IBM Field Engineering

Theory of Operation

2860 Selector Channel

SY27-2220-1

PREFACE

This manual describes the theory of operation of the 2860 Selector Channel. It is assumed that the reader has a knowledge of ALD interpretation and of the basic circuits used in the 2860.

This manual is divided into six chapters and two appendices. Chapter 1, "Introduction", describes the 2860 Selector Channel data control and flow, stressing conceptual and overall objectives. Chapter 2, "Functional Units", describes individual logic units of the 2860. Chapter 3, "Principles of Operation", presents a detailed analysis of 2860 operations. Chapter 4, "Features", describes the existing features of the 2860. Chapter 5, "Power Supplies and Control", describes the power distribution and control within the 2860 and the power control interface with the other units of the system. Chapter 6, "Console and Maintenance Features", describes the 2860 CE control panel and associated test logic, as well as the logout, diagnostic, error checking, and status byte maintenance capabilities of the 2860.

Appendix A, "Unit Characteristics", gives the characteristics of the 2860. Appendix B, "Special Circuits" describes the special circuits of the 2860. Appendix C, "ALD Contents by Prefix", lists the 2860 ALD's by prefix and the 2860 function(s) contained in the ALD's.

Following most paragraph heads are bullets (key statements preceded by •) which summarize significant points about the subject. The bullets serve two functions: (1) they provide the customer engineer (CE) with the key points of the topic, and (2) they provide quick reference for review and recall for the CE who is familiar with the machine. The more detailed text following the bullets provide the nonclassroom student with the material necessary for self- instruction.

The diagrams supporting the text are divided into two groups: (1) purely instructional diagrams, and (2) maintenance-oriented diagrams. Examples of the first group are high-level block diagrams and diagrams that show general data flow and timing considerations. These diagrams are generally not affected by engineering changes, and, if they include AND/OR logic blocks, the blocks are drawn in positive logic convention and do not maintain exact ALD line names. The instruction diagrams, which are contained in this manual, are numbered consecutively within a chapter (for example, 1-1 is the first figure in Chapter 1,

Second Edition (August 1974)
This is a reprint of SY27-2220-0 incorporating changes released in Technical Newsletter SY22-6927 dated March 6, 1973.

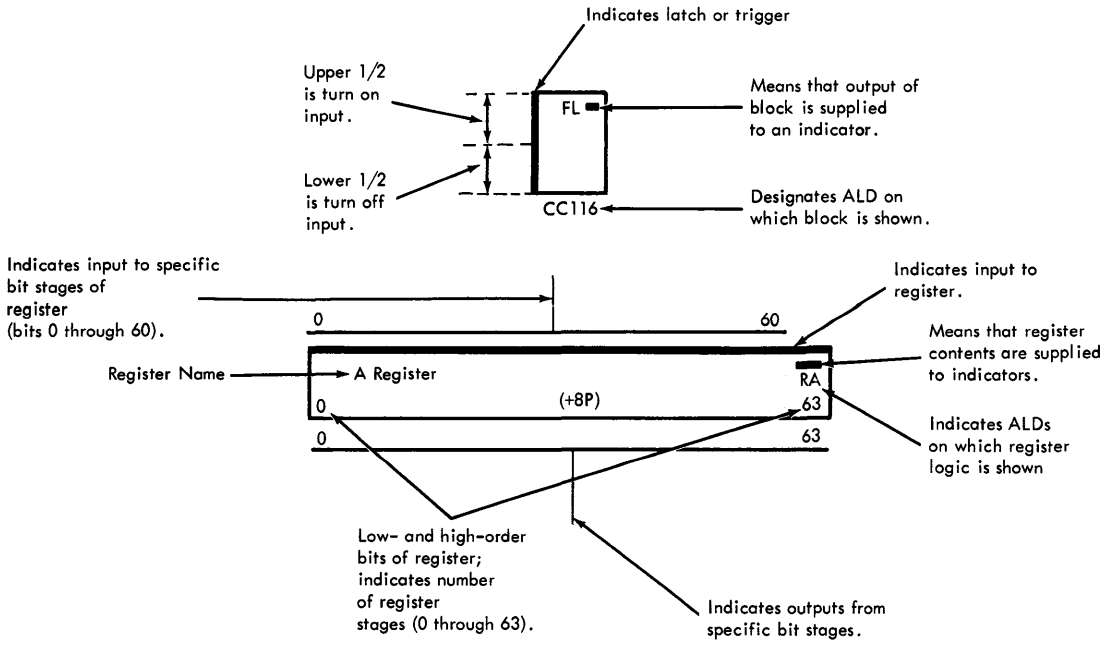
Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

This manual has been prepared by the IBM System Products Division, Product Publications, Dept. B97, PO Box 390, Poughkeepsie, N.Y. 12602. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be sent to the above address. Comments become the property of IBM.

©Copyright International Business Machines Corporation 1968

2-7 is the seventh figure of Chapter 2). The diagrams of the second group are referenced in this manual (e.g., Diagram 3-1, FEMDM) but are placed in the 2860 FE Maintenance Diagram Manual to allow ready reference during maintenance and to facilitate updating the diagrams to new engineering levels. These diagrams are grouped by categories; each category may contain different types of diagrams (e.g., Category 4 contains diagrams designated Diagrams 4-1, 4-2, etc., which pertain to functional units of the 2860).

The symbols and references on all diagrams are standardized to provide easy identification of the diagrammed blocks. Symbols and references that may not be obvious to the reader are:



The following related manuals may be used in conjunction with this manual:

FE Maintenance Diagrams Manual, *IBM 2860 Selector Channel,* SY27-2221

FE Maintenance Manual, *IBM 2860 Selector Channel,* SY22-2893

FE Theory of Operation:

IBM Power Supplies–SLT, SLD, ASLT, MST, SY22-2799

IBM SLT Technology–Packaging, Tools, Wiring Change Procedure, SY22-2800

Illustrated Parts Catalog, *IBM 2860 Selector Channel,* S123-0426

Maintenance Library Installation Manual, *IBM System/360 and System/370, 2860 Selector Channel and 2870 Multiplexer Channel,* SY22-9510

Reference manuals:

IBM System/360 Principles of Operation, GA22-6821

IBM System/370 Principles of Operation, GA22-7000

IBM System/360 and System/370 I/O Interface Channel to Control Unit, Original Equipment Manufacturers' Information, GA22-6974

FETOM, *IBM 2860 Storage Channel,* SY22-2906

FEDM, *IBM 2860 Storage Channel,* SY22-6607

FEMM, *IBM 2860 Storage Channel,* SY22-2907

CONTENTS

ILLUSTRATIONS

TABLES

ac alternating current
Addr address
ADR address
ADR IN address in
ADR O address out
A FUL A-register full
amp ampere(s)
ASR auxiliary storage request
ATTN attention
Auto automatic
Aux auxiliary

B FUL B-register full
BC byte counter
BCL byte count latches
BCR byte count register
BCU bus control unit
BI bus in
BO bus out
BTU British thermal unit

CA command address
CAR command address register
CAW channel address word
CB circuit breaker
CC chain command
CC condition code
CCA channel-to-channel adapter
CCU channel controller unit
CCW channel control word
CCWF channel control word fetch
CCWV channel control word valid
CD chain data
CDA chain data
CE customer engineer
CH E channel end bit
CHAN channel
Chk check
com common
COM O command out
CPU central processor unit
CSW channel status word
CT count
CTB count boundary
CTL channel control check
CU control unit
CUE control unit end bit

DA data address
DAB data address boundary
DAR data address register
dc direct current
DEV E device end bit
DIAG diagnostic simulate interface

EC engineering change
EOT end of test
EPO emergency power off

FEDM Field Engineering Diagram Manual
FEMDM Field Engineering Maintenance Diagram Manual
FEMI Field Engineering Manual of Instruction
FEMM Field Engineering Maintenance Manual
FETOM Field Engineering Theory of Operation Manual
FLR flag register
FLT fault location test
ft feet

Gt BC and CT-1 IL Gate byte count and count minus 1, incorrect length (to adder)

H I/O halt I/O
hex hexadecimal
HSDA high-speed direct access

I CTL interface control check
IC instruction counter
ID identity
IL incorrect length
in. inches
IND indicator
INTF DSBLD interface disabled
I/O input/output
IPL initial program load
IRG inter-record gap
IRPT interrupt

KVA kilo-volt amperes

lbs pounds
LCS large capacity storage
Ldd loaded
LS WD last word trigger
LWT last word trigger

M multiplex
M margin (Voltage)
Mach machine
Max. maximum
MC marginal check
MCW maintenance control word
MDBO memory data bus out
min. minimum
MKB mark B
MOD modifier bit
MPS modular power supplies

NO SEL no selection
ns nanoseconds

OE OR exclusive
OP IN operational in
OP O operational out

P parity
P IRPT polling interrupt

PCI program controlled interrupt
PFCU parallel file control unit
PN part number
pos position
PROG program check
PROG LD FLT program load-fault location test
PROG LD IPL program load-initial program load
PROT protection (storage) check
PSCE peripheral storage control element
PSW program status word

Rcvr receiver
RD read
reg register
Rel Humidity relative humidity
RET S retain storage
RST restart

S simplex
S CHK storage check
S I/O start I/O
SAB storage address bus
SBO storage bus out
SBI storage bus in
SEL O select out
SILI suppress incorrect length indication
SIM simulate
SLI suppress (incorrect) length indication
SPR storage protect register
SRL systems reference library
SS singleshot
ST CY storage cycle
STA IN status in
STOR store
SU setup
SVC O service out

T0-T7 time 0 through time 7
T I CH transfer in channel latch
TB terminal board
TIC transfer in channel
T CY transfer in channel cycle latch
T I/O test I/O

U CHK unit check bit
U EXC unit exception bit
UAB unit address bus
UABO unit address bus out
UABI unit address bus in
UAR unit address register
usec microseconds

v volts
V ac volts alternating current
V dc volts direct current

WLR wrong length record
WR write

Z address channel status word address 64 decimal
Z + 1 address channel address word address 72 decimal

=CT byte count latches equal count B
=ZRT byte count latches equal zero

This chapter describes the overall function and capabilities of the 2860 Selector Channel. Included in this chapter is a general description of the system application of the unit, general unit capabilities, I/O configurations, and operations performed by the channel.

PURPOSE

The 2860 Selector Channel (channel) directs the flow of information between I/O devices and main storage, thus relieving the processor of the task of communicating directly with the I/O devices. Channel operation is such that once an I/O operation is successfully initiated, the processor is released (in most cases), and may perform data processing operations concurrent with channel data transfer operations.

I/O operations and, consequently, channel operations are initiated by the processor. When the channel is selected to control an I/O operation, it performs initial operations to determine whether both the channel and the addressed I/O device are available to perform the operation. If either the channel or I/O device is unavailable, the channel relays information to the processor indicating the reason why the operation cannot be performed. Assuming both the channel and the I/O device are available, the channel selects the I/O device and releases the processor. Once the processor is released, the channel independently obtains commands and controls the flow of data to and from the selected I/O device until the operation is complete.

Note: For the initial program load operation and a portion of the fault location test operations, the channel is not released until required data has been transferred. When the operation is complete, the channel notifies the processor and, when instructed by the processor, stores information about the condition (status) of both the channel and I/O device at the end of the operation, and the extent of the data transfer operation.

Channel operations may be halted by the processor at any point in an operation previously initiated by the processor. In addition, the channel or an I/O device serviced by the channel may be tested by the processor to determine the operating status or condition of these units at the time the test is initiated.

DESCRIPTION

The following paragraphs describe the channel in terms of use with different system configurations, the different models available, physical characteristics, and interface lines.

System Configurations

The different system configurations in which the 2860 Selector Channel is used are illustrated in Figure 1-1. When used in the System/360 Model 65, Model 67-1, and Model 75, the Selector Channel interfaces with the processor and main storage units and with up to 8 I/O control units (see Figure 1-1, a). The basic version of the channel (version 000) is used with Models 65 and 67-1, while version 002 of the channel is used with Model 75. The difference between the two versions is the active state polarities of the CPU and BCU interface lines, which are positive on Models 65 and 67 and negative on Model 75.

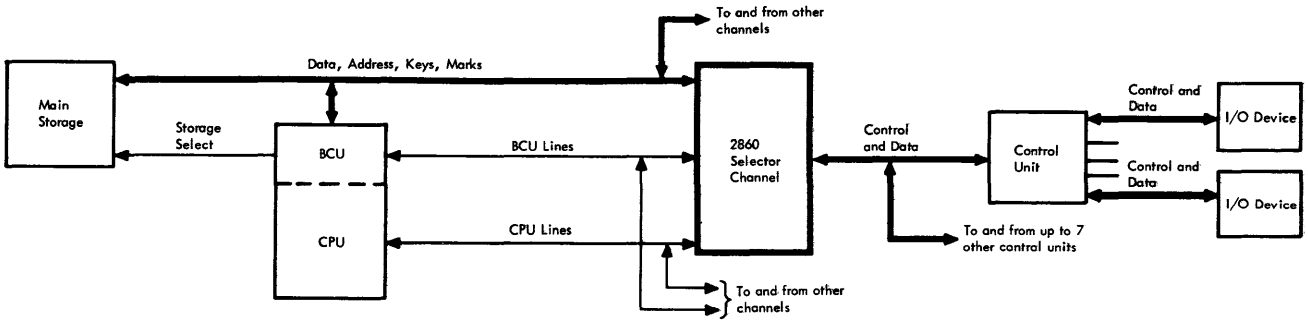
In System/360 Model 67-2, the channel interfaces with the 2846 Channel Controller rather than a processor, and versions 003 or 004 (positive active-state polarities) of the channel are used (see Figure 1-1, b). In System/360 Model 91, the channel interfaces with the Peripheral Storage Control Element (PSCE) rather than a processor (see Figure 1-1, c). In this case, version 005 of the Selector Channel is used. Version 005 differs from the basic version in that some interface lines are added and delted to accommodate the Model 91 requirements, and the active-state polarities of the PSCE/channel lines are negative.

Models and Physical Description

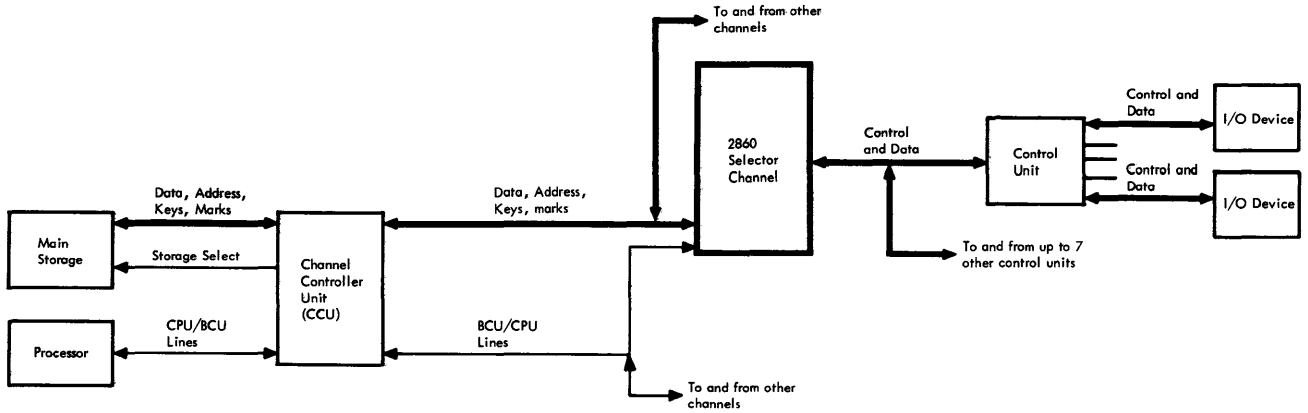
Three models of the 2860 Selector Channel are available:

Model 1 – Frame with one channel.
Model 2 – Frame with two channels.
Model 3 – Frame with three channels.

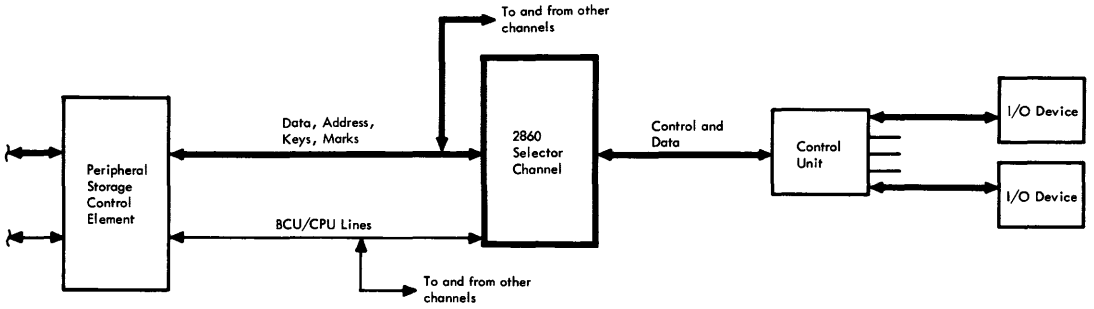
Model 1 contains what is referred to as the basic channel, and Models 2 and 3 contain this basic channel plus additional channels as indicated. For a physical representation of the 2860 Selector Channel, refer to the FEMDM for the 2860.



a. Selector Channel System Configuration for Model 65, 67-1 and 75 Systems.



b. Selector Channel System Configuration for Model 67-1.



c. Selector Channel System Configuration for Model 91.

Figure 1-1. 2860 Selector Channel System Configurations

The CE panel on each channel allows the CE to simulate or manually control operations. Switching the channel to test mode (at the CE panel) forces the channel off-line to the CPU and places the channel to a simulate CPU condition. Most operations normally initiated by the CPU can then be manually initiated at the CE panel. When the channel is in test mode, the CE panel switches may be set (if desired) to permit the channel to simulate main storage or to simulate an I/O device on the I/O interface. For simulate main storage operations, the channel does not have access to main storage. For simulate I/O interface operations, the channel does not have access to attached I/O devices. During test-mode operations, the simulate storage and simulate I/O interface functions may be used separately, simultaneously, or not at all.

The power and marginal checking controls are mounted on the front of the channel frame. Power on/off operations and marginal checking can be performed on a frame basis only; i.e., power operations and marginal checking cannot be performed on an individual channel basis.

Channel Interfaces

- CPU, BCU, I/O, and power interfaces defined at channel.
- Interface connects unit electrically.
- Simplex and multiplex lines are used.

For the 2860 Selector Channel, four interfaces are defined at the channel: the CPU interface, the BCU interface, the I/O interface, and the power interface. These interfaces electrically connect the channel to other units, to a gate within the 2860 frame, or to logic areas within the individual channel. The CPU interface connects the channel to the CPU (Models 65, 67-1, or 75) or an equivalent unit (Channel Control Unit for Model 67-1 or Periferal Storage Control Element for Model 91.) The BCU interface connects the channel to the BCU (part of the CPU) and main storage for Models 65, 67-1, and 75, or equivalent unit (Channel Controller Unit for the Model 67-2 or Periferal Storage Control Element for the Model 91). The standard I/O interface connects the channel to the attached control units. The power interface connects the channels (on a frame basis) to the interlocking system power control logic.

Note: The 2860 CPU, BCU, and I/O interfaces may interface with other units than those described; when this is the case, refer to Chapter 4, "Features".

All interface connections are made with pluggable cable connectors. The wires within the cables are referred to as interface lines.

Two types of interface lines are used at the channel interfaces: simplex and multiplex. (See the example of simplex and multiplex lines in Figure 1-2.) A simplex line connects one unit to only one other unit. A multiplex line connects one unit to more than one other unit. A multiplex line, though used by more than one unit, is used by only one unit at a given time. The CPU, BCU and I/O interfaces and the lines associated with each are described in the following text.

CPU Interface

The CPU interface (defined at the channel) with its associated signal lines can be considered in three groups (Figure 1-3): (1) CPU operation lines provide the channel with control signals needed for normal program operation; (2) diagnostic lines are used by the processor to assume diagnostic control of the channel; (3) fault locating test (FLT) lines control the channel when the processor performs fault locating tests. Field engineering aid signals exist at probes near the CPU interface connector to aid the customer engineer in locating faults in the channel.

CPU Operation Lines. The operation lines provide the control signals required for normal channel instruction operations. A description of these lines and signals follows:

1. Unit Address Bus Out (UABO) – consists of nine multiplex lines (eight data, one parity) to the CPU interface and receivers in each channel. The UABO bits must be valid (active) a minimum of 75 ns before the channel receives the 'select channel' signal and must stay active until the 'select channel' signal is dropped. The UABO bits provide the unit address of the I/O device specified by a Start I/O, Halt I/O, or Test I/O instruction, or by an FLT or IPL operation.
2. Unit Address Bus In (UABI) – consists of nine multiplex lines (eight data, one parity) driven from each channel CPU interface. The UABI bits are guaranteed valid at the CPU interface when the channel sends a 'release' signal to the CPU interface in reply to an 'interrupt response' signal. The UABI bits are valid for the duration of the 'release' signal.
3. Select Channel – a simplex signal to the CPU interface at each channel. The 'select channel' signal is activated to initiate an IPL operation or an I/O instruction at a specific channel. The 'select channel' signal rises 75 ns (minimum) after the UABO bits and IPL or instruction signals are active. The 'select channel' signal must stay active until the channel sends a 'release' signal to the CPU interface.
4. Start I/O – one multiplex line received at all channel CPU interfaces. The 'start I/O' signal becomes active at least 75 ns before the 'select channel' signal, and must stay active until after the 'release' signal is generated by the channel. The 'start I/O' signal indicates that the channel is to execute a start I/O operation.

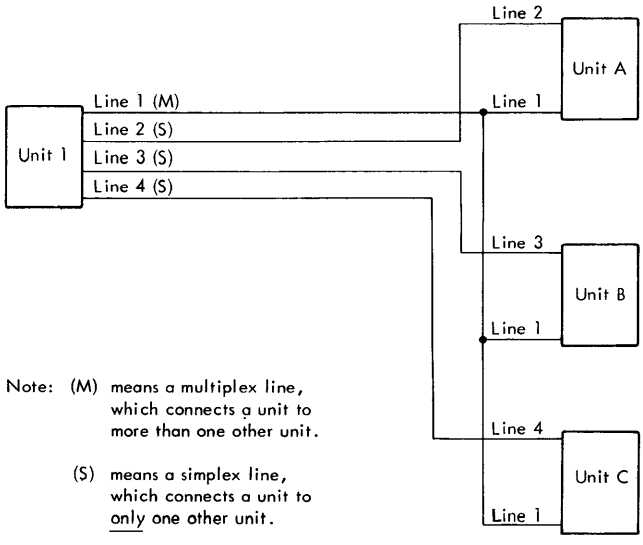


Figure 1-2. Multiplex and Simplex Lines

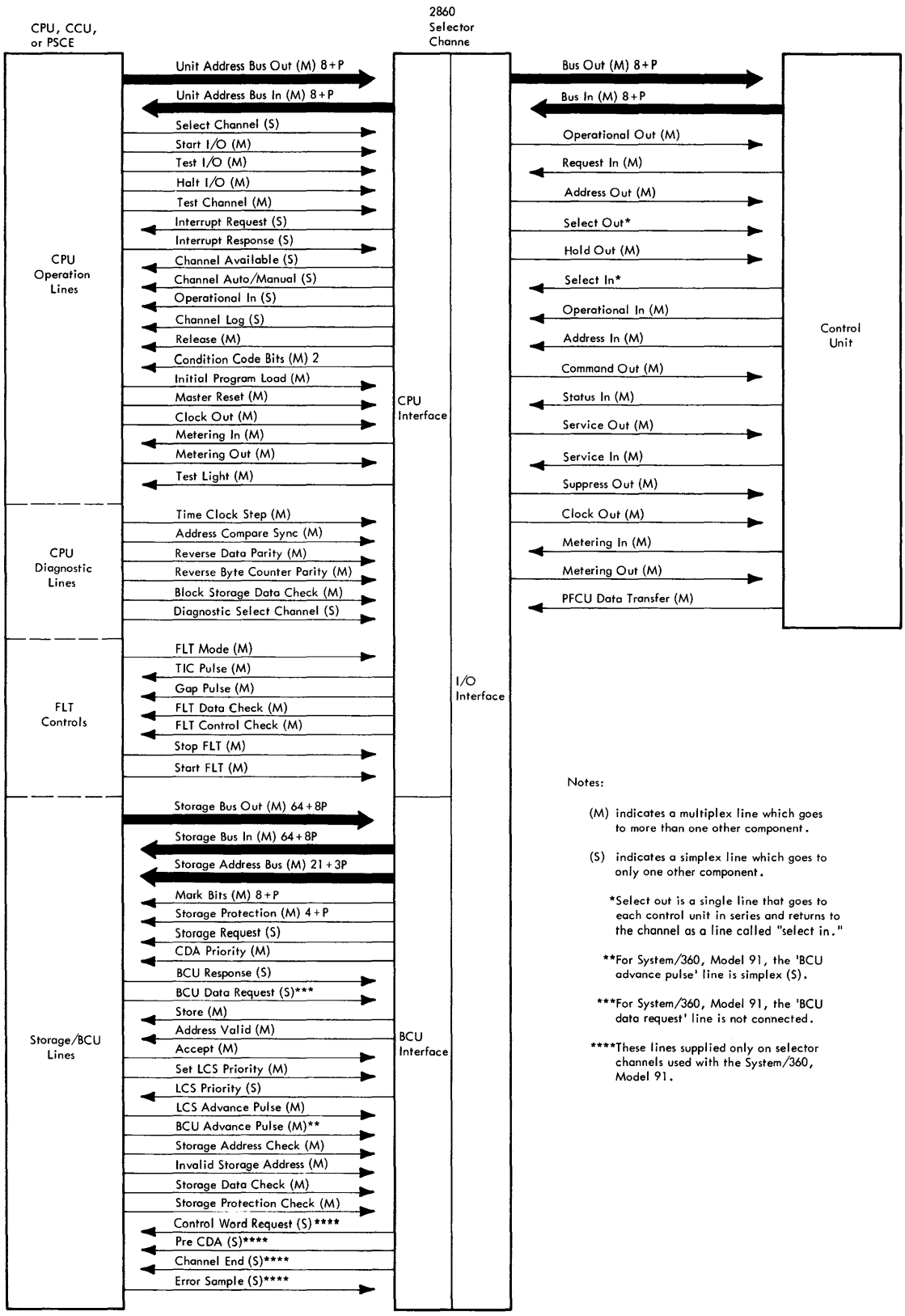


Figure 1-3. Selector Channel Signal Interface Lines

5. Test I/O – one multiplex line received at all channel CPU interfaces. The 'test I/O' signal becomes active at least 75 ns before the 'select channel' signal, and must stay active until after the 'release' signal is activated by the channel. The 'test I/O' signal indicates that the channel is to execute a test I/O operation.
6. Halt I/O – one multiplex line received at all channel CPU interfaces. The 'halt I/O' signal becomes active at least 75 ns before the 'select channel' signal, and must stay up until after the channel activates the 'release' signal. The 'halt I/O' signal indicates that the channel is to execute a halt I/O operation.
7. Test Channel – one multiplex line received at all channel CPU interfaces. The 'test channel' signal becomes active at least 75 ns before the 'select channel' signal, and must stay active until after the 'release' signal is activated by the channel. The 'test channel' signal indicates that the channel is to execute a test channel operation.
8. Interrupt Request – a simplex line from the channel to the CPU interface. The 'interrupt request' signal is activated when an 'interrupt' condition is recognized by the channel and stays active until the channel either receives an 'interrupt response' signal at the CPU interface or until the channel is selected for a test I/O operation. If the 'interrupt request' signal is caused by a polling interrupt, the channel can be selected for a halt I/O, start I/O or test I/O operation, any of which will cause the 'interrupt request' signal to be deactivated. Receipt by the channel of an 'interrupt response' signal also deactivates the 'interrupt response' signal.
9. Interrupt Response – a simplex line to each channel CPU interface. The 'interrupt response' signal is in reply to an 'interrupt request' signal from the channel and allows the channel to store a channel status word (CSW) at byte location 64. The 'interrupt response' signal stays active until the 'release' signal is activated by the channel.
10. Channel Available – a simplex line from each channel to the CPU interface. The 'channel available' signal is active when the channel is on line. The line is inactive if the channel power is down, if the channel is in test mode or is meter disabled, or if the channel is not attached to the system.
11. Channel Auto/Manual – a simplex line from each channel to the CPU interface that lights an indicator on the CE console (in those systems requiring this indication) when the channel is in automatic mode.
12. Operational In - a simplex line from each channel to the CPU interface that lights an indicator on the CE console (in those systems requiring this indication) when an I/O device activates the 'operational in' signal on the I/O interface.
13. Channel Log – a simplex line from each channel to the CPU interface. The 'channel log' signal lights an indicator on the CE console (in those systems requiring the indication) when the channel LOG ON MACH CHK (log on machine check) switch is in the down position and the channel is not in test mode.
14. Release – a multiplex line from each channel to the CPU interface. The 'release' signal terminates the communication between the CPU (or equivalent unit) and the channel for the following three conditions:
    a. A 'release' signal for an instruction operation indicates to the CPU (or equivalent unit) that the condition codes are valid and that the instruction operation may be terminated.
    b. A 'release' signal in reply to an 'interrupt response' signal from the CPU interface indicates that the channel has completed storing the CSW into main storage beginning at location 64 (decimal) and has cleared the interrupt condition.
    c. A 'release' signal for an IPL operation indicates that the IPL operation has been successfully completed. After receipt of the 'release' signal the CPU is free to access main storage address 0 to obtain the initial PSW.
15. Condition Code Bits – two multiplex lines from each channel to the CPU interface. The condition code bits indicate the results of an operation performed by the channel. The condition code bits are valid at the time the channel activates the 'release' signal.
16. Initial Program Load (IPL) – one multiplex line received at all channel CPU interfaces. The 'IPL' signal causes each channel to do a general reset and the selected channel to perform the IPL operation from the I/O device specified by the UABO bits. The 'IPL' signal is 180 ns (minimum) in duration and must arrive at the channel after the 'select channel' signal is received. The 'IPL' signal must overlap the 'select channel' signal for at least 180 ns.
17. System Reset – a multiplex line to all channel CPU interfaces. The 'system reset' signal is 2.2 ± 1.8 usec in duration and causes a complete channel, control unit, and I/O device reset operation. All control units and I/O devices are reset except those devices that are performing an operation independent of the control unit, such as a tape unit in rewind.
18. Clock Out (not CPU Stopped) – a multiplex line to all channel CPU interfaces. The 'clock out' signal indicates that the CPU is not in a halt or wait state. The channel may not change the enable or disable meter conditions while the 'clock out' signal is active at the channel.
19. Metering In (Intermediate Unit Run CPU Clock) – a multiplex line from all channels to the CPU interface. The 'metering in' signal forces the CPU clock to run as long as the signal is present.

20. Metering Out (CPU Clock Running) – a multiplex line to all channel CPU interfaces. The 'metering out' signal is active any time the customer meter is running.
21. Test Light (Turn On Test Indicator) – a multiplex line from all channels to the CPU interface. The 'test light' signal is active any time that a channel 'log on machine check' (LOG ON MACH CHK) switch is in the up position or the channel is in test mode.

Diagnostic Lines. The diagnostic lines may be used by the CPU to determine the validity of the channel checking functions:

Note: The diagnostic lines are not connected for those systems without diagnostic capabilities.

1. Reverse Data Parity – a multiplex line to the channel CPU interface that is activated by decoding the maintenance control word (MCW) in the CPU. The 'reverse data parity' signal causes a reversal of the parity bit from the simulate I/O interface register and blocks the activation of the 'bus in parity check' signal. The 'reverse data parity' signal remains active until the MCW is changed by a diagnose instruction in the CPU.
2. Reverse Byte Counter Parity – a multiplex line to the channel CPU interface that causes the byte counter parity to be wrong after the byte counter is updated in the selected channel. The signal is under the direct control of the MCW in the CPU and remains active until the MCW is changed by a diagnose instruction.
3. Block Storage Data Check – a multiplex line to the channel CPU interface that causes the selected channel to block turn on of the channels 'channel data check' or 'channel control check' latches. The 'block storage data check' signal allows invalid CCW's to be brought into the channel to test sections of the channel checking circuitry. The 'block storage data check' signal is under the direct control of the MCW in the CPU and remains active until the MCW is changed by a new diagnose instruction.
4. Diagnostic Select Channel – a simplex line to each channel CPU interface. The 'diagnostic select channel' signal enables the other three diagnostic control signals to be used, and the specified channel forces a simulate I/O interface condition within the channel. The 'diagnostic select channel' signal is under direct control of the MCW in the CPU and remains active until the MCW is changed by a new diagnose instruction.

   Whenever the 'diagnostic select channel', 'block storage data check', 'reverse data parity', and 'reverse byte counter parity' signals are active simultaneously, the diagnostic reverse-byte counter parity function is inhibited and channel adder parity checks are blocked.
5. Time Clock Step – a multiplex line to the channel CPU interfaces that provides a controlling gate for the auto-restart maintenance function (initiated at the channel CE panel in the test mode). The 'time clock step' signal is generated by the interval timer in the processor and is 200 ns (minimum) in duration with a 16.7-ms period on 60 Hz machines (20-ms period for 50 Hz).
6. Address Compare Sync – a multiplex line from the CPU interface. The 'address compare sync' signal is generated at the processor and controlled at the system console. A storage address sent to the BCU (or CCU or PSCE) is compared with a configuration set in the console address keys. The 'address compare sync' signal is generated and sent to all attached channels if the addresses match. The 'address compare sync' signal is primarily a scoping aid.

FLT Control Lines. The FLT control lines provide the communication required between the CPU interface and the channel for the channel to control the storage of FLT's under control of the CPU.

1. FLT Mode – a multiplex line to all channel CPU interfaces. The 'FLT mode' signal is active as long as the FLT operation is in progress.
2. TIC Pulse – a multiplex line from all channels to the CPU interface. The 'TIC pulse' signal duration is 350 ± 30 ns, starting when the first TIC command is executed after the 'IPL' latch is reset. The 'TIC pulse' signal indicates that one of the two FLT storage areas (buffers) is filled with FLT tests.
3. Gap Pulse – a multiplex line from all channels to the CPU interface. This signal is active for 1 ms (minimum) whenever an end-of-block condition is recognized by the I/O device providing the FLT's. The 'gap pulse' signal is not generated if an error is detected in a data block of FLT tests.
4. FLT Data Check – a multiplex line from all channels to the CPU interface. The 'FLT data check' signal indicates that a data error has been detected. The 'FLT data check' signal is active from the time the error is detected until the 'start FLT' signal is received by the channel.
5. FLT Control Check – a multiplex line from all channels to the CPU interface. The 'FLT control check' signal is activated when a control error, command reject, or exceptional condition situation is detected. The 'FLT control check' signal indicates that the channel is unable to proceed with the FLT operation.
6. Stop FLT – a multiplex line to all channel CPU interfaces. The 250 ± 50-ns 'stop FLT' signal commands the operating channel to stop transmission of FLT tests to main storage and wait. The I/O device providing the FLT's proceeds to the end of the current block and disconnects from the I/O interface.
7. Start FLT – a multiplex line to all channel CPU interfaces. The 250 ± 50-ns 'start FLT' signal commands the

selected channel to restart data transmission to main storage. The channel fetches the command at storage location 128 (decimal) and continues loading FLT tests. The channel always restarts loading FLT tests from the beginning of the data block being transmitted at the time the 'stop FLT' was received. If the 'stop FLT' signal is received during an error backspace by the I/O device, the backspace operation is completed. A subsequent 'start FLT' signal causes the channel to fetch a read command CCW that specifies load FLT tests into the first FLT main storage location.

BCU Interface

The BCU (bus control unit) interface (defined at the channel) carries the data and control information necessary to the channel when accessing main storage. Signals entering and leaving the channel via the BCU interface enable the channel to receive commands and transfer data doublewords to or from main storage. For System/360 Models 65, 67-1, and 75, the channel shares the same bus control unit used by the CPU to each storage. In these models, control of CPU/channel priority is a function of the bus control unit. For Model 67-2, the Channel Controller Unit assumes the bus control unit functions, while in Model 91, the Peripheral Storage Control Unit assumes the bus control unit functions.

1. Storage Bus Out (SBO) – consists of 72 multiplex lines (64 data, 8 parity) to each attached channel's BCU interface. The SBO signals become active after the 'BCU response' and 'accept' signals deactivate and 200 ± 40 ns after the 'BCU advance pulse' signal is activated. The SBO signals are active for a minimum of 300 ns and can be gated either to control registers or to the A-register in the channel.
2. Storage Bus In (SBI) – consists of 72 multiplex lines (64 data, 8 parity) from each attached channel's BCU interface. The SBI signals become active when the channel receives the 'BCU data request' signal and deactivate when 'BCU data request' signal is deactivated.
3. Storage Address Bus (SAB) – consists of 24 multiplex lines (21 data, 3 parity) from each attached channel's BCU interface. The SAB signals are active for the duration of the 'BCU response' signal at the channel. The SAB bits specify the doubleword location in main storage to be accessed by the channel.
4. Mark Lines – consist of nine multiplex lines (eight data, one parity) from each attached channel's BCU interface. The mark bits indicate which of the eight bytes of a doubleword are to be stored on a channel store operation. The mark bits are active as long as the 'BCU response' signal is active at the channel.
5. Storage Protection Lines – consist of five multiplex lines (four data, one parity) from each attached channel's BCU interface. The 'storage protection' lines carry the storage protection key which allows the channel to access protected areas of storage, both on store and fetch operations. The storage protection bits are active as long as the 'BCU response' signal is active at the channel.
6. Storage Request – a simplex line from each attached channel's BCU interface. The 'storage request' signal requests priority for a storage cycle. The 'storage request' signal is deactivated when the channel receives the 'BCU data request' signal.
7. BCU Response – a simplex line to channel's BCU interface. The 'BCU response' signal is activated when storage priority is assigned to a 'storage request' signal from channel and an address is required on the SAB lines. The 'BCU response' signal remains active as long as an address is required on the SAB. The 'BCU response' signal must stay active for a minimum of 500 ns.
8. BCU Data Request – a simplex line to the channel's BCU interface. The 'BCU data request' signal rises and stays up as long as data is required on the SBI lines. The 'BCU data request' signal rises for both a store and a fetch operation, even though data is gated to the SBI lines only during a store operation. The 'BCU data request' signal must overlap the 'accept' signal for at least 140 ns prior to receipt of the 'BCU advance pulse' signal.

   Note: For 2860 Selector Channels used with the System/360, Model 91, the channel generates the 'BCU data request' signal by delaying the 'BCU response' signal approximately 150 ns.

9. Store – a multiplex line from each attached channel's BCU interface. The 'store' signal is active as long as the 'BCU response' signal is active at the channel and indicates that the channel is engaged in a store operation.
10. Address Valid – a multiplex line to the channel's BCU interface. The 'address valid' signal is activated when the address on the SAB is valid (stable). The 'address valid' signal is delayed from the rise of the 'BCU response' signal long enough to provide a deskew delay for the address on the SAB. The 'address valid' signal is deactivated when the 'BCU response' signal is deactivated.
11. Accept – a multiplex line to the channel's BCU interface. The 'accept' signal is activated at least 140 ns before the 'BCU data request' signal drops. The 'accept' signal notifies the channel that originated the 'storage request' signal that the storage cycle is started.

The next 'BCU advance pulse' signal pertains to that 'storage request' signal. The 'accept' signal duration is 200 ± 50 ns with a maximum of 50 ns overlapping the 'BCU advance pulse' signal.

12. Set LCS (Large Capacity Storage) Priority – a multiplex line to the channel's BCU interface. The 'set LCS priority' signal is activated when the LCS address is on the SAB. The requesting channel turns on the 'LCS priority' latch and waits for an 'LCS advance pulse' signal. The 'BCU advance pulse' signal is ignored. The 'set LCS priority' signal duration is 500 ns (maximum) with a minimum of 140 ns overlapped by the 'BCU data request' signal. (The 'BCU data request' signal is not supplied for the Model 91).
13. LCS Priority – a simplex line from each attached channel's BCU interface. The 'LCS priority' signal is activated when the channel receives a 'set LCS priority' signal. The 'LCS priority' signal indicates that the channel is requesting a storage location in an LCS. The 'LCS priority' signal falls when the 'BCU advance pulse' signal is activated.
14. LCS Advance Pulse – a multiplex line to the channel's BCU interface. The 240 ± 60-ns 'LCS advance pulse' signal arrives at the channel 200 ± 40 ns before the SBO bits become valid and indicates that the channel can prepare to gate data into the channel from the LCS.
15. BCU Advance Pulse – a multiplex line (simplex for Model 91) to the channel's BCU interface. The 240 ± 60-ns 'BCU advance pulse' signal arrives at the channel 200 ± 40 ns before the SBO bits become valid and indicates that the channel can prepare to gate data from storage into the channel.
16. Storage Address Check – a multiplex line to the channel's BCU interface that indicates a parity error was detected in the SAB bits, storage protect key bits, or the mark bits. The 'storage address check' signal must overlap the 'BCU advance pulse' signal for 150 ns (minimum).
17. Invalid Storage Address – a muliplex line to the channel's BCU interface that is activated when the address on the SAB refers to a non-existent storage location. The 'invalid storage address' signal must overlap the 'BCU advance pulse' signal for 150 ns (minimum).
18. Storage Data Check – a multiplex line to the channel's BCU interface. The 'storage data check' signal is activated when a data parity error is detected in information sent to main storage on a store operation, or in information coming from main storage on a fetch operation. The 130 ns (minimum) 'storage data check' signal rises between 100 and 200 ns after data normally appears on the SBO lines.
19. Storage Protection Check – a multiplex line to the channel's BCU interface. The 'storage protection check' signal indicates that an attempt was made by the channel to access a protected storage area with an improper storage protect key. The 'storage protection check' signal must overlap the 'BCU advance pulse' signal for 150 ns (minimum).
20. Control Word Request (System/360, Model 91, only) – a simplex line from the 2860 Selector Channel (BCU interface) to the PSCE which identifies the current 'storage request' signal as a request for a channel address word (CAW) or a channel command word. The 'control word request' signal rises after the channel 'storage request' signal and falls when the 'storage request' signal falls.
21. Pre CDA (System/360, Model 91, only) – a simplex line from the 2860 Selector Channel (BCU interface) to the PSCE. During a write CDA operation, this line will rise with the next to last channel 'storage request' signal, and fall when the 'control word request' signal is raised.
22. Channel End (System/360, Model 91, only) – a simplex line from the 2860 Selector Channel (BCU interface) to the PSCE. This signal rises when the channel terminates an operation and raises the 'interrupt request' signal. The 'channel end' signal has a duration of approximately 150 ns.
23. Error Sample (System/360, Model 91, only) a simplex line from the PSCE to the 2860 Selector Channel (BCU interface). This signal is used to sample channel errors during store operations. The channel 'interrupt request' signal or chain command operation will be suppressed until the last 'error sample' (for data) signal has been received by the channel. The duration of the 'error sample' signal is approximately 150 ns.

I/O Interface

The connection between the channel and the attached control units is designated the I/O interface (Figure 1-3). The I/O interface information format and signal sequences are common to all control units. The interface consists of a set of lines that connect a number of control units to a channel. Up to eight control units can be connected to a single channel. Except for the select out line, all communications between the channel and the I/O units occur over multiplex lines. Only one control unit at any one time can be electrically connected to the channel. A control unit remains electrically connected on the interface throughout a complete data transfer sequence or until the channel initiates a disconnect operation.

The I/O interface can accommodate up to 256 directly addressable I/O devices. The actual number of control units and devices that can be used is limited only by data rate, electrical power, and address considerations.

The rise and fall of all signals transmitted over the interface is controlled by interlocked signal responses between

the channel and control unit. This interlocking removes the dependence of the interface on circuit speed and allows application to a wide variety of circuits and data rates. Interlocking also permits the connection of control units with different circuit speeds to a single channel. Definitions of the I/O interface lines follow.

Buses. Each of two buses (designated bus in and bus out) is a set of nine multiplex lines (eight data, one parity) between all attached control units and the channel. Odd parity must be maintained on the buses for all transfer operations.

1. Bus-Out Lines - used to transmit address bytes, command bytes, control order bytes, and data bytes to the control units. The type of information transmitted over the bus-out lines is indicated by the active signal to the control units. The period during which information on the bus-out lines is valid is controlled by the duration of the active signal.
2. Bus-In Lines – used to transmit address bytes, status bytes, sense information, and data bytes to the channel. A control unit can place and maintain information on the bus-in lines only when its 'operational in' signal is active, except in the case of the control unit busy sequence. The type of information transmitted over the bus-in lines is indicated by the active signal from the control unit. The period during which information on the bus-in lines is valid is controlled for the duration of the active signal.

Selection Control Lines. The 'selection control' signals are used to select or reset I/O devices attached to the control units. A description of the selection control lines follows:

1. Operational Out – a multiplex line from the channel to all attached control units used for interlocking purposes. Except for the 'suppress out' signal, all signals from the channel are significant only when the 'operational out' signal is active. Whenever the 'operational out' signal is inactive, all lines from the control unit to the channel drop and any operation currently in process over the I/O interface is reset.
2. Select Out – a line from the channel to the highest-priority control unit and from any control unit to the next lower control unit in priority. The 'select out' signal, together with the 'select in' signal, forms a loop for polling all attached control units. A control unit can raise the 'operational in' signal only when a 'select out' signal is received. A control unit that does not require selection propagates the 'select out' signal to the next control unit. A control unit that propagates the 'select out' signal cannot raise the 'operational in' signal or respond with a control unit busy sequence until the channel again activates the 'select out' signal.
3. Hold Out – a multiplex line from the channel to all attached control units that is used to enable the 'select out' signal at the control units. The 'select out' signal is considered active at the control unit only when the 'hold out' signal is active. The 'hold out' signal rises when the 'select out' signal rises at the channel. The 'hold out' signal is active at the last attached control unit long before the 'select out' signal is propagated through the intervening control units. When the 'select out' and 'hold out' signals drop at the channel, the fall of the 'select out' signal must be propagated, but the 'hold out' signal drops almost simultaneously at all attached control units; thus, no time is lost waiting for the 'select out' signal to drop at all control units serially.
4. Select In – a line from the driver in the last attached control unit to the channel. The 'select in' signal is actually the 'select out' signal after propagation of the 'select out' signal through the attached control units.
5. Operational In – a multiplex line from all attached control units to the channel. An active 'operational in' signal indicates the selection of a control unit by the channel (except for the 'control unit busy' sequence). The 'operational in' signal is activated only if the 'select out' and 'hold out' signals are active at the control unit; the 'operational in' signal is raised in response to the rise of the 'select out' signal and blocks the propagation of the 'select out' signal to the next control unit. Once raised, the 'operational in' signal stays active until all required information is transmitted between the channel and the control unit. The 'operational in' signal drops after the 'select out' signal falls and after the rise of the signal from the channel that is associated with the transfer of the last byte of information.
6. Suppress Out – a multiplex line from the channel to all attached control units. The 'suppress out' signal is used either alone or in conjunction with other signals from the channel to force data suppression, status suppression, command chaining, or a selective reset.

   Operations whose rate of data transfer can be adjusted without overrunning are subject to suppression of data by the 'suppress out' signal. Completely buffered I/O devices and start-stop devices fall in this category. When the 'suppress out' signal is active at the control unit, the control unit can not raise the 'service in' signal.

   Whenever the channel is unable to immediately handle a polling interrupt status condition in a control unit, the 'suppress out' signal is raised. The control unit cannot then attempt to initiate a polling interrupt sequence to present suppressable-type status information. The 'suppress out' signal suppresses only the initiation of the polling interrupt sequence by the control unit; if the 'suppress out' signal rises after an interrupt sequence has started, the polling interrupt (status) sequence proceeds normally. A chain command operation is indicated to the control unit if the 'suppress out' signal is active when 'service out' signal is raised in response to the 'status in' signal. A chain command operation requires that another command be supplied to the I/O device in operation

immediately following the presentation of the device end status bytes, provided no unusual conditions are encountered during execution of the current operation. Reselection of any I/O device attached to the control unit resets the chain command condition in the control unit. Status conditions such as unit check, unit exception, control unit end, or attention terminate a chain command operation.

A selective reset is indicated whenever the 'suppress out' signal is active and the 'operational out' signal drops. This sequence causes the 'operational in' signal to fall and the working I/O device and its status to be reset. The operation proceeds to a normal stopping point with no further data transfer. The I/O device is busy throughout the reset. A selective reset is issued as a result of a malfunction detected at the channel or a selection timeout condition in the channel.

7. Request In – a multiplex line from all attached control units to the channel. The 'request in' signal is terminated at the channel and is not recognized or used by the 2860.

Tag Lines. The I/O interface tag lines between the channel and control units are used for special sequences and for interlocking and controlling the transfer of information on the 'bus in' and 'bus out' lines. A description of each tag line follows:

1. Address Out – a multiplex line from the channel to all attached control units that provides for I/O device selection and disconnect operations. On an I/O device selection operation, the 'address out' signal causes the control units to decode the unit address on the 'bus out' lines. The control unit recognizing the address responds by raising the 'operational in' signal when the incoming 'select out' signal rises with the 'address out' signal active (except when the control unit is busy). The 'address out' signal normally rises only when the 'select out', 'select in', and 'operational in' signals are inactive at the channel. Once the 'address out' and 'select out' signals are active, the 'address out' signal stays active until either the 'select in' or 'operational in' signal rises, or until the 'status in' signal falls in the case of the control unit busy sequence. During an I/O device selection operation, the 'address out' signal cannot be up concurrently with another 'tag line' signal to the control unit.

   On a disconnect operation with 'select out' signal or 'hold out' signal inactive and the 'address out' signal active, the operating control unit drops the 'operational in' signal, thereby disconnecting from the I/O interface. The 'address out' signal remains active until the 'operational in' signal drops. Mechanical motion of the I/O device continues to a normal stopping point. Status information is then generated and presented to the channel when available.
2. Address In – a multiplex line from all attached control units to the channel. The 'address in' signal informs the channel that the address of the currently selected I/O device has been placed on the 'bus in' lines. A channel responds to the 'address in' signal with the 'command out' signal. The 'address in' signal stays up until the 'command out' signal rises. The 'address in' signal falls upon the recognition of the 'command out' signal by a control unit. The 'command out' signal falls upon the recognition by the channel of the fall of the 'address in' signal. The 'address in' signal cannot be active concurrently with any other 'tag line' signal from the control unit.
3. Command Out – a multiplex line from the channel to all attached control units. The 'command out' signal is used as a response to the 'address in' signal, the 'status in' signal, or the 'service in' signal. The rise of the 'command out' signal in response to an active 'address in' signal during the initial selection sequence indicates that the channel has a command byte on the 'bus out' lines. The 'command out' signal remains active until the fall of the associated 'address in', 'status in', or 'service in' signal. The 'command out' signal cannot be active concurrently with any other 'tag line' signal from the channel except during an interface disconnect sequence when the 'address out' signal may be active.

   During a control-unit-initiated sequence, a 'command out' signal in response to an 'address in' signal means proceed. During a channel-initiated selection sequence, a 'command out' signal indicates that a command byte is available on the 'bus out' lines. During the transfer of a data byte, a 'command out' signal response to a 'service in' signal always means stop. During status handling operations, a 'command out' signal in response to a 'status in' signal means stack status. When the 'command out' signal rises to indicate proceed, stack status, or stop, the 'bus out' lines must contain all zeros but not necessarily correct parity. Although correct parity is provided, the 'bus out' lines are not checked for parity or decoded by the control unit under these circumstances.
4. Status In – a multiplex line from all attached control units to the channel. The 'status in' signal informs the channel when the selected I/O device places a status byte on the 'bus in' lines. The channel responds with either a 'service out' or 'command out' signal, depending upon whether or not the status byte is accepted. The 'status in' signal cannot be active concurrently with any other 'tag line' signal to the channel. Once active, the 'status in' signal stays active until the rise of a 'tag line' signal from the channel (or until the 'select out' signal falls during a control unit busy sequence). The 'status in' signal must fall to allow the responding 'tag line' signal from the channel to fall.
5. Service Out – a multiplex line from the channel to all attached control units. The 'service out' signal is activated by the channel as a response to a 'service in' or 'status in' signal.

On a read-type operation, the 'service out' signal rises to indicate that the channel accepts the data byte offered on the 'bus in' lines. On a write-type operation, the 'service out' signal is activated to indicate that the channel has provided the requested data byte on the 'bus out' lines. During chain command operations, a 'service out' signal in response to a 'status in' signal while the 'suppress out' signal is active indicates to the control unit that the chain command operation is in progress and that the status byte is accepted by the channel. The 'service out' signal stays up until the fall of the associated 'service in' or 'status in' signal. The 'service out' signal should not be active concurrently with any other 'tag line' signal from the control unit except during an interface disconnect sequence (when the 'address out' signal may be active).

6. Service In – a multiplex line from all attached control units to the channel. The 'service in' signal is raised to the channel when the selected I/O device is ready to transmit or receive a byte of information. The channel responds to the 'service in' signal with a 'service out', 'command out', or (during an interface disconnect) 'address out' signal. During a read-type operation, the 'service in' signal is activated when information is available on the 'bus in' lines. During a write-type operation, the 'service in' signal is activated when information is required on the 'bus out' lines. The 'service in' signal cannot be active concurrently with any other 'tag line' signal to the channel. The 'service in' signal stays active until the rise of the 'service out', 'command out', or 'address out' signal. The 'service in' signal cannot drop if a 'tag line' signal from the channel is not activated, nor rise if the 'service out' signal is not dropped.
7. PFCU Data Transfer – a signal to the channel from a high-speed control unit. The 'parallel file control unit (PFCU) data transfer' signal is activated by a high-speed I/O device a minimum of 8 usec before the first 'service in' signal is activated by the control unit. The 'PFCU data transfer' signal remains active until the channel accepts the ending status byte from the control unit. However, the signal may be activated and deactivated during keying operations, provided that the signal is always raised at least 8 usec before the next 'service in' signal is received. The 'PFCU data transfer' signal is not activated by the actual PFCU for data operations of less than 16 bytes; however, other high-speed I/O devices are not restricted in this manner. In the channel, an active 'PFCU data transfer' signal blocks a 'PCI interrupt' operation and is available for use with the high-speed direct access storage priority feature.

Metering Control Lines. The I/O interface metering control lines condition the elapsed time meters and enable/disable switches located in the attached units. A description of each of the lines follows:

1. Metering Out – a multiplex line from the channel to all attached control units. The 'metering out' signal conditions all other usage meters in assignable units connected to the channel via the I/O interface. The 'metering out' signal is active whenever the CPU customer meter and the channel usage meter are recording time.
2. Metering In – a multiplex line from all attached control units to the channel. The 'metering in' signal forces the CPU customer meter to run if the channel is meter-enabled. The 'metering in' signal indicates that one of the attached control units or associated I/O devices is meter-enabled and working.
3. Clock Out – a multiplex line from the channel to all attached control units. The 'clock out' signal provides the CPU interlock control signal necessary for changing the enable/disable metering states of the units. The 'clock out' signal (which originates in the CPU) must be inactive for either a channel or the associated control units to enable or disable their respective metering circuits. The 'clock out' signal is inactive when the CPU is in the wait, halted, or check stop state.

Sequence Controls. Sequences of I/O interface signals generate specific operational conditions. The definitions of these conditions follow:

1. Proceed – indicated whenever the channel activates the 'command out' signal in response to an 'address in' signal at any time other than during a channel-initiated selection sequence. The proceed operation indicates to the I/O device that the normal operating sequences across the I/O interface are to continue.
2. Stop – indicated by a 'command out' signal in response to a 'service in' signal or as a result of an interface disconnect operation that occurs before the normal 'channel end' status is generated for the current operation. The Stop command indicates to the I/O device that the channel is ending the current operation. Upon receipt of the Stop command, the I/O device proceeds to the normal ending point without sending any further data to the channel. The I/O device remains busy until the ending status byte is available and is presented and accepted by the channel.
3. Stack Status – indicated by a 'command out' signal in response to a 'status in' signal. A stack-status operation causes the retention of status information at the control unit until that status byte is accepted by the channel. When the 'command out' signal is activated, a command byte of all 0's (with correct parity) is on the 'bus out' lines. On a 'stack status' operation, the control unit disconnects from the interface after the 'select out' from the channel drops. The 'command

out' signal remains active until the 'operational in' signal falls.

4. Suppress Data – controlled by the 'suppress out' signal. I/O devices whose rate of data transfer can be adjusted without causing an overrun condition are subject to suppression of data. Completely buffered I/O devices and start-stop devices fall in this category. When the 'suppress out' signal is active at the control unit, the control unit must not raise the 'service in' signal. The 'suppress out' signal must be active for at least 250 ns before the 'service out' signal falls to insure suppression of subsequent data.
5. Accept Data – indicates that the information placed on the 'bus in' lines has been accepted by the channel. The accept data operation consists of the channel raising the 'service out' signal in response to a 'service in' signal during a read-type operation.
6. Data Ready – indicates that the channel has placed the requested information on the 'bus out' lines. The data ready operation consists of the channel raising the 'service out' signal in response to the 'service in' signal during a write-type operation.
7. Suppress Status – indicated whenever the channel is unable to immediately handle interrupt-causing status conditions and raises the 'suppress out' signal. With 'suppress out' signal active, the control unit must not attempt a selection sequence to present suppressable-type status information. The 'suppress out' signal must be active at least 250 ns before the 'select out' signal rises at the control unit for proper suppression of status. The 'suppress out' signal only suppresses the initiation of the interface selection by the control unit; if the 'suppress out' signal rises after a status sequence starts, the status sequence proceeds normally.
8. Accept Status – indicates that the status byte on the 'bus in' lines is accepted by the channel. The accept status operation consists of the channel activating the 'service out' signal in response to the 'status in' signal.
9. Command Chaining – indicates that another command for the I/O device currently in operation will immediately follow the presentation of the device end status byte to the channel. When the chain command operation is in progress, the channel raises the 'suppress out' signal before the 'service out' signal is raised in response to the 'status in' signal.

   When a chain command operation is indicated at the time the device end status byte is presented, the indication is valid until the channel reselects the I/O device or until the 'suppress out' signal falls. Reselection of any I/O device attached to the control unit resets the chain command condition in the control unit. Status conditions such as unit check or unit exception terminate chain command operations in the channel. To insure recognition of a chain command operation by the control unit, the 'suppress out' signal must be up at least 250 ns before the 'service out' signal rises in response to the 'status in' signal. Conversely, if a chain command operation is not to be indicated, the 'suppress out' signal must be inactive at least 250 ns before the rise of the 'service out' signal.
10. Interface Disconnect – directs the control unit to deactivate all signals on the I/O interface. An interface disconnect operation is indicated: (1) if the 'address out' signal is active and the 'select out' signal is inactive at least 250 ns before the completion of any signal sequence; or (2) if the 'address out' signal is active at least 250 ns while the 'select out' signal is active, and the 'select out' signal drops while the 'address out' signal remains active. In this case, the 'address out' signal may be active concurrently with another 'tag line' signal from the channel.

    When a normal ending point is reached, the control unit attempts to present any generated status to the channel. If the interface-disconnect operation is performed before the initial status byte is accepted by the channel or after the device end status byte for an operation is accepted by the channel, the control unit does not attempt to present the channel with a status byte as a result of the interface disconnect operation. If disconnected while performing an operation, the I/O device remains busy until the device-end status byte is accepted by the channel. If the interface disconnect operation is performed when the I/O device is not busy, no status is generated nor is the I/O device made busy.
11. Selective Reset – indicated whenever the 'suppress out' signal is active and the 'operational out' signals drops. A selective reset operation causes the I/O device currently in operation to be reset along with the associated status byte in the control unit. As a result of the selective reset operation, the 'operational in' signal falls. The current I/O device operation proceeds to a normal stopping point with no further data transfer. The I/O device currently operating over the I/O interface is the only device reset, even on multi-device control units. The 'suppress out' signal rises approximately 300 ns before the 'operational out' signal drops. The 'operational out' signal remains inactive 6 usec.
12. System Reset – indicated whenever the 'operational out' and 'suppress out' signals are inactive concurrently. All on-line control units and attached I/O devices, along with the associated status bytes, are reset. The control units are in a busy state throughout this procedure. Off-line control units and I/O devices are not affected by the system reset operation. The 'operational out' and 'suppress out' signals are inactive 6 usec on a system reset operation. A system

reset operation is often referred to as a general reset.

13. Control Unit Busy – indicates that the addressed control unit is busy or has status pending for an I/O device other than the one addressed. The addressed control unit presents a status byte either by responding with status information as in the initial selection sequence or by responding with a control unit busy indication during a device selection sequence.

    The control unit busy sequence begins when the channel gates the unit address register contents to the 'bus out' lines and raises the 'address out' signal. The 'select out' signal rises 400 ns later. Each control unit attempts to decode the address on the 'bus out' lines. When the 'select out' signal rises at the addressed control unit, the control unit blocks the propagation of 'select out' signal, raises the 'status in' signal, and places the control-unit-busy status byte on the 'bus in' lines. The 'operational in' signal does not rise. The control unit busy status byte consists of the status modifier bit and the busy bit. After the channel accepts the status byte, the channel deactivates the 'select out' signal. The control unit responds to the fall of the 'select out' signal by dropping the 'status in' signal and disconnecting from the interface ('operational in' signal drops). The channel holds the 'address out' signal active until the 'status in' signal drops.

CHANNEL AND I/O SYSTEM DEFINITIONS

The following paragraphs define operations and information-type codes and words pertinent to the channel and I/O system as a whole. Definitions from the system viewpoint are brief and intended only to introduce operations and terms required for an understanding of channel operations. (See "Channel Operations" in this chapter.) For more detailed information regarding the desired items, refer to the SRL defining the System/360 Principles of Operation, the System/360 I/O Interface (see "Preface" in this manual), or the SRL for the particular control unit and/or the I/O device of interest.

I/O Instructions

I/O instructions are the basic means by which the processor initiates, stops, or tests channel, control units and I/O devices operations. The processor is capable of issuing four I/O instructions: Start I/O, Halt I/O, Test I/O and Test Channel. Diagram 3-2, FEMDM, illustrates the instructions as generated by a typical processor unit.

The Start I/O instruction is issued to the channels (along with a unit address and 'select channel' signal, designating the I/O device and specific channel) when the processor desires an I/O operation to be performed. If the selected channel is available, it selects the I/O device (if available) specified by the unit address, fetches command information from storage, releases the CPU, and controls I/O operations designated by the command information. When the operation(s) initiated by the Start I/O instruction are complete, the channel informs the processor (when permitted to do so by the processor), sends the processor information pertaining to the operation, and stores information pertaining to the status of the I/O device, control unit, and channel in main storage.

The Halt I/O instruction is issued by the processor primarily to terminate an I/O operation already in progress. When the processor decodes a Halt I/O instruction, it sends a 'halt I/O' signal and unit address (via the 'unit address bus out' lines) to all attached channels and a 'select channel' signal to a specific channel. When the channel receives the 'halt I/O' and 'select channel' signals, the channel terminates any I/O operation in progress and informs the processor of the results of the operation by sending the processor one of four possible condition codes. In addition, the channel stores information (the channel status word) in main storage pertaining to the condition of the I/O device, control unit and channel at the time the halt I/O operation is completed.

The Test I/O instruction is issued by the processor to clear an interrupt condition pertaining to a specific I/O device. If the channel contains an interrupt condition pertaining to the I/O device specified by the Test I/O instruction, the Test I/O instruction clears the channel interrupt condition. If the channel contains an interrupt condition for an I/O device other than the one specified by the Test I/O instruction, the interrupt condition is not cleared, and the channel informs the processor of the condition by sending an appropriate condition code. If no interrupt condition exists in the channel (and the channel is not busy), the channel selects the I/O device specified by the Test I/O unit address to obtain status information, then releases the processor. If an interrupt condition is cleared or the I/O device is selected to obtain the status byte, the channel stores the channel and I/O status bytes in main storage. Regardless of the results of the Test I/O operation, the channel sends the processor one of four condition codes (0 through 3); the condition code informs the processor of the Test I/O operation results.

The Test Channel instruction is issued when the processor wishes to test the status of the channel. In this case, the channel informs the processor of its status by sending one of four condition codes (0 through 3) to the processor. Current channel operations are not affected by the Test Channel instruction.

I/O Operations

I/O operations are initiated by the Start I/O instruction. There are five basic I/O operations: read, read backward, sense, write, and control. I/O operations are specified in

the channel command word (CCW) that the channel obtains from main storage after receipt of a 'start I/O' signal, 'select channel' signal, and unit address.

The read operation is performed to transfer data bytes from a selected I/O device to main storage in ascending-order address locations. Transfer of data for the read operation is controlled by the channel based upon control information received in the CCW. (The read operation may also be initiated by an IPL or FLT operation.)

The read backward operation is performed to transfer data bytes from a selected I/O device to main storage in descending-order address locations. Transfer of data for the read backward operation is controlled by the channel based upon control information received in the CCW.

The sense operation is performed to transfer information pertaining to the selected I/O device from the I/O device to main storage. The number of bytes transferred to complete the sense operation is dependent upon the type of I/O device from which the sense information is obtained. The channel controls the transfer of sense information to main storage; channel operations for the sense and read operations are identical. Channel control of the sense operation is based upon control information received in the CCW.

The write operation is performed to transfer data bytes from main storage to a selected I/O device. Data bytes are obtained from ascending-order address locations in main storage. The channel controls the transfer of data to the I/O device based upon control information received in the CCW.

The command operation is performed to instruct the selected I/O device to perform an operation not involving the transfer of data. For example, a command operation is performed to instruct a tape unit to perform a backspace operation. In this case, the channel sends the command, obtained from the channel control word (CCW), to the I/O device. Since data handling is not involved, further channel operations are dependent upon other control information received in the CCW.

Chaining

A channel, when finished with an operation specified by a CCW, can continue start I/O (or IPL or FLT) operations with the same I/O device by fetching a new CCW. The fetching of a new CCW when the channel has completed operations specified by the present CCW is referred to as chaining, and the CCW's belonging to such a sequence are said to be chained. During chaining operations, a chain of CCW's is normally fetched from consecutive doubleword locations in main storage. (The chain of CCW's is obtained from ascending-order address locations.) Two chains of CCW's located in different storage areas can be coupled by means of a Transfer in Channel (TIC) command contained in a CCW in the first chain of CCW's. All CCW's in a chain apply to the I/O device specified in the original Start I/O instruction.

There are two types of chaining: chaining of data addresses, and chaining of commands, controlled respectively by the chain data address (CDA) flag and the chain command (CC) flag in the CCW. These flags, if active, specify the chaining operations to be performed by the channel when the current CCW operation is completed. If the CDA flag is active, the CC flag is ignored by the channel. If a TIC CCW is fetched by the channel as a result of chaining operations, the state of the CDA and CC flags remains unchanged; i.e., the flag values from the preceding CCW are maintained.

On a TIC operation, the channel fetches the next CCW from a new storage location specified in the TIC CCW rather than the next sequential address location in main storage. A chain data operation is specified to continue a read, read backward, or write I/O operation that is presently in progress. Each new CCW that is fetched specifies that a specific number of bytes are to be transferred to or from main storage, and also specifies the starting address location in main storage from which data is to be obtained or to which data is to be stored.

A chain command operation is specified when the program requires that different I/O operations (read, read backward, write, sense, or control) be performed. For example, a chain command operation could be used to cause a read operation to be performed, and then when a new CCW is fetched, to cause a control operation to be performed. Chain command operations are also used to continue the same operation (for example, read); however, due to the manner in which chain command operations are performed, a chain data operation is normally specified if the I/O operation does not change (that is, if different operations are not specified during the chain of CCW's).

Interrupts

Interrupt conditions are a means by which the channel informs the processor that an I/O operation has terminated as a result of data transfer completion, detection of an error during the course of an operation, or detection of an unusual condition at an I/O device. A program controlled interrupt (PCI) flag also causes an interrupt condition, although the current I/O operation progresses normally. The PCI flag is used by the control program to check the progress of an operation.

When the processor is able to acknowledge the 'interrupt request' signal from the channel (processor not masked off to interrupts), the channel sends the unit address of the I/O device involved in the operation to the processor for storage in the PSW. In addition, the channel stores status information pertaining to the I/O device, control unit, and channel at address location 64 (decimal) in main storage.

Polling

Any time the channel is not executing an operation or a chain of operations, attempting a logout operation, or holding a pending interrupt condition, all attached operational control units are scanned for interruption conditions. This scanning is called polling. During a polling operation the channel scans all attached control units in descending order of priority. The first control unit to respond to the polling operation is serviced by the channel to the exclusion of any other control units that might require service. The control units are serviced one by one in order of priority unless channel polling operations are stopped. Polling operations are stopped when the channel receives an I/O instruction which causes the channel to enter an operating sequence. In addition, channel polling operations are inhibited when the channel contains a pending interrupt or polling interrupt condition. The information sent to the channel by the control unit when a polling interrupt condition is detected consists of the 'operational in' signal and followed by the 'address in' signal and a unit address byte. Subsequently, the channel performs operations which recognize that the polling interrupt condition exists for a particular I/O device, cause the status byte causing the polling interrupt condition to be stacked back to the control unit, and clear the stacked status if the processor sends the channel an 'interrupt response' signal (in response to the 'interrupt request' signal from the channel). If the channel is unable to clear the stacked status due to receipt of an instruction or non-receipt of the 'interrupt response' signal, the status remains stacked at the control unit, and channel polling operations are terminated.

Condition Codes

During execution of an I/O instruction, the results of tests by the CPU, channel and control unit, and I/O device are used to send one of four condition codes to the CPU. The condition codes are sent at the same time that the CPU is released to end operations specified by the current instruction and proceed with operations specified by the next instruction. The condition codes indicate whether the channel performed the operation specified by the instruction or the reason the operation was not completed. The CPU stores the condition codes in the program status word (PSW) for later decision making by branch-on-condition operations.

The channel sends condition codes (designated 0 through 3) in binary form on two lines to the CPU. Condition code responses to I/O instructions from the processor follow:

Instruction	Code 0	Code 1	Code 2	Code 3
Start I/O	Available	CSW Stored	Busy	Unavailable
Test I/O	Available	CSW Stored	Working	Unavailable
Halt I/O	Not Working	CSW Stored	Halted	Unavailable
Test Channel	Not Working	CSW Ready	Working	Unavailable

Channel Address Word

The channel address word (CAW) contains the storage protection key bits and the address of the first CCW associated with the Start I/O instruction. The CAW is always located in main storage address location 72 (decimal). The channel always fetches the CAW from main storage at the beginning of a start I/O operation. When the start I/O operation has progressed to the point where a 'release' signal is sent to the CPU, the program is free to change the contents of the CAW for another start I/O operation. The fields in the CAW (Figure 1-4) are allocated as follows:

1. Bit positions 0-3 specify the storage protection key for all commands associated with the start I/O operation.
2. Bit positions 4-7 of the CAW must contain zeros or a program violation exists and the start I/O operation is terminated by the channel.
3. Bit positions 8-31 specify the storage location of the first CCW for the instruction. The three low-order bits of the command address must be all 0's in order to specify the CCW address on doubleword boundaries.

If the command address is not on doubleword boundaries or specifies an invalid storage location, the channel terminates the start I/O operation and causes the status bytes of the CSW to be stored in main storage (location 64, decimal) with the program check bit active.

Channel Address Word (CAW)

Fixed address in storage of 72_{10}

0-3 Storage Protection Key
4-7 Program Check if not all zeros
8-31 Command Address
29-31 Program Check if not all zeros

Figure 1-4. Channel Address Word Format

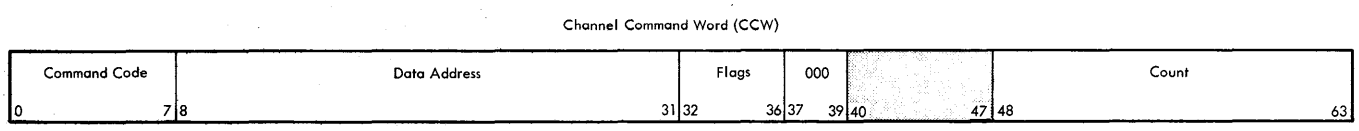
Channel Command Word

The channel command word specifies the command to be executed, the area in main storage associated with the command (if any), and the action to be taken by the channel upon completion of the command. CCW's can be located at any non-restricted address location in main storage. When a CCW is fetched from main storage by the channel, information from the CCW is stored in channel registers. The first (and possibly only) CCW is fetched by the channel during a start I/O initial selection routine. If the channel is performing chaining operations, additional CCW's are fetched when the I/O operation specified by the current CCW has progressed to the point where another CCW is required by the channel.

The fields in the CCW (Figure 1-5) are allocated as follows:

1. Bits 0 through 7 specify a command code, which specifies the I/O operation to be performed.
2. Bits 8 through 31 specify a data address, which is the location of a byte in storage. This is the first data location referred to in the area defined by the CCW.
3. Bit 32 is the CDA flag; when active, the flag specifies chaining of data doubleword addresses and causes the command obtained from the first CCW in the chain to be maintained in the channel for use with succeeding CCW's in the chain.
4. Bit 33 is the CC flag. When the CC flag is on and the CDA flag is off, a chain command operation is specified. A chain command operation causes the channel to fetch the next CCW and obtain the new command from the next CCW upon normal completion of the current operation. The command from the new CCW is then supplied to the same I/O device with which the channel was previously operating.
5. Bit 34 is the suppress incorrect length indication (SILI) flag, which determines whether or not an incorrect length indication detected by the channel is presented to the program (in the CSW). When the SILI flag is active and the CDA flag is inactive, an incorrect length indication is suppressed. If the CCW has active CC and SILI flags, a chain command operation is performed. The absence of the SILI flag or the presence of the CDA flag (SILI flag active or inactive) ends the operation and causes the program to be notified when an incorrect length condition is detected. An incorrect length condition occurs when the end of a data block at the I/O device is reached before the count in the channels count register is exhausted, or when the channel count is less than the device record length.
6. Bit 35 is the skip flag. An active skip flag suppresses the transfer of information to main storage during a read, read backward, or sense operation. With the skip flag active, the channel receives data from the control unit, but the data is not stored.
7. Bit 36 is the PCI flag. A PCI flag causes an enabled channel to interrupt the program as soon after fetching the CCW as possible. The primary use of the PCI flag is to cause the channel to store the CSW as soon as possible without interferring with the channel operation in progress. In this sense, the PCI flag is used primarily as a programming aid.
8. Bit positions 40 through 47 in a CCW are ignored by the channel.
9. Bits 48 through 63 make up the count field. The count specifies the number of byte locations in main storage that will be used by the current operation, provided that the data block at the I/O device is at least as large as the count obtained from the CCW.

Bit positions 37 through 39 of every CCW, other than a TIC CCW must contain all 0's. If not all 0's, the channel detects a program check condition. If the first CCW specified by the CAW does not contain the required 0's, the channel start I/O operations are terminated and the status portion of the CSW is stored. If this zero check condition occurs during a chain data operation, the operation is terminated. If the zero check condition occurs during a chain command operation, the new operation is not started, and an interrupt condition is generated in the channel.



0-7	Command Code
8-31	Data Address
29-31	Program Check if not all zeros on TIC CCW
32	Chain Data Address Flag (CDA)
33	Chain Command Flag (CC)
34	Suppress Length Indication Flag (SLI)
35	Skip Flag
36	Program Controlled Interrupt Flag (PCI)
37-39	Program Check if not all zeros except on TIC CCW
40-47	Ignored
48-63	Count

Figure 1-5. Channel Command Word Format

CCW Command Codes

The command code in the CCW instructs the channel and I/O device what operation to perform. The configuration of the four lowest-order bits in the command field identifies the operation to channel. The high-order bits of the command field are ignored by channel after the byte is checked for correct parity. The channel recognizes four operations:

Write or control (transfer bytes from main storage to control unit).

Read or sense (transfer bytes from control unit to main storage in ascending-order address locations).

Read backward (transfer bytes from control unit to main storage in descending-order address locations).

Transfer in channel (branch operation).

Commands that start read, read backward, sense, write, and control operations cause all eight bits of the command code to be transmitted to the I/O device. Some of the bit positions of these command codes may contain modifier bits. Modifier bits give the I/O device additional details on how the command is to be executed.

The channel detects a program check condition if the invalid command code is detected in the command field. When the first CCW specified by the CAW contains an invalid command, the channel terminates the start I/O operation and stores the status bytes portion of the CSW in main storage. When the invalid code is detected during a chain command operation, a new operation is not started, and an interrupt condition is generated in the channel. For a chain data operation, the first command code in the CCW chain remains in the channel until a CCW arrives with the CDA flag inactive (with the exception of a TIC CCW where the flags are not inspected).

CCW Storage Area Definition

The storage area in main storage used on an I/O operation is defined by CCW's. A CCW defines the storage area (sequential doubleword storage address) by specifying the address of the first byte and the number of bytes that the area contains. The address of the first byte appears in the data address field of the CCW. The number of bytes contained in the storage area is specified in the count field.

A program check condition is generated in the channel if the channel refers to a storage address not in the system. Reference to a non-existent storage location and detection of an invalid CCW address during chain data operations result in detection of an interrupt condition in the channel. The channel causes the I/O device to end the operation when this condition is detected. During chain command operations, the detection of an invalid address prevents the channel from performing a new operation and causes an interrupt condition in the channel.

The count field in the CCW can specify any number up to and including 65,535. Except for a CCW specifying a transfer-in-channel operation, the count must not be zero. Whenever the count field in the CCW initially contains a count of zero, a program check condition is detected by the channel. If this occurs in the first CCW specified by the CAW, the operation specified by the command field is not started and the status portion of the CSW is stored to end the start I/O operation. If a count of zero is detected during a chain data operation, the channel causes the I/O device to end the operation. When a count of zero is detected during a chain command operation, a program check condition is detected by the channel. This prevents the channel from starting the new operation and causes an interrupt condition in the channel.

During a write or control operation, the channel may fetch data and CCW's from main storage before the information is needed by the channel. When the I/O operation uses data and CCW's from a location near the end of available storage area, such prefetching may cause the channel to refer to storage locations that do not exist. The channel may also detect a count of zero in a prefetched CCW. Any programming errors detected by the channel during prefetching of data or CCW's do not cause an error indication until the channel actually attempts to use the information. If the I/O device or the channel ends the operation before the invalid information is used, the error condition is never detected and thus is not stored in the CSW.

Channel Status Word

The channel status word (CSW) consists of a channel status information and an I/O device byte which provides the program with the channel and/or I/O device conditions under which an I/O operation has ended. At the end of a channel controlled I/O operation, the CSW (Figure 1-6) is stored by the channel in main storage at address location 64. The CSW for a given channel ending operation remains available at this location until cleared by the CPU or until replaced by a CSW for another channel CSW store operation. For a detailed description of the CSW and the channel logic involved in assembling the CSW, refer to "CSW Status Byte" in Chapter 6.

CHANNEL OPERATIONS

The 2860 Selector Channel, when instructed by the processor, directs the flow of information between I/O devices and main storage. Channel activities relieve the processor of the need to maintain direct communications with the I/O devices; in addition, the channel provides a common I/O interface for connecting I/O devices with different data transfer rates into the system.

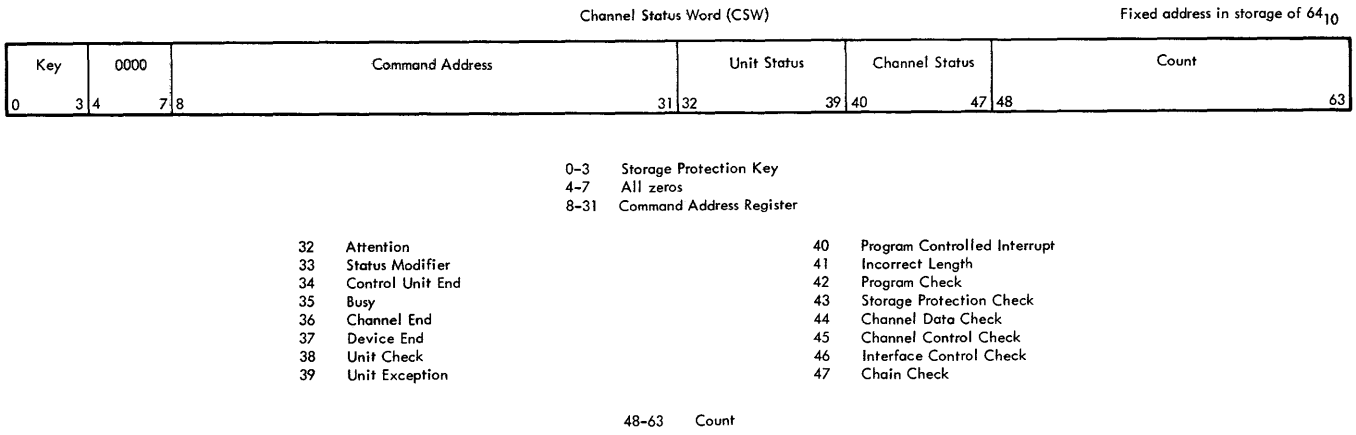


Figure 1-6. Channel Status Word Format

Channel operations are initiated by signals originating in the processor. In most cases, the processor formulates the channel control signals from the program with which the processor is working. (Exceptions are the IPL and FLT operations, where the 'channel control' signals are formulated by switch settings and the program, rather than by the program). The channel uses the channel control signals supplied by the processor to initiate a sequence of operations and signals compatible with any control unit. (The control unit, in turn, interfaces I/O devices, and converts standard I/O interface signals into forms acceptable for use by the attached I/O devices.)

After an I/O operation is initiated at the channel, the channel assembles or disassembles data from or to the I/O devices (via the I/O interface). In addition, the channel controls the transmission of command and data doublewords between the channel and main storage via the BCU interface (Diagram 3-1, FEMDM). To control these operations, the channel maintains and updates a data address and a count (obtained from a CCW in main storage). The address value specifies the destination or source location of data in main storage, while the count value specifies the amount of data to be transferred. When an I/O operation is complete or is terminated by an error condition, the channel informs the processor of the results. When permitted to do so by the processor, the channel then places information in main storage (the CSW) pertaining to the I/O operation. This information is available to the program until altered by the program or until another CSW is stored.

The channel also provides the I/O devices with a means of informing the program of conditions at the device requiring program attention. When an I/O device produces signals that require program attention, the channel converts the signals into other signals recognizable by the processor.

For all but the IPL and a portion of the FLT operations, the channel independently controls I/O operations after being initially instructed by the processor. With the channel controlling the I/O operations, the processor is released to perform other activities. Thus, channel controlled I/O operations are performed at the same time the processor is performing non-related operations.

During channel I/O operations, the channel accesses main storage to obtain CCW's (if required) and to transfer data doublewords to or from main storage locations. A doubleword consists of eight bytes, with each byte comprised of eight data bits and one parity bit. On all channel transfers, odd parity must be maintained on a byte basis. Information is transferred between the channel and control unit on a byte basis and between the channel and main storage on a doubleword basis.

The 2860 Selector Channel operates only in the burst mode. This means that the channel remains logically connected to the control unit until the entire specified operation is complete, or an error condition terminates the operation. When a chain command operation is specified, the channel operates in the burst mode over the entire sequence of commands.

Most channel I/O operations are initiated by a Start I/O instruction. When the processor decodes a Start I/O instruction, the specified channel receives a 'select channel' signal, a 'start I/O' signal, and a unit address identifying the I/O device which is to perform the I/O operation. When these signals and addresses are presented to the channel, the channel (if not busy) fetches the CAW from address location 72 (decimal) in main storage. The CAW identifies the address location in main storage from which the channel is to fetch the first CCW. The CCW specifies the command to be executed and the main storage area, if any, to be used during the I/O operation. While the CAW is being fetched, the channel attempts to select the I/O device specified by the start I/O unit address. The channel attempts the selection by: (1) gating the unit address byte from the 'unit address bus out' lines into the channel's unit address register; (2) gating the unit address register contents to the I/O interface 'bus out' lines; (3) activating the 'address out' signal; and (4) activating the 'select out' signal. If the

addressed I/O device is unavailable, the 'select out' signal passes through all attached control units and returns to the channel as a 'select in' signal. If the I/O device is available, the propogation of the 'select out' signal is suppressed at the control unit to which the addressed I/O device is attached. This causes the control unit to activate the 'operational in' signal which causes the channel to drop the 'address out' signal. The control unit then activates the 'address in' signal and places the unit address of the I/O device on the 'bus in' lines to the channel. The channel responds by sending a command code (requesting the status byte from the control unit) and a 'command out' signal to the control unit. The control unit, in turn, responds by raising its 'status in' signal and placing its status byte on the 'bus in' lines to the channel. The channel then examines the status byte to determine if the control unit is available to perform the I/O operation. If the control unit is available, the channel proceeds with the start I/O operation; if not, the start I/O operation is terminated.

After the CAW has been fetched by the channel and while the channel is selecting the I/O device, the channel fetches the CCW from the main storage address location designated by the address obtained from the CAW. Providing the channel successfully selects the I/O device and obtains an error-free CCW, the channel sends a 'release' signal to the processor and is ready to control the I/O operation specified by the CCW. When the 'release' signal is sent to the processor, further channel operations are independant of the processor; conversely, the processor is free to perform other operations independent of the channel operations.

Assume that the channel has performed the initial selection operations and is ready to begin controlling the I/O operation specified by the CCW. If the I/O operation (specified by the command from the CCW) involves the transfer of information or data bytes between the channel and control unit, the channel is conditioned by the command to respond to 'service in' signals from the control unit. In this case, data is transferred between the channel and control unit on a byte basis each time the channel responds to a 'service in' signal with a 'service out' signal. As doublewords are assembled in the channel (read-type operation) or required by the channel for disassembling (write-type operation), the channel accesses main storage to store or fetch data doublewords.

For I/O operations that do not require byte transmission to or from the I/O device and that do not require continued operation of the control unit, the channel receives a status byte with the channel end bit active immediately after the control unit receives the command byte. This means that the channel portion of the I/O operation is complete and frees the channel to perform other operations.

If the I/O operation involves the transfer of data bytes between the channel and I/O device, data transfers may end before the I/O device reaches the end of a data block, or before the the I/O device mechanical portion of the operation is complete. In this case, the channel receives a status byte with only the channel end bit active and is freed to perform other operations.

If the channel and I/O device complete the I/O operation at the same time, the channel receives a status byte with both the channel end and device end bits active. In this case, the channel is still freed to perform other operations; however, the presence of the device end bit indicates that the I/O device is also available to perform other operations, if required.

When, at the end of an I/O operation, the channel receives a status byte with the channel end bit (and possible the device end bit), the channel ends the I/O operation by an interrupt routine. The interrupt condition causes the channel to send an 'interrupt request' signal to the processor for the purpose of storing the CSW. When the processor honors the 'interrupt request' signal, the channel receives an 'interrupt response' signal which causes the channel to control the storage of the CSW. The CSW contains: (1) an address and count that indicate the amount of main storage locations accessed during the operation; (2) a channel status byte which indicates any error conditions detected during the operation; and (3) a unit status byte which indicates the status of the control unit and I/O device involved in the I/O operation.

The channel can continue different operations with the same I/O device (without receiving another Start I/O instruction) by performing chain command operations. When a chain command operation is specified (by an active CC flag in the current CCW), receipt of the device end status byte by the channel causes the channel to stop I/O device operations, disconnect the I/O device from the I/O interface, and fetch a new CCW from main storage. The channel then reselects the I/O device, provides the device with the command obtained in the new CCW and, if data transfers are specified, controls the transfer of information between main storage and the I/O device. For chain command operations, the channel end status bit is ignored by the channel until the operation specified by the last CCW in the chain is completed. The device end bit is used by the channel to initiate each chain command operation. When the operation specified by the last CCW in the chain is complete, the channel responds to the channel end and device end bits by initiating an interrupt routine as previously described.

On a system basis each channel and each I/O device is identified by a unique address. The channel and I/O device address for a particular channel operation is obtained from the program by the processor. (Diagram 3-2, FEMDM is a typical example of processor handling of the program information containing the I/O instruction, channel address and I/O device address). The channel address (three bits) is decoded in the processor and activates the simplex 'select channel' signal to the specified channel. The I/O device address (eight bits plus parity) is sent to the channels on the multiplex 'unit address bus out' lines. When supplied to the control units by the channel, the unit address is decoded partially by the control unit and partially by the I/O device. Since

each channel may service up to eight control units, and a control unit may service several I/O devices (depending upon the control unit/I/O device configuration), the eight-bit unit address may be used by the channel to address up to 256 I/O devices. The number of valid I/O device addresses applicable for a given channel depends upon the number of operational I/O devices accessible to the channel via the attached control units. If during an I/O instruction operation, the channel receives a unit address for an unavailable I/O device, the channel terminates the operation when the I/O device fails to respond to a selection attempt. An I/O device, control unit, or channel is unavailable when it is not attached in the system, when its power is off, or when it is not meter enabled.

Timing and Sequencing

The channel contains internal timing and sequencing logic to control portions of the operations performed by the channel. The precise usage of the timing and sequencing logic is dependent upon the channel operation performed as well as the routine being performed during the operation.

The main channel timing device is the channel clock. (See "Clock" in Chapter 2.) The clock is turned on to time routines within a channel I/O operation and turned off when timing signals are no longer required.

Sequencing allows the channel to perform routines and operations within an I/O operation; the channel contains five sequence latches. Each sequence latch, when turned on, allows the channel to perform a specific routine or operation required during a channel I/O operation. The channel usage of the sequence latches (designated 1 through 5) is unique in that more than one sequence latch may be on at a given time, depending upon the channel operation performed. However, each sequence has a basic function with regard to channel operation. For some channel operations, the basic function of the sequence is not performed, but the sequence latch may be turned on to permit the channel to perform other necessary decisions and/or register-to-register transfers. The channel sequences, their basic functions, and other functions associated with the sequence, are as follows:

1. Sequence 1. The basic function of the 'sequence 1' signal is to permit the channel to perform setup operations required before read-type or write-type byte transfer operations can begin between the channel and control unit. Setup operations are performed for a channel I/O operation when the first CCW is established in the channel or when a chain command operation occurs. During setup operations, the 'sequence 1' latch is turned on when the 'status in' signal is received from the control unit. For read-type operations, the latch is turned off after the channel examines the control unit status byte and determines that the control unit and channel are ready to begin read byte transfer operations.

   For write-type operations, the 'sequence 1' latch is turned off after the channel fetches two doublewords from storage (unless only one doubleword is specified by the CCW) and determines that the channel and control unit are ready to begin write byte transfer operations.
2. Sequence 2. The basic function of the 'sequence 2' signal is to permit the channel to transfer bytes from the B-register to the control unit on write-type operations or to transfer bytes from the control unit to the B-register on read-type operations.
3. Sequence 3. The basic function of the 'sequence 3' signal is to permit the channel to decrement the count in the count register by eight each time a doubleword transfer to or from the B-register occurs. If a doubleword is to be transferred to or from the B-register, the 'sequence 3' signal permits the channel to perform this operation. The 'sequence 3' signal also permits the channel to sense an impending 'last word' condition and, for a read-type operation, to initiate a storage request for the purpose of storing data in main storage, if required. For read CDA operations, the 'sequence 3' signal permits the channel to fetch a new CCW immediately after the last doubleword of the CDA-modified read-type command is stored. For a write CDA operation, the 'sequence 3' signal permits the channel to fetch the first doubleword of the new CCW when the last doubleword of the old CCW is in the B-register.
4. Sequence 4. The basic function of the 'sequence 4' latch is to permit the channel to increment the data address register value by the equivalent of eight bytes for read-type and write-type operations. After incrementing, the data address register specifies the main storage location of the next data doubleword to be stored or fetched by the channel. For write-type commands, the 'sequence 4' latch permits the channel to initiate a storage request for the purpose of fetching data from main storage. Note that data address incrementing precedes the fetch-data storage request for write-type operations and follows the store-data storage request for read-type operations. Entry into the sequence 4 routine from the sequence 3 routine is normally automatic but may be temporarily blocked during a CDA operation while the channel waits for a new CCW required by the CDA operation.
5. Sequence 5. The basic function of the 'sequence 5' signal is to end a channel operation specified by a CCW. The sequence 5 routine may be entered as a result of successful completion of the operation or as a result of a check condition or error detected during performance of the operation. Entry into the sequence 5 operation normally terminates channel operations initiated by an I/O instruction. An exception is when the sequence 5 routine is entered after successful completion of an operation specified by a CCW with an active chain command flag. In this case, the sequence 5 signal permits the channel to fetch a new CCW, perform another setup operation, and continue I/O operations.

Polling

Any time the channel is not executing an operation or a chain of operations, attempting a logout operation, or holding an interrupt condition, all attached and operational control units are scanned for interrupt conditions. This scanning is called polling. A polling operation initiated by the channel scans all attached control units in descending order of priority. During polling operations, the channel activates the 'select out' signal; the 'select out' signal is then serially propagated through all attached control units. Assuming that no control unit has outstanding status (interrupt condition), the 'select out' signal is propagated through each of the control units and returns to the channel as a 'select in' signal. When the channel receives the 'select in' signal, the 'select out' signal is deactivated and held inactive for a predetermined period by a 2.2ns singleshot signal. After the singleshot times out, the 'select out' signal is again activated and the scanning process is repeated (Figure 1-7). The priority of a specific control unit is determined by its relative position on the select out/select in loop. The first control unit to receive the 'select out' signal has highest priority and the last control unit on the loop (before the 'select in' signal enters the channel) has lowest priority.

A control unit that has outstanding interrupt (status) condition when the rise of the 'select out' signal is detected, stops the propagation of the 'select out' signal, and sends an 'operational in' signal followed by an 'address in' signal and a unit address byte to the channel (Figure 1-7). The channel gates the unit address byte into the unit address register and requests a status byte from the control unit. When the control unit sends the status byte to the channel, the channel stacks the status byte back to the control unit and causes the control unit to disconnect from the I/O interface.

The channel now has an 'interrupt request' signal active to the CPU interface and further polling operations are inhibited. If the processor responds to the 'interrupt request' signal, the channel reselects the control unit with stacked status and obtains the status byte from the control unit. When the status byte is received, the channel stores the status byte portion of the CSW and sends the unit address of the I/O device involved in the polling interrupt operation to the CPU interface.

Condition Codes

Condition code bits are generated by the channel (or by the CPU if a channel selected for an operation is unavailable). One of four possible condition codes is generated by the channel for I/O operations, and identifies the results of an operation or test initiated by the processor and performed by the channel and/or I/O device. The channel sends the appropriate condition code to the CPU interface along with a 'release' signal when the operation results have been determined by the channel. The processor stores the condition code (received in binary form on two 'condition code' lines) in its PSW for optional decision making by the program. Binary condition codes 00, 01, 10, and 11 are referred to throughout this manual as condition codes 0, 1, 2, and 3, respectively.

For the four I/O instructions (Start I/O, Test I/O, Halt I/O and Test Channel) the condition code responses and their meaning are shown in Table 1-1. All condition codes are generated by the channel, except when the channel is unavailable. When the channel is unavailable, the processor internally generates condition code 3 and terminates the I/O instruction.

Table 1-1. Condition Code Responses

Instruction	Code 0	Code 1	Code 2	Code 3
Start I/O	Available	CSW Stored	Working	Unavailable
Test I/O	Available	CSW Stored	Working	Unavailable
Halt I/O	Interrupt in Channel	CSW Stored	Halted	Unavailable
Test Channel	Available	CSW Ready	Working	Unavailable

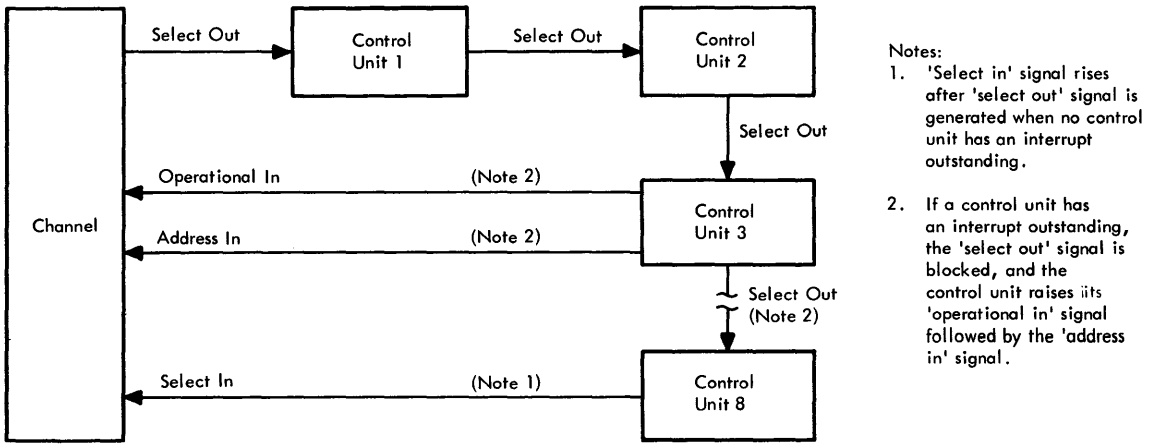


Figure 1-7. Channel Polling Operation

A summary of condition code generation for the Start I/O instruction is shown in Figure 1-8. In addition to condition code generation, the figure summarizes status handling operations (channel status word and CPU 'release' signal conditions) associated with the generation of condition codes. Note that, for a Start I/O instruction, the generation of condition codes by the channel is dependent upon conditions in the channel, addressed control unit and addressed I/O device during initial selection operations. After operations are started, status information in the channel status word is the primary means by which the processor determines the results of the operation.

Condition code generation for the Test I/O, Halt I/O, and Test Channel instructions are as shown in Table 1-1.

Start I/O

A Start I/O instruction is issued to an available channel for the purpose of initiating a command to a specific I/O device. Commands (I/O operations) which may be initiated in the channel and I/O device are: read, read backward, write, sense, and control.

A start I/O operation is initiated at an available channel when the channel receives the 'select channel' and 'start I/O' signals along with a unit address byte. If the channel is busy with another operation or has an interrupt condition pending when the 'start I/O' and 'select channel' signals are received, the channel sends condition code 2 and a 'release' signal to the CPU interface to end the start I/O operation.

Note: If the channel is unavailable, the processor internally generates the condition code 3 and the 'release' signal to end the start I/O operation.

Assuming the channel is unavailable, not busy, and does not contain an interrupt condition, the channel gates the unit address byte from the 'unit address bus out' lines to its unit address register and fetches the CAW from storage location 72. The CAW contains the storage address of the channel command word and the storage protection key, which are gated to the data address register and storage protect register, respectively. Any errors detected in the CAW or unit address byte cause the channel to store the CSW and to send condition code 1 and a 'release' signal to the processor; i.e., the start I/O operation is terminated.

Assuming no errors are detected, the channel now starts two parallel operations to (1) fetch the CSW specified by the address obtained from the CAW; and (2) select the I/O device specified by the unit address in the unit address register. The CCW fetch operation is initiated when the channel activates the 'storage request' signal to the BCU interface. When the 'BCU response' signal is received, the channel gates the CCW address received in the CAW from the data address register to the storage address bus (SAB) and waits for the 'BCU advance pulse' signal. When received, this signal is delayed in the channel until the CCW is available to the channel on the SBO lines. The delayed signal gates the CCW information (command, data address, flags and count) into the respective channel registers. While the CCW is still available on the SBO lines, the channel checks the CCW for correct parity and proper program formatting. If an error is detected, the channel terminates the start I/O operation by sending condition code 1 and the 'release' signal to the CPU interface; in this case, the channel stores the CSW. If no error is detected, the channel turns on the 'CCW valid' latch to indicate that a valid CCW has been gated into the channel. During the CCW fetch operation, the CCW address in the data address register is incremented by one doubleword. With a valid CCW in the channel, the channel gates the updated value into the command address register. The updated command address is available for use to fetch other CCW's if chaining is indicated by the first CCW.

The second operation performed while the CCW is being fetched is the selection of the I/O device specified by the unit address. The channel performs this operation by gating the unit address register contents to the I/O interface 'bus out' lines and activating the 'address out' signal. At least 400 ns later the 'select out' signal is activated by the channel. Assuming the addressed control unit is available and decodes its own address, receipt of the 'select out' signal causes the control unit to respond with an 'operational in' signal, which causes the channel to drop the 'address out' signal. With the fall of the 'address out' signal, a not-busy control unit activates the 'address in' signal and places the address of the selected device on the 'bus in' lines to the channel. The channel then compares the unit address received with the unit address sent to insure that the proper I/O device has been selected. An improper unit address on the 'bus in' lines results in an interface control check condition in the channel, and a selective reset operation to the I/O device occurs.

If the two unit addresses compare, the parallel CCW fetch and I/O device selection operations converge. With the 'CCW valid' latch on and no detected errors, the operation proceeds. The channel gates the command from the command register to the 'bus out' lines and (after a deskew delay) activates the 'command out' signal. The control unit responds by activating its 'status in' signal and gating its status byte to the 'bus in' lines. If the status byte is all zeros, the channel sends condition code 0 and a 'release' signal to the CPU interface. The channel and I/O device are now ready to proceed with operations specified by the command received in the CCW.

If errors are detected (unit address does not compare, 'bus in' parity is wrong, or errors are detected in the CCW), the channel gates all zeros to the 'bus out' lines instead of the CCW command code in the command register. The control unit responds to the All Zeros command with a status byte. The channel then disconnects the control unit

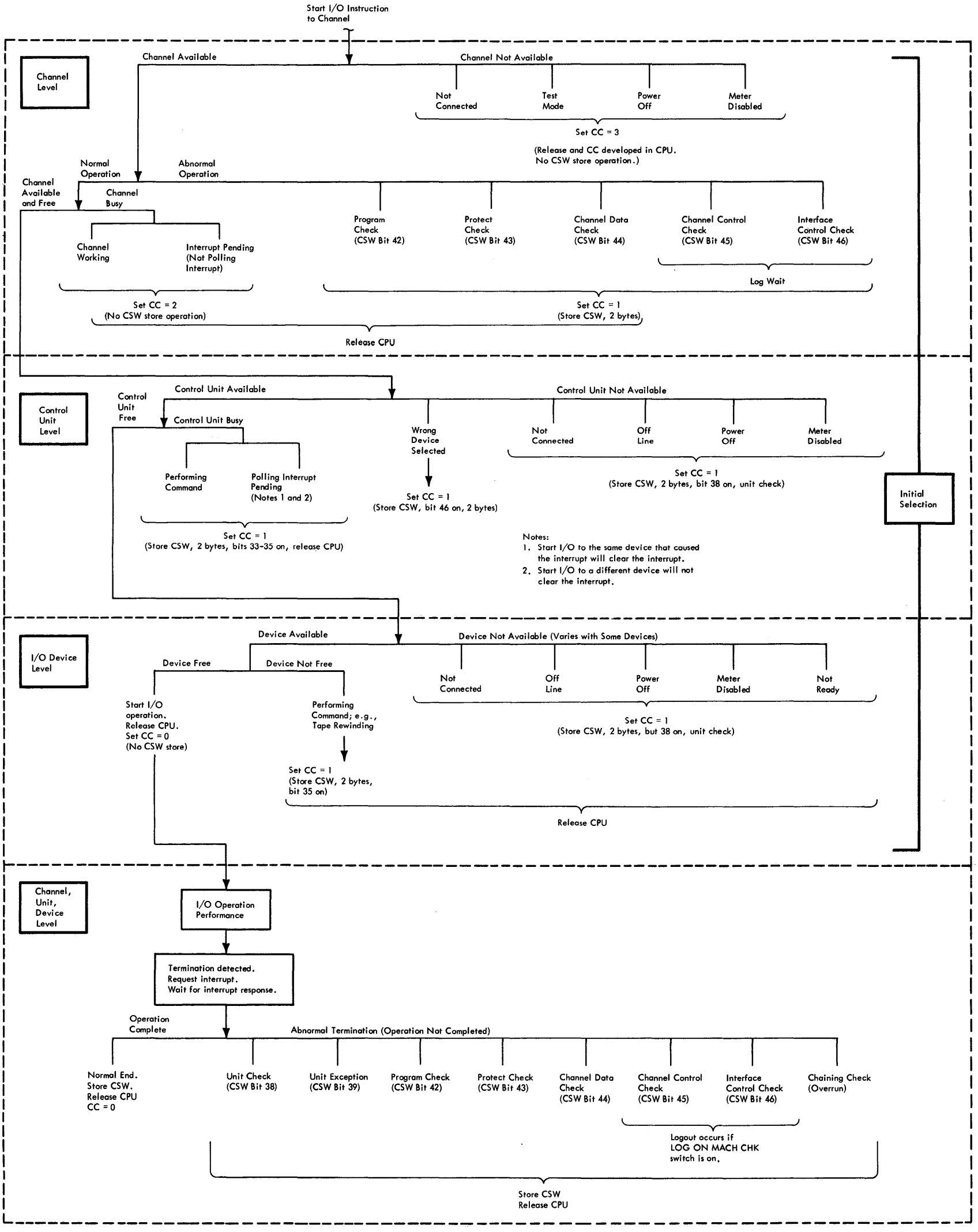


Figure 1-8. Condition Code (and Status) Summary for Start I/O Instruction

from the I/O interface and requests a storage cycle to store the CSW. When the 'BCU response' signal is received by the channel, the channel and unit status bytes in the CSW are stored in main storage, and the channel activates the 'release' signal and condition code 1 to the CPU interface. The channel is then free to receive another instruction. If errors are not detected by the channel, but the control unit responds to the 'command out' signal with anything except zero status, the start I/O operation is ended and the CSW status bytes are stored. The storage protection key, command address, and count in the channel registers are not stored in the CSW under these conditions; the entire CSW is stored if the channel is chaining CCW's and a not all zeros status byte is received by the channel during the reselection sequence.

The condition code responses are interpreted as follows for a start I/O operation:

Condition Code 0 – The addressed channel and I/O device are operational and are not engaged in the execution of any previously started operations. No 'interrupt' conditions exist in the channel or control unit. No programming or equipment errors were detected during the start I/O initial selection routine. The I/O device has accepted and started executing the command as issued.

Condition Code 1 – This condition code indicates that the instruction either was not accepted by the channel, control unit, or I/O device, or was completed and the status bytes have been stored in the CSW. Condition code 1 can be caused by equipment or programming errors or by a response of other than an all zeros status byte to an initial command (except for a command immediate operation.)

Condition Code 2 – This condition code is sent to the CPU by an addressed channel that is already actively engaged in the execution of an instruction. A channel with a pending interrupt condition also produces condition code 2.

Condition Code 3 – This code indicates that the channel, control unit, or I/O device is not available to process the instruction. If the 'channel available' signal to the CPU interface is inactive, the processor internally generates condition code 3 and terminates the start I/O operation. A channel is unavailable when in test mode, not meter-enabled, or not attached to the system. A control unit or device is considered unavailable for the same reasons as the channel; however in this case, the channel generates condition code 3 and activates the 'release' signal. Condition code 3 and the 'release' signal are supplied to the CPU interface to terminate the start I/O operation.

Read

The channel controls read operations after a start I/O initial selection routine has been successfully completed and command bits 6 and 7 (in the command register) are equal to 1 and 0, respectively. Read operations are performed to transfer data from an I/O device to main storage in ascending-order address locations.

The channel begins read operations by first clearing the initial status received from the control unit during the initial selection routine. After the initial status condition is cleared, the channel (see Diagram 3-1, FEMDM) begins to gate data bytes, presented by the control unit on the 'bus in' lines, to the B-register. A data byte is gated into the B-register (and a corresponding mark-B register bit is set) when the channel responds to a 'service in' signal from the control unit by issuing a 'service out' signal. The B-register location to which each byte is gated is controlled by the count in the byte count register. The byte count register has the capability of counting from 0 through 7 and is advanced each time a data byte is gated into the B-register. Thus, when the channel accepts a data byte from the control unit, the encoded count from the byte count register (via the byte count latches) enables gating into a specific B-register byte location. For the first data byte received from the control unit, the byte count register may specify gating to any B-register byte location, depending upon the starting address received from the CCW and subsequently entered into the byte count register.

The channel continues to gate data bytes into the B-register until the B-register boundary is reached (a data byte has been gated into byte location 7). When the B-register is full, the channel transfers the contents of the B-register to the A-register, requests a storage cycle to store the data in the A-register, and starts loading data bytes from the control unit into the B-register for the next data doubleword. While the data in the A-register is being stored, the channel decrements the count value in the count register by an amount equivalent to eight bytes, indicating that a data doubleword specified by the CCW is being stored in main storage.

During the storage cycle to store the A-register data, the channel raises, the 'storage request' signal to the BCU interface. When the channel receives a 'BCU response' signal, the address information required to store the A-register data is gated to the BCU interface (data address register contents to the SAB, storage protect key from the storage protect register to the 'storage protection' lines, mark bits from the mark-A register to the mark lines, and a 'store' signal to the 'store' line). Subsequently, the channel receives a 'BCU data

request' signal which gates the data from the A-register to the 'storage bus in' (SBI) lines; the channel then waits for a 'BCU advance pulse' signal indicating that the data has been stored. When the data address is no longer required on the SAB, the channel updates the data address register contents by an amount equivalent to eight bytes in preparation for storage of the next data doubleword.

Read operations continue (with the channel filling the B-register, transferring the B-register data to the A-register, storing the A-register data, decrementing the count register contents and incrementing the data address register contents) until all data specified by the count in the count register has been transferred. During the last portion of a read operation, the channel examines bits in the flag register to determine if chaining operations are to be performed. Assuming a chaining operation is not specified, the channel terminates the read operation by obtaining the ending status byte from the control unit, disconnecting the control unit from the I/O interface and sending an 'interrupt request' signal to the CPU interface. When the channel receives an 'interrupt response' signal, it stores the channel status word, sends the address of the I/O device involved in the read operation to the CPU interface, and releases the processor with a 'release' signal and condition code 0.

Read Backward

The channel controls read backward operations after a start I/O initial selection routine has been successfully completed, and command register bits 4 and 5 are logic ones while bits 6 and 7 are logic zeros. Read backward operations are performed to transfer data from an I/O device to main storage in descending-order address locations.

Channel read backward operations are similar to read operations with the following exceptions: data is gated into the B-register (and mark-B register bits are set) in the reverse order of that for the read operation (that is, data is gated to B-register positions 7 through 0, respectively); the data address register is decremented (rather than incremented) after each data doubleword storage cycle.

Reverse gating to the B-register is accomplished at the byte count encoder, since the byte count register counts in the forward direction for both read and read backward operations.

Write

The channel controls write operations after a start I/O initial selection routine has been successfully completed and command bits 6 and 7 (in the command register) are equal to 0 and 1, respectively. Write operations are performed to transfer data from main storage to an I/O device. Data is obtained from ascending-order address locations in main storage.

After the initial selection routine is competed, the channel initiates operations to obtain the first two data doublewords from storage (assuming that the count from the CCW specifies that at least two data doublewords are required for the write operation). Transfer of data bytes to the I/O device does not begin until both data doublewords have been received by the channel. (See Diagram 3-1, FEMDM.)

To fetch the first data doubleword, the channel activates the 'storage request' signal to the BCU interface. When the channel receives a 'BCU response' signal, it gates the address of the first data doubleword from the data address register to the SAB and the storage protect key from the storage protect register to the 'storage protection' lines. The channel then awaits the 'BCU advance pulse' signal, indicating that the first data doubleword is forthcoming on the SBO lines. The channel delays the 'BCU advance pulse' signal until the data is present, then gates the data doubleword into the A-register.

When the data address is no longer required on the SAB, the channel increments the address in the data address register by an amount equivalent to eight bytes in preparation for fetching the second data doubleword. Incrementing is accomplished by gating the data address register contents plus 1 to the adder, then gating the results back into the data address register.

With the first data doubleword in the A-register and the data address updated, the channel again raises the 'storage request' signal to request a storage cycle for the second data doubleword. Concurrent with the storage request, the channel transfers the first data doubleword from the A-register to the B-register.

When the channel receives the 'BCU response' signal, it again gates the data address and storage protect key to the BCU interface and waits for the 'BCU advance pulse' signal from the BCU interface. When the 'BCU advance pulse' signal is received, it is delayed until the second data doubleword is on the SBO lines. The delayed signal is used to cause the data to be gated into the A-register.
With two data doublewords in the channel, the channel is prepared to begin transferring data bytes to the selected I/O device. Transfer of a data byte to the B-register begins with the receipt of a 'service in' signal from the control unit, signifying that the control unit is ready to receive a data byte. The channel responds by gating a data byte from the B-register byte position designated by the byte counter to the 'bus out' lines; with the data byte on the 'bus out' lines, the channel raises its 'service out' signal, enabling the control unit to accept the data byte. As the data byte is transferred to the control unit, the channel updates the byte counter to "point" at the next B-register byte location.

The first data byte of the first data doubleword may be gated to the control unit from any byte location in the B-register, depending upon the address value originally obtained from the CCW and gated into the byte counter. Following receipt of the first data byte, the control unit drops its

'service in' signal causing the channel to drop the 'service out' signal. The exchange of 'service in' and 'service out' signals between the control unit and channel continues until the B-register is empty (data has been transferred from byte location 7 of the B-register).

With the B-register empty, the channel transfers the second data doubleword from the A-register to the B-register and, if the count register value so specifies, initiates a storage request to fetch the next data doubleword from main storage. (The count is decremented and the data address incremented prior to fetching the third data doubleword. The count reflects bytes sent to the control unit.) With the second data doubleword in the B-register, the channel is prepared to transfer data bytes to the control unit on a 'service in'/'service out' basis as previously described.

The channel continues to fetch data doublewords from storage and transfer data bytes from the B-register until the count value in the B-register indicates that further data doublewords are not required from storage. At this point, the channel transfers data bytes already in the channel to the control unit until the byte count versus count comparison indicates that the last data byte has been transferred.

At a specified time, prior to transfer of the last data byte, the channel examines the flag register bits to determine if chaining is required. Assuming chaining is not required, the channel terminates the write operation after the 1st data byte is transferred by obtaining status information from the control unit and disconnecting the control unit from the I/O interface. Following the I/O interface disconnect operation, the channel sends an 'interrupt request' signal to the CPU interface. When an 'interrupt response' signal is received at the channel, the channel stores the channel status word, sends the unit address of the I/O device involved in the write operation to the CPU interface 'unit address bus-in' lines, and generates a 'release' signal and condition code 0 to the CPU interface.

Sense

The sense command initiates a sense operation at the selected I/O device. Channel operations for the sense command are the same as for a read operation except that the channel transfers sense bytes (rather than data bytes) from the I/O device to main storage. The sense data is stored in an ascending-order storage address location. The starting address in main storage is the data address specified in the current CCW.

The program uses the sense command to obtain detailed information about the status of the I/O device. This sense information can specify, for example, that a magnetic tape unit is loaded, that the stacker in a card reader is full, or that some unusual conditions happened in the preceding operation. The number of sense bytes and the meaning of the bits within those bytes are peculiar to the type of I/O device; refer to the appropriate functional specifications for the device for a description of sense information.

When a CCW command indicates a sense operation, the channel inspects all five flag bits in the flag register. Command register bit positions 0 through 3 can contain modifier bits, which are meaningful to the I/O device but not to the channel. Bit positions 4, 6, and 7 must be 0; bit position 5 must be 1.

Control

The control command is treated by the channel as a write-type command. It is only at the device or control unit that the true nature of the command can be decoded and executed. The operations that can be initiated by a control command and the control command codes are specified in the functional description of each type of I/O unit.

A control command initiates a control operation at the I/O device. Examples are backspacing, rewinding magnetic tape, or positioning the access mechanism on a disk file. For most control functions, the entire order is encoded in the modifier bits of the command code and no control information is contained in the storage location specified by the data address in the CCW. In some cases, such a multiple-byte addressing for files, further information required for the operation is obtained from main storage starting at the data address specified in the CCW. The channel transfers the additional data from main storage just as in a write operation.

A command immediate operation results if the I/O device needs no additional information to perform the operation specified by the command byte from the channel. In this case, the channel receives a status byte with the 'channel end' bit active as soon as the command is decoded by the control unit and I/O device. The channel is not required to access main storage for additional data. No incorrect length indication condition is detected by the channel regardless of the fact that a CCW specifying a control command must not contain a count of zero. Whether a control function is performed as a command immediate operation or not depends on the operation and the type of device.

A CCW indicating a control operation is inspected for the CDA, CC, SILI, and PCI flags. The skip flag is ignored by the channel. On a command immediate operation, the presence of the CC flag causes the channel to perform a chain command operation, but the presence of the CDA flag suppresses chaining whether the CC flag is on or off.

Command register bit positions 0 through 5 can contain modifier bits which are meaningful to the I/O device but not to the channel. Bit positions 6 and 7 must be 1 to indicate a control operation. A special case involving a control command has been defined for all I/O devices. If the control command code bits 0 through 5 are 0's and bits 6 and 7 are 1's, the command is defined as a No Operation command.

A No Operation command is handled by the channel exactly as any other control command. The I/O unit, however, initiates no operation, but responds to the command by sending an ending status byte back to the channel.

Transfer in Channel

The channel performs a 'transfer in channel' (TIC) operation when a CCW is received by the channel which specifies a TIC command. The TIC operation is performed by the channel to fetch a new CCW from the storage location specified by the data address field of the TIC CCW; i.e., the channel branches to a new chain of CCW's in main storage.

When the TIC CCW is received by the channel, the channel checks the CCW for proper program formatting and gates the TIC CCW data address field from the SBO lines into the data address register. Assuming no formatting error is detected by the channel, the channel activates the 'storage request' signal to the BCU interface. When the 'BCU response' signal is received by the channel, the channel gates the data address register contents to the SAB; as previously stated, the data address specifies the address of the next CCW to be fetched by the channel.

While the channel is fetching the new CCW, the channel increments contents of the data address register by 1 (the equivalent of eight bytes) by gating the data address register contents to the adder. Subsequently, the channel gates the updated address from the adder into the command address register where it is available for use in fetching subsequent CCW's if chaining is indicated by the new CCW.

When the channel receives the 'BCU advance pulse' signal, it delays the signal until the new CCW is on the SBO lines. When the CCW is available, the channel checks it for proper program formatting and gates the CCW information into the appropriate channel registers. A detected error in the new CCW causes a program check condition in the channel and causes the channel to end the operation. Assume that no error is detected. In this case, the channel proceeds with operations to establish the new CCW in the channel in the same manner as for any other non-TIC CCW.

A valid TIC operation may be performed by the channel during either a chain command operation or a chain data operation. However, certain restrictions (dictated by architectural requirements and channel design) are placed upon the use of TIC CCW's. The restrictions are (1) the first CCW fetched by the channel must not contain a TIC command; and (2) the channel (while chaining) must not receive two consecutive CCW's with TIC commands. If either restriction is violated, the channel detects a program check condition and terminates the operation in progress. The channel terminates the operation by causing an interrupt condition in the channel and activating the 'interrupt request signal to the CPU interface. When the channel receives an 'interrupt response' signal (via the CPU interface) it stores the CSW with the program check bit active and sends the 'release' signal to the CPU interface.

A TIC CCW received by the channel contains the following: (1) an address field (SBO bits 8 through 31) in which the three low-order bits (29 through 31) must be all zeros to specify doubleword boundaries for the next CCW (if these bits are not all 0's, the channel detects a program check condition); (2) bits 0 through 3, which are ignored by the channel; (3) a TIC command in the command field (command register bit 4 equals a logic 1 and bits 0 through 3 and 5 through 7 equal logic 0's); and (4) the remaining CCW bits which are ignored by the channel. A TIC CCW can have a count field of zero, since the count field is not gated into the channel and checked for a 'not all 0's' error condition.

Note: The channel performs a pseudo TIC operation to fetch the first CCW of a start I/O operation (after fetching the CAW). However, the TIC operation performed in this case is forced by the channel in order that the channel may use the data address register contents (obtained from the CAW) to fetch the first CCW. This operation is not to be confused with a TIC operation initiated as a result of the channel receiving a CCW with a TIC command.

Data Address Chaining

Chain data operations are performed by the channel when the CDA flag in the current CCW is active and the channel nears completion of the read-type or write-type operation specified by the current CCW. Chain data (CDA) operations are programmed to allow the channel to access different areas of main storage when storing or fetching blocks of data from or to the same I/O device.

When the channel receives a CCW with an active CDA flag, the channel first controls the read-type or write-type operation specified by the command in the CCW. For read CDA operations, the channel assembles doublewords from the control unit in the B-register, transfers the B-register contents to the A-register and stores the A-register contents as during a normal read operation. When the last doubleword of the current CCW is assembled in the channel, the channel stores the doubleword and then fetches a new CCW. While the channel is fetching the new CCW, the channel may continue to receive data bytes from the I/O device, and assemble these bytes in the channel's B-register.

When the new CCW is received by the channel, the channel gates the flag, data address, and count fields from the CCW into the respective channel registers and then establishes the CCW in the channel. Successful read CDA operations performed by the channel depend upon the number of bytes received from the I/O device by the channel while

the CCW was being fetched, as well as the count value from the new CCW and doubleword, or byte boundary specified by the three low-order bits of the data address from the new CCW. If the bytes received by the channel are not compatible with the count value and/or low-order data address bits, the channel detects a chain check condition to terminate the operation. In any event, if the channel receives more than 16 bytes from the I/O device before the new CCW is established in the channel, the channel detects an overrun condition; this also results in a chain check condition and terminates the read CDA operation. If an error or check condition is not detected during the read CDA operation, the channel continues to control the transfer of data from the I/O device to main storage using the control information obtained from the new CCW.

For a write CDA operation, the channel fetches the new CCW from main storage after the last doubleword of the current CCW is received by the channel. While the CCW is being fetched, the channel continues to transfer data bytes specified by the current CCW to the I/O device. When the new CCW is obtained by the channel, the CCW is established in the channel as described for a read CDA operation. However, for the write CDA operation, the count value and low-order address bit comparisons performed for the read CDA operation are not performed, since the data from the previous CCW is already in the channel and being transferred to the I/O device. Assuming the new CCW is established in the channel without error, the channel can begin fetching a doubleword from main storage for the new CCW while the channel completes the transfer of data from the previous CCW. When the last data byte of the previous CCW is transferred to the I/O unit, the control information from the new CCW is used by the channel to begin transferring data specified by the new CCW to the control unit.

For both the read CDA and write CDA operations, the channel continues chaining operations until a CCW is obtained by the channel without an active CDA flag.

Command Chaining

The channel performs chain command operations to control successive I/O operations with the same I/O device; chain command operations are performed by the channel under a single start I/O instruction.

From the standpoint of the I/O device, a chain command CCW applies to an entire data block rather than a portion of the block. For example, if a chain command CCW specifies that 500 bytes are to be read from a data block of 1500 bytes, the next CCW will not specify that the remaining 1000 bytes be read; i.e., the I/O device continues to the end of the data block, stops, and waits for the next command from the channel.

The channel initiates a chain command operation when (1) the CC flag is active, (2) the CDA flag is inactive, and (3) the channel receives a status byte with the device end bit active. When these conditions are present, the channel performs operations to disconnect the control unit from the I/O interface. After the I/O interface disconnect operation, the channel performs parallel operations to reselect the I/O device and fetch the next CCW from the sequential address location contained in the channels command address register.

When the CCW has been received from main storage and established in the channel and the I/O device has been successfully reselected, the channel sends the command from the new CCW to the control unit. (With the CCW established in the channel, the channel is prepared to control any operations required by the new command.) When the control unit and I/O device decode the command, the control unit responds by sending the channel a status byte. The channel then examines the status byte for an all zeros condition (control unit and I/O device ready to perform the command). If the all zeros condition is present, the channel commands the control unit to remove the status byte from the 'bus-in' lines and begin the operation specified by the command. If data transfer operations (read-type or write-type) are involved, the channel controls the transfers as for a normal read-type or write-type operation. The channel continues to perform chain command operations until a CCW without a CC flag (and inactive CDA flag) is received. After the operation specified by this CCW is completed, the channel terminates the chain command operations by activating the 'interrupt request' signal to the CPU interface. When the 'interrupt response' signal is received, the channel stores the CSW and sends the 'release' signal to the CPU interface.

The channel performs chain command operations only if no unusual conditions are detected in the operation specified by the current CCW. If a condition such as a data check or incorrect length (with the SILI flag inactive) is detected, the channel terminates chain command operations. In addition, errors detected during the actual chain command operation (such as a detected interface control check, channel control check or program check condition) cause the channel to terminate the chain command operation.

The sequence of chained CCW's can be changed by the I/O device. This occurs when the control unit presents the channel with a status byte containing the status modifier and device end bits. The combination of the status modifier bit, device end bit, active CC flag and inactive CDA flag causes the channel to increment the command address register contents by eight bytes before fetching the next CCW. This means that the next CCW will be fetched from a main storage address location 16 bytes higher than the address from which the current CCW was obtained. Thus, the I/O unit can cause the channel to "jump" one CCW in the sequence of CCW's when chain command operations are performed.

Skipping

Skipping operations prevent the channel from storing information in main storage. Only the read, read backward, and sense operations are affected. Channel skipping operations are performed only if the skip flag is active in the channel's flag register.

During channel skipping operations, data is transferred from the control unit into the channel as during normal read-type operations. All channel read-type operations are performed, except that the channel does not activate the 'storage request' signal to access main storage. The overall result of the skipping operation is that the channel receives the data specified by the CCW count, holds the data in the A-register briefly, and then resets the A-register; i.e., the data is lost.

For chain command operations, an entire block of data from the I/O device is skipped. However, on chain data operations a selected portion of a data block may be skipped, a new CCW fetched by the channel, and the next portion of the data block stored in main storage. Thus, channel skipping operations provide the programmer with a method of storing only selected data blocks or portions of data blocks during chaining operations.

Test I/O

The Test I/O instruction is normally issued to the channel to clear a channel interrupt condition pertaining to a specific I/O device or to test the status of a specific I/O device. Channel test I/O operations are dependent upon conditions in the addressed channel and/or the addressed I/O device at the time the Test I/O instruction is issued.

A test I/O operation is initiated at the channel when the channel receives a 'test I/O' signal, 'select channel' signal and unit address via the CPU interface. If the channel is engaged in another operation (busy) when the test I/O instruction is initiated, the channel sends condition code 2 and a 'release' signal to the CPU interface to end the test I/O operation.

If the channel contains an interrupt condition, the test I/O unit address is compared with the unit address in the unit address register. If the unit addresses do not compare, the channel sends condition code 2 and the 'release' signal to the CPU interface; i.e., the interrupt condition does not pertain to the I/O device addressed by the Test I/O instruction, and the interrupt condition remains pending in the channel.

If the unit addresses compare, the channel stores the CSW, thereby clearing the interrupt condition. After the CSW is stored, the channel ends the test I/O operations by sending condition code 1 and the 'release' signal to the CPU interface.

If the channel is not busy and does not contain an interrupt condition, the channel attempts to select the I/O device specified by the test I/O unit address. If the selection attempt is successful and the status byte from the control unit is all zeros (I/O device available for operation), the channel ends the operation by sending condition code 0 and the 'release' signal to the CPU interface. If the I/O device is busy or the channel detects an error condition during the selection attempt, the channel stores the CSW in main storage location 64 (decimal) and ends the test I/O operation by sending condtion code 1 and the 'release' signal to the CPU interface. If the I/O device is unavailable, the channel sends condition code 3 and the 'release' signal to the CPU interface to end the test I/O operation. The CSW, when stored, identifies any error condition detected during the test I/O operation, while the condition codes identify channel and I/O device conditions detected by the Test I/O instruction. A summary of the test I/O condition codes and their meaning follows:

Condition Code 0 – means that both the channel and I/O device are available for an operation.

Condition Code 1 – means that the test I/O operation resulted in the channel storing the CSW. The interrupt conditions indicated in the CSW were reset in the I/O device and cleared in the channel if the interrupt conditions pertained to the addressed device. The presence of the unit check, channel control check, or the interface control check bits in the absence of the channel end or device end bits can be due either to a condition caused by the preceding operation or to an equipment error detected during execution of the test I/O operation. A busy bit indicates a busy I/O device if present alone, or a busy control unit if the status modifier bit is also present.

Condition Code 2 – indicates that no action has been taken by the channel because the channel is actively working on a previously initiated command, or has an interrupt condition pending from some device other than the device addressed by the test I/O operation.

Condition Code 3 – indicates that the channel, control unit or I/O device is unavailable (not attached to the system, has power off, is not meter enabled, or has a malfunction preventing normal operation).

Halt I/O

The Halt I/O instruction is normally issued to the channel to terminate a channel operation and cause the channel to disconnect the operating I/O device from the I/O interface. Channel halt I/O operations are dependent upon conditions in the addressed channel and/or the addressed I/O device at the time the Halt I/O instruction is issued.

A halt I/O operation is initiated at the channel when the channel receives a 'halt I/O' signal, a 'select channel' signal and a unit address via the CPU interface. If the channel contains a pending interrupt condition, the channel immediately ends the halt I/O operation by sending condition code 0 and a 'release' signal to the CPU interface. The interrupt condition remains pending in the channel.

If the channel is not busy, the channel attempts to select the I/O device addressed by the halt I/O unit address. If the I/O device is not available (indicated when the channel receives a 'select in' signal during the selection attempt), the channel terminates the halt I/O operation by sending condition code 3 and the 'release' signal to the CPU interface. If the channel determines that the I/O device is available (an all zeros status byte is received from the addressed I/O device), the channel performs operations to disconnect the I/O device from the I/O interface. Subsequently, the channel activates the 'storage request' signal for the purpose of storing the CSW. After the CSW is stored, the channel terminates the halt I/O operation by sending condition code 1 and the 'release' signal to the CPU interface.

If the channel is working (busy), the Halt I/O instruction causes the channel to disconnect the operating I/O device from the I/O interface. The channel accomplishes this by activating the 'address out' signal and after a specified period, deactivating the 'select out' signal. The control unit recognizes this signal sequence by deactivating all I/O interface signals to the channel. Meanwhile, the channel activates the 'interrupt request' signal to the CPU interface. The channel then terminates the halt I/O operation by sending condition code 2 and the 'release' signal to the CPU interface. The channel interrupt condition remains pending.

When the channel receives a Halt I/O instruction while performing an I/O device reselection sequence during a chain command operation, channel operations continue until the initial status byte is received from the control unit. The channel then activates the 'interrupt request' signal and ends the halt I/O operation by sending condition code 2 and the 'release' signal to the CPU interface.

A summary of the condition codes generated for a halt I/O operation and their meaning follows:

Condition Code 0 – means that an interrupt condition is pending in the channel.
Condition Code 1 – indicates that the channel and unit status bytes portion of the CSW was stored.
Condition Code 2 – indicates that an I/O operation was halted by the Halt I/O instruction and that the channel has an interrupt condition pending.
Condition Code 3 – indicates that the channel control unit or I/O device was found to be unavailable (not attached to the system, power off, meter disabled, in the test mode, or contains a malfunction which precludes normal operation).

Test Channel

The Test Channel instruction is issued to the channel to determine the condition of the channel. The Test Channel instruction is initiated when the processor activates the 'test channel' and 'select channel' signals to the channel.

Note: If the channel is unavailable, the CPU internally generates condition code 3 and the 'release' signal to end the test channel operation.

Since no I/O device selection is required by the Test Channel instruction, a unit address is not sent to the channel. In the channel, test channel operations test, but do not alter, the state of the channel. Upon receipt of the 'test channel' signal and 'select channel' signal, the channel generates one of three condition codes and sends a 'release' signal to the CPU interface to end the test channel operation. (Recall that the fourth condition code is generated by the processor if the channel is unavailable; in this case, the 'test channel' and 'select channel' signals are not sent to the channel). The condition codes are stored in the processors PSW for program reference, if required.

A summary of the condition codes generated for a test channel operation and their meanings follows:

Condition Code 0 – the channel is not engaged in an operation and does not contain a pending interrupt condition.
Condition Code 1 – the channel contains an interrupt condition that will cause the CSW to be stored when the processor provides the channel with an 'interrupt response' signal.
Condition Code 2 – the channel is performing an operation started by some previous instruction.
Condition Code 3 – the channel is unavailable (not attached in the system, has power off, is meter disabled, is in the test mode, or contains a malfunction precluding system operation.)

Channel Ending Operations

- An ending sequence (sequence 5) causes an interrupt condition.
- The interrupt condition can be cleared by an 'interrupt response' signal to the channel in response to an 'interrupt request' signal from the channel.
- Information pertinent to the channel is stored in main storage in the CSW where it is available for use by the program.

- The CSW contains the channel and unit status bytes, the command address from the channels command address register, and the count from the count register.

- Information in the CSW is an indication of the conditions under which an operation was terminated, whether the operation was completed, and (if not completed), the extent of the operation performed.

A channel ending sequence may be initiated during a start I/O, test I/O, halt I/O, or IPL operation due to detection of an error condition, nonselection of an I/O device, or completion of the operation.

The purpose of the channel ending sequence (sequence 5) is to terminate the operation in progress, and where applicable, send a condition code and 'release' signal to the CPU interface. Depending upon the condition which caused the channel to enter the sequence 5 routine, the channel may or may not enter an interrupt routine to store the CSW.

The channel enters a sequence 5 ending routine for any of the following reasons:

1. A program check condition is detected during initiation of start I/O operation.
2. The initial status byte from control unit does not equal all zeros on a device selection routine.
3. An interface control check condition is detected (Refer to "Interface Control Check" in Chapter 6).
4. The control unit is found to be busy (control unit busy condition) during a device selection sequence.
5. A channel control check condition is detected. (Refer to "Channel Control Check" in Chapter 6.)
6. A chain check condition is detected. (Refer to "Chain Check" in Chapter 6.)
7. A program check condition is detected. (Refer to "Program Check" in Chapter 6.)
8. The channel receives a 'status in' signal during a read-type or write-type operation before the operation specified by the CCW is complete.
9. The channel receives a 'storage protect check' signal or 'invalid address check' signal while controlling a store or fetch operation.
10. The channel operation is successfully completed and the channel is to end the operation.
11. The channel is unable to perform the operation initiated by the processor. (The channel may be working on a previously initiated operation or may contain a pending interrupt condition.)

Depending upon which event caused the channel to enter the sequence 5 ending routine, the channel performs operations to: (1) disconnect the I/O device from the I/O interface, if the device is connected; (2) calculate a residual count value if a read-type or write-type operation was prematurely terminated and the SILI flag is inactive; (3) send condition code 2, or 3 and a 'release' signal to the CPU interface if the channel or I/O device is busy or unavailable (respectively); and/or (4) enter an interrupt routine for the purpose of storing the CSW if the operation was terminated under conditions requiring storage of the CSW. (For example, an interface control check, channel control check, program check, invalid storage address, storage protect check, or control unit busy condition during a start I/O operation causes the channel to enter an interrupt routine to store the CSW).

In summary, the channel enters a sequence 5 ending sequence to terminate an operation that has just been initiated, is in progress, and is terminated by detection of an error condition, or has been successfully completed.

I/O Interrupts

I/O interrupt conditions provide a means for the processor to change state in response to conditions in the channel or attached I/O devices. Interrupt conditions can be caused by detected errors (in the channel, control unit, or I/O device), by the ending of an I/O operation, or by operator intervention at the I/O device.

The channel processes all interrupt conditions by issuing an 'interrupt request' signal to the CPU interface. If an I/O device contains an interrupt condition, the device attempts to send a 'status in' signal and status byte to the channel to cause an interrupt condition in the channel. I/O device interrupt conditions are: channel end, device end, attention, control unit end, unit check or unit exception. If the channel receives a status byte when the CC flag is active, the channel end and device end bits are not considered interrupt conditions unless accompanied by one of the other status interrupt bits. The channel interprets the unit check and unit exception bits as an interrupt condition only when these bits are active in the initial status byte obtained during selection of an I/O device. Once the channel has selected an I/O device and commenced operations specified by the command, the unit check and unit exception bits will appear only in a status byte in which the channel end, control unit end or device end bit is active. In this case, the unit check and unit exception bits are interpreted by the channel as an interrupt condition.

Either a program controlled interruption (caused by an active PCI flag) during data transfer or the execution of a Halt I/O instruction can cause the channel to activate the 'interrupt request' signal without ever having received the ending status byte from the I/O device. If the channel detects a program check condition while starting a chain command operation, a channel-generated Test I/O command is sent to the control unit. The channel then receives a status byte from the control unit and activates the 'interrupt request' signal to the CPU interface. The unit status byte may or may not contain all zeros.

If the channel is not engaged in an operation and does not have an interrupt condition pending, the channel polls the

attached control unit for an interrupt condition. If a control unit has a pending interrupt condition, it responds by sending first the unit address byte and then the status byte to the channel. The channel then stacks the status byte back to the control unit and activates the 'interrupt request' signal to the CPU interface. If the channel then receives an 'interrupt response' signal (via the CPU interface), the channel reselects the I/O devices to obtain the stacked status byte and store the CSW.

Interrupt Priority

Channel 'interrupt request' signals are not synchronized with the activity in the processor and an interrupt condition can be pending from more than one I/O device on a given channel at the same time. The priority among the devices on any one channel is determined by cabling when the devices are installed on the channel. The first device to receive and respond to the 'select out' signal when the channel is polling has the highest priority. Interrupt priority among the channels is determined at CPU. The lowest-numbered channel with an outstanding interrupt condition has top priority unless masked. For example, channel two would have priority over channel three. The status condition that caused the channel to activate the 'interrupt request' signal is held at the I/O device or in the channel until the channel receives an 'interrupt response' signal via the CPU interface. An interrupt condition is never intentionally lost.

An interrupt condition can be cleared only when the channel to which the device is attached is not masked by the program at the processor, and after the execution of the current instruction in the processor ends. If a masked channel accepts status for an interrupt condition from an I/O device, the interrupt condition is cleared immediately after the end of the processor instruction that removes the mask, and before the next processor instruction is executed. If more than one channel is unmasked at the same time, the interrupt condition is cleared from the channel that has the highest priority among those with active 'interrupt request' signals. If the I/O device status byte for an interrupt condition is not in the channel by the time the mask is removed, the interrupt condition cannot be cleared immediately after the end of the instruction that removes the mask. This delay can occur regardless of how long the 'interrupt' condition has existed in the I/O device.

Program Controlled Interruption

The PCI flag permits the program to cause an I/O interrupt condition in the channel while the channel is performing an I/O operation. The PCI operation functions as a programming aid; its prime purpose is to give an indication rather than directly control an operation.

The PCI operation is initiated in the channel by the presence of an PCI flag obtained from the CCW. The PCI flag can be in the first CCW specified by a Start I/O instruction or in any later CCW fetched during chaining operations. Whenever the PCI flag is on in the current CCW, the channel attempts to interrupt the program for the purpose of storing the CSW as soon as possible after starting a data transfer I/O operation. Neither the presence of a PCI flag or successful performance of the interrupt routine by the channel prevents the channel from executing the current I/O operation. The CSW can be stored by a channel controlled PCI operation while the I/O operation being performed by the channel is in progress; a CSW containing the PCI bit indicates to the program how far the channel I/O operation had proceeded when the CSW was stored. If the pending PCI operation is not performed while the channel I/O operation is in progress, the CSW is stored at the end of the I/O operation with the normal ending conditions indicated, and the PCI bit is active.

When the PCI bit causes the channel to store the CSW before an I/O operation is complete, the channel end bit in the CSW in inactive. If the PCI CSW is stored during an I/O operation and the channel detected data errors in the operation before the ending CSW is stored, the channel data check bit is inactive. The PCI CSW store operation, stores only the PCI bit; no other channel status bits are stored. An error condition in the channel is not reset when the channel stores a PCI CSW; at the end of the operation, the CSW is stored again. This CSW contains the channel status bits without an active PCI bit if the PCI was previously honored. The command address in the CSW identifies the storage location of the next sequential CCW. Presence of the channel end bit with an active PCI bit indicates to the program that the operation is ended, and that the PCI condition was not honored before the ending interrupt occurred.

If a channel chaining operation occurs before the PCI flag causes the CSW to be stored, the PCI condition remains active in the channel when the new CCW is received. This applies to both chain data and chain command operations. The PCI flag condition is not changed by a channel TIC operation during chaining. If a new CCW is fetched with the PCI flag active before the PCI CSW is stored as a result of a previous PCI flag, the channel attempts to store only one PCI CSW.

The channel checks for an active PCI flag on every CCW except a TIC CCW. On a TIC CCW, flags are not gated to the flag register and the flags from the preceding CCW remain in the channel. The PCI flag is ignored by the channel during all IPL operations.

Test Facilities

The selector channel test facilities are designed to provide methods for determining what channel areas should be

tested and what portions of the over-all system should be included in the test.

The primary channel test facilities consist of the logout function and the test mode function. The logout function provides the channel with a means of storing three logwords during channel auto-mode operations. The three logwords are stored in main storage (starting at location 304, decimal) only when the channel CE panel LOG ON MACH CHK switch is on and the channel detects a channel control check or interface control check condition. When these conditions exist, channel operations are stopped and the channel sends an 'interrupt request' signal to the CPU interface. When the channel receives an 'interrupt response' signal, the three logwords are sequentially stored by the channel. The logwords contain information pertinent to channel conditions at the time the check condition was detected. At the customer's option, the logwords can be printed out to provide a visual indication of these conditions.

The test mode function is manually initiated at the CE panel and allows the customer engineer to use the CE panel controls to simulate various channel operations. If the LOG ON MACH CHK switch is on during test-mode operations, channel detection of either a channel control check, interface control check, or channel data check causes the channel to stop operations. In this case, the three logwords are not stored, but channel conditions are displayed on the channel's CE panel indicators. The channel remains in the stopped condition until reset.

Other CE panel controls allow the customer engineer to operate the channel to simulate the I/O interface or to simulate storage. Either or both simulation operations may be performed, or the channel may be switch-controlled to access main storage or an attached I/O device. Most operations normally initiated by the processor can be simulated in the test mode using the CE panel switches. (While in the test mode, the channel is electrically disconnected from the CPU interface.)

This chapter describes the logic units composing the 2860 Selector Channel. Each functional unit is described separately. Simplified diagrams, flow charts, and timing charts are included where necessary to aid in the understanding of unit operations and/or unit timing. Where necessary, references are made to diagrams in the FEMDM for the 2860 Channel Selector.

UNIT ADDRESS REGISTER

The unit address register is a nine-bit register (eight bits plus parity) which receives and stores the address of the I/O unit involved in a current I/O operation. Once the unit address is stored in the register, it is available for outgating or comparison purposes during the course of an operation.

Block Diagram Description

- Unit address byte gated to unit address register from:
  1. 'Unit address bus out' lines (start I/O, test I/O, halt I/O channel free, IPL and FLT operations).
  2. 'Bus in' latches for I/O interrupt.
  3. CE panel UNIT ADDRESS switches during test mode operations.
- 'Address compare' logic compares address in 'bus-in' latches with address in unit address register during setup routine; mismatch indicates wrong device responded and interface control check condition results.
- 'Unit address compare' logic compares address on 'unit address bus out' lines with address in unit address register during test I/O operation with interrupt condition in channel. Mismatch results in CC2 to CPU; match causes test I/O operation to clear pending interrupt.
- Unit address register outputs supplied to:
  1. 'Bus out' latches for I/O device selection.
  2. SBI lines for storage during logout or IPL end operations.

The unit address register and associated ingate logic, ingate powering logic, comparison logic, and 'unit address bus in' gating logic are shown in Figure 2-1.

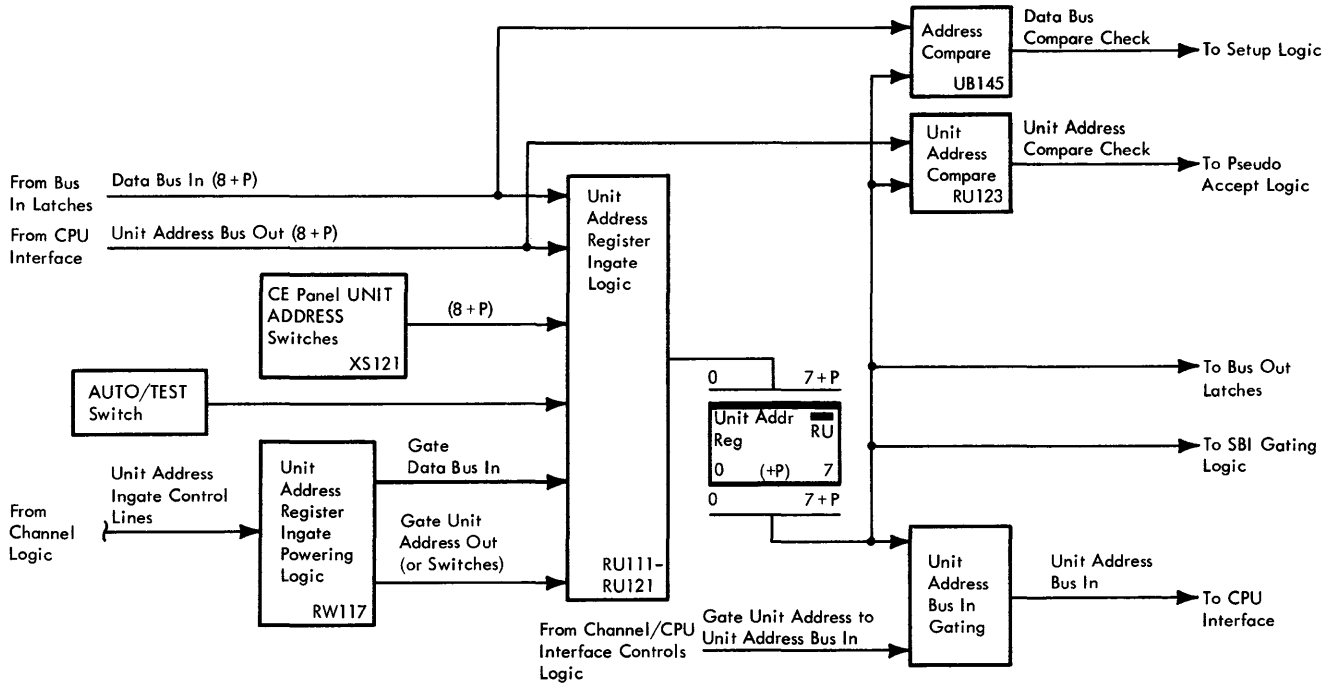


Figure 2-1. Unit Address Register, Gating Logic and Comparison Logic, Block Diagram.

The unit address byte is gated into the unit address register from one of three sources. The three sources, and the operations associated with the gating from each source, are: (1) the CPU interface from the 'unit address bus out' lines for start I/O, test I/O, halt I/O (with channel free), IPL or FLT operations; (2) the I/O interface via the 'bus in' latches ('data bus in' lines) upon an I/O interruption; and (3) the CE panel UNIT ADDRESS switches during simulate CPU test mode operations. Unit address ingating is accomplished at the 'unit address register ingate' logic as determined by the position of the AUTO/TEST switch on the CE panel, and gating signals from the 'unit address register ingate powering' logic. With the AUTO/TEST switch in the AUTO position, the 'unit address bus out' lines are ingated when the 'gate unit address out' signal is generated; in the TEST position, the 'gate unit address out' signal ingates the contents of the UNIT ADDRESS switches. The 'data bus in' bits are ingated when the 'gate data bus in' signal is generated.

Once the unit address register is loaded, the contents may be gated to the CPU interface or I/O interface and are also available for comparison with a unit address byte from the CPU interface or I/O interface depending upon the current channel operation.

Two comparison operations are performed in the channel; an address comparison, and a unit address comparison. The address comparison is performed by the 'address compare' logic and is a bit-to-bit comparison of the unit address register contents and the data on the 'data bus in' lines from the 'bus in' latches. The results of this comparison are monitored by the channel during a setup operation involving selection of an I/O device by the channel. If the correct I/O device has been selected, the unit address byte from the 'bus in' latches and the unit address register are identical. In this case, the 'data bus compare check' line is inactive, and the setup operation is allowed to continue. If the address comparison is not successful, the 'data bus compare check' line is activated and, when gated during the setup operation, results in an 'interface control check' condition which terminates the current I/O operation.

The unit address comparison is performed by the 'unit address compare' logic and is a bit-to-bit comparison of the unit address from the CPU interface ('unit address bus out' lines) and the contents of the unit address register. The comparison results are monitored by the channel during a Test I/O instruction. If the unit addresses do not compare, the 'unit address check' line is activated to indicate that the channel contains a pending interrupt from an I/O device other than the one specified by the Test I/O instruction. In this case, the appropriate condition code is sent to the CPU and the test I/O operation is terminated. If the 'unit address check' line is not active, the test I/O operation continues, and the address byte on the 'unit address bus out' lines is gated into the unit address register. The unit address register contents are supplied to the 'bus out' latches, the 'storage bus in' gates, or the 'unit address bus in' gates depending upon the operation in progress.

The 'bus out' latches receive the unit address register contents when selection of the I/O device is required during a Start I/O, Test I/O, Halt I/O, IPL or FLT instruction. After gating into the 'bus out' latches, the unit address is subsequently presented to the control unit on the 'bus out' lines and is used by the control unit to select the addressed I/O device.

The unit address register contents are gated to the 'storage bus in' (SBI) lines for a logout operation (during storage of log word 2) or at the end of an IPL operation. In the case of the logout operation, the unit address identifies the I/O device with which the channel was working prior to the logout operation. In the case of the IPL operation, the unit address identifies the I/O device which handled the IPL operation.

Diagram 4-1, FEMDM, summarizes the ingating, comparison, and outgating operations associated with the unit address register.

Detailed Description

Simplified logic for the unit address register and associated ingating logic, comparison logic, and outgating logic is shown in Diagram 4-2. Ingating logic, the unit address register, comparison logic, and outgating logic are described separately in the following text.

Ingating Logic

- 'Gate BI to UAR' signal activated by polling interrupt operation to gate address from 'bus in' latches to unit address register.
- 'Polling interrupt request', 'address in', 'T2', and 'not T3' signals activate 'gate bus in' signal.
- 'Gate bus in' signal activates 'machine reset or ingate reset' signal (reset unit address register) and turns on 'bus in to register' latch (gate address byte from 'bus in' latches to unit address register).
- 'Gate UAB to UAR' signal activated by 'turn on setup IPL', 'fetch CAW', or 'test I/O or halt I/O and select in' signal to gate address from 'unit address bus out' lines into unit address register.
- Active 'CPU accept interrupt' signal inhibits 'gate UAB to UAR' signal.

Ingating to the unit address register is performed at the 'unit address register ingating' logic under control of signals from the 'unit address register ingate powering' logic and from the

AUTO/TEST switch on the CE panel. The 'unit address register ingate' logic receives unit address bits from three sources; the 'data bus in' lines from the 'bus in' latches, the UNIT ADDRESS switches on the CE panel, and the 'unit address bus-out' lines from the CPU interface.

The 'data bus in' lines are connected to eight 'bus in' ANDs which are gated by the 'gate bus in to unit address register' signal ('gate BI to UAR' signal) from the 'unit address register ingating powering' logic. Gating the 'bus in' ANDs causes a one level at any of the eight gates to turn on the corresponding latch of the unit address register. The 'gate BI to UAR' signal is activated by a polling interruption. When a control unit supplies the channel with an 'address in' signal as a result of a channel polling operation, the 'polling interrupt request' signal from the 'polling interrupt request' latch is raised. This signal is AND'ed with the 'address in' signal and the 'T2' and 'not T3' clock signals to produce a 'gate bus in' signal. The 'gate bus in' signal has two functions: (1) to produce a signal which resets the unit address register; and (2) to set the 'bus in to register' latch which produces the 'gate BI to UAR' signal. The reset signal ('machine reset or ingate reset') is activated by OR'ing the 'gate bus in' signal. Shortly after the reset signal is presented to the unit address register, the 'bus in to register' latch output is presented to the 'unit address bus' gates and the bit values on the 'data bus in' lines are gated to the unit address register. However, the unit address register cannot latch while the 'machine reset or ingate reset' signal is present. The 'machine reset or ingate reset' signal is normally turned off by the rise of the 'T3' clock signal. With the reset level removed from the unit address register, the bit values present at the output of the 'bus in' gates are latched into the unit address register. The fall of the 'machine reset or ingate reset' signal is delayed by logic and used to reset the 'bus in to register' latch; the delayed reset of the 'bus in to register' latch ensures that the 'bus in' bits are present at the unit address register latches for a sufficient time to permit latching. The reset/set timing described above is shown in Figure 2-2.

Either the 'unit address but out' data or the contents of the UNIT ADDRESS switches are available for gating to the unit address register, depending upon the setting of the CE panel AUTO/TEST switch and upon the condition of the CPU clock and the channel metering logic.

The condition of the CPU and channel metering logic is reflected by the 'block program control' signal. When the 'block program control' signal is present, channel operation in the system is not permissible; in this case, the channel may operate only in the simulate CPU mode. With the 'block program control' signal inactive, the 'auto mode' level from the AUTO position of the AUTO/TEST switch is AND'ed to generate a 'not simulate CPU' level which enables nine ANDs receiving the 'unit address bus out' bits from the CPU interface. After AND'ing, the nine 'unit address bus out' bits are individually OR'ed and supplied to the nine 'unit address bus' ANDs. If the 'block program control' signal is active, the contents of the UNIT ADDRESS switches are AND'ed through nine gates by the 'simulate CPU' level. In turn, these bit values are OR'ed to the same nine lines previously described for the 'unit address bus-out' bits.

At the 'unit address bus' gates, the unit address bits are gated when the 'gate unit address bus to unit address register' signal ('gate UAB to UAR' signal) is generated. This signal, along with the 'machine reset or ingate reset' signal is produced as a result of (1) a 'turn on setup IPL' signal, which gates the unit address used to select the I/O device containing the IPL into the unit address register; (2) the 'fetch CAW' signal (result of a Start I/O instruction), which gates the address of the I/O device that is to perform the instructions into the unit address register; and (3) the 'test I/O or halt I/O and select in' signal in the absence of a 'CPU accept interrupt' signal, which gates the address of the I/O device to be tested or halted into the unit address register. In the latter case, the presence of a 'CPU accept interrupt' signal signifies one of the following: (1) that the CPU has responded to a previous 'interrupt request' signal from the channel; (2) that the channel is stopped with a logout 'interrupt request' pending; (3) that a simulated 'interrupt response' has been initiated at the CLEAR INTERRUPT switch on the CE panel with the channel in the test mode; or (4) that an interface reset is occurring. Any of the above events has precedence over the Test I/O or Halt I/O instruction and 'unit address bus to unit address register' ingating is inhibited accordingly.

When 'unit address bus out' bits are gated to the unit address register, the signal originating the gating causes activation of the 'machine reset or ingate reset' signal and

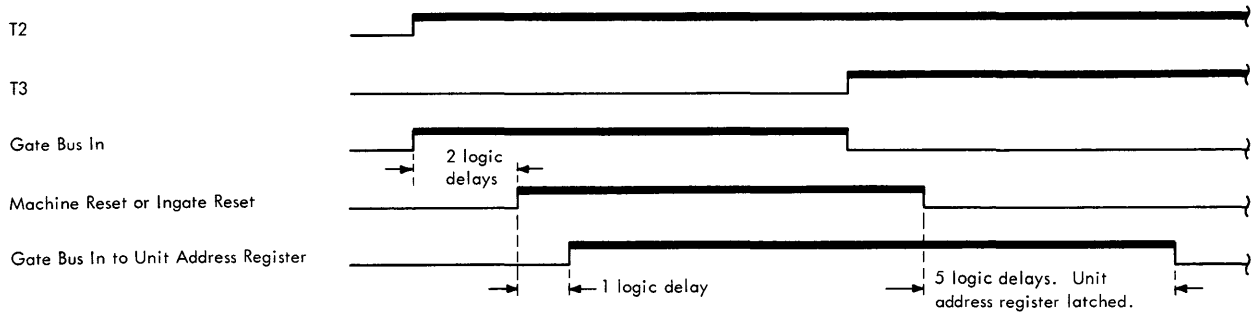


Figure 2-2. Unit Address Register Bus-In Gate Timing

also turns on the 'unit address bus to register' latch. The relative timing of the latch output ('gate unit address out to unit address register' signal) and the reset signal is similar to that previously described for the 'data bus in' gating.

Unit Address Register Logic

- Unit address register consists of nine latches (eight data bits and one parity).
- Parity latches reset to 1 state: data latches to 0 state. Gated logic 1 bit to data latch turns latch on; gated logic 0 bit to parity latch turns latch off.

The unit address register logic is shown in both block and simplified form in Diagram 4-2. The register consists of nine latches, eight data bit latches and one parity bit latch. With the exception of the parity bit latch ('unit address register bit P' latch), all latches are identical. A data latch is set to the 1 state by the coincidence of a logic 1 level and a gating signal at either of the latch ingate ANDs. Prior to ingating, each latch is reset by the 'machine reset or ingate reset' signal. (See "Ingating Logic" for timing of the reset and gating signals.) The parity bit latch is unique in that a reset signal to the unit address register forces a 1 level into the parity bit latch while resetting all other latches to 0; thus, the unit address register maintains odd parity when reset. For ingating operations, the absence of a parity bit (logic 1 level) at the parity bit latch resets the latch and a logic 0 level is reflected at the latch output.

Comparison Logic

- 'Address compare' logic activates 'data bus in compare check' signal when 'bus in' bits and unit address register bits do not match (on bit-to-bit comparison).
- 'Data bus in compare check' signal monitored only during setup operation by 'address in' signal; active signal terminates operation and inactive signal allows operation to continue.
- 'Unit address compare' logic activates 'unit address compare check' signal when unit address bus-out and unit address register bits do not match (on bit-to-bit comparison).
- 'Unit address compare check' signal significant on Test I/O instruction when channel is not working and interrupt is pending. Active signal results in CC2 to CPU; inactive signal causes test I/O operation to clear interrupt.

The 'address compare' logic and the 'unit address compare' logic (Diagram 4-2) each consist of nine sets of exclusive-OR gates which feed a single OR gate. In the 'address compare' logic, each exclusive-OR gate compares a bit from the 'data bus in' lines with a corresponding bit from the unit address register. If the two bits at any exclusive-OR gate fail to match, the 'data bus compare check' signal is activated. The 'data bus compare check' signal is only significant during a setup operation initiated by a start I/O, halt I/O, test I/O, chain command, IPL or FLT operation. The 'data bus compare check' line is monitored after the control unit supplies the channel with the unit address of an I/O device and the address byte has been entered into the channel's 'bus in' latches (upon receipt of the 'address in' signal from the control unit). When monitored, an active 'data bus compare check' line causes turn-on of the 'interface control check' latch to terminate the operation; if the signal line is inactive, the operation continues.

In the 'unit address compare' logic, each exclusive-OR gate compares a bit from the 'unit address bus out' lines with a corresponding bit from the unit address register. If the two bits at any exclusive-OR gate fail to match, the 'unit address compare check' signal is activated. The 'unit address compare check' signal is significant during a Test I/O instruction when an interrupt is pending in the channel and the channel is not operating. If the 'unit address compare check' signal is inactive (indicating that the interrupt in the channel pertains to the device addressed by the Test I/O instruction) when the signal is monitored, the channel generates a 'pseudo accept' signal and clears the interrupt from the channel; if the 'unit address check' signal is active (and significant), condition code 2 is sent to the CPU indicating that the channel contains an interrupt which does not pertain to the device addressed by the Test I/O instruction.

Unit Address Register Outgating Logic

The contents of the unit address register are presented to the 'bus out' latches, the SBI gates, and the 'unit address bus in' gates. The unit address register bits are gated into the 'bus out' latches for subsequent transfer to the control units upon start I/O, test I/O, halt I/O, chain command, IPL and FLT operations (Diagram 4-2). The unit address bits are gated to the SBI lines for storage in main storage during log word 2 of a logout operation or at the end of an IPL operation. Bits 0 through 7 of the unit address register are gated to SBI positions 24 through 31 on logword 2 and IPL endings; the parity bit is gated to SBI position 56 on logword 3 only.

The unit address bits are gated to the CPU interface through the 'unit address bus in' logic (Diagram 4-2) when the CPU raises the 'interrupt response' signal to honor an

'interrupt request' signal previously initiated by the channel. Outgating to the 'unit address bus in' lines is accomplished when the channel receives the 'interrupt response' signal from the CPU; however, outgating is inhibited if the channel is performing a test operation ('simulate CPU' or 'simulate accept interrupt' signals), an interface reset operation is in process, or a logout operation is to be performed ('logout interrupt' signal). To preclude sending a "bad parity" address byte to the CPU, the 'unit address register outgating' logic contains a parity generator which generates a unit address parity generated bit for gating to the 'unit address bus in' parity line. To generate the parity bit, the parity generator receives bits 0 through 7 from the unit address register.

CLOCK

- Clock can generate 8 sequential timing signals ('T0' through 'T7').
- Each signal is 100 ns duration and stays active until clock is turned off.
- Clock turned on when channel operation requires timing signals.
- 'Clock control' latch turned on.
- If 150 ns since last clock sequence and 'stop' signal inactive, 'turn on clock' signal output of 'clock control' latch activates 'T0' signal.
- Conditions activating 'T0' are delayed 100 ns; after delay, 'T1' latch is turned on producing 'T1' signal.
- 'T1' signal delayed 100 ns to turn on 'T2' latch; 'T2' signal delayed 100 ns to turn on 'T3' latch, etc.
- 'Turn off clock' signal activated when channel operation requires absence of timing signals.
- 'Clock control' latch turns off and degates 'T0' signal; 'not T0' degates outputs of 'T1' through 'T7' latches.
- Turn-off of 'clock control' latch triggers 150 ns SS to inhibit 'T1' through 'T7' delay lines and turn off associated latches.
- Active 'stop' signal prevents turn on, turn off, and further generation of clock signals.

The channel clock produces timing signals which provide internal timing for the channel. Clock timing signals control the gating of channel registers and control lines, thereby regulating data flow through the channel.

The clock is capable of producing eight sequential timing signals designated 'T0' through 'T7' (Figure 2-3). Each sequential signal that is generated rises 100 ns after the rise of the preceeding signal and stays active until all activated timing signals are turned off. The status of the clock timing lines is displayed on the CLOCK indicators (T0 through T7) on the CE panel.

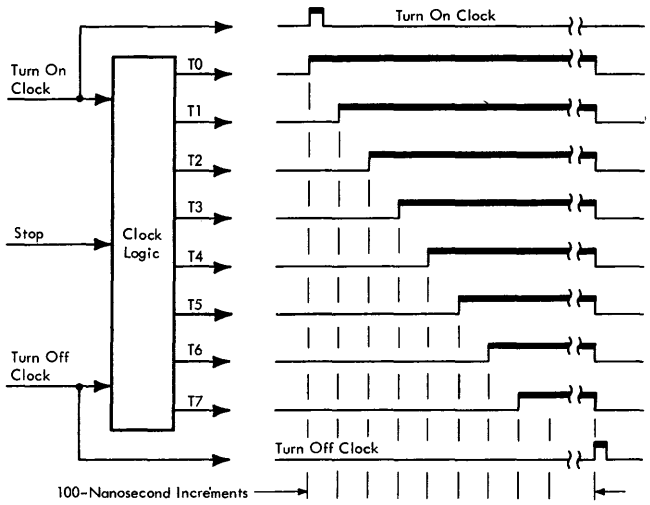


Figure 2-3. Channel Clock Sequencing

Clock operation is initiated by a 'turn on clock' signal which is generated when a channel operation requires timing signals. The clock is turned off by the 'turn off clock' signal when the timing requirements for a particular operation have been satisfied, or when the channel enters a new operation requiring a new timing sequence.

Once started, the clock may be turned off at any time during the generation of timing signals or may remain on an indeterminate time after the T7 signal is generated. For the logout operation and certain diagnostic operations, the clock is prevented from advancing by a 'stop' signal. While the 'stop' level is present the clock is prevented from turning on, turning off, or advancing; clock operation is allowed to resume when the 'stop' level is removed.

Positive logic for the clock is shown in Diagram 4-3. The clock consists of the following logic circuits: (1) a 'clock control latch' and associated logic which control clock turn-on and turn-off times; and (2) seven timing latches (T1 through T7) each of which has an associated input control AND, an input time delay circuit, and an output AND.

The clock is turned on by any of the events listed in Diagram 4-3. Any listed event, in the absence of a 'turn off' signal, turns on the 'clock control' latch. Providing the clock

has been off at least 150 ns since the last clock sequence, the 'turn on clock' level is AND'ed to produce the 'T0' signal. (A 150 ns period between clock turn-off and the next clock sequence permits decay of delay-line signals throughout the clock before a new sequence is initiated.) As long as the 'T0' signal is active and a 'not stop' level is present, the clock will continue to advance until the 'T7' signal is activated. To produce the 'T1' signal, the same conditions which produced the 'T0' signal are AND'ed with the 'not stop' level at the T1 latch 'input AND gate' logic. After a delay of less than 100 ns, the 'input AND gate' signal turns on the T1 latch and the resulting output is AND'ed with the 'T0' signal at the T1 latch' output AND gate' logic to produce the 'T1' signal. In a similar manner, signals 'T2' through 'T7' are produced, with the generation of each signal dependent upon the presence of the preceding timing signal ('T1' signal required to produce the 'T2' signal, 'T2' signal required to produce the 'T3' signal, etc.) and the 'not stop' level.

The clock 'turn off' signal is caused by any of the events listed in Diagram 4-3. The 'turn off' signal turns off the 'clock control' latch, causing the 'T0' signal to drop. When the 'T0' signal drops, the 'output AND gate' logic for each clock latch is disabled, causing all activated timing signals to drop simultaneously. Concurrent with the turn-off of the 'T0' signal, the 'turn off clock' output of the 'clock control' latch fires two 150-ns singleshots. The singleshots produce two 'turn off' signals which reset the 'T1' through 'T7' latches and delay lines and also prevent generation of the 'T0' signal for a 150-ns period. As previously stated, the 150-ns period ensures that all clock delay line signals have decayed before a new clock sequence is started.

The 'stop' signal is generated when the channel is in AUTO mode, the LOG ON MACH CHK switch on the CE panel is down, and a machine check condition resulting from an interface control check or channel control check condition occurs. If the channel is in test mode, the 'stop' signal is activated by either a machine check or channel data check condition and remains activated until manually cleared by the CE. For auto mode operation, the 'stop' signal causes a logout operation. In this case, the clock is stopped (if sequencing when the 'stop' signal is activated) during the logout operation and allowed to resume sequencing at the completion of the operation.

When the 'stop' signal is activated, the 'input AND gate' logic to latches 'T1' through 'T7' is inhibited, and the next clock latch cannot turn on.

Note: If the conditions necessary to satisfy an 'input AND gate' logic are met before the 'stop' signal occurs, the associated clock latch may or may not be turned on. Therefore, the error condition that causes the 'stop' signal may have occurred either at the latest active clock time or at the previous clock time.

Most clock turn-on/turn-off sequences are conditional. That is, once the clock is started, the clock turn off time is dependent upon the course or results of subsequent operations. In other cases, the clock is unconditionally turned on and off by the rise and fall, respectively of the same signal. This signal is designated the 'turn on clock direct' signal and is activated by the following conditions: (1) by a polling interrupt condition when the 'address in' or 'status in' signal is present; (2) by a 'machine reset' signal; or (3) by a Test I/O instruction when the channel has an interrupt condition pending. When activated, the 'turn on clock direct' signal raises the 'turn on' signal to the 'clock control' latch and, after an approximate 40-ns delay, raises the 'turn off' signal to the latch. The early arrival of the 'turn on' signal turns on the 'clock control' latch which remains turned on until the 'turn on' signal falls. While the latch is on, the clock sequences in the manner previously described. When the 'turn on clock direct' signal falls, the 'turn on' level to the 'clock control' latch falls, but the 'turn off' signal is held active for approximately 40 ns (due to logic delays) to turn off the 'clock control' latch. The fall of the 'turn on clock direct' signal also fires the two 150-ns singleshots to turn off any activated clock timing latch.

STORAGE ADDRESS BUS GATING AND CONTROL

The storage address bus (SAB) gating and control circuits control the flow of storage address bits from the channel to the BCU interface. Twenty-one address bits plus three parity bits are provided by the channel for each access to main storage, and the source of address bits within the channel depends upon the operation being performed. The SAB gating and control circuits monitor control signals to determine when address bits are required on the SAB, the type of operation being performed and, consequently, the source of the address bits within the channel.

Five types of addresses are gated by the SAB gating and control circuits: (1) channel address word (CAW) address; (2) channel control word (CCW) addresses; (3) channel status word address (Z-address); (4) data addresses; and (5) logword addresses. The CAW address is fixed (72 decimal or 48 hex) and is gated to the SAB lines during start I/O operations to obtain the CAW which contains the address of the first CCW. The CCW address may specify any nonrestricted storage address, and is gated to the SAB to obtain the CCW required for a current operation. The Z-address is fixed (64 decimal or 40 hex) and is gated to the SAB lines when the channel status word is stored in main storage. Data addresses are gated to the SAB lines on read-type and write-type operations and specify the main storage areas into which data is stored or from which data is fetched. Log word addresses are fixed and specify the storage addresses at which channel log words are to be stored. For each channel

logout operation, three log words (consisting of a double word) are sequentially stored in main storage at addresses 304, 312, and 320 (decimal). These logwords are designated log word 1, log word 2, and log word 3 and the address for each is individually gated to the SAB lines during the logout sequence.

Block Description

- 'SAB gating' logic receives data address register bits; 'gate data address to SAB' signal gates bits to SAB on read-type or write-type data storage cycles or TIC fetch CCW operation.
- 'SAB gating' logic receives command address register bit; 'gate command address to SAB' signal gates bits to SAB when CAW is fetched or other than first CCW is fetched.
- For CSW store operation, 'gate Z address' signal forces address 64 to SAB via 'SAB gating' logic.
- For logout operation, sequential addresses 394, 312, and 320 are forced to SAB via 'SAB gating' logic.
- For CAW fetch cycle, 'gate CAW address' and 'gate command address to SAB' signals force address 72 to SAB via 'SAB gating' logic.

The SAB gating and control circuits (Figure 2-4) consist of the 'data address gating control' logic, the 'command address gating control' logic, the 'Z address gating control' logic, the 'CAW address gating control' logic, and the 'SAB gating' logic. 'Log control' logic supplies gating signals for SAB gating; however, due to the numerous functions of the 'log control' logic, a separate description of this logic is not contained in this paragraph. (Refer to "Logout Operations" in Chapter 6 of this manual for logout address information.)

Data address register bits 0 through 20, P0, P1, and P2 are applied to the 'SAB gating' logic. When the data address register bits are to be gated to the SAB, 'channel control' signals to the 'data address gating control' logic activate the 'gate data address to SAB' signal, causing the data address bits to be gated. The data address register contents are gated to the SAB for storage cycles on which read-type or write-type data transfers occur, upon a fetch cycle to obtain the first CCW of a start I/O operation, or upon a fetch cycle for a TIC operation.

The contents of the command address register are also applied to the 'SAB gating' logic and are gated to the SAB when the 'gate command address to SAB' signal is activated. This signal is activated by the 'command address gating control' logic when 'channel control' signals indicate that (1) the CAW is to be fetched from storage, or (2) that a second or subsequent CCW is to be fetched from storage. In the first case, the command address register is in the reset condition and conains all zeros with good parity (the CAW address is formed by the contents of the command register plus certain bits forced by the 'gate CAW address' signal). In the second case, the command address register contains the updated CCW address previously obtained from the CAW.

For operations involving storage of the CSW, the 'gate Z address' signal from the 'Z address gating control' logic is activated by 'channel control' signals. At the 'SAB gating' logic, the 'gate Z address' signal forces logic 1's into the SAB P0, P1, and bit 17 lines. SAB lines not gated by the 'gate Z address' signal are at a logic 0 level.

For logout operations, the 'log control' logic supplies the 'SAB gating' logic with gating signals for each of the three log words. When the log 1 word is stored, gating signals from the log control logic forces logic 1's into the SAB P0, bit 15, bit 18, bit 19, and P2 lines (address 304). For the log 2 word, gating signals force 1's onto the SAB P0, bit 15, bit 18, bit 19, and bit 20 lines (address 312). For the log 3 word, 1's are forced onto the P0, bit 15, and bit 17 lines (address 320). During the gating of a log word address, any SAB line not gated by a 'log' signal remains at the logical 0 level.

Gating of the CAW address is unique in that both the 'gate command address to SAB' and 'gate CAW address' signals

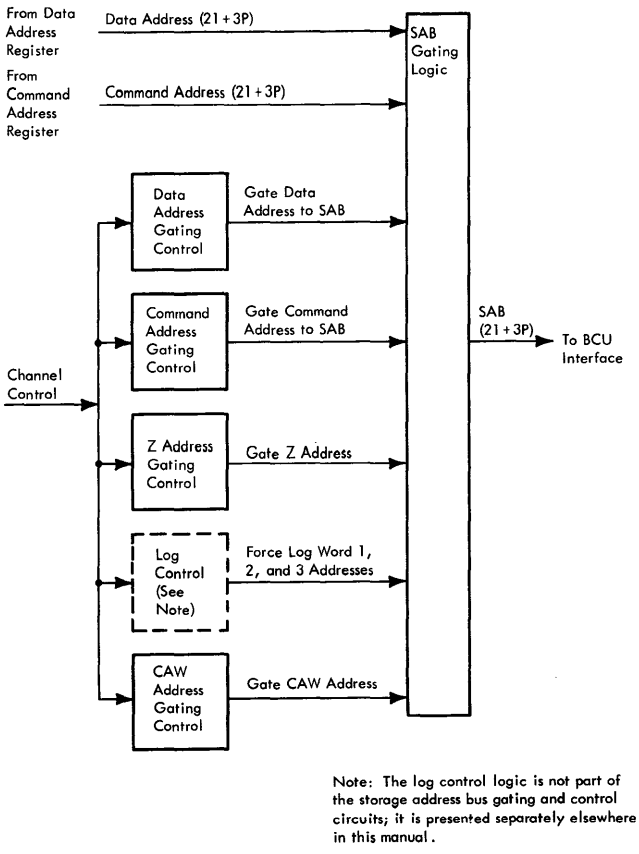


Figure 2-4. Storage Address Bus Gating and Control

are required to gate the proper address (72 decimal) to the SAB. As previously described, the 'gate command address to SAB' signal is produced by the 'command address gating control' logic and, for a CAW fetch cycle, gates the "all 0's with good parity" contents of the command address register to the SAB. At the same time, the 'CAW address gating control' logic produces the 'gate CAW address' signal which forces logical 1's onto the SAB bit 17 and bit 20 lines. Thus, for the CAW address, 1's are present on the three SAB parity lines and on the bit 17 and bit 20 lines, while 0's are present on the remaining lines to provide an address of 72 (decimal).

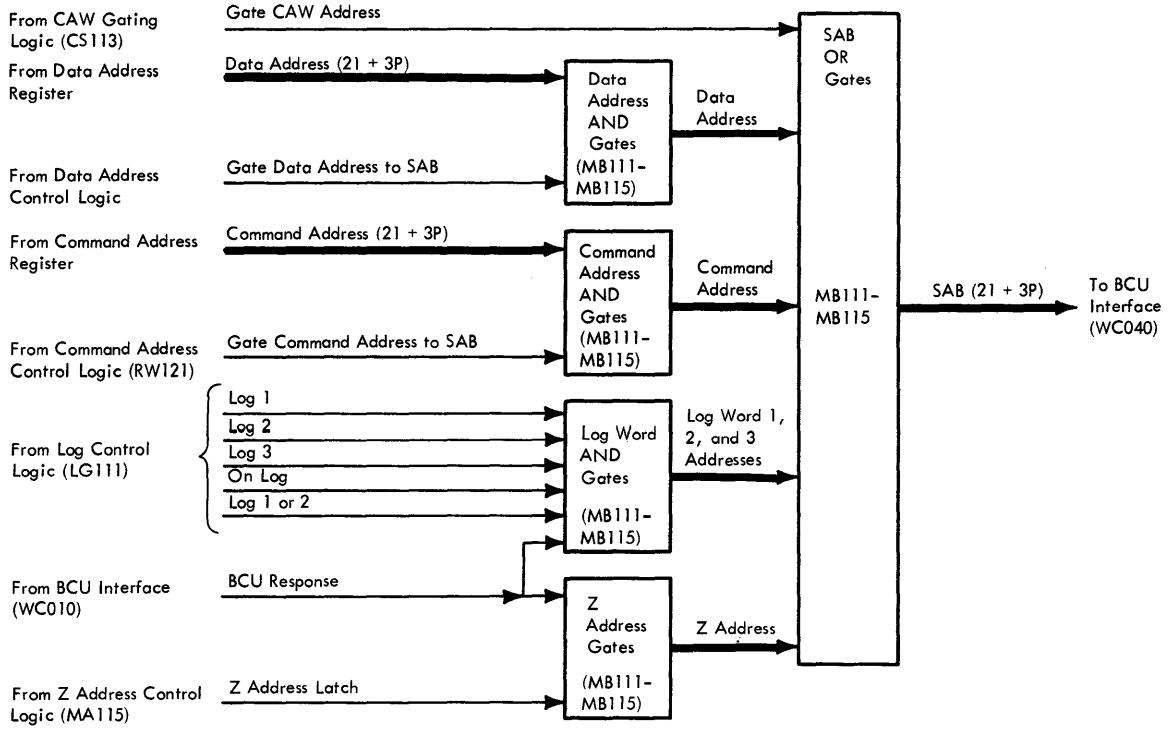
Detailed Description

Detailed and separate descriptions of the 'SAB gating' logic, 'CAW address gating control' logic, 'data address gating control' logic, 'command address gating control' logic, and the 'Z address gating control' logic are provided in the following paragraphs.

SAB Gating Logic

- 'SAB gating' logic OR's addresses to SAB lines for:
  1. CAW fetch cycles (address 72, decimal).
  2. Data store or fetch cycles (data address register contents).
  3. TIC operation to fetch CCW (data address register contents).
  4. Non-TIC CCW fetch operation (command address register contents).
  5. Logout operation (sequential addresses 304, 312, and 320).
  6. CSW storage operation (Z-address, or 64 decimal).

The 'SAB gating' logic (Figure 2-5) consists of a group of OR gates, each of which receives inputs from two or more gated AND's. In Figure 2-5, the AND circuits and signals associated with a particular gating function, i.e., CAW, data address, command address, log address, and Z-address, are shown by functional grouping. Note that the 'BCU



Fixed Addresses in Channel

Name	Decimal	Hex.	SAB Bits Forced to One Level*							
			P0	15	P1	17	18	19	20	P2
Z Address	64	40	1	0	1	1	0	0	0	0
CAW Address**	72	48	1	0	1	1	0	0	1	1
Log Word 1	304	130	1	1	0	0	1	1	0	1
Log Word 2	312	138	1	1	0	0	1	1	1	0
Log Word 3	320	140	1	1	0	1	0	0	0	0

Legend:

* Any SAB bits not listed (0-14 and 16) are 0's for all fixed addresses.

** The CAW address is obtained by gating the command address register (all 0's with good parity) and forcing 1's into positions 17 and 20 with the 'gate CAW address' signal.

Figure 2-5. SAB Gating Logic Block Diagram

response' signal (indicating that a storage request previously initiated by the channel has been honored) is required to gate the 'log word' and 'Z address' AND. The 'BCU response' signal is also required to activate the 'gate data address to SAB', 'gate command address to SAB' and 'gate CAW address to SAB' signals. Therefore, gating to the SAB cannot be accomplished until the unit controlling the BCU interface indicates that the address is required on the SAB lines.

Addresses supplied to the SAB by the data address register and command address register are program variable, but the CAW, log, and Z-addresses are fixed. For each fixed address, the table in Figure 2-5 indicates the logic 1 and 0 levels on the SAB lines. Bits not indicated in the table are at a logical 0 level for all fixed addresses.

Simplified, positive logic for the SAB gating logic is shown in Diagram 4-4. Each SAB line is driven by an OR gate which receives inputs from two or more gated ANDs. In the case of the 'SAB bit 17' and the 'SAB bit 20' lines, the output OR receives the output of an AND/OR combination, as well as the 'gate CAW address' signal. Except for the CAW gating, only one gating signal is present for a given storage operation. For example, during a data read or write operation, only the 'gate data address' signal is generated; therefore, only the bits from the data address register are gated to the SAB. In the case of the CAW address both the 'gate command address to SAB' and 'gate CAW address' signals are generated simultaneously to place logic 1's on the SAB bit P0-P7' (P0), 'SAB bit P8-P15' (P1), 'SAB bit 17', 'SAB bit 20', and 'SAB bit P16-P20' (P2) lines.

CAW Address Gating and Control Logic

The CAW address gating control logic (Figure 2-6) consists of an AND which controls generation of the 'gate CAW address' signal. Since the CAW is fetched from storage location 72 only at the beginning of a start I/O sequence, the 'start I/O latch' and 'not transfer in channel' (TIC) signals are required to condition the AND. (The presence of both the 'start I/O latch' and 'TIC' signals indicates a CCW fetch cycle, rather than a CAW fetch cycle.) Assuming that a CAW fetch cycle is indicated and that the channel is not in the 'simulate storage' test mode, the rise of the 'BCU response' signal causes the 'gate CAW address' signal to rise.

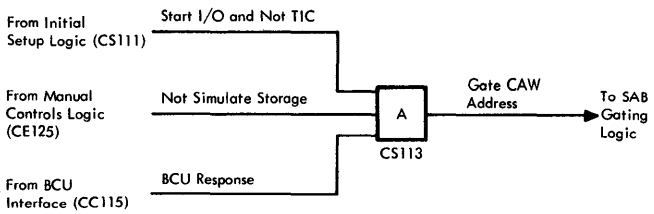


Figure 2-6. CAW Address Gating Control Logic

The fall of the 'BCU response' signal causes the 'gate CAW address' signal to fall. The 'BCU response' signal is provided by the BCU (or equivalent unit) in response to a 'storage request' signal previously issued by the channel.

Data Address Gating Control Logic

- 'Gate data address to SAB' signal gates data address register contents to SAB lines.
- Data Address register contents gated to SAB:
  1. For read or write data doubleword storage accesses.
  2. To fetch first CCW of a start I/O operation.
  3. To fetch CCW for TIC operation.
- For read and write operations, 'gate data address' and 'BCU response powered' signal OR'ed to activate 'gate data address to SAB' signal.
- For fetch first CCW or TIC operation, 'TIC cycle', and 'CCW fetch', and 'BCU response powered' signals AND'ed to activate 'gate data address to SAB' signal.
- For test mode 'manual fetch' (CE panel FETCH switch) or 'manual store' (CE panel STORE switch), 'manual store or fetch' and 'BCU response powered' signals AND'ed to activate 'gate data address to SAB' signal.

The 'data address gating control' logic (Figure 2-7) generates the 'gate data address to SAB' signal on a storage cycle requiring an address from the data address register. In general, the 'gate data address to SAB' signal is generated on (1) a storage cycle involving a data transfer to or from storage (read or write operation), (2) a storage cycle to obtain the first CCW of start I/O operation, or (3) a storage cycle manually initiated with the FETCH or STORE switches on the CE panel when the channel is in test-mode operation.

For normal operation ('not manual store or fetch' operation) data address gating is inhibited when (1) the CSW is to be stored ('Z address' latch on), (2) a log operation is in progress ('inhibit on log' signal is active), or (3) a 'read CDA' or 'write CDA' CCW fetch storage cycle is requested ('retain storage' signal activated). In each of the above cases, the address is gated from an area other than the data address register.

Assume that the channel has not forced 'retain storage' operation ('not turn on CCW fetch and remember BCU response' signal in Figure 2-7). If a storage cycle is requested to store or fetch data for a read-type or write-type operation, a 'gate data address' signal is generated by AND'ing the absence of inhibiting signals ('Z address latch', 'inhibit on log', 'CCW fetch', 'manual store or fetch', and 'remember BCU response and accept'). In turn, the 'gate data address' signal is OR'ed to enable one input to the 'data address' AND. In the absence of a 'retain storage' signal, a second input to

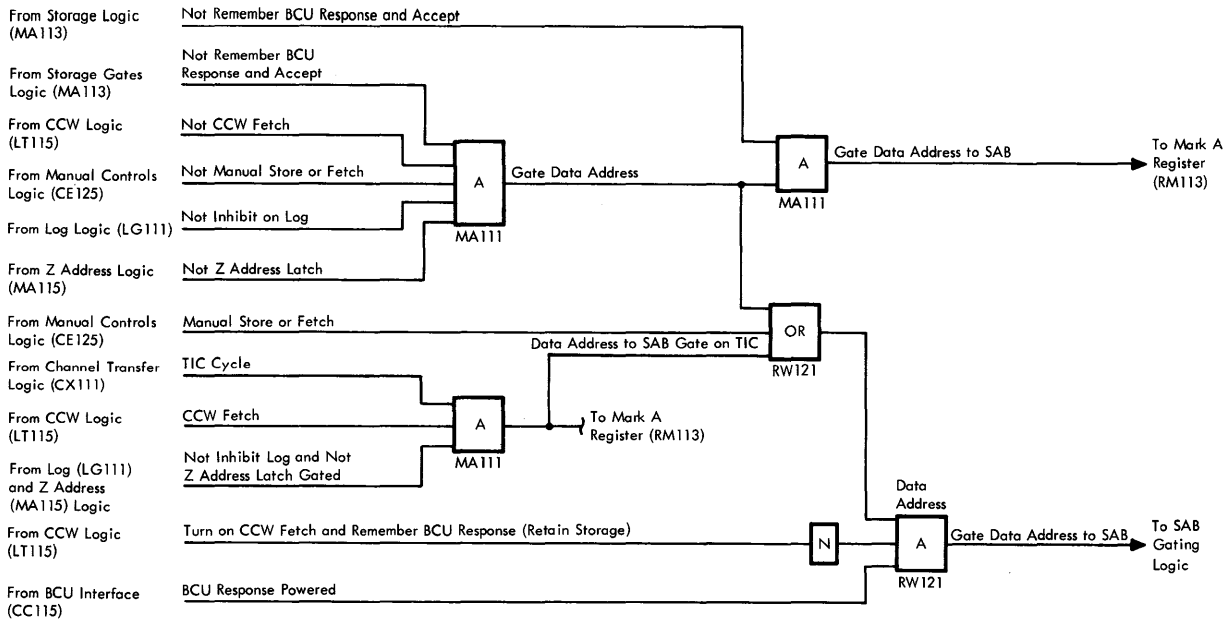


Figure 2-7. Data Address Gating Control Logic

the 'data address' AND is enabled, allowing the 'BCU response powered' signal to generate the 'gate data address to SAB' signal.

When a storage cycle is requested to fetch the first CCW (as a result of an actual TIC operation), the 'data address to SAB gate on TIC' signal is generated by the 'TIC cycle' and 'CCW fetch signals and the absence of the 'inhibit log' and 'Z address latch gated' signals. With the OR satisfied, the 'data address' AND is enabled and the 'BCU response powered' signal generates the 'gate data address to SAB' signal (assuming the 'not retain storage' signal is present).

Note: For test mode simulate storage operation, the 'BCU response powered' signal is inhibited at the channel BCU interface and the 'gate data address to SAB' signal is not generated.

For test mode operation, pressing the CE panel FETCH or STORE switch satisfies the OR to enable the 'data address' AND. The arrival of the 'BCU response powered' signal from the BCU interface in response to the 'storage request' signal (generated when the FETCH or STORE switch was pressed) generates the 'gate data address to SAB' signal as previously described. A second 'gate data address to SAB' signal is generated by the 'data address gating control' logic. This signal is sent to the mark-A register where it is used to gate the contents of the mark-A register to the mark bus.

The 'data address to SAB gate on TIC' signal is also sent to the mark-A register where it forces correct mark parity bits to the mark bus on the initial or TIC CCW fetch cycle. (For the initial or TIC CCW fetch cycle, the mark bus receives logic zero levels on all lines except the parity bit line.)

Command Address Gating Control Logic

- 'Gate command address to SAB' signal gates command address register contents to SAB.
- Signal activated on CAW fetch and non-TIC CCW fetch storage cycles.

The 'command address gating control' logic (Figure 2-8) generates the 'gate command address to SAB' signal on each storage cycle requiring an address from the command address register. Gating of the command address to the SAB lines is required (1) on a CAW fetch storage cycle, and (2) on all CCW fetch storage cycles except the initial CCW fetch following the CAW fetch cycle or a CCW fetch cycle due to a TIC command.

Providing a logout operation ('inhibit on log' level) or CSW store operation ('Z address latch gated' signal) is not in progress, the 'gate command address to SAB' signal is produced by the 'BCU response' signal when the 'CCW fetch' signal and the 'not TIC cycle' signal are AND'ed to enable the 'gate command address' AND. The 'CCW fetch' and 'not TIC cycle' signals are active for both the CAW fetch cycle and all CCW fetches except the initial CCW fetch or a TIC

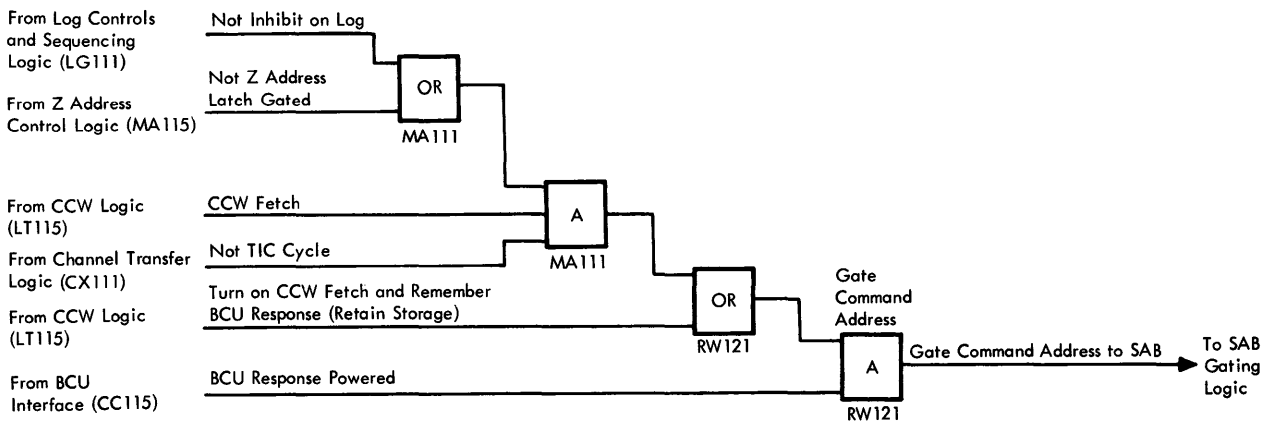


Figure 2-8. Command Address Gating Control Logic

CCW fetch. For read CDA or write CDA operations, the 'retain storage' signal ('turn on CCW fetch and remember BCU response' signal) is active for the CCW fetch cycle, and permits activation of the 'gate command address to SAB' signal when the 'BCU response' signal is present.

Z-Address Gating Control Logic

- 'Z address latch' signal forces address 64 to SAB lines for CSW store operation.
- 'PCI or interrupt', 'CPU interrupt response', 'not setup', 'not CDA latch', and 'not CCW fetch gated' signals activate 'Z address latch' signal.
- 'BCU response' signal turns on 'latch Z address' latch.
- 'Z address' latch turns on: address 64 sustained on SAB lines.
- 'Z address latch gated' signal gates all mark bits to mark lines.
- 'Storage cycle complete' signal turns off 'latch Z address' and 'Z address' latches; address 64 and mark bits degated.

The 'Z address gating control' logic (Figure 2-9) generates a 'Z address latch' signal when the CSW is to be stored in main storage. The 'Z address latch' signal forces address 64 to the SAB lines.

The 'Z address gating control' logic contains a 'Z address' latch, a 'latch Z address' latch, and a 'block Z address' latch. When the 'Z address' latch is on, the address 64 (decimal) is forced onto the SAB. To store the CSW, the channel must either (1) send an 'interrupt request' signal to the CPU and await the return of an 'interrupt response' signal before requesting a storage cycle, or (2) force a CSW storage cycle without referring to the CPU. In either case, the channel activates the 'PCI or interrupt' signal to enable one input to the 'set Z address' AND. When the 'CPU interrupt response' signal is raised (either as a result of the 'interrupt response' signal from the CPU or the 'pseudo accept interrupt' signal forced by the channel), a second input to the 'set Z address' AND is enabled.

Providing channel operations ('setup' signal active, 'chain data' latch on, or 'CCW fetch gated' signal active) do not preclude a CSW store operation, the 'set Z address latch' signal is activated and OR'ed to produce the 'Z address latch' signal. The 'set Z address latch' signal is also applied to the 'Z address' latch, but is unable to turn on the latch until the 'latch Z address' latch is turned on. The 'latch Z address' latch is turned on by the 'BCU response' signal to signify that the storage request to store the CSW has been honored. With the 'latch Z address' latch on, the reset level to the 'Z address' latch is removed, allowing the latch to be turned on by the 'set Z address latch' signal. With both the 'latch Z address' latch and the 'Z address' latch on, the 'Z address latch' signal to the 'SAB gating' logic is sustained. (To force the Z-address to the SAB, the 'Z address latch' signal is gated at the 'SAB gating' logic by the same 'BCU response' signal that sets the 'latch Z address' latch.) In addition to forcing the address to the SAB, the 'Z address latch' signal is supplied to the 'channel storage control' logic, where it is used to produce other channel gating signals required during storage of the CSW, and to the 'polling interrupt release' logic where it enables generation of a 'release' signal to the CPU.

A second 'Z address control' logic output, the 'Z address latch gated' signal, is produced by AND'ing the on conditions of the 'Z address' latch and the 'latch Z address' latch. The 'Z address latch gated' signal is used at the mark-A register to force mark bits to all positions of the mark bus. In addition, the 'Z address latch gated' signal (1) resets the 'suppress out' latch, (2) enables the channel to gate the 'control unit status' and 'channel

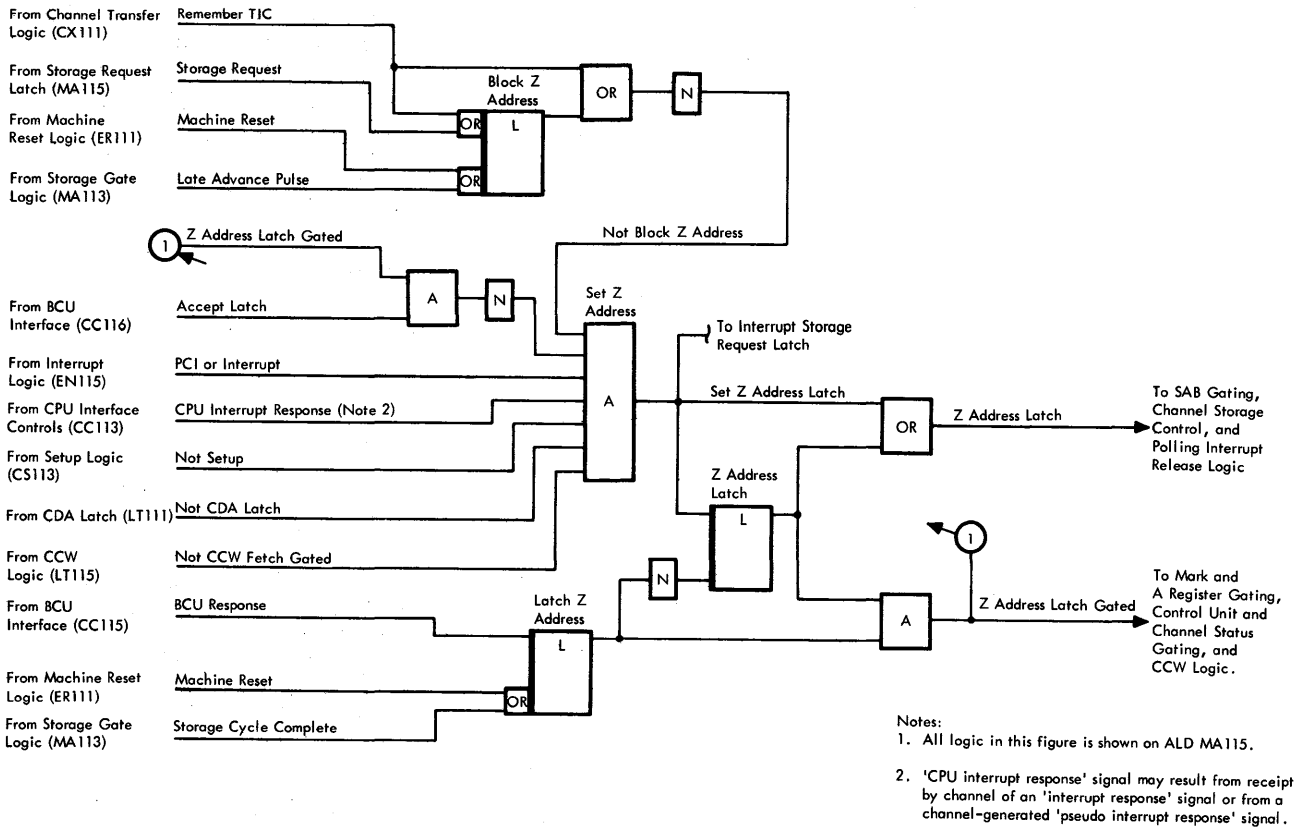


Figure 2-9. Z-Address Gating Control Logic

status' bytes to the SBI lines, and (3) inhibits the gating of the command address register contents to the SAB lines. In the 'Z address control' logic, the 'Z address latch gated' signal is applied to an AND which also receives the 'accept latch' signal as a second input' Coincidence of the two signals inhibits the 'set Z address' AND to remove the 'set Z address latch' level; however, the 'Z address' latch remains on until the 'latch Z address' latch is turned off at the end of the CSW storage cycle, or as a result of a 'machine reset' signal.

The 'block Z address' latch blocks the turn-on of the 'Z address latch' (1) when a 'storage request' signal has been issued for any purpose other than to store the CSW, or (2) when a TIC operation is in progress. In the first case, the output of the 'storage request' latch turns on the 'block Z address' latch to inhibit turn-on of the 'Z address' latch. (Storage requests to store the CSW are processed by the 'interrupt storage request' latch rather than by the 'storage request' latch.) For a TIC operation, the 'block Z address' latch is turned on by the 'remember TIC' signal to inhibit gating of the Z address to the SAB lines. (On TIC storage cycles, the data address register contents are gated to the SAB.) To block activation of the 'Z address latch' signal at the earliest possible time after the TIC cycle begins, the 'remember TIC' signal is OR'ed to independently generate the 'block Z address' signals prior to the time the 'block Z address' latch is turned on.

STORAGE BUS OUT LOGIC

- SBO logic ingating sources are:
  1. SBO lines from main storage.
  2. CE panel SBO switches.
  3. Mark-B register.

- SBO data from storage is gated to channel SBO lines for all operations where information is fetched from storage.

- CE panel SBO switch contents are gated to channel SBO lines for 'manual store', 'simulate store', and 'load data address' test mode operations.

- Mark-B register contents are gated to channel SBO lines when a wrong length record condition occurs during a write CDA operation.

- Zero tests are performed during CAW and CCW fetches by testing groups of bits on the SBO lines for all 0's.
- For a CAW fetch cycle, the validity of the storage protect key and the CCW addresses is checked by zero tests.
- For a TIC CCW fetch cycle, the validity of the next CCW address is checked for zero testing.
- For a non-TIC CCW fetch cycle, the validity of flag information and the count field bits is checked by 0 testing.
- Gated SBO bits are applied to ingates of:
  1. A-register.
  2. Storage protect register.
  3. Command register.
  4. Data address register.
  5. Flag register.
  6. Count register.
- SBO gating into the registers is function of operation being performed in channel.

The SBO logic (Figure 2-10) controls the gating of SBO bits into the channel and performs tests to determine whether specified groups of SBO bits contain all 0's. SBO bits may be gated from one of three sources: (1) the 72 SBO multiplex lines from main storage; (2) 72 SBO switches on the channel CE panel; and (3) the mark-B register. Ingating is controlled by signals generated by the channel control logic. After ingating, the SBO bits are distributed to individual channel registers where they are further gated as required by the operation in progress. In addition, certain groups of the gated SBO bits are tested at the 'SBO zero check' logic for the presence of binary 1's; the results of the zero tests are sent to the channel status logic where they are gated (if so required by the operation in progress) to determine if an error exists. An error detected by an SBO zero test sets the 'program check' latch in the channel.

The SBO logic is shown in positive logic form in Diagram 4-5. SBO gating is accomplished at three sets of AND's; one set of 72 AND's for the SBO bits from storage (channel SBO bits), one set of 62 AND's for the CE panel switches ('SBO switch' lines), and one set of 6 AND's for the mark-B bits from the mark-B register. Outputs from the three sets of AND's are routed to 72 OR's. The outputs of the OR's are distributed to channel circuits.

Channel SBO bits are AND'ed to the 'SBO bits' lines at all times except when the channel is operating in the test mode, or a wrong length record condition occurs during a 'write chain data' (write CDA) operation. The 'channel SBO' bits are gated by the 'not simulate storage or manual store or load data address' level, which enables all ANDS receiving the 'channel SBO' bits.

For certain test mode operations, bit values set into CE panel SBO switches are gated through the SBO logic. Test mode operations that cause the SBO switch contents to be gated (and, consequently, inhibit 'channel SBO' bits from being gated) are: (1) simulate storage, which results when the CE panel SIM STOR switch is on (down); (2) manual store, which is initiated by pressing the CE panel STORE switch; and (3) load data address, which is initiated by pressing the CE panel LOAD DATA ADDRESS switch. Any of the above test-mode operations activates the 'simulate storage or manual store or load data address' signal to enable the AND's conditioned by the settings of the SBO switches. With these gates enabled, the values in the following CE panel switches are gated to the indicated SBO lines within the channel:

Switches	SBO Lines
COMMAND FIELD	P0, 1-7
ADDRESS FIELD	P1, P2, P3, 8-31
FLAG FIELD	P4, 32-39
P5, 40-47	P5, 40-47
COUNT FIELD	P6, P7, 48-63

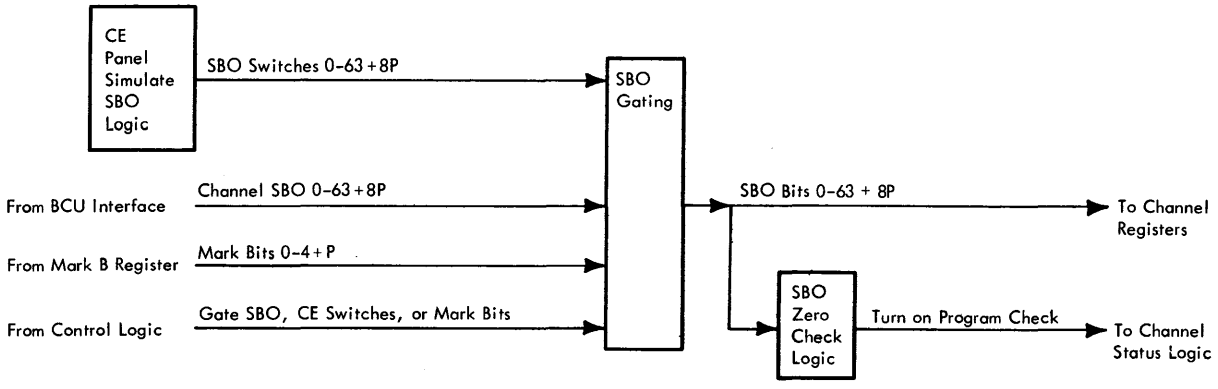


Figure 2-10. Storage Bus-Out Gating Logic, Block Diagram

Gating from the mark-B register through the SBO gates is performed when a wrong length record condition is recognized during a write CDA operation. On a write CDA operation, the channel fetches a new CCW before the count from the old CCW is exhausted. To clear the count register for the new CCW count and yet retain the old count until completion of the write operation specified by the old CCW, the old count is transferred to the five highest-order positions of the mark-B register. If the old count write operation ends with a wrong length record condition, the mark-B register contents must be transferred back to the count register so that the channel may calculate a residual count value; the residual count is placed in the CSW. In this case, the 'write and wrong length record' signal inhibits gating of 'channel SBO' and 'SBO switches' bit values, while the 'write CDA and wrong length record' signal gates the bits from the mark-B register to the channels 'SBO bits' lines. Bits 0 through 4 and the parity bit from the mark-B register are gated to SBO lines 63 through 59 and P7, respectively.

Zero tests are performed upon four groups of SBO bits, and the result of each test is significant only for a specified channel operation. Failure of a zero test results in a program check condition. The four zero tests, the operation during which the tests are significant, and the test results required to prevent a program check condition follow:

SBO Bit Group	Operation	Normal Results (No Error)
4 through 7 and 29 through 31	CAW fetch cycle	All 0's
29 through 31	TIC CCW fetch cycle	All 0's
37 through 39	Non-TIC CCW fetch cycle	All 0's
48 through 63	Non-TIC CCW fetch cycle	Not all 0's

The reasons for each zero test are described separately in the following paragraphs.

For a CAW fetch cycle, the zero test (bits 4 through 7 and 28 through 31) checks the validity of (1) the storage protect key and (2) the CCW address contained in the CAW. In the first case, the storage protect key is contained in bits 0 through 3 of the CAW and, consequently, is present on SBO lines 0 through 3 on a CAW fetch cycle. Further, the storage protect key is gated into the storage protect register and on subsequent store or fetch operations is parity checked before being sent to storage. Since bits 4 through 7 and P0 of the CAW are also entered into the storage protect register for parity checking purposes but do not form a part of the storage protect key, these bits (4 through 7) are programmed for logic 0 values. Should a binary 1 be present in bits 4 through 7, subsequent parity checks of the storage protect register contents would be invalid; therefore, if the zero test detects a binary 1 in bits 4 through 7 on a CAW fetch operation, the 'program check' latch is turned on to preclude a possible illegal access to storage on future store or fetch operations. In the second case, the CCW address is contained in bits 8 through 31 of the CAW. As an architectural requirement, the storage address of a CCW must be located on a doubleword boundary. To address storage on doubleword boundaries, the three low-order bits of the storage address must be 0's. Thus, the three low-order CCW address bits (CAW or SBO bits 29 through 31) are zero tested for the doubleword criteria. If the bits are not 0's the 'program check' latch is turned on to end the operation.

On a TIC CCW fetch cycle, the zero test checks the validity of the next CCW address. The TIC CCW on the SBO lines contains the address of the next CCW in bits 8 through 31. Since the doubleword boundary criteria applies to all CCW addresses, low-order bits 29 through 31 are tested and must be all 0's if the CCW address is to be considered valid. If the zero test indicates other than all 0's, the 'program check' latch is turned on to end the operation.

On a non-TIC CCW fetch cycle the validity of the flag bits and byte count bits contained in the CCW is checked by performing zero tests. The flag field is contained in CCW bits 32 through 36. On a CCW fetch cycle, these bits are entered into the flag register. However, bits 37 through 39 are also entered into the flag register for parity checking purposes and are preprogrammed for logic 0 values. Thus, a binary 1 detected for bits 37 through 39 indicates a program error and the 'program check' latch is turned on accordingly. Zero testing of the count bits (48 through 63) is performed to ensure that at least one data transfer is specified by the CCW (as is the case when one or more logic 1's are contained in the count bits). Since a non-TIC CCW usually specifies a data transfer and a zero count specifies that no bytes are to be transferred, the two events are incompatible and the 'program check' latch is turned on accordingly.

Note: On a TIC CCW, the flag and count registers are not loaded, and zero tests on SBO bits to these registers are not significant.

Gated SBO bits from the SBO logic are supplied to gating inputs of several channel registers. Depending upon the operation in progress, specific SBO bits are gated into these registers as required. Registers receiving SBO bits, the SBO bits associated with each register, and the operations during which SBO bits are gated into the registers are as follows:

Register	SBO Bits	SBO Gating to Register
A-register	0-63 (+8P)	Upon data fetch for write operations
Store protect register	0-7 (+P0)	Upon a CAW fetch
Command register	0-7 (+P0)	Upon a CCW fetch
Data address register	8-31 (+3P)	Upon a CAW or CCW fetch
Flag register	32-39 (+P)	Upon a non-TIC CCW fetch
Count register	48-63 (+2P)	Upon a non-TIC CCW fetch

For more information regarding gating to channel registers, refer to the individual descriptions of these registers in this chapter.

STORAGE PROTECT REGISTER

- The storage protect register and associated logic furnishes a four-bit (plus parity) storage protect key to storage on each storage access.
- The key furnished must be proper if storage access is to be successful.

The storage protect register and associated logic (Figure 2-11) furnishes a storage protect key (four bits plus parity) to main storage on each storage access by the channel. In main storage, the storage protect key is examined to determine if the storage area addressed by the channel may be legally accessed. If the storage protect key is not proper, access to the storage area is not permitted and a 'storage protect check' signal is generated and supplied to the channel.

Block Diagram Description

- A storage protect key (four bits plus parity) is sent to main storage by the channel on each storage cycle to gain access to protected storage areas.
- A storage protect key is obtained by the channel and entered into the storage protect register on each CAW fetch cycle.
- The storage protect register receives nine bits, with bits 0 through 3 (plus a parity bit) forming the storage protect key, and bits 4 through 7 (all 0's) provided for parity checking purposes.
- On storage cycles involving CAW fetches, logout, manual store, CSW store, and IPL end condition store operations, a master key of all 0's with good parity is gated to storage, rather than the key from the storage protect register.
- Bits 0 through 3 plus parity from the storage protect register are stored in main storage (via the SBI lines) on logout operations.
- Bits 0 through 3 from the storage protect register are stored in main storage (via the SBI lines) on a CSW store operation.
- Storage protect register parity is checked at the end of the CAW fetch cycle, and results in a 'channel control check' condition if odd parity is not present.

The storage protect register and associated logic (Figure 2-11) consists of a nine-position register (eight bits plus parity), ingating AND's, 'parity checking' logic, 'storage

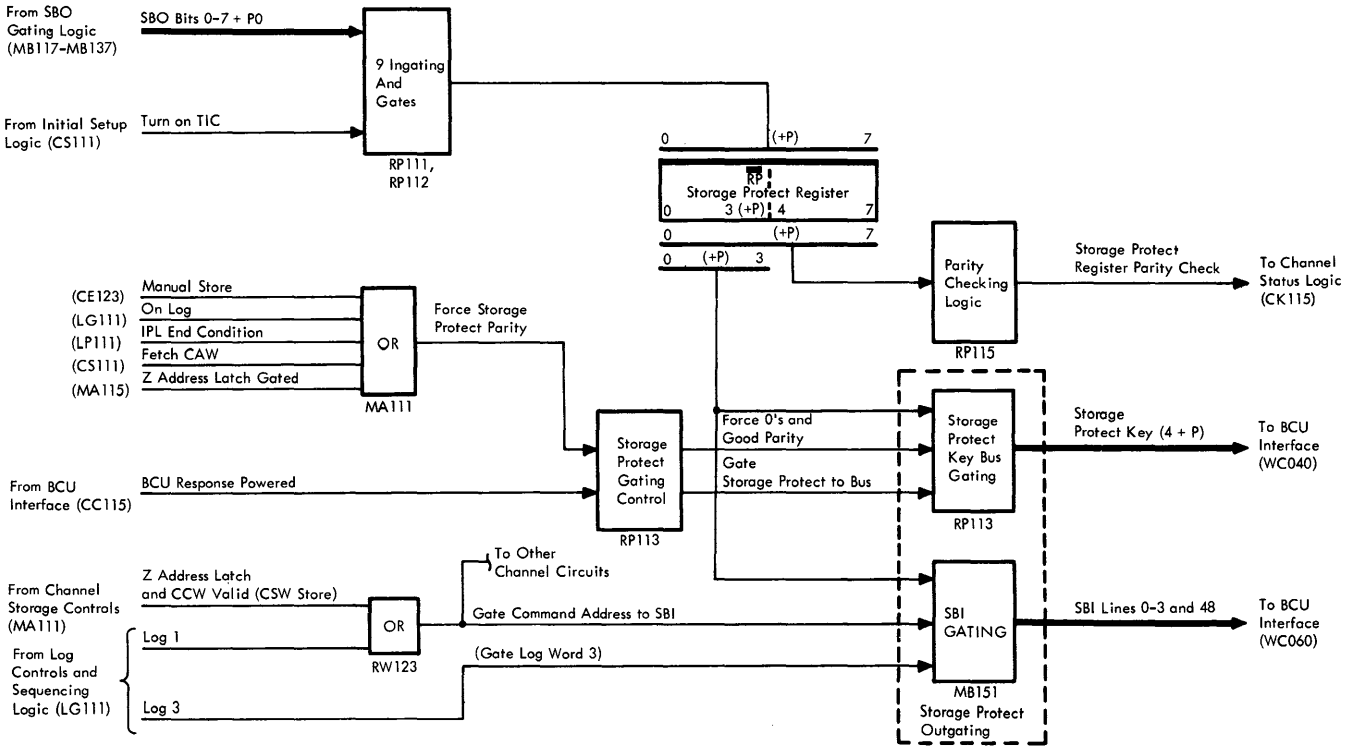


Figure 2-11. Storage Protect Register and Associated Logic, Block Diagram

protect key bus gating' logic, and a portion of the SBI gating logic. Additional logic includes the 'gating control' logic. Five positions of the storage protect register (bits P0 and 0 through 3) have indicators on the CE panel. Indicators are not furnished for the other positions because bits in these positions are not part of the storage protect key.

'Storage protect key' bits are gated through the 'storage protect key bus gating' logic to the channels BCU interface for transfer to main storage. The key gated to the BCU interface may be obtained by forcing all zeros with good parity (designated the master key) through the 'storage protect key bus gating' logic. Whether the contents of the storage protect register or the master key is gated depends upon the purpose of the storage access.

In general, the master key is sent to storage when the channel must have access to storage, but the key in the storage protect register is not necessarily valid. The key in the storage protect register may not be valid because of the manner in which the key is gated into the register. Gating into the storage protect register is accomplished only on a CAW fetch cycle following a Start I/O instruction from the CPU. Once the register is loaded, a new key cannot be obtained until a new Start I/O instruction is received by the channel, and a subsequent CAW fetch cycle is performed. (Subsequent data transfer or command chaining operations do not change the key.) Thus, when the channel accesses storage to obtain a CAW, to store log words, to store a CSW, or to store the channel and I/O addresses on an IPL end condition operation, the storage protect key in the register may not be valid and thus would not permit access to storage. Since a master key of all zeros with good parity always allows storage access, the master key is sent to storage for storage operations described above. In addition, the master key is also sent to storage on all test mode manual store operations performed by the channel, since it is not necessary to fetch a CAW (and consequently, a storage protect key) to perform manual store operations.

When the channel receives a Start I/O instruction and subsequently accesses storage to obtain the CAW, the 'fetch CAW' signal is OR'ed to produce the 'force storage protect parity' signal (Figure 2-11). The 'force storage protect parity' signal inhibits outgating of the storage protect register contents, thus forcing zeros to bit positions 0 through 3 of the 'storage protect key' lines. At the same time, the 'force storage protect parity' signal, in coincidence with the 'BCU response' signal from the BCU interface, forces a logic 1 onto the 'storage protect key' parity bit line. When the CAW subsequently becomes available and is gated through the SBO logic, CAW (or SBO) bits 0 through 7 plus P0 are gated into the storage protect register by the 'turn on TIC' signal.

Note: Bits 4 through 7 do not form part of the storage protect key. These bits are all zeros and are gated into the storage protect register only for parity checking purposes. While these bits are on the SBO, they are examined for an all zeros condition; if the bits are not all zeros, the channels 'program check' latch is turned on and the current operation is terminated.

The 'turn on TIC' signal is active only when CAW data is on the SBO; i.e., when the 'start I/O' latch is turned on and the delayed 'BCU advance pulse' signal is active to indicate that SBO data is present for gating. Once the storage protect key has been gated into the storage protect register, the channel uses the storage protect key when accessing storage (except when the master key is used, as previously described). The key (bits 0 through 3 plus parity) from the storage protect register is gated through the 'storage protect key bus gating' logic by the 'BCU response' signal in the absence of a 'force storage protect parity' signal.

Note: The 'BCU response' signal is received by the channel at the beginning of each storage cycle.

To summarize gating from the storage protect register, the storage protect key in the register is gated to storage on all data transfers to and from storage except those transfers involving manual store, logout, fetch CAW, CSW store, and IPL end condition operations.

On logout operations and CSW store operations, the storage protect key (bits 0 through 3 plus parity) is gated to the SBI lines for storage at logword or CSW addresses.

For logout operations, the 'log 1' gating signal is OR'ed to produce a 'gate command address to SBI' signal. This signal gates bits 0 through 3 from the storage protect register to SBI lines 0 through 3 for storage in address location 304 of main storage. When the third log word is stored, the 'log 3' gating signal gates the parity bit from the storage protect register to SBI line 48 for storage in address location 320 of main storage.

For CSW store operations, the 'Z address latch and CCW valid' signal is OR'ed to activate the 'gate command address to SBI' signal. This signal gates bits 0 through 3 of the storage protect register to SBI lines 0 through 3 for storage in address location 64 in main storage.

Parity checking of the storage protect register is performed by the parity checking logic to determine if correct (odd) parity has been maintained in the transfer of the storage protect key into the register. If the register contains an even number of 1's the 'SPR parity check' signal is activated. At the end of the CAW fetch cycle (after the key has been gated into the storage protect register), the 'SPR parity check' signal is strobed and, if active, turns on the 'channel control check' latch which causes termination of the current start I/O operation.

Note: The 'SPR parity check' signal is strobed at the end of each storage cycle. However, since the storage protect register bits remain unchanged after the CAW fetch cycle, the

possibility of an 'SPR parity check' condition occurring on other than the CAW fetch cycle is remote.

Detailed Logic Description

- Storage protect register consists of nine flip latches.
- Gating into the register is accomplished by AND'ing the 'turn on TIC' signal with SBO bits P0 and 0 through 3.
- The register is reset by a 'start I/O reset' or a 'machine reset' signal.

Simplified logic for the storage protect register and associated logic is shown in Diagram 4-6. The storage protect register is composed of nine flip-latches. An AND associated with the 'turn on' input to each latch receives the 'turn on TIC' signal and the appropriate SBO bit. For example, the 'bit 0' latch receives SBO bit 0 and the 'bit 1' latch receives SBO bit 1. Each latch (except the 'parity bit P0' latch) is reset by either a 'start I/O reset' signal or the 'machine reset' signal. The 'parity bit P0' latch is reset only by the 'start I/O reset' signal.

Storage Protect Register Ingating on CAW Cycles

- On a CAW fetch cycle, the 'BCU response' signal:
  1. Resets the storage protect register latches.
  2. Forces a logic 1 to the 'storage protect tag parity' line so that the master key (all zeros) required for the CAW fetch cycle has correct parity.
- CAW bits P0 and 0 through 7 are gated into the storage protect register as a result of the delayed 'BCU advance pulse' signal from storage.
- Storage protect register bits P0 and 0 through 3 form the storage protect key.
- All storage protect register bits are checked for odd parity at the end of the CAW cycle.
- A 'storage protect parity check' condition turns on the 'channel control check' latch to terminate the start I/O operation.

Assume that the channel has received a Start I/O instruction ('start I/O' latch is on) and has requested a storage cycle to fetch the CAW. When the CAW storage request is honored, the channel receives a 'BCU response' signal (Diagram 4-6) which is AND'ed with the 'start I/O' (and 'not TIC') level to generate the 'start I/O reset' signal. The 'start I/O reset' signal is OR'ed to reset (turn off) all latches of the storage protect register. At the same time, the 'BCU response powered' signal is AND'ed with the 'force storage protect parity' signal (generated by the 'fetch CAW' signal) to force a logic 1 to the 'storage protect tag parity' line. The presence of the 'force storage protect parity' signal also inhibits the bit 0 through bit 3 ANDs, causing logic 0's to appear on the 'storage protect tag 0 through 3' lines. Thus, on the CAW fetch cycle, the 'BCU response' signal resets the storage protect register latches and also forces correct parity for the master key (all zeros) present on the 'storage protect tag' lines.

The 'BCU response' signal is present for a sufficient duration (approximately 500 ns) to ensure that the master key is present on the 'storage protect key' lines for the required time. However, the 'BCU response' signal falls, removing the 'reset' signal from the storage protect register prior to the time CAW bits are gated into the storage protect register latches. Approximately 200 ns before the CAW data is available at the channel, the channel receives a 'BCU advance pulse' signal signifying that the data is forthcoming. The 'BCU advance pulse' signal is delayed and used to generate the 'turn on TIC' signal. The 'turn on TIC' signal is active while the CAW data is present on the SBO lines and gates SBO bits P0 and 0 through 7 into storage protect register latches P0 and 0 through 7 respectively.

Although only bits P0 and 0 through 3 form the storage protect key, the outputs of all storage protect register latches are supplied to exclusive-OR circuits and checked for odd parity. If parity is not odd, the 'SPR parity check' signal is activated. At the end of the CAW fetch cycle, the 'SPR parity check' signal is examined (by the channel status logic) and if the signal is active, the 'channel control check' latch is turned on to terminate the start I/O operation.

If a 'machine reset' signal occurs, storage protect register latches 'bit 0' through 'bit 7' reset to logic 0's and the 'parity bit P0' latch is set to a logic 1 to maintain odd parity.

Storage Protect Key Bus Gating

- For 'CAW fetch', logout, 'IPL end condition', 'CSW store', and 'manual store' storage cycles, a master key of all zeros with good parity is supplied to main storage by the channel.
- For all other storage cycles requested by the channel, the key bits in the storage protect register are gated to main storage.

The 'storage protect key' bus consists of five lines designated storage protect tag parity, and 0 through 3. If a particular storage cycle is not a CAW fetch, logout, IPL end condition, CSW store ('Z address latch gated' signal active) or manual store operation, the absence of the 'force storage protect

parity' signal enables an AND (Diagram 4-6) which also receives the 'BCU response' signal. When the 'BCU response' signal is active, the five AND's to the 'storage protect tag' lines are enabled, and the contents of storage protect register latches P0 and 0 through 3 are gated to the 'storage protect tag' lines. If gating from the storage protect register is inhibited by one of the operations previously designated, the 'BCU response' signal forces a logic 1 to the 'storage protect tag parity' line, and the remaining 'storage protect tag' signals are logic 0's.

SP Gating on Logout and CSW Operations

- On a logout operation, storage protect register bits 0 through 3 are stored as part of log word 1 and the parity bit as part of log word 3.

- On a CSW store operation, storage protect register bits 0 through 3 are stored as part of the CSW.

On logout and CSW store operations, the storage protect key contained in the storage protect register is stored in main storage. For logout operations, the 'log 1' signal is OR'ed and applied to four AND's which also receive bits 0 through 3 from the storage protect register. The outputs of the four AND's are designated SBI bits 0 through 3. Since logword 1 is stored in address location 304 in storage, the storage protect key is stored in the first 4 bit positions of that byte address. On logword 3, the 'gate logword 3' signal gates the 'parity bit P0' bit to SBI position 48. Since logword 3 is stored at address 320 in main storage, the parity bit is stored in the last bit position of the sixth byte. On a CSW store operation, the 'Z address latch CCW valid' signal is OR'ed to gate storage protect register bits 0 through 3 to SBI lines 0 through 3. Since the CSW is stored in storage location 64, the storage protect key is stored in the first four bits of that address. The key parity bit is not stored on a CSW store operation.

COMMAND REGISTER

- Command register stores command code bits from CCW.

- Command code specifies I/O device and channel operation.

- Command register consists of nine latches with CE panel indicators.

- SBO bits 0-7 are gated into command register on first CCW fetch.
  a. If command chaining is specified, a new command is gated into register each CCW fetch cycle.
  b. If data chaining is specified, command gated into register on first CCW is retained throughout chaining operation.

- Command register parity is checked at 'bus out' latches when command byte is gated to control unit.

- Read command is forced into command register during IPL operation.

- Command register contents are gated to SBI on logout sequence.

The function of the command register is to store command bits received in the CCW; command bits are gated from the command register to the control unit (via the 'bus out' latches) and specify the operation to be performed by the I/O device. In addition, bits from the command register are decoded by the channel to determine whether the command code specifies an invalid, read-type, read backward, or write-type operation.

The command register (Diagram 4-7) consists of nine latches (eight data and one parity) with COMMAND REGISTER indicators on the CE panel for each latch. Command register parity is checked at the 'bus out' latches when the command byte is sent to the control unit.

The command register receives command bits from the SBO bit 0 through SBO bit 7 and bit P0 lines. Commands are first gated into the command register (with the 'raw advance SS' signal) when the first CCW of a start I/O operation is available on the SBO lines. As soon as the command is gated into the register, 'command decoding' logic in the channel (Diagram 4-7) decodes significant bits of the command to check for an invalid or read backward code. If the code is invalid (bits 4 through 7 are all zeros), the channel 'program check' latch is turned on and the operation is terminated. If the channel decodes a Read Backward command (Diagram 4-7), control logic in the channel performs two significant operations: (1) complements the DAB before it is entered into the adder and byte counter; and (2) sets up data address gating logic to decrement the data address rather than increment the address on an update operation.

After the channel determines that the CCW is valid and that no errors exist, the 'gate command' latch is turned on and strobes ANDs in the 'command decoding' logic (Diagram 4-7) to determine if the command code specifies a read-type or write-type operation. A read-type operation is specified if command register bits 5 or 6 are active and bit 7 is inactive. A write-type operation is specified if command register bit 7 is active. Read (RD) and write (WR) indicators are located on the CE panel to indicate the decoding results.

If chain command operations are being performed by the channel, new commands are gated into the command register

on each CCW fetch. If chain data operations are being performed, the command gated into the register upon receipt of the first CCW in the chain of CCW's remains valid, regardless of the number of CCW's fetched during the chaining operation.

For an IPL operation, a read command is forced into the command register. This results when the 'force read operation' signal is activated to turn on the command register bit 7 and parity latches. (Prior to the read command being forced, the command register is reset to all zeros with odd parity by the 'machine or ingate reset' signal.)

Of the nine command register latches, the parity latch is unique in that it provides an active parity bit when turned off (not latched). All other command register latches provide active outputs when turned on.

During a logout sequence, positions 0 through 7 of the command register are gated to SBI positions 8 through 15 (respectively) during storage of log word 2; the parity latch output is gated to SBI position 62 during storage of log word 3.

DATA ADDRESS REGISTER

- The data address register provides the storage address of the first CCW.
- Subsequently, the data address register provides the storage address for data transfers.
- The DAB portion of the data address register identifies the byte location in the B-register into which the first data byte is loaded (read-type operations) or from which the first data byte is gated (write-type operations).

The data address register provides a means of addressing main storage to obtain the first CCW and, thereafter, a means of addressing main storage on read-type or write-type data doubleword transfers. In addition, a portion of the data address register, designated the data address byte portion, identifies a byte location in the doubleword capacity of the B-register. The DAB is effective only when the data address bits from a CCW are gated into the register. The DAB determines the B-register byte location to or from which the first data byte will be gated (depending upon whether a read-type or write-type operation is specified). The data address register consists of 27 latches (24 data and 3 parity); all latches have indicators on the CE panel.

Block Diagram Description

The data address register latches 0 through 20 (Figure 2-12) identify the first CCW storage address and subsequent data storage addresses. Three other latches (21 through 23) contain the DAB bits, and three latches contain the parity bits for the three bytes of the register.

Gating logic associated with inputs to the data address register provides for gating bits into the register from the SBO gating logic and from the adder.

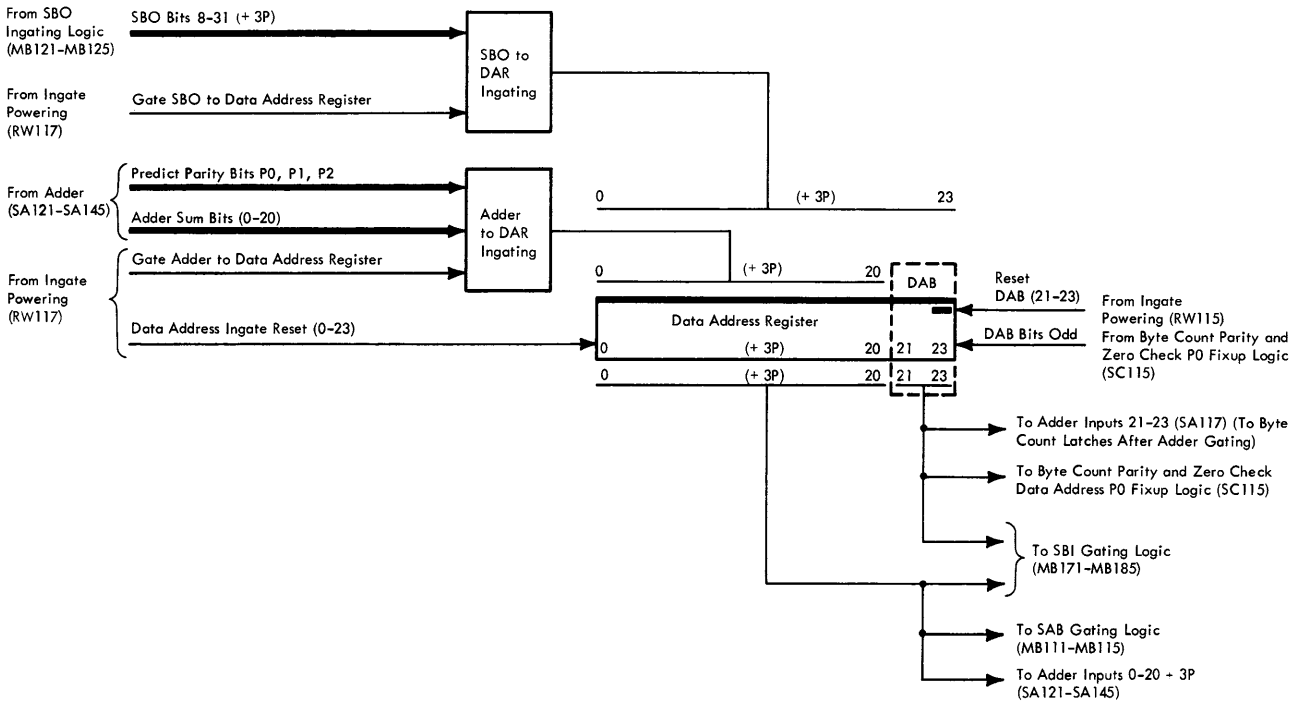


Figure 2-12. Data Address Register, Block Diagram

CAW Fetch Cycle Operations

- On a CAW fetch cycle, the address of the first CCW is gated into the data address register from the SBO.
- Data address register bits 21 through 23 must be all zeros to specify a "doubleword boundary" CCW address; otherwise, the 'program check' latch is turned on.

For a CAW fetch cycle, the address of the first CCW is gated into the register from SBO bit positions 8 through 31 (and P1, P2, and P3). When CAW bits are present on the SBO lines, the 'data address ingate reset' signal is first activated to reset the data address register to all zeros with good parity. Subsequently, the 'gate SBO to data address register' signal is raised, gating SBO bits 8 through 31 plus the 3 parity bits into the data address register. These bits specify the storage address location of the first CCW. Since each CCW must be addressed on a doubleword boundary (as an architectural requirement) data address register bits 21 through 23 must be all zeros. These bits are checked for all zeros at the SBO gating logic and, if not all zeros, cause turn-on of the 'program check' latch which terminates the start I/O operation. After the CAW is gated to the data address register, bits 0 through 20 are available for gating to the SAB lines during the succeeding CCW fetch cycle. The DAB bits (all zeros) have no significance after the CAW bits are gated into the data address register.

First CCW Fetch

- Data address register bits 0 through 20 and the three parity bits are gated to the SAB lines on the first CCW fetch cycle.
- Data address register bits are gated to the adder, incremented by one (equivalent to eight bytes), then gated to the command address register.
- When the CCW is available on the SBO lines, the data address register is reset, and the address of the first data word (assuming data transfer is specified by the CCW) is gated into the data address register from the SBO lines.
- Bits 0 through 20 specify the storage location to or from which data is transferred on a read-type or write-type operation.
- DAB bits 21 through 23 define the first data byte within the first doubleword to be transferred.
- The DAB is gated to the adder and the byte count register.
- The parity bit for data address register bits 16 through 23 is changed if the DAB contains an odd number of ones or remains unchanged if an even number of ones.
- The DAB is reset to all zeros after gating to adder and byte count register is accomplished.

When the CAW fetch cycle is complete, the first CCW fetch cycle is initiated. At the beginning of the cycle, (signified by a 'BCU response' signal to the channel), data address register bits 0 through 20 plus the three parity bits are gated to the SAB lines (via the 'SAB gating' logic) to address main storage for the CCW. The address in the data address register is also gated to the adder where it is subsequently incremented by one (corresponding to 8 bytes) and transferred to the command address register. (If data or command chaining is involved in the current operation, the addresses of subsequent CCW's are obtained from the command address register.)

When the first CCW is available on the SBO lines, the 'data address ingate reset' signal resets the data address register to all zeros with good parity and the subsequent 'gate SBO to data address register' signal gates SBO bits 8 through 32 (plus three parity bits) into the data address register. Assuming that the command in the CCW specifies a read-type or write-type operation, bits 0 through 20 of the data address register now specify the main storage address for the first data transfer; the data transfer may be either a store or fetch operation. Bits 21 through 23 (the DAB) specify the byte within the doubleword at which data transfers begin.

Following gating of the CCW bits to the data address register, the DAB is gated to the adder and to the byte count register. After this gating occurs, the 'reset DAB' signal resets the DAB bits to zero so that storage will be addressed only on doubleword boundaries on future data transfers to or from storage. However, before the DAB bits are reset, they are examined at the 'byte count parity and zero check DA P0 fixup' logic to determine if the DAB is composed of an odd or even number of ones. If the DAB bits contain an odd number of ones, the 'DAB bits odd' signal is raised and the parity bit associated with data address register bits 16 through 23 is changed. This parity bit is changed so that byte 3 in the data address register will maintain correct parity when the DAB is reset. If the DAB bits contain an even number of ones, the 'DAB bits odd' signal is not raised and the parity bit remains unchanged.

Read-Type or Write-Type Data Transfers

- Data address register bits 0 through 20 plus three parity bits are gated to the SAB lines on data doubleword transfer to or from main storage.

- These bits are also gated to the adder, incremented by 1 (eight bytes) and gated back into the data address register to specify the address of the next data doubleword transfer.

- On 'read backward' operations the data address is decremented by one, rather than incremented.

When the current I/O operation involves a data transfer (read-type or write-type operation), data address register bits 0 through 20 are gated through the 'SAB gating' logic to address the storage location to or from which data is to be transferred. In addition, bits 0 through 20 are gated to the adder where they are incremented by one (assuming that a 'read backward' operation is not in progress, in which case the bits are decremented by one). The incremented (or decremented) output of the adder (adder sum bits 0 through 20) and three predicted parity bits (P0, P1, and P2, one bit for each byte of the data address register) are applied to the 'adder to DAR ingating' logic. After the data address has been on the SAB lines a sufficient time to access storage, the 'data address ingate reset' signal is raised to reset the data address register. Following the reset, the 'gate adder to data address register' signal is raised to gate 'adder sum' bits 0 through 20 into the data address register. The data address register now contains the address of the storage location to or from which data will be transferred on the next storage cycle. The above described action (gating of data address to the SAB and the adder, and subsequent gating of the 'adder sum' bits into the data address register) continues until all data bytes specified by the current CCW are transferred.

SBI Gating

On logout operations, the contents of the data register are gated through the SBI logic for storage in main storage. Bits 0 through 23 are gated to SBI lines 32 through 55 for storage in logword 2 of the logout sequence. The three parity bits (P0, P1, and P2) are gated to SBI lines 57 through 59 for storage in log word 3 of the logout sequence.

Logic Diagram Description

Positive logic for the data address register is shown in Diagram 4-8. Separate descriptions (based on Diagram 4-8) of the 'bit 0' through 'bit 20' latches, the DAB latches, and the parity (bit P0, P1, and P2) latches are provided in the following text.

Bit 0 through Bit 20 Latches

The bit 0 through bit 20 latches are identical and may be turned on by bits gated from an SBO line or from the adder. Since these latches are identical, only the 'bit 0' latch is described in this text. The 'bit 0' latch is reset before each ingating operation by the 'data address ingate reset' signal. When CAW or CCW bits are on the SBO lines, the AND receiving the 'SBO bit 8' signal is gated by the 'gate MDBO to data adddress register' signal. If a logic 1 level is on the 'SBO bit 8' line at the gating time, the 'bit 0' latch is turned on, placing a logic 1 on the 'data address bit 0' output of the latch. A logic 0 on the 'SBO bit 8' line at gating time inhibits the AND, and the 'bit 0' latch remains off.

For data transfer operations between main storage and the selector channel, the 'bit 0' through 'bit 20' latches are updated (incremented or decremented) after each data transfer in preparation for the next data transfer. When updating has been completed by the adder and the 'data address' bits are no longer required on the SAB lines, the 'data address ingate reset' signal again resets the 'bit 0' latch. The 'bit 0' latch AND receiving the 'adder sum 0' bit is then gated by the 'gate adder to data address register' signal. A logic 1 on the 'adder sum 0' line turns on the 'bit 0' latch while a logic 0 on the line leaves the latch off.

DAB Latches

The 'bit 21' through 'bit 23' (DAB) latches receive bits only from the 'SBO bit 29' through 'SBO bit 31' lines, respectively (Diagram 4-8). Although bits are gated to the DAB latches on the CAW and CCW fetch cycles as previously described for the 'bit 0' latch (in the preceding paragraph), gating to the DAB latches is significant only on the CCW fetch cycle. (On the CAW cycle, the DAB must be all zeros, but the zero test is made before gating to the latches occurs. Once the CAW bits are gated into the DAB latches, they do not affect channel operations.) On the CCW cycle, the DAB latches are reset by the 'data address ingate' signal, SBO bits 29 through 31 are applied to the ANDs for the 'bit 21' through 'bit 23' latches, respectively. After the DAB latches are reset, the 'gate MDBO to data address register' signal gates the SBO bits into their respective latch. A logic 1 on an SBO line turns on the respective DAB latch and a logic 0 leaves the latch off.

After the DAB bits have been gated to the adder and byte count register, the DAB latches receive a 'reset DAB' signal which turns off the latches. For subsequent channel operations for the current CCW, the DAB latches remain off and the logic 0 outputs of these latches specify data doubleword

transfers to or from main storage on doubleword boundaries.

Before the DAB latches are reset after a CCW fetch cycle, the outputs of the DAB latches are decoded by two AND/AND/OR combinations.

The two outputs developed by this decoding ('read DAB 4 or read backward DAB 3' and 'read DAB 0 or read backward DAB 7') are used by the 'read CDA controls' logic to determine if overrun occurred during a 'read CDA' CCW fetch cycle or to determine which gating and reset signals must be generated in the event overrun did not occur. In addition, the 'read DAB 0 or read backward DAB 7' signal is sent to the 'service out-service in' logic where, if active, it provides one of the conditions necessary for the generation of the 'service out' signal to the I/O device on a read operation.

Parity Latches

The data address register (Diagram 4-8) has three parity latches (P0, P1, and P2), one for each eight-bit data byte of the data address register. The 'bit P2' latch is unique and is described in this text following the description of the 'bit P0' and 'bit P1' latches.

The parity 'bit P0' and 'bit P1' latches are turned on to provide a logical 1 output by the 'data address ingate reset' signal. Since this signal also turns off the data address register 'bit 0' through 'bit 15' latches, odd parity is maintained for the first two data byte positions when the reset occurs. When data is gated from the SBO lines or the adder, the 'gate MDBO to data address register' signal or 'gate adder to data address register' signal (respectively) strobes an AND at the reset input to the 'bit P0' and 'bit P1' latches. If the gated bit to the 'bit P' or 'bit P1' latch is a logic 0 level, the latch is turned off to present a logic 0 at the latch output. If the gated bit is a logical 1 level, the latch remains on.

The data address register 'bit P2' latch is unique in that it contains provisions for sensing whether the state of the latch is to change before the DAB is reset following a CCW fetch operation. In addition, the 'bit P2' latch (like the 'bit P0' and 'bit P1' latches) is turned on to present a logic 1 at its output when the 'data address ingate reset' signal occurs. The 'bit P2' latch also has provisions for gating from the SBO P3 line and from the adder ('predetermined parity P2' bit). Ingating from these lines is as described for the 'bit P0' and 'bit P1' latches.

The sensing portion of the 'bit P2' latch consists of three ANDs and an OR (Diagram 4-8). This logic is controlled by: (1) the sensed output of the 'bit P2' latch; and (2) the 'DAB bits odd' signal, which is active only when the DAB latches contain an odd number of ones and while the DAB bits are being gated to the adder. As previously stated, if the 'DAB bits odd' signal is active, the state of the 'bit P2' latch will be changed, regardless of whether it is in the logic 1 or logic 0 state. While the 'DAB bits odd' signal is not active, the latch contents remain unchanged (and thus, may be altered by the 'data address ingate reset' signal or by gating from the SBO line or adder under the circumstances previously described).

Assume that the DAB contains an odd number of 1's and that the 'bit P2' latch is in the logic 1 state; this means that when the DAB bits are gated to the adder, the 'DAB bits odd' signal is raised and the 'bit P2' latch is reset to the logic 0 state. This is accomplished by applying the 'DAB bits odd' ('not DAB bits even') signal to OR gate 4, removing one of the inputs which enables the OR. The input to OR gate 4 from AND gate 2 is also inactive (at the disabling level) since AND gate 1 is enabled by the sense output of the 'bit P2' latch and the disabling level from AND gate 2. With AND gate 1 enabled, AND gate 2 is disabled and the AND gate 2 output does not enable OR gate 4. Also, AND gate 3 is disabled and the AND gate 2 output does not enable OR gate 4. Also, AND gate 3 is disabled by the output of enabled AND gate 1, and, thus, the third input to OR gate 4 is at the disabling level. With the above conditions present, the output of OR gate 4 is at the disabling ('not OR gate 4') level, which causes a reset signal to be applied to the 'bit P2' latch. Thus, the state of the latch is changed from a logic 1 level to a logic 0 level because the DAB latches contained an odd number of bits. For a second example, assume that the DAB contains an odd number of 1's and that the 'bit P2' latch contains a logic 0 level. In this case, AND gate 1 is disabled by the sensing output of the 'bit P2' latch; this provides enabling inputs to both AND gates 2 and 3. OR gate 2 is enabled by the 'DAB bits even' signal (before the 'DAB bits odd' signal is raised), enabling the second input to AND gate 3. In turn, the output of OR gate 4 is held at the enabling level. With the 'OR gate 2' output enabled, a second input to AND gate 2 is enabled. When the 'DAB bits odd' signal is activated, the output of AND gate 2 is raised to turn on the 'bit P2' latch.

If the 'DAB bits odd' signal is not raised when the DAB bits are gated to the adder, AND gate 2 remains disabled and OR gate 2 remains enabled. In this case, the content of the 'bit P2' latch is not affected.

FLAG REGISTER

- Consists of a nine-bit flag register:
  1. Bits 0-4 are CDA, CC, SILI, Skip and PCI flag positions, respectively.
  2. Bits 5-7 (all zeros) and P used for parity checking.

- Indicators provided for all but 5-7 bit positions.

The flag register and associated logic (Figure 2-13) consists of a 9-bit register (eight data bits and one parity bit), 'flag register gating' logic, 'flag register gating control' logic, and 'parity checking' logic. Bits 0 through 4 of the flag register

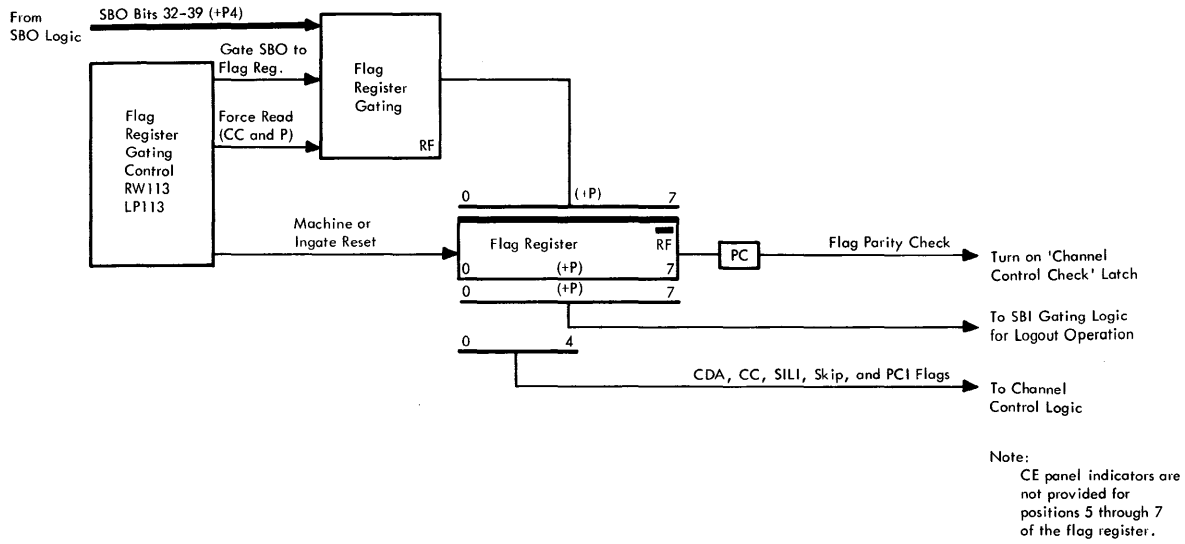


Figure 2-13. Flag Register, Block Diagram

are reserved for flag bits which control portions of channel operations initiated by a Start I/O, IPL, or FLT instruction. Bits 5 through 7 and the parity bit are used for flag register parity checking. Indicators are provided on the CE panel only for the parity bit and bits 0 through 4 (locations C19 through C23). Indicators are not provided for bits 5 through 7, since these bits must always be zero to constitute a properly formatted flag field. Bits 0 through 4 are designated as follows:

Bit 0 – CD or CDA flag
Bit 1 – CC flag
Bit 2 – SILI flag
Bit 3 – Skip flag
Bit 4 – PCI flag

Flag Register Block Description

- Data from CCW gated to flag register:
  1. 'Machine or ingate reset' signal resets flag register.
  2. 'Gate SBO to flag register' signal activated to gate SBO bits 32 through 39 plus P4 into flag register.
  3. 'Parity checking' logic checks flag register parity; parity error turns on 'channel control check' latch.

- For IPL or FLT operations, 'force read operation' signal (activated when channel forces first CCW) activates flag register CC and parity bits.

- CDA, CC, SILI, Skip, and PCI flag bits monitored by channel control logic; active flag bits cause channel operations to be altered or suppressed (depending on channel operation in progress and active flag bit).

- All flag register bits supplied to SBI logic for storage during log word 2 (bit 0-7) and log word 3 (bit P) of logout operation.

Data can be gated into the flag register either from the SBO lines (SBO bits 32 through 39) or as the result of an IPL or FLT operation. Data is gated into the flag register from the SBO lines when the channel fetches a CCW from main storage (Figure 2-13). When the CCW is on the SBO lines, the 'flag register gating control' logic activates the 'machine or ingate reset' signal to reset the flag register. Subsequently, the 'gate SBO to flag register' signal is activated to gate SBO bits 32 through 39 and P4 into the flag register via the 'flag register gating' logic.

With the flag field bits in the register, the 'parity checking' logic checks the bits for correct (odd) parity. If even parity is detected, the 'flag parity check' signal is activated and the 'channel control check' latch is turned on to end the operation.

For the IPL or FLT operations, the 'force read operation' signal (activated when the channel forces the first IPL or FLT CCW) activates the CC and parity bits in the flag register.

Flag bits 0 through 4 (CDA, CC, SILI, Skip, and PCI) are supplied to channel control logic; the control logic monitors the flag bits, and depending upon the channel operation in progress, uses the flag bit to suppress, perform, or alter operations as specified by active flag bits. All flag register bits (0 through 7 plus P) are supplied to SBI gating logic where they are available for storage during a logout operation. For logout operations, bits 0 through 7 are gated to SBI lines 16 through 20, respectively, for storage as part of log word 2. The parity bit is gated to the SBI line 62 for storage as part of log word 3.

Flag Register Logic Description

The flag register and associated logic are shown in Diagram 4-9. Flag register operations are described in detail in the following paragraphs.

Flag Register Gating Control

- 'Machine or ingate reset' signal resets flag register data latches for logic 0 outputs and 'parity' latch for logic 1 output.

- 'Machine or ingate reset' signal activated by:
  1. 'TIC' signal to reset latches and prevent TIC CCW bits from being gated to latches.
  2. 'Machine or setup reset' signal.
  3. 'Gate SBO to count, data address registers' signal when non-TIC CCW is on SBO lines.

- 'Gate SBO to count, data address register' signal also turns on 'SBO flag register' latch.

- 'Gate SBO to flag register' signal gates 'SBO bit P4', and 'SBO bit 32' through 'SBO bit 39' values to 'parity', 'CDA', 'CC', 'SILI', 'skip', 'PCI' and 'FLR 5' through 'FLR 7' latches, respectively.

- For IPL or FLT operations, 'force read operation' signal activates parity and CC flag bits.

The 'flag register gating control' logic (Diagram 4-9) provides a 'machine or ingate reset' signal to reset the flag register latches, and provides a 'gate SBO to flag register' signal to gate the flag bits (contained in the CCW) into the flag register latches.

An active 'machine or ingate reset' signal turns off all flag register data latches and turns on the parity latch. This provides a flag register output of all zeros with correct parity.

Note: The parity latch is actually off (ALD RF111) when reset; however, the off condition reflects a logic 1 at the latch output. For simplification, Diagram 4-9 shows the latch turned on to reflect a logic 1.

The 'machine or ingate reset' signal is activated by one of three signals: 'TIC', 'machine or setup reset' or 'gate SBO to count, data address register'.

The 'TIC' signal is activated during a TIC operation. At the 'flag register gating control' logic, the 'TIC' signal activates the 'machine or ingate reset' signal and prevents latching of the 'SBO flag register' latch. Thus, the 'TIC' signal prevents data bits in the TIC CCW from being gated into the flag register. For a TIC CCW, the flag field bits (on the SBO lines) are of no significance; thus, the flag register latches are reset and held to all zeros with good parity to prevent possible erroneous detection of a 'flag parity check' condition.

The 'machine or setup reset' signal also activates the 'machine or ingate reset' signal to reset the flag register data latches. The 'machine or setup reset' signal is activated by the channels 'machine reset' or 'setup' signals and resets the flag register latches (as well as other channel registers and latches) prior to the channel starting or continuing a channel operation.

The 'gate SBO to count, data address registers' signal (generated when a non-TIC CCW is on the SBO lines) activates the 'machine or ingate reset' signal and also turns on the 'SBO flag register' latch. In this case, the 'machine or ingate reset' signal is activated to reset the flag register latches prior to gating flag-field SBO bits into the flag register latches. Following turn on of the 'SBO flag register' latch, the 'gate SBO to flag register' signal is activated. This signal gates the 'SBO bit 32' through 'SBO bit 39' and 'SBO bit P4' values into the 'CDA', 'CC', 'SILI', 'skip', 'PCI', 'FLR 5' through 'FLR 7', and 'parity' latches, respectively. When the 'machine or ingate reset' signal falls, turn off of the 'SBO flag register' latch is delayed to permit the 'gate SBO to flag register' signal to latch the SBO bit values into the flag register latches.

For IPL or FLT operations, a 'force read operations' signal is activated for the first portion of the operation. This signal is ORed at the output of the flag register 'parity' and 'CC' latches to activate the 'flag parity' and 'CC flag' bits. These bits are sustained by the 'force read operation' signal until the first IPL or FLT 'chain command' operation is initiated.

Flag Register Latches

- 'CDA flag' signal enables channel to perform read CDA or write CDA operations.

- 'CC flag' and 'not CDA flag' signals enable channel to perform chain command operations.

- 'CC flag' and 'CDA flag' signals enable channel to perform 'jump command address on CDA' operation.

- 'SILI flag' signal prevents channel from detecting incorrect length condition.

- 'Skip flag' signal prevents channel from storing data during read-type operations.

- 'PCI flag' signal causes channel to store CSW as soon as possible during current operation.

- Parity error activates 'flag parity check' signal; 'channel control check' latch turns on to end operation.

- Flag and parity bits available at SBI gating logic for storage on logout operation.

The nine flag register latches (Diagram 4-9) are designated as follows: 'parity', 'CDA', 'CC', SILI', 'skip', 'PCI', and 'FLR 5' through 'FLR 7'. All latches are off to reflect an inactive bit at the output and on to reflect an active bit at the output. (The 'parity' latch is actually off to reflect an active flag parity bit, but is shown "on" for convenience in Diagram 4-9.)

The 'CDA' latch, when on, provides a CDA flag output which enables the channel to perform read CDA or write CDA operations. The 'CC' latch provides a 'CC flag' signal when on. Providing the 'CDA' latch is off, the active 'CC flag' signal enables the channel to perform chain command operations. If the 'CDA' latch is on, an active 'CC flag' signal enables the channel to perform 'jump command address for CDA' operations, if so designated by a status byte from the I/O device.

If the 'SILI' latch is on, the 'SILI flag' signal prevents the channel from detecting an 'incorrect length' condition. With the 'skip' latch on during a read, read backward or sense operation, the 'skip flag' signal prevents the channel from storing data received from the I/O device during the read-type operation.

The on condition of the 'PCI' latch provides a 'PCI flag' signal. The 'PCI flag' signal causes the channel to request an interrupt at the earliest possible time to store the CSW; the PCI operation does not stop channel operations specified by the command in the command register.

The outputs of all flag register latches are applied to exclusive-OR logic ("odd" logic blocks on Diagram 4-9) in the flag register 'parity checking' logic. If a parity error (even parity) is present in the flag register latches during read-type or write-type operations, the 'flag parity check' signal is activated when the 'sample flag parity' or 'sample flag parity write CDA' signal strobes the 'parity check' results. The 'flag parity check' signal turns on the 'channel control check' latch to terminate the operation in progress.

The 'PCI flag', 'skip flag', 'SILI flag', 'CC flag', 'CDA flag', and 'flag parity' outputs of the flag register latches are also supplied to the SBI gating logic. During a logout operation, the active or inactive state of the flag bits are stored in positions 16 through 20 of log word 2 and flag parity bit in position 62 of log word 3.

COUNT REGISTER CIRCUITS

The count register circuits (Figure 2-14) maintain an updated count value representing the number of bytes to be transferred for a current CCW when the channel is engaged in data transfer operations. The count register circuits also monitor the count value for count characteristics used by the channel control logic to determine the progress and status of data transfer operations at specified times. Channel control logic usage of the count characteristic signals is dependent upon the channel operation in progress.

Block Diagram Description

- Count register circuits consist of:
  1. 'Count ingating' logic.
  2. Count register (20 bits, designated 7 through 23, P0, P1 and P2)
  3. 'Count register channel decoding' logic

- Initially, count register receives count from CCW fetched from main storage or COUNT FIELD switches (SBO bits 48-63 gated to count register positions 8-23).

- Count register bits 7-23 are gated to adder, added with DAB bits, and the sum is gated back into count register.

- During data transfer operations, count register bits 7-23 are gated to adder, decremented by eight bytes, and the results are gated back to count register.

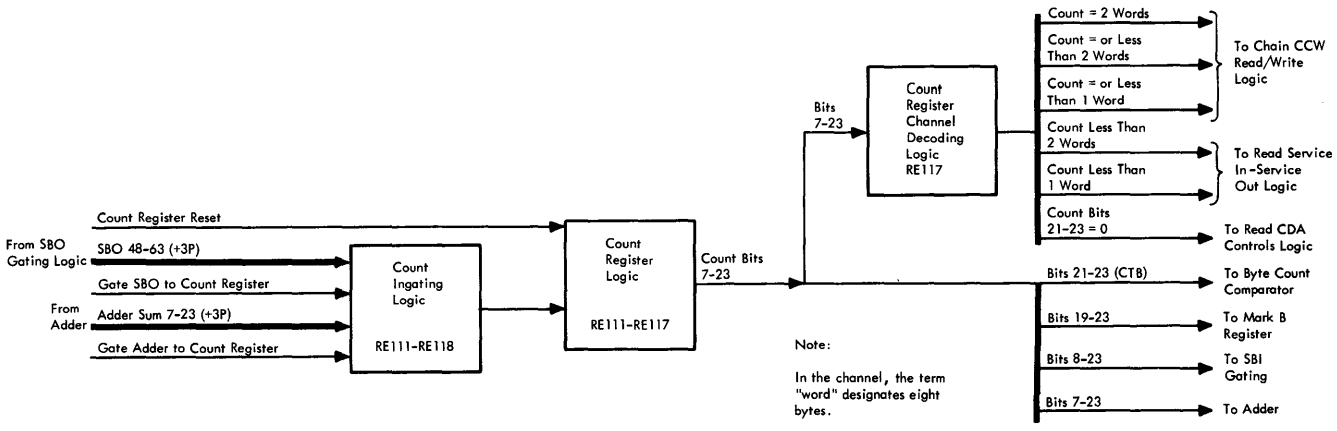


Figure 2-14. Count Register Circuits, Block Diagram

- For write CDA operation, count register bits 19 through 23 are gated to mark-B register to maintain count; detection of incorrect length condition causes count to be gated from mark-B register to count register for residual count operations.

- CTB (count register bits 21-23) compared with BC register bits to detect last byte of data transfer operation.

- Count register bits 7-23 available at SBI gates 48-63 for 'CSW store' or logout operations.

- Count characteristic signals decoded by 'count register channel decoding' logic; signals identify last word, error, and retain storage conditions for specific operations.

The count register circuits (Figure 2-14) consist of the 'count ingating' logic, the count register (bits 7 through 23 plus 3P) and the 'count register channel decoding' logic.

The 'count ingating' logic receives SBO bits 48 through 63 from the SBO gating logic and adder sum bits 7 through 23 from the adder. During data handling operations, SBO bits 48 through 63 are initially gated into count register positions 8 through 23 via the 'count ingating' logic from either a CCW fetched from main storage or from the CE panel COUNT FIELD switches; the values in the COUNT FIELD switches are gated to the count register only when the channel is in the test mode and simulating storage operations. (Position 7 of the count register does not initially receive a count bit, but is available to receive a carry bit from the adder if subsequent DAB plus count operations generate a carry bit.) Before data transfer operations begin, bits 8 through 23 in the count register are gated to the adder where they are added with bits 21 through 23 (DAB) of the data address register. The results (used by the 'count register channel decoding' logic and channel control logic to detect specified control conditions and the last byte of the transfer operations) are then gated into the count register via the 'adder sum 7 through 23' lines. During data transfer operations, bits 7 through 23 of the count register are gated to the adder and decremented by the equivalent of eight bytes each time the channel accesses main storage (or simulates a storage access in the test mode). After decrementing, the results are gated back into the count register via the 'adder sum 7 through 23' lines. If a write CDA operation is in progress, bits 19 through 23 of the count register are gated to the mark register when the channel fetches the next CCW. (With the significant count bits in the mark-B register, the channel continues transferring data specified by the current CCW, and the count register is free to accept the count bits from the new CCW.) If an incorrect length condition is detected during transfer operations for the current write CDA CCW, the count bits in the mark-B register are gated through the 'count ingating' logic to the count register via the SBO 59 through 63 lines; these count bits may then be used to calculate a residual count which defines the difference between the bytes specified for transfer by the CCW and the bytes actually transferred.

Count register bits are also sent to the 'byte count comparator' logic, the 'SBI gating' logic and the 'count register channel decoding' logic (Figure 2-14). The 'byte count comparator' logic receives count register bits 21 through 23 (referred to as 'count B' or CTB). These bits are compared with the byte count register bits throughout channel controlled data transfer operations; when the CTB and byte count bits compare, all data bytes specified by the current CCW have been gated into or out of the B-register. Count register bits 8 through 23 are supplied to the SBI gating logic (SBI gates 48 through 63) for gating to storage on a 'CSW store' or logout operation.

The 'byte count comparator' logic (Figure 2-14) receives count register bits 7 through 23. These bits are monitored by the 'count register channel decoding' logic throughout data transfer operations to provide count characteristic signals. In general, the count characteristic signals enable the channel to detect a last word condition, error condition, or retain storage condition. In many cases, the count characteristic signals are operation oriented; e.g., read, write, read CDA, or write CDA. The count characteristic signals are 'count equals 2 words', 'count equal or less than 2 words', 'count equal or less than one word', 'count less than 2 words', 'count less than 1 word' and 'count bits 21 through 23 equal zero'.

Note: The term word as shown in channel logic designates 8 bytes; except when signal line names are designated in text throughout this manual, the term "doubleword" is substituted for the term "word". A detailed description of the function of each count characteristic signal is contained in "Count Characteristic Signals".

Detailed Count Register Logic Description

Logic details of the count register circuit are shown in Diagram 4-10. The count register consists of 20 latches (17 data and 3 parity) designated 'bit 7' through 'bit 23', 'bit P0', 'bit P1' and 'bit P2'. COUNT indicators for all but the high-order bit 7 and the 'bit P0' latches are located at positions B19 through B26 on the CE panel; the indicator for the 'bit 7' latch (CT7) is located at position F27, and no indicator is provided for the 'bit P0' latch. Count register operations for read, read CDA, write, write CDA commands, the IPL and FLT operations, and the count characteristic signals provided by the 'count register channel decoding' logic are described separately in the following paragraphs.

Count Register Read/Write Operation

- For start I/O operation, 'SBO to count register' latch is turned on when CCW is on SBO lines; count field gated from SBO lines 48-63, P6 and P7 into count register positions 8-21, P1 and P2.

- For IPL or FLT operation, initial count of 24 decimal forced into count register (bits 19 and 20 latches turned on) by 'turn on CCW valid IPL' signal.

- After start I/O CCW count is in count register, count bits 7-23, P0, P1 and P2 are gated to adder, added to DAB, and gated back to count register.

- Bits from 'adder sum 7-23, P0, P2 and P2 lines gated to count register when 'adder to count register' latch turned on.

- Count register reset conditions:
  1. Initial gating from SBO lines activates 'count ingate reset' and 'reset count 21-23' signals. (Reset to all zeros with active parity bits).
  2. Count update activates 'count ingate reset' signal; all bits reset except 21-23.
  3. When BC = CTB condition detected, count register is reset in same manner as initial gating reset.

- Incorrect length detection causes counter contents to be gated to adder for residual count computation; results are gated back to adder.

For a read or write operation, the count register (Diagram 4-10) receives a significant count when a CCW for a start I/O operation is presented to the channel on the SBO lines. (For the beginning of the IPL and FLT operations, the channel forces a significant count of 24 decimal, into the count register instead of receiving the count from a CCW; i.e., the count register 'bit 19' and 'bit 20' latches are turned on by the 'turn on CCW valid IPL' signal.) When the CCW is present on the SBO lines, the 'gate SBO to count register' signal (derived from the 'raw advance SS' signal) is activated to turn on the 'SBO to count register' latch (Diagram 4-10).

The latch output gates SBO bits 48 through 63, P6 and P7 into count register positions 8 through 23 and P1 and P2 (respectively) via the count ingate ANDs. After the I/O device is selected and the channel has determined that the CCW is valid, count register bits 7 through 23 plus bits P0, P1 and P2, are gated to the adder where they are added to bits 21 through 23 (the DAB) from the data address register. The resulting sum is gated gack to the count register via the 'adder sum 7 through 23' lines by turning on the 'adder to count register' latch (Diagram 4-10). With the 'count plus DAB' sum in the count register, the three low-order count register positions (bits 21 through 23) contain a value designated count B (CTB). The CTB, when equal to the value in the byte count register, identifies the transfer of the last data byte for the read or write operation.

When, during I/O byte transfers, a B-register boundary is reached ('byte count latches equal zero' condition), the channel initiates a storage request to store (read operation) or fetch (write operation) a data doubleword. During the subsequent sequence 3 routine, the count register contents are gated to the adder and decremented by 1 in position 20 (count value is decremented by the equivalent of eight bytes). The decremented count is then gated back to the count register via the 'adder sum 7 through 23' (and P0, P1, and P2) lines. Each time the channel enters a storage routine to store or fetch a data doubleword, the count register contents are decremented in the manner described.

When count register positions 7 through 20 are equal to zero ('count equal or less than 2 words' signal active), the channel's 'last word' trigger is turned on and the channel begins sensing for a byte count equal CTB (BC = CTB) condition. When the BC = CTB condition is sensed, the count register is reset to all zeros, indicating that all required bytes for the read or write operations have been transferred.

Count register reset conditions are dependent upon the operation performed. Upon initial ingating from the SBO lines, both the 'count ingate reset' and 'reset count 21, 22, 23' signals (Diagram 4-10) are activated to reset count register positions 7 through 23 to 0's and the three parity latches to 1's. For an 'update count' (decrement) operation, only the 'count ingate reset' signal is activated, resetting positions 7 through 20 of the count register before the decremented count is gated back to the count register. The CTB positions (bits 21, 22, and 23 of the count register) remain unchanged. When the BC = CTB condition is sensed on the last doubleword transfer, the count register is reset in the same manner as described for initial gating from the SBO lines.

If data transfer operations are ended before the specified number of bytes has been transferred and an incorrect length (IL) condition is detected, a 'BC and CT-1, IL' signal is generated (ALD EN111). This signal causes the current byte count in the byte counter to be subtracted (at the adder) from the CTB in the count register to obtain a 'residual count' value. The results of the subtraction are gated into the count register, thereby restoring the count register count to a value representing the true number of bytes not transferred for the current CCW. This operation is effectively the reverse of the DAB plus count operation; the residual count value is available at the SBI gating logic for storage as part of the CSW.

Count Register Write CDA Operation

- Count register bits 19-23 and P2 gated to mark-B register positions 4-0 and P when channel fetches new CCW.
- Mark-B register contains count bits to control current CCW data transfer ending.
- Count bits from new CCW gated into count register while current CCW write operations performed.
- Detailed incorrect length condition turns on 'SBO to count register' latch; count from mark-B register gated back into count register via SBO lines 59-63 and P7.
- Residual count operation performed as follows:
  1. Gate count register value minus 8 bytes to adder.
  2. Adder performs residual count operation.
  3. Turn on 'adder to count register' latch; gate residual count value into count register.

Count register operations during a write CDA operation are as described for the write operation, except for one portion of the write CDA operation. On write CDA operations, the channel must fetch a new CCW while data from the old CCW is still being transferred under control of the old count. So that the current count may be maintained when the count from the new CCW arrives, the channel transfers count register bits 19 through 23 and P2 to mark-B register positions 4 through 0, and P, respectively (Diagram 4-10). The transfer is effected when either the count characteristic 'count equals 2 words' signal or 'count less than 2 words' signal is active along with the 'write', 'CDA latch', 'not sequence 3', 'sequence 4' and 'T2 clock' signals (ALD CW113). With these signals active, the 'gate CTB to mark B' signal is activated, and count register bits 19 through 23 (Diagram 4-10) are gated into the mark-B register. Channel sensing for the CTB = BC condition is then performed using the count in the count-B register. When the new CCW arrives, the 'count field' bits are gated from the SBO lines into the count register as described in "Count Register Read/Write Operation".

If an 'incorrect length' condition is detected while the count bits in the mark-B register are controlling current CCW write transfers, a 'write CDA incorrect length, sequence 5' signal is generated to turn on the 'SBO to count register' latch (Diagram 4-10). In this case, the resulting 'SBO to count register' signal gates SBO bits 59 through 63 and P7 into count register positions 19 through 23 and P2. SBO bits 59 through 63 and P7 represent the count bits obtained from the mark-B register. With the original count reentered into the count register, the channel computes the residual count value by gating the contents of the count register (minus 8 bytes) and the two's complement of the byte count register to the adder. The results of the computation are gated back into the count register via the adder sum 7 through 23 lines when the 'gate adder to count WLR' signal is activated to turn on the 'adder to register' latch (Diagram 4-10). The residual count in the count register is then available at SBI gating positions 48 through 63 for the CSW store operation that follows.

Count Register SBI Gating

Count register positions 8 through 23 are available at SBI gating positions 48 through 63 (Diagram 4-10) for gating to main storage during a CSW store operation or during storage of log word 1 during a logout operation. Count register parity bits are available at SBI gating positions 53 through 55 for gating to main storage when log word 3 is stored during a logout operation.

Count Characteristic Signals

- 'Count equals 2 words' signal effective only during write CDA operation; gates CTB bits from count register to mark-B register and turns on 'retain storage' latch.
- 'Count bits 21 through 23 equal zero' signal inhibits detection of overrun condition during read operations when last word trigger is on and 'BCL = 0' signal is active.
- 'Count less than 2 words' signal:
  1. For write CDA operation, gates CTB bits from count register to mark-B register and turns on 'retain storage' latch.
  2. For read CDA operation, turns on 'chain check' latch when new CCW is received and channel had already received more data bytes than count specifies (A-register is full and 'BC greater than CTB' signal active).
- 'Count equal or less than 2 words' sampled during read-type and write-type operations for last word condition after data transfer operations has begun.
- 'Count equal or less than 1 word' signal sampled during sequence 1 (initial selection routine), read CDA and write CDA for last word condition prior to CCW assuming control of data transfers.
- 'Count less than 1 word' signal sampled during read CDA operation for chain check condition. When new CCW is available, signal is sampled prior to and after count plus DAB operations to determine if chain check condition exists.

Note: The term "word" in the count characteristic signal names refer to eight bytes. The count characteristic signals are activated by the 'count register channel decoding' logic

which monitors the outputs of the count register. These signals are used by channel logic to detect the last word condition for all read-type and write-type operations and to detect errors or events for specific channel operations. Each of the count characteristic signals is described separately in the following text.

The 'count equals 2 words' signal (Diagram 4-10) is effective only during write CDA operations. The signal is activated when all count register bits except bit 19 are zeros. The active 'count equals 2 words' signal is OR'ed (ALD LT113) to activate a 'count equal or less than 2 words' signal. With the 'write' signal and CDA flag active, the channel generates signals during the sequence 4 routine which: (1) gate the CTB bits from the count register to the mark-B register; and (2) turn on the channel's 'retain storage' latch.

The 'count bits 21 through 23 equal zero' signal (Diagram 4-10) inhibits detection of an overrun condition during read operations when the 'last word' trigger is on. The signal is active when count register bits 21 through 23 are all zeros. Since the 'last word' trigger is on and count bits 21 through 23 equal zero, all specified data bytes for the read operation in progress have been received by the channel; i.e., the count register has been reset. An overrun condition will occur during a read operation if the 'last word' trigger is on, the value in the byte count latches equals zero, and bits 21 through 23 are not equal to zero (the 'BCL = 0' and 'last word' trigger conditions specify that the operation is complete (B-register full, last doubleword); if count bits 21 through 23 equal other than zero, the count specifies that more data is required to complete the operation). A detected overrun condition turns on the 'chain check' latch to indicate the detected error.

The 'count less than 2 words' signal (Diagram 4-10) is effective for both write CDA and read CDA operations. The signal is activated when all count register bits except bit 20 equals zeros.

For write CDA operations, the 'count less than 2 words' signal provides the same function as described for the 'count equals 2 words' signal (gates the CTB bits to the mark-B register and turns on the 'retain storage' latch).

For the read CDA operation, the 'count less than 2 words' signal is sampled when the channel receives a new CCW. If the new count is less than 16 bytes ('count less than 2 words' signal active), the 'read CDA A-register loaded' signal is active, and the 'BC greater than CTB' signal is active, the channel has already received more data bytes from the control unit than specified by the count in the new CCW. Thus, a chain check condition exists and the chain check latch is turned on. If the 'count less than 2 words' signal is inactive under the conditions described, no chain check condition exists.

The 'count equal or less than 2 words' signal (Diagram 4-10) is sampled during read-type and write-type operations for a last word condition after data transfers have begun. The sampling occurs just prior to the time the count register contents are decremented as the result of a storage access. When the active 'count equal or less than 2 words' signal is sampled, the 'last word' trigger is turned on. The 'count equal or less than 2 words' signal is activated by either the 'count equals 2 words', 'count less than 2 words' or 'count equal or less than 1 word' signal.

The 'count equal or less than 1 word' signal is effective during read CDA and write CDA operations and during sequence 1 of the initial selection routine. This signal is sampled when a valid CCW is received by the channel and before the CCW information assumes control of data transfer operations. If the count in the new CCW specifies only one doubleword transfer, the 'count equal or less than 1 word' signal is activated; when sampled, the active signal causes turn-on of the 'last word' trigger. The 'count equal or less than 1 word' signal is activated when count register bits 7 through 19 equal all zeros and either bit 20 equals zero or bits 21 through 23 equal zeros.

The 'count less than 1 word' signal (Diagram 4-10) is effective only during read CDA operations. The signal line is sampled at two different times during a read CDA operation to determine if a chain check condition exists. When the new CCW is in the channel, the 'count less than 1 word' signal is first sampled prior to the count + DAB operation. If, during this sampling, the 'read CDA A register loaded' signal is active, the 'chain check' latch is turned on since the channel has already received more bytes than specified by the count in the new CCW. After the count + DAB operation is performed, and the 'count less than 1 word' and 'BC greater than CTB' signals are active at this sampling, the 'chain check' latch is turned on. In this case, the chain check condition exists because only one doubleword is to be transferred, and the B-register already contains more data bytes than specified by the CTB value. If the first sampling of the 'count less than 1 word' results in a chain check condition, the second sampling is not performed.

ADDER

- Adder consists of bit 0-23 positions.
- Positions 21-23 are full adder; two inputs may be added.
- Positions 0-20 increment or decrement by 1 only.
- Position 20 receives ±1 input (increment or decrement).
- Carries or borrows processed from full adder bit 21 position through adder positions 0.
- Six 4-bit groups comprise adder positions 0-23; group 1 consists of low-order bits 20-23 and group 6 consists of high order bits 0-3.

- 'Parity predict' logic generates predicted parity bits from gated register bits and group 1 through group 6 bits; predicted parity bits gated to registers.

- 'Parity generator' logic generates result parity bits from adder sum bits.

- 'Parity check' logic compares predicted parity and result parity bits; mismatch turns on 'channel control check' latch when results are sampled.

The adder (Diagram 4-11) is capable of performing update data (increment or decrement) operations and limited full add operations. The adder consists of 24 positions designated 0 through 23; position 0 is the high-order position and position 23 is the low-order position. Adder positions 0 through 23 are divided into 4-bit groups designated group 1 through group 6. Group 1 consists of low-order bits 20 through 23, group 2 of bits 16 through 19, etc., with group 6 consisting of high-order bits 0 through 3.

Low-order positions 21 through 23 are full-adder positions; i.e., two values may be added at these positions to obtain a sum value. Positions 0 through 20 are only capable of incrementing or decrementing one value by 1; however, these positions process carries (when incrementing) or borrows (when decrementing) from position 21 (the full adder high-order position).

Position 20 of the adder (Diagram 4-11) receives the plus or minus 1 input for an increment or decrement operation. Carries or borrows resulting from the operation are processed up to and including position 0.

The adder outputs are applied to adder latches. Each latch in the adder is actually a polarity hold (Diagrams 4-12 and 4-13). When the 'latch adder' signal is inactive, the adder sum outputs of the polarity holds equal the inputs to the polarity holds. When the 'latch adder' signal is activated, the bit values present in the polarity holds at that time are locked and remain stable until the signal drops.

The adder is used by the channel to perform the following operations: (1) increment the data address register value by 1 on each storage transfer for read, write or sense operations; (2) decrement the data address value by 1 on each storage transfer for a read backward operation; (3) add data address register bits 21 through 23 (the DAB) to the count register contents during CCW setup operations; (4) decrement the count register contents by 1 on each storage transfer; (5) increment the command address register contents by 1 each time a new CCW is fetched during chaining operations; and (6) calculate a residual count value by adding the twos' complement of the byte count register to the count register minus 1 value with hot 1's applied to the adder 21 through 23 positions (See Diagram 4-12).

Gated parity bits and data bits for each of the above operations are applied to the adder 'parity predict' logic (Diagram 4-11) which also receives group 1 through group 6 inputs from the adder groups. From these inputs, the 'parity predict' logic generates three predicted parity bits, which are gated to the appropriate registers along with the adder sum bits.

The adder sum bits are also applied to the adder 'parity generator' logic which generates parity bits ('result parity' in Diagram 4-11) for each of the three 8-bit adder sum bytes. The result parity bits and predicted parity bits are compared at the 'parity check' logic. If the bits do not match on a bit to bit basis, the 'adder parity check' signal is activated. When sampling during the adder operation, the active 'adder parity check' signal turns on the 'channel control check' latch to end the operation in progress. The 'adder parity check' signal remains inactive if the 'result parity' and 'predicted parity' bits match. Adder operations are described in greater detail in the following paragraphs.

Adder Increment Data Address

- Data address register bits gated to adder.

- 'Gate data address plus 1' signal changes register bit value at adder sum 20 output of polarity hold.

- If register bit equals 1, 'group 1 carry' signal generated; group 2 register bit inputs changed at polarity hold outputs according to value of inputs.

- 'Latch adder' signal locks incremented 'adder sum' value in polarity holds.

- 'Gate adder to data address register' signal gates adder sum bits to data address register.

Adder operations to increment the data address register contents by 1 (eight bytes) begin when the 'gate data address +1 to adder' and 'gate data address to adder' signals are activated (Diagram 4-12). The 'gate data address to adder' signal gates data address register bits 0 through 20 to adder positions 0 through 20. The 'gate data address +1 to adder' signal is applied to adder position 20 where it activates the 'full add carry or increment' (+1) signal. This signal is OR'ed, then exclusive OR'ed with gated data address bit 20 ('group 1, bit 4' signal). If the 'group 1, bit 4' signal equals 1, the output of the exclusive OR provides an adder sum 20 equals 0 bit from the position 20 polarity hold. At the same time, the 'full add carry or increment' and 'group 1 bit 4 = 1' signals are AND'ed to generate a 'group 1 carry' signal to adder group 2. If the 'group 1, bit 4' signal equals 0, the position 20 polarity hold output provides an adder sum 20 equals 1 output and the 'group 1 carry' signal is not generated.

When the 'group 1 carry' signal is activated, the group 2 through group 6 blocks change data register inputs to polarity holds as required to provide an adder output incremented

by 1. Diagram 4-13 shows adder group 3 logic and the 'group 2 carry' signal from the 'group 2' logic (the operation of 'group 2' through 'group 6' logic is identical.) Note in Diagram 4-13 that an active 'group 2 carry' signal changes the value of the register bit input 'group 3 bit 1' signal to the 'position 15' polarity hold at the 'adder sum 15' output of the polarity hold. Also, if the 'group 3, bit 1' signal equals 1, the active 'group 2 carry' signal changes the value of the 'group 3, bit 2' signal (from the register) at the 'adder sum 14' output of the 'position 14' polarity hold. (If the 'group 3, bit 1' signal equals 0, the input and output of the 'position 14' polarity hold are the same). If the 'group 2 carry' signal is inactive, the 'adder sum' outputs of all polarity holds in group 3 are the same as the inputs from the register.

For the data address register incrementing operation, a 'group 1 carry' signal initiated at position 20 (Diagram 4-12) is propogated through the remaining groups to the extent necessary to provide an adder sum 0 through adder sum 20 output which is incremented by 1 from the input value.

When the 'latch adder' signal is activated, the incremented values in the position 0 through position 20 polarity holds are locked. Subsequently, the 'gate adder to data address register' signal gates the adder sum bits into the data address register. (If the value in the data address register, before incremented, is from the CAW or a TIC CCW, the 'gate adder to command address register' signal gates the adder sum bits to the command address register, rather than the data address register.)

The 'adder parity check' signal is strobed when the 'adder sum' bits are gated to the data address register. If the signal is active, the 'channel control check' latch is turned on to end the operation.

Adder Decrement Data Address

- Data address register bits gated to adder.
- 'Read backward' and 'sequence 4 and not T4' signals change register bit value at adder sum 20 output of polarity hold.
- If register bit equals 0, 'group 1 borrow' signal generated; group 2 register bit inputs changed at polarity hold outputs according to value of inputs.
- 'Latch adder' signal locks decremented adder sum value in polarity holds.
- 'Gate adder to data address register' signal gates adder sum bits to data address register.

Adder operations to decrement the data address register contents by 1 (read backward operation) begin when the 'gate data address to adder', 'read backward' and 'sequence 4 and T4' signals are activated (Diagram 4-12). The 'gate data address to adder' signal gates data address register bit 0 through 20 to adder positions 0 through 20. The 'read backward' and 'sequence 4 and not T4' signals are ANDed to activate the 'decrement position 20' signal. As during the 'data address increment' operation, the 'decrement position 20' signal changes the value of the 'data address 20' bit ('group 1, bit 1' signal) at the 'adder sum 20' output of the 'position 20' polarity hold. If the 'group 1, bit 4' signal equals 0, the 'decrement position 20' signal also activates the 'group 1 borrow' signal. The 'group 1 borrow' signal is supplied to the 'group 2' logic to change values of bits at polarity hold outputs as required.

An active 'group 2 borrow' signal changes the value of the group 3, bit 1 input to the position 15 polarity hold at the adder sum output of the polarity hold (Diagram 4-13). If the 'group 3, bit 1' signal equals 0, the 'group 2 borrow' signal activates the 'bit 1 carry or borrow' signal; this signal causes the value of the group 3 bit 2 input to change at the adder sum 14 output of the position 14 polarity hold.

For the data address decrementing operation, a 'group 1 borrow' signal initiated at position 20 (Diagram 4-12) is propogated through the remaining groups to the extent necessary to provide an adder sum 0 through adder sum 20 output which is decremented by 1 from the input value.

When the 'latch adder' signal is activated, the decremented values in the adder polarity holds are locked. Subsequently, the 'gate adder to data address register' signal gates the adder sum bits into the data address register.

An adder parity check condition will turn on the 'channel control check' latch to end the operation.

Adder DAB Plus Count Operations

- 'Gate DAB to adder' and 'gate count to adder' signals gate DAB and count register values to adder.
- DAB bits and count register bits 21-23 added by full-adder positions 21 through 23.
- 'Bit carry' signals generated by each position (21-23) as determined by input values to each position and input/output values of lower-order positions.
- 'Full add carry' signal to position 20 generated, if necessary, by position 21 logic.
- 'Latch adder' signal locks summed value in adder polarity holds; 'gate adder to count register' signal gates adder sum bits to count register.

Adder operations to add the DAB (data address register bits 21 through 23) to the count register value are performed

during CCW setup operations. This adder operation begins when the 'gate DAB to adder' and 'gate count to adder' signals are activated. The 'gate count to adder' signal gates count register bits 7 through 23 to adder positions 7 through 23. The 'gate DAB to adder' signal gates the DAB bits to positions 21 through 23 of the adder (Diagram 4-12). At position 23 of the adder, the 'data bit 23' signal (derived from low-order bit 1 of the byte count register) is exclusive OR'ed with count register bit 23. If both bits are the same, the 'count 23 exclusive OR'ed with data bit 23' signal is inactive and the adder sum 23 output of the positions 23 polarity hold is at a logic 0 level. If the two inputs are unalike, the adder sum 23 output is at the logic 1 level.

If both 'data bit 21' and 'count register 23' signals are logic 1's, the 'position 23' logic supplies a 'change position 23' signal to an exclusive OR at the input to the position 22 polarity hold (Diagram 4-12). At position 22, the 'data bit 22' and 'count register 22' signals are exclusive OR'ed, with bits of equal value deactivating the 'count 22 exclusive OR'ed with data bit 22' signal. If this signal is inactive with the 'change position 22' signal from position 23, the 'adder sum 22' output of the 'position 22' polarity hold is at a logic 1 level; if the 'count 22 exclusive OR'ed with data bit 22' signal is inactive with the 'change position 22' signal inactive, a logic 0 level is present on the 'adder sum 22' output. Conversely, if the 'data bit 22' and 'count register bit' signals are not alike, the active 'count 22 exclusive OR'ed with data bit 22' signal causes a logic 1 on the 'adder sum 22' line if the 'change position 22' signal is inactive, or a logic 0 if the 'change position 22' signal is active. The described position 22 operations are summarized as follows:

Data Bit 22 Value	Count Reg. 22 Value	Change Position 22 Signal	Adder Sum 22 Value
1	1	active	1
1	1	inactive	0
0	0	active	1
0	0	inactive	0
1	0	active	0
1	0	inactive	1
0	1	active	0
0	1	inactive	1

At adder position 22, the 'change position 21' signal is activated if: (1) the 'data bit 22' and 'count register 22' signals are both 1's; or (2) if the 'data bit 23' and 'count register 23' signals are both 1's and the 'count register 22 exclusive OR'ed with data bit 22' signal is active. The 'change position 21' signal is processed by position 21 in the manner described for position 22.

At position 21, a 'full add carry' signal is generated if: (1) the 'data bit 21' and 'count register 21' signals are both 1's; (2) the 'data bit 22' and 'count register 22' signals are both 1's, and either or both of the 'data bit 23' and 'count register 23' signals are 1's; or (3) both the 'data bit 23' and 'count register 23' signals are 1's, either or both 'data bit 22' and 'count register 22' signals are 1's, and either or both the 'data bit 21' and 'count register 22' signals are 1's.

If the 'full add carry' signal is genrated, it is applied to position 20 where it activates the 'full add carry or increment' signal. The operation of position 20 with and without an active 'full add carry or increment' signal is the same as described for the data address incrementing operation (Refer to "Adder Increment Data Address"). The operation of other adder increment positions are also described in "Adder Increment Data Address", except that count register bits are present at these positions, rather than data address register bits.

A typical 'DAB plus count' operation for adder positions 20 through 23 is as follows:

	Full Adder Positions			
Adder Positions	20	21	22	23
DAB value (5)		1	0	1
Count Bits 20-23 (3)	0	0	1	1
Inputs & carrys	No Group 1 Carry	Different & Full-Add Carry	Different & Bit Carry	Same & Bit Carry
Adder Sum Outputs (8)	Full-Add 1 carry ←	0	0	0

When the 'latch adder' signal is activated, the values in the polarity holds are locked; subsequently, the 'gate adder to count register' signal gates the adder sum 7 through adder-sum 23 and predicted parity bits to the count register.

Adder Decrement Count

Adder operations are performed to decrement the count register contents by 1 (8 bytes) for channel data doubleword transfers. Except as follows, adder operations for count decrementing are the same as described for data address decrementing. (Refer to "Adder Decrement Data Address".)

The decrement count operations begin when the 'gate count to adder' and 'gate count minus 1 to adder' signals are activated (Diagram 4-12). The 'gate count to adder' signal gates count register bits 7 through 23 to adder positions 7 through 23. The 'gate count minus 1 to adder' signal activates the adder 'decrement position 20' signal. From this point, adder decrementing operations are identical to the data address decrementing operation. (Count bits 21 through 23 to full-add positions 21 through 23 are not changed at the 'adder sum 21' through 'adder sum 23' outputs.) The 'latch adder' signal locks the decremented values in the polarity holds, and the 'gate adder to count register' signal gates the 'adder sum 7' through 'adder sum 23' and 'predicted parity' bits to the count register. The 'adder parity check' signal is strobed and, if active, turns on the 'channel control check' latch to end the operation.

Adder Increment Command Address

Adder operations are performed to increment the command address register contents by 1 (8 bytes) each time the channel fetches a new CCW during chaining operations. Except as follows, adder operations for command address incrementing are the same as described for data address incrementing. (Refer to "Adder Increment Data Address.")

The increment count operations begin when the 'gate command address to adder' and 'gate command address plus 1 to adder' signals are activated (Diagram 4-12). The 'gate command address to adder' signal gates command address register bits 0 through 20 to adder positions 0 through 20. The 'gate command address plus 1 to adder' signal activates the adder 'full add carry or increment' signal. From this point, adder incrementing operations are identical to the data address incrementing operation.

The 'latch adder' signal locks the incremented values in polarity holds, and the 'gate adder to count register' signal gates the 'adder sum 0' through 'adder sum 20' and 'predicted parity' bits to the command address register.

Adder Residual Count Calculation

- Two's complement of BCR gated to adder.
- Count gated to adder.
- "Hot 1's" applied to adder positions 21 through 23.
- Full add carry condition inhibits 'decrement position 20' signal; no 'full add carry' signal generated.
- Not full add carry condition enables 'decrement position 20' signal.

The adder performs residual count calculations when the channel detects an IL condition and requires a value in the count register to reflect the number of bytes not transferred for the read-type or write-type operation.

Adder residual count operations begin when the 'gate count to adder', and 'gate byte count and count -1 IL' signals are activated (Diagram 4-12). Since the purpose of the residual count operation is to subtract the byte count from the count in the count register, the 'gate byte count and count -1, IL' signal gates the two's complement of the byte count register bits to adder full-add positions 21 through 23. (The two's complement is gated, since the full-adder portion is only capable of summing inputs.)

In addition, the 'gate byte count and count -1, IL' signal is applied to the exclusive-OR input to the 'position 23' polarity hold latch and to the bit carry logic of adder positions 21 through 23'; this places "hot 1's" to these positions. The 'gate byte count and count -1, IL' signal also: (1) inhibits generation of the 'full add carry' signal to the 'bit 20' position; and (2) enables or disables generation of the 'decrement position 20' signal depending upon whether a full add carry condition exists at the 'position 21' logic. (A full add carry condition inhibits generation of the 'decrement position 20' signal.)

The 'gate count to adder' signal gates count register bits 7 through 23 to adder positions 7 through 23.

At adder position 23 (Diagram 4-12) the two's complement of byte count register bit 1 ('data bit 23' signal) is exclusive-OR'ed with the 'count register 23' signal. If the two signals have different values, the 'count 23 exclusive-OR'ed with data bit 23' signal is activated. This signal is further exclusive-OR'ed with the "hot 1" at the input to the position 23 polarity hold; this exclusive OR'ing provides a logic 0 adder sum 23 output from the position 23 polarity hold. Conversely, if the 'count 23 exclusive-ORed with data bit 23' signal is not activated ('count register 23' and 'data bit 23' signals are of different values), the "hot 1" is exclusive-OR'ed to produce a logic 1 adder sum 23 output.

A 'change position 22' signal to position 22 is generated, except when both gated inputs to position 23 are logic 0's. If both position 23 inputs are logic 1's, the 'change position 22' signal is activated by AND'ing the 'data bit 23' and 'count register 23' signals. If the two inputs have different values, the 'count register 23 exclusive-OR'ed with data bit 23' is ANDed with a "hot 1" to activate the 'change position 2' signal. If both position 23 inputs are 0's, the 'change position 2' signal is not activated.

Operations at the position 22 logic are dependent upon the values of the input signals and whether the 'change position 2' signal is active or inactive. Exclusive-OR'ing of the 'data bit 22' and 'count register 22' signals are as described for position 23. If the 'count 22 exclusive-OR'ed with data bit 22' signal is activated and the 'change position 22' signal is inactive, the adder sum 22 output of the polarity hold is a logic 1; if the 'change position 22' signal is active, the adder sum 22 output is a logic 0. Conversely, if the 'change position 22' signal is active, an active 'count 22 exclusive-OR'ed with data bit 22' signal provides a logic 0 adder sum 22 output and an inactive 'count 22-exclusive OR'ed with data bit 22' signal provides a logic 1 adder sum 22 output.

Like the 'position 23' logic, the 'position 22' logic generates a bit carry ('change position 21')signal, except when both position 22 inputs equal 0's or both position 23 inputs equal 0's.

Position 21 operation to provide an adder sum 21 output is as described for position 22; i.e., the value of 'adder sum 21' signal is dependent upon the values of the two position 21 inputs and whether the 'change position 21' signal is active or inactive.

The 'full add carry' logic associated with position 21 detects whether a full add carry condition exists (Diagram 4-12). A full add carry condition exists except when both gated inputs to positions 21, 22, or 23 equal 0's. A full add

carry condition is AND'ed with a "hot 1" to activate the 'full add carry and gate BC and count-1 IL' signal. This signal inhibits both the 'full add carry' and the 'decrement position 20' signals to the position 20 logic. In effect, this allows the adder to provide the outputs on the 'adder sum 7' through 'adder sum 20' lines that are the same values as the count register bit 7 through bit 20 inputs. In this case, only the adder sum 21 through adder sum 23 outputs are altered as a result of summing the two's complement of the byte count register bits and count register bits 21 through 23.

If the two's complement of the byte count register bits are all 0's and count register bits 21 through 23 are all 0's, the full add carry condition is not detected and the 'decrement position 20' signal is activated at adder position 20. This permits decrementing of adder positions 7 through 20 as required by the count bit values to these positions.

The two following examples illustrate the reason for inhibiting the 'decrement position 20' signal when both gated inputs to positions 21 through 23 are 0's and for activating the 'decrement position 20' signal when the gated inputs are not all 0's.

Example 1 (Subtract Byte Count 7, Decimal, From Count 8, Decimal)

	20		21	22	23	
Adder Position						
BCR Value			1	1	1	(7 Decimal)
2's Complement BCR Value			0	0	0	
				↓		
				Add		
Count Bit Values	1		0	0	0	(8 Decimal)
		Decrement				
Adder Sum	0	20 ←	0	0	1	(1 Decimal)

Example 2 (Subtract Byte Count 4, Decimal, From Count 12, Decimal)

	20		21	22	23	
Adder Position						
BCR Value			1	0	0	(4 Decimal)
2's Complement BCR Value			0	1	1	
				↓		
				Add		
Count Bit Values	1		1	0	0	(12 Decimal)
		No Decrement				
Adder Sum	1	20 ←	0	0	0	(8 Decimal)

The 'latch adder' signal locks the subtracted values in the polarity holds, and the 'gate adder to count register' signal gates the adder sum 0 through adder sum 20 and predicted parity bits to the command address register.

COMMAND ADDRESS REGISTER

- Command address register contains updated address of second and subsequent CCW's to be fetched from storage.
- The register is reset to all zeros with odd parity prior to fetching the CAW on a start I/O operation.
- First CCW address is updated by 1 (eight bytes) and entered into the command address register.
- When each new CCW is fetched, the address in the command address register is gated to the SAB, then updated.
- The command address register contents are gated to the SBI on a logout sequence or CSW store operation.

The command address register maintains an updated storage address to be used for fetching consecutive CCW's when this is a requirement of the current channel operation. The command address register (Figure 2-15) consists of 24 latches (21 data and 3 parity latches). The contents of the command address register are displayed on 24 COMMAND ADDRESS indicators on the CE control panel.

The command address register is used during a channel operation initiated by a Start I/O instruction. Prior to receipt of the CAW from storage, the command address register is reset to all zeros with odd parity by the 'command address ingate reset' signal. When the CAW, which contains the address of the first CCW, arrives at the channel, the CCW address is gated into the data address register. Subsequently, the contents of the data address register are gated to storage to obtain the first CCW. The address in the data address register is also gated to the adder, where it is incremented by 1 to specify a storage address eight bytes (one doubleword) higher than the address originally in the data address register. The incremented address is then gated to the command address register. If the first CCW obtained from storage specifies chaining, the address in the command address register is gated to the SAB at the time the channel requires the next CCW. Assuming that a second CCW is required, the address in the command address register, in addition to being gated to the SAB, is also gated to the adder input where it is again incremented by 1 and gated back into the command address register. The address in the command address register is continually updated each time a new CCW is fetched until CCW's from sequential addresses are no longer required by the current operation.

Under special conditions, the address in the command address register may be updated twice before the next CCW is fetched. As a result, the address in the command address

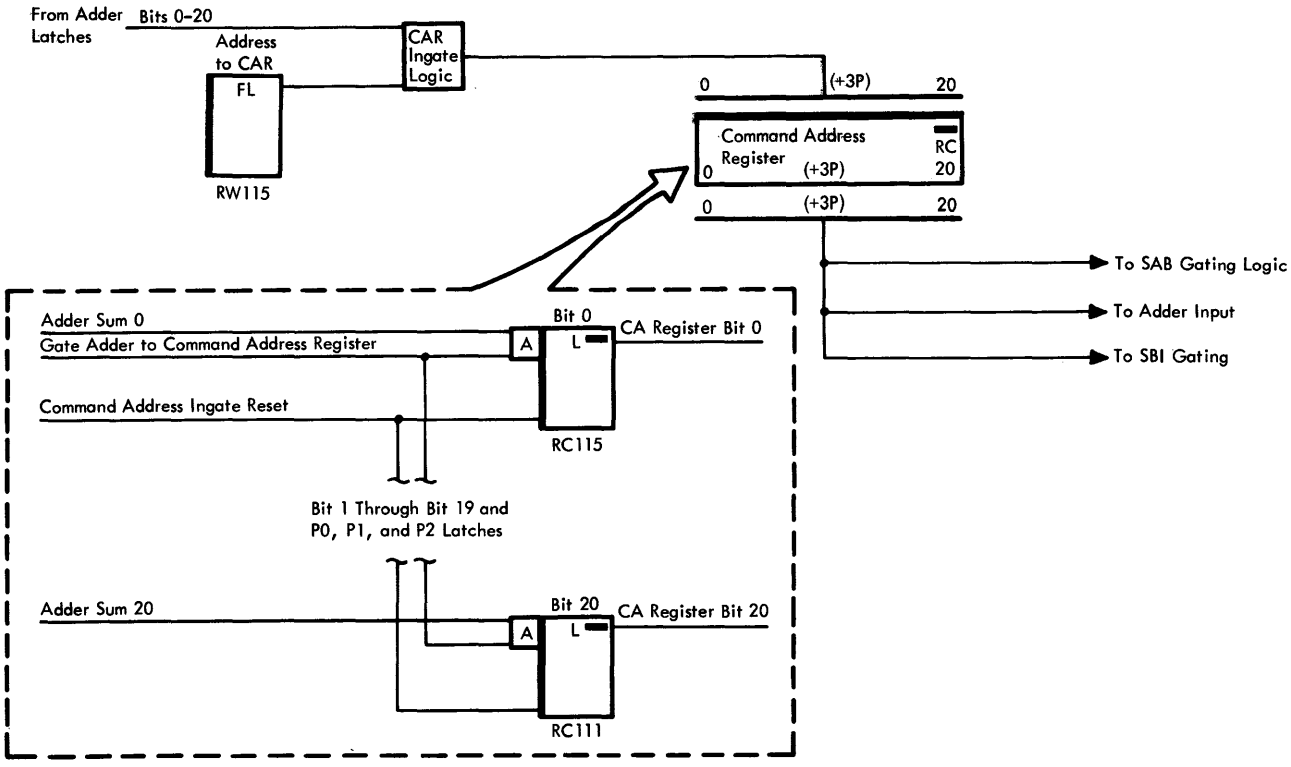


Figure 2-15. Command Address Register

register specifies a CCW address 16 bytes higher than the address of the current CCW; i.e., the second update causes the channel to "jump" one CCW and fetch the second consecutive CCW instead. The second increment of the address in the command address register occurs when the chain command flag in the channel is on, and the status byte from the I/O device contains active status modifier and device ending bits.

Command address register bit positions 0 through 20 are gated to positions 8 through 28 of the SBI on three conditions: (1) during log word 1 of a logout sequence; (2) during a CSW store operation with the 'CCW valid' trigger active; and (3) during a CSW store operation with the 'CDA' latch on. Command address register parity bits P0, P1, and P2 are gated to SBI positions 49 through 51 on log word 3 of a logout sequence.

BYTE COUNTER CIRCUITS

The byte counter circuits control the gating of information to and from the eight-byte B-register during the transfer of information (on a byte basis) between the channel and the I/O device. In addition, the byte count circuits perform bit comparisons which are used to determine the progress of the information transferred through the channel so that (depending upon the operation in progress) ending sequence, check conditions, control functions, and certain reset functions may be initiated at the proper times.

Figure 2-16 is a simplified block diagram of the byte counter circuits. Basically, these circuits consist of a byte counter, a byte count encoder, and a byte count comparator. The byte counter is capable of counting from 0 to 7 on a wraparound basis. In other words, the counter can count from 0 through 7 and begin counting again from 0 through 7 without first being reset. The byte counter maintains a count which, when encoded by the byte count encoder, represents a corresponding byte position of the B-register. For each successful start I/O, IPL or FLT operation initiated in the channel, the byte counter is reset to zero. For a start I/O operation, the DAB (bits 21 through 23) from the data address register is gated into the byte counter (following the reset) to specify the byte position of the B-register into or from which the first data byte is to be gated. The three bit output of the byte counter is supplied to the byte count encoder where it is encoded to select ingating or outgating lines to the B-register. Each time a data byte is gated to or from the B-register, the byte counter is changed to "point" at the next byte position of the B-register. This process of gating into or from the B-register based upon the count in the byte counter at the time, and then changing the byte counter count, continues until all data specified by the

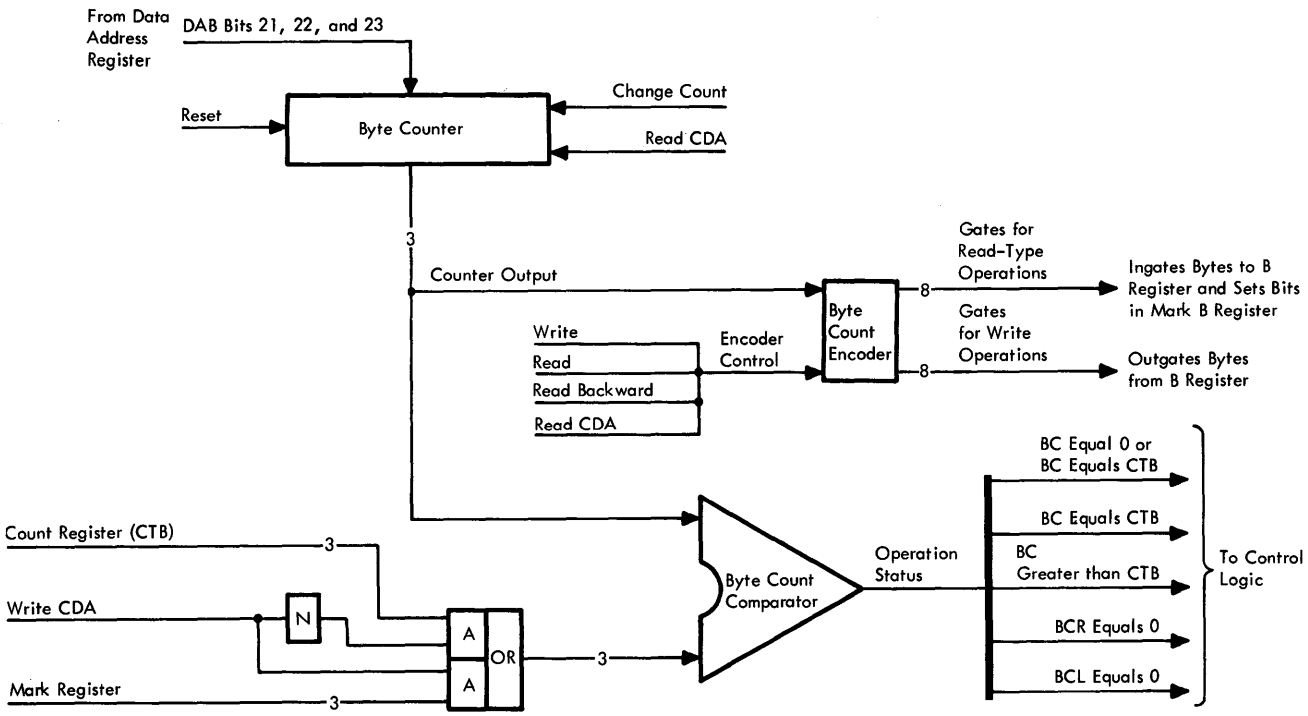


Figure 2-16. Byte Counter Circuits

current operation is transferred, or until the operation is prematurely terminated because of a detected error or wrong length record condition. For a read chain data operation, the byte counter is reset to 0 after the last data byte specified by the current CCW is gated into the B-register. If the I/O device involved in the read CDA operation continues to supply the channel with data while the new CCW is being obtained from storage, the counter must count to provide the necessary encoded gating signals to the B-register. In this case, the new DAB (available after the new CCW is in the channel), is gated to the byte counter and is effectively OR'ed with the count existing in the byte counter. For a typical example, assume the byte count register (BCR) contains a value of 1 (one data byte has been gated) and the DAB equals 4. After the DAB is OR'ed into the BCR, the new BCR value is 5. This operation of effectively adding the DAB and the existing count provides a byte counter output which is used for two purposes: (1) it is used by the byte count encoder to "point" at a byte position of the B-register as previously described; and (2) it is used by the byte count comparator and the channel control logic to determine if the address specified by the DAB and the number of bytes entered into the B-register are compatible for continued operation.

Note: For a read CDA operation, programming limitations are placed upon the new DAB count versus the number of bytes gated into the B-register before the DAB arrives. If these limitations are exceeded, the operation is terminated. (See "Byte Count Encoder" in this chapter.)

The byte count encoder receives the count output of the byte counter and encodes this count to gate data bytes to or from the B-register. The outputs of the byte count encoder are 16 gating lines. Eight gating lines are used to gate data bytes into the B-register and corresponding bits into the mark-B register; the other eight gating lines are for gating data bytes from the B-register. Because gating bytes into and from the B-register are mutually exclusive operations, and because of the different gating requirements for each operation, control of the byte counter is dependent upon the channel operation [write, read (or sense), read backward, or read CDA] being performed at the time the counter outputs are being encoded. For example, if the channel is performing a write operation, the byte count encoder must encode one of the gating lines which gates a byte from the B-register; if a read operation is being performed, the byte count encoder must encode one of the lines which gates a byte to the B-register. For a read backward operation, bytes are gated into the B-register in the reverse order for that of a read operation; thus, the byte count encoder must reverse the order in which gating lines to the B-register are encoded. (For example, if the counter output specifies gating line 0, the byte count encoder must encode the count of specify ingating the 7 to the B-register.) For read CDA operations, data is doublegated into the B-register (see "Byte Count Encoder" in this Chapter) and, for each

byte to be doublegated, the byte counter must encode the counter input to select two gating lines to the B-register.

The byte count comparator monitors the progress of read-type and write operations by (1) examining the byte counter (BC) output, and (2) by comparing the count B output of the count register with the BC output. The byte count comparator outputs ('BC equals 0 or BC equals CTB', 'BC equals CTB', 'BC greater than CTB', 'BCL equals 0' and 'BCR equals 0') are sent to the channel control logic, and are gated at various times during channel operations to determine the subsequent operational sequence to be performed. It is beyond the scope of this text to fully describe the function of each byte count comparator output, since each performs different functions for the different channel operations. (See Chapter 3, "Principles of Operation "). For example, during a read operation, one function of the 'BC latches equals 0' signal (when active) is to enable the contents of the B-register to be gated to the A-register for subsequent storage in main storage (assuming the A-register is not already full and the read operation is not terminated for some reason). During a write operation, the 'BC latches equal 0' signal (when active) initiates a data transfer operation from the A-register to the B-register, assuming that the A-register contains data from main storage and that the write operation is not terminated for some reason. The above examples describe only two of several functions of the 'BC latches equal 0' signal. Like the 'BC latches equal 0' signal, the 'BC equals CTB' signal has several functions in the channel control logic. To cite one example, if the 'BC equals CTB' signal is active during a write operation, data chaining is not involved, and the end of data transfer is indicated ('last word' trigger in on), the active 'BC equals CTB' signal initiates a channel ending sequence (sequence 5). The 'byte count greater than CTB' output is significant only during read chain data operations and is sampled at the 'chain check' logic to determine if a chain check condition exists.

Operation of the byte count comparator during a write CDA operation is unique in that the CTB (required for the BC to CTB comparison) is contained in the mark-B register for a portion of the write CDA operation. During this time, the CTB contents of the mark-B register are gated to the byte count comparator for comparison with the BC. The CTB is placed in the mark-B register just prior to fetching the next CCW so that data transfers from the current write operation may continue (which requires the current count) while the new count in the new CCW is entered into the count register.

Byte Counter

The byte counter isdescribed functionally at the block diagram level (see "Byte Counter Simplified Description") and at the logic diagram level (see "Byte Counter Logic Description").

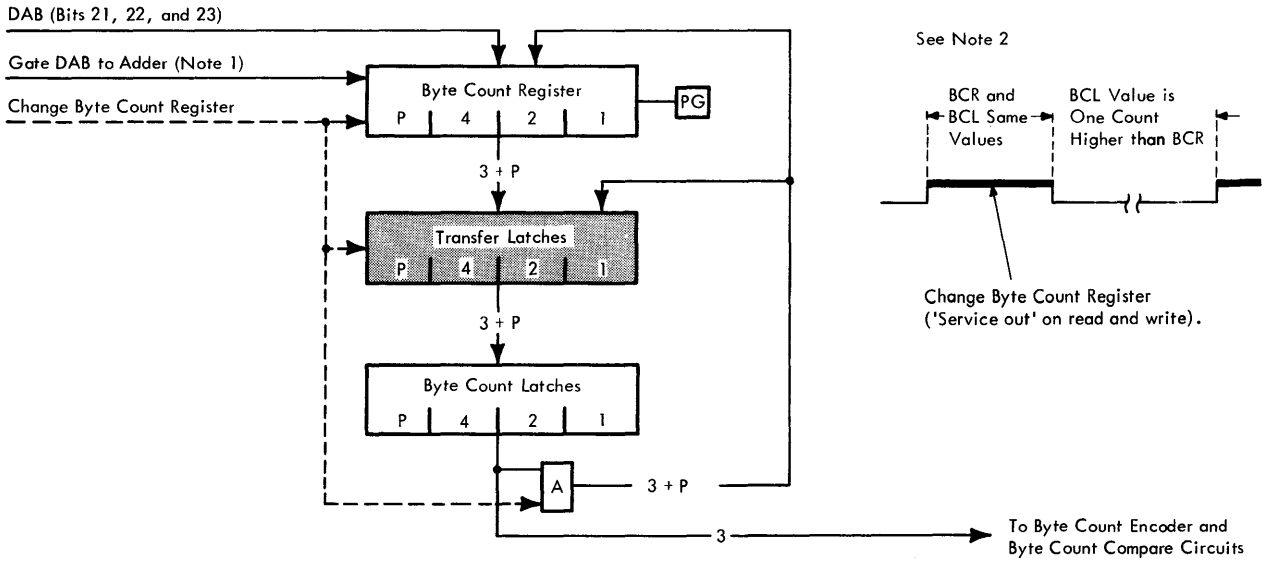
Byte Counter Simplified Description

The function of the byte counter, as stated in the previous paragraph, is to provide count outputs which, when encoded by the byte count encoder, select individual byte positions of the B-register for data ingating or outgating. The byte counter (Figure 2-17) consists of: (1) a byte count register; (2) a group of transfer latches; and (3) a group of byte count latches. Since the minimum byte count is 0 and the maximum is 7, the above register and two sets of latches each consist of four stages: three count bit stages and a parity bit stage. The count stages are designated decimally; i.e., stages 1, 2, and 4.

The basic operation of the byte counter is illustrated in Figure 2-17. Assuming that a CCW specifying a read-type or write-type operation has just been entered into the channel, the 3-bit DAB (which specifies byte, word, or doubleword boundaries) is gated into the byte count register by the 'gate DAB to adder' signal. A parity generator in the byte count register circuits assigns odd parity to the register.

The outputs of the byte count register are applied to the transfer latches. The transfer latches upgrade the count in the byte count register and transfer the upgraded count to the byte count latches. The byte count latches, at the end of the operation described above, contain a count one higher than the byte count register (or in the case where the byte count register contains a count of 7, the byte count latches contain a count of 0 due to the "wraparound" manner in which the byte counter counts). In Figure 2-17, note 1 illustrates the 1 and 0 values contained in the byte count register, transfer latches, and byte count latches after a DAB with a count of four is gated into the byte count register. Note that the byte count register contains a decimal value of four and that the transfer and byte count latches contain a decimal value of five.

Once the DAB count has been entered into the byte counter, the byte counter must be advanced each time a data byte is gated to or from the B-register. Assume that the byte count register contains a count of four as indicated in the previous paragraph and that a read operation is in progress. While I/O data is being gated into the B-register (as a result of the 'service out' signal), the byte count register receives a 'change byte count register' signal which gates the contents of the byte count latches to the byte count register. For the duration of the 'change byte count register' signal, the byte count register and byte count latches contain a count of five; in other words, the count of the byte count register equals that of the byte count latches. (See the waveform and illustration indicated by note 2 in Figure 2-17). When the 'change byte count register' signal falls, the count in the byte count latches is changed to a 6 by the upgrade operation of the transfer latches. The manner in which the byte counter count is changed is significant in that the byte count latch outputs remain unchanged during the count change of the byte count register, allowing the byte count encoder to



Notes:

1. 'Gate DAB to adder' signal gates DAB (on each non-TIC CCW) to the byte count register. Below is an example of a DAB of four (decimal) after gating is complete.

P	4	2	1	Position
0	1	0	0	Byte Count Register
1	1	0	1	Transfer Latches
1	1	0	1	Byte Count Latches

2. The example below illustrates the first updating of the byte count register following the ingating of the DAB illustrated in Note 1. Refer to the waveform indicated by "Note 2" in this Figure.

	P	4	2	1	Position
During Updating, BCL = BCR.	1	1	0	1	Byte Count Register
	1	1	0	0	Transfer Latches
	1	1	0	1	Byte Count Latches
After change, BCL = BCR + 1.	1	1	0	1	Byte Count Register
	1	1	1	0	Transfer Latches
	1	1	1	0	Byte Count Latches

Figure 2-17. Byte Counter Simplified Block Diagram

encode the byte count latch outputs and gate the appropriate B-register byte position during the change period.

For a write operation, the byte counter operates as described for the read operation, except that the 'change byte count register' signal is initiated as a result of the 'service out' signal to the I/O device. During the time the BCR count equals the BCL count, the BCL count is encoded and the appropriate byte is gated from the B-register.

Table 2-1 illustrates byte counter stepping beginning with the counter is a reset condition. Bit values are shown for the byte count register, transfer latches, and byte count latches for both the period during which the change (stepping) is performed and after the change is completed. Note that, in each case (counts 0 through 7 and the wrap-around count of 0 as shown in Figure 2-17) the count in the byte count register equals the count in the byte count latch during the count change; in each case, after the count change is completed, the count in the transfer and byte count latches is one count ahead of the count in the byte count register.

Byte Counter Logic Description

The byte counter is shown in logic form in Diagram 4-14 and consists primarily of the 'DAB ingating' logic, the 'change' logic, the BCR logic, the 'transfer latch' logic, and the 'byte count latch' (BCL) logic. Prior to any operation for which the byte counter is to begin a new count, all four byte count register stages receive a reset signal causing the BCR 1, 2, and 4 stages to go to the logic 0 state and the BCR P stage to the logic 1 state. The reset signal is generated (1) during a machine reset or setup sequence, since a new DAB count will be required for subsequent operation

Table 2-1. Byte Counter Stepping

Operation	BC Register					Transfer Latch					BC Latch				
	P	4	2	1	Value	P	4	2	1	Value	P	4	2	1	Value
Reset Byte Counter	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1
Change BC (during change)	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1
Not Change BC (after change)	0	0	0	1	1	0	0	1	0	2	0	0	1	0	2
Change BC (during change)	0	0	1	0	2	0	0	1	0	2	0	0	1	0	2
Not Change BC (after change)	0	0	1	0	2	1	0	1	1	3	1	0	1	1	3
Change BC (during change)	1	0	1	1	3	0	0	0	0	0	1	0	1	1	3
Not Change BC (after change)	1	0	1	1	3	0	1	0	0	4	0	1	0	0	4
Change BC (during change)	0	1	0	0	4	0	1	0	0	4	0	1	0	0	4
Not Change BC (after change)	0	1	0	0	4	1	1	0	1	5	1	1	0	1	5
Change BC (during change)	1	1	0	1	5	1	1	0	0	4	1	1	0	1	5
Not Change BC (after change)	1	1	0	1	5	1	1	1	0	6	1	1	1	0	6
Change BC (during change)	1	1	1	0	6	0	1	1	0	6	1	1	1	0	6
Not Change BC (after change)	1	1	1	0	6	0	1	1	1	7	0	1	1	1	7
Change BC (during change)	0	1	1	1	7	0	0	0	0	0	0	1	1	1	7
Not Change BC (after change)	0	1	1	1	7	1	0	0	0	0	1	0	0	0	0
Change BC (during change)	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Not Change BC (after change)	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1

of the byte counter; (2) when the last data transfer to storage has begun during a read CDA operation ('read CDA latch' signal), since the byte counter must monitor any DAB bytes that are gated into the channel before the new DAB arrives; and (3) during a write operation when the byte counter contents equal the count register 'CTB' contents ('write byte count equal count reset' signal), since this situation usually indicates that the last write operation data transfer between the I/O device and channel has occurred.

With the byte count register reset to a count of zero and the transfer latches enabled by the 'not change byte count register' signal, the outputs of byte count register stages 1, 2, and 4 are decoded by the transfer latches to enter a logic 1 in the byte count latch 1 stage; a logic 0 is entered into the remaining byte count latch stages.

For channel operations (read-type or write), the DAB is gated into the byte count register through the 'DAB ingate' logic. If the channel operation is other than a read backward operation, the DAB values contained in bit positions 21, 22, and 23 of the data address register are individually AND'ed with the 'gate DAB to adder' signal; providing the 'T4' signal from the channel clock is not active, the gated DAB bits (logic 1's) are entered into byte count register stages 1, 2, and 4, as appropriate. Byte count register odd parity is established by an exclusive-OR circuit (logic block designated "Odd" in Diagram 4-14) which receives as inputs DAB bits 21 through 23. If the DAB bits consist of an odd number of 1's, the DAB odd output of the odd logic block is activated and (assuming other than read backward operation), when gated by the 'gate DAB to adder' signal,

provides a reset signal ('DAB bits odd') to the 'BCR P' stage. If the DAB contains an even number of 1's, the reset to the 'BCR P' stage is not generated since this stage was set to a logic 1 by the reset signal preceeding the ingating of the DAB.

For a read backward operation, the DAB contained in the data address register specifies a read forward byte count that is not compatible with the combined read backward operation of the byte counter and byte count encoder. Because the byte counter is only capable of counting forward, and to obtain the necessary byte counter/byte count encoder operating compatibility, the two's complement of the DAB is gated into the BCR. In order that the DAB complement may be gated to the byte count register, the activated 'read backward' signal enables the DAB ingate AND's (Diagram 4-14) receiving the DAB complement bits and disables the DAB ingate gates receiving the actual DAB values. When the DAB complement is gated to the BCR by the 'gate DAB to adder' signal, the transfer latches decode the output of the BCR to upgrade the BCL count as previously described. The following illustration shows a typical DAB count, the counts' complement, the BCR count advance, and the B-register position gated for each BCR count.

Read Backward Operation

DAB Count	DAB Complement	BCR Advance	B-Register Byte Position Gated
100 (4)	011 (3)	011 (3)	4
		100 (4)	3
		101 (5)	2
		110 (6)	1
		111 (7)	0

To establish odd parity when gating the DAB complement into the BCR, the actual DAB bit values are examined by the exclusive-OR circuit as previously described. However, the parity bit for the DAB complement is, in all cases, the opposite value of the parity bit for the actual DAB value (See Table 2-2). For this reason, the AND receiving the not DAB odd output of the exclusive-OR circuit (Diagram 4-14) is enabled by the 'read backward' signal while the AND receiving the DAB odd output of the exclusive OR is inhibited. Thus, if the actual DAB bits contain an even number of 1's, the 'BCR P' latch is reset to a logic 0 by the 'DAB bits odd' signal; if the DAB bits contain an odd number of 1's, the 'BCR P' latch remains in the on condition to provide a logic 1 output.

The byte counter 'change' logic (Diagram 4-14) provides signals which advance the byte counter each time a data byte is gated to or from the B-register. One 'change' logic output, the 'change byte count register' signal, initiates updating of the count in the BCR and BCL 1, 2, and 4 stages. A second output, the 'change byte count parity' signal, initiates updating of the BCR and BCL parity stages when activated coincident with the 'change byte count register' signal. Both of the above 'change' logic outputs are activated during a read-type operation by the 'gate bus in to B-register' signal (derived from the channel's 'service out' signal in response to the I/O interface 'service in' signal); during a write operation, the two 'change' logic outputs are activated by the 'change byte count register and write' signal (derived from the channel's 'service out' signal in response to the control unit's 'service in' signal).

Updating of each byte counter count stage (BCR and BCL 1, 2, and 4 stages) is accomplished in a similar manner. For simplification, updating of only the BCR 2, and BCL 2 stages is described in detail in the following text. Diagram 4-15 shows the logic for the BCR 2, transfer latch 2, and BCL 2 stages; two examples of updating are shown in the figure. Consider the 'update BCR from 5 to 6' example. Prior to the count change (update) operation, the 'BCL 2' stage (at the OR output) contains a logic 1 which is to be transferred to the 'BCR 2' stage at the beginning of the change operation. The 'BCR 1' stage contains a logic 1 which will be changed to a logic 0 at the beginning of the change operation. The transfer latch output, prior to the updating operation, is at the logic 1 level. This logic 1 level is the result of AND'ing the 'not BCR 2', 'BCR 1' and 'not change byte count register' signals at AND gate 2 of transfer latch 2; the logic 1 level at the transfer latch output is OR'ed at the 'BCL 2' stage to produce the logic 1 level (previously mentioned) at the OR output.

When the 'change byte count register' signal rises, the 'BCL 2' stage is latched and the logic 1 output of the 'BCL 2' AND turns on the 'BCR 2' stage; this produces a logic 1 level at the 'BCR 2' output. (Simultaneous with the status change from logic 0 to 1 of the 'BCR 2' stage, the 'BCR 1' stage is changed from a logic 1 to a logic 0.) When the status change in the 'BCR 1' and 'BCR 2' stages is completed, AND gate 1 of transfer latch 2 is enabled by the 'not BCR

Table 2-2. DAB/DAB Complement Versus Parity Bits

DAB Values (4, 2, 1)	DAB Complement (4, 2, 1)	DAB Parity Bit	DAB Complement Parity Bit
000	111	1	0
001	110	0	1
010	101	0	1
011	100	1	0
100	011	0	1
101	010	1	0
110	001	1	0
111	000	0	1

1', 'BCR 2', and 'BC latch bit 2' (from the BCL 2 AND gate) signals. With AND gate 1 enabled, the output of the 'transfer latch 2' stage is sustained at the logic 1 level. During the duration of the 'change byte count register' signal, the outputs of the BCR 2, transfer latch 2, and BCL 2 stages are at the logic 1 level. After the fall of the 'change byte count register' signal (updating completed) all stages mentioned above are maintained at a logic 1 level output: the BCR 2 stage because it was latched during the change; the transfer latch 2 stage because AND gate 1 is enabled by the not BCR 1, BCR 2, and latch back output of the stage; and the BCL 2 stage (OR output) because the transfer latch output is at the logic 1 level.

Consider the 'update BCR from 6 to 7' example of Diagram 4-15. Prior to the updating operation, the 'BCR 2', 'not BCR 1' and 'not change byte count register' signals enable gate 1 of transfer latch 2, thus producing logic 1 levels at the transfer latch and BCL 2 OR outputs. When the 'change byte count register' signal rises, the output of the 'BCR 2' stage remains unchanged (since both the BCR 2 and BCL 2 stages contained logic 1 levels prior to the rise of the signal); however, the output of the BCR 1 stage is changed from a logic 0 to a logic 1 level. With both the 'BCR 1' and 'BCR 2' signals present to the transfer latch 2 AND's, the output of the transfer latch drops to a logic 0 level. Meanwhile, the BCL 2 stage is latched by the 'change byte count register' signal to maintain the logic 1 level at the BCL 2 OR output. When the 'change byte count register' signal falls, the BCL 2 stage AND is disabled, and the output of this stage drops to a logic 0 level. Thus, at the end of the change, the BCR 2 stage contains a logic 1 level, and the outputs of the transfer latch 2 and BCL 2 stages are at the logic 0 level.

Updating of the byte counter parity stages is similar to that previously described for the BCR 2 and BCL 2 stages. The major exception is that 'change' logic is provided to introduce even parity (a parity error) during diagnostic routines controlled by signals via the CPU interface.

For normal operation, the 'change byte count parity' signal is activated as a result of OR'ing the 'change byte count register' signal. The 'change byte count parity' signal gates the AND in the 'BCL P' stage. If the 'BCL P' stage contains a logic 1, the 'BCL P to BCR P transfer or reverse parity' output of the exclusive-OR gate (OE block in Diagram 4-14) in the absence of the 'diagnose reverse byte count parity' signal, is activated to set the 'BCR P' stage for a logic 1 output; if the 'BCL P' stage contains a logic 0, the 'exclusive OR' gate output is not activated and the AND at the reset input to the 'BCR P' stage is enabled. This permits the 'change byte count parity' signal to gate through the AND and reset the 'BCR P' stage for a logic 0 output.

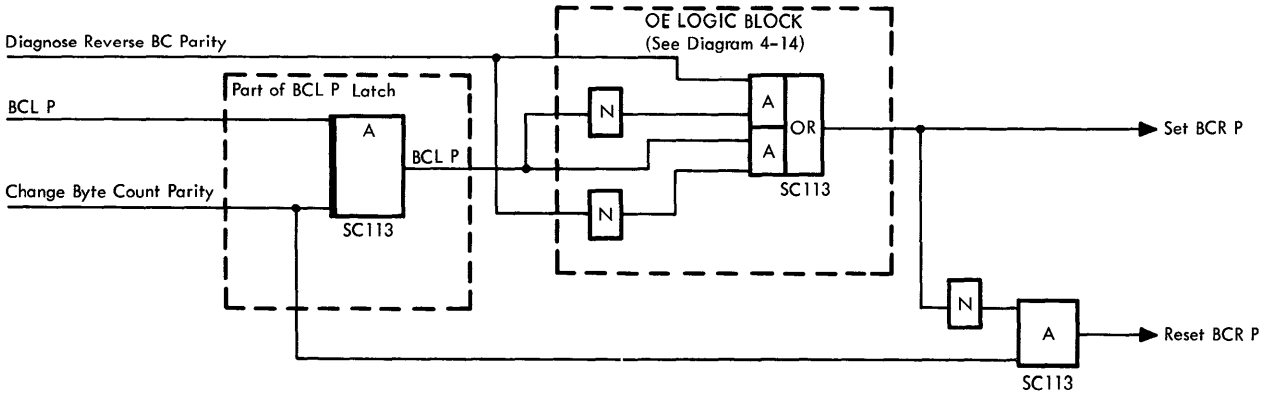
If the 'diagnose reverse byte count parity' signal is active, updating the byte counter results in just the opposite effect at the output of the exclusive-OR gate. If a logic 1 is contained in the 'BCL P' stage when the 'change byte count parity' signal is activated, the 'not BCL P to BCR P transfer or reverse parity' output of the exclusive-OR gate is AND'ed to reset the 'BCR P' stage; a logic 0 in the 'BCL P' stage activates the 'BCL P to BCR P transfer or reverse parity' signal to set a logic 1 into the 'BCR P' stage. In either case, even parity has been introduced into the byte count register. The 'byte count parity check' signal (output of ODD logic block shown on Diagram 4-14) is activated upon detection of the even parity, and a channel control check condition results.

Note: Detection of a byte count parity check condition while the 'diagnose reverse byte count parity' signal is active will cause a channel control check condition only when the counter is updated (due to B-register byte gating) and not during DAB ingating. Byte counter parity updating, both for normal and diagnostic operations, is summarized in Figure 2-18.

As previously discussed ("Byte Counter Circuits"), a read CDA operation may involve the transfer of I/O data to the channel while a new DAB (contained in the CCW) is being obtained. Thus, when the DAB is available for ingating, the byte counter already contains a significant count. If the DAB count is equal to 4 decimal (which is common due to programming limitations in the case under consideration), the DAB is gated into the byte count register; however, parity is not generated in the usual manner. (Refer to Diagram 4-14.) With DAB equal to 4 on a read CDA operation, the DAB parity generating logic (AND/AND/OR logic with DAB bits odd output) is inhibited by the output of the enabled AND receiving the activated 'gate DAB parity to byte counter' and 'read DAB 4 or read backward DAB 3' signals. To generate correct parity, the 'change byte count parity' signal is activated (shortly after the DAB is gated into the BCR) by the 'sequence 4 read CDA and read DAB 4 or read backward DAB 3' signal. With the 'change byte count parity' signal activated, the 'transfer latch parity' and 'BCL P' stages generate correct parity for the byte counter as during a normal update operation.

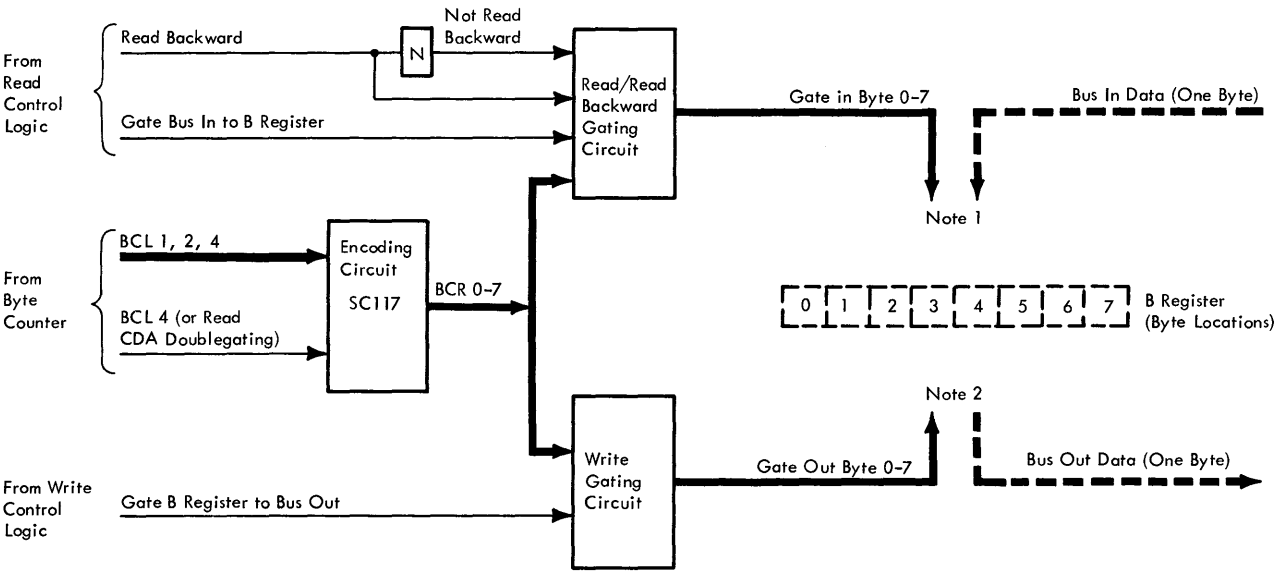
Byte Count Encoder

The byte count encoder is shown in block form in Figure 2-19 and in logic form in Diagram 4-16. The byte count encoder consists of three circuits: the encoding circuit, the read/read backward-gating circuit, and the write-gating circuit. The encoding circuit encodes the BCL 1, 2, and 4 inputs from the byte counter for write, read (or sense), read backward, or read CDA operations. Eight encoded outputs of the encoder circuit, 'BCR equals 0' through 'BCR equals 7' signals are sent to both the write-gating and read/read backward-gating circuits. For a write operation, one of the



Notes:
1. Normal Operation:
   BCL P and Not Diagnose Reverse BC Parity = Set BCR P = Odd Parity
   Not BCL P and Not Diagnose Reverse BC Parity = Reset BCR P = Odd Parity
2. Diagnostic Operation:
   BCL P and Diagnostic Reverse BC Parity = Reset BCR P = Even Parity
   Not BCL P and Diagnostic Reverse BC Parity = Set BCR P = Even Parity
   } Equals Channel Control Check on BC Change

Figure 2-18. Byte Counter Parity Updating



Notes:
1. Active 'gate in byte' line gates data byte to B register byte location; e.g., active 'gate in byte 0' signal gates data byte to B register location 0.
2. Active 'gate out byte' line gates data byte from B register byte location; e.g., active 'gate out byte 0' signal gates data byte from B register location 0.

Figure 2-19. Byte Counter Encoder Block Diagram

encoder output signals to the write-gating circuit is activated and designates one of eight 'gate out byte' signals from the write-gating circuit. The designated 'gate out byte' signal is then activated when the 'gate B-register to bus out' signal rises to gate data from the respective B-register byte location. For a read, sense, or read backward operation, one of the encoder output signals to the read/read backward-gating circuit is activated and designates one of eight 'gate in byte' signals from the 'read gating' circuit. The designated 'gate in byte' signal is then activated when the 'gate bus in to B-register' signal rises to gate data into the respective B-register byte location. For a read CDA operation (assuming that the new DAB is not available as described in "Byte Counter Circuits"), the encoding circuit activates two BCR output signals for each of the first three bytes gated into the B-register and activates one line for each byte ingated thereafter. Activation of two BCR lines simultaneously (doublegating) is controlled by the BCL 1, 2, and 4 inputs and the BCL 4 (or read CDA doublegate) input to the encoder circuits. Doublegating results in the gating of the same data byte into two different B-register locations when the 'gate bus in to B-register' signal rises. For the principles of doublegating and programming limitations resulting from the method of doublegating, see "Read CDA Encoding" in this chapter.

A more detailed description of byte counter encoder operations is presented in the following paragraphs.

Write Operation Encoding

For the following description of write operation encoding, refer to Diagram 4-16. Assume that the BCL count equals 1, that a write operation is in progress, and that data is available in the B-register for outgating. With the BCL count equal to 1, the 'BCR equals 0' output of the encoding logic AND is activated by the 'BCL 1', 'not BCL 2', and 'not BCL 4' inputs. With the 'BCL 4' (or read CDA double gate) signal inactive, all other encoding logic AND's are inhibited. In the 'write gating' logic, the activated 'BCR equals 0' signal enables one input to the 'gate out byte 0' AND. When the I/O device is ready to receive 'byte 0' data from the B-register, the channel activates the 'gate register B to bus out' signal which enables the other AND input to activate the 'gate out byte 0' signal. This signal gates byte 0 from the B-register for presentation to the I/O device (via the channel 'bus out' latches).

Subsequent to the gating of byte 0 and prior to the next B-register byte gating operation, the BCL count is advanced to a count of 2, enabling the 'BCR equals 1' signal; in turn, this enables one input to the 'gate out byte 1' AND in the 'write gating' logic.

When the 'gate register B to bus out' signal again rises, the 'gate out byte 1' signal is activated to gate the data from the byte 1 position of the B-register. This encoding and gating process continues until the data in each byte position of the B-register has been sequentially transferred to the I/O device, or until the operation is terminated for some reason. (For example, an error condition occurs, all requested data has been transferred, or a Halt I/O instruction is received by the channel.) Table 2-3 lists the BCL inputs to the 'encoding' logic, the 'BCR output' signal activated for each BCL count, and the 'gate out byte' signal activated for each active BCR signal.

Read Operation Encoding

For the following description of read operation encoding refer to Diagram 4-16.

Note: Read and sense operations are identical within the channel; thus, descriptions of read operation encoding also pertain to the sense operation.

Encoding logic AND's are enabled by the BCL 1, 2, and 4 inputs in the same manner as described for write operations. (See "Write Operation Encoding.") Therefore, the description of read operation encoding is limited to operations performed at the 'read/read backward gating' logic. For the read operation, the 'read backward' signal is inactive, enabling eight of the sixteen AND's of the 'read/read backward gating' logic. Assume that the 'BCR equals 0' signal is active and that data is available for gating to the B-register. When the 'gate bus-in to B-register' signal rises, the AND receiving the 'BCR equals 0', 'not read backward', and 'gate bus into B-register' signals is enabled to activate the 'gate in byte 0' signal. In turn, the activated 'gate in byte 0' signal gates data from the 'bus in' latches into the byte 0 position of the B-register. When gating is completed, the 'BCR equals 0' signal is deactivated and the 'BCR equals 1' signal is activated. During the time the next I/O data byte is available for gating, the 'gate bus into B-register' signal rises to activate the 'gate in byte 1' signal, thus causing the byte to be gated into byte position 1 of the B-register. The encoding and gating operation continues as described above (in succession, the 'BCR 2' through 'BCR 7' lines enable activation of the 'gate in byte 2' through 'gate in byte 7' lines, respectively), until the B-register is full or the read operation is otherwise terminated.

Table 2-3 lists the order in which the 'BCR 0 through 7' signals are activated and, as a result, the order in which the 'gate in byte' signals are activated for a read (or sense) operation. Note that, due to the order of gating, the I/O data bytes are gated into the B-register from the 'low address' byte position (byte 0) to the 'high address' byte position (byte 7).

Table 2-3. Byte Count Encoder Gating Signal Selections (Refer to Diagram 4-16, FEMDM.)

Encoding Circuit		Write Gating Circuit	Read/Read Backward Circuit	
BCL Input (4-2-1)	BCR Output (BCR 0-7)	'Gate out byte' Output Activated*	'Gate in byte' Output Activated**	
			Read	Read Backward
001	0	0	0	7
010	1	1	1	6
011	2	2	2	5
100	3	3	3	4
101	4	4	4	3
110	5	5	5	2
111	6	6	6	1
000	7	7	7	0

*Signal is activated when 'write gating' circuit is gated by 'gate B-register to bus out' signal. Activated signal gates respective B-register byte to 'bus-out' latches.

**Signal is activated when 'read/read backward' circuit is gated by 'gate bus in to B-register' signal. Activated signal gates 'bus-in' data byte to respective B-register location.

Read Backward Operation Encoding

For the following description of the read backward operation encoding, refer to Figure 4-16. Encoding logic AND's are enabled by the BCL 1, 2, and 4 inputs in the same manner as described for write operations (see "Write Operation Encoding"). For the read backward operation, the read backward input to the read/read backward-gating logic is active, enabling the eight read backward AND's. Assume that the 'BCR equals 0' signal is active and that an I/O data byte is available for gating to the B-register. When the 'gate bus in to B-register' signal rises, the AND receiving the 'BCR equals 0' and 'read backward' signals is enabled. This activates the 'gate in byte 7' signal which, in turn, gates the I/O data byte into the byte 7 position of the B-register. After the gating is completed, the 'BCR equals 0' signal is activated. When the next I/O data byte is available for gating, the 'gate in byte 6' signal is activated by the coincidence of the 'BCR equals 1', 'read backward', and 'gate bus into B-register' signals. The encoding and gating operation continues as described above (in succession, the 'BCR 2' through 'BCR 7' signals enabling activation of the 'gate in byte 5' through 'gate in byte 0' signals, respectively) until the B-register is full or the read backward operation is otherwise terminated. Table 2-3 lists the order in which the 'BCR 0 through 7' signals are activated and, as a result, the order in which the 'gate in byte' signals are activated for a read backward operation. Note that, due to the order of gating, the I/O data bytes are gated into the B-register from the high address byte position (byte 7) to the low address byte position (byte 0).

Read CDA Encoding

Read CDA encoding is unique in that doublegating (gating of the same data byte) into two different byte positions of the B-register occurs under certain conditions.

As previously stated (see "Byte Counter Circuits") a read CDA operation requires that the B-register contents be stored and a new CCW (containing a new DAB) be obtained from storage after the count specified in the old CCW is exhausted. When the last data-to-storage transfer specified by the old CCW begins, the byte count register is reset and the byte count encoder receives a BCL count of 1. In addition, the BCL 4 (or read CDA double gate) input to the encoder is activated. The byte count encoder is now at the initial state required to perform doublegating to the B-register while awaiting receipt of the new CCW.

Before the new CCW arrives, four pertinent possibilities exist regarding receipt and gating of data from the I/O device involved in the read CDA operation: (1) the I/O device

will not present data to the channel for gating to the B-register; (2) the I/O device will present between one and four data bytes for gating into B-register positions 0 through 3; (3) the I/O device will present more than four but not more than eight data bytes for gating; or (4) the I/O device will present more than eight data bytes for gating.

For all four cases, programming considerations are of importance in understanding the read CDA encoding operation of the byte count encoder. In other words, the DAB value in the new CCW has been programmed taking into consideration the probable number of I/O data bytes which will be gated into the B-register by a particular type I/O device before the new DAB value is available.

In case one above, where no I/O data bytes are gated, the DAB in the new CCW may specify B-register gating on any byte boundary. For example, if the DAB specifies a count of 5, the byte count encoder receives a BCL count of 6. In the encoder (Diagram 4-16), the BCL 6 count activates the 'BCR 5' signal, which in turn enables one input to the 'gate in byte 5' AND. When the first I/O byte is available for gating, the 'gate bus-in to B-register' signal activates the 'gate in byte 5' signal, causing the I/O data byte to be gated to the byte 5 position of the B-register. Succeeding I/O data bytes will be sequentially gated to the B-register as described in "Read Operation Encoding".

In case 2 above, (between one and four bytes gated into the B-register before the DAB arrives), the DAB arriving in the new CCW must designate a count specifying an address on single-word boundaries (DAB equals 4) or on doubleword boundaries (DAB equals 0). Otherwise, a channel chain check condition results. Figure 2-20 shows an example of byte counter encoding where doublegating of one data byte to the B-register occurs before arrival of the new DAB. As illustrated, byte 1 from the I/O device is doublegated into the byte 0 and byte 4 positions of the B-register. At the byte count encoder (Diagram 4-16), this is accomplished by enabling the BCR equals 0 and BCR equals 4 AND's with the BCL count of 1 and the activated 'BCL 4' (or read CDA doublegate) signal. When the I/O data byte is available, the 'gate bus in to B-register' signal rises, activating the 'gate in byte 0' and 'gate in byte 4' lines to gate the data byte to the byte 0 and byte 4 B-register positions. In the example of Figure 20, the byte count register is updated to 1 after the first byte is gated. Doublegating ends after the first byte is gated into the B-register due to the arrival and gating of the new DAB (DAB equals 4) to the byte counter. When the DAB is gated, it is added to the existing count of 1 already contained in the byte counter, providing a byte count register count of 5 (BCL 6 to the byte count encoder). Therefore, when the next I/O data byte is available for gating, the byte count encoder raises the 'gate in byte 5' signal (Diagram 4-16) to gate the data into the byte 5 position of the B-register. In this case, data entered in the byte 0 position of the B-register is reset. Singlegating continues until the B-register is "full"; data has been gated into byte positions four through seven. Had the DAB contained a doubleword boundary count of 0 in the example of Figure 2-20, gating operations to the B-register would remain valid (a chain check condition would not result), since the second data byte would be stored in the byte 1 position of the B-register, and the byte 4 position would be reset. To understand how a channel chain check condition would result if the new DAB count were other than on singleword or doubleword boundaries (Figure 2-20 example), assume that the new DAB contained a count of 5. In this case, adding the existing 1 count in the byte count register with the DAB count of 5 yields 6 (BCL count of 7 to the encoder). When

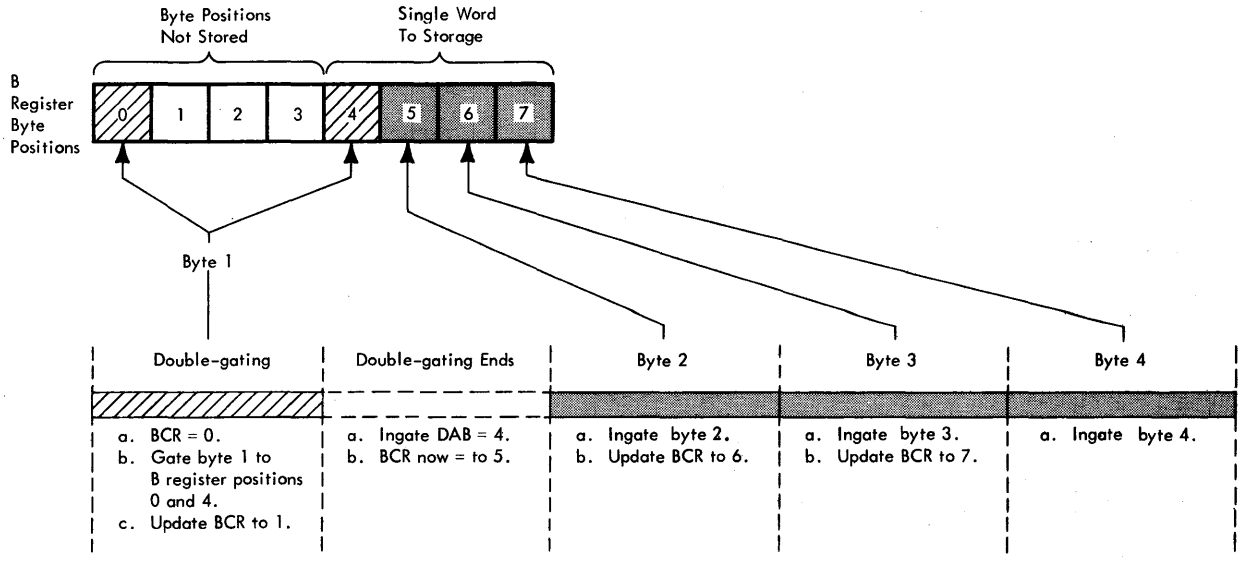


Figure 2-20. Example of Read CDA Doublegating Requiring DAB on Singleword Boundaries

the next I/O data byte arrived, the byte count encoder would gate it into position 6 of the B-register. Thus, two consecutive I/O data bytes would be contained in the byte 4 and byte 6 positions, with the byte 5 position unoccupied with data. Since I/O data must be gated into sequential byte positions of the B-register to be valid, a channel chain check condition would result in the above described circumstances.

In case 3, (more than four but less than nine data bytes gated into the channel before the new DAB arrives), the DAB in the new CCW must designate a count specifying an address on doubleword boundaries (DAB equals 0). Otherwise, a channel chain check condition results. Figure 2-21 illustrates an example where seven I/O data bytes are gated into the B-register before the new DAB is gated into the B-register. As shown, the first three I/O bytes are doublegated into the bytes 0 and 4, bytes 1 and 5, and bytes 2 and 6 positions of the B-register. Doublegating is accomplished from the byte count encoder (Diagram 4-16) by activating two 'gate in byte' lines for each of the three bytes in the manner previously described [BCL count and active 'BCL 4' (or read CDA doublegate) signal activating the appropriate 'BCR equals N' signals which are further gated by the 'gate bus-in to B register' signal]. When the fourth I/O data byte is gated into position 3, doublegating is ended, and positions 4 through 7 of the B-register are reset so that succeeding I/O data bytes may be gated into these positions. In the example (Figure 2-21), succeeding I/O data bytes may be gated into the byte 4 through byte 6 positions of the B-register before the new DAB is gated into the byte count register. Since, when the DAB is gated, the byte count register already contains a count of 7 (BCL=0 to the byte count encoder), the DAB must equal 0; otherwise, when added to the existing byte count register count, the byte count encoder would specify gating to other than the byte 7 position of the B-register. Since all other B-register byte positions contain valid I/O data and since I/O data bytes must be

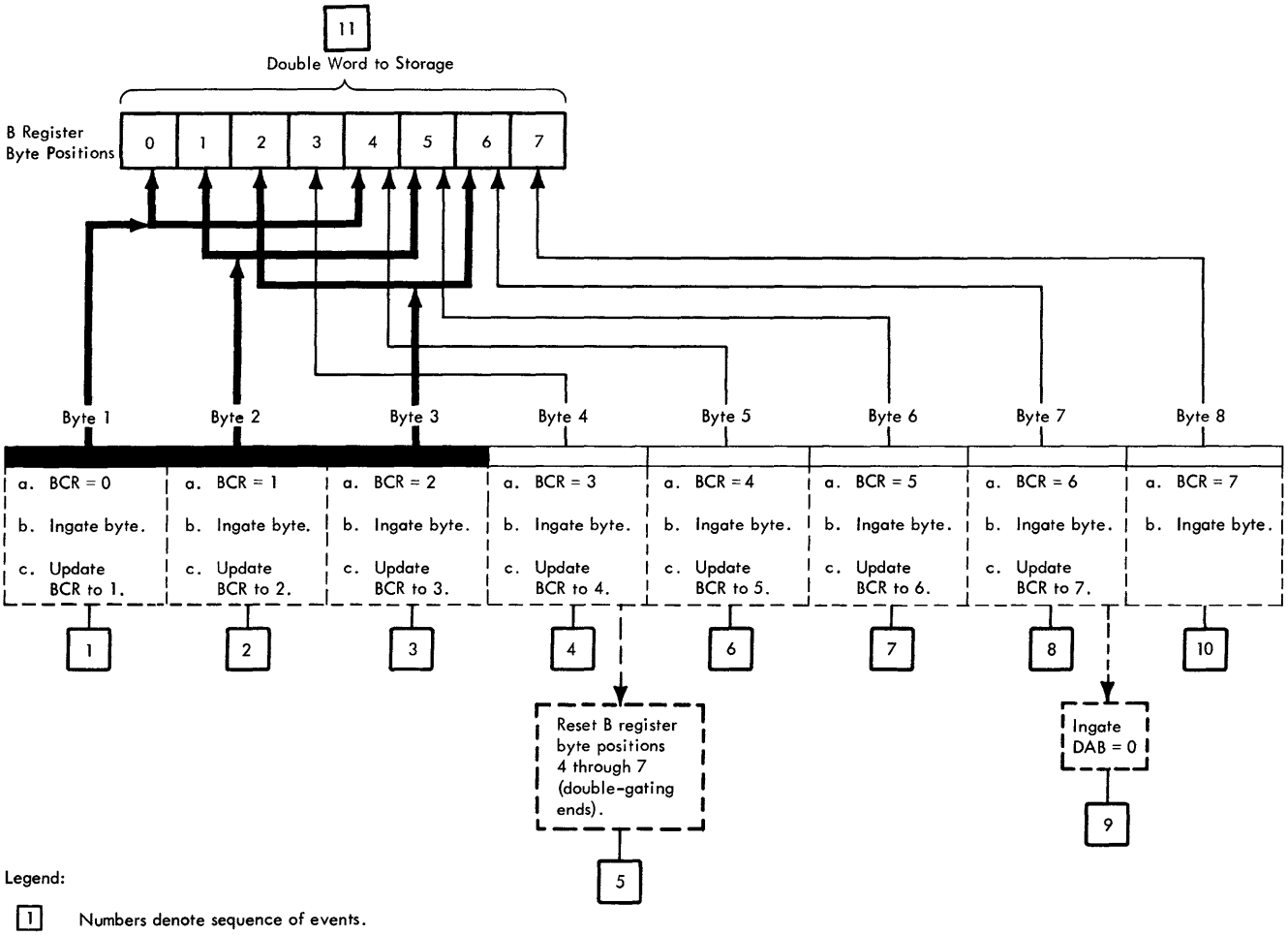


Figure 2-21. Example of Read CDA Doublegating Requiring DAB on Doubleword Boundaries

gated into consecutive B-register byte positions, a DAB that does not specify a doubleword boundary results in a channel chain check condition.

In case 4, more than eight I/O data bytes are presented for B-register gating before the new DAB arrives. When this occurs, the channel transfers the B-register contents into the A-register and continues singlegating data bytes into the B-register. If the B-register is filled before the new DAB arrives, an overrun condition is detected to turn on the 'chain check' latch. If the B-register is not filled (A-register full), the DAB must equal 0 so that the byte counter count is not altered and B-register consecutive gating can continue; otherwise, a chain check condition is detected.

Byte Count Compare Logic

The byte count compare logic (Diagram 4-17) monitors the progress of data transfers through the B-register; output signals from this logic are used by the channel control logic (depending upon the operation in progress) to initiate data transfers, ending sequences, determine error conditions, and initiate reset and other control operations. For all but a write CDA operation, the 'byte count compare' logic continuously compares the three low-order bits (CTB bits 21, 22, and 23) of the count register with the BCL outputs of the byte counter circuits. The comparison is performed at three exclusive-OR gates (Diagram 4-17). When the CTB count and the BCL count are equal, the OR gate output fed by the exclusive-OR gates is deactivated, providing an active 'byte count equals CTB' signal to one input of an AND. If the 'last word' trigger is on (indicating the last data transfer of the current CCW is forthcoming) and the current operation is not a write CDA operation, the 'gate byte count equals 0 latch' signal strobes the AND to turn on the 'byte count equals CTB' trigger. For both read-type and write-type operations, the 'byte count equals CTB' signal is generated when data bytes are gated to or from (respectively) the B-register.

With the 'byte count equals CTB' trigger on, the output of this trigger: (1) sets the 'byte count equals CTB' latch; (2) is sent to the status logic (ALD CK121) where it may be instrumental in initiating turn-on of the 'sequence 5' latch; (3) is sent to the write controls logic (CW111) where, for a write operation is generates a 100 ns 'write byte counter equal counter reset' signal; (4) is sent to the 'write CDA controls' logic (ALD CW113) where it is used to initiate a chain data operation if the CDA flag is active; (5) is sent to the 'sequence 5 gating' logic (ALD EN111) to initiate gating signals for the ending sequence; and (6) is sent to the 'simulate interface controls' logic (ALD RS115) where, under specified conditions it is used to generate a 'simulate read gate' signal. The 'byte count equals CTB' trigger output is also OR'ed (Diagram 4-17) to activate the 'BC equals 0 or BC equals CTB' signal. The presence of this signal is one of the requirements to turn on the 'B-register full' latch during a read operation or to initiate a sequence 3 operation during a write operation.

The 'byte count equals CTB' latch signal initiates a sequence 5 operation (ending sequence) if data chaining is not indicated. (See Diagram 4-17.) The latch output is also used in the following channel control circuits: (1) manual controls circuits (ALD CW113) for the simulate interface operations; (2) 'channel status' logic (ALD CK111) to determine if an incorrect length condition exists; (3) 'read operation update count' logic (ALD CR115) to inhibit further decrementing of the count register; and (4) 'sequence 5 gating' logic to inhibit incorrect length operations.

If the current operation is a write CDA operation and a new CCW is being fetched from storage, the 'byte count to CTB' comparison is made between the count from the byte counter (BCL 1, 2, and 4) as previously described, and the mark-B register (bits 0 through 2). In this case, the current count (from the old CCW) has not been exhausted and data bytes are still being transferred from the channel to the I/O device. So that the count in the incoming (new) CCW may be gated into the count register without interfering with the data transfers specified by the old CCW, the old count is transferred to the mark-B register and, in turn, supplied to the byte count comparator for comparison purposes. When the mark bits and BCL bits compare, and the AND conditioned by the 'write and CDA' signal (Diagram 4-17) is gated by the 'gate byte count equals 0 latch' signal, the 'byte count equals CTB' trigger is turned on. Consequently, the 'byte count equals CTB' latch is turned on and the 'byte count equals 0 or byte count equals CTB' signal is activated as previously described.

The 'byte count equals 0' latch (Diagram 4-17) is turned on when the byte counter (Diagram 4-14) detects a count of 0 in the byte count latches. The activated output of the 'byte count equals 0' latch: (1) activates the 'byte count equals 0 or byte count equals CTB' signal previously described; (2) is sent to the 'status' logic (ALD CK121) where it may be instrumental in initiating turn on of the 'sequence 5' latch; (3) is sent to the 'write controls' logic (ALD CW111) where, during a write operation, it initiates turn-off of the 'B-register full' trigger; (4) is sent to the 'read CDA controls' logic (ALD CR121), where it is used to determine if an overrun condition exists; and (5) is sent to the 'SBI gate bits 16 through 19' logic (ALD MB161), where it is gated to the SBI bit 18 position on log word 3 during a log-out operation.

One other output, the 'byte count greater than CTB' signal, is supplied by the 'byte count compare' circuit. This signal is significant only during a read CDA operation and, if active, will cause a channel chain check condition if the count register contains a new count of less than one word (less than 8 bytes). The status of the 'byte count greater than CTB' signal is determined by comparing bits 21 through 23 from the count register with BCL bits 1, 2, and 4 from

the byte counter. Note 1 in Diagram 4-17 indicates the bit values which will activate the 'byte count greater than CTB' signal.

B-REGISTER

- Has an eight-byte (doubleword) capacity.
- On read-type opertion, data is gated in from 'bus in' latches and gated out the A-register.
- On write-type operation, data is gated in from A-register and gated out to 'bus out' latches.
- Data is gated into B-register from A-register in doublewords; data is gated out of B-register into A-register in doublewords.
- Data is gated to the B-register from the 'bus in' latches and gated to the 'bus out' latches from the B-register in bytes.
- B-register has no parity checking circuits.

The B-register (Diagram 4-18) consists of 72 latches (64 data and 8 parity) and associated control circuits. The latches are arranged in eight bytes, designated from high-order to low-order positions as bytes 0 through 7. The 64 'data bit' latches are on to indicate active bits, and the 8 parity latches are off (not latched) to indicate active parity bits.

On a read-type operation, the B-register functions as an assembler, assembling data bytes from an I/O device into doublewords to be transferred to the A-register. On a write-type operation, the B-register functions as a disassembler, gating the doublewords received from the A-register to the I/O device a byte at a time. B-REGISTER indicators are provided on the CE panel for bytes 0 through 7 (including parity).

For a read-type operation (Diagram 4-18) bytes from the I/O interface are gated into the B-register by active 'gate in byte 0' through 'gate in byte 7' signals. These signals are activated by the byte count encoder in the byte counter circuits at appropriate times to gate data bytes from the 'bus in' latches to specific B-register byte locations.

For write-type operations (Diagram 4-18) bytes from the B-register are gated to the I/O interface by active 'gate out byte 0' through 'gate in byte 7' signals. These signals are activated by the byte count encoder in the byte counter circuits at appropriate times to gate data bytes from the 'bus in' latches to specific B-register byte locations.

For write-type operations (Diagram 4-18) bytes from the B-register are gated to the I/O interface by active 'gate out byte 0' through 'gate out byte 7' signals. These signals are activated by the byte count encoder in the byte counter circuits at appropriate times to gate data bytes from the 'bus-in' latches to the I/O interface.

For A-register to B-register transfers (write-type operation), the B-register is first reset by the 'reset B-register' signal (Diagram 4-18). Subsequently, the 'gate register A to B' latch is turned on, and the active 'gate register A to register B' signal gates A-register bits 0 through 63 (plus eight parity bits) into the B-register.

When, during a read-type operation, data is gated from the B-register to the A-register, the 'B to A' latch is turned on, activating the 'gate B-register to A-register' signal (Diagram 4-18). This signal gates B-register bits 0 through 63 (plus eight parity bits) into the A-register.

A-REGISTER

- Has an eight byte (one doubleword) capacity.
- SBO bits gated into A-register and gated out to the B-register on a write-type operation.
- B-register bits gated into the A-register and out to the SBI lines on a read-type operation.
- Data doublewords (only) gated into or out of A-register.
- Parity is checked when A-register bits are gated to the SBI lines during a store operation.

The A-register consists of 72 latches (64 data, 8 parity) and associated control circuitry. The latches are arranged in eight bytes designated bytes 0 through 7 (from the high-order to low-order positions, respectively). The 64 data bit latches are on to indicate active bits, and the eight parity latches (P0 through P7) are off (not latched) to indicate active parity bits. Once loaded, the contents of the A-register remain stable until new data is gated into the register or a 'machine reset' signal occurs. The A-register acts as a buffer between the B-register and the SBO and SBI storage lines.

A-REGISTER indicators for bytes 0 through 7 (including parity) are provided on the CE panel.

A-register operation is different for a read-type, write-type, or test mode manual operation. For these operations, the A-register reset and ingating signals are active simultaneously, but because of logic delays the reset signal falls before the ingating signal falls (ALD RW111). A-register read-type, write-type and manual operations are described separately in the following paragraphs. Descriptions are based upon Diagram 4-19.

A-Register Read-Type Operation

- 'B to A' latch turned on when B-register is full; 'gate B-register to A-register' signal gates B-register contents to A-register.
- On storage of A-register contents, 'gate A to SBI' signal gates A-register bits 0-63 to SBI lines.
- SBI logic generates parity bits from gate A-register 0-63 bits; generated parity gated to SBI lines.
- A-register parity bits and generated parity bits compared for channel data check condition.

During read-type operations, a data doubleword is gated into the A-register from the B-register each time a B-register boundary is reached (B-register is full as indicated by a BCL = 0 condition).

When the BCL = 0 condition is detected, the 'read and BC equal 0 and not read CDA A-register loaded' signal (Diagram 4-19) turns on the 'B to A' latch to gate B-register bits 0 through 63 (plus eight parity bits) into the A-register. For the first doubleword of a read operation, all byte positions may not be stored, depending upon the DAB value received in the read operation CCW; i.e., data bytes may be gated into all or only some B-register byte positions. The data bytes to be stored are identified by mark bits gated into the mark-B register when the data bytes are gated into the B-register. When the B-register contents are gated to the A-register, the mark bits are gated from the mark-B register to the mark-A register and identify the A-register byte to be stored. (Mark bits are gated to the BCU interface mark lines prior to gating of the A-register contents to the BCU interface SBI lines).

With the data doubleword in the A-register, the channel requests a storage cycle; after the storage request is honored and the channel receives (or in the case of the Model 91, generates) the 'BCU data request' signal, the 'gate A to SBI' signal (Diagram 4-19) activates the 'gate A-register to SBI' signal. This signal gates A-register bits 0 through 64 to SBI lines 0 through 64. The A-register 0 through 64 bits are also applied to the SBI 'parity generator' logic where parity bits are generated for the eight A-register bytes. The generated parity bits (P0 through P7) are gated to the SBI lines by the 'gate SBI parity' signal (delayed from the 'gate A to SBI' signal). Note that when the channel is under diagnostic control and the 'diagnose block storage data check' signal is active, the A-register parity bits are gated to the SBI lines, rather than the generated parity bits (Diagram 4-19). This permits the channel to store bad parity data bytes from the A-register during diagnostic operations.

For normal operation, the A-register parity is checked by the SBI parity checking logic. This logic compares the A-register parity bits with the generated parity bits at exclusive-OR logic (Diagram 4-19). If the two sets of parity bits do not match (on a bit-by-bit basis) the 'SBI parity check' signal is activated and, when sampled, turns on the 'channel data check' latch.

If an incorrect length condition is detected during the read operation, the 'read WLR short' signal turns on the 'B to A' latch, causing the B-register contents to be gated to the A-register (Diagram 4-19). The channel then requests a storage cycle to store the A-register data and terminates the operation.

For a read operation normal ending, data byte transfers can end at any byte location within the last doubleword; i.e., the last doubleword can contain from one to eight bytes for storage. As for the first doubleword, mark bits specify which bytes within the doubleword are to be stored. With the possible exception of the first and last doubleword transfers for the read operation, all bytes in the A-register are stored in main storage during a store operation.

A-Register Write-Type Operation

- For write-type operation, the A-register receives bits from the SBO lines.
- 'Gate SBO to A full write' signal activates 'register A ingate or machine reset' signal; resets the A-register and turns on the 'SBO to A' latch.
- SBO bits gated to A-register.
- Last byte gated from B-register; 'turn off A full write' signal activated.
- B-register reset; 'gate register A to B' latch is turned on.
- A-register contents gated into B-register.

During write-type operations, a data doubleword is gated into the A-register from the SBO lines. When the A-register is full, the contents of the A-register are then gated to the B-register and another storage request is initiated by the channel to fill the A-register.

After the channel requests a storage cycle and the requested data doubleword is on the SBO lines, the 'gate SBO to A full write' signal (activated by the 'raw advance SS' signal) activates the 'register A ingate or machine reset' signal and turns on the 'SBO to A' latch. The 'register A ingate or machine reset' signal resets the A-register and the 'gate SBO to A-register' signal gates SBO bits 0 through 63 and P0 through P7 to the A-register (Diagram 4-19).

When all bytes have been gated from the B-register to the I/O interface, the 'turn off A full write' signal initiates a reset to the B-register and turns on the 'gate register A to B' latch. The 'gate A to B' output of the latch gates A-register bits 0 through 63 and P0 through P7 to the B-register. Each

time the A-register receives a data doubleword from the SBO lines and all bytes in the B-register are gated to the I/O interface, the described A-register operations are performed.

A-Register Manual Operations

- For manual store operation, pressing STORE pushbutton gates CE panel SBO switch values into A-register.
  1. If SIMUL STOR switch off, storage is accessed to store A-register contents.
  2. If SIMUL STOR switch on, A-register contents not stored.

- For manual fetch operation, pressing FETCH pushbutton causes data doubleword to be gated into A-register.
  1. If SIMUL STOR switch off, data is fetched from storage.
  2. If SIMUL STOR switch on, data is fetched from CE panel SBO switches.

When the channel is in the test mode, the A-register receives data doublewords from the SBO switches on the CE panel, from main storage via the SBO lines, or from the B-register. Data doublewords are gated from the B-register into the A-register as described for normal read-type operations.

For a test mode manual store operation, pressing the STORE pushbutton turns on the 'storage request' latch; in turn the 'storage request' signal activates the 'gate SBO to A-register' signal which turns on the 'SBO to A' latch. The 'gate SBO to A-register' signal from the latch gates the values in the CE panel SBO switches into the A-register. If the SIMUL STOR (simulate storage) switch is off, the 'storage request' signal is sent to the BCU interface and the 'manual store' signal gates the A-register contents to the SBI lines for storage in main storage. If the SIMUL I/O switch is on, the 'storage request' signal is not sent to the BCU interface. Pressing the STORE pushbutton activates the 'manual store' signal; however, A-register bits are inhibited at the SBI gating logic and are not actually stored.

For a manual fetch operation, the channel initiates a storage cycle when the CE panel FETCH pushbutton is pressed. If the SIMUL STOR switch is off, the channel accesses main storage. When the data doubleword is on the SBO lines, the 'gate SBO to A-register' signal is activated (by the 'raw advance SS' signal) to turn on the 'SBO to A' latch. The resulting 'gate SBO to A-register' signal gates SBO bits 0 through 63 to the A-register. If the SIMUL STOR switch is on, the channel does not access storage but internally simulates signals normally sent to and received from the BCU interface. When the 'BCU advance pulse' signal is simulated, the 'raw advance SS' signal activates the 'gate SBO to A-register' signal; this causes the values set into the CE panel switches to be gated to the A-register. For more detailed information on A-register test mode operation refer to Section 1, Chapter 6 of this manual.

CHANNEL ADDRESS LOGIC

- Channel address is represented by three binary bits.

- Channel address is gated to storage upon an IPL end condition or a logout operation.

- Channel address is displayed on CE panel.

- Channel address is wired into channel during installation.

The channel address logic (Figure 2-22) provides a three-bit binary code which identifies the channel within the system. The channel address is used for three purposes: (1) to provide a visual display on three CHAN ADR (channel address) indicators on the channel's CE panel; (2) on an IPL operation to provide the channel address to storage so that the CPU has a means of identifying the channel that has performed the IPL operation; and (3) on a logout operation to provide the channel address to storage so that the channel involved in the logout operation may be identified if the log information is subsequently recorded for visual observation.

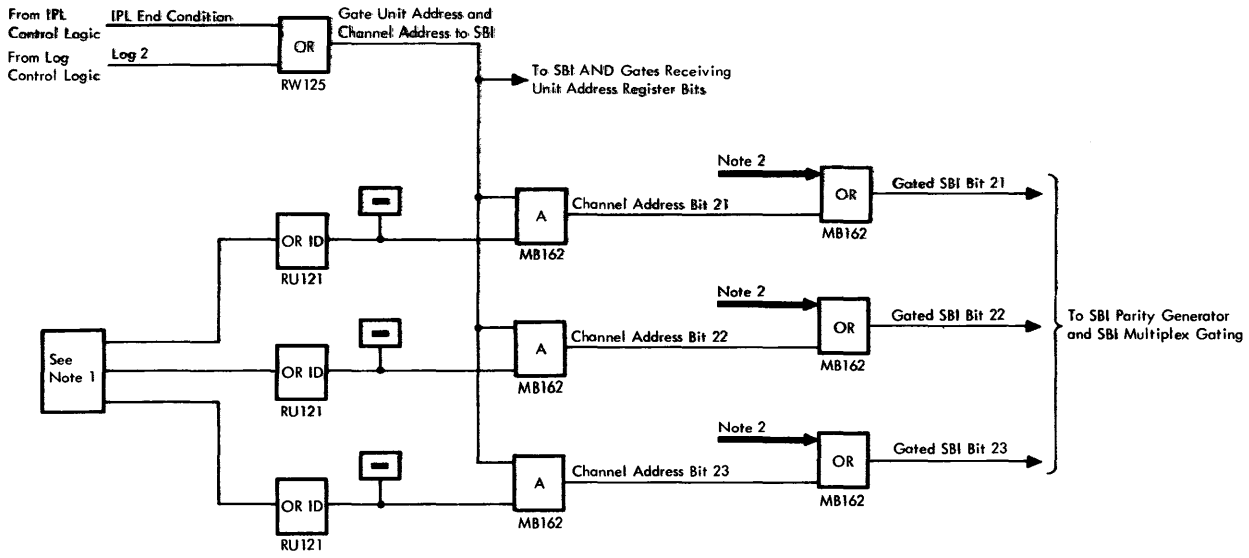
On an IPL operation, the channel address is gated to storage, (by the 'IPL end' signal) at the end of the IPL operation for storage in byte 3 of address 0. On the channel logout operations, the channel address is gated to storage (by the 'log 2' signal) during the transfer of logword 2. For both operations, the three channel address bits are gated to the 'SBI bit 21' through 'SBI bit 23' lines.

An assigned channel address is wired into three drivers in the channel at the time of installation. This is accomplished by grounding the input of any driver which is to produce a logic 1 at its output.

UNIT ADDRESS BUS-OUT LOGIC

- 'Unit address bus out' logic:
  1. Gates unit address bus-out bytes from CPU interface when channel is in auto mode.
  2. Gates values in CE panel UNIT ADDRESS switches through logic when the channel is in the test mode.

- Gated bits specify the unit address of the I/O device to be used for a CPU instruction, IPL operation, or test mode operation.



Notes:
1. The channel address is wired when the 2860 is installed. Grounding a driver input produces a logical 1 at the driver output.

2. Other gated inputs to SBI 21-23 OR gates are:
   a. A register bits 21-23.
   b. Command address register bits 13-15.
   c. Byte count register bits 4, 2, and 1.

Figure 2-22. Channel Address Logic

- 'Unit address bus out' bits gated into unit address register.

- Values in CE panel UNIT ADDRESS switches are gated through logic and into units address register when channel is in the test mode.

The 'unit address bus out' logic (Figure 2-23) consists of nine sets of AND/AND/OR combinations. Each combination gates either a 'unit address bus out' value (from the CPU interface) or a CE panel 'unit address switch' value into the channel.

When the CE panel AUTO/TEST switch is set to AUTO and the 'not simulate CPU' level is present, nine AND's receiving 'unit address bus out' bits (UABO bits 0 through 7 and P) are enabled. When the CPU initiates a Start I/O, Test I/O, or Halt I/O (channel free) instruction, an IPL operation or an FLT operation, the CPU places a unit address byte (eight bits plus parity) on the 'unit address bus out' lines. This byte is gated through the channels 'unit address bus out' logic and is available for comparison with the unit address register contents or for gating into the unit address registers (depending upon the operation in progress).

With the CE panel AUTO/TEST switch to TEST and the 'simulate CPU' signal active, the 'unit address bus out' logic AND's receiving the 'unit address bus out' lines are disabled and the AND's receiving the CE panel 'unit address switch' values are enabled. The values in the CE panel UNIT ADDRESS switches are gated through the 'unit address bus out' logic and are available for gating into the unit address register or comparison with the unit address register contents (depending upon the test operations performed.)

SBI GATING LOGIC

- CSW gated to SBI lines for store CSW operation; SBI gating logic generates parity bits for CSW bytes.

- For logout operation:
  1. 'Log word 1' information (CSW) gated to SBI lines; SBI gating logic generates parity bits for log word 1 bytes.
  2. After log word 1 gating, log word 2 information gated to SBI lines; SBI gating logic generates parity bits for log word 2 bytes.
  3. Log word 3 information gated to SBI lines; SBI gating logic generates parity bits for log word 3 bytes.

- For IPL operation, unit and channel addresses gated to SBI lines at end of IPL operation. SBI gating logic generates parity bits for log word 2 bytes.

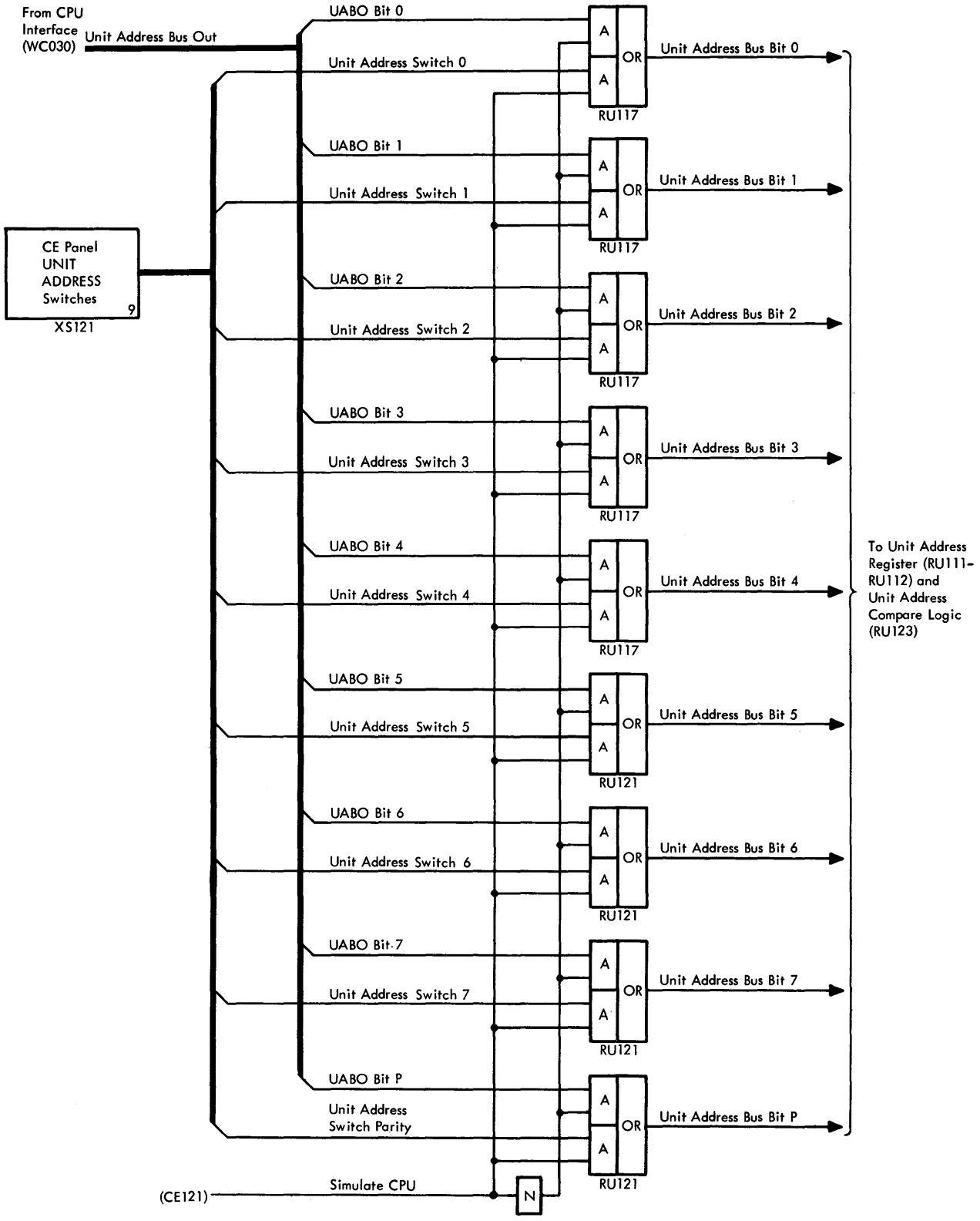


Figure 2-23. Unit Address Bus-Out Logic

- For read-type operations:
  1. A-register data bits gated to SBI lines.
  2. SBI gating logic generates parity bits for A-register bytes.
  3. Generated parity bits and A-register parity bits compared for channel data check condition.

The SBI gating logic (Figure 2-24) provides the channel with a means of gating information from the channel to the SBI lines for storage in main storage. Information gated to the SBI logic is as follows: (1) CSW information stored during a channel CSW store operation; (2) 'log word 1', 'log word 2' and 'log word 3' information stored during a logout operation; (3) A-register data stored during a read-type storage operation; and (4) channel and unit address information stored during an IPL end storage operation. For a CSW store operation, gating signals gate bytes of the CSW through ANDs; the outputs of the AND's are OR'ed to the SBI lines, and to SBI parity generating logic. The parity generating logic generates eight parity bits, one for each byte of the CSW, and presents these bits to the SBI parity lines. The parity generating logic is necessary, since the CSW is composed of bits from various channel sources. Diagram 4-20 shows the information comprising the CSW. Note that the CSW is composed of the storage protect key bits from the storage protect register, the command address register data bits, the control unit status byte ('bus in' latches, bits 0 through 7), the channel status byte (composed of the outputs of individual latches in the channel), and the count register bits. Since parity for all CSW bytes has not been established in the channel (e.g., the channel status byte), the SBI parity generating logic provides a means of generating correct parity. For the CSW store operation, all bytes of the CSW are gated to the SBI ('gate command address register to SBI', 'gate control unit status to SBI', 'gate channel status to SBI', and 'gate count register to SBI' signals activated) except for a CSW store operation when the 'start I/O' latch is on or a 'halt I/O' signal is received when the channel is free. (When the 'start I/O' latch is on during a CSW store operation,the channel has detected an error before operations specified by the Start I/O command begin; when a 'halt I/O' signal is received with the channel free, the channel is not engaged in an operation.) In these two cases only the control unit and channel status bytes are gated to the SBI. The 'gate channel status to SBI' signal gates the 'channel status latches' values and the 'gate control unit status to SBI' gates the 'bus in' latches 0 through 7 bits.

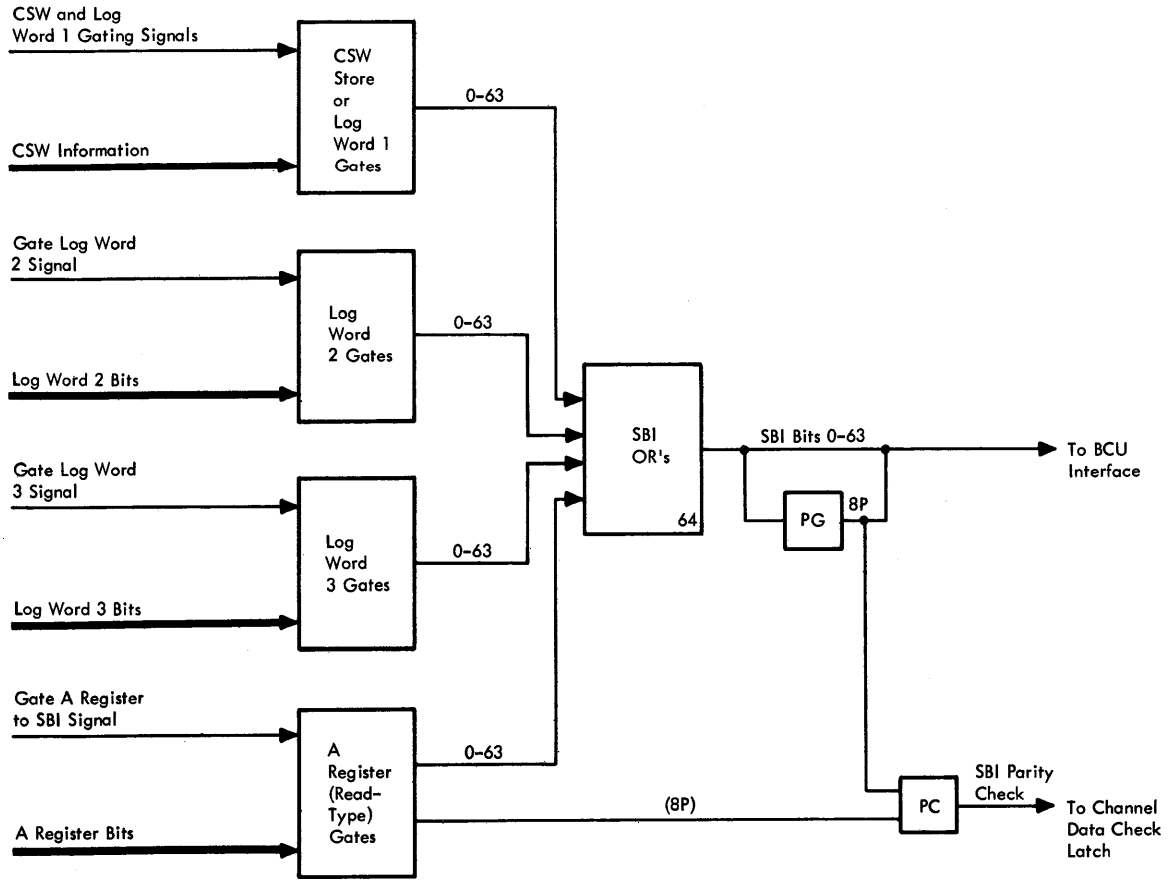


Figure 2-24. SBI Gating Logic, Block Diagram

The SBI gating logic provides gating AND's for the three log words sequentially gated to storage during a logout operation. Diagram 4-20 shows the information gated to the SBI lines for each of the three log words. For log word 1, the entire CSW is gated to the SBI lines for storage at main storage location 304, decimal. For log word 2, the 'gate log word 2' and 'gate unit address and channel address to SBI' signal are activated to gate the 'log word 2' information to the SBI lines. 'Log word 2' information consists of the contents of channel registers and bits identifying the channel address and is stored in main storage location 312 (See Diagram 4-20.) SBI parity generating logic generates parity bits for the bytes comprising log word 2. For log word 3, the 'gate log word 3' signal is activated to gate the 'log word 3' information to the SBI lines. Log word 3 data (stored in main storage location 320, decimal) consists of the outputs of various channel latches, various channel signal lines, and the parity bits from channel registers. (See Diagram 4-20.) The SBI parity generator generates parity bits for the bytes comprising log word 3.

For an IPL operation, the channel address bits (wired into the channel) and the contents of the unit address register are gated to the SBI lines at the end of the IPL operation. At the SBI gating logic, the channel and unit address bytes are gated to the SBI lines when the 'gate unit address and channel address to SBI' signal is activated (Diagram 4-20). SBI parity generating logic generates the required parity bits for the channel and unit address bits.

For read-type operations, the A-register contents (bits 0 through 63) are gated through SBI AND's to the SBI lines when the 'gate A-register to SBI' signal is activated (Diagram 4-20). SBI parity generating logic generates parity bits for the gated A-register bytes. In addition to supplying the generated parity bits to the SBI parity line, the parity generating logic also sends the generated parity bits to 'SBI parity checking' logic. The 'SBI parity checking' logic also receives the eight parity bits from the A-register and 'gate A-register to SBI' signal. At the 'SBI parity checking' logic, the eight generated parity bits are compared with the eight parity bits from the A-register. If the bits do not match while the 'gate A-register to SBI' signal is active, the 'SBI parity check' signal is activated to turn on the 'channel data check' latch. For more detailed information on gating A-register data to the SBI lines, refer to "A-Register" in this Chapter.

BUS-OUT LATCHES

- Consists of nine positions; each position is polarity hold.
- 'Bus out' latches enable channel to gate data bytes, unit address bytes, and command bytes to control unit via I/O interface.
- B-register bytes gated to 'bus out' latches (write operation) by gating signals from byte count encoder; data bits latched into polarity holds.
- Unit address bytes gated through 'bus out' latches by 'setup' signal.
- Command bytes gated through 'bus out' latches by 'gate command out to bus in' signal.
- 'Gate zeros to bus out' signal forces Stop command of all zeros with good parity into 'bus out' latches.

The 'bus out' latches (Figure 2-25) enable the channel to gate data bytes, address bytes, and command bytes to the control unit via the I/O interface 'bus out' lines. The 'bus out' latches consist of nine positions, with each position functioning as a polarity hold. Individual gating logic is associated with each position; 'latch bus out' (control and reset) signal and the 'gate zeros to bus out' signal are associated with all positions.

B-register data bytes are gated through the 'bus out' latches (Figure 2-25) during write-type operations. 'B-register gates' logic associated with the 'bus out' latches receives bytes 0 through 7 from the B-register; for each byte, a corresponding gating line from the byte count encoder is supplied to the 'B-register gates' logic.

An active gating line gates the corresponding B-register byte to the 'bus out' latches. As each byte is gated into the 'bus out' latches, the 'latch bus out' signals are activated to latch the byte values into the 'bus out' latches (polarity holds).

During initial selection or chain command setup operations the unit address byte in the unit address register latches are gated through the 'bus out' latches (via the unit address register gates by the 'setup' signal). The 'latch bus out' signals are not activated, and the unit address bits gated into the 'bus out' latches appear at the 'bus out' latches outputs without being latched into the polarity holds.

Following selection of an I/O device during initial setup or chain command setup operations, the channel gates the command byte from the command address register (via the 'command register' gates) by activating the 'gate command out to bus out' signal. The 'latch bus out' signals are not activated, and the command bits gated into the 'bus out' latches appear at the 'bus out' latches outputs without being latched into the polarity holds.

For channel operations requiring that a stop command byte be sent to the I/O device, the channel activates the 'gate zeros to bus out' signal (Figure 2-26). This signal causes all zeros with good parity to appear at the 'bus out' latch outputs. Operations which cause the 'gate zeros to bus out' signal to be activated are: (1) a halt I/O channel free or test I/O condition when the 'setup' and 'sequence 2' latches are off, the channel is not engaged in simulate I/O

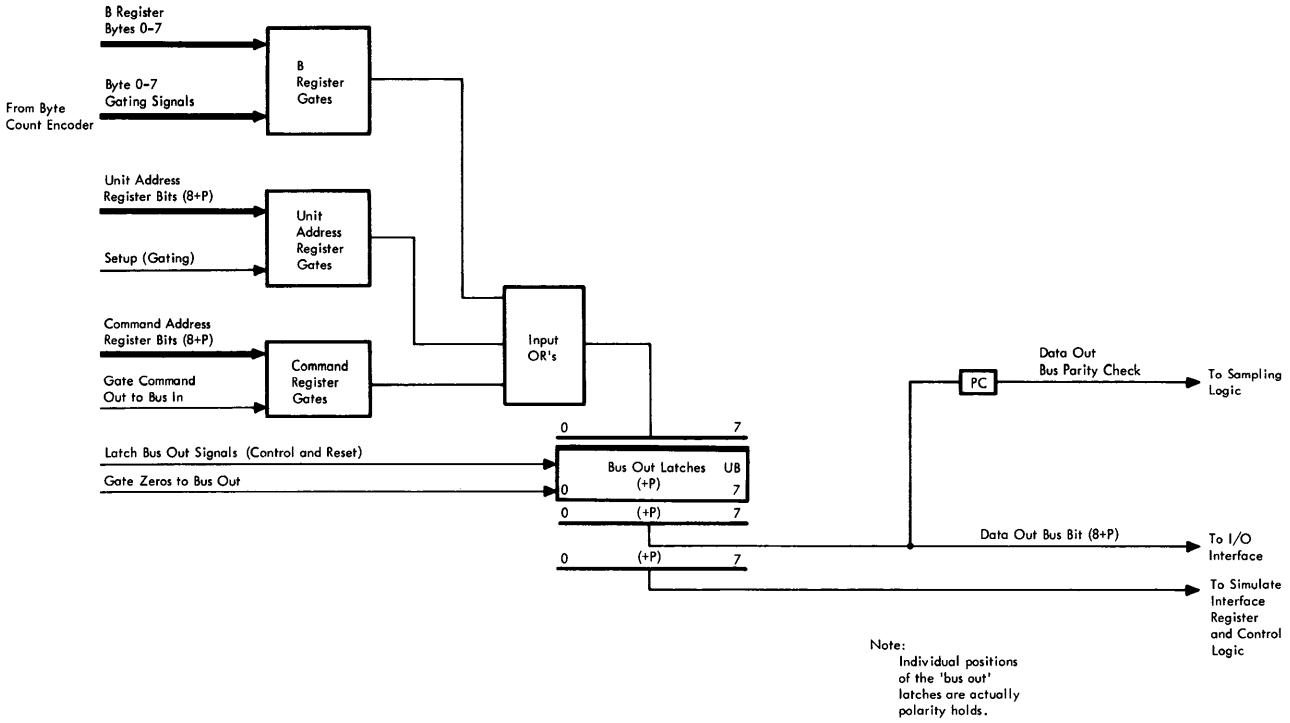


Figure 2-25. Bus-Out Latches, Block Diagram

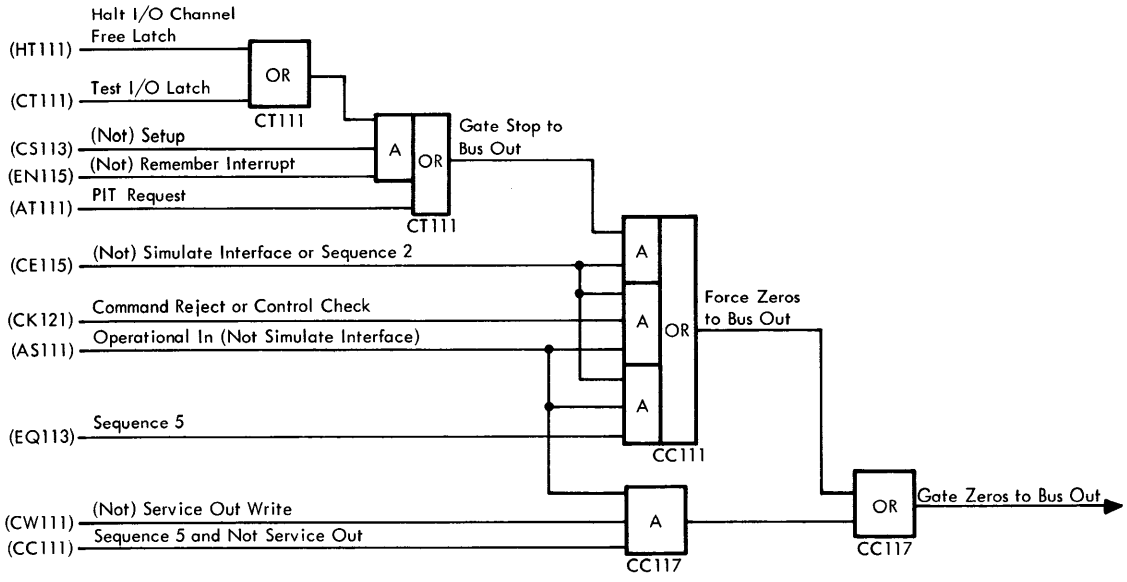


Figure 2-26. Gate Zeros to Bus-Out Logic

interface operations, and an interrupt condition is not pending in the channel ('force zeros to bus out' signal activated); (2) a 'polling interrupt request' is in the channel, the 'sequence 2' latch is off, and the channel is not simulating I/O interface operations ('force zeros to bus out' signal activated); (3) an error is detected to activate the channel's 'command reject or control check' signal with the 'operational in' signal active, the 'sequence 2' latch off, and the channel not simulating I/O interface operations ('force zeros to bus out' signal activated); (4) the channel enters a 'sequence 5' routine with the 'operational in' signal active, the 'sequence 2' latch off and the 'simulate interface' signal inactive; and (5) the channel enters a 'sequence 5' routine with the 'operational in' signal active and the 'service out write' and 'service out' signals inactive.

The 'bus out' latches and associated logic are shown in detail in Diagram 4-21. Separate descriptions of data byte gating (write-type operation), unit address byte gating and command byte gating are presented separately in the following paragraphs.

Bus-Out Data Byte Gating

- Nine polarity holds comprise 'bus out' latches.
- 'Gate out byte 0' through 'gate out byte 7' signals individually activated to gate respective B-register bytes into 'bus out' latches.
- 'Latch bus out (+)' signal latches values gated into polarity holds.
- 'Latch bus out (-)' signal (delayed) prevents changes at inputs to polarity holds from affecting outputs.
- Drop of 'service-in' signal deactivates 'latch bus out' signals, resetting 'bus out' latches.

The 'bus out' latches and associated gates and control logic (Diagram 4-21) provide a buffer for transferring individual data bytes from the B-register to the control unit via the I/O interface 'bus out' ('data out bus bits') lines. Each of the nine polarity holds comprising the 'bus out' latches has eight input data gates, one for each byte (0 through 7) from the B-register. (See insert showing 'bus out bit 0' position in Diagram 4-21.) Each input gate at the individual position receives a different 'gate out byte' gating input from the byte count encoder and a different 'B-register bit' input; e.g., the eight data gates at the 'bus out bit 0' position receive 'gate-out byte 0' through 'gate-out byte 7' gating signals and B-register bits 8, 16, 24, 32, 40, 48, and 56, respectively.

When a particular B-register byte is gated to the 'bus out' latches, the appropriate 'gate-out byte' signal gates the nine bits comprising the byte through the 'bus out' latches. For example, an active 'gate-out byte 0' signal gates B-register byte 0 bits (0 through 7 plus parity) through the 'bus out' latches.

When the data byte is gated into the polarity holds comprising the 'bus out' latches, the 'service-in', 'not gate zeros to bus out', and 'write service-out' signals activate first the 'latch bus out (+)' signal and then the 'latch bus out (-)' signal. The 'latch bus out (+)' signal locks (latches) the data byte values into the polarity holds, and the delayed 'latch bus out (-)' signal prevents possible changes at the polarity hold inputs from affecting the outputs. Thus, the data byte gated into the 'bus out' latches is stable until the 'service-in' signal drops (indicating the data byte is no longer required on the 'bus out' lines). When the 'service-in' signal falls, the two 'latch bus out' signals are deactivated to reset the polarity holds and enable the next data byte to be gated into the 'bus out' latches.

For channel test mode or diagnostic operations, data bytes gated through the 'bus out' latches are also gated to the simulate interface register. (The bits are also present on the I/O interface 'bus out' lines; however the I/O interface 'service-in' and 'service-out' lines are inhibited at the channel. Thus, channel to control unit byte transfers are inhibited.

Bus-Out Unit Address Gating

- 'Setup' signal gates unit address register byte through polarity holds of 'bus out' latches.
- Outputs of polarity holds held stable by 'setup' signal.

'Bus out' gating for the nine unit address bits (0 through 7 plus parity) from the unit address register is provided by a single AND at the input to each of the nine polarity holds comprising the 'bus out' latches (Diagram 4-21). During initial or chain command setup operations, the 'setup' signal is activated and gates the unit address through the polarity holds to the I/O interface 'bus out' lines. For unit address byte gating, the two 'latch bus out' signals are not activated. Thus, the unit address bits gated to the polarity hold's inputs are held stable at the polarity hold's outputs by the 'setup' gating signal. The fall of the 'setup' signal degates the unit address byte to the 'bus in' latches.

Bus-Out Command Gating

- 'Gate command out to bus out, address-in' signal gates command register byte through polarity holds of 'bus-out' latches.
- Outputs of polarity holds held stable by 'gate command out to bus out, address-in' signal.

'Bus out' gating for the nine command register bits (0 through 7 plus parity) is provided by a single AND at the input to each of the polarity holds comprising the 'bus out' latches (Diagram 4-21). During an initial or chain command selection routine (after the unit address byte is gated to the 'bus out' latches), the channel activates the 'gate command out to bus out, address-in' gating signal. This signal gates the command register byte through the polarity holds to the I/O interface 'bus out' lines. For command register byte gating, the two 'latch bus out' signals are not activated. Thus, the command byte bits gated to the polarity hold's inputs are held stable at the polarity hold's outputs by the 'gate command out to bus out, address-in' gating signal. The fall of this signal degates the unit address byte to the 'bus in' latches.

Bus-Out Zeros Command Gating

- 'Gate zeros to bus out' signal activated to place stop command byte to all zeros with good parity on 'bus out' lines.
- 'Gate zeros to bus out' signal activates 'latch bus out (-)' signal.
- 'Latch bus out (-)' signal places zeros in data polarity holds of 'bus out' latches and a one at parity polarity hold output.

For operations where the channel may be required to internally generate and send a stop command byte to the control unit, the 'gate zeros to bus out' signal is activated (Diagram 4-21). (The stop command byte, when accompanied by a 'command out' signal from the channel, causes the operating I/O device to proceed to its normal ending point without sending further 'service-in' signals to the channel). The 'gate zeros to bus out' signal is initiated for a 'halt I/O, channel free', 'test I/O', 'polling interrupt request', 'command reject or control check, or a 'sequence 5' routine (Figure 2-26). Note that the 'gate zeros to bus out' signal cannot be generated while the channel is simulating the I/O interface and that in all but one case ('sequence 5' routine, 'not service-out write'), the 'sequence 2' latch must be off.

When the 'gate zeros to bus out' signal (Diagram 4-21) is activated, the signal is OR'ed to activate the 'latch bus out (-)' signal. This signal forces zeros into all data bit polarity holds and prevents other signals which may be inadvertently gated to the 'bus out' latch from affecting the zero outputs of the polarity holds. In addition, the 'gate zeros to bus out' signal is OR'ed at the parity bit polarity hold output to produce a logic 1 data bus-out parity (bus-out parity) bit. Thus, while active, the 'gate zeros to bus out' signal provides a stop command byte of all zeros with proper parity on the 'bus out' lines. Depending upon the channel operation in progress, the channel may or may not raise the 'command out' signal to the I/O interface; i.e., the stop command byte may or may not be gated into the control unit.

Bus-Out Parity Checking

- 'Data out bus parity check' signal activated if even parity detected on 'bus out' lines.
- For unit address or command byte transfers, active 'data out bus parity check' signal causes turn on of 'channel control check' latch.
- For data transfers, active 'data out bus parity check' signal causes turn on of 'channel data check' latch.

Outputs of the 'bus out' latches are continuously checked for correct (odd) parity by exclusive-OR parity checking logic (Diagram 4-21). 'Data out bus' ('bus out') bits 0 through 8 and P are applied to the parity checking logic. If odd parity is not present, the 'data out bus parity check' signal is activated and sent to sampling logic. If a unit address byte or command byte is being gated through the 'bus out' latches, an active 'data out bus parity check' signal is sampled and causes turn-on of the 'channel control check' latch. This terminates the operation in progress. If a data byte is being gated through the 'bus out' latches, an active 'data out bus parity check' signal is sampled and causes turn-on of the 'channel data check' latch. This does not terminate the write operation in progress, but will break a chain command operation, if the operation is specified.

BUS-IN LATCHES

- 'Bus in' latches buffer data, address, and status bytes from I/O interface and 'simulate interface' logic into channel.
- Nine polarity holds comprise 'bus in' latches.

The 'bus in' latches (Diagram 4-22) enable the channel to transfer data bytes, address bytes, and status bytes received from the control unit (via the I/O interface 'bus in' lines) to channel registers and logic. For simulate interface operations, the 'bus in' latches provide a means of transferring data bytes from the simulate interface register. In addition, unit address bytes and simulated status bytes generated by the simulate interface control logic are transferred via the latches to channel registers and logic.

The 'bus in' latches consist of nine positions, with each position functioning as a polarity hold. Individual gating logic is associated with each position. 'Latch bus in (-)', 'latch bus in (+)' and 'simulate interface' signals provide

control for polarity hold gating and latching. Indicators on the CE panel provide a visual display of the 'bus in' latch contents.

'Bus in' latch data byte, address byte, status byte, and simulate interface operations are described separately in the following paragraphs.

Bus-In Data Byte Handling

- Data byte bits on 'bus-in bit 0-7 and P' lines AND'ed to polarity hold's outputs.
- 'Service-out' signal rises.
- 'Latch bus in (+)' signal locks data bits in polarity holds.
- Delayed 'latch bus in (-)' signal inhibits input AND's to polarity holds.
- Contents of 'bus in' latches checked for correct (odd) parity: detected even parity causes turn on of 'channel data check' latch.
- 'Service-out' signal drops, resets 'bus in' latches (polarity holds) and enables input AND's.

The 'bus in' latches (Diagram 4-22) provide a transfer path for data bytes from the control unit to the B-register during read-type operations. For read-type operations, the 'bus in' latches receives data bytes on the 'bus-in bit 0' through 'bus in bit 7' and 'bus in bit P' lines. With the two 'latch bus in' signals inactive, the data bits placed on these lines are AND'ed through input gates to the polarity holds. Logic values at the inputs to the polarity holds appear on the 'data in bus bit 0' through 'data in bus bit 7' and 'data in bus bit P' outputs of the polarity holds (Diagram 4-22). When the channel activates the 'service-out' signal (accepting the data byte from the control unit), the 'service-out' signal is AND' ed with the 'not read, sequence 3, CCW valid' and 'not CDA block service-out' signals to activate the 'latch bus-in (+)' signal. This signal locks (latches) the data bits into the polarity holds of the 'bus in' latches. After a logic delay, the 'latch bus in (-)' signal is activated to inhibit the input gates receiving the data byte bits. This prevents possible logic changes at the input gates from affecting the polarity hold outputs once the data byte is latched. After the channel gates the data byte into the B-register, the 'service out' signal drops, causing the two 'latch bus in' signals to deactivate. This resets the polarity holds and enables the input gates to the polarity holds for the next data byte input.

While the data byte is in the 'bus in' latches, the 'data in bus bit 0-7 and P' outputs are checked for a parity error by the 'bus in parity checking' logic (logic blocks designated "odd" in Diagram 4-22). If a parity error (even parity) is detected, the 'data in bus parity check' signal is activated and sent to sampling logic. If the signal is active when sampled, the 'channel data check' latch is turned on. Turn on of this latch does not terminate the read operation but does break chain command operation, if the operation is indicated.

Bus-In Address Byte Handling

- Unit address byte bits on 'bus in bit 0-7 and P' lines AND'ed to polarity hold outputs.
- Bits not latched into 'bus in' latches (polarity holds).
- Unit address bytes from 'bus out' latches used for comparison with unit address register contents on I/O device selection.
- Unit address byte from 'bus in' latches gated into unit address register on polling interrupt sequence.
- 'Bus in parity checking' logic checks address byte parity; parity error causes turn on of "interface control check" latch.

The 'bus in' latches (Diagram 4-22) provide a transfer path for unit address bytes from the control unit (via the I/O interface). The unit address bits are applied to the 'bus in bit 0 through 7 and P' lines and are AND'ed to the output of the polarity holds. Since the channel does not respond to an 'address in' signal from the control unit until after channel operations involving the unit address byte are completed, the control unit sustains the unit address bits on the 'bus in' lines. For this reason, the 'latch bus in (-)' and 'latch bus in (+)' signals are not activated; thus, the polarity holds comprising the 'bus in' latches are not latched and the polarity hold outputs are sustained by the stable unit address bit inputs.

Unit address bytes from the 'bus in' latches are supplied to 'unit address compare' logic and to the unit address register 'input gating' logic. For initial and chain command setup operations, the unit address byte from the 'bus in' latches is compared with the unit address in the unit address register. A successful comparison indicates that the control unit has responded with the address of the desired I/O device, and setup operations are continued by the channel. A comparison mismatch indicates that the control unit has responded with the wrong I/O device address and an interface control check condition is detected to terminate the operation.

For a polling interrupt routine, the channel gates the unit address byte from the 'bus in' latches into the unit address register. When this occurs, the unit address register contents are available for gating to the CPU interface 'unit address bus out' lines during an interrupt routine.

The unit address byte from the 'bus in' latches is also parity checked by the 'bus in parity checking' logic. If incorrect parity is detected, the 'data in bus parity check' signal activated. If active when sampled, ('start I/O latch', 'chain command latch', 'CCW valid' or 'command reject or control check' signal active when 'address-in' signal is active) the 'data in bus parity check' signal turns on the 'interface control check' latch to terminate the operation in progress.

When the channel drops the 'address-out' signal (for an I/O device selection routine) or raises the 'command out' signal (for a 'polling interrupt' sequence), the control unit removes the unit address byte from the 'bus in' lines; this removes the unit address bits from the 'bus in' latches output lines.

Bus-In Status Byte Handling

- Status byte AND'ed to 'bus in' latch (polarity holds) outputs.
- 'Latch status byte' signals activate two 'latch bus in' signals; status bits latched into polarity holds.
- Outputs are used by channel logic, dependant upon operation in progress.
  1. Available for CSW store operation.
  2. Gate 'jump command address CC' logic.
  3. Gate 'turn on interrupt end' logic.
  4. Gate 'turn off select out, channel free' and 'turn on chain command latch' logic.
  5. Examine for all zeros on initial or CC setup operations.
- Status bits checked by 'bus in parity checking' logic; error turns on 'interface control check' latch.

The 'bus in' latches (Diagram 4-22) provide a transfer path for status bytes from the control unit to channel logic. When a status byte is received on the 'bus in bit 0' through 'bus in bit 7' and 'bus in bit P' lines, the outputs of the polarity holds comprising the 'bus in' latches follow the status byte inputs.

Following receipt of the status byte, the channel turns on the 'latch status byte' trigger. The 'latch status byte trigger' signal is AND'ed with the 'not read, sequence 3, CCW valid' and 'not CDA block service out' signals to activate the 'latch bus in (+)' signal (Diagram 4-22). This signal latches the status bits into the polarity holds comprising the 'bus in' latches. After a logic time delay, the 'latch bus in (-)' signal is activated to inhibit the input AND's to the polarity holds. This prevents possible logic changes on the 'bus in bits' lines from affecting the outputs of the polarity holds.

Status bits on the 'data in bus bit' lines from the 'bus in' latches are distributed to channel logic as follows: (1) 'data in bus bits' 0 through 7 plus P are supplied to the SBI gating logic for gating to storage during a 'CSW store' operation; (2) 'data in bus bits' 1 and 5 are supplied to 'jump command address' logic and, when active during a chain command operation, causes the channel to fetch a CCW from a storage location eight bytes higher than the location from which the current CCW was fetched; (3) 'data in bus bits' 4 ('channel end' bit) and 5 ('device end' bit) gates the 'turn on interrupt end' logic and providing either of the bits is active and channel conditions are proper, activates the 'turn on interrupt end' signal to turn on the channels' interrupt latch; (4) 'data in bus bit' 5 ('channel end' bit) gates the 'turn off select out, channel free' logic and 'turn on chain command' latch logic for chain command operations; and (5) during an initial or chain command setup operation, 'data in bus bit 0' through 'data in bus bit 7' outputs are examined by the 'interrupt control' logic for all zeros. In case 5, if all zeros are detected the operation is allowed to continue; if not, the operation may be terminated.

The status bits on the 'data in bus' lines from the 'bus in' latches are also checked for correct (odd) parity to the 'bus in parity checking' logic. If parity is even, the 'data in bus parity check' signal is activated and, when sampled, turns on the 'channel control check' latch. If the channel is not already engaged in terminating the operation, turn on of the 'interface control check' latches terminates the operation in progress.

Bus-In Simulate Interface Operation

- 'Simulate interface' signals enable 'manual set bit' AND's to 'bus in' latches (polarity holds); 'bus in bit' AND's are inhibited.
- Data bytes, unit address bytes, and simulated status bytes are gated by 'simulate interface control' logic through 'bus in' latches (via 'manual set bits' lines.)

When the channel is simulating I/O interface operations (during test-mode or 'CPU diagnostic control' operations), an active 'simulate interface' signal is supplied to the 'bus in' latches (Diagram 4-22). The 'simulate interface' signal activates the 'latch bus in (-)' signal which inhibits the 'bus in bit' AND's to the polarity hold inputs. At the same time, the 'simulate interface' signal enables the 'manual set bit' AND's to the polarity hold inputs.

During simulate I/O interface operations, data bytes for read operations are gated from the simulate interface register to the 'manual set bit 0' through 'manual set bit 7' and 'manual set parity' lines. Data bytes on these lines are

AND'ed to the polarity hold inputs; outputs of the polarity holds follow the gated 'manual set bits' inputs.

Unit address bytes and simulated status bytes (simulated by the 'simulate interface control' logic) are also gated through the polarity holds comprising the 'bus in' latches in the manner described for data bytes. When a device end status byte is simulated by the 'simulate interface control' logic, the status byte ('manual set bit 5' active) is gated through the 'bus out' latches in the manner described. In addition, the 'simulate unit free' signal (which generates the 'device end' status byte) is OR'ed at the output of the 'bus in bit 5' polarity hold (Diagram 4-22) to provide an active 'bus in bit 5' signal to the 'bus in parity checking' logic.

Note: The 'bus in bit 5' signal to the 'bus in parity checking' logic is also activated (after a delay) by the output of the 'bus in bit 5' polarity hold; simulated I/O interface timing considerations with respect to 'bus in parity' checking necessitates the requirement to OR the 'simulate unit free' signal at the output of the 'bus in bit 5' polarity hold. For more detailed information on simulate interface operations, refer to "Simulate Interface Register and Control" in Chapter 6.

MARK-B REGISTER

The mark-B register and associated logic (Figure 2-27) perform two functions for channel operation. For read-type operations, the function of the mark-B register is to store mark bits at bit positions corresponding to B-register byte positions receiving data bytes from the I/O interface. For write CDA operations, the mark-B register stores the significant count register bits while the channel is fetching a new CCW.

The mark-B register consists of nine positions (eight mark bits and one parity); contents of the mark-B register are displayed at MARK-B indicators on the CE panel.

Mark-B Register Simplified Description

- For read-type operations, mark bit activated in mark-B register each time data byte gated into B-register.
- Mark-B register positions 0-7 correspond to B-register byte positions 0-7, respectively.
- Gating signal that gates byte into B-register activates corresponding mark-B register bit.
- Parity change' logic maintains odd parity for mark-B register.
- For read CDA operations, doublegating activates two mark bit positions simultaneously.
- When doublegating ceases, either position 0-3 or 4-7 is reset, dependent upon:
  1. Whether operation is read or read backward.
  2. Value of DAB received in new CCW.
- For write CDA operations, significant count bits gated to mark-B register from count register.
- 'BC equals CTB' condition detected with CTB bits in mark-B register.
- 'Incorrect length' detection causes channel to gate count bits back to count register.

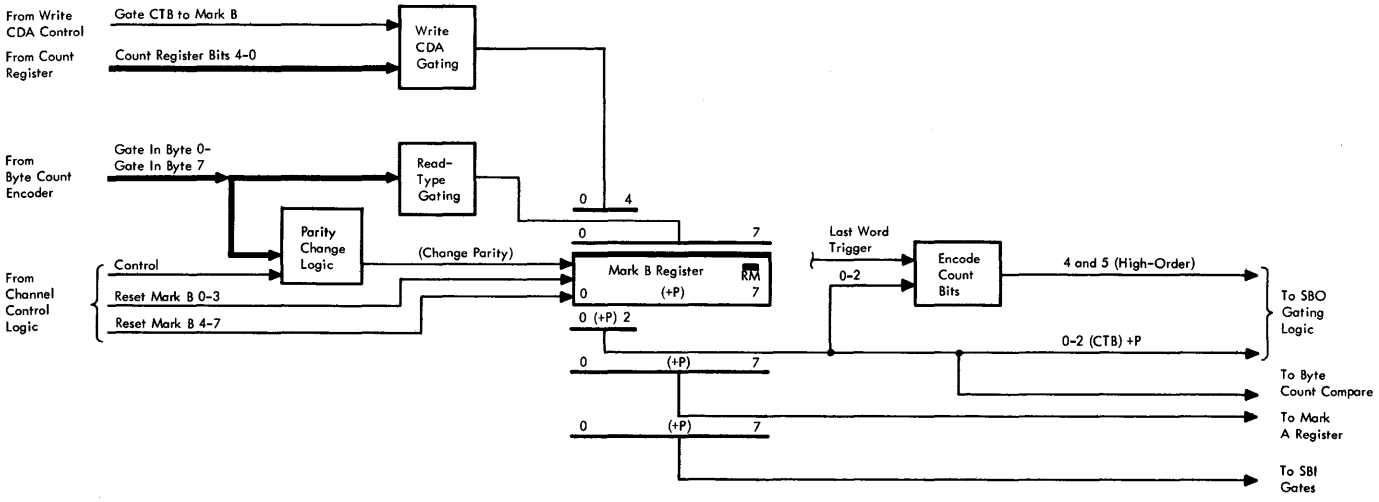


Figure 2-27. Mark-B Register, Block Diagram

For read-type operations, an active mark bit is sent to storage for each data byte (within a doubleword) that is to be stored in main storage. Since data bytes to be stored are individually gated into the B-register to assemble the doubleword, corresponding mark bits are activated in the mark-B register when the bytes are gated into the B-register. 'Read-type gating' logic (Figure 2-27) receives the same 'gate in byte 0' through 'gate in byte 7' signals (from the byte count encoder) that gate data bytes into the B-register. Each time a 'gate in byte' signal is activated the corresponding mark bit in the mark-B register is activated and a data byte is gated into the B-register. Since the mark-B register bits are activated on a bit, rather than a byte basis, 'parity change' logic associated with the mark-B register monitors the gating of mark bits into the register and changes the mark-B register parity position to maintain odd parity.

For read CDA operations, the first four data bytes are doublegated to the B-register; therefore two mark bit positions of the mark-B register are activated simultaneously when doublegating is performed. When doublegating ceases, the 'reset mark-B (4-7)' signal or the 'reset mark B (0 -3)' signal resets the designated mark bit positions. Which of the two signals is generated depends upon: (1) whether the operation is a read or read backward operation; and (2) the number of active mark bits (or bytes to the B-register) versus the value of the DAB (data address register bits 21 through 23) from the new CCW.

For read-type operations, mark-B register bits 0 through 7 plus P are gated to the mark-A register when the data contents of the B-register are transferred to the A-register; i.e., a data doubleword has been assembled into the B-register, the corresponding mark bits are active in the mark-B register, and the channel initiates operations to store the data doubleword.

For write CDA operations, low-order count register bits 23 through 19 are gated to mark-B register positions 0 through 4, respectively, when operations are initiated to fetch a new CCW. The count register bits are gated through 'write CDA gating' logic (Figure 2-27) by the 'gate CTB to mark-B' signal. With the significant count bits in the mark-B register, output bits 0 through 2 (CTB bits) are supplied to the 'byte count compare' logic. This logic monitors the mark-B register bits for a 'BC equals CTB' condition, indicating that the last byte for the current CCWhas been gated from the B-register to the control unit.

If an incorrect length condition is detected while the count bits are in the mark-B register, bits 0 through 2 are supplied to the SBO gating logic for gating back into the count register. In addition, 'encode count bits' logic examines the CTB values of mark register bits 0 through 2 and the status of the 'last word' trigger to determine the proper values for high-order count bits 4 and 5. Encoded bits 4 and 5 are also supplied to the SBO gating logic for gating to the count register. Once gated from the mark-B register to the count register, the count values are used by the channel to calculate a 'residual count' value.

Mark register bits 0 through 7 plus P are also supplied to the SBI gating logic. During a logout operation mark bits 0 through 7 are stored as part of logword 2, and the mark parity bit as part of logword 3.

A more detailed description of mark-B register operations, based upon Diagram 4-23, is provided in the following paragraphs.

Mark-B Register Detailed Description

- 'Mark-B bit 0' through 'mark-B bit 7' latches turned on by 'gate in byte 0' through 'gate in byte 7' bits for read-type operations.
- 'Parity change' logic updates 'mark-B parity' latch each time 'mark-B bit' latch is turned on.
- 'Count register bit 23' through 'count register bit 19' gated to 'mark-B bit 0' through 'mark-B bit 4' latches for write CDA operation.
- 'Count register bit P2' output gated to 'parity change' logic.
- CTB on 'mark register bit 0' through 'mark-B register bit 2' lines monitored by byte count compare logic.
- Detected incorrect length condition causes count bits ('mark-B register bit 0' through 'mark-B register bit 2', 'encode mark bit 3' and 'encode mark bit 4') to be gated to count register via SBO lines.

The mark-B register and associated logic (Diagram 4-23) consist of nine latches designated 'mark-B bit 0' through 'mark-B bit 7' and 'mark-B parity' latches, as well as associated 'parity change' and 'encode count bits' logic. For read-type operations, the 'mark-B bit 0' through 'mark-B bit 7' latches may be turned on by the 'gate in byte 0' through 'gate in byte 7' signals, respectively. When a data byte is gated (or doublegated) into the B-register, the 'gate in byte' signal(s) from the byte count encoder that gate the data byte to the B-register also turn on mark-B register latch(es). For example, if the channel is singlegating a data byte into the B-register byte 0 position, the active 'gate in byte 0' signal turns on the 'mark bit 0' latch. If the channel is doublegating a data byte into B-register positions 0 and 4, the active 'gate in byte 0' and 'gate in byte 4' signals turn on the 'mark-B bit 0', and 'mark-B bit 4' latches. For read-type operations, the 'mark-B parity' latch is updated by the parity change logic' each time a 'gate in byte' signal turns on a 'mark-B bit' latch.

For a write CDA operation 'count register bit 23' through 'count register bit 19' are gated into the 'mark-B bit 0' through 'mark-B bit 4' latches, respectively. The 'count register bit P2' output is gated to the 'parity change' logic (Diagram 4-23). These bits are gated by the 'gate CTB to mark-B' signal which is activated when the channel detects a count value equal to or less than 16 bytes with the CDA flag active. While the channel is fetching the next CCW, the 'mark-B register bit 0' through 'mark-B register bit 2' outputs (CTB) are monitored by the byte count compare logic for a 'BC = CTB' condition. This condition indicates that the last byte of the current CCW has been gated from the B-register.

If an incorrect length condition is detected while the significant count bits are in the mark-B register, 'mark-B register bits 0 through 2' signals plus the 'encoded mark bit 3' and 'encoded mark bit 4' signals (Diagram 4-23) are gated to the count register via the SBO gating logic. The channel may then use the count to calculate a residual count value.

Mark register reset operations, parity change logic operation, and encode count bits operations are described separately in the following paragraphs.

Mark-B Reset Operations

- 'Reset mark-B 0-3', 'reset mark-B (4-7)' and 'reset mark-B parity' signals activated simultaneously when:
  1. B-register is reset during read operation.
  2. BC = CTB condition is detected for write CDA operation.
  3. 'Machine reset or setup reset' signal is activated.

- Assume (1) a read forward operation, (2) doublegating has ended, and (3) the new CCW is not established. The 'read CDA, sequence 3 reset latch' signal activates 'reset mark-B (4-7)' signal.

- Assume (1) a read forward operation, (2) doublegating is active, and (3) the new CCW DAB is on singleword boundaries. The 'read DAB 4 or read backward DAB 3' signal activates 'reset mark-B 0-3' signal.

- Assume (1) a read forward operation, (2) doublegating is active, and (3) the new CCW DAB is on doubleword boundaries. The 'read DAB 0 or read backward DAB 7' signal activates 'reset mark-B (4-7)' signal.

- For read backward operation, same conditions cause opposite 'reset mark-B' signal to be activated.

'Mark-B bits 0 through 3' and 'mark-B bits 4 through 7' latches are reset by two different reset signals: 'reset mark-B 0 through 3' and 'reset mark-B' signals, respectively. The 'mark-B parity' latch is turned on by the 'reset mark parity' signal (see Diagram 4-23). For all channel operations involving the mark-B register, except the read CDA operation, all three reset signals are activated simultaneously. The three active signals turn off all 'mark-B bit' latches and turn on the 'mark-B parity' latch.

Since the mark-B register is used only during read-type or write CDA operations, the three 'reset mark-B' signals are activated only for these operations or by the 'machine or setup reset' signal (Figure 2-28). For read or sense operations, all three 'reset mark-B' signals are activated after the mark-B register bits have been transferred to the mark-A register; the three signals are activated by the 'reset B-register, read, sequence 3' signal. For a write CDA operation (assuming the operation is normal and the significant count

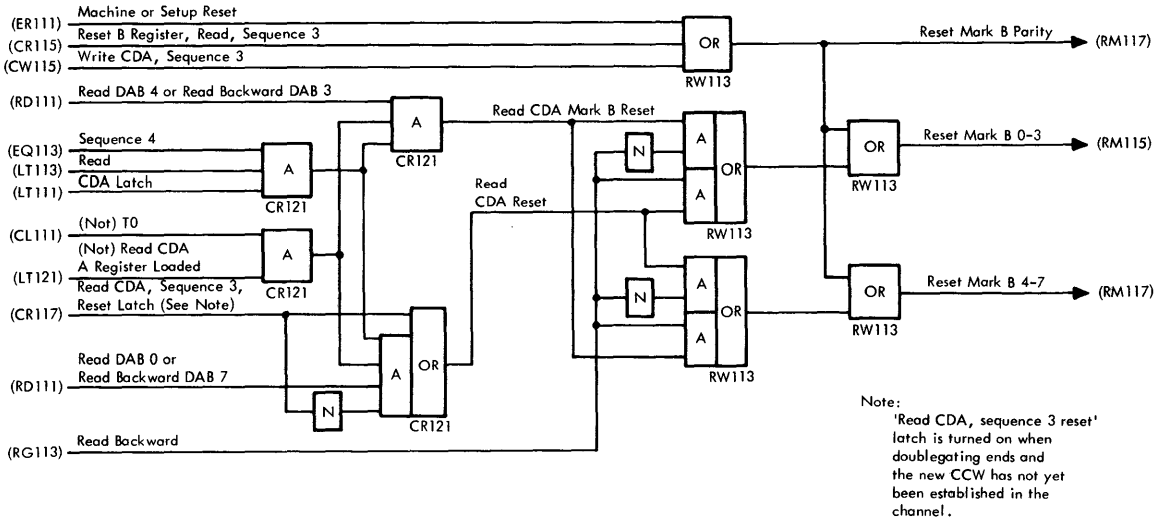


Figure 2-28. Mark-B Register Reset Logic

bits are in the mark-B register), the three 'reset mark-B' signals are activated after the 'BC = CTB' condition is detected for the current CCW; for this condition, the 'write CDA, sequence 3' signal activates the three signals to reset the mark-B register.

For a read CDA or read backward CDA operation, several possibilities exist with respect to activation of the three 'reset mark-B' signals. Each possibility is described in the following text.

Assume that the operation is a read CDA operation ('not read backward' signal in Figure 2-28), and that the channel is enabled for doublegating operations. If no mark bits are doublegated into the mark-B register before the new CCW is established in the channel, none of the three 'reset mark-B' signals is activated, doublegating is disabled, and turn-on of mark bits for the new CCW proceeds as during a normal read operation. Assume that doublegating begins and that four mark bits are doublegated to the mark-B register before the new CCW arrives; i.e., all mark-B register latches are on. In this case, doublegating ends, and the 'read CDA, sequence 3 reset' latch is turned on. The 'read CDA, sequence 3 reset latch' signal (Figure 2-28) is OR'ed to activate the 'read CDA reset' signal. This signal is AND'ed with the 'not read backward' signal and then OR'ed to activate the 'reset mark-B (4-7)' signal. This signal resets the 'mark-B bit 4' through 'mark-B bit 7' latches. If further mark bits are activated before the new CCW assumes control, singlegating sequentially turns on the 'mark-B bit 4' through 'mark-B bit 7' latches. If all latches are turned on by singlegating, the mark-B register bits are transferred to the mark-A register, the 'read CDA A-register loaded' signal is activated, and all three 'reset mark-B' signals are activated by the 'reset B-register, read, sequence 3' signal. If the new CCW still has not been established, the 'mark-B bit 0' through 'mark-B bit 7' latches may be sequentially turned on by singlegating before an overrun condition is detected to end the read CDA operation.

Assume that at least one mark bit has been doublegated, doublegating is still active, and the new CCW is established in the channel; i.e., the 'mark-B bit 0' and 'mark-B bit 4' latches are on and doublegating is still active. In this case, the DAB from the new CCW must specify either singleword or doubleword boundaries or the operation is terminated. If singleword boundaries are specified, the 'read DAB 4 or read backward DAB 3' signal is AND'ed to activate the 'read CDA mark B' reset signal (Figure 2-28). This signal is further AND'ed with the 'not read backward' signal to activate the 'reset mark-B 0-3' signal which turns off any active 'mark-B bit 0' through 'mark-B bit 3' latches. Subsequently, any of the 'mark-B bit 5' through 'mark-B bit 7' latches not already on are sequentially turned on until the 'mark-B bit 7' latch is on. Thus, for a singleword boundary DAB received for a read forward operation, only the 'mark-B bit 4' through 'mark-B bit 7' latches are on for the first data doubleword transfer (assuming that a count of at least four bytes is specified by the new CCW). If doubleword boundaries are specified while doublegating is still active, the 'read DAB 0 or read backward DAB 7' signal is AND'ed to activate the 'read CDA reset' signal. This signal is AND'ed with the 'not read backward' signal to activate the 'reset mark-B (4-7)' signal. 'Mark-B bit' latches are then sequentially turned on beginning with the first latch not on in the 'mark-B bit 0' through 'mark-B bit 3' group.

For a read backward operation, the same conditions described for the read forward operation can exist. However, since mark bits are activated in the reverse order for a read backward operation, exactly the opposite 'reset mark-B' signal is activated for the conditions described for a read forward operation. For example, a condition where doublegating is still active and the new CCW is established with the DAB specifying singleword boundaries causes the 'reset mark-B (4-7)' signal to be activated. Recall that for a read forward operation, the same conditions cause the 'reset mark-B 0-3' signal to be activated.

Mark-B Parity Change Logic

- 'Parity change' logic changes value in 'mark-B parity' latch each time a 'gate in byte' signal is activated during a read-type operation.

- Alternating turn-off, turn-on of the 'mark-B parity' latch during read-type operation maintains odd parity for mark-B register.

- For write CDA operation, 'count register bit P2' signal is gated by 'gate CTB to mark-B' signal. Logic 0 turns off 'mark-B parity' latch; logic 1 leaves latch on.

- For write CDA operation, a detected incorrect length condition causes value in 'mark-B parity' latch to change if 'last word' trigger is on, one or more CTB bits in mark-B latches equal 1, and count bit in 'mark-B bit 3' latch equals 1.

The 'change parity' logic associated with the 'mark-B parity' latch (Diagram 4-23) monitors inputs to the mark-B register latches to maintain odd parity for the register.

For read-type operations, the 'parity change' logic monitors the 'gate in byte 0' through 'gate in byte 7' signals to change the state of the 'mark-B parity' latch as required. Assume that the mark-B register is in the reset condition ('mark-B parity' latch is on) and that a read-type operation is in progress. The active 'mark-B parity' bit is sensed at OR gate 1, disabling one input to the OR. AND gate 2 is disabled by the absence of a 'change' signal, disabling the second input to OR gate 1 and enabling one input to AND gate 4. With the output OR gate 1 at the 'not' level, AND gate 2 is held disabled and AND gate 3 is disabled, enabling a second

input to AND gate 4. When a 'gate in byte' signal is activated, the 'change' signal is activated to enable the third input to AND gate 4. With AND gate 4 satisfied, the 'mark-B parity' latch is turned off by the output of AND gate 4. Th 'gate in byte' signal then drops, leaving a 'mark-B bit' latch on and the 'mark-B parity' latch off. (If read CDA doublegating is active, two 'mark-B bit' latches are on and the 'mark-B parity' latch is off; however, since one of the two 'mark-B bit' latches will be reset, the 'mark-B parity' latch is "tracking" odd parity on a singlegating basis. When doublegating ends and the appropriate 'mark-B bit' latches are reset, the 'mark-B parity' latch will reflect correct parity for the mark-B register.)

With the 'mark-B parity' latch off, the 'not mark-B parity' bit enables the output of OR gate 1; this enables one input to AND gate 2 and one input to AND gate 3. In the absence of a 'change' signal, the 'not' output of AND gate 4 enables the second input to AND gate 3 and a second input to AND gate 2. With AND gate 3 completely enabled, AND gate 4 is held disabled. When a 'gate in byte' signal activates the 'change' signal, the third input to AND gate 2 is enabled this satisfies the AND, causing the 'mark-B parity' latch to be turned on. When the 'change' signal falls, the 'mark-B parity' latch remains on to provide odd parity for the mark-B register. For read-type operations, the 'mark-B parity' latch is alternately turned off and on by the 'parity change' logic in the manner described.

For a write CDA operation, the mark-B register latches are reset before the significant count register bits are gated to the mark-B register. Thus, the 'mark-B parity' latch is on. If the 'count register bit P2' signal is a logic 0, the 'gate CTB to mark-B' signal and the logic 0 level are AND'ed to activate the 'change' signal in the 'parity change' logic (Diagram 4-23). The 'parity change' logic then turns off the 'mark-B parity' latch in the manner described for a read-type operation. If the 'count register bit P2' signal is a logic 1, the 'change' signal is not activated and the 'mark-B parity' latch remains on.

If the channel detects an incorrect length condition while significant count bits are in the mark-B register, the value in the 'mark-B parity' latch may be changed before the count bits are gated back to the count register. Whether the value in the latch is changed depends upon: (1) the status of the 'last word' trigger; (2) the value of the count bit in the 'mark-B bit 3' latch; and (3) the values of the CTB bits in the 'mark-B bit 0' through 'mark-B bit 3' latches (Diagram 4-23). The value in the 'mark-B parity' latch is changed only if one or more CTB bits equal logic 1 ('mark-B register bit 0 or 1 or 2' signal), the count bit in the 'mark-B bit 3' latch is a logic 1 ('mark-B register bit 3' signal) and the 'last word' trigger is on ('last word trigger' signal). When these conditions are present, the 'write IL sequence 5' signal is AND'ed to activate the 'change' signal. The 'parity change' signal then changes the value in the 'mark-B parity' latch as described for read-type operations. To understand the reason for changing the parity bit, refer to "Mark-B Encoder Count Bits Logic".

Mark-B Encoder Count Bits Logic

- 'Encoder count bits' logic applicable only for write CDA, incorrect length detection.
- Encodes count bits for gating to count register in accordance with status of 'last word' trigger and value of CTB bits.
- If 'last word' trigger is off, encoded outputs are value originally gated to mark-B register.
- If 'last word' trigger is on, encoded outputs are a value 8 less than originally gated to mark-B register.

The function of the 'encoder count bits' logic (Diagram 4-23) is to encode the two high-order count bits contained in the mark-B register during a write CDA operation. The two high-order encoded count bits and the three low-order CTB bits are available for gating to the count register (via the 'SBO gating' logic) if an incorrect length condition is detected.

For a write CDA operation, the significant count bits are gated to the mark-B register when the count is equal to or less than 16 bytes (data from two doublewords are yet to be gated to the I/O interface) and the 'last word' trigger is off. If an incorrect length condition is detected before the 'last word' trigger is turned on, the same count value gated into the mark-B register must be gated out of the mark-B register. If the incorrect length condition is detected after the 'last word' trigger is turned on, the count value gated from the mark-B register must be reduced by 8 bytes.

Recall that low-order count register bits 23 through 21 (CTB) are gated to the 'mark-B bit 0' through 'mark-B bit 2' latches, respectively. Upon detection of an incorrect length condition these bits are gated to the count register CTB positions unchanged. However, these bits along with the status of the 'last word' trigger are also used to determine the values of the two high-order count bits adjacent to the CTB bits.

At the mark-B register, the CTB values on the 'mark-B register bit 0, 1, and 2' lines are applied to an OR gate. A logic 1 on any line activates the 'mark-B register bit 0, 1 or 2' signal (Diagram 4-23). With this signal active and the 'last word' trigger off, the 'encode mark bit 3' signal is activated and the 'encode mark bit 4' signal is inactive. Thus, a count value greater than eight bytes and less than 16 bytes is gated to the count register. This is the value originally gated into the mark-B register.

If the 'mark-B register bit 0, 1, or 2' signal is inactive (CTB equals 0) and the 'last word' trigger is off, the 'encode mark-B bit 4' signal is activated and the 'encode mark-B bit 3' signal remains inactive. This represents a count value of 16 bytes (originally gated to the mark-B register).

If the 'last word' trigger is on (data from only one doubleword remains to be gated to the I/O interface) and the 'mark-B register bit 0, 1 or 2' signal is active, both the 'encode mark bit 3' and 'encode mark bit 4' signals remain inactive to represent a count of less than eight bytes. This is a value eight bytes less than the value originally gated into the mark-B register.

If the 'last word' trigger is on and the 'mark-B register bits 0, 1 or 2' signal in inactive (CTB equals 0), the 'encode mark bit 3' signal is activated and the 'encode mark bit 4' signal remains inactive. This represents a count value of eight bytes and is less than the count value of 16 bytes originally gated into the mark-B register.

Table 2-4 summarizes the operation of the 'encoder count bits' logic. In the table, the active 'mark-B register bit 0, 1 or 2' signal is referred to as "CTB $\geq$ 1" (CTB greater than or equals 1); an inactive signal is referred to as "CTB = 0".

Table 2-4. Mark-B Encoded Count Values

Last Word Trigger Status	Low-Order Count Bits 0-2	Encode Mark Bit 3	Encode Mark Bit 3 (High-Order)	Encoded Count Value
Not LWT	CTB $\geq$ 0	0	1	16 bytes
Not LWT	CTB $\geq$ 1	1	0	>8 and <16 bytes
LWT	CTB = 0	1	0	8 bytes
LWT	CTB $\geq$ 1	0	0	<8 bytes

MARK-A REGISTER

- Mark-A register latches 0-7 and P receive inputs from mark-B register on read-type operations.
- Mark-A register bits gated to storage on 'mark bit 0-7 and P' lines by 'BCU response powered' signal.
- For CSW store (all bytes), logout, and manual store operation, 'mark-A output gating' logic activates all mark bit signals to storage.
- For CSW store (channel and control unit status only) operation, 'mark-A output gating' logic activates 'mark bits 4, 5, and P' signals to storage.
- For IPL end condition operation, 'mark-A output gating' logic activates 'mark bits 2, 3, and P' signals to storage.
- For CCW fetch operations, 'mark-A output' gating logic activates 'mark bit P' to storage.
- For write-type operations, reset condition of mark-B register (all zeros, P bit active) is gated to storage.

The mark-A register and associated 'mark-A output gating' logic (Diagram 4-24) enable the channel to gate mark bits to storage for the various store-type operations performed by the channel.

The mark-A register consists of nine latches designated 'bit 0' through 'bit 7' and 'bit P'. These latches act as a buffer between the mark-B register and the mark lines to storage for channel read-type storage operations. During read-type operations, the contents of the mark-B register (mark-B register bits 0 through 7 plus P) are gated into the mark-A register 'bit 0' through 'bit 7' and 'bit P' latches by the 'gate mark-B to mark-A' signal each time the mark-B register contents are ready for transfer. Before activating the 'gate mark-B to mark-A' signal, the 'machine or ingate reset' signal resets the mark-A register latches. With the 'bit P' latch reset, a logic 1 is active at the latch output; all other latches have a logic 0 output when reset. When the mark-B register bits are stored in the mark-A register, the 'mark-A register bit 0' through 'mark-A register bit P' outputs are applied to AND's which are gated by AND'ing the 'gate data address to SAB', 'BCU response powered', and 'not IPL end condition' signals. When these signals are active simultaneously, the outputs of the mark-A register latches are AND'ed then OR'ed to main storage on the 'mark bit 0' through 'mark bit 7' and 'mark bit P' lines. In main storage, an active mark bit causes the corresponding data byte in the data doubleword gated from the channel to be stored.

For other channel store operations, the mark-A register outputs are not gated, and mark bits to main storage are activated by OR'ing gated signals to the appropriate 'mark bit 0' through 'mark bit 7' and 'mark bit P' lines (Diagram 4-24).

For a CSW store operation ('Z address latch gated' signal), test-mode, manual store operation, or logout operation ('on log' signal) all mark bit lines to main storage are activated when the channel receives the 'BCU response' signal. In the 'mark-A output gating' logic (Diagram 4-24) the 'BCU response' signal activates the 'gate all marks' and 'gate parity' signals. These signals are OR'ed to activate the 'mark bit 0' through 'mark bit 7' and 'mark bit P' signals to main storage. Thus, for the CSW store operation, logout operation (logwords 1, 2, and 3 stored) and the manual store operation, all bytes of the doublewords sent to storage by the channel are stored in the appropriate storage locations. For the CSW store operation, only the channel and control unit status bytes are stored when the 'start I/O' latch is on or the 'halt I/O' signal is active. These bytes are located in byte positions 4 and 5 of the CSW. For this operation (Diagram 4-24) the 'start I/O or halt I/O' and 'not on log' signals prevent generation of the 'gate all marks' signal. With the 'Z address latch gated' signal active, the 'BCU

response' signal activates the 'gate marks 4 and 5' and 'gate parity' signals; these signals activate the 'mark bit 4', 'mark bit 5' and 'mark bit P' lines to main storage.

At the end of an IPL operation ('IPL end condition' signal active), the channel stores the channel and unit addresses in byte locations 2 and 3 of storage location 0. To enable storage of the two bytes, the 'IPL end condition' signal is AND'ed with the 'BCU response' signal (Diagram 4-20) to activate the 'gate marks 2 and 3' and 'gate parity' signals. These signals activate the 'mark bit 2', 'mark bit 3' and 'mark bit P' signals to main storage.

For a TIC CCW fetch or a normal CCW fetch operation, the 'data address to SAB on TIC cycle' or 'command address to SAB gate' signal, respectively, activates the 'gate parity' signal. This signal activates the 'mark bit P' signal to main storage. (Although marks are not required for fetch operations, the 'mark bit P' line to main storage is activated to prevent storage from detecting a mark parity error.) For normal write-type operations, the mark-A register latches are in a reset condition; i.e., only the 'mark-A register bit P' signal is active. Each time a data doubleword is fetched from main storage, the 'gate data address to SAB', 'BCU response powered' and 'not IPL end condition' signals are AND'ed to gate the active 'mark-A register bit P' signal to the 'mark bit P' lines. Thus, the mark lines to storage always have correct parity for both store and fetch operations.

PARITY CIRCUITS

- Based on exclusive-OR functions.
- Odd parity is considered good parity.
- Parity can be checked or generated by the same circuit types.

General Description

In the 2860 the prevalent method of verifying accurate information transfer is to transmit an odd number of binary 1's in each basic group of information (odd parity) and check the information group after transfer to see if the parity is still odd. A detected even parity after a transfer signifies an error. Information is usually transferred in groups called bytes. A byte consists of nine bits; eight data bits and one control bit. The control bit is called the parity (P-bit and is used to maintain an odd 1-bit count in its associated byte.

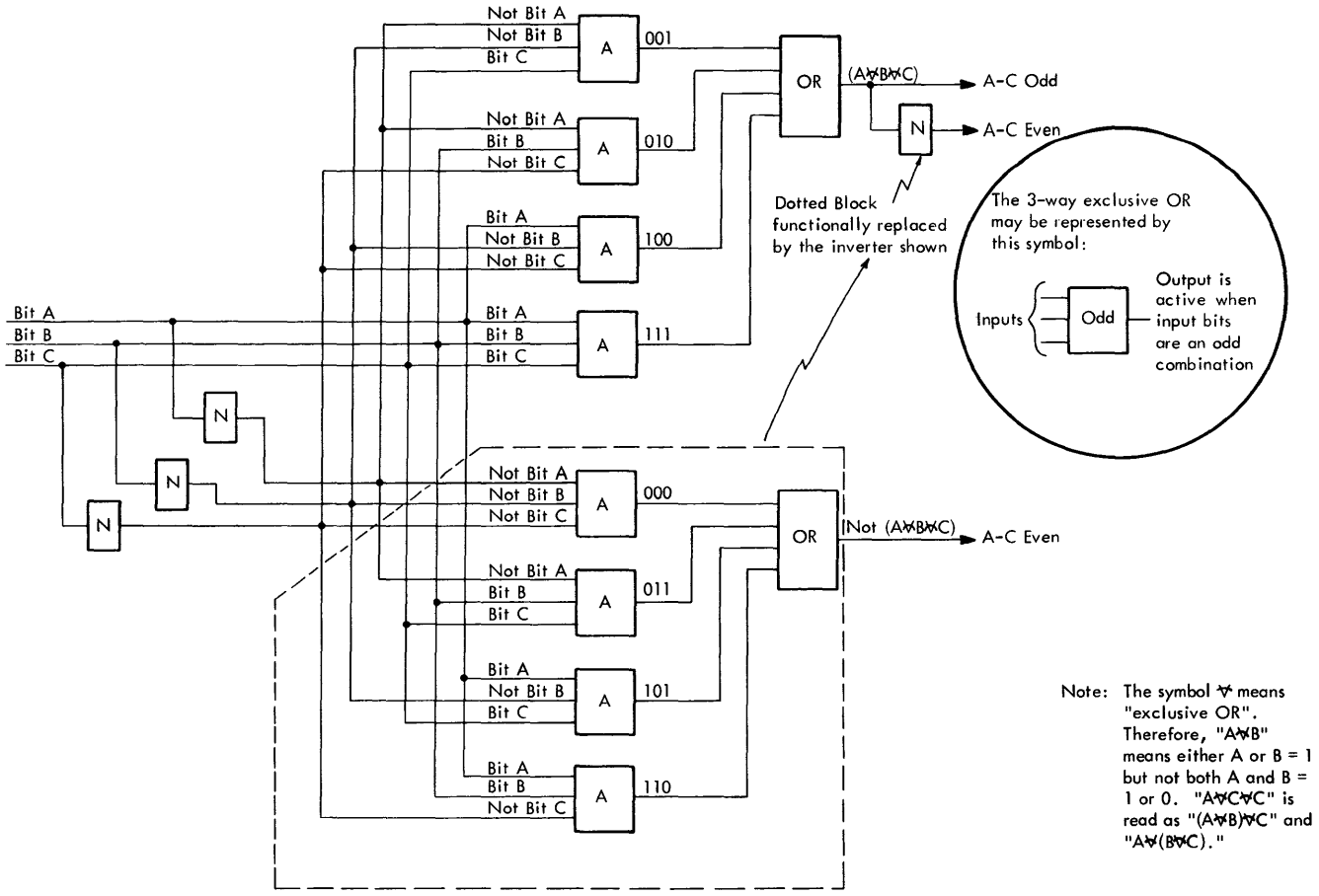
The circuits that generate or check parity are often called parity trees. There is little difference between circuits used for parity generation and circuits used for parity checking. Both are based on the use of exclusive-OR functions, as shown in Figure 2-29. For parity checking, an even number of binary 1's in the 9-bit byte indicates an error condition. For parity generation, an even number of 1's in the eight data bits of the byte indicate that the P-bit should be set to 1 to maintain odd parity. Refer to Figure 2-30.

The exclusive-OR (in various configurations) is the basic logic of parity circuits (Figure 2-29). A two-input exclusive-OR has an active output when either one or the other of its inputs is active (equal to a logic 1) but not when both inputs are active or both inputs are inactive.

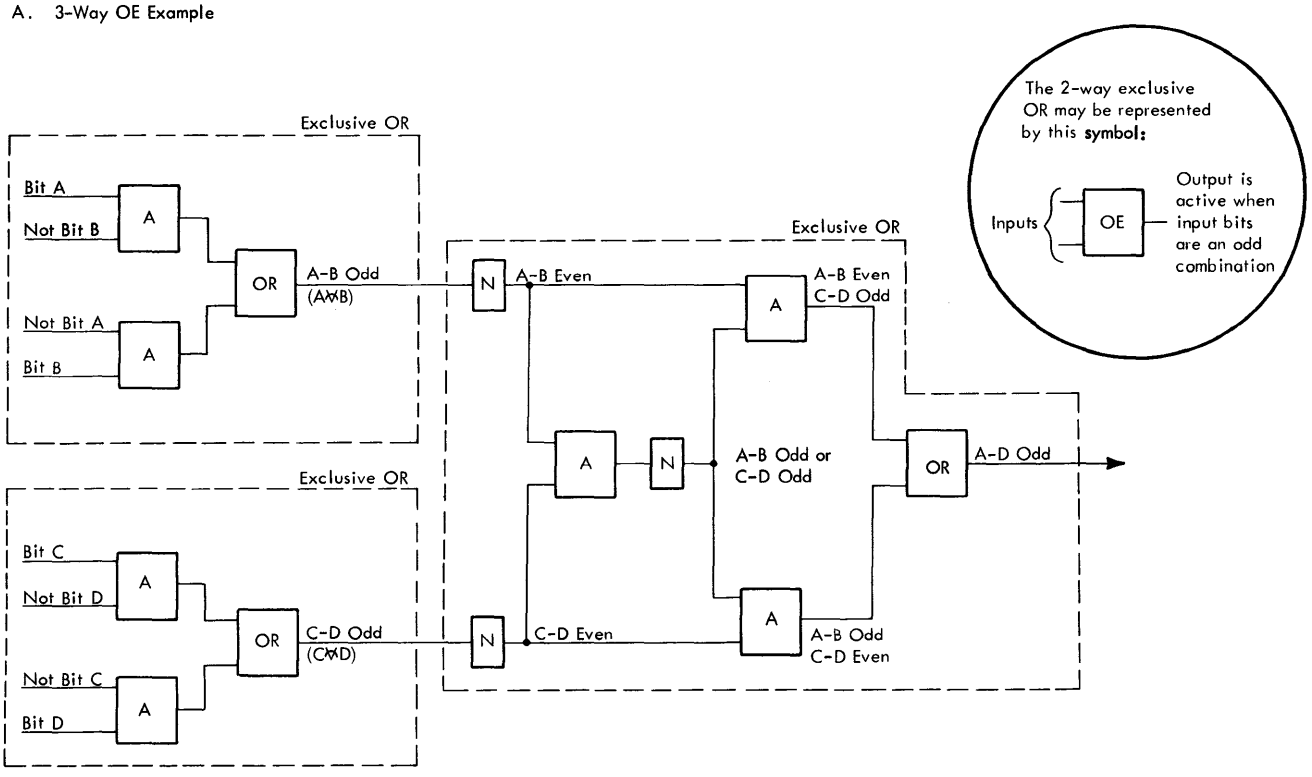
Operation and Location

The operation of the various parity checking and parity generating circuits in the channel is explained in the description of the associated functional unit in this chapter. A list of the channel's 'parity generating' and 'parity checking' logic and their locations in ALD logic follows:

Storage Bus-In Parity Generate and A-register Parity Check	
P0	MB153
P1	MB157
P2	MB163
P3	MB167
P4	MB173
P5	MB177
P6	MB183
P7	MB187
Flag Register Parity Check	RF113
Storage Protect Register Parity Check	RP115
Adder Parity Generate and Parity Check	
P0	SA157
P1	SA161
P2	SA163
Byte Counter Parity Check	SC115
Bus-In Parity Check	UB117
Bus-Out Parity Check	UB143
Unit Address Bus-In Parity Generate	RU116



A. 3-Way OE Example



B. 2-Way OE Examples

Figure 2-29. Three-Way and Two-Way Exclusive OR's, Functional Diagrams

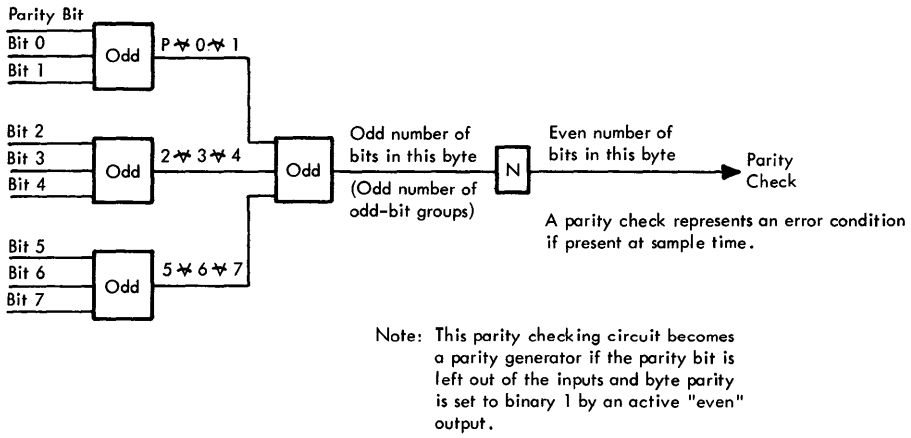


Figure 2-30. Bus-In and Bus-Out Parity Checking

CHAPTER 3. PRINCIPLES OF OPERATION

This chapter describes the principles of operation for the 2860 Selector Channel, and provides separate descriptions of the start I/O, halt I/O, test I/O and test channel operations. In addition, separate descriptions are provided for the channel initial program load (IPL) operation, the fault locating test (FLT) operation, polling interrupt operations, and program controlled interrupt (PCI) operations as well as other operations peripheral to the above operations. For the overall operation of the 2860 Selector Channel, see Diagram 3-1, FEMDM (Selector Channel Unit Data and Control), as well as other figures referenced throughout this chapter.

Note: The terms "BCU interface" and "CPU interface" used throughout this text are defined at the 2860 Selector Channel. In systems where the channel is not actually connected to BCU and CPU interfaces, the appropriate interface designations should be substituted for the "BCU interface" and "CPU interface" terms. For example in the Model 67-2 system, the Selector Channel interfaces with the 2846 Channel Controller and the terms "CPU-channel element interface" and "channel-storage element interface" may be substituted for the "CPU interface" and "BCU interface" terms, respectively. In the Model 91 system, the term "PSCE/channel interface" may be substituted for both the "CPU interface" and "BCU interface" designations.

START I/O INSTRUCTION

The Start I/O instruction is the basic instruction by which channel operations are initiated to control the flow of information between an attached I/O device and main storage or to control I/O device operations. The Start I/O instruction, originated at the CPU, initiates the channel read-type and write-type operations. Read-type operations include: (1) the read operation performed to transfer data from a selected I/O device to ascending-order address locations in main storage; (2) the read backward operation performed to transfer data from a selected I/O device to descending-order address locations in main storage; and (3) the sense operation performed to transfer status information pertinent to the selected I/O device to main storage. (Channel operations for the read and sense commands are identical; thus, the description of channel read operations in this chapter pertains to both the read and sense commands.)

Channel write-type operations include: (1) the write operation performed to transfer data from main storage to a selected I/O device; and (2) the control operation performed to initiate an operation at a selected I/O device that does not involve transfer of data.

Channel start I/O operations begin upon receipt of a 'start I/O' signal, device (unit) address byte, and the 'select channel' signal from the CPU interface. Providing the channel is prepared to handle the Start I/O instruction, it performs an initial selection routine, during which the CAW is fetched from main storage, the I/O device specified by the device address is selected (if available for operation), the first CCW is fetched from main storage, and the CPU is released to perform other operations. With a successful initial selection routine complete, the channel controls the read-type or write-type operations without the necessity for further control by the CPU.

Channel operations described in the following paragraphs as a result of receiving a start I/O Instruction are (1) the initial selection routine, (2) the read operation, (3) the write operation, (4) the write chain data (write CDA) operation, and (5) the read chain data (read CDA) operation. Descriptions are provided at both the simplified flow chart and detailed flow chart levels.

Initial Selection Routine

The initial selection routine for a Start I/O instruction prepares both the channel and the I/O device to perform the command(s) initiated by the Start I/O instruction. Basically, the channel performs parallel operations during the initial selection routine to: (1) determine that the I/O device specified by the Start I/O instruction is available to handle the required operation; and (2) fetch from storage the information words which ultimately specify the operation to be performed. Once the parallel operations are successfully completed, the channel checks to see that the proper I/O device has been selected, gates the command information to the selected I/O device, checks the control unit and I/O device status bits for a "go" condition, and then releases the CPU. At this point both the channel and the I/O device are ready to begin the operation initiated by the Start I/O instruction.

Diagram 5-1 is a simplified flow chart showing the basic operations performed during a successful initial selection routine. Each major operation indicated in Diagram 5-1 is further detailed in Diagram 5-2. The initial selection routine

for a start I/O operation is described first at the simplified level ("Simplified Initial Selection Routine") using Diagram 5-1 as the basis for the description, and then at the detailed level ("Detailed Initial Selection Routine") using Diagram 5-2 as the basis for the description.

Simplified Initial Selection Routine

- 'Start I/O' signal, 'select channel' signal, and unit address byte from CPU initiate initial selection routine.
- If channel available, channel initiates storage request to fetch CAW from address location 72; address is channel generated and gated to SAB upon receipt of 'BCU response' signal.
- When 'BCU response' signal for CAW fetch cycle falls and the 'accept' signal is received from the BCU interface, parallel operations to select I/O device begin.
- 'Setup' latch turned on; unit address byte gated to I/O interface and 'address out' signal activated.
- 'Select out' signal sent to I/O interface.
- If I/O device available, channel receives 'operational in' signal, unit address of I/O device, and 'address in' signal via I/O interface; completion of I/O device selection awaits completion of parallel CAW and CCW fetch operations.
- CAW received on SBO lines; channel checks for proper format; CAW contains CCW address and storage protect key.
- Channel initiates storage request to fetch CCW.
- 'BCU response' signal gates CCW address and storage protect key to BCU interface.
- CCW received on SBO lines; channel checks for proper format and gates to appropriate channel registers.
- 'CCW valid' trigger is turned on.
- Unit address from control unit compared with unit address from CPU to ensure correct device responded.
- Command byte and 'command out' signal sent to control unit.
- Control unit drops 'address in' signal; channel drops 'command out' signal.
- Control unit places status byte on 'bus in' lines and raises 'status in' signal; channel checks for all 0's status byte (control unit and I/O device ready).
- 'Sequence 2' latch turns on; 'release' signal and CC=0 sent to CPU; initial selection routine complete.

Assume that the channel is polling its attached control units (propagating the 'select out' signal) for a pending interrupt condition at any control units (see Diagram 5-1). When the channel is polling, it is not engaged in handling a CPU instruction or an interrupt condition, and is therefore free to accept a CPU instruction.

With the channel polling, the arrival from the CPU interface of a 'start I/O' signal, 'select channel' signal, and the unit address byte associated with the Start I/O instruction, breaks the channel polling routine and causes the channel to begin the initial selection routine. During the initial selection routine the channel first performs two parallel operations: (1) fetches first the CAW and then the CCW from main storage; (2) attempts to select the I/O device specified by the unit address from the CPU interface. The parallel operations are interdependent at various points in the routine and, where this is so, it is indicated in Diagram 5-1.

When the channel polling routine is interrupted, a 'storage request' signal is initiated by the channel to fetch the CAW from storage address Z+1 (72 decimal or 48 hex). The CAW storage address is forced by the channel and is always address 72 decimal. When the 'storage request' signal is acknowledged, the channel receives a 'BCU response' signal from the BCU interface requesting the channel to gate the CAW address to the storage address bus (SAB). When the 'BCU response' signal falls and the 'accept' signal is received from the BCU interface, the parallel I/O device setup routine (Diagram 5-1) begins with the turn-on of the 'setup' latch. With the 'setup' latch on, the unit address byte, originally obtained from the CPU interface and gated into the channel's unit address register, is gated on the 'bus out' lines to all control units attached to the channel. After a significant delay to allow the unit address bits to stabilize, the channel sends an 'address out' signal to all control units signifying a unit address byte is on the 'bus out' lines. The channel then turns on the 'select out' latch to propogate the 'select out' signal through the attached control units. When the 'select out' signal reaches the control unit which has the addressed I/O device attached, the control unit sends the channel an 'operational in' signal. The 'operational in' signal indicates that an I/O device is attached for operation. The control unit then sends the channel an 'address in' signal signifying that the unit address byte is present on the 'bus in' lines. At this point, the parallel I/O device setup routine remains in a "wait" state until the parallel CAW and CCW fetch

routines are successfully completed (as indicated by turn-on of the 'CCW valid' trigger).

Recall that the parallel CAW and CCW fetch operations started when the channel initiated a storage request for the CAW, the storage request was subsequently acknowledged by receipt of a 'BCU response' signal, and the CAW address (72 decimal) was gated to the SAB. When the CAW is available to the channel on the SBO lines from main storage, it is gated into the channel and checked for proper formatting. The CAW contains the storage address of the CCW and the 'storage protect key' bits required for access to the CCW storage location. (These 'storage protect key' bits are also valid for access to storage locations specified by the CCW after completion of the initial selection routine.) With an error-free CAW in the channel, the channel initiates a second 'storage request' signal to obtain the CCW. When the 'storage request' signal is acknowledged (by the 'BCU response' signal), the channel gates the CCW address and storage protect key obtained from the CAW to the SAB and 'storage protection' lines, respectively. When the CCW is available to the channel on the SBO lines, the CCW is gated into the channel and checked for proper formatting. If formatting is proper, the 'CCW valid' trigger is turned on signifying successful completion of the CCW fetch cycle. At this point in the initial selection routine, the parallel 'initial selection routine' operations are complete, and the remaining operations are serial in nature (see Diagram 5-1).

With the 'CCW valid' trigger on and the 'address in' signal active, the channel gates the unit address byte from the I/O interface 'bus in' lines into the channel and compares this unit address with the unit address originally sent to the control unit. A successful comparison indicates that the proper I/O device has been selected and is operationally connected to the channel. With a successful selection indicated, the channel 'setup' latch is turned off, and the command byte received in the first CCW is gated from the command register to the I/O interface 'bus out' lines. After a significant deskewing delay, the channel sends the control unit a 'command out' signal, indicating that command byte is on the 'bus out' lines and available to the control unit and I/O device for decoding. The control unit responds to the 'command out' signal by dropping the 'address in' signal, indicating that the command byte on the 'bus out' lines has been accepted. With the drop of the 'address in' signal, the channel drops the 'command out' signal.

Following the fall of the 'command out' signal, the control unit raises the 'status in' signal to the channel and places the control unit and I/O device status bits on the 'bus-in' lines to the channel. The 'status in' signal causes the channel to turn on the 'sequence 1' latch. With the 'sequence 1' latch on, the channel performs an 'all zeros' check on the status bits received from the control unit. If the status bits are all 0's (control unit and I/O device ready to perform the received command and operationally connected to the channel), the channel 'sequence 2' latch is turned on. This causes the channel to generate an internal 'accept' signal indicating that the channel is capable of performing the Start I/O instruction originally received via the CPU interface. Following generation of the 'accept' signal, the channel sends a 'release' signal and a condition code of 0 to the CPU interface. The 'release' signal causes the CPU to disconnect from the channel, permitting parallel CPU and channel operations to proceed independently. Condition code 0 is stored in the CPU's current PSW for future reference.

With the 'release' signal sent to the CPU, the channel initial selection routine is complete and the channel is ready to independently control read-type or write-type operations. To determine the type of operation to be performed, the channel decodes specified bits of the command received in the CCW. See Diagram 5-1 for reference to simplified read-type and write-type operation flow charts.

Detailed Initial Selection Routine

The initial selection routine for a Start I/O instruction is described in detail in the following five paragraphs: "Channel Polling and Start I/O Instruction Handling", "CAW Fetch Operation", "CCW Fetch Operation", "Parallel I/O Device Setup" and "Setup Completion". Throughout the text in the above cited paragraphs, reference is made to Diagram 5-2, FEDM.

Channel Polling and Start I/O Instruction Handling. A summary and a detailed description of channel polling and start I/O operations follow:

- Channel polling operations initiated when channel not engaged in operation, interrupt not pending and 'select out' signal is activated.
- Receipt of 'select in' signal indicates no attached I/O device has interrupt pending.
- Receipt of 'operational in' and 'address in' signals indicates attached I/O device has pending interrupt; channel stacks the status byte and sends an 'interrupt response' signal to the CPU interface.
- With channel free and no interrupt pending, receipt of 'start I/O' and 'select channel' signals causes 'start I/O' latch to turn on.
- 'Select in SS' signal turns off 'select out' latch.

- 'CAW fetch' and 'CCW fetch' latches turned on by 'start I/O' latch and 'select in' or 'polling interrupt request' signals.

- Unit address from CPU interface 'unit address bus out' lines gated into unit address register.

- Command address register reset.

Assume that the channel is not engaged in performing an operation initiated by an instruction from the CPU interface and is not handling an interrupt condition. When this is the case, the channel initiates a polling operation to determine if any attached device has an interrupt pending (due to status presented as a result of ending a previous operation initiated by the channel or, in some cases, due to status presented as a result of some unusual condition. For example, when some I/O devices are entered into the system by a power-on operation, the device requests an interrupt to present status.)

Polling is initiated when the channel 'select out' latch is turned on (Diagram 5-2). The resulting 'select out' signal is propagated through the attached control units in a predetermined sequence. That is, the 'select out' signal is presented first to control unit 1, then to control units 2 through 8 (maximum control unit configuration) in that order (see Figure 3-1). If no control unit has an interrupt pending, the channel receives a 'select in' signal from the last control unit in the chain to turn off the 'select out' latch. (The 'select in' signal is actually the 'select out' signal after propagation through the attached control units.) Assuming that the channel is still free (not working), and the channels 'start I/O', 'halt I/O' and 'test I/O' latches are not on, the 'select in' signal falls when the fall of the 'select out' signal has propogated through the attached control units. To ensure that sufficient time is allowed for the 'select in' signal to fall, the turn-on of the 'select out' latch for the next polling operation is held off by a 'select in singleshot' signal of approximately 2.2 usec in duration. After the singleshot times out, the 'select out' latch is again turned on to perform another polling operation. The channel "polling loop" described above continues as long as no control unit contains an interrupt condition and no operation is initiated from the CPU interface. If a control unit interrupt condition is found pending during a polling operation, the channel receives an 'operational in' signal, unit address byte and 'address in' signal from the control unit to enter a polling interrupt routine. Whether or not a control unit interrupt condition is pending and a 'start I/O' signal is presented to the channel before the 'select in' signal arrives at the channel, the channel 'start I/O' latch is turned on; when the 'select in' signal arrives at the channel or a polling interrupt request is detected, the 'select out' latch is turned off as previously described.

With the 'start I/O' latch on, the channel "polling loop" is broken when the 'select in' signal turns off the 'select out' latch. With the 'start I/O' latch on, the unit address byte (supplied via the CPU interface) is gated from the 'unit address bus out' lines (eight bits plug parity) to the unit address register (Diagram 5-2). Also, with the 'start I/O' latch on, the 'select in' signal turns on the 'CAW fetch' and 'CCW fetch' latches and resets the command address register in preparation for channel operations performed when CAW data is fetched during the storage cycle that follows. At this point in the initial selection routine for a Start I/O instruction, parallel operations are initiated to: (1) fetch first the CAW (see "CAW Fetch Operation") and then the CCW (see "CCW Fetch Operation") from main storage, and (2) at a specified time during the CAW fetch cycle, select the I/O device specified by the unit address in the channels' unit address register (see " Parallel I/O Device Setup").

CAW Fetch Operation. A summary and a detailed description of the CAW fetch operation.

- 'Storage request' latch turned on; 'storage request' signal activated at BCU interface.

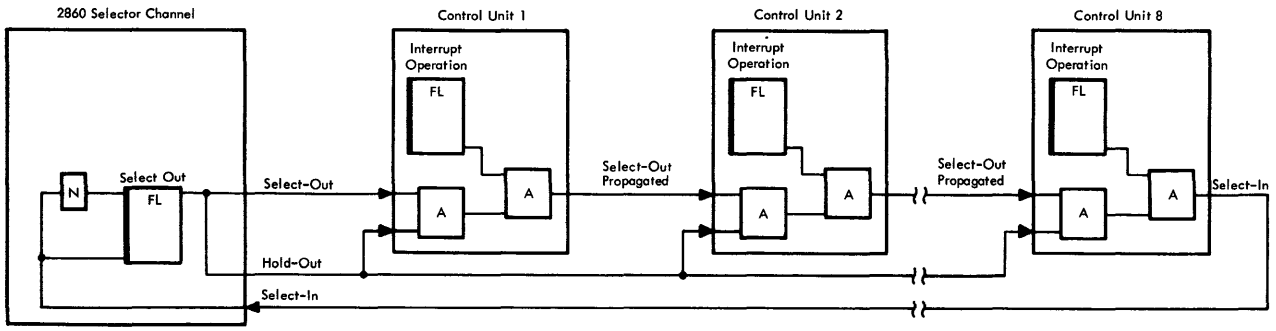


Figure 3-1. Channel Polling

- 'Storage cycle' trigger turned on.
- Channel receives 'BCU response' signal.
- Channel generated CAW address (72 decimal) gated to SAB; storage protect register contents gated to 'storage protection' lines.
- Channel receives 'accept' signal, indicating storage has been selected and the next 'BCU advance pulse' signal is for channel use.
- Channel receives 'BCU advance pulse' signal.
- Delayed 'BCU advance pulse' signal:
  1. Gates CAW bits 8-31 (+3P) from SBO lines into data address register.
  2. Gates CAW bits 0-7 (+P) from SBO lines into storage protect register.
  3. Gates zero check of CAW bits 4-7 and 29-31; if these bits are not all 0's, program check condition terminates operation.
- If CAW is valid (no program check condition detected), 'TIC' latch is turned on.

With both the channel's 'CAW fetch' and 'CCW fetch' latches on, the 'storage request' latch is turned on and a 'storage request' signal is sent to the BCU interface. (Model 2860 Selector Channels used with the System/360, Model 91, supply a 'control word request' signal to the BCU interface following the 'storage request' signal to the BCU interface, indicating that the request is for a control word.) This 'storage request signal is issued to request a storage cycle to obtain the CAW. The 'storage request' signal also turns on the 'storage cycle' trigger. The output of this trigger activates and inhibits applicable channel circuitry for the CAW fetch cycle. With the "CCW fetch' latch on and the 'TIC cycle' latch off, the channel awaits the 'BCU response' signal from the BCU interface. The arrival of the 'BCU response' signal signifies that the storage cycle previously requested has been honored and that the CAW address from the channel is required on the SAB. Since this is a CAW fetch cycle, the arrival of the 'BCU response' signal causes the channel to force address 72 decimal (48 hex) to the SAB.

Note: Since this is a CAW fetch cycle, the command address register contains no valid address (was previously reset to all zeros) and the command address update operation is of no significance on this storage cycle.

With address 72 decimal gated to the SAB, the channel waits for the 'BCU response' signal to drop indicating that the address is no longer required on the SAB. Once the 'accept' signal is received, the channel then awaits a 'BCU advance pulse' signal, signifying that the CAW is forthcoming on the BCU interface SBO lines. When the 'BCU advance pulse' signal arrives, the channel delays the signal to allow time for the CAW to arrive and stabilize on the SBO lines. The delayed 'BCU advance pulse' signal (designated 'raw advance singleshot') gates SBO bits 8 through 31 (plus 3 P bits) into the data address register and SBO bits 0 through 7 (plus 1 P bit) into the storage protect register. The data address entered into the data address register designates the storage location of the first CCW for the start I/O operation. The bits entered into positions 0 through 3 of the storage protect register, when gated to storage during the CCW fetch cycle, permit access to the storage location designated by the address in the data address register. As the CAW is gated into the channel, bits 4 through 7 and 29 through 31 are checked for all 0's (Diagram 5-2). (The results of the check are gated by the 'turn on TIC' signal, which, in turn, is generated by the delayed 'advance' signal from the CAW storage cycle.) If the checked bits are other than all 0's, a program violation exists, and a program check condition is detected by the channel. A program check condition results in the termination of the operation in progress. Assuming that the checked bits are all 0's and the storage protect register parity is odd, the CAW fetch cycle has been successfully completed. With the 'start I/O' latch still on, the TIC latch is turned on by the 'turn on TIC' signal to begin the CCW fetch operation. (See Diagram 5-2.)

CCW Fetch Operation. A summary and a detailed description of the CCW fetch operation follow:

- 'TIC' latch is turned on; 'storage request' latch turns on.
- Delayed 'advance' signal from CAW fetch cycle turns on 'TIC cycle' latch.
- Data address register contents plus one (eight bytes) gated to adder.
- Channel receives 'BCU response' signal.
- 'TIC' latch turns off; address in data address register gated to SAB and key in storage protect register gated to 'storage protection' lines.
- Channel sends 'address valid' signal to BCU interface.
- Channel receives 'BCU data request' signal followed by 'accept' signal from BCU interface.
- 'BCU response' signal falls and triggers 'remember BCU response SS' signal.

- 'Remember BCU response' latch turns on.
- Adder is latched.
- Channel receives 'BCU advance pulse' signal.
- Adder output gated to command address register.
- Delayed 'advance' signal gates CCW to channel registers:
  1. SBO bits 0-7 (+P) to command register.
  2. SBO bits 8-31 (+3P) to data address register.
  3. SBO bits 32-39 (+P) to flag register.
  4. SBO bits 48-63 (+2P) to count register.
- SBO bits 48-63 and 37-39 samples for all 0's (no error); bits 48-63 sampled for other than all 0's (no error).
- If sampling detects error, program check condition terminates operation; otherwise operation conintues.
- 'CCW valid' latch turns on.
- 'TIC cycle' trigger and 'CCW fetch' latch turn off.

For the beginning of the first CCW fetch cycle, the 'TIC' latch is turned on (even though an actual TIC operation is not indicated) for four primary reasons: (1) to block zero checking of insignificant CAW bits while the CAW bits are still on the SBO lines; (2) to inhibit the turn-on of the 'CCW valid' trigger by the delayed 'late advance pulse' signal from the CAW fetch cycle; (3) to turn on the channels 'storage request' latch which (in the absence of a program check condition due to CAW zero checking) initiates a 'storage request' signal to the BCU interface for the first CCW; and (4) to permit the channel to use the address in the data address register (rather than the command address register) to obtain the CCW. (See Diagram 5-2.)

In the first case above, consider that different SBO bits are checked for a non-TIC CCW fetch cycle than are checked for a TIC CCW fetch cycle; also, for a TIC CCW fetch cycle, only bits zero checked during a CAW fetch cycle are zero checked. Thus, with the 'TIC' latch on to inhibit checking of bits normally zero checked during a non-TIC cycle, insignificant CAW bits still on the SBO are prevented from causing detection of a program check condition to erroneously terminate the operation.

When the delayed 'advance' signal from the previous CAW fetch cycle falls (Diagram 5-2), the 'TIC cycle' latch is turned on to update the address contained in the data address register. The updating operation consists of gating the address plus 1 value (equivalent to advancing the address by eight bytes) from the data address register to the adder. Later in the CCW fetch cycle operation, the updated address is gated from the adder to the command address register. If chaining is specified, the updated address in the command address register is used to fetch the next CCW.

Recall that a 'storage request' signal was supplied to the BCU interface with the turn-on of the 'storage request' latch. (Model 2860 Selector Channels used with the System/360, Model 91, supply a 'control word request' signal to the BCU interface following the 'storage request' signal. At the Model 91, the 'control word request' signal signifies that the storage request is for a CAW or CCW.) At this point, the channel awaits the receipt of a 'BCU response' signal from the BCU interface; this signal indicates that the storage request for the first CCW has been honored. When the 'BCU response' signal is received, the 'TIC' latch is turned off and the 'inhibit CCW valid' signal deactivated in that order. This prepares the channel to check the program validity of the CCW when it becomes available on the SBO lines. (CAW bits are no longer present on the SBO lines at this time.)

In addition, the 'BCU response' signal (Diagram 5-2) gates the CCW address (previously obtained during the CAW fetch cycle) from the data address register to the SAB and gates the storage protect key (also obtained during the CAW fetch cycle) from the storage protect register to the BCU interface 'storage protection' lines. After a deskew delay of between 10 and 30 ns to allow the address on the SAB to stabilize, the channel sends an 'address valid' signal to the BCU interface. At this point in the operation, the channel awaits receipt of the 'BCU advance pulse' signal indicating that the CCW is forthcomin on the SBO.

However, while awaiting the arrival of the CCW, the channel completes the data address register plus 1 update operation previously initiated when the data address register contents (plus 1) were gated to the adder. The updating operation is completed by first turning on the 'remember BCU response' singleshot (Diagram 5-2) when the 'BCU response' signal falls. The 'remember BCU response singleshot' signal has a duration of between 100 and 200 ns and allows time for the adder to perform the data address register plus one addition. With the turn-on of the 'remember BCU response' singleshot, the 'remember BCU response' latch is turned on which, in turn, latches the adder. Note that, in addition to an active 'remember BCU response' signal, the channel's 'accept' latch must be turned on to latch the adder. The 'accept' latch is turned on following receipt of the 'BCU response' signal by coincidence of a 'BCU data request' signal and 'accept' signal from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is not supplied via the BCU interface, but is generated in the channel by delaying the 'BCU response' signal approximately 150 ns.

(The 'BCU data request' signal is supplied to the channel via the BCU interface for all fetch and store cycles; the 'accept' signal is supplied to the channel to indicate that the requested storage cycle has actually begun.) With the adder latched, and following the timeout of the 'remember BCU response singleshot' pulse, the updated CCW address is gated to the command address register upon receipt of the 'BCU advance pulse' signal from the BCU interface. This completes the update CCW address operation, leaving the command address register with a CCW address specifying a storage location eight bytes higher than the address from which the incoming CCW was obtained.

With the timeout of the 'remember BCU response singleshot' signal complete and the channel-delayed 'advance' signal active, the first CCW is present on the BCU interface SBO lines and is gated into the channel. When the gating occurs, SBO bits 0 through 7 (plus 1 P) are gated into the command register, SBO bits 8 through 31 (plus 3 P) are gated into the data address register, SBO bits 32 through 39 (plus 1 P) are gated into the flag register and SBO bits 48 through 63 (plus 2 P) are gated into the count register.

Note: The 'storage protect key' bits gated into the storage protect register during the CAW fetch cycle are retained; i.e., gating of the CCW into the channel does not affect the storage protect register contents, since the key in this register is valid for subsequent storage accesses specified by the CCW.

When the CCW is gated, the channel performs three parallel operations (Diagram 5-2): (1) gates SBO bits for a CCW zero check condition; (2) samples command bits from the CCW for the presence of a TIC command; and (3) turns on the 'CCW valid' latch. For the first operation, the channel checks SBO bits 48 through 63 and 37 through 39 to determine if a program violation exists. If bits 48 through 63 do not equal zero and bits 37 through 39 do equal zero, no program violation exists. In the second case above, the channel samples command bits 4 through 7 for a TIC command, since a TIC command in the first CCW is a program violation. If a CCW program violation exists, either as a result of a zero check condition or detection of a TIC command, the 'program check' latch is turned on and the operation terminated by a 'sequence 5' ending routine; otherwise, the operation continues. In case 3 above, turn on the 'CCW valid' latch causes turn off of both the 'TIC cycle' trigger and 'CCW fetch' latch. With the 'CCW fetch' latch off, and an error-free CCW gated into the channel registers, the CCW fetch cycle has been successfully completed.

Parallel I/O Device Setup. A summary and a detailed description of 'parallel I/O device' operations follow:

- I/O device setup operation begins when 'not BCU response' and 'accept' signals from CAW fetch cycle present in channel.
- 'Setup' latch turns on.
- Unit address register contents gated to 'bus out' lines.
- 'Address out' latch turned on.
- Parity of unit address in 'bus out' latches checked; parity error terminates operation.
- With no parity error, 'select out' latch turned on.
- Receipt of 'select in' signal indicates unsuccessful selection; operation is terminated.
- On successful selection, 'operational in' signal received by channel.
- 'Address out' latch turns off.
- 'Address in' signal and unit address byte received by channel.

Selection of the I/O device specified by the Start I/O instruction begins when the 'BCU response' signal (received as a result of the 'storage request' signal issued for the CAW) drops (Diagram 5-2) and the 'accept' signal for the CAW storage cycle is present in the channel. With the above channel conditions present, the 'setup' latch is turned on and the 'CAW fetch' latch is turned off to begin the parallel I/O device setup operation. (For this operation, gating of the command address +1 value into the command address register is not significant, since the channel at this point contains no command address data.) With the 'setup' latch on, the unit address contained in the unit address register is gated to the 'bus out' latches for presentation to the attached control units on the I/O interface 'bus out' lines. In addition, the 'setup' signal turns on the channel clock. When the clock activates the 'T4' signal (and 'not T5' signal), the channels 'address out' latch is turned on. The resulting 'address out' signal is sent to all attached control units, notifying these units that a unit address is available on the 'bus out' lines for control unit decoding. At the same time the 'address out' latch is turned on, a 'check bus out parity' signal is generated to check the parity of the unit address in the 'bus out'

latches. If a parity error is detected, the 'channel control check' latch is turned on and the start I/O operation terminated by a 'sequence 5' ending routine. Assuming 'bus out' parity is correct, the 'select out' latch is turned on when the channel clock activates the 'T7' signal. The resulting 'select out' signal is sent to the I/O interface and sequentially propagated through the attached control units to determine if the I/O device specified by the unit address is available to the channel. If a 'select in' signal is returned to the channel, it indicates that the addressed I/O control unit is unavailable (off-line or not attached in the system). In this case, the channel's 'no selection' latch is turned on and the channel sends condition code 3 and the 'release' signal to the CPU interface.

If the address of the I/O device is properly decoded (recognized) by a control unit, the control unit responds by blocking the propagation of the 'select out' signal and sending an 'operational in' signal to the channel. The 'operational in' signal causes the channel's 'address out' latch to turn off. The control unit then sends the channel an 'address in' signal indicating that the unit address byte is on the 'bus in' lines.

When the CCW fetch cycle (parallel operation) has progressed to the point where the 'CCW valid' trigger is on, the channel's 'address in gated' signal is activated to complete the parallel I/O device setup, CAW fetch and CCW fetch operations. The remaining operations performed during the start I/O initial selection routine are serial in nature, and are described in the following paragraph.

Setup Completion. A summary and a detailed description of the setup completion operation follow:

- 'Bus in' unit address compared with 'unit address register' unit address:
  1. Unsuccessful comparison terminates channel operation (wrong I/O device responded).
  2. Successful comparison allows 'setup completion' operation to continue.
- 'Setup' latch turned off by 'T4' signal.
- 'Gate command' latch turns on.
- Channel decodes command address register bits for read-type or write-type operation.
- Command register bits gated to 'bus out' latches.
- DAB (data address register bits 21-23) and count register values gated to adder ('T0' clock time).
- DAB gated to byte count register ('T0' clock time).
- Summed DAB plus count value latched into adder and gated to count register ('T4' clock time).
- 'Bus out' parity checked for command byte parity error; error terminates operation and no error allows operation to continue.
- 'Command out' signal activated ('T7' clock time).
- 'Address in' signal to channel drops; channel deactivates 'command out' signal.
- Channel receives status byte and 'status in' signal.
- Status byte parity checked at 'bus in' latches.
- No 'bus in' parity error causes 'sequence 1' latch to turn on; clock turns off, then on.
- Count sampled (at 'T0' time); 'last word' trigger turns on if count less than nine.
- Status byte sampled for all 0's (no error); error causes termination of operation.
- 'Sequence 2' latch turns on ('T2' clock time).
- 'Release signal and CC equal 0 sent to CPU (channel releases CPU); 'setup completion' operation successfully completed.

Setup completion operations begin after the first CCW has been obtained from storage and the unit address from control unit has been received by the channel (Diagram 5-2).

With the unit address in the channel (via the 'bus in' latches), the channel compares this address with the address in the unit address register to ensure that the I/O device that actually responded during the parallel I/O device setup operation was the device specified. The results of the comparison are sampled after turn-on of the channel clock by coincidence of the 'operational in' and 'address in' signals from the I/O interface. If an address mismatch is detected, the channel's 'interface control check' latch is turned on by the 'T1' clock signal and the operation is terminated by a 'sequence 5' ending routine. If no mismatch is detected, the 'setup' latch and channel clock are turned off by the 'T4' signal from the channel clock.

With the 'setup' latch off, the 'gate command' latch is turned on. Turn-on of the 'gate command' latch causes the channel to perform the following operations: (1) channel command decoding logic (ALD LT113) is gated to determine if the command specified by command register

bits 5 through 7 is a read-type (read or sense) or a write-type (write or control) operation; and (2) the 'read or write' signal is activated by the 'gate command' latch causing the command bits in the command register to be gated to the control unit via the I/O interface 'bus out' latches. When the command bits are gated, the channel clock is turned on. In the first case above, appropriate channel control logic is set up for the read-type or write-type operation which follows the setup completion portion of the start I/O initial selection routine.

Note: A fifth operation, read backward, is not gated by the 'gate command' signal, but is decoded directly from the command register (bits 4, 6, and 7) to perform DAB complement operations at the byte counter and to set up the adder for decrementing data addresses. In this case, direct decoding of the Read Backward command is necessary due to the setup completion operations following the gating of the command to the 'bus out' latches.

Following gating of the command bits to the 'bus-out' latches and turn-on of the channel clock (Diagram 5-2), the clocks 'T0' signal gates the three least significant bits (bits 21 through 23, referred to as the DAB) of the data address register plus the contents of the count register to the adder. At the adder the two values are summed. When the 'T4' clock signal is generated, the 'DAB plus count' value is latched into the adder and then gated back into the count register. The 'DAB plus count' value (now in the count register) is used by channel logic to identify the byte location of the last byte of the last doubleword specified by the CCW. Figure 3-2, A illustrates a data transfer on a byte basis and a successful ending with the last data byte properly located as a result of adding the DAB and count. Figure 3-2, B is a hypothetical situation showing the error that would result if the DAB and count were not added.

At the same time that the DAB and count are gated to the adder, the DAB is gated to the byte counter (Diagram 5-2). As previously stated, the DAB identifies the byte location of the first byte of the first doubleword to be transferred. For a detailed description of the byte counter operation refer to "Byte Counter Circuits" in Chapter 2.

When the channel clock generates the 'T2' signal, the command byte in the 'bus out' latches is checked for odd parity. If 'bus out' even parity is detected, or an adder parity error was detected during the 'DAB plus count' operation, the channel's 'channel control check' latch is turned on to activate the 'machine check' signal; this causes termination of the operation by a sequence 5 ending routine. If no error exists, the 'command out' signal is activated by the 'T7' clock signal. The 'command out' signal is sent to the control units via the I/O interface, causing the selected control unit to process the command on the 'bus out' lines. When the command is processed, the control unit drops the 'address in' signal to the channel. The fall of the 'address in' signal deactivates the channels 'command out' signal to the control units. When the 'command out' signal drops, the selected control unit sends the channel a 'status in' signal. Upon receipt of the 'status in' signal, the channel samples the status byte at the 'bus in' latches for correct parity. If parity is incorrect, the channel 'interface control check' latch is turned on and the operation terminated by a sequence 5 ending routine.

The 'status in' signal, in addition to causing the bus-in parity sample, turns off the channel clock and turns on the 'sequence 1' latch (Diagram 5-2). With the 'sequence 1' latch on, the 'status in' signal active, and no bus-in parity error, the channel clock is again turned on. The 'T0' clock signal latches the status byte into the 'bus in' latches and also causes the channel to sample the count value in the count register. If the count register count value (count plus the DAB) is less than nine, the channel's 'last word'

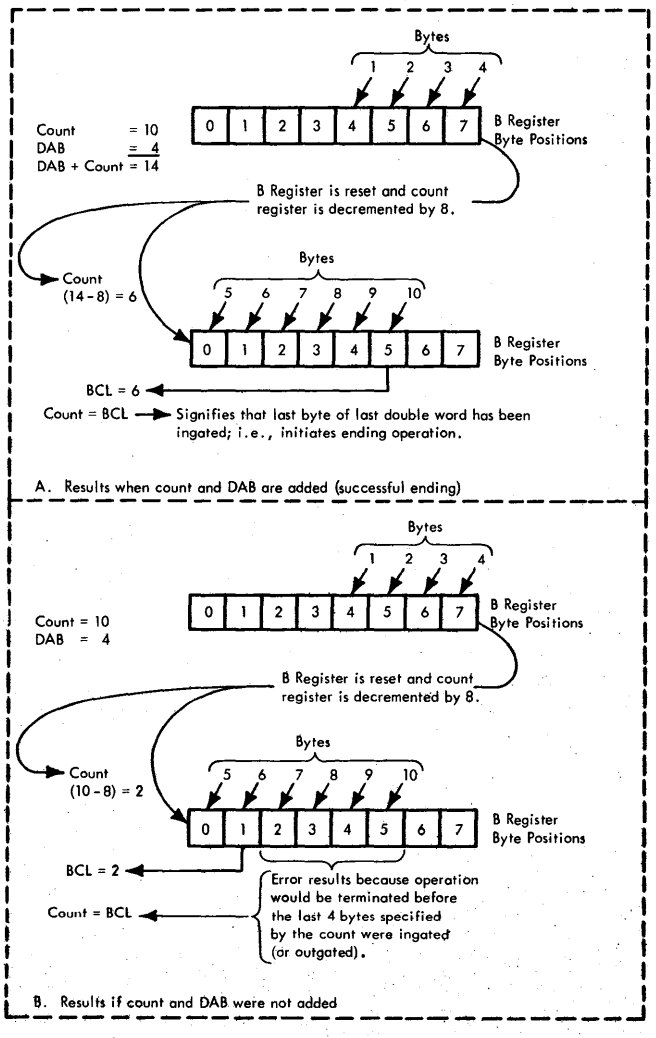


Figure 3-2. Example of Necessity for DAB + Count Operation

trigger is turned on, indicating that all data specified by the CCW count will be transferred after the first data doubleword is stored in main storage.

When the control unit status byte is latched into the 'bus in' latches, the channel examines the status bits for all 0's. If the status bits are not all 0's, (or a Halt I/O instruction has been received by the channel), the operation is terminated by a sequence 5 ending routine (Diagram 5-2). Otherwise, the channel's 'sequence 2' latch is turned on by the clock 'T2' signal, causing the channel to generate an internal 'accept' signal. In turn, the 'accept' signal generates a 'release' signal to the CPU interface. The 'release' signal, when accompanied by a condition code (two bits) of 0 indicates that the channel has successfully completed the initial selection routine and is ready to control read-type or write-type operations specified by the CCW command without further control by the CPU. Upon receipt of the 'release' signal and condition code, the CPU drops a 'start I/O' signal to the channel and stores the condition code in its PSW for future reference.

As previously indicated, with the setup completion portion of the initial selection routine completed, the channel begins the read-type or write-type operations specified by the command in the command register. Note that Diagram 5-2 references other FEMDM Diagrams for the read-type and write-type operations. Refer to "Read Operation", "Write Operation", "Read CDA Operation" and "Write CDA Operation" for a description of these operations.

Read Operation

- Read operations begin following setup completion operations; read command is in command register and 'sequence 2' latch is on.
- Channel receives data bytes (one at a time) from I/O interface; bytes are gated into B-register.
- BCL = 0 condition indicates B-register full condition.
- B-register contents gated to A-register.
- Channel initiates store operation to store A-register contents.
- Channel decrements count register contents by eight bytes.
- Channel increments data address register contents by eight bytes.
- Operations repeated to store data doublewords.
- 'Last word' trigger turned on when count less than 17.
- On last data doubleword, 'BC = CTB' indicates last byte of read operation in B-register.
- B-register to A-register, data storage and data address update operations performed.
- Channel enters sequence 5 routine to end operation.

The read operation is performed by the channel to control the transfer of data from an I/O device to main storage.

Note: The channel does not distinguish between read and sense commands. Therefore, channel read operations and sense operations are identical, even though the control unit and I/O device internal operations are different for the two commands.

The channel begins read operations when 'start I/O' and 'select channel' signals along with a unit address are received from the CPU interface, an initial selection routine is completed, and the command from the CCW specifies a read operation.

Basic operations performed by the channel for a read operation are shown in Figure 3-3. The following description is provided to introduce the basic read operations. The read operation is described in the following paragraphs at the simplified flow chart level ("Simplified Read Operation") and at the detailed flow chart level ("Detailed Read Operation"). The channel begins the read operation after completion of the setup completion portion of the initial selection routine (see "Setup Completion" in this Chapter). With a read command specified and the 'sequence 2' latch on, the channel enters the B-register data handling loop operation. During this operation the B-register is loaded with data from the I/O device one byte at a time until the byte count latches are equal to zero (BCL = 0); the BCL = 0 condition specifies that a data byte has been loaded into the last B-register byte position (B-register full condition).

With the B-register full, the channel performs parallel operations to: (1) gate the B-register contents to the A-register (B-register to A-register operation); store the data received by the A-register in main storage ('data storage' operation); and (3) decrement the count in the count register by eight bytes so that the count will indicate that the parallel storage operation has occurred (update count operation). At the completion of the B-register to A-register, data storage, and update count operations, the channel performs a data address update operation to advance the address in the data address register an amount equivalent to eight bytes. The data address update operation is performed to provide a sequential address location at which the next data doubleword is to be stored. With the completion of the data

address update operation, the first read sequence is complete and the channel may begin the next read sequence (Figure 3-3).

The channel continues to perform the operations previously described for each read sequence until the next to the last read sequence is detected (at which time the 'last word' trigger is turned on). For the last read sequence, the B-register data handling loop operation loads bytes from the I/O device into the B-register until a BC = CTB condition is detected; this indicates the last data byte specified by the original CCW count has been gated into the B-register. When the BC = CTB condition is detected, the channel performs the B-register to A-register operation, and data address update operation as previously described. Note that the update count operation is not performed since the count register has been reset to 0. Upon completion of the data address update operation, the channel performs a sequence 5 routine during which it: (1) receives a status byte from the attached control unit; and (2) disconnects the control unit from the I/O interface.

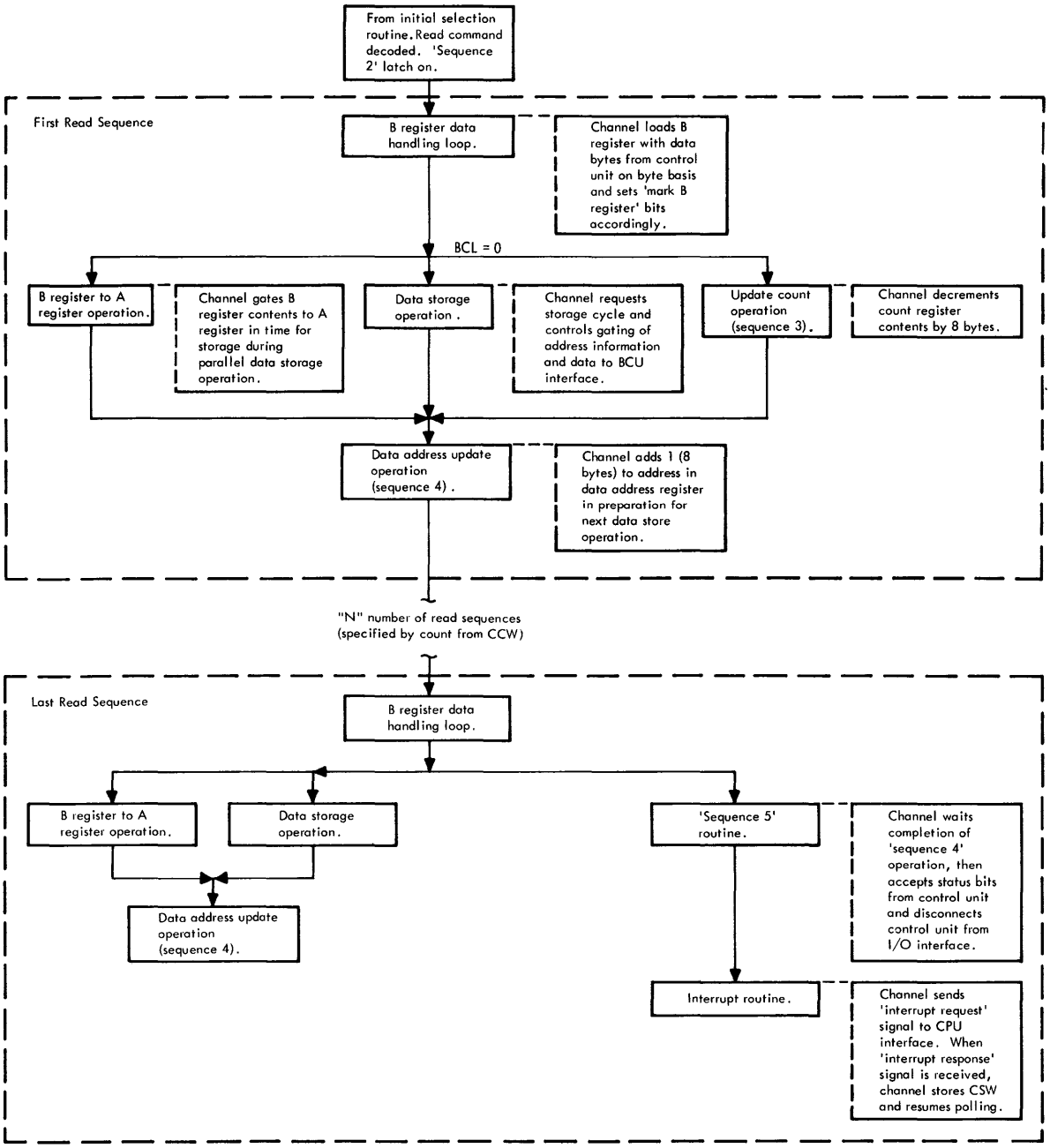


Figure 3-3. Basic Read Operations

Subsequent to the sequence 5 routine, the channel enters an interrupt routine during which the channel sends an 'interrupt request' signal to the CPU interface. When the 'interrupt request' signal is honored, the channel stores the channel status word at storage location 64, sends the unit address of the I/O device involved in the read operation to the CPU, and sends a 'release' signal to the CPU to resume channel polling operations.

Simplified Read Operations

The read operation is shown in simplified flow chart form in Diagram 5-4. This diagram illustrates a read operation in which three data storage cycles are required to transfer all data specified by the count contained in the count register. Entry into the read operation shown in Diagram 5-4 is from the simplified flow chart for the initial selection routine (Diagram 5-1).

At the end of the initial selection routine specifying a read operation, the 'sequence 1' and 'sequence 2' latches are on and the 'read' signal is active. Before the actual read operation can commence, the channel must perform initial selection routine cleanup operations. These operations are performed to clear the initial status-in byte and signal which were presented to the channel by the control unit at the end of the initial selection routine. To clear the initial status-in condition, the channel generates a 'service out' signal to the control unit, causing the control unit to drop the 'status in' signal and remove the status byte from the 'bus in' lines. With the drop of the 'status in' signal, the channel drops the 'service out' signal and turns off the 'sequence 1' latch.

At this point, the channel is conditioned to handle read operation data flow and enters the 'B-register data handling loop' operation (Diagram 5-4). When the control unit has a data byte available, it sends a 'service in' signal to the channel. The 'service in' signal causes the channel to generate a 'service out' signal which: (1) activates the 'service out' signal to the control unit (via the I/O interface) indicating that the channel accepts the data presented; (2) latches the data byte into the 'bus in' latches prior to the fall of the 'service in' signal from the control unit; (3) steps the byte counter to indicate receipt of the data byte; (4) gates the data byte into the B-register byte position indicated by the count in the byte count register. (B-register ingating occurs before stepping of the byte counter is completed, so that the data byte will be gated into the byte position specified by the byte counter prior to being stepped. See "Byte Counter Circuits" in Chapter 2); and (5) sets the mark-B register bit corresponding to the B-register byte position.

When the data byte is gated into the B-register, the 'BCL = 0' logic is gated to determine if the present data byte is being gated into the last B-register byte position. Assuming the BCL does not equal zero (and the 'service in' signal from control unit has dropped causing the channel's 'service out' signal to drop), the channel awaits another 'service in' signal from the control unit. When the 'service in' signal arrives, the described channel operations are again performed; the 'service in, service out' B-register loading loop continues until the 'BCL = 0' condition is detected to turn on the 'BCL = 0' latch. The resulting 'BCL = 0' signal turns on the 'B-register full' latch which inhibits the channel from responding to a 'service in' signal from the control unit; in effect, this stops the 'B-register data handling loop' operation (Diagram 5-4). In addition, the 'BCL = 0' signal initiates turn-on of the 'sequence 3' latch causing the channel to begin three parallel operations: B-register to A-register transfer; data storage; and update count. With the turn-on of the 'sequence 3' latch, the channel clock is turned on to time related events performed during the three parallel operations.

The 'B-register to A-register transfer' operations (Diagram 5-4) actually begin when the 'BCL = 0' condition is detected. The 'BCL = 0' signal generates a 'gate B-register to A-register' signal which gates data from the B-register to A-register on a byte position-to-byte position basis; the 'BCL = 0' signal also generates a 'gate mark-B to mark-A register' signal which gates bits from the mark-B register to the mark-A register. At clock 'T2' time, the 'A-register full' latch is turned on indicating that the data in the A-register is stable and available for gating to storage. At the same time, the 'BCL = 0' latch is turned off, the B-register is reset, and the 'B-register full' latch is turned off. With the turn-off of the 'B-register full' latch, the channel can again respond to 'service in' signals with a 'service out' signal; thus, the next B-register data handling loop operation may begin, even though the parallel data storage and update count operations are not complete. At this point in the 'B-register to A-register transfer' operation, the 'A-register full' latch remains on until the data is no longer required on the SBI lines (indicated by receipt of an 'accept' signal from the BCU interface during the 'data storage' operation).

The parallel 'data storage' operation (Diagram 5-4) begins with the turn-on of the 'storage request' latch at clock 'T0' time. This raises the 'storage request' signal to the BCU interface to request a storage cycle for the purpose of storing the data contained in the A-register. In addition, the 'storage request' signal turns on the 'storage cycle' trigger to inhibit and enable appropriate channel logic as required for the storage cycle. When the channel receives the 'BCU response' signal from the BCU interface ('storage request' signal has been honored), the channel: (1) gates data address register bits 0 through 20 to the BCU interface SAB lines to specify the storage location at which the A-register contents are to be stored;

(2) gates storage protect register bits 0 through 3 to the BCU interface 'storage protection' lines to permit access to the storage location specified by the data address on the SAB; (3) gates mark-A register bits to the BCU interface 'mark bits' lines to specify which data bytes in the A-register doubleword are to be stored; and (4) turns on the 'store' latch to send a 'store' signal to the BCU interface indicating that the purpose of the requested storage cycle is to store information.

Subsequently, the channel receives a 'BCU data request' signal from the BCU interface which causes the channel to gate the contents of the A-register to the BCU interface SBI lines.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is not supplied via the BCU interface, but is generated in the channel by delaying the 'BCU response' signal by approximately 150 ns.

When the storage cycle progresses past the point where A-register data is no longer required on the SAB, the channel receives an 'accept' signal (accompanied by the fall of the 'BCU data request' signal) via the BCU interface. This signal initiates the generation of a signal ('turn off A-register full') which turns off the 'A-register full' latch. Receipt of the 'BCU advance pulse' signal completes the data storage operation.

The parallel update count operation (Diagram 5-4) begins at the same time the 'data storage' operation begins. At clock 'T0' time, the count register contents are gated to the adder where the equivalent of eight bytes is subtracted from the count (count value minus 1). At clock 'T4' time, the adder is latched and the contents of the adder gated back to the count register. This completes the update count operation with the count register value now indicating that one data storage operation has occurred.

At clock 'T5' time, the channel's 'sequence 4' latch is turned on and the channel clock is turned off to begin the 'update data address' operation. Following turn-on of the 'sequence 4' latch, the 'sequence 3' latch is turned off and the channel clock is turned on (Diagram 5-4). With the 'sequence 4' latch on, the contents of the data address register are gated to the adder where the equivalent of eight bytes is added (data address plus 1). At clock 'T4' time, the adder is latched, and the adder contents gated back into the data address register. At clock 'T6' time, the 'sequence 4' latch is turned off to end the 'data address update' operation. With the operation complete, the data address register now contains the address location at which the next data doubleword is to be stored.

Recall that when the 'B-register full' latch was turned off during the 'B-register to A-register transfer' operation, that the 'B-register data handling' operation for the second data transfer began (Diagram 5-4). When the B-register is full, the three parallel operations (B-register to A-register transfer, data storage and count update) are again performed as previously described. However, since (in the Diagram 5-4 example) the second doubleword assembled in the B-register is the next to the last doubleword to be transferred to storage, the count in the count register indicates less than 17 bytes (equal to or less than two words). Before the updated 'count minus 1' value is latched into the adder, the count is sampled by channel logic and the 'last word' trigger is turned on. Note that the 'last word' trigger is turned on if the count is less than 17 bytes because the last word condition is sampled just prior to count updating. When the update count operation is completed, the 'sequence 4' latch is turned on and the update data address operation is performed.

In the meantime, the B-register data handling loop operation for the last data transfer is in progress (Diagram 5-4). With the 'last word' trigger on, the channel is monitoring data byte gating to the B-register for a BC = CTB condition. When this condition is detected, the last byte specified by the original count is being gated into the B-register. Detection of the BC = CTB condition breaks the data handling loop by: (1) turning on the BC = CTB trigger; (2) turning on the 'sequence 5' latch to initiate an ending operation; (3) resetting the count register to zero, indicating that all specified bytes for the read operation have been received (zero count for CSW store); and (4) turning on the 'B-register full' latch to indicate completion of data gating.

With the data handling loop for the last doubleword transfer complete, the B-register to A-register transfer operation, the data storage operation, and the update data address operation are performed as previously described. (For the B-register to A-register transfer operation, the 'BCL = CTB' latch is turned off when the 'A-register full' latch is turned on, rather than the 'BCL = 0' latch as in previous 'B-register to A-register transfer' operations.) When the update data address operation is complete, the 'sequence 4' latch is turned off.

With both the 'sequence 4' and 'sequence 3' latches off, the channel enters a 'sequence 5' ending routine (Diagram 5-4). With the 'sequence 5' latch on, the channel inhibits the generation of 'service out' signals in response to possible 'service in' signals from the control unit. This prevents further gating of data from the control unit into the B-register in the event the I/O device presents more data than originally specified by the count from the CCW. The channel then places all 0's on the 'bus out' lines to the control unit; in the event that the control unit does send the channel a 'service in' signal, the channel 'wrong length record' and 'command out' latches are turned on. With all 0's on the 'bus out' lines, and a 'command out' signal to the control unit, the control unit responds by dropping the 'service in' signal. If the channel is not presented with a 'service in' signal, the 'wrong length

record' and 'command out' latches are not turned on.

In either case, the channel awaits receipt of a 'status in' signal from the control unit, indicating that the control unit has placed a status byte (containing the channel end and, possibly, the control unit and device end bits) on the 'bus in' lines to the channel. Upon receipt of the 'status in' signal, the channel sends the control unit a 'service out' signal indicating acceptance of the status byte. Following activation of the 'service out' signal, the channel turns off the 'select out' latch, dropping the 'select out' signal to the control unit. The control unit responds by dropping the 'status in' and 'operational in' signals to disconnect the control unit from the channel. Subsequently, the channel turns on the 'interrupt' latch to send an 'interrupt request' signal to the CPU interface. With the 'interrupt' latch on, the 'sequence 5' and 'sequence 2' latches are turned off, indicating a successful control unit disconnect and entry into an interrupt routine.

During the interrupt routine (Diagram 5-4) the channel awaits an 'interrupt response' signal from the CPU. When the signal is received, the channel: (1) gates the unit address of the I/O device which handled the read operation to the CPU for entry into the PSW; (2) stores the channel status word; (3) performs channel cleanup operations [turns off the 'CCW valid' latch, the 'gate command out' (read or write) latch, and the 'storage cycle' latch], (4) gates condition code 0 and a 'release' signal to the CPU, and (5) resumes polling operations.

Detailed Read Operations

The detailed read operation descriptions in the following paragraphs are based upon Diagram 5-5. Separate descriptions are provided for the following operations: initial channel conditions; B-register data handling loop; B-register to A-register transfer; data storage; update count; update data address; and read operation ending sequence.

Initial Channel Conditions. Assume that a successful start I/O initial selection routine for a read operation has been completed, that the 'sequence 3, 4 and 5' latches are off, and that the 'B-register full' and 'A-register full' latches are off (Diagram 5-5). At 'T2' clock time, the channel sends a 'service out' signal to the control unit in response to the initial 'status in' signal presented during the initial selection routine and waits for the control unit to drop its 'status in' signal. When the 'status in' signal drops, the channel turns off the 'sequence 1' latch, deactivates the 'service out' signal and turns off the channel clock. At this point, the channel's 'sequence 2' latch is on, a read command has been decoded by the channel, and the channel is ready to begin handling data for the read operation.

B-Register Data Handling Loop. A summary and a detailed description of B-register data handling loop operations follow:

- Channel receives 'service in' signal from control unit.

- Channel activates 'service out' signal:
  1. Raises 'service out' signal to control unit.
  2. Causes data byte to be latched into 'bus-in' latches and parity of data byte to be checked.
  3. Causes byte counter to be updated.
  4. Causes data byte to be gated into B-register byte location specified by byte counter.
  5. Causes channel to sample for BCL = 0 and BC = CTB condition.

- BCL = 0 condition indicates B-register full condition:
  1. 'BCL = 0' trigger turns on.
  2. 'B-register full' latch turns on; prevents channel from activating 'service out' signal in response to 'service in' signal.
  3. 'Sequence 3' latch turned on.

- 'Last word' trigger turned on when count less than 17 bytes (two doublewords).

- BC = CTB condition indicates last byte of read operation has been gated to B-register:
  1. 'BC = CTB' trigger and latch turned on.
  2. Count register is reset.
  3. 'B-register full' latch turned on.
  4. 'Sequence 3' latch turned on.

Data handling for read operations begins when the control unit presents the channel with a 'service in' signal (Diagram 5-5) indicating that a data byte is present on the 'bus in' lines. Providing the channel is not performing a sequence 5 operation (due to a device initiated ending, or an ending sequence initiated by a check condition), the channel sends a 'service out' signal to the control unit indicating acceptance of the data byte. With the generation of the 'service out' signal, a 'latch bus-in' signal is generated to latch the data into the 'bus in' latches and a 'sampling' signal is generated for the bus-in data parity check circuit. The 'service out' signal also triggers a 130 ns singleshot which updates the byte counter and simultaneously gates the data in the 'bus in' latches to the B-register byte position specified by the byte counter. Since the byte counter output is not changed until the fall of the 130 ns signal, the data from the 'bus in' latches is gated into the B-register byte position specified by the byte counter prior to the update operation. (See "Byte Counter Circuits" in Chapter 2.) At the same time data is gated into the B-reg-

ister byte position, the mark-B register bit corresponding to the B-register byte position is set to a logic 1 level. (A logic 1 level mark bit indicates that data in the corresponding byte position is to be stored in main storage; if the bit is a logic 0, the corresponding data byte will not be stored.) The same signal used to gate the 'bus-in' data byte to the B-register also generates a signal to gate the 'byte count latches equal 0' and BC = CTB logic. If a BCL = 0 condition exists, it indicates that the B-register is full (a data byte has been loaded into the last byte position of the B-register). If a BC = CTB condition exists with the 'last word' trigger on, it indicates that the last data byte specified by the CCW count for the current read operation has been gated into the B-register.

For the present, assume that neither the BCL = 0 nor BC = CTB conditions exist. In this case, the channel awaits the drop of the 'service in' signal from the control unit. When the signal drops, the channel drops its 'service out' signal and waits for the next 'service in' signal from the control unit.

The 'service in, service out' exchange continues until either the BCL = 0 (without the 'last word' trigger turned on) condition is detected or the BC = CTB (with the 'last word' trigger turned on) condition is detected. Assume that the BCL = 0 condition is detected during the gating of a data byte into the B-register. This means, as previously described, that the data byte is being gated into the last byte position of the B-register: i.e., the B-register byte boundary has been reached. Detection of the BCL = 0 condition turns on the 'BCL = 0' trigger; in turn, the trigger output turns on the 'B-register full' latch, the 'sequence 3' latch, and the channel clock to complete the 'B-register data handling loop' operation ('service in, service out' exchange between the control unit and channel).

Assume that instead of a BCL = 0 condition, that the channel detects a BC = CTB condition with the 'last word' trigger on (see B-register data handling loop, last data transfer on Diagram 5-5). (The 'last word' trigger would have been turned on during the previous read sequence when a count of less than 17 bytes was detected in the count register.) As previously described, the BC = CTB condition indicates that the last byte of the read operation is being gated into the B-register and that the read operation will terminate after storage of the B-register data. Upon detection of the BC = CTB condition, the 'BC = CTB' trigger is turned on followed by turn-on of the 'BC = CTB' latch; the count register is then reset to all zeros with good parity (when stored as part of the CSW, the zero count indicates that all data bytes specified for the read operation were transferred). Turn-on of the 'BC = CTB' trigger also causes turn-on of the 'B-register full' latch and 'sequence 3' latch to complete the B-register data handling loop operation for the last data transfer of the read operation.

B-Register to A-register Transfer Operation. A summary and a detailed description of the B-register to A-register transfer operation follow:

- 'Read and BC equal 0' signal activated when 'A-register full' latch is off; turns on:
  1. 'Sequence 3' latch.
  2. Channel clock.
  3. 'B to A' latch.
  4. 'Mark-B to mark-A' latch.

- 'B to A' signal gates B-register contents to A-register.

- 'Mark-B to mark-A' signal gates mark-B register bits to mark-A register.

- 'A-register full' latch turns on, 'B-register full' latch turns off, 'BCL = 0' latch or 'BC = CTB' trigger turns off, and B-register is reset.

With the 'B-register full' latch on, the channel is prevented from responding to further 'service in' signals from the control unit (Diagram 5-5). In the absence of the 'A-register full' signal and the presence of a BCL = 0 condition, the 'BCL = 0' signal generates a 'read and BC equal 0' signal which turns on the 'sequence 3' latch and the channel clock. In addition, the 'read and BC equal 0' signal causes the 'B to A' latch and the 'mark-B to mark-A' latch to turn on. With the 'B to A' latch on, the data in the B-register is gated into the A-register. Turn-on of the 'mark-B to mark-A' latch gates the mark bits contained in the mark-B register to the mark-A register.

Note: If the 'A-register full' latch is on due to a previous read-operation storage request, the 'A-register full' condition indicates that main storage is not storing data at the channel data-transfer rate. In some cases, this may result in a 'status in' signal from the control unit when the channel is finally able to respond to the control unit's 'service in' signal with a 'service out' signal ('service out' signal response cannot occur until the 'B-register to A-register transfer' operation is complete). For example, assume the channel is servicing a high-speed drum which must have a response to its 'service in' signal within a specified period. If the channel is unable to respond within the specified time, the high-speed-drum control unit will, when the channel finally does send a 'service out' signal, respond with a status byte and 'status in' signal indicating an overrun condition at the I/O device.

To complete the 'B-register to A-register transfer' operation (Diagram 5-5), the 'A-register full' latch is turned on by the clock 'T2' signal. The same signal ('read, sequence 3, and T2') that turns on the 'A-register

full' latch also turns off the 'BCL = 0' latch (or 'BC = CTB' trigger), the 'B-register full' latch, and resets the B-register. If the B-register to A-register transfer operation is not the last transfer of the read operation (not BC = CTB) the reset of the 'B-register full' latch permits the channel to resume 'B-register data handling' operations (see B-register to A-register transfer, first transfer in Diagram 5-5). At this point, the B-register to A-register transfer operation is complete and further channel operations (data storage, update count, and update data address) are being performed.

Data Storage Operation. A summary and a detailed description of the data storage operation follow:

- 'Storage request' latch turns on; 'storage request' signal to BCU interface is activated.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal:
  1. Data address register bits 0-20 (+3P) gates to SAB.
  2. Mark-A register bits 0-7 (+P) gated to mark lines.
  3. Storage protect register bits 0-3 (+P) gated to 'storage protection' lines.
  4. 'Store' signal gated to BCU interface.
- Channel sends 'address valid' signal to BCU Interface.
- Channel receives 'BCU data request' signal; channel gates A-register contents to SBI lines.
- A-register byte parity generated and checked at SBI gating logic; detected parity error turns on' channel data check' latch.
- Channel receives 'accept' signal; 'accept' latch turns on.
- Fall of 'BCU response' signal turns on 'remember BCU response' latch.
- 'A-register full' latch turned off by 'remember BCU response latch' and 'accept latch' signals.
- Channel receives 'BCU advance pulse' signal.
- 'Accept', and 'remember BCU response' latches and 'storage cycle' trigger turns off.

Data storage operations (Diagram 5-5) begin with the turn-on of the 'storage request' latch by the 'T0' clock signal and subsequent turn-on of the 'storage cycle' trigger. With the 'storage request' latch on, a 'storage request' signal is sent to the BCU interface for the purpose of requesting a storage cycle to store data contained in the A-register. With the 'storage cycle' trigger on, applicable channel controls are activated or inhibited in preparation for a data transfer storage cycle.

With the 'storage request' signal active, the channel awaits receipt of the 'BCU response' signal. Upon receipt of the 'BCU response' signal, the channel gates the following to the BCU interface: (1) data address register bits 0 through 20 (+3P) to the SAB; (2) mark-A register bits 0 through 7 (+P) to the mark bits lines; (3) storage protect register bits 0 through 3 (+P) to the 'storage protection' lines. In addition, the 'BCU response' signal is gated by the 'gate store to BCU' signal (ALD MA111) to activate the 'store' signal to the BCU interface. When received in main storage, the data address on the SAB indicates the doubleword location at which data in the A-register is to be stored. Active mark bits specify that corresponding A-register bytes are to be stored at byte locations within the doubleword location specified by the data address. In storage units having storage protection facilities, the storage protect key permits access to the storage location specified by the data address. The 'store' signal indicates that main storage is to perform a storage operation.

With the address information and 'store' signal on the appropriate BCU interface lines, the channel awaits the receipt of a 'BCU data request' signal from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is generated within the channel by delaying the 'BCU response' signal by approximately 150 ns.

Upon receipt of the 'BCU data request' signal, the A-register contents are gated to the 'storage bus-in' lines and to a parity generator circuit where parity bits are generated for the eight A-register bytes. The generated parity bits are then gated to the SBI lines and are also compared with the eight parity bits originally contained in the A-register. Approximately 200 ns after the 'BCU data request' signal rises, the result of the parity comparison is strobed to determine if an SBI parity error exists. If a parity error does exist (generated parity bits do not match A-register parity bits), the 'SBI parity check' signal is activated and the 'channel data check' latch is turned on. For a read operation, the channel data check condition does not terminate read operations. However, the channel data check condition is indicated in the channel status word stored at the end of the read operation. If a chain command operation is indicated at the end of the read operations the channel data check condition prevents the channel from performing the chain command operation.)

With the A-register bits on the SBI, the channel awaits receipt of the 'accept' signal from the BCU interface. When received, the 'accept' signal (which indicates that

the requested storage cycle has begun) turns on the channel 'accept' latch. Also, the fall of the 'BCU response' signal turns on the channel 'remember BCU response' latch. In turn, the 'remember BCU response latch' and 'accept latch' signals turn off the 'A-register full' latch. Subsequently, the channel receives a 'BCU advance pulse' signal indicating that the data doubleword has been stored. In the channel, the 'BCU advance pulse' is delayed and causes turn-off of the 'accept', and 'remember BCU response' latches as well as the 'storage cycle' trigger. With these latches and trigger off, the data storage operation is complete.

Update Count Operation. The count in the count register is updated (decremented by an amount equivalent to eight bytes) for all read storage cycles except the last cycle of the read operation (Diagram 5-5). The last cycle is indicated by the on condition of the 'last word' trigger. Assuming the 'last word' trigger is off, the channel enters an update count operation when, after the clock is turned on at the end of the previous B-register data handling loop operation, the 'T0' clock signal is activated. With the activation of the 'T0' signal, the count register contents are gated to the adder where 1 (equivalent to eight bytes) is subtracted from the count (count minus 1 operation). Subsequently, channel logic is sampled by the 'T3' clock signal for a count register count of less than 17 bytes. If the count is less than 17, the 'last word' trigger is turned on to indicate that the next doubleword assembled in the B-register is the last of the current read operation. Note that the count is sampled by the 'T3' clock signal, which is prior to the time ('T4' clock signal) that the decremented count is latched into the adder and gated into the count register. Thus, the count is monitored for count of less than 17 bytes because the storage operation presently in progress is reflected in the count. In other words, the last word condition, when present, is detected just before the counter is decremented to reflect a count of eight or less bytes.

After the decremented count is gated into the count register, the 'T5' clock signal turns on the 'sequence 4' latch. At this point, the update count operation is complete and the update data address operation begins.

Update Data Address. Entry into the 'update data address' operation for a read operation may be due either to completion of the update count operation, or during the last transfer, indicated when the 'last word' and 'BC = CTB' triggers are on. In the first case, the update data address operation cannot begin until the update count operation is complete, since both operations require use of the adder. In the second case, ('last word' trigger on), the update count operation is not performed since the count register is reset to zero. In either case, the update data address operation is identical. With the 'sequence 3' latch on, the 'sequence 4' latch on, and the 'A-register full' latch off, (Diagram 5-5) the channel clock is turned off. Subsequently, the 'sequence 3' latch is turned off and the channel clock turned back on to time the 'update data address' operation. With the 'sequence 4' latch on, the data address in the data address register is gated to the adder (by the 'not T6' clock signal) where the equivalent of eight bytes is added to the data address (data address +1 operation). At clock 'T4' time, the adder is latched and the updated address is gated into the data address register. At clock 'T6' time, the 'sequence 4' latch is turned off indicating that the update data address operation is complete. Providing the update data address operation was not the last to be performed during the current read operation (not BC = CTB condition with 'last word' trigger on), the channel again enters the B-register data handling operation. If no further data is to be handled ('last word' trigger on and BC = CTB condition) and a chain data operation is not indicated, the 'BC = CTB' trigger is turned on. The active 'BC = CTB trigger' signal turns on the 'BC = CTB' latch, which in turn causes the 'sequence 5' latch to turn on. With the turn-on of the 'sequence 5' latch the data address update operation is complete and the channel is prepared to perform a read operation ending sequence.

Read Operation Ending Sequence. A summary and a detailed description of the read operation ending sequence follow:

- 'Service out' signal response to 'service in' signal inhibited.
- 'Stop' command (all 0's + P) gated to 'bus out' lines.
- Channel clock turns off.
- 'Sequence 4' latch turns off.
- If channel receives 'service in' signal:
  1. 'Wrong length record' latch turns on.
  2. Channel activates 'command out' signal.
  3. Control unit decodes Stop command and drops 'service in' signal.
  4. Channel drops 'command out' signal and awaits 'status in' signal.
- If channel does not receive 'service in' signal, it awaits 'status in' signal.
- Channel receives 'status in' signal:
  1. Channel clock turns on.
  2. Status byte is latched into 'bus in' latches.
  3. 'Status in end' latch turns on.

- Channel activates 'service out' signal; status byte latched into 'bus in' latches.
- Channel turns off 'select out' latch.
- Control unit drops 'status in' and 'operational in' signals.
- Channel 'status in end' latch turns off.
- Channel 'interrupt' latch turns on.
- 'Interrupt request' signal activated to CPU interface.
- Channel receives 'interrupt response' signal from CPU interface.
- Unit address register contents gated to 'unit address bus out' lines.
- 'Z address' latch and 'storage request' latch turn on.
- Channel receives 'BCU response' signal; address 64 gated to SAB, all mark bits to mark lines and 'store' signal to BCU interface.
- Channel receives 'BCU data request' signal; CSW gated to SBI lines.
- 'Accept' signal turns off 'interrupt' latch.
- 'BCU advance pulse' signal turns off 'CCW valid' latch; 'gate command' latch turns off and 'storage cycle' latch turns off.
- Channel sends 'release' signal and CC 0 to CPU interface.

The read operation ending sequence (sequence 5) begins when the 'sequence 4' latch is turned off at the end of the data address update operation (Diagram 5-5). With the 'sequence 5' latch on (turned on at the end of the last data handling loop operation) the channel is inhibited from generating a 'service out' signal in response to further 'service in' signals from the control unit. While waiting for turn-off of the 'sequence 4' latch, the channel gates all zeros with good parity to the control unit on the I/O interface 'bus out' lines. The all zeros bus-out bits represent a stop command to the I/O device when, if necessary, the channel must gate a 'command out' signal to the control unit later in the 'sequence 5' routine. At clock time 'T6' (clock was turned on for the update data address operation), the channel clock is turned off.

Assume that the 'sequence 3' and 'sequence 4' latches are turned off (Diagram 5-5) and that the 'select out' and 'operational in' signals are still active, indicating that the channel and control unit are still operationally connected. If the control unit has a 'service in' signal active, the channels 'wrong length record' latch is turned on and a 'command out' signal is gated to the control unit. With all 0's on the 'bus out' lines, the control unit responds to the 'command out' signal by dropping its 'service in' signal. With the drop of the 'service in' signal, the channel drops its 'command out' signal and awaits a 'status in' signal from the control unit.

If the control unit does not present a 'service in' signal to the channel, the 'wrong length record' latch is not turned on and the 'command out' signal is not activated; in this case, the channel waits for a 'status in' signal from the control unit. Receipt of the 'status in' signal indicates that a status byte is on the 'bus in' lines to the channel. The channel responds to the 'status in' signal by turning on the channel clock. When the 'T2' clock signal is activated, the status byte is latched into the 'bus-in' latches. Subsequently, the 'T3' clock signal turns on the 'status in end' latch (ALD EN111) to gate a 'service out' signal to the control unit. The 'service out' signal initiates generation of 'latch bus-in' signals, latching the status byte into the 'bus in' latches.

With the 'chain command' and 'IPL' latches off and the 'sequence 1 and sequence 2' signal inactive, the 'select out' latch is turned off by the 'T3' clock signal, causing the 'select out' signal to drop. The control unit responds to the drop of the 'select out' signal by dropping its 'status in' and 'operational in' signals. With the drop of these two I/O interface signals, the control unit is disconnected from the channel. The drop of the 'status in' signal turns off the 'status in end' latch which was originally turned on to gate the previous 'service out' signal. At 'T7' time, the channel clock is turned off. The resulting 'not T1' signal and channel end or device end bit from the status byte activate the 'turn on interrupt end' signal which turns on the 'interrupt' latch. With the 'interrupt' signal active, the 'sequence 5' and 'sequence 2' latches are turned off. At this point, the 'sequence 5' routine is complete. In other words, the status byte has been received from the control unit, the control unit has been disconnected from the I/O interface, and channel controls associated with I/O interface operations have been cleared. With the 'interrupt' latch on, the channel sends an 'interrupt request' signal to the CPU interface and enters an interrupt routine (Diagram 5-5). When an 'interrupt response' signal is received from the CPU interface, the I/O device address contained in the unit address register is gated to the 'unit address bus-in' lines for storage in the CPU's current PSW. The 'interrupt response' signal also turns on the 'Z-address' latch and causes a 'storage request' signal to be sent to the BCU interface. The 'Z-address' latch is turned on to enable the address for the channel status word (address 64) to be gated to the SAB when the 'storage request' signal is honored. When the 'BCU response' signal is supplied to the

channel: (1) the Z-address (generated internally by the channel) is gated to the SAB and a 'store' signal is sent to the BCU. In addition, all mark bits are activated and gated to the mark bit lines. This specifies that all eight bytes comprising the channel status word are to be stored. Upon receipt of the 'BCU data request' signal, the channel gates the channel status word to the SBI lines. The channel status word is degated from the SBI lines when the 'BCU data request' signal drops.

Note: For the 2860 Selector Channels used with System/360 Model 91, the 'BCU data request' signal is generated by delaying the 'BCU response' signal approximately 150 ns.

Subsequently, the receipt of an 'accept' signal from the BCU interface turns off the 'interrupt' latch. At this point, the channel waits for a 'BCU advance pulse' signal from the BCU interface. With the receipt of the 'BCU advance pulse' signal, the 'CCW valid' latch is turned off by the delayed 'advance' signal. The 'gate command' (read/write) latch is then turned off by the 'not CCW valid' signal, and the 'store cycle' trigger is turned off by the delayed 'late advance pulse' signal. Subsequently, the channel gates condition code 0 and a 'release' signal to the CPU interface. With the generation of the 'release' signal the interrupt routine for the read operation is complete and the channel resumes polling operations.

Read Operation Timing. A timing chart for a read operation (Diagram 5-6) shows the relationship of signals and operations performed as described in the preceding paragraphs. Diagram 5-6 shows the timing for the B-register data handling loop operations, the B-register to A-register transfer, data storage, update count, update data address, and sequence 5 operations described in previous paragraphs.

Write Operations

The write operation is performed by the channel to control the transfer of data from main storage to the I/O device. The channel begins write operations when a Start I/O instruction is received from the CPU interface, an initial selection routine is completed, and the command from the CCW specifies a write operation. To begin the write operation, the channel first fetches two doublewords from storage before transferring any data to the control unit. When the two doublewords are in channel registers, data transfers to the control unit begin on a byte-by-byte basis. As soon as the first doubleword transfer is completed, the channel fetches another doubleword from storage (if required by the CCW count) and begins transferring data bytes of the second doubleword to the control unit. This sequence of fetching a doubleword from storage each time one doubleword has been transferred to the control unit continues until the count indicates that all required data has been obtained from storage. As soon as all data contained in the channel is transferred to the I/O device, the channel ends the write operation by accepting a status byte from the control unit and performing an I/O interface disconnect operation. Subsequently, the channel sends an 'interrupt request' signal to the CPU interface to store the channel status word and provide the CPU interface with the unit (device) address and condition code 0.

Diagram 5-7 is a simplified flow chart showing the basic channel operations performed during a write operation. Operations shown on Diagram 5-7 are shown in greater detail in Diagram 5-8. The write operation is described first at the simplified level ("Simplified Write Operation") using Diagram 5-9 as the basis for the description, and then at the detailed level ("Detailed Write Operation") using Diagram 5-8 as the basis for the description.

Simplified Write Operation

- Write operation begins after initial selection routine is complete.

- Channel fetches first data doubleword.
  1. 'Storage request' latch and 'storage cycle' triggers turned on.
  2. 'BCU response' signal gates data address register contents and storage protect key to BCU interface.
  3. Channel receives 'BCU advance pulse' signal.
  4. Delayed 'BCU advance pulse' signal gates data doubleword from SBO lines to A-register; 'A-register full' latch turns on.
  5. A-register contents gated to B-register, 'A-register full' latch turns off, and 'B-register full' latch turns on.
  6. Data address register contents incremented by eight bytes.

- If 'last word' trigger off, channel fetches second data doubleword:
  1. Delayed 'BCU advance pulse' signal gates data doubleword into A-register.
  2. 'A-register full' latch turns on.

- Channel clears initial status byte:
  1. Raises 'service out' signal.
  2. 'Status in' signal drops; channel turns off 'latch status byte' trigger and drops 'service out' signal.

- Data from first data doubleword gated from B-register to control unit one byte at a time.

- BCL = 0 condition detected when all bytes from first doubleword are transferred; A-register contents transferred to B-register; 'A-register full' latch off.
- Channel decrements count register value by eight bytes.
- Channel increments data address register value by eight bytes; if 'last word' trigger not on, channel fetches next data doubleword.
- Data from second data doubleword gated from B-register to control unit; BCL = 0 detected when all bytes transferred.
- A-register contents transferred to B-register.
- Count register sampled; count of less than 17 turns on 'last word' trigger.
- Count register decremented; data address register value incremented.
- Data from last data doubleword gated to control unit; BC = CTB condition detected when last byte of write operation transferred.
- Channel ends operation with sequence 5 routine (receive status byte and perform I/O interface disconnect) and interrupt routine (store CSW, send unit address and CC = 0 to CPU).

Assume that a successful initial selection routine has been completed, that all sequence latches except the 'sequence 1' and 'sequence 2' latches are off, that the 'status in' signal from the control unit is still active, and a write command is specified (see Diagram 5-7). Also, assume that the count in the count register indicates that three doublewords must be fetched from storage to transfer all required data.

The write operation begins with the turn-on of the 'storage request' latch; this causes a 'storage request' signal to be sent to the BCU interface requesting a storage cycle to fetch the first data doubleword from storage. Also, with the 'storage request' latch on, the 'storage cycle' trigger is turned on to condition channel logic for the storage fetch operation in progress. When the 'storage request' signal is honored, the channel receives a 'BCU response' signal which causes address information for the first data doubleword to be gated to the BCU interface. Address information gating consists of: (1) gating bits 0 through 20 (+3P) from the data address register to the SAB; and (2) gating bits 0 through 3 (+P) from the storage protect register to the 'storage protection' lines. (Mark bits are not required for fetch storage cycle operations; however, the mark-P bit is activated.) When the storage cycle has progressed to the point where data is forthcoming on the SBO lines, the channel receives a 'BCU advance pulse' signal from the BCU interface. Since the 'BCU advance pulse' signal precedes the data on the SBO lines, the channel delays the signal approximately 150 ns and uses the delayed signal to gate data from the SBO lines into the A-register; with the gating of data into the A-register, the 'A-register full' latch is turned on. At this point, the first of the two required doublewords is in the channel.

Since a second doubleword must be fetched from storage, the channel's clock and 'sequence 4' latch are turned on: (1) to turn off the 'A-register full' latch and gate the data doubleword from the A-register to the B-register; and (2) to update the data address in data address register. In the first case, the 'A-register to B-register' transfer is performed to prepare the A-register to receive the second doubleword from storage. In the second case, the data address is updated eight bytes (data address register plus one) so that the data address register specifies the sequential address location of the second doubleword. After the data address is updated, the 'sequence 4' latch is turned off; the channel is now prepared to fetch the second data doubleword (Diagram 3-9).

Operations performed to fetch the second data doubleword are similar to those performed to fetch the first doubleword. Just prior to the turn-off of the 'sequence 4' latch, the channel turns on the 'storage request' latch and 'storage cycle' trigger as during the first fetch cycle. Upon receipt of the 'BCU response' signal from the BCU interface, the channel gates the address information (data address and storage protect key) to the BCU interface and awaits a 'BCU advance pulse' signal from the BCU interface. Upon receipt of the 'BCU advance pulse' signal, the second doubleword is gated from the SBO lines into the A-register in the same manner described for the first doubleword. With data in the A-register, the 'A-register full' latch is turned on.

With both the 'B-register full' and the 'A-register full' latches on, the channel begins operations to clear the initial 'status in' byte and signal presented by the control unit at the end of the initial selection routine. To clear the 'status in' condition, the channel sends a 'service out' signal to the control unit. The control unit responds by dropping the 'status in' signal. This causes the channel to: (1) turn off its 'sequence 1' latch; (2) turn off the 'latch status byte' trigger to remove the status byte from the 'bus in' latches; and (3) drop the 'service out' signal. With this sequence of events complete, the channel is ready to begin transferring data to the control unit.

Transfer of data from the B-register to the control unit ('data handling for write' operation) begins when the channel receives a 'service in' signal indicating the control unit is ready to receive a data byte. Upon receipt of the 'service in' signal, the channel gates a data byte from the

B-register byte position specified by the count in the byte counter. The data byte is gated through the channel's 'bus out' latches to the control unit on the 'bus out' lines. While the data byte is being gated from the B-register, the channel updates the byte counter and monitors the channel logic for a BCL = 0 condition. (A BCL = 0 condition signifies that the byte being transferred is from the last byte position of the B-register; i.e., all specified data from the first data doubleword has been transferred.)

The control unit, upon receipt of the data byte and 'service out' signal, drops the 'service in' signal, causing the channel to drop its 'service out' signal. Assuming that a BCL = 0 condition is not detected, the channel waits for the next 'service in' signal and again responds with the 'service out' signal to furnish another data byte to the control unit. The exchange of 'service in' and 'service out' signals continues until the BCL = 0 condition is detected.

When the BCL = 0 condition is detected, the 'A-register full' latch is turned off, the B-register is reset, and the second data doubleword is gated from the A-register to the B-register. Following the A- to B-register transfer operation, the count in the count register is decremented eight bytes; the decremented count value reflects the fact that all specified data in the first data doubleword has been transferred to the control unit.

When the count in the count register has been updated, the address in the data address register is updated (incremented) eight bytes (data address plus 1). At the completion of the update data address operation, the data address register contains the address of the sequential storage location from which the third data doubleword is to be fetched.

With the data address updated, the channel initiates operations to fetch the third doubleword from storage (Diagram.5-7). The fetch operation begins with the turn-on of the 'storage request' latch, followed by turn-on of the 'storage cycle' trigger. Upon receipt of the 'BCU response' signal, the channel gates the address information to the BCU interface and waits for a 'BCU advance pulse' signal from the BCU interface. When the 'BCU advance pulse' signal is received, it is delayed as previously described until data is available on the SBO lines. The delayed signal ('raw advance SS') causes the channel to gate the data doubleword from the SBO lines into the A-register and to turn on the 'A-register full' latch (Diagram 5-7).

When the storage request for the third doubleword is initiated, the channel begins transferring the second data doubleword from the B-register into the control unit. Data is transferred on a byte-by-byte basis in the manner described for transfer of data bytes from the first doubleword ('service in'/'service out' exchange between control unit and channel) until all eight data bytes are transferred: i.e., BCL = 0 condition is detected.

When the BCL = 0 condition is detected, the channel: (1) turns off the 'A-register full' latch, (2) resets the B-register and 'B-register full' latch, (3) transfers the third data doubleword from the A-register to the B-register, and (4) turns on the 'B-register full' latch. Just prior to decrementing the count register (so that the count register count will indicate that the second doubleword has been transferred to the control unit), the channel samples the count register value for a count of 16 or less bytes. Since, for the write operation illustrated by Diagram 5-7 the count does indicate 16 or less bytes, the 'last word' trigger is turned on, indicating that the third doubleword (now in the B-register) is the last to be transferred for the write operation. After the 'last word' trigger is turned on, the count in the count register is decremented eight bytes.

Upon completion of the count register updating, the data address in the data address register is updated by adding an amount equivalent to eight bytes. Since no further data is to be fetched for the write operation, the updated data address is available only for gating to storage in the event a logout operation should be initiated prior to the end of the write operation. In this case, the updated storage address indicates the storage access extent of the write operation.

When the 'B-register full' latch is turned on (prior to completion of the update data address operation), the channel begins transferring the third doubleword to the control unit on a byte basis as previously described for transfer of the first and second doublewords. As each data byte is transferred to the control unit, the channel (because the 'last word' trigger is on) samples for a BC = CTB condition. A BC = CTB condition signifies that the byte being transferred is the last byte to be sent to the control unit for the write operation.

After the BC = CTB condition is detected and transfer of the last data byte is complete ('service in' signal drops), the count register is reset to all zeros and the 'sequence 5' latch is turned on. The count register is reset to zero so that, when stored in the CSW at the end of the write operation, the count indicates that all specified bytes were transferred.

With the 'sequence 5' latch on, the channel enters a sequence 5 ending routine to obtain status from the control unit and to disconnect the control unit from the I/O interface. Status is received when the control unit raises the 'status in' signal and places a status byte on the 'bus in' lines to the channel. The channel responds by latching the status byte into the 'bus in' latches and sending a 'service out' signal to the control unit. Subsequently, the channel drops the 'select out' signal which was raised when the I/O device was selected during the initial selection routine. The control unit responds to the drop of the 'select out' signal by dropping the 'status in' and 'operational-in' signals. With the drop of these two signals, the

control unit is disconnected from the I/O interface.

The channel then enters an interrupt routine to store the CSW and to present the unit address of the device involved in the write operation to the CPU interface for entry into the I/O old PSW. When the channel receives an 'interrupt response' signal from the CPU interface, it gates the unit address register contents to the CPU interface 'unit address bus-in' lines. Subsequently, the channel sends a 'storage request' signal to the BCU interface. When the channel receives a 'BCU response' signal, it forces the Z-address (64 decimal) to the SAB and waits for a 'BCU data request' signal from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the channel generates the 'BCU data request' signal by delaying the 'BCU data response' signal by approximately 150 ns.

When the 'BCU data request' signal is received, the channel gates the CSW to the SBI lines and waits for a 'BCU advance pulse' signal from the BCU interface. Upon receipt of the signal, the channel performs cleanup operations, sends condition code 0 and a 'release' signal to the CPU interface, and resumes polling operations.

Detailed Write Operation

The detailed write operation descriptions in the following paragraphs are based upon Diagram 5-10.

Initial Channel Conditions. Assume that a successful start I/O initial selection routine for a write operation has been completed (Diagram 5-8), that the 'sequence 3, 4, and 5' latches are off, and that the 'last word' trigger and 'B-register full' and 'A-register full' latches are off. In addition, the 'status in' signal to the channel is active. The 'sequence 1' and 'sequence 2' latches are on and the channel has decoded a write command (bit 7 of the command register is a logic 1 level). The channel clock is on and has generated the 'T2' clock signal.

Fetch First Data Doubleword. A summary and a detailed description of the fetch first data doubleword operation follow:

- 'Storage request' latch turns on; 'storage request' signal to BCU interface activated.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Data address bits 0-20 (+3P) gated to SAB; storage protect register bits 0-3 (+P) gated to 'storage protection' lines.
- Channel receives 'BCU advance pulse' signal.
- 'Raw advance SS' and 'write' signals activate 'gate SBO to A write' signal.
- 'A-register full' latch turns on; 'SBO to A-register' trigger turns on.
- Data doubleword gated from SBO lines to A-register.
- 'Sequence 4' latch turns on; clock turns on.
- 'A-register full' latch turns off.
- Data address register contents plus 1 (eight bytes) gated to adder.
- Adder latched; incremented data address gated from adder to data address register.
- A-register contents gated to B-register; 'B-register full' latch turns on.

Before data transfer operations to the control unit can begin, the channel must fetch two doublewords from storage. (Diagram 5-8 assumes that a total of three doublewords must be fetched to perform the entire write operation.) This paragraph describes operations performed to fetch the first data doubleword.

With the 'write' signal active, the 'storage request' latch is turned on by the 'T3' clock signal to send a 'storage request' signal to the BCU interface (Diagram 5-8). Turn-on of the 'storage request' latch also causes the 'storage cycle' trigger to turn on and activate and inhibit applicable channel logic for the store operation.

When the channel receives a 'BCU response' signal, indicating that the 'storage request' signal has been honored, the data address register contents (bits 0 through 20 plus 3P) are gated to the SAB, and storage protect register bits 0 through 3 (plus parity) are gated to the 'storage protection' lines. The channel then awaits receipt of the 'BCU advance pulse' signal from the BCU interface, signifying that the first data doubleword will be on the SBO lines in approximately 200 ns. The 'BCU advance pulse' signal is delayed within the channel to generate a 'raw advance singleshot' signal which is active while the data is on the SBO lines. This signal activates the 'gate SBO to A-write' signal which turns on the 'A-register full' latch. In addition,

the 'gate SBO to A-write' signal turns on the 'SBO to A-register' trigger; the output of this trigger gates the data doubleword from the SBO lines into the A-register.

Recall that the 'sequence 1' latch is still on (Diagram 5-8); at this stage in the write operation, the on condition of the 'sequence 1' latch indicates that the second doubleword has not yet been fetched into the channel and that further channel operations must be performed before the second doubleword can be obtained.

With the 'B-register full' latch off and the 'A-register full' latch on, the 'sequence 4' latch is turned on. When the 'sequence 4' latch is on, the 'A-register full' latch is turned off and the channel clock is turned on. The clock is turned on to time an update data address operation and to time the transfer of the A-register data to the B-register.

Before the second doubleword can be fetched from storage, the address in the data address register must be incremented by an amount equivalent to eight bytes; otherwise, the same address location from which the first doubleword was obtained would again be accessed to obtain duplicate data for the second doubleword. Update data address operations begin when address bits 0 through 20 (+3P) are gated to the adder where 1 is added at bit position 20 (equivalent of adding eight bytes); the gating occurs with turn-on of the 'sequence 4' latch and the absence of the 'T6' clock signal. The incremented data address is latched into the adder by the 'T4' clock signal and gated back into the data address register. With the update data address operation complete, the data address register now contains the address location of the second data doubleword to be fetched from storage.

The 'T5' clock signal turns on the channel's 'gate A-register to B-register' latch to: (1) gate the first data doubleword into the B-register; and (2) to turn on the 'B-register full' latch. With the first data doubleword in the B-register, the A-register is prepared to accept the second data doubleword, when available. When the 'T6' clock signal is activated, the 'sequence 4' latch is turned off indicating that the 'update data address' and A-register to B-register transfer operations are complete. With the 'sequence 4' latch off, the channel clock turns off. At this point in the write operation, the channel is prepared to fetch the second doubleword from storage.

The status of channel logic conditions at the end of the fetch first data doubleword operations and prior to the fetch second data doubleword operations are: 'status in' signal is active; 'A-register full' latch is off; 'B-register full' latch is on; 'sequence 1' and 'sequence 2' latches are on and 'sequence 3', 'sequence 4', and 'sequence 5' latches are off; the first data doubleword is in the B-register; the data address register contains the address of the second doubleword; and the clock timing signals are inactive.

Fetch Second Data Doubleword. A summary and a detailed description of the fetch second data doubleword operations follow:

- Second data doubleword fetched and gated into A-register.
- 'A-register full' latch turns on.
- Channel raises 'service out' signal.
- Control unit drops 'status in' signal.
- 'Sequence 1' latch turns off; status byte in 'bus-in' latches reset.

Before turn-off of the 'sequence 4' latch (Diagram 5-8), the 'storage request' latch is turned on by the 'T2' clock signal causing a 'storage request' signal to be sent to the BCU interface. Turn-on of the 'storage request' latch causes the 'storage cycle' trigger to turn on which, in turn, activates and inhibits applicable channel control logic in preparation for the storage cycle. At this point, the channel awaits a 'BCU response' signal from the BCU interface.

Upon receipt of the 'BCU response' signal, the channel gates data address register bits 0 through 20 (+3P) to the SAB, and storage protect register bits 0 through 3 (+P) to the 'storage protection' lines. The channel then awaits receipt of the 'BCU advance pulse' signal from the BCU interface, signifying that the second data doubleword will be on the SBO lines shortly. The 'BCU advance pulse' signal is delayed (See "Fetch First Data Doubleword") to generate signals which gate the second data doubleword from the SBO lines into the A-register and turn on the 'A-register full' latch.

With the 'sequence 1' latch still on and the 'B-register full' latch on, the channel begins operations to clear the initial status byte from the I/O interface so that data transfer operations to the control unit can begin. To clear the status byte, the channel activates the 'service out' signal (by coincidence of the 'sequence 1 and 2', 'write', 'B-register full', 'A-register full' and 'status in' signals) causing a 'service out' signal to be sent to the control unit. The control unit responds to the 'service out' signal by dropping the 'status in' signal. The drop of the 'status in' signal is AND'ed in the channel with the 'sequence 1' and 'sequence 2' signals, causing the 'sequence 1' latch to turn off (Diagram 5-8). In addition, the drop of the 'status in' signal: (1) degates the 'service out' signal to the control unit causing this signal to deactivate; (2) turns off the 'latch status byte' trigger which, in turn, turns off the 'bus in'

latches. (The 'bus in' latches were previously latched during the initial setup routine when the status byte was received from the control unit.) At this point, the fetch second data doubleword operations are complete and the channel begins a data handling for write operation to transfer B-register data (first data doubleword) to the control unit.

Data Handling For Write Operation. A summary and a detailed description of the data handling for write operation follow:

- Channel receives 'service in' signal; control unit is requesting data byte.
- Channel generates 'service out' signal.
- 'Service in' signal causes channel to:
  1. gate data byte specified by byte counter to 'bus out' latches.
  2. latch data byte into 'bus out' latches.
  3. advance byte counter at end of byte transfer (with 'change byte count register write' signal.
  4. check for BCL = 0 (last byte in B-register gated to control unit) condition.
- Service in/service out exchange continues until BCL = 0 condition is detected.
- 'B-register full' latch turns off; 'sequence 3' latch and clock turn on.
- With doubleword in A-register:
  1. 'A-register full' latch turns off.
  2. 'BCL = 0' trigger turns off.
  3. B-register is reset.
  4. 'Gate register A to B' latch is turned on.
  5. A-register contents transferred to B-register.
  6. 'B-register full' latch turns on.
- Channel decrements count register value by eight bytes ('count minus 1' gated to adder); adder is latched, and results gated back to count register).
- Prior to count decrementing, channel samples count for last word condition (count less than 17 bytes). 'Last word' trigger turned on if condition is present.
- 'Sequence 4' latch turns on.
- Channel increments data address register contents (data address plus 1 gated to adder); adder is latched, and results gated back to data address register).
- 'Sequence 4' latch turns off.
- If 'last word' trigger off, 'storage request' latch turns on to fetch another data doubleword.

Write data transfer is performed on a byte-by-byte basis; byte transfers begin when the 'sequence 2' latch is on, the 'sequence 1' latch is off, and a 'service in' signal is received from the control unit indicating that the control unit is ready to receive data (Diagram 5-8). Assuming that a 'sequence 5' ending routine is not in progress and the 'B-register full' latch is on, the channel responds to the 'service in' signal by generating a 'service out' signal to the control unit. In the meantime, the 'service in' signal turns on the channel's 'write service in' latch. With the 'write service in' latch on, the B-register data byte specified by the count in the byte counter is latched into the 'bus-out' latches for presentation to the control unit. The 'write service in' signal also causes a 120 ns singleshot to fire and produce a 'change byte count register write' signal. This signal is applied to the byte counter to advance the byte count, thereby indicating that a data byte has been transferred from the B-register to the control unit. (Since the current byte count must be maintained until the byte transfer in progress is complete, the byte counter output is not updated until the fall of the 120-ns 'change byte count register write' signal.) In addition, the 120-ns 'change byte count register write' signal gates comparison circuits to check for a BCL = 0 condition to determine if the byte being transferred is from the last byte position (position 7) of the B-register. If the 'last word' trigger is on (indicating that the last word to be transferred for the write operation is in the B-register), the comparison circuits are also sampled for a BC = CTB condition; the BC = CTB condition with the 'last word' trigger on indicates that the last byte of the write operation is being transferred to the control unit. The 'change byte count register write' signal also generates a sampling pulse which is used to check for proper byte counter parity and 'bus-out' data parity. Detection of a byte counter parity error turns on the 'channel control check' latch and the write operation is subsequently terminated. Detection of a 'bus out' data parity error causes turn-on of the 'channel data check' latch, but does not terminate transfer operations.

Assume that the operations described above did not result in the detection of an error, BCL = 0 or BC = CTB condition. The control unit, upon receipt of the 'service out' signal, obtains the data byte from the 'bus out' latches and drops its 'service in' signal to the channel. When the 'service in' signal drops, the channel drops its 'service out' signal and waits for the next 'service in' signal from the control unit. Each time the channel receives a 'service in' signal, the channel performs the operation previously described.

For the first data doubleword, the first data byte to be transferred (written) may be located in any B-register byte position (depending upon the byte value specified by bits 21 through 23 of the data address obtained from the CCW and subsequently entered into the byte counter). Thus, any number of data bytes from one to eight may be transferred to the control unit before data from the last B-register byte position (byte position 7) is transferred. In other words, from one to eight 'service in/service out' exchanges between the channel and control unit can occur before all specified data from the first doubleword is transferred.

Assume that the 'service in/service out' exchange continues until the BCL = 0 condition ('last word' trigger off) is detected (Diagram 5-8). Upon detection of the BCL = 0 condition, the 'BCL = 0' trigger is turned on. Turn-on of this trigger turns off the 'B-register full' latch. Assuming the 'A-register full' latch is on, as is the case when the second data doubleword is in the A-register, the 'BCL = 0' signal also turns on the 'sequence 3' latch and the channel clock.

Note: Since the BCL = 0 condition was detected as a result of the 'service in' signal, the drop of the 'service in' signal and subsequent drop of the 'service out' signal occurs for the last byte in the same manner as for other byte transfer.

With the 'sequence 3' latch on and the 'T0' clock signal active: (1) the 'A-register full' latch is turned off; (2) the 'BCL = 0' trigger is turned off; (3) the B-register is reset; and (4) the 'gate register A to B' latch is turned on. With the turn-on of the 'gate register A to B' latch, the A-register data is transferred into the B-register and the 'B-register full' latch is turned on.

At clock 'T0' time, an update count operation (Diagram 5-10) also begins when the count register count is gated to the adder where it is decremented by one (eight bytes). Prior to the actual change of the count register contents, the 'T3' clock signal gates comparison logic to determine if a count of less than 17 bytes is contained in the count register. If the count is less than 17, the 'last word' trigger is turned on to indicate that all data required for the write operation is present in the channel. At 'T4' clock time, the adder is latched, and the decremented count is gated into the count register. At 'T5' clock time, the 'sequence 4' latch is turned on in preparation for an update data address operation. The update data address operation is necessary in the event a third doubleword must be fetched from storage.

The update data address operation (Diagram 5-8) begins with the turn-off of the channel clock and subsequent turn-off of the 'sequence 3' latch. The channel clock is again turned on by the 'not sequence 3', 'sequence 4', and 'not A-register full' signals. In the meantime, the data address in the data address register is gated to the adder by the 'sequence 4' and 'not T6' signals where the equivalent of eight bytes is added to the data address. The 'T4' clock signal latches the adder and gates the adder output to the data address register. At clock 'T6' time, the 'sequence 4' latch is turned off to end the update data address operation.

Assume that the 'last word' trigger is not on. In this case, the 'storage request' latch is turned on and a data fetch storage routine is initiated for the next doubleword. At the same time, the data handling for write operation begins to transfer the second doubleword from the B-register to the control unit.

The data fetch storage routine for the third and (if necessary) subsequent data doublewords is performed in the same manner as that described for fetching the second data doubleword, except that channel operations to clear the initial status byte are not performed. With the 'sequence 1' latch off, the channel performs the data handling for write operation while the next doubleword is being fetched from main storage.

Last Word Data Handling. A summary and detailed description of the last word data handling operation follow:

- 'BCL = 0' trigger turns on when all data from next to last doubleword has been transferred to control unit.
- 'A-register full' latch turns off.
- A-register contents gated to B-register.
- Count register contents gated to adder.
- 'Last word' trigger turns on (count less than 17 bytes).
- Adder latched; adder output gated to count register.
- Update data address operation performed.
- Bytes from last data doubleword transferred to control unit until BC = CTB condition detected (last byte of operation transferred).
- 'BC = CTB' trigger turns on; 'BC = CTB' latch turns on.
- Byte counter is reset.
- 'B-register full' latch turns off.
- Count register is reset.
- 'Sequence 5' latch turns on.
- Channel performs sequence 5 and interrupt routines.

Assume that the second doubleword has been completely transferred to the control unit (BCL = 0 detected) and that the third data doubleword is the last of the current write operation (Diagram 5-8). When the BCL = 0 condition is detected for the second data doubleword transfer, the 'BCL = 0' trigger is turned on and the 'B-register full' latch is turned off. Assuming that the third doubleword is in the A-register ('A-register full' latch is on), the 'sequence 3' latch and clock are turned on to begin an update count operation.

Note: If the 'A-register full' latch is off, the third doubleword has not yet been entered into the A-register. This indicates that main storage is not providing data at the channel data-transfer rate. In some cases, this may result in a 'status in' signal from the control unit when the channel is finally able to respond to the control unit's 'service in' signal with a 'service out' signal ('service out' response cannot occur until the 'B-register full' latch is on). Assume the channel is servicing a high-speed drum which requires response to its 'service in' signal within a specified period. If the channel is unable to respond within the specified time, the control unit will, when the channel finally does send a 'service out' signal, present a 'status in' signal and byte indicating an overrun condition at the I/O device.

At 'T0' clock time, the 'A-register full' latch is turned off, the third doubleword is gated to the B-register, and the count register contents minus one (minus eight bytes) are gated to the adder. At 'T3' clock time, the count comparison logic is gated and the 'last word' trigger is turned on. (Since the third data doubleword for the Diagram 5-8 example is the last for the current write operation, the count in the count register is less than 17 bytes). Subsequently, at 'T4' clock time, the adder is latched and the adder contents (decremented count value) gated to the count register. At 'T5' clock time, the 'sequence 4' latch is turned on. The channel then performs an update data address operation as previously described. With the 'B-register full' latch on and the 'last word' trigger on, the channel enters a data handling for write operation to transfer the appropriate number of last doubleword bytes from the B-register to the control unit.

As each data byte is transferred (service in/service out exchange), the channel samples for a BC = CTB condition (Diagram 5-8). When the BC = CTB condition is detected, it indicates that the last data byte specified by the original CCW count is being transferred to the control unit.

When detected, the BC = CTB condition activates a signal to turn on the 'BC = CTB' trigger. Turn-on of this trigger: (1) turns on the 'BC = CTB' latch; and (2) triggers a 110-ns singleshot which produces a signal to reset the byte counter and turn off the 'B-register full' latch. Turn-on of the 'BC = CTB' latch resets the count register and turns on the 'sequence 5' latch, causing the channel to enter a sequence 5 ending routine. The sequence 5 ending routine for the write operation is identical to that described for the read operation (See "Read Operation Ending Sequence"). During the sequence 5 ending routine, the channel obtains a status byte from the I/O device, performs operations to disconnect the control unit from the I/O interface, and performs operations to clear channel I/O interface controls. Upon completion of the sequence 5 routine, the channel enters an interrupt routine to store the channel status word, and to furnish the unit address and condition code 0 to the CPU interface. For more detailed information on the interrupt routine, refer to "Read Operation Ending Sequence."

Write Chain Data Operation

Write CDA operations permit the channel to transfer data specified by more than one CCW from main storage to the control unit selected during the initial selection routine without interrupting the CPU; in addition, the write CDA operation creates a continuous record on the I/O device. In general, the channel controls the transfer of all data specified by the first CCW obtained during the initial selection routine; then, since the first CCW specifies a write CDA operation, the channel fetches the second CCW from main storage and controls the transfer of all data specified by the second CCW. Only the command of the first CDA CCW is sent to the control unit; as long as the CDA flag is on in following CCW's, the commands in these CCW's are ignored by the channel. As long as each new CCW specifies a CDA operation, the channel continues to fetch CCW's and control the transfer of data specified by the CCW's. (For example, if the second CCW specifies data chaining, the channel will fetch a third CCW at the appropriate time and control the transfer of data specified by that CCW.) Write CDA operations also permit each CCW to specify a different starting address in storage, thus specifying the storage area from which data for that CCW is to be obtained.

Write CDA operations are described in the following paragraphs at three levels: (1) the first level ("Basic Write CDA Operations") introduces the basic operations performed; (2) the second level ("Simplified Write CDA Operations") presents a simplified description of the basic write CDA operations; and (3) the third level ("Detailed Write CDA Operations") presents a detailed description of the write CDA operations.

Basic Write CDA Operations

- Channel fetched first two data doublewords as during normal write operation.

- Channel clears 'sequence 1' and initial status condition.
- Channel transfers bytes from first doubleword, first CCW to control unit.
- BCL = 0 detected; second doubleword; first CCW transferred to B-register.
- Count register contents decremented; data address register contents incremented.
- Count less than 17 turns on 'CDA' latch; significant count bits transferred to mark-B register and 'retain storage' latch turned on.
- Channel fetches last data doubleword, first CCW, and gates last data doubleword into A-register.
- Channel fetches second CCW, checks program validity, and gates bits to data address, flag, and count registers.
- Channel transfers bytes from second doubleword, first CCW, to control unit.
- BCL = 0 detected; last doubleword, first CCW, transferred to B-register.
- 'Last word' trigger turns on.
- Channel fetches first data doubleword, second CCW, and gates into A-register.
- BC = MKB detected; channel adds DAB and count from new CCW and gates results to count register.
- DAB gated to byte counter.
- 'CDA' latch turns off; channel updates data address and fetches second data doubleword, second CCW.
- Channel transfers bytes from first data doubleword, second CCW to control unit.

The channel begins write CDA operations (Diagram 5-9) upon completion of the initial selection routine when the first CCW command specifies a write operation and the CDA flag bit is active. At the completion of the initial selection routine, the 'sequence 1' and 'sequence 2' latches are on.

To begin the write CDA operation, the channel fetches the first data doubleword from storage and gates it into the A-register. Subsequently, the first data dobleword is transferred from the A-register to the B-register and the address in the data address register is updated by an amount equivalent to eight bytes in preparation for fetching the second data doubleword. After the update data address operation is complete, the channel activates the 'storage request' signal to the BCU interface and fetches the second data doubleword from storage. When the second data doubleword is received, it is gated into the A-register. With data doublewords in both the A- and B-registers, the channel turns off the 'sequence 1' latch, clears the initial status, and begins to transfer the appropriate data bytes of the first doubleword from the B-register to the control unit. Since the DAB value (Diagram 5-9 example) received from the first CCW is 6, the channel transfers two data bytes from the B-register (byte locations 6 and 7) before a BCL = 0 condition is detected. When the BCL = 0 condition is detected (indicating that the last byte of the first data doubleword has been transferred), the channel transfers the second data doubleword of the first CCW to the B-register. The channel also decrements the count register by an amount equivalent to eight bytes to indicate that all data from the first doubleword has been transferred. After the count is decremented, a count value of 11 (B-hex) is contained in the count register.

Subsequent to the update count operation, the channel updates the address in the data address register by eight bytes in preparation for fetching the third (and last) doubleword for the first CCW. Because the CDA flag bit is active and the count register count value is less than 17, the channel's 'CDA' latch is turned on, the significant count register bits are transferred to the mark-B register, and the 'retain storage latch is turned on.

The 'retain storage' latch is turned on to indicate (within the channel) that two consecutive storage fetch cycles will be performed by the channel to obtain: (1) the third data doubleword of the first CCW; and (2) the second CCW. Since all data from the first CCW may not be transferred by the time the second CCW (with a new count) is entered into the channel, the significant count bits for the first CCW are transferred to the mark-B register; thus, the BC = CTB comparison required to detect the transfer of the last data byte of the first CCW is made with the count bits in the mark-B register.

Note: If the count bits were not transferred to the mark-B register and the count bits from the second CCW were gated into the count register before all data bytes for the first CCW were transferred, the count for the first CCW would be lost and detection of the last data byte transfer would not be possible.

With the channel conditioned for a retain storage sequence, the channel generates a 'storage request' signal and fetches the third (and last) data doubleword from storage. When the last doubleword for the first CCW is gated into the A-register, the channel fetches the second CCW from storage. Upon receipt of the second CCW on the SBO lines, the channel gates the data address bits into

the data address register, the flag bits into the flag register, and the count bits into the count register. The channel also turns off the 'retain storage' latch and turns on the 'CCW valid' latch to indicate that the second CCW is in the channel.

In the meantime, the channel has been transferring the eight bytes of the second data doubleword (for the first CCW) to the control unit. When the last byte of this doubleword has been transferred (BCL = 0 detected), the channel transfers the last doubleword for the first CCW from the A-register to the B-register and turns on the 'last word' trigger. Turn-on of the 'last word' trigger indicates that the last data doubleword of the first CCW is in the B-register, ready for transfer to the control unit; the 'last word trigger' signal is required in order for the channel to select the transfer of the last data byte for the first CCW.

The 'sequence 3' latch is turned off, and the count in the mark-B register is not decremented to reflect transfer of the second data doubleword for the first CCW. Update of the count is not required, since (to detect the last data byte for the first CCW) only the three low-order bits are compared with the count in the byte count latches, and decrementing the count by eight would not change the status of these bits. Recall that the count value transferred to the mark-B register was 11 (01011); decrementing the count by eight (00011) would not change the three low-order bits.

With the 'sequence 3' latch off, the channel fetches the first data word of the second CCW and gates it into the A-register. In addition, the channel begins transferring bytes of the last data doubleword for the first CCW to the control unit. As each byte is transferred ('last word' trigger is on), the channel samples for a BC = MKB condition, indicating that the last data byte of the first CCW is being transferred. (In the example of Diagram 5-9, three data bytes are transferred before the BC = MKB condition is detected.)

When the BC = MKB condition is detected, the 'sequence 3' latch is turned on and the first data doubleword for the second CCW is transferred from the A-register to the B-register.

The channel then begins operations to condition the channel to control data transfers specified by the second CCW. These operations consist of: (1) gating the new DAB (three low-order bits of the data address register) into the byte count register to identify the first byte to be transferred from the B-register; (2) adding the DAB and count register contents and gating the sum back into the count register to locate the last data byte of the last doubleword for the second CCW; and (3) turning off the 'CDA' latch to indicate that the first two operations have been completed.

The channel then updates the address in the data address register by the equivalent of eight bytes in preparation for fetching the second data doubleword of the second CCW. After the update data address operation, the channel turns on the 'storage request' latch and fetches the second data doubleword for the second CCW from storage. When received, the data doubleword is gated into the A-register. While the fetch data cycle is in progress, the channel begins transferring bytes from the first data doubleword for the second CCW to the control unit.

From this point on, the channel continues to fetch data doublewords and transfer bytes to the control unit as during a normal write operation (See "Write Operations"). Recall that in the example of Diagram 5-9, the second CCW indicated that further data chaining was not required. If further data chaining had been indicated, the 'CDA' latch would be turned on when the count register value became less than 17, and write CDA operations would continue.

Simplified Write CDA Operations

A simplified write CDA operation is illustrated in flowchart form in Diagram 5-10; the simplified write CDA operations are divided into subroutines indicated by dashed lines ('fetch first data doubleword, first CCW', etc.) Each subroutine indicated on the flowchart is described in a separate paragraph, beginning with "Initial Channel Conditions".

Initial Channel Conditions. This paragraph describes the channel conditions at the beginning of the write CDA operation. A successful initial selection routine has been completed (see Diagram 5-1), leaving the 'sequence 1' and 'sequence 2' latches on, the 'sequence 3', 'sequence 4', and 'sequence 5' latches off, and the 'status in' signal from the control unit active. The channel has decoded a write command (command register bit 7 active) and the 'gate command out' latch is on (Diagram 5-13). In addition, the count from the first CCW specifies that 13 data bytes are to be transferred from storage to the control unit beginning with the byte at storage address location 2006. Since data is fetched from storage on doubleword boundaries, three data doubleword fetches will be required to obtain the specified 13 data bytes. (The last two bytes from the first data doubleword, all eight bytes of the second data doubleword, and the first three bytes from the third data doublewords.) The CDA bit in the flag register is active, indicating that data chaining is to be performed. The channel clock is turned on and the 'T3' clock signal has been activated. With the above described conditions present, the channel is ready to fetch the first data doubleword for the first CCW of the write CDA operation.

Fetch First Data Doubleword, First CCW. A summary and a simplified description of 'fetch first data doubleword, first CCW' operations follow:

- 'Storage request' latch turns on.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Data address gated to SAB; 'storage protect' key gated to 'storage protection' lines.
- Channel receives 'BCU advance pulse' signal from BCU interface.
- Delayed signal gates data from SBO lines to A-register.
- 'A-register full' latch turns on.
- A-register contents transferred to B-register.
- 'A-register full' latch turns off; 'B-register full' latch turns on.
- Data address updated eight bytes.

The channel operations to fetch the first data doubleword for the first CCW begin with the turn-on of the 'storage request' latch. This causes a 'storage request' signal to be sent to the BCU interface requesting a storage cycle to fetch the first data doubleword (for the first CCW) from storage. Also, with the 'storage request' latch on, the 'storage-cycle' trigger is turned on to condition channel logic for the storage fetch operation in progress. When the 'storage request' signal is honored, the channel receives a 'BCU response' signal which causes address information for the first data doubleword to be gated to the BCU interface. Address information gating consists of: (1) gating bits 0 through 20 (+3P) from the data address register to the SAB; and (2) gating bits 0 through 3 (+P0) from the storage protect register to the 'storage protection' lines. (Mark bits are not required for a fetch storage cycle operation.) When the storage cycle has progressed to the point where data is forthcoming on the SBO lines, the channel receives a 'BCU advance pulse' signal from the BCU interface. Since the 'BCU advance pulse' signal precedes the data on the SBO lines, the channel delays the signal and uses the delayed signal to gate data from the SBO lines into the A-register; with the gating of data to the A-register, the 'A-register full' latch is turned on. At this point, the first of the two required doublewords is in the channel.

Since a second doubleword must be fetched from storage, the channel's clock and 'sequence 4' latch are turned on to: (1) turn off the 'A-register full' latch, gate the data doubleword from the A-register to the B-register, and turn on the 'B-register full' latch; and (2) to update the data address in data address register. In the first case, the 'A-register to B-register' transfer is performed to prepare the A-register to receive the second doubleword from storage. In the second case, the data address is updated eight bytes (data address register plus 1) so that the data address register specifies the address location of the second doubleword for the first CCW. After the data address is updated, the 'sequence 4' latch is turned off; the channel is now prepared to fetch the second data doubleword for the first CCW (Diagram 5-10).

Fetch Second Data Doubleword, First CCW. A summary and a simplified description of the 'fetch second data doubleword, first CCW' operation follow:

- Second doubleword fetched and gated to A-register; 'A-register full' latch turns on.
- Channel raises 'service out' signal; 'status in' signal drops.
- 'Latch status byte' trigger turns off; 'service out' signal drops.

Operations performed to fetch the second data doubleword are similar to those performed to fetch the first doubleword (Diagram 5-10). Just prior to turn-off of the 'sequence 4' latch, the channel turns on the 'storage request' latch and 'storage cycle' trigger as during the first fetch cycle. Upon receipt of the 'BCU response' signal from the BCU, the channel gates the address information (data address and storage protect key) to the BCU interface and awaits a 'BCU advance pulse' signal from the BCU interface. Upon receipt of the 'BCU advance pulse' signal, the second doubleword is gated from the SBO lines into the A-register in the same manner described for the first doubleword. With data in the A-register, the 'A-register full' latch is turned on.

With both the 'B-register full' and the 'A-register full' latches on, the channel begins operations to clear the 'status in' byte and signal presented by the control unit at the end of the initial selection routine. To clear the 'status in' condition, the channel sends a 'service out' signal to the control unit. The control unit responds by dropping the 'status in' signal. This causes the channel to: (1) turn off its 'sequence 1' latch; (2) turn off the 'latch status byte' trigger to remove the latch signal from the 'bus-in' latches; and (3) drop the 'service out' signal. With this

sequence of events complete, the channel is ready to begin transferring the first data doubleword for the first CCW to the control unit.

Data Handling, First Data Doubleword, First CCW. A summary and a simplified description of data handling, first data doubleword, first CCW operations follow:

- Channel receives 'service in' signal; raises 'service out' signal.
- Data byte gate from B-register position specified by byte counter; data byte gated through 'bus-out' latches control unit.
- Byte counter updated.
- Service in/service out exchange continues until BCL = 0 condition detected.
- 'B-register full' latch turns off.
- 'A-register full' latch turns off.
- A-register contents transferred to B-register; 'B-register full' latch turns on.
- Count register value decremented; data address register incremented.

Transfer of data from the first doubleword for the first CCW (data handling for write operation) begins when the channel receives a 'service in' signal indicating the control unit is ready to receive a data byte (see Diagram 5-10). Upon receipt of the 'service in' signal, the channel gates a data byte from the B-register byte position specified by the count in the byte counter. Since, for the example in Diagram 5-10, the starting data byte address is 2006, the byte counter contains a count of six at the beginning of the data transfer; thus, byte position six is the first byte position gated to the control unit. The data byte is gated through the channel's 'bus-out' latches to the control unit on the 'bus-out' lines. While the data byte is being gated from the B-register, the channel updates the byte counter and monitors the channel logic for a BCL = 0 condition. (A BCL = 0 condition signifies that the byte being transferred is from the last byte position of the B-register; i.e., all specified data from the first data doubleword for the first CCW has been transferred.) In Diagram 5-10, the byte counter is updated during the first data byte transfer to a count of seven, and the BCL = 0 condition is not detected.

The control unit, upon receipt of the first data byte and 'service out' signal, drops the 'service in' signal, causing the channel to drop its 'service out' signal. The channel then waits for the next 'service in' signal and responds with the 'service out' signal to gate the second byte to the control unit. In Diagram 5-10, the second data byte is gated from B-register byte position 7 (last byte position of the B-register), and the BCL = 0 condition is detected.

After the transfer of the second data byte is complete ('service in' signal drops causing the 'service out' signal to drop), the BCL = 0 condition causes: (1) the 'B-register full' latch to turn off; (2) the 'A-register full' latch to turn off; (3) the B-register to be reset; (4) the second data doubleword for the first CCW to be gated from the A-register to the B-register; and (5) the 'B-register full' latch to be turned on. Following the A-to-B register data transfer, the count in the count register is decremented by eight bytes; the decremented count reflects that all specified data in the first data doubleword has been transferred to the control unit. In Diagram 5-10, note that the count register contains a count of 11 (B-hex) after the update count operation.

When the count in the count register has been updated, the address in the data address register is updated eight bytes (data address plus one). At the completion of the update data address operation, the data address register contains the address of the storage location from which the third data doubleword is to be fetched. At this point, data handling for the first data doubleword of the first CCW is complete.

Write CDA Setup. A summary and a simplified description of the write CDA setup operations follow:

- 'CDA' latch turns on when count less than 17.
- Significant count register bits gated to mark-B register.
- 'Retain storage' latch turns on.

The write CDA setup operations are performed during a write CDA operation when the count register contains a count value of less than 17 and the update data address operation is complete (Diagram 5-10). When a count value of less than 17 is detected, the 'CDA' latch is turned on, and count register bits 19 through 23 are gated to mark-B register bit positions 4 through 0, respectively. With the count in the mark-B register, the count register contents are no longer required for operations involving the first CCW. Thus, when the second CCW is fetched into the channel, the new count may be entered into the count register even though data handling operations for the first CCW are still in progress. After the mark-B register receives the first CCW count, BC to CTB comparisons to detect transfer of the last data byte of the first CCW are made using the count in the mark-B register (CTB = mark-B comparison). Following the transfer of count bits to the

mark-B register, the 'retain storage' latch is turned on to complete the write CDA setup operations. The 'retain storage' latch is turned on in preparation for fetching two doublewords from storage in succession, the last data doubleword for the first CCW, and the second CCW.

Fetch Last Doubleword of First CCW. A summary and a simplified description of the fetch last doubleword of first CCW operations follow:

- 'Storage request' latch turns on.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Address information gated to BCU interface.
- Channel receives 'BCU data request' signal; 'storage request' signal to BCU interface degated.
- Channel receives 'accept' signal.
- 'CCW fetch' latch turns on.
- Channel receives 'BCU advance pulse' signal.
- 'Raw advance SS' signal gates data from SBO lines to A-register.
- 'A-register full' latch turns on.

Channel operations to fetch the last doubleword of the first CCW begin with the turn-on of the 'storage request' latch (Diagram 5-10).

Note: The 'storage request' latch remains on for the entire fetch last doubleword of the first CCW operation and is turned off after the 'storage request' signal for the second CCW has been honored; however, the 'storage request' signal to the BCU interface is degated while the 'BCU data request' or 'accept latch' signal are active at the channel.

After turn-on of the 'storage request' latch, the fetch last doubleword of first CCW operation (Diagram 5-10) continues with the turn-on of the 'storage cycle' trigger. The 'storage cycle' signal activates applicable channel logic for the storage fetch operation. With the turn-on of the 'storage cycle' trigger, the channel enters a retain storage routine and awaits a 'BCU response' signal from the BCU interface. Upon receipt of the 'BCU response' signal, the channel gates address information to the BCU interface (data address register bits 0 through 20 plus 3P to the SAB, and storage protect register bits 0 through 3 plus P to the 'storage protection' lines). Subsequently, the channel receives a 'BCU data request' signal; while present, this signal degates the 'storage request' signal to the BCU interface. When the 'BCU response' signal falls (indicating that the address information is no longer required), the 'remember BCU response' latch is turned on and the channel waits for an 'accept' signal from the BCU interface. Receipt of the 'accept' signal indicates that the storage cycle for the last data doubleword of the first CCW has started in main storage. The 'remember BCU response' signal and 'accept' signal are AND'ed to turn on the 'CCW fetch' latch indicating that a second storage fetch operation for the second CCW is required. (See "Fetch Second CCW".) When the channel is used with the System/360, Model 91, a 'control word request' signal is also raised, indicating that the fetch cycle is for a control word.

Following receipt of the 'accept' signal, the channel receives a 'BCU advance pulse' signal, indicating that the last data doubleword is forthcoming on the SBO lines. The 'BCU advance pulse' signal is time delayed by the channel until the data is present on the SBO lines. The delayed 'BCU advance pulse' signal ('raw advance SS') causes the data to be gated into the A-register and causes the 'A-register full' latch to be turned on. With the turn-on of the 'A-register full' latch, the fetch operation for the last data doubleword of the first CCW is complete.

Fetch Second CCW. A summary and a simplified description of the 'fetch second CCW' operation follow:

- 'Storage request' signal to BCU interface activated when 'BCU data request' and 'accept latch' signals fall.
- Channel receives 'BCU response' signal.
- Command address register contents gated to SAB.
- 'Retain storage' latch turns off.
- Channel receives 'BCU advance pulse' signal.
- SBO bits 8-31 (+3P) gated to data address register, 32-39 (+P) to flag registers, and 48-63 (+2P) to count register.
- SBO bits 4-7 checked for 'TIC' command, 37-39 for all 0's and 48-63 for not all 0's.
- If TIC command not detected and 0's check successful, 'CCW valid' latch turns on.

The fetch second CCW channel operation begins with the turn-on of the 'CCW fetch' latch (Diagram 5-10). Recall that this latch is turned on when the 'accept' signal for the last data doubleword of the first CCW is received by

the channel. The 'CCW fetch' signal conditions the channel to gate the contents of the command address register to the SAB when required; in addition, the 'CCW fetch' signal conditions channel logic to perform the necessary updating and zero checking operations required for a CCW fetch storage cycle.

Recall, also, that the 'storage request' latch turned on at the beginning of the fetch last doubleword of first CCW operation, is still on. Thus, when the 'BCU data request' signal from the fetch last doubleword of first CCW operation falls and the 'accept' latch is turned off by receipt of the 'BCU advance pulse' signal, the 'storage request' signal to the BCU interface is activated. With the 'storage request' signal active and the 'CCW fetch' latch on, the channel waits for a 'BCU response' signal from the BCU interface. Upon receipt of the 'BCU response' signal, the channel gates the address of the second CCW from the command address register to the SAB. The channel's 'retain storage' latch is then turned off and the channel awaits receipt of the 'BCU advance pulse' signal indicating that the second CCW is forthcoming on the SBO lines.

Note: The 'retain storage' latch is turned off by coincidence of the 'CCW fetch' and the 'storage cycle complete' signals; the 'storage cycle complete' signal is generated by delaying the 'BCU advance pulse' signal for the fetch last doubleword of first CCW operation. Since the arrival of the 'BCU response' signal may be delayed due to preemption of storage (by the CPU or a higher priority channel) the 'retain storage' latch may be turned off earlier in the sequence than shown in Diagram 5-10. However, this does not affect fetch second CCW operations.

When the 'BCU advance pulse' signal for the second CCW arrives at the channel, it is delayed and causes the CCW to be gated into the appropriate channel registers. SBO bits 8 through 31 (+3P) are gated to the data address register, SBO bits 32 through 39 (+P) to the flag register, and SBO bits 48 through 63 (+2P) to the count register. While the CCW is still on the SBO lines, command bits 4 through 7 are checked for a TIC command. Assuming a TIC command is not present, SBO bits 37 through 39 are checked for an all 0's (no error) condition and SBO bits 48 through 63 are checked for a not all 0's (no error) condition. Assuming no errors are detected, the 'CCW valid' latch is turned on, indicating that the second CCW is in the channel and is properly formatted. With the 'CCW valid' latch on, the fetch second CCW operation is complete.

Data Handling, Second Doubleword, First CCW. Data handling operations for the second data doubleword of the first CCW (Diagram 5-10) begin following turn-on of the 'B-register full' latch at the end of the data handling, first doubleword, first CCW operation. Since all eight bytes of the second doubleword (Diagram 5-10) are to be transferred, eight service in/service out exchanges between the channel and the control unit are required before all data is transferred from the B-register and the BCL = 0 condition is recognized. After all data bytes have been transferred (BCL = 0 detected), the 'sequence 3' latch is turned on (assuming that the third data doubleword for the first CCW is in the A-register).

Although the 'sequence 3' latch is turned on at this time, an update count operation is not performed, as is usually the case. Instead, the 'sequence 3' latch is turned on so that the 'last word' trigger may be turned on when the BCL = 0 condition is detected.

Following turn-on of the 'sequence 3' latch, the 'A-register full' latch is turned off and the third (last) data doubleword of the first CCW is transferred from the A-register to the B-register. With the last data doubleword of the first CCW in the B-register and the 'CDA' latch on, the 'last word' trigger is turned on to end the data handling, second data doubleword, first CCW operation.

Fetch First Doubleword of Second CCW. With the turn-on of the 'last word' trigger (Diagram 5-10), operations to fetch the first doubleword of the second CCW begin. With the 'CCW valid' latch on, turn-on of the 'last word' trigger turns on the 'storage request' latch to activate the 'storage request' signal to the BCU interface. Following turn-on of the 'storage request' latch, the 'sequence 3' latch is turned off, allowing data handling operations for the third (and last) data doubleword of the first CCW to begin.

Turn-on of the 'storage request' latch also turns on the 'storage cycle' trigger; the output of this trigger activates applicable channel logic for the fetch operation in progress. When the channel receives a 'BCU response' signal from the BCU interface, it gates the required data address information to the BCU interface (bits 0 through 20, plus 3P from the data address register to the SAB, and bits 0 through 3 plus P from the storage protect register to the 'storage protection' lines). Subsequently, the channel receives a 'BCU advance pulse' signal, indicating that the first data doubleword of the second CCW is forthcoming on the SBO lines. The channel time delays the 'BCU advance pulse' signal until the data is present on the SBO lines; the delayed signal causes the data doubleword to be gated into the A-register and causes the 'A-register full' latch to turn on. Turn-on of the 'A-register full' latch indicates that the data doubleword is in the A-register and that the fetch first data doubleword of second CCW operation is complete.

Data Handling, Third Doubleword, First CCW. Data handling operations to transfer the data bytes of the last doubleword, first CCW, from the B-register to the control unit begin after turn-off of the 'sequence 3' latch during the 'fetch first doubleword of second CCW' operation

(Diagram 5-10). As each data byte is transferred (service in/service out exchange between the channel and control unit), the count in the byte count latches is compared with the count in the mark-B register. When the two counts are equal (BC = mark-B detected), the last data byte required by the original count from the first CCW is being transferred to the control unit. Recall that, for the Diagram 5-10 example, the three least significant bits in the mark-B register indicate a value of three; thus, three data bytes (from B-register byte positions 0, 1, and 2) must be transferred to the control unit to complete the transfer of all data bytes specified by the first CCW. When the third data byte is being transferred, the BC = mark-B condition is detected, causing the 'BC = CTB' trigger to turn on and the 'B-register full' latch to turn off. With the trigger and latch in the condition indicated, all data specified by the first CCW has been transferred to the control unit. The 'BC = CTB trigger' signal also attempts to turn on the 'BC = CTB' latch; however the 'BC = CTB' latch is held reset by the 'CDA latch' signal.

In addition, detection of the BC = mark-B condition causes the byte counter to be reset so that the DAB from the second CCW can be gated into the byte counter during the second CCW setup operation. With the byte counter reset, data handling operations for the third data doubleword of the first CCW are complete.

Second CCW Setup Operation. A summary and a simplified description of the second CCW setup operation follow:

- BC = mark-B detection turns on 'sequence 3' latch.
- 'A-register full' latch turns off.
- First data doubleword of second CCW transferred from A- to B-register.
- Updated command address value gated from adder to command address register.
- DAB and count register values gated to adder; sum gated back to count register.
- DAB gated to byte counter.
- 'Sequence 4' latch turns on.
- Count register value sampled for 'count less than 9' value:
  1. Count less than 9, 'last word' trigger remains on.
  2. Count greater than 9, 'last word' trigger turns off.
- 'BC = CTB' trigger and 'CDA' latch turn off.

The second CCW setup operations (Diagram 5-10) are performed to prepare the channel to begin data transfer operations for data doublewords specified by the second CCW. When the BC = mark-B condition is detected, the 'sequence 3' latch is turned on. This causes the 'A-register full' latch to turn off and the first data doubleword of the second CCW to be transferred from the A-register to the B-register. Turn-on of the 'sequence 3' latch also turns on the channel clock, causing the updated command address (command address plus equivalent of eight bytes) to be latched into the adder and subsequently gated from the adder into the command address register.

Note: The command address register plus 1 (eight bytes) value was previously gated to the adder when the 'CCW fetch' latch was turned on.

The command address register value is updated in the event a third CCW is required during the write CDA operation; if a third CCW is not required (as is the case in the Diagram 5-10 example), the updated command address is placed in the channel status word at the end of the write CDA operation.

As the updated command address is being transferred into the command address register, data address register bits 21 through 23 (the DAB) plus the count in the count register are gated to the adder. Subsequently the DAB plus count sum is latched into the adder and gated back into the count register. The count register now contains a count which is used to identify the last data byte to be transferred from the last data doubleword of the second CCW.

When the DAB plus count value is gated to the adder, the DAB bits are also gated into the byte count register to identify the first byte to be transferred from the first data doubleword of the second CCW. (Since, in the Diagram 5-10 example, the first data byte to be transferred is obtained from storage address 4050, the DAB specifies doubleword boundaries, and all eight bytes of the first doubleword are to be transferred to the control unit).

After the DAB plus count operation is performed, the 'sequence 4' latch is turned on and the value in the count register is sampled for a count value of less than nine. If a value of less than nine is detected, it indicates that all data for the second CCW is presently in the channel, and the 'last word' trigger remains turned on. In Diagram 5-10, the second CCW specifies the transfer of 30 data bytes; thus, a count value of less than nine is not detected, and the 'last word' trigger, as well as the 'BC = CTB' trigger and the 'CDA' latch, are turned off. Turn-off of the 'CDA' latch completes the second CCW setup operations, and the channel is ready to perform operations to fetch the second data doubleword of the second CCW.

Fetch Second Data Doubleword of Second CCW. Channel operations to fetch the second data doubleword of the second CCW (Diagram 5-10) begin with an update data address operation. The address in the data address register is updated eight bytes; after updating, the data address specifies the storage location of the second data doubleword. Following updating of the data address, the 'storage request' latch is turned on to activate the 'storage request' signal to the BCU interface.

While data handling operations for the first doubleword of the second CCW are in progress, the channel continues operations to fetch the second data doubleword from storage. When the data doubleword is received by the channel, the 'A-register full' latch is turned on and the data doubleword is gated into the A-register to complete the fetch second data doubleword of second CCW operation.

Data Handling for First Data Doubleword, Second CCW. Operations to transfer data bytes for first data doubleword of the second CCW are performed in the same manner as described in "Data Handling, First Data Doubleword, First CCW". That is, data is transferred on a byte-by-byte basis due to exchange of 'service in' and 'service out' signals between the control unit and channel until a BCL = 0 condition is detected. Since all eight bytes of the first data doubleword for the second CCW are to be transferred, eight service in/service out exchanges are necessary to transfer the data. When the BCL = 0 condition is detected, channel operations continue as for a normal write operation. If the second CCW contains an active CDA bit, all operations previously described are repeated to fetch the third CCW and begin transferring data specified by that CCW. Otherwise, the channel ends the current write operation with a sequence 5 routine when all specified bytes have been transferred.

Detailed Write CDA Operations

Detailed descriptions of the write CDA operations are based upon Diagram 5-11. For simplicity, the detailed write CDA operation is described by subroutine.

Detailed Initial Channel Conditions. It is assumed that a successful initial selection routine has been completed, and that the channel has sent a 'release' signal to the CPU interface (Diagram 5-11). Other channel conditions at the start of the write CDA operation are: (1) 'sequence 1' and 'sequence 2' latches on; (2) 'sequence 3', 'sequence 4', and 'sequence 5' latches off; (3) 'status in' signal active; (4) write command is specified by active command register bit 7; (5) 'gate command out' (read or write) latch is on; and (6) the CDA bit in the flag register is active. With the above described conditions present, the channel is ready to begin write CDA operations.

Detailed Fetch First Doubleword of First CCW. A summary and a detailed description of the fetch first doubleword of first CCW operation follow:

- 'Start I/O' latch turns off; 'storage request' latch turns on.
- Clock turns off.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Data address register bits 0-20 (+3P) gated to SAB.
- Storage protect register bits 0-3 (+P) gated to 'storage protection' lines.
- Channel receives 'BCU data request' signal; 'storage request' signal to BCU interface degated.
- 'BCU response' signal drops; 'remember BCU response' latch turns on.
- Channel receives 'accept' signal.
- 'Storage request' latch turns off; 'accept' latch turns on.
- Channel receives 'BCU advance pulse' signal.
- 'Raw advance SS' signal gates data from SBO lines into A-register.
- 'A-register full' latch turns on.
- 'Sequence 4' latch turns on.
- 'A-register full' latch turns off; channel clock turns on.
- Data address register contents plus 1 gated to adder.
- Adder latched; adder output gated to data address register ('T4' time).
- A-register contents transferred to B-register; 'B-register full' latch turns on ('T5' time).
- 'Sequence 4' latch turns off.

Operations to fetch the first doubleword of the first CCW (Diagram 5-11) begin with the turn-on of the 'storage request' latch causing a 'storage request' signal to be sent to the BCU interface. The 'storage request' latch is turned on as a result of the CPU dropping the 'start I/O' signal which causes the channel's 'start I/O' latch to turn off

(Diagram 5-2). Turn-off of the 'start I/O' latch (1) turns on the 'storage request' latch and (2) turns off the channel clock (Diagram 5-11). At the time the channel clock is turned off, the 'T3' clock signal is active, allowing the 'storage request' latch to be turned on.

With the 'storage request' signal active, the 'storage cycle' trigger is turned on to activate the applicable channel logic for the fetch storage cycle. After the 'storage cycle' trigger is turned on, the channel awaits receipt of a 'BCU response' signal, indicating that the 'storage request' signal for the first data doubleword of the first CCW has been honored.

When the channel receives a 'BCU response' signal, the data address register contents (bits 0 through 20 plus 3P) are gated to the SAB and storage protect register bits 0 through 3 (plus parity) are gated to the 'storage protection' lines.

After the 'BCU response' signal is received, the channel awaits receipt of the 'BCU data request' signal. When this signal is received, the 'storage request' signal to the BCU interface is degated. Subsequently, the 'BCU response' signal to the channel drops, causing the 'remember BCU response' latch to turn on. The channel then receives an 'accept' signal from the BCU interface. This signal signifies that main storage has started the requested storage cycle. In the channel, the 'accept' signal turns off the 'storage request' latch and turns on the 'accept' latch. The channel then awaits receipt of the 'BCU advance pulse' signal from the BCU interface signifying that the first data doubleword for the first CCW will be on the SBO lines in approximately 200 ns. The 'BCU advance pulse' signal is delayed within the channel to generate a 'raw advance singleshot' signal which is active while the data is on the SBO lines. The 'raw advance singleshot' signal activates the 'gate SBO to A-write' signal which turns on the 'A-register full' latch. In addition, the 'gate SBO to A-write' signal turns on the 'SBO to A-register' latch; the output of this latch gates the data doubleword from the SBO lines into the A-register.

Recall that the 'sequence 1' latch is still on; at this stage in the write operation, the on condition of the 'sequence 1' latch indicates that the second doubleword for the first CCW has not yet been fetched into the channel, and that update data address operations must be performed before the second doubleword can be obtained.

With the 'B-register full' latch off and the 'A-register full' latch on, the 'sequence 4' latch is turned on. When the 'sequence 4' latch is on, the 'A-register full' latch is turned off and the channel clock is turned on. The clock is turned on to time the update data address operation and to time the transfer of the A-register contents to the B-register.

Before the second data doubleword of the first CCW can be fetched from storage, the address in the data address register must be incremented by eight bytes; otherwise, the same address location from which the first doubleword was obtained would again be accessed to obtain duplicate data for the second doubleword. Update data address operations begin when address bits 0 through 20 (+3P) are gated to the adder where 1 is added at bit position 20 (equivalent of adding eight bytes); the gating occurs with the turn-on of the 'sequence 4' latch and the absence of the 'T6' clock signal. The incremented data address is latched into the adder by the 'T4' clock signal, and gated back into the data address register. With the update data address operation complete, the data address register now contains the address location of the second data doubleword for the first CCW to be fetched from storage.

At 'T5' clock time, the channel's 'gate A-register to B-register' latch is turned on to: (1) gate the first data doubleword for the first CCW into the B-register; and (2) to turn on the 'B-register full' latch. With the first data doubleword in the B-register, the A-register is prepared to accept the second data doubleword, when available. At 'T6' clock time, the 'sequence 4' latch is turned off, indicating that the update data address and A-register to B-register transfer operations are complete. At this point in the write operation, the channel is prepared to fetch the second doubleword from storage.

Since the CDA bit is active, the value in the count register is examined at 'T5' clock time of the update data address operation (Diagram 5-11). If a count value of less than 17 bytes is detected after fetching the first data doubleword, the channel's 'CDA' latch is turned on to indicate that only two data doublewords are to be fetched for the first CCW, and that write CDA setup operations are required. (See "Detailed Write CDA Setup".)

Note: During the initial selection routine (Diagram 5-2), the value in the count register is sampled after the DAB plus count operation for a count of eight or less bytes. If the count is eight or less, only one doubleword is to be fetched for the first CCW and the 'last word' trigger is turned on at that time. The first data doubleword is then fetched as described above, and the 'CDA' latch turned on when the count is sampled for a count of less than 17 bytes as described above. This is a special case for the write CDA operation; i.e., only one data doubleword is fetched from storage and then the second CCW is fetched. See "Detailed Write CDA Setup" for a more detailed description of this special case operation.

Assuming that neither a count of less than nine nor a count of less than 17 has been detected, the status of channel logic conditions at the end of the fetch first data doubleword operations and prior to the fetch second data doubleword operations is: 'status in' signal is active; 'A-register full' latch is off; 'B-register full' latch is on; 'sequence 1' and 'sequence 2' latches are on, and 'sequence

3', 'sequence 4', and 'sequence 5' latches are off; the first data doubleword is in the B-register, the data address register contains the address of the second doubleword; and the clock timing signals are still active.

Detailed Fetch Second Data Doubleword, First CCW. A summary and detailed description of the fetch second data doubleword, first CCW operation are as follows:

- 'Storage request' latch turned on by 'T5' clock signal.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Data address gated to SAB; storage protect bits gated to 'storage protection' lines.
- 'BCU data request' and 'accept' signals received by channel as during fetch first data doubleword, first CCW operation.
- Channel receives 'BCU advance pulse' signal.
- Data gated into A-register; 'A-register full' latch turns on.
- Channel activates 'service out' signal.
- 'Status in' signal drops.
- 'Sequence 1' latch turns off.
- 'Latch status byte' trigger turns off.
- Channel deactivates 'service out' signal.

Assuming that a second data doubleword is to be fetched for the first CCW, channel operations to fetch the data doubleword are performed as described in the following text. Before the turn-off of the 'sequence 4' latch (Diagram 5-11) the 'storage request' latch is turned on by the 'T5' clock signal causing a 'storage request' signal to be sent to the BCU interface. Turn-on of the 'storage request' latch causes the 'storage cycle' trigger to turn on, which in turn activates and inhibits the applicable channel control logic in preparation for the storage cycle. At this point, the channel awaits a 'BCU response' signal from the BCU interface.

Upon receipt of the 'BCU response' signal, the channel gates the data address register bits to the SAB, and the applicable storage protect register bits to the 'storage protection' lines. The channel then awaits receipt of the 'BCU advance pulse' signal from the BCU interface signifying that the second data doubleword will be on the SBO lines shortly. The 'BCU advance pulse signal is delayed (see "Detailed Fetch First Data Doubleword") to generate signals which gate the second data doubleword from the SBO lines into the A-register and turn on the 'A-register full' latch.

With the 'sequence 1' latch still on and the 'B-register full' latch on, (Diagram 5-11), the channel begins operations to clear the initial status byte and 'status in' signal so that data transfer operations to the control unit can begin. To clear the initial status condition, the channel's 'service out' signal is activated (by coincidence of the 'sequence 1 and 2', 'write', 'B-register full', 'A-register full', and 'status in' signals) causing a 'service out' signal to be sent to the control unit. The control unit responds to the 'service out' signal by dropping the 'status in' signal. The drop of the 'status in' signal is AND'ed in the channel with the 'sequence 1' and 'sequence 2' signals, causing the 'sequence 1' latch to turn off. In addition, the drop of the 'status in' signal: (1) degates the 'service out' signal to the control unit causing this signal to drop; (2) turns off the 'latch status in' trigger which, in turn, turns off the 'bus-in' latches. (The 'bus in' latches were previously latched during the initial setup routine when the status byte was received from the control unit.) At this point, the fetch second data doubleword, first CCW operations are complete and the channel begins a data handling for write operation to transfer B-register data (first data doubleword for first CCW) to the control unit.

Detailed Data Handling, First Doubleword, First CCW. A summary and a detailed description of the data handling, first doubleword, first CCW operations follow:

- Channel receives 'service in' signal and activates 'service out' signal.
- 'Service in write' latch turns on.
- B-register byte (specified by byte counter) latched into 'bus out' latches.
- 'Change byte count register write' signal activated to:
  1. Advance byte counter.
  2. Gate comparison logic for BCL = 0 condition.
  3. Sample byte counter parity.
  4. Sample 'bus out' parity.
- Control unit drops 'service in' signal.
- Channel drops 'service out' signal and turns off 'service in write' latch.
- Service in/service out exchange continues until BCL = 0 condition detected.

- 'BCL = 0' trigger turns on.
- 'B-register full' latch turns off.
- 'Sequence 3' latch and clock turn on.
- 'A-register full' latch and 'BCL = 0' trigger turn off; B-register is reset.
- A-register contents transferred to B-register.
- Count register count minus 1 value gated to adder.
- Adder latched; adder output gated to count register.
- 'Sequence 4' latch turns on; clock turns off.
- 'Sequence 3' latch turns off; clock turns on.
- Data address register contents plus 1 gated to adder.
- Adder latched; adder output gated to data address register.
- 'Sequence 4' latch turns off.

Write data transfer is performed on a byte-by-byte basis (Diagram 5-11). Byte transfers begin when the 'sequence 2' latch is on, the 'sequence 1' latch is off, and a 'service in' signal is received from the control unit indicating that the control unit is ready to receive data. Assuming that a 'sequence 5' ending routine is not in progress (no error has been detected by the channel) and the 'B-register full' latch is on, the channel responds to the 'service in' signal by activating the 'service out' signal to the control unit. In the meantime, the 'service in' signal turns on the channel's 'write service in' latch. With the 'write service in' latch on, the B-register data byte specified by the count in the byte counter is latched into the 'bus out' latches for presentation to the control unit. The 'write service in' signal also causes a 120 ns singleshot to fire and produce a 'change byte count register write' signal. This signal is applied to the byte counter to advance the byte count, thereby indicating that a data byte has been transferred from the B-register to the control unit. (Since the current byte count must be maintained until the byte transfer in progress is complete, the byte counter output is not updated until the fall of the 120 ns 'change byte count register write' signal.) In addition, the 120 ns 'change byte count register write' signal gates comparison circuits to check for a BCL = 0 condition to determine if the byte being transferred is from the last byte position (position 7) of the B-register.

Another function of the 'change byte count register write' signal is to generate a sampling pulse which is used to check for proper byte counter parity and 'bus out' data parity. Detection of a byte counter parity error turns on the 'channel control check' latch and the write CDA operation is subsequently terminated. Detection of a 'bus out' data parity error causes turn-on of the 'channel data check' latch, but does not terminate transfer operations.

Assume that the operations described above did not result in the detection of an error, or a BCL = 0 condition. The control unit, upon receipt of the 'service out' signal, obtains the data from the 'bus out' lines and drops its 'service in' signal to the channel. When the 'service in' signal drops, the channel drops its 'service out' signal, turns off the 'write service in' latch, and waits for the next 'service in' signal from the control unit. Each time the channel receives a 'service in' signal, the channel performs the operations previously described.

For the first data doubleword of the first CCW, the first data byte to be transferred (written) may be in any B-register byte location (depending upon the byte value specified by bits 21 through 23 of the data address obtained from the CCW and subsequently entered into the byte counter). Thus, any number of data bytes from one to eight may be transferred to the control unit before data from the last B-register byte position (byte position 7) is transferred. In other words, from one to eight service in/service out exchanges between the channel and control unit can occur before all specified data from the first doubleword for the first CCW is transferred.

Assume that the service in/service out exchange continues until the BCL = 0 condition ('last word' trigger off) is detected. Upon detection of the BCL = 0 condition, the 'BCL = 0' trigger is turned on. Turn-on of this trigger turns off the 'B-register full' latch; assuming the 'A-register full' latch is on, as is the case when the second data doubleword for the first CCW is in the A-register, the 'BCL = 0' signal also turns on the 'sequence 3' latch and the channel clock.

Note: Since the BCL = 0 condition was detected as a result of the 'service in' signal, the drop of the 'service in' signal and subsequent drop of the 'service out' signal occurs for the last byte in the same manner as for other byte transfers.

With the 'sequence 3' latch on and the 'T0' clock signal active: (1) the 'A-register full' latch is turned off; (2) the 'BCL = 0' trigger is turned off; (3) the B-register is reset; and (4) the 'gate register A to B' latch is turned on. With the turn-on of the 'gate register A to B' latch, the A-register data is transferred into the B-register and the 'B-register full' latch is turned on.

At 'T0' clock time, an update count operation also begins when the count register value is gated to the adder where it is decremented by one (equivalent to eight bytes). Prior to the actual change of the count register contents, the 'T3' clock signal gates comparison logic to determine if a count of less than 17 bytes is contained in the register. If the count is less than 17, the 'last word' trigger is turned on to indicate that all data required for the first CCW of the write CDA operation is present in the channel. (For the Diagram 5-11 example, the count is greater than 17.) At 'T4' clock time, the adder is latched, and the decremented count is gated into the count register. At 'T5' clock time, the 'sequence 4' latch is turned on in preparation for an update data address operation.

The update data address operation is necessary to fetch the third data doubleword for the first CCW from storage. The update data address operation begins with the turn-off of the channel clock and subsequent turn-off of the 'sequence 3' latch. The channel clock is again turned on by the 'not sequence 3', 'sequence 4', and 'not A-register full' signals. In the meantime, the data address in the data address register is gated to the adder by the 'sequence 4' signal, where the equivalent of eight bytes is added to the data address. At 'T4' clock time, the adder is latched and gated to the data address register. At 'T6' clock time, the 'sequence 4' latch is turned off to end the update data address operation.

Assume that at this point in the write CDA operation, the count register contains a count of less than 17 bytes and that the channel must fetch one more data doubleword for the first CCW. This means that the channel must perform write CDA setup operations in preparation for fetching the last data doubleword and then the next CCW. At this point, data handling operations for the first doubleword of the first CCW are complete.

Detailed Write CDA Setup Operation.

Write CDA setup operations are performed to prepare the channel to fetch the last data word of the first CCW and then to fetch the second CCW. A special write CDA setup case exists when the count in the first CCW specifies that only one data doubleword is required to transfer all data to the control unit. Normal write CDA setup operations and the special case write CDA setup operations are described separately in the following two paragraphs.

Normal Write CDA Setup Operations. A summary and a detailed description of normal write CDA setup operations follow:

- Count register bits 19-23(+P) transferred to mark-B register 4-0 (+P) positions, respectively.
- Transfer caused by 'T2', 'sequence 4', and 'count less than 17' signals.
- 'CDA' latch turned on by 'T5' signal.
- 'Retain storage' latch turns on.

Assume that the count register contains a count of less than 17 bytes, but greater than 8 bytes; i.e., the last data doubleword of the first CCW has not been obtained from storage (Diagram 5-11). Also, assume that the 'sequence 4' latch is on, that the 'sequence 3' latch is off, and that the channel clock is on to begin an update data address operation. With the chain data flag bit active and a count of less than 17 in the count register, bits 19 through 23 and P2 of the count register are gated to bit positions 4 through 0 and P (respectively) of the mark-B register by the 'T2', 'sequence 4', and 'count less than 17' signals. With the count now in the mark-B register, the BC to CTB comparison to determine when the last data byte of the first CCW is transferred is made with count B in the mark-B register (BC = mark B condition). At 'T5' clock time, the channel's 'CDA' latch is turned on to condition channel logic for the fetch last data doubleword, first CCW, fetch second CCW, and second CCW setup operations that are subsequently performed. For example, with the turn-on of the 'CDA' latch, the address bits in the command address register plus the equivalent of eight bytes are gated to the adder for updating later in the write CDA operation.

At 'T2' clock time ('sequence 4' signal active), the 'retain storage' latch is turned on. With the 'retain storage' latch on, channel logic necessary to initiate the storage cycles for the last data doubleword of the first CCW and the second CCW is enabled. For example, the active 'retain storage' signal prevents turn-off of the 'storage request' latch until both the data doubleword and second CCW fetch storage cycles are initiated. In addition, the 'retain storage' signal enables turn-on of the 'CCW fetch' trigger after the storage cycle for the last data doubleword of the first CCW begins. With the turn on of the 'retain storage' latch, the normal write CDA setup operations are complete, and channel operations to fetch the last data doubleword of the first CCW are started.

Special Write CDA Setup Operations. Special write CDA setup operations are required when the first CCW contains a count, that when added to the DAB, specifies that less than nine bytes are required from storage for the first CCW. In other words, only one data doubleword is to be fetched for the first CCW. Detection of the count less than nine condition occurs during the last portion of the initial selection routine (Diagram 5-8). With the count less than nine condition present, a signal designated 'count

equal or less than eight bytes' is activated (ALD LT113) and AND'ed with a 'sample last word condition' signal to turn on the 'last word' trigger. (Recall that during the normal write CDA setup operations, the 'last word' trigger was not on.) In this special case, the 'sample last word condition' signal is activated by the coincidence of the following signals: 'sequence 1', 'status in', 'not sequence 5', 'not sequence 2' and 'not T1' clock time. Following the first fetch operation for the only data doubleword of the first CCW, the count register contents are gated to the mark-B register and the 'CDA' latch is turned on during the 'sequence 4' routine when a count of less than 17 bytes is detected. The 'retain storage' latch is not turned on (ALD CW113, 'not one word boundary' signal) since the 'last word' trigger is already on. Following turn-on of the 'CDA' latch, the 'CCW fetch' latch is turned on (ALD CW115, 'write CDA one doubleword' signal) and the 'storage request' latch is turned on for the CCW fetch cycle. Turn-on of the 'CCW fetch' latch completes the special case write CDA setup operation.

Detailed Fetch Last Doubleword of First CCW and Fetch CCW. A summary and a detailed description of the fetch last doubleword of first CCW and fetch CCW operation follow:

- 'Storage request' latch turns on; 'storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- Data address and storage protect bits gated to BCU interface.
- 'Address valid' latch turns on.
- Channel receives 'BCU data request' signal.
- 'Storage request' signal to BCU interface degated.
- 'BCU response' signal falls; 'remember BCU response' latch turns on.
- Channel receives 'accept' signal; 'accept' latch turns on.
- 'CCW fetch' latch turns on.
- 'BCU data request' signal falls.
- Channel receives 'BCU advance pulse' signal.
- Last data doubleword, first CCW, gated into A-register; 'A-register full' latch turns on.
- Channel receives 'BCU advance pulse' signal; 'accept' latch turns off.
- 'Storage request' signal to BCU interface activated.
- Channel receives 'BCU response' signal.
- Command address register bits gated to SAB.
- 'Storage cycle complete' signal (data fetch cycle) turns off 'retain storage' latch.
- Channel receives 'accept' signal; 'accept' latch turns on.
- Channel receives 'BCU advance pulse' signal.
- CCW checked for TIC command and proper format.
- CCW bits gated into data address, flag, and count registers.
- With no TIC command detected and no CCW errors detected, 'CCW valid' latch turns on.
- 'CCW fetch' latch turns off.
- 'Storage cycle' latch turns off.

Assuming that a normal write CDA setup operation is performed ('last word' trigger not yet on), the channel initiates operations to fetch the last data doubleword of the first CCW (Diagram 5-11). These operations begin with the turn-on of the 'storage request' latch by the 'T2' and 'sequence 4' signals. With the turn-on of the 'storage request' latch, data handling operations for the next to last data doubleword begin, and operations to fetch the last data doubleword continue with the turn-on of the 'storage cycle' trigger.

Note: For the System/360, Model 91, the channel activates the 'pre CDA' signal at this point indicating that the last doubleword of a Write CDA operation is being fetched.

With the 'retain storage' latch on, the channel enters a retain storage routine to fetch the last data doubleword. When the channel receives a 'BCU response' signal from the BCU interface, bits 0 through 20 (plus 3P) are gated from the data address register to the SAB and bits 0 through 3 (plus P) are gated from the storage protect register to the 'storage protection' lines. After a deskewing delay to allow the gated bits to stabilize, the channel turns on its 'address valid' latch. With the 'address valid' latch on, an 'address valid' signal is sent to the BCU interface

signifying that the address is on the SAB. The 'storage request' latch remains on so that the request for the CCW can be initiated as soon as possible during the retain storage routine. The channel then awaits a 'BCU data request' signal from the BCU interface. For write-type operations, data is not gated to the SBI, however, the 'BCU data request' signal is required by the channel in order to enable certain channel control logic, For example, an active 'BCU data request' signal degates the 'storage request' signal to the BCU interface and enables turn-on of the 'accept' latch when the 'accept' signal is received from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is generated in the channel by delaying the 'BCU response' signal by approximately 150 ns.

When the 'BCU response' signal to the channel falls, the channel's 'remember BCU response' latch is turned on. Turn-on of the 'remember BCU response' latch is necessary in order for the channel to initiate operations for the CCW. Upon receipt of the 'accept' signal from the BCU interface, indicating that the storage cycle to fetch the last data doubleword of the first (current) CCW has started, the channel's 'accept' latch is turned on. Subsequently, the 'CCW fetch' latch is turned on by the coincidence of the 'remember BCU response' signal and the 'accept' signal.

Note: Model 2860 Selector Channels used with the System/360, Model 91, supply a 'control word request' signal to the BCU interface with the turn-on of the 'CCW fetch' latch, indicating that the storage request is for a control word, rather than a data doubleword.

In the meantime, the channel is awaiting receipt of the 'BCU advance pulse' signal for the last data doubleword of the first CCW, indicating that the data doubleword is forthcoming on the SBO lines.

Upon receipt of the 'BCU advance pulse' signal for the last data doubleword of the first CCW, the channel delays the signal until the data doubleword is present on the SBO lines. The delayed 'BCU advance pulse' signal ('raw advance singleshot') turns on the 'SBO to A-register' latch causing the last data doubleword for the first CCW to be gated into the A-register. The 'BCU advance pulse' signal is also delayed to activate a 'storage cycle complete' signal which is AND'ed with the 'CCW fetch' signal to turn off the 'retain storage' latch. (When the 'accept' signal for the second CCW storage fetch cycle is received by the channel, the 'storage request' latch is turned off.)

With the 'storage request' latch still on (due to the active 'retain storage' signal), the fall of the 'BCU data request' signal and turn-off of the 'accept' latch by the 'BCU advance pulse' signal gates the 'storage request' signal to the BCU interface. The channel then awaits receipt of the 'BCU response' signal, indicating that the 'storage request' signal for the second CCW has been honored. When the channel receives the 'BCU response' signal it is AND'ed with the 'CCW fetch' signal to gate the contents of the command address register to the SAB. Recall that the command address register contains the address of the next CCW.

The channel then awaits the fall of the 'BCU response' signal and receipt of the 'accept' signal for the second CCW. Receipt of the 'accept' signal indicates that the storage cycle for the second CCW has begun in the channel; the 'accept' signal turns on the 'accept' latch and turns off the 'storage request' latch.

The channel then waits for the 'BCU advance pulse' signal, indicating that the second CCW is forthcoming on the SBO lines. The 'BCU advance pulse' signal is delayed, as previously described, until the CCW is on the SBO lines. While the CCW is on the SBO lines, the command bits (SBO bits 0 through 7) are sampled for a TIC command. If a TIC command is indicated, the channel enters a TIC routine, during which (1) the CCW is gated into the channel registers, (2) appropriate zero checks are made on CCW bits, and (3) a storage cycle is initiated to fetch a new CCW from the storage location specified by the data address bits obtained from the second CCW.

Assuming that a TIC command is not detected, SBO bits 37 through 39 are examined for all 0's (no error) and SBO bits 48 through 63 (count bits) are examined for a 'not all 0's' condition (no error). If an error condition is detected, the channel's 'program check' latch is turned on and the write CDA operation is terminated.

If no errors are detected, the 'raw advance SS' signal causes SBO bits 8 through 31 (+3P) to be gated into the data address register, bits 32 through 39 (+P) to be gated into the flag register, and bits 48 through 63 (+2P) to be gated into the count register.

After the CCW is gated into the channel registers, the 'CCW valid' latch is turned on (Diagrams 5-11) by coincidence of the 'late advance pulse' and 'CCW fetch' signals. Subsequently, the 'CCW fetch' latch is turned off by coincidence of the 'late advance pulse' signal and the 'CCW valid' signal. With the turn-off the the 'CCW fetch' latch, the fetch last data doubleword of the first CCW and fetch second CCW operation is complete. At this point, the channel enters the data handling routine (already in progress) for the next to the last data doubleword for the first CCW. Recall that data handling operations for this data doubleword began with turn-on of the 'B-register full' latch when the second data doubleword for the first CCW was transferred into the B-register.

Detailed Data Handling, Next to Last Data Doubleword, First CCW. Data handling operations for the next to last data doubleword are similar to those described for the

first data doubleword. The control unit and channel exchange 'service in' and 'service out' signals to transfer B-register data bytes until the last data byte is transferred (BCL = 0 condition is detected). (Recall that, unless a special case write CDA setup operation was performed, the 'last word' trigger is still off.) Upon detection of the BCL = 0 condition, the channel turns on the 'BCL = 0' trigger and turns off the 'B-register full' latch (Diagram 5-11). With the last data doubleword of the first CCW in the A-register, the 'sequence 3' latch and channel clock are turned on.

Since the significant bits of the count are now in the mark-B register, and the three low-order bits already indicate the number of bytes to be transferred to the control unit from the B-register, an update count operation is not performed when the 'sequence 3' latch is turned. At this point in the write CDA operation, the 'sequence 3' latch is turned on so that the 'last word' trigger may be turned on and a 'storage request' signal to fetch the first data doubleword of the second CCW may be initiated by the channel.

At 'T0' clock time, the 'A-register full' latch is turned off, the last data doubleword of the first CCW is transferred to the B-register, and the 'B-register full' latch is turned on. Recall that the 'CDA' latch is on and that the 'BC = CTB' trigger has not yet been turned on. The 'last word' trigger is turned on by AND'ing the 'sequence 3', 'CDA latch', and 'T4' clock signals. Turn-on of the 'last word' trigger indicates that the last data doubleword is in the B-register and completes the data handling, next to last data doubleword, first CCW operations.

Detailed Fetch First Data Doubleword, Second CCW. With the 'CCW valid' latch on, the 'last word' trigger on, the 'sequence 3' latch on, and the 'storage cycle complete' signal inactive, the 'storage request' latch is turned on (Diagram 5-11). With this latch on, a 'storage request' signal is supplied to the BCU interface to begin operations for fetching the first data doubleword of the second CCW. When the 'storage request' latch is turned on, the 'sequence 3' latch is turned off.

Operations to fetch the first doubleword of the second CCW proceed with the turn-on of the 'storage cycle' trigger to activate applicable channel control logic for the fetch operation. With the 'storage request' signal active at the BCU interface, the channel waits for a 'BCU response' signal from the BCU interface. Upon receipt of the 'BCU response' signal, the channel gates the data address from the data address register to the SAB and the 'storage protect' key from the storage protect register to the 'storage protection' lines. The channel then waits for a 'BCU advance pulse' signal and, when received, delays it until the first data doubleword for the second CCW is on the SBO lines. When the data doubleword is available, the 'raw advance SS' signal gates the data into the A-register and turns on the 'A-register full' latch. With the 'sequence 1' latch off, the channel enters the data handling routine for the last doubleword of the first CCW. Recall that the data handling operations have already begun with the turn-on of the 'B-register full' latch when the last doubleword was transferred to the B-register.

Detailed Data Handling, Last Doubleword, First CCW. Data handling operations for the last doubleword of the first CCW (Diagram 5-11) proceed in the manner described in previous text; i.e., individual bytes from the B-register are transferred to the control unit due to the exchange of 'service in' and 'service out' signals. Recall that, at this point in the write CDA operation, the 'last word' trigger is on, the 'CDA' latch is on, and that the significant count bits are in the mark-B register. As each data byte is transferred, the byte count is compared with the count-B in the mark-B register for a BC = mark-B condition.

When the BC = mark-B condition is detected, it indicates that the last byte of the last data doubleword for the first CCW has been transferred to the control unit. In order words, all data bytes specified by the first CCW have been transferred. Detection of the BC = mark-B condition causes the 'BC = CTB' trigger to turn on, the 'B-register full' latch to turn off, and the byte counter to be reset. With the byte counter reset, data handling for last data doubleword, first CCW operations are complete and the channel begins the second CCW setup operations.

Detailed Second CCW Setup Operations. A summary and a detailed description of the second CCW setup operations follow:

- 'Sequence 3' latch and channel clock turned on.
- 'A-register full' latch turned off.
- A-register contents transferred to B-register.
- 'B-register full' latch turned on.
- 'Write CDA command end' latch turned on.
- 'Latch adder command address' trigger turned on.
- Command address plus 1 value latched into adder.
- Adder output gated to command address register; 'latch adder command address' trigger turned off.
- 'DAB plus count' gated to adder.
- Adder latched; adder output gated to count register.

- DAB gated to byte counter; DAB bits in data address register reset.
- 'Sequence 4' latch turns on.
- 'DAB plus count' value sampled for count less than nine condition.
  1. 'Last word' trigger remains on if count less than nine.
  2. 'Last word' trigger turned off if count greater than nine.
- 'BC = CTB' trigger turns off.
- 'CDA' latch turns off.
- 'Write CDA command end' latch turns on.
- Data address register contents plus one gated to adder.
- Adder latched; adder output gated to data address register.
- If 'last word' trigger on, 'CDA' latch and 'CCW fetch' latch turned on; 'storage request' signal initiated for next CCW.
- If 'last word' trigger off, channel initiates 'storage request' signal for next data doubleword.

Second CCW setup operations are performed to complete the necessary CCW calculations required before data transfer operations specified by the second CCW can begin. Assume that the BC = mark-B condition has been detected, and that the first data doubleword of the second CCW is in the A-register. With the 'A-register full' latch on, the 'sequence 3' latch and channel clock are turned on by the coincidence of the 'write', 'CDA latch', 'last word trigger', 'A-register full' and 'BC = CTB' signals.

Following turn-on of the 'sequence 3' latch and the channel clock, the 'A-register full' latch is turned off, the first data doubleword for the second CCW is gated from the A-register to the B-register, and the 'B-register full' latch is turned on. The first data doubleword for the second CCW is now in the B-register, ready for data byte transfers to begin after completion of the second CCW setup operations.

Recall that the 'CDA' latch is still on and the 'BC = CTB' trigger is on. When these conditions are present, the 'write CDA command end' latch is turned on. With this latch on, the 'T2' clock signal turns on the 'latch adder command address' trigger, causing the command address (plus the equivalent of eight bytes) to be latched into the adder and the 'gate command address +1 to adder' signal to drop. At 'T0' clock time, the updated command address is gated from the adder into the command address register. At 'T1' clock time, the 'latch adder command address' trigger is turned off to clear the updated command address from the adder. With the above operations complete, the command address register contains the address of the next consecutive CCW.

At 'T0' clock time, the three low-order bits of the DAB plus the count in the count register are gated to the adder. (This operation occurs while the updated command address is being gated from the adder latches to the command address register.) In addition, the DAB is gated into the byte counter where it is used to specify from which B-register byte location the first data byte of the first data doubleword will be gated. At 'T4' clock time, the DAB plus count value is latched into the adder, and gated to the count register. The count register now contains a modified count which is used to detect the last byte transfer for the last data doubleword of the second CCW.

After the DAB plus count operation, the DAB in the data address register is reset to all 0's (doubleword boundary) since all data bytes for subsequent data doublewords (with the possible exception of the last data doubleword) will be transferred to the control unit. Prior to the DAB reset, the DAB bits are examined to determine if the data address register P2 bit will be affected by the reset. If the P2 bit is affected, the P2 bit value is corrected at the 'P2-bit' latch.

At 'T6' clock time, the updated count in the count register is sampled for a last word condition (count of less than nine). Detection of a last word condition prevents turn-off of the 'last word' trigger, indicating that the one data doubleword required for the second CCW is in the B-register. If the count is greater than eight bytes, the 'last word' triggger is turned off.

The 'sequence 4' latch is also turned on by the 'T6' clock signal so that the channel may update the data address register contents prior to fetching the second data doubleword (if required). With both the 'sequence 3' and 'sequence 4' latches on, the 'BC = CTB' trigger is turned off, causing the 'CDA' latch to turn off. With the drop of the 'CDA' signal, the 'write CDA command end' latch is turned off to end the 'second CCW setup' operations. The channel must now perform an update data address operation, turn on the 'storage request' trigger to fetch the second data doubleword, and begin data handling operations for the first data doubleword of the second CCW.

If the 'last word' trigger is still on, (as would be the case if the count register contained a modified count of less than nine), data handling operations will begin (assuming the 'CDA flag' bit in the second CCW is not active) following turn-on of the 'sequence 3' latch (Diagram 5-11). If the CDA flag bit is active with a count of less than nine, the 'CDA' latch and 'CCW fetch' latch are turned on, and a 'storage request' is initiated for the next CCW. In this case, data handling operations for the

first data doubleword of the second CCW begin following turn-off of the 'write CDA command end' latch from the previous CDA operation.

In summary, the detailed write CDA operations performed as described in the preceding paragraphs are repeated each time the new CCW contains an active chain data flag bit. When a CCW is obtained by the channel without an active chain data flag, the write CDA operation ends in the same manner as described for a write operation. That is, once the last data byte has been transferred to the control unit, the channel enters a 'sequence 5' routine to obtain status and clear the I/O interface. Subsequent to the 'sequence 5' routine, the channel enters an interrupt routine to store the channel status word and present the unit address and appropriate condition code to the CPU interface.

Read Chain Data Operation

Read chain data (read CDA) operations permit the channel to transfer data specified by more than one CCW from the control unit selected during the initial selection routine to main storage without interrupting the CPU; the read CDA operation also permits data from a continuous record on an I/O device to be stored in different areas of main storage. In general, the channel controls the transfer of all data specified by the first CCW obtained during the initial selection routine; then, since the first CCW specifies a read CDA operation, the channel fetches the second CCW from storage and controls the read transfer of all data specified by the second CCW. As long as each new CCW specifies a read CDA operation, the channel will continue to fetch CCW's and control the transfer of data specified by the CCW's. (For example, if the second CCW specifies data chaining, the channel will fetch a third CCW at the appropriate time and control the read transfer of data specified by that CCW. Read data chaining also permits each CCW to specify a different starting address location in storage, thus specifying the storage area into which data for that CCW will be stored.)

Read CDA operations are described in the following paragraphs at three levels: (1) the first level ("Basic Read CDA Operations") introduces the basic operations performed; (2) the second level ("Simplified Read CDA Operations") presents a simplified description of the basic read CDA operations; and (3) the third level ("Detailed Read CDA Operations") presents a detailed description of the read CDA operations.

Basic Read CDA Operations

- For first read sequence, channel:
  1. Loads B-register on byte-by-byte basis until BCL = 0 condition detected.
  2. Transfers B-register contents to A-register.
  3. Decrements value in count register eight bytes.
  4. Controls storage of A-register data in main storage.
  5. Increments (read) or decrements (read backward) value in data address register.
- Each read sequence identical until next to last read sequence.
- During next to last read sequence, 'last word' trigger is turned on.
- During last read sequence, BC = CTB detected when last data byte is gated into B-register.
- Channel performs read CDA setup operations.
  1. Transfers last doubleword from B-register to A-register.
  2. 'CDA' latch turned on.
  3. 'Retain storage' latch turned on.
  4. Byte counter reset.
  5. 'Read CDA doublegating' signal activated.
- Channel free to gate data bytes into B-register while storing last data doubleword, first CCW, and fetching second CCW.
- Before second CCW is received by channel, data bytes (if received from the control unit) are gated into B-register (read forward) as follows:
  1. First three bytes doublegated to positions 0 and 4, 1 and 5, and 2 and 6, respectively. (Mark-B register bits gated accordingly).
  2. Byte positions 4-7 of B-register reset (Mark-B register bits 4-7 also reset).
  3. Other bytes received gated to B-register positions 3-7, sequentially.
  4. When B-register full, contents transferred to A-register.
  5. Channel can singlegate eight more data bytes into B-register, then overrun condition occurs if CCW has not yet been received by the channel.
- Channel receives second CCW and checks for TIC command and proper formatting; TIC command causes new CCW to be fetched and improper formatting causes termination of operation.
- Channel gates data address, flag, and count bits into channel registers.
- 'CCW valid' latch turns on.
- Channel compares DAB and count values from new CCW against bytes received by channel for operating compatibility.

- Non-compatibility terminates operation.

- Assuming compatibility:
  1. Command address register contents incremented.
  2. DAB added to count and gated to count register.
  3. DAB gated to byte counter.

- Channel resumes read operation under new CCW control.

The channel begins read CDA operations (Diagram 5-12) when a 'start I/O' signal is received from the CPU interface, an initial selection routine is completed, the command from the CCW specifies a read operation, and the CDA bit in the CCW is active. The following description is provided to introduce the basic read CDA operations.

The channel enters the read CDA operation from the setup completion portion of the initial selection routine (see "Setup Completion" in this Chapter). With the 'read' signal and 'CDA flag' bit active and the 'sequence 2' latch on, the channel enters the B-register data handling loop operation. During this operation, the B-register is loaded with data from the I/O device one byte at a time until the BCL = 0 condition is detected; the BCL = 0 condition specifies that a data byte has been loaded into the last B-register position (B-register is full).

With the B-register full, the channel performs parallel operations to: (1) gate the B-register contents to the A-register (B-register to A-register operation); (2) store the A-register contents in main storage; and (3) decrement the count in the count register eight bytes so that the count will indicate that the parallel storage operation has occurred (update count operation). At the completion of the B-register to A-register operation, data storage operation, and update count operation, the channel performs a data address update operation to increment (read) or decrement (read backward) the address in the data address register an amount equivalent to eight bytes. The data address update operation is performed to provide a sequential address location at which the next data doubleword is to be stored. With the completion of the data address update operation, the first read sequence is complete and the channel may begin the next read sequence (Diagram 5-12).

The channel continues to perform the operations previously described for each read sequence until the next to the last read sequence; during this sequence, the 'last word' trigger is turned on. For the last read sequence, the B-register data handling loop operation loads bytes from the I/O device into the B-register until a BC = CTB condition is detected; this indicates that the last data byte specified by the original count has been gated into the B-register. When the BC = CTB condition is detected, the channel performs the read CDA setup operations. During the read CDA setup operation, the channel: (1) transfers the last data doubleword for the first CCW to the A-register; and (2) prepares channel logic to store the last data doubleword for the first CCW and then fetch the second CCW. Channel logic preparation includes turn-on of the 'storage request' latch (which remains on until the storage request for the second CCW has been honored), turn-on of the 'CDA' and 'retain storage' latches, reset of the byte counter, turn-off of the 'CCW valid' latch and 'last word' trigger, and activation of the 'read CDA doublegating' signal. When the read CDA setup operations are complete, the channel is conditioned to: (1) start gating bytes into the B-register for the first data doubleword of the second CCW on a doublegating basis; and (2) begin operations to store the last data doubleword of the first CCW and fetch the second CCW.

Since the channel cannot anticipate which byte location within the first data doubleword will be specified as the starting byte location by the second CCW, data byte gating into the B-register for the first data doubleword is performed in a manner which allows the CCW to specify other than doubleword boundaries (with certain limitations). If the channel receives data bytes from the control unit before the second CCW arrives in channel, the first three data bytes received are doublegated into the B-register. For a read (forward) operation, the first data byte is gated to positions 0 and 4, the second data byte to positions 1 and 5, and the third data byte to positions 2 and 6. After the third data byte is gated into the B-register, doublegating is turned off, and B-register byte positions 4 through 7 are reset (as well as mark-B register bits 4 through 7). If subsequent data bytes are received from the control unit, they are singlegated into B-register byte positions 3 through 7 in that order (fourth byte is gated to position 3, fifth byte to position 4, sixth byte to position 5, seventh byte to position 6 and eighth byte to position 7). For a read backward CDA operation, gating into the B-register is the reverse of that described for the read operation.

While B-register data handling, first data doubleword, second CCW operations are in progress, the channel (in the retain storage mode) stores the last data doubleword of the first CCW at the storage location indicated by the address in the data address register. Following the data store operation ('storage request' latch is still on but the 'storage request' signal to the BCU interface is degated while the 'BCU data request' or 'accept latch' signals are active), the channel turns on the 'CCW fetch' latch and fetches the second CCW from the storage location indicated by the address in the command address register. When the second CCW is available to the channel on the SBO lines, data address, flag, and count bits are gated into the respective channel registers. While the CCW is on the SBO lines, the command bits are checked for a TIC command and specified CCW bits are checked for programming validity. Assuming no errors are detected, the 'CCW valid' latch is turned on indicating that the second

CCW is in channel. Since setup operations must be performed for the second CCW, the channel inhibits further gating into the B-register ('inhibit service out' signal) until the setup operations are complete. The 'CCW fetch' latch is also turned off, signifying the end of the fetch second CCW operation.

The second CCW setup operations are performed to: (1) update the command address register in the event a third CCW is required; (2) determine whether a chain check condition exists by comparing the count in the byte counter (number of bytes gated into B-register) with the data address register's three low-order bits (DAB), and by examining the count obtained from the second CCW; (3) add the DAB to the count register contents and return the sum to the count register; (4) OR the DAB into the byte count register, thereby effectively adding the DAB to the count already existing in the byte counter; and (5) turn off the 'CDA' latch and deactivate the 'inhibit service out' signal to permit data handling operations to resume.

A chain check condition detected during the second CCW setup operation results in termination of the read CDA operation and can be caused by any of the following conditions: (1) the DAB does not specify doubleword boundaries, and the first data doubleword has been gated into the B-register and subsequently transferred to the A-register (no chain check condition is detected if the DAB specifies doubleword boundaries), the first doubleword is in the A-register, no more than eight bytes (second data doubleword) have been gated into the B-register, and the count is 16 or greater); (2) the count in the CCW is less than the number of bytes already in the B-register; (3) at least one data byte has been doublegated, doublegating is still active, and the DAB does not specify singleword (DAB = 4 for read or three for read backward) or doubleword (DAB = 0 for read or seven for read backward) boundaries; and (4) data bytes are in the B-register, the 'read CDA doublegating' signal has been turned off, and the DAB does not specify doubleword boundaries. (See "Read CDA Encoding" in Chapter 2.) Assuming that a chain check condition is not detected (and the first doubleword is not in the A-register), the channel turns off the 'inhibit service out' signal, and B-register data handling loop, first data doubleword, second CCW operations continue until the B-register is full.

With the B-register full, the channel performs the B-register to A-register, data storage and update count operations as previously described to store the first data doubleword of the second CCW. After performing the above operations, the channel performs an update data address operation to complete the first read sequence of the second CCW. From this point, the channel continues normal read operations. That is, for each data doubleword specified by the second CCW, the B-register is loaded (singlegating) on a byte-by-byte basis until the register is full; then the B-register to A-register, data storage, update count and update data address operations are performed. If the second CCW contains an active CDA flag bit, read CDA operations are continued by fetching a third CCW. Otherwise, when the last data doubleword of the second CCW is being stored, the channel enters a sequence 5 ending routine to obtain a status byte from the control unit and clear the I/O interface. After the sequence 5 routine is complete, the channel enters an interrupt routine to store the channel status word and to send the unit address register contents and condition code 0 to the CPU interface.

Simplified Read CDA Operations

A simplified read CDA operation is illustrated in flow chart form in Diagram 5-13. In this diagram, the simplified read CDA operations are divided into subroutines indicated by dashed lines (initial conditions and initial status cleanup; data handling, first data doubleword, first CCW, etc.). Each subroutine indicated on the flow chart is described in separate paragraphs below.

Initial Conditions and Initial Status Cleanup. This paragraph describes the channel conditions at the beginning of the read CDA operation and subsequent operations performed to clear initial status conditions which are present at the completion of the initial selection routine. At the completion of the initial selection routine, the following conditions are present in the channel: (1) the 'sequence 1' and 'sequence 2' latches are on; (2) the 'sequence 3', 'sequence 4', and 'sequence 5' latches are off; (3) the 'status in' signal from the control unit is active; (4) command register bit 5 or bit 6 is active, and bit 7 is inactive to specify a read command; (5) the 'gate command out' latch is on; (6) the CDA bit is active; and (7) the channel clock is on. In addition, for the Diagram 5-13 example, the count from the first CCW specifies that 13 data bytes are to be transferred from the control unit to storage beginning with the byte at storage address location 2006. Since data is stored on doubleword boundaries, three data doublewords must be sent to storage to store the specified 13 data bytes. (The last two bytes of the first data doubleword, all eight bytes of the second data doubleword, and the first three bytes of the last doubleword are stored.)

Before the actual read CDA operation can commence, the channel must perform initial selection routine cleanup operations. These operations are performed to remove the initial status in condition which was presented to the channel by the control unit at the end of the initial selection routine. To remove the initial status in condition, the channel generates a 'service out' signal to the control unit, causing the control unit to drop the 'status

in' signal. With the drop of the 'status in' signal, the channel drops the 'service out' signal and turns off the 'sequence 1' latch. At this point, the channel is conditioned to control read CDA data flow, and enters the data handling, first data doubleword, first CCW operation (Diagram 5-13).

Data Handling, First Data Doubleword, First CCW. A summary and a simplified description of the data handling, first data doubleword, first CCW operations follow:

- Channel receives 'service in' signal and activates 'service out' signal.
- Data byte from control unit latched into 'bus in' latches.
- Data byte gated into B-register.
- Byte counter advanced.
- Exchange of 'service in' and 'service out' signals continues until BCL = 0 condition is detected.
- BCL = 0 condition turns on 'BCL = 0' latch and 'B-register full' latch.
- 'Sequence 3' latch and clock turn on.

This paragraph describes operations to gate data bytes from the control unit into the B-register byte positions for the first data doubleword of the first CCW. (Note that, for the Diagram 5-13 example, only two data bytes are to be gated into the B-register for the first data doubleword.) When the control unit has a data byte available, it sends a 'service in' signal to the channel. The 'service in' signal causes the channel to activate the 'service out' signal which: (1) raises the 'service out' signal to the control unit, indicating that the channel accepts the data byte presented; (2) latches the data byte into the 'bus in' latches prior to the fall of the 'service in' signal from the control unit; (3) steps the byte counter to indicate receipt of the data byte; (4) gates the data byte into the B-register byte position indicated by the count in byte counter (gating into the B-register occurs before stepping of the byte counter is complete indicating that the data byte will be gated into the byte position specified by the byte counter count prior to changing the byte counter output. See "Byte Counter Circuits" in Chapter 2); and (5) sets the mark-B register bit corresponding to the B-register byte position.

When the data byte is gated into the B-register, the BCL = 0 logic is gated to determine if the present data byte is being gated into the last B-register byte position. Assuming the BCL = 0 condition is not detected (and the 'service in' signal from control unit has dropped, causing the channel's 'service out' signal to drop), the channel waits for another 'service in' signal from the control unit. When the 'service in' signal arrives, the channel operations described above are again performed. The exchange of 'service in' and 'service out' signal continues until the BCL = 0 condition is detected to turn on the 'BCL = 0' latch. (For the example in Diagram 5-13, the BCL = 0 condition is detected when the second data byte has been gated into the B-register.) The resulting 'BCL = 0' signal turns on the 'B-register full' latch which inhibits the channel from responding to a 'service in' signal from the control unit; in effect, this stops the 'B-register data handling loop' operation. In addition, the 'BCL = 0' signal initiates turn-on of the 'sequence 3' latch, causing the channel to begin three parallel operations: transfer first data doubleword of first CCW to A-register, store first data doubleword of first CCW, and update count, first data doubleword, first CCW. With the turn-on of the 'sequence 3' latch, the channel clock is turned on to time related events performed during the three parallel operations. With the clock on, data handling, first data doubleword, first CCW operations are complete.

Transfer First Data Doubleword of First CCW to A-Register. A summary and a simplified description of the transfer first data doubleword of first CCW to A-register operations follow:

- 'BCL = 0' signal causes activation of 'gate B-register to A-register' and 'gate mark-B to mark-A register' signals.
- B-register contents transferred to A-register; mark-B register bits transferred to mark-A register.
- 'A-register full' latch turns on, 'B-register full' latch turns off, 'BCL = 0' latch turns off, and B-register reset.
- Channel can begin gating data bytes into B-register.

The transfer first data doubleword of first CCW to A-register operations (Diagram 5-13) actually begin when the BCL = 0 condition is detected. The 'BCL = 0' signal causes activation of the 'gate B-register to A-register' signal which gates data from the B-register to the A-register on a byte position-to-byte position basis; the 'BCL = 0' signal also causes activation of the 'gate mark-B to mark-A register' signal which gates bits from the mark-B register to the mark-A register. At 'T2' clock time, the 'A-register full' latch is turned on indicating that the data in the A-register and bits in the mark-A register are stable and available for gating to storage. At the same time, the 'BCL = 0' latch is

turned off, the B-register is reset, and the 'B-register full' latch is turned off. With the turn-off of the 'B-register full' latch, the channel can again respond to 'service in' signals with a 'service out' signal; thus, the next B-register data handling loop operation may begin, even though the parallel data storage and update count operations are not complete. At this point in the transfer first data doubleword of the first CCW to A-register operation, the 'A-register full' latch remains turned on until the channel receives the 'accept' signal for the parallel store first data doubleword of first CCW operation.

Store First Data Doubleword of First CCW. A summary and a detailed description of the store first data doubleword of first CCW follow:

- 'Storage request' latch turns on.
- 'Storage cycle' trigger turns on.
- Channel receives 'BCU response' signal:
  1. Data address register bits 0-20 (+3P) gated to SAB.
  2. Storage protect register bits 0-3 (+P) gated to 'storage protection' lines.
  3. Mark-A register bits gated to mark lines.
  4. 'Store' signal activated to BCU interface.
- Channel receives 'BCU data request' signal; A-register data gated to SBI lines.
- Channel receives 'accept' signal; 'A-register full' latch turns off.

The parallel store first data doubleword of first CCW operation (Diagram 5-13) begins with the turn-on of the 'storage request' latch by the 'T0' clock signal. This raises the 'storage request' signal to the BCU interface, requesting a storage cycle to store the data contained in the A-register. In addition, the 'storage request' signal turns on the 'storage cycle' trigger to activate and inhibit channel logic as required for the storage cycle. When the channel receives the 'BCU response' signal from the BCU interface ('storage request' signal has been honored), the channel: (1) gates data address register bits 0 through 20 (+3P) to the SAB to specify the address location in main storage at which the A-register contents are to be stored; (2) gates storage protect register bits 0 through 3 (+P) to the 'storage protection' lines to permit access in main storage to the storage location specified by the data address on the SAB; (3) gates mark-A register bits to the BCU interface mark lines to specify which data bytes in the A-register are to be stored (an active mark bit specifies that the corresponding A-register byte is to be stored); and (4) activates the 'store' signal to the BCU interface, indicating that the purpose of the requested storage cycle is to store information.

Subsequently, the channel receives a 'BCU data request' signal from the BCU interface which causes the channel to gate the contents of the A-register to the SBI lines for storage in main storage.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is not supplied via the BCU interface but is generated in the channel by delaying the 'BCU response' signal approximately 150 ns.

When the storage cycle progresses past the point where A-register data is no longer required on the SAB, the 'BCU data request' signal falls to degate the data from the SAB. When the 'accept' signal is received, the 'A-register full' latch turns off. Turn-off of this latch, and subsequent receipt of the 'BCU advance pulse' signal, complete the store first data doubleword of first CCW operation.

Update Count, First Data Doubleword, First CCW. The parallel update count, first data doubleword, first CCW operation (Diagram 5-13), begins at the same time the store first data doubleword of first CCW operation begins. At 'T0' clock time, the count register contents are gated to the adder where the equivalent of eight bytes is subtracted from the count ('count minus 1') to adder. At 'T4' clock time, the adder is latched and the contents of the adder gated back to the count register. At 'T5' clock time, the channel's 'sequence 4' latch is turned on and the channel clock is turned off, providing the parallel storage cycle has progressed to the point where the 'A-register full' latch is off. This completes the update count, first data doubleword, first CCW operation, with the count now reflecting that one data storage operation has occurred. Note that, for the Diagram 5-13 example, the modified count register value before the update operation equaled 19 (13 hex) and that following the update operation the count value equaled 13 (B hex).

Update Data Address, First Data Doubleword, First CCW. Update data address, first data doubleword, first CCW operations follow completion of the update count, first data doubleword, first CCW operation. After the 'sequence 4' latch is turned on, the 'sequence 3' latch is turned off and the channel clock is turned on (Diagram 5-13). With the 'sequence 4' latch on, the contents of the data address register are gated to the adder where eight bytes are added (data address plus one to adder). (For a read backward operation, eight bytes are subtracted from the data address register contents.) At 'T4' clock time, the adder is latched, and the adder contents gated back into the data address register. At 'T6' clock time, the 'sequence 4' latch is turned off to end the update data

address, first data doubleword, first CCW operation. With the operation completed, the data address register now contains the sequential address location at which the next data doubleword is to be stored.

Second Data Doubleword for First CCW Operations. Second data doubleword for first CCW operations (Diagram 5-13) are similar to the operations described in the preceding paragraphs for the first data doubleword of the first CCW. Recall that when the 'B-register full' latch was turned off during the transfer first data doubleword of first CCW to A-register operation, the data handling, second data doubleword, first CCW operation began; that is, the transfer of data bytes from the control unit to the B-register began due to the exchange of 'service in' and 'service out' signals between the control unit and the channel. When the B-register is full, the three parallel operations (transfer second data doubleword, first CCW, to A-register, store second data doubleword, first CCW, and update count register) are performed in the manner described for the first data doubleword. However, since (in the Diagram 5-13 example) the second data doubleword assembled in the B-register is the next to the last data doubleword to be transferred to storage for the first CCW, the count in the count register indicates less than 17 bytes (equal to or less than two doublewords). Before the count minus one value is latched into the adder, the count is decoded by channel logic and the 'last word' trigger is turned on. When the update count register operation is completed, the 'sequence 4' latch is turned on and the update data address operation is performed. With the completion of the update data address operation, second data doubleword for first CCW operations are complete, and the channel proceeds with the data handling, last data doubleword, first CCW operations.

Data Handling, Last Data Doubleword, First CCW. A summary and a simplified description of the data handling, last data doubleword, first CCW operations follow:

- Control unit and channel exchange 'service in' and 'service out' signals to gate into B-register until BC = CTB detected.
- 'BC = CTB' trigger turns on.
- 'B-register full' latch turns on.
- Count register is reset to zero count.
- 'Delay service in' latch turns on.
- 'Suppress out' signal is raised to control unit.
- 'Sequence 3' latch turns on.

Data handling operations for the last data doubleword of the first CCW (Diagram 5-13) begin with the turn-off of the 'B-register full' latch during the transfer second data doubleword, first CCW, to A-register operation. With the 'last word' trigger on, the channel is monitoring the gating of bytes into the B-register for a BC = CTB condition. Recall that for the Diagram 5-13 example, three data bytes are to be gated into the B-register for the last data doubleword of the first CCW. When the last data byte is gated into the B-register, the BC = CTB condition is detected to turn on the 'BC = CTB' trigger, reset the count register, and turn on the 'B-register full' latch. Turn-on of the 'B-register full' latch inhibits further exchanges of 'service in' and 'service out' signals between the control unit and the channel, thus preventing further gating of data bytes into the B-register. However, because the chain data flag bit is active, the channel's 'delay service in' latch is turned on while the last data byte is being gated into the B-register. Turn-on of the 'delay service in' latch raises the 'suppress out' signal to the control unit. If the control unit is servicing a buffered I/O device or a start-stop I/O device, the 'suppress out' signal prevents 'service in' signals from being generated until the 'suppress out' signal is dropped. Thus, the possibility of an overrun condition occurring in the channel due to the gating of too many data bytes into the channel before the second CCW is in the channel is reduced. (The 'suppress out' signal is dropped considerably later than the turn-off of the 'B-register full' latch; recall that the channel is able to respond to 'service in' signals when the 'B-register full' latch is turned off.) The 'suppress out' signal does not prevent non-bufferred or other than start-stop I/O devices from raising a 'service in' signal.

With the 'B-register full' latch on and the BC = CTB condition detected, the channel's 'sequence 3' latch is turned on to end the data handling, last data doubleword, first CCW operation. The channel is now ready to perform the read CDA setup operations.

Read CDA Setup Operations. A summary and a simplified description of the read CDA setup operations follow:

- Last data doubleword transferred from B-register to A-register.
- Mark-B register bits transferred to mark-A register.
- 'A-register full' latch turns on.
- 'Storage request' latch turns on.
- 'Storage cycle' trigger turns on.
- 'CDA' latch turns on.
- 'Read CDA doublegating' signal activated.

- 'Retain storage' latch turns on.
- Byte counter is reset to zero.
- 'CCW valid' latch turns off.
- 'Last word' trigger turns off.
- 'Delay service in' latch turns off; 'suppress out' signal drops.

The read CDA setup operations (Diagram 5-13) are performed to: (1) prepare the channel to store the last data doubleword of the first CCW; (2) fetch the second CCW; and (3) prepare the channel to begin accepting data bytes for the first data doubleword of the second CCW. Read CDA setup operations begin with the turn-on of the channel clock.

At 'T0' clock time, the channel transfers the last data doubleword for the first CCW from the B-register into the A-register, transfers the bits from the mark-B register into the mark-A register, and turns off the 'B-register full' latch. Transfer of the last data doubleword into the A-register prepares the channel to gate the data doubleword to main storage during the store operation following the read CDA setup operations. At 'T2' clock time, the 'A-register full' latch is turned on, signifying that the last data doubleword is in the A-register.

In preparation for storage of the last data doubleword and subsequent fetch of the second CCW, the 'storage request' latch is turned on by the 'T0' clock signal. Turn-on of the 'storage request' latch turns on the 'storage cycle' trigger which activates and inhibits appropriate channel logic for the data store and subsequent CCW fetch operations. Both the 'storage request' latch and 'storage cycle' trigger remain on until the last data doubleword has been stored and the fetch request for the second CCW has been honored. (However, the 'storage request' signal to the BCU interface is degated while the 'BCU data request' and 'accept latch' signals are present in the channel.)

Note: For 2860 Selector Channels used with the System/360, Model 91, the channel activates the 'pre CDA' signal to the Model 81 following turn-on of the 'storage request' latch.

The 'T0' clock signal also turns on the CDA latch. Turn-on of the 'CDA' latch then activates the 'read CDA doublegating' signal to the byte count encoder, permitting the channel to doublegate data bytes from the control unit into the B-register. (Doublegating operations may begin when the 'B-register full' latch is turned off or, in the case of devices affected by the 'suppress out' signal, when the 'suppress out' signal is deactivated.)

With the 'CDA' latch on, the 'retain storage' latch is turned on when the 'storage request' latch is turned on. The resulting 'retain storage' signal prevents turn-off of the 'storage request' latch until the fetch request for the second CCW has been honored.

At 'T2' clock time, following turn-on of the 'CDA' latch, the byte counter is reset to zero in preparation for controlling the gating of data bytes into the B-register until the second CCW is in the channel. At the same time the byte counter is reset, the 'CCW valid' latch is turned off. Turn-off of the 'CCW valid' latch indicates that a valid CCW is not in the channel, and that any data byte gated into the B-register is under control of channel logic; that is, channel logic has been conditioned to control data byte gating until the second CCW is in the channel.

With the turn-off of the 'CCW valid' latch, the 'last word' trigger is turned off. At 'T6' clock time, the channel's 'delay service in' latch is turned off to drop the 'suppress out' signal to the control units. With the drop of the 'suppress out' signal, the read CDA setup operations are complete. At this point, the channel is conditioned to: (1) store the last data doubleword of the first CCW and fetch the second CCW; (2) at the same time, control the gating of data bytes from the control unit into the B-register; and (3) remain in a 'sequence 3' condition until the 'CCW valid' latch is turned on.

Store Last Data Doubleword, First CCW. A summary and a simplified description of the store last data doubleword, first CCW operations follow:

- Channel receives 'BCU response' signal.
- Data address register bits gated to SAB, storage protect key to 'storage protection' lines, mark-A register bits to 'mark' lines and 'store' signal to BCU interface.
- Channel receives 'BCU data request' signal.
- A-register contents gated to SBI lines: 'storage request' signal to BCU interface degated.
- 'BCU response' signal falls; 'remember BCU response' latch turns on.
- Channel receives 'accept' signal; 'accept' latch turns on.
- 'CCW fetch' latch turns on.
- Channel receives 'BCU advance pulse' signal.

Channel operations to store the last data doubleword of the first CCW began with the turn-on of the 'storage request' latch and 'storage cycle' trigger during the read CDA setup operations. With the turn-on of the 'storage

cycle' trigger, the channel enters a retain storage routine and awaits a 'BCU response' signal from the BCU interface (Diagram 5-13). Upon receipt of the 'BCU response' signal, the channel gates address information to the BCU interface (data address register bits 0 through 20 plus 3P to the SAB, mark-A register bits 0 through 7 plus P to the BCU interface 'mark' lines, and storage protect register bits 0 through 3 plus P to the 'storage' protection lines). In addition, the channel activates the 'store' signal to the BCU interface.

Subsequent to receipt of the 'BCU response' signal, the channel receives a 'BCU data request' signal from the BCU interface. This signal causes the channel to gate the last data doubleword for the first CCW from the A-register to the SBI lines for storage at the address location in main storage indicated by the data address on the SAB. While active at the channel, the 'BCU data request' signal also degates the 'storage request' signal to the BCU interface. (The 'storage request' signal remains degated until both the 'BCU data request' and 'accept latch' signals are inactive.)

Note: When the 2860 Selector Channel is used with the System/360, Model 91, the 'BCU data request' signal is generated in the channel by delaying the 'BCU response' signal by approximately 150 ns.

When the 'BCU response' signal falls (indicating that the address information is no longer required by main storage), the 'remember BCU response' latch is turned on, and the channel waits for an 'accept' signal from the BCU interface. Receipt of the 'accept' signal indicates that the storage cycle to store the last data doubleword of the first CCW has started. At this point, the 'remember BCU response' signal and 'accept' signal are AND'ed to turn on the 'CCW fetch' latch, indicating a second storage access is required to fetch the second CCW. The 'A-register full' latch is also turned off at this time. (See "Fetch Second CCW".) Subsequently, the fall of the 'BCU data request' signal reactivates the 'storage request' signal to the BCU interface for the purpose of requesting a CCW fetch storage cycle.

After receipt of the 'accept' signal, the channel receives a 'BCU advance pulse' signal, indicating that storage of the last data doubleword for the first CCW has been completed by main storage. Receipt of the 'BCU advance pulse' signal completes the store last data doubleword, first CCW operations.

Fetch Second CCW. A summary and a simplified description of the fetch second CCW operations follow:

- 'CCW fetch' and 'storage request' signals active.
- Channel receives 'BCU response' signal.
- Command address register contents gated to SAB.
- 'Retain storage' latch turns off.
- 'Storage request' latch turns off.
- Channel receives 'BCU advance pulse' signal.
- SBO bits 8-31 gated to data address register, bits 32-39 to flag register, and bits 48-63 to count register.
- Channel checks for TIC command; if detected, fetches new CCW.
- Channel checks SBO bits 37-39 for all 0's and 48-63 for 'not all 0's' (no error).
- With no error, 'CCW valid' latch turns on.

Channel operations to fetch the second CCW begin with the turn-on of the 'CCW fetch' latch. (Recall that the 'retain storage' signal is still active, preventing turn-off of the 'storage request' latch.) Thus, the 'storage request' signal to the BCU interface is activated when the 'BCU data request' and 'accept latch' signals from the data store operation are no longer active. The active 'CCW fetch' signal conditions the channel to gate the contents of the command address register to the SAB, when required; in addition, the 'CCW fetch' signal conditions channel logic to perform the necessary updating and zero checking operations required for a 'CCW fetch' storage cycle.

Note: For the System/360, Model 91, turn-on of the 'CCW fetch' causes a 'control word request' signal to be supplied to the Peripheral Storage Control Element (indicating that the 'storage request' is for a CCW) and causes the 'pre CDA' signal to drop.

With the 'storage request' signal active and the 'CCW fetch' latch on, the channel waits for a 'BCU response' signal from the BCU interface. Upon receipt of the 'BCU response' signal, the channel gates the address of the second CCW from the command address register to the SAB. The channel's 'retain storage' latch and 'storage request' latch are turned off and the channel awaits receipt of the 'BCU advance pulse' signal, indicating that the second CCW is forthcoming on the SBO lines.

Note: The 'retain storage' latch is turned off by coincidence of the 'CCW fetch' signal, and the 'storage cycle complete' signal; the 'storage cycle complete' signal is generated by delaying the 'BCU advance pulse' signal for the 'store last doubleword, first CCW' operation. Since the arrival of the 'BCU response' signal for the 'fetch second CCW' operation depends upon when the 'storage

request' signal for the CCW (made after the channel receives the 'BCU advance pulse' signal from the data store cycle) is honored, the 'retain storage' latch may be turned off earlier in the sequence than shown in Diagram 5-13. However, this does not affect fetch second CCW operations. The 'storage request' latch remains on until the 'accept' signal is received.

When the 'BCU advance pulse' signal for the second CCW arrives at the channel, it is delayed and causes the CCW to be gated into the appropriate channel registers. SBO bits 8 through 31 (+3P) are gated to the data address register, SBO bits 32 through 39 (+P) to the flag register, and SBO bits 48 through 63 (+2P) to the count register. While the CCW is still on the SBO lines, command bits 4 through 7 are checked for a TIC command. Assuming a TIC command is not present, SBO bits 37 through 39 are checked for an all zeros (no error) condition. Assuming no errors are detected, the 'CCW valid' latch is turned on, indicating that the second CCW is in the channel and is properly formatted. With the 'CCW valid' latch on, the fetch second CCW operation is complete.

Data Handling, First Data Doubleword, Second CCW. A summary and a simplified description of the data handling first data doubleword, second CCW operations follow:

- For read forward operation, first three data bytes are doublegated into B-register unless 'CCW valid' latch is turned on first.
- Doublegating gates first byte to B-register positions 0 and 4, second byte to positions 1 and 5, and third byte to positions 2 and 6. (Mark-B register bits set, accordingly.)
- B-register byte positions 4-7 and mark-B register bits 4-7 reset.
- Fourth through eighth bytes gated to positions 3-7, respectively.

While the last data doubleword for the first CCW is being stored and the second CCW is being fetched, the channel is enabled to begin gating data from the control unit into the B-register (Diagram 5-13).

Note: To fully understand the channel operations performed during the data handling first data doubleword, second CCW operations, a knowledge of the byte counter operation is required. See "Byte Counter Circuits" in Chapter 2.

Data handling, first data doubleword, second CCW operations are enabled by the 'read CDA doublegating', 'not B-register full', and 'suppress out' signals activated during the read CDA setup operations. When the channel receives the first 'service in' signal from the control unit, the data byte on the 'bus in' lines is gated into B-register byte positions 0 and 4, assuming a read forward operation is in progress. (Gating of the same data byte into two B-register byte positions is designated "doublegating"). Doublegating continues (providing the second CCW is not received by the channel as indicated by turn-on of the 'CCW valid' latch) until the three data doublewords have been gated into the B-register (byte 1 to B-register positions 0 and 4, byte 2 to positions 1 and 5, and byte 3 to positions 2 and 6). For each byte doublegated, the two corresponding mark bits in the mark-B register are set.

If the 'CCW valid' latch is not on by the time the byte count latches reach a count of 4, the 'read CDA doublegating' line is deactivated, and the fourth data byte is singlegated into B-register byte position 3. Before the fourth byte is gated into the B-register, B-register byte positions 4 through 7 (bits 32 through 63) and the mark-B register bit positions 4 through 7 are reset (assuming a read forward operation is in progress). If further data bytes are received from the control unit before the 'CCW valid' latch is turned on, they are singlegated into B-register byte positions 4 through 7. (Byte 5 is gated into B-register byte position 4, byte 6 to position 5, byte 7 to position 6 and byte 8 to position 7.) However, recall that once the 'CCW valid' latch is on, the channel is inhibited from gating further data bytes into the B-register until the second CCW setup operations are performed. By inhibiting further data byte gating, the channel can compare the number of bytes gated into the B-register with the DAB and count values received in the second CCW to determine if a chain check condition exists. (See "Second CCW Setup" operations.)

Second CCW Setup. A summary and a simplified description of the second CCW setup operations follow:

- Second CCW setup initiated when 'CCW valid' latch turns on.
- Command address register contents incremented eight bytes.
- 'Sequence 4' latch turned on.
- DAB versus byte count register value examined for chain check condition:
  1. 'Read CDA doublegating' signal active, byte count register value greater than 0, and DAB not on single-word or doubleword boundaries causes chain check condition.
  2. 'Read CDA doublegating' signal inactive and DAB not on doubleword boundaries causes chain check condition.

3. A-register full and DAB not on doubleword boundaries causes chain check condition.

- Both A- and B-registers full and channel receives another data byte causes overrun and chain check conditions.

- Count versus BCR value examined for chain check condition:
  1. BCR value larger than CCW count causes chain check condition.
  2. A-register full, DAB on doubleword boundaries, BCR value greater than one, and count less than 16 causes chain check condition.

- No chain check condition detected; turn off clock with 'sequence 3' and 'sequence 4' signals.

- Turn off 'sequence 3' latch with 'sequence 4' signal and no clock signals.

- Turn on clock with 'sequence 4' and 'not sequence 3' signals.

- DAB and count gated to adder; sum gated back to count register.

- DAB OR'ed into byte counter; effectively added to existing byte count.

- DAB bits reset; data address register P2 bit changed if DAB bits odd number of ones.

- 'Sequence 4' latch turns off.

- 'CDA latch' turns off.

- 'Inhibit service out' signal deactivates.

The second CCW setup operations (Diagram 5-13) begins when the 'CCW valid' latch is turned on ("Fetch Second CCW") indicating that a valid CCW is in the channel. With 'service out' signal response to a 'service in' signal inhibited, the channel updates the command address in the command address register by eight bytes. (The command address register contents plus one are gated to the adder, latched into the adder latches, and the updated results gated back into the command address register.)

Following updating of the command address, the 'sequence 4' latch is turned on to enable further second CCW setup operations to be performed. With the turn-on of the 'sequence 4' latch, the channel samples the number of bytes gated into the B-register versus the starting byte address specified by the DAB for a possible chain check condition. In addition, the channel samples the number of bytes gated into the channel versus the count received in the second CCW for a possible chain check condition. (Detection of a chain check condition results in the channel entering a 'sequence 5' ending routine.)

With regard to the DAB (three low-order bits of the data address register) versus the number of bytes gated into the channel (indicated by the value in the byte counter), comparisons which can cause a chain check condition are described below. If the 'read CDA doublegating' signal is active and the byte count register value does not equal zero [date byte(s) have been doublegated into the B-register], the DAB must specify singleword or doubleword boundaries. If singleword boundaries are specified (DAB equals four for read or three for read backward), the data in the last four byte positions are to be stored; if doubleword boundaries are specified (DAB equals zero for read or seven for read backward) data in all eight byte positions are to be stored. In either case, with the 'read CDA doublegating' signal active and data bytes in the B-register, operations may continue and a chain check condition is not detected. If, in the described case, the DAB specifies byte boundaries (DAB other than zero or four for read or three or seven for read backward), a chain check condition is detected , since data bytes already in the B-register would not be stored. For example, if the first data byte has been doublegated to B-register byte positions zero and four (read operation) while the DAB specifies a beginning byte location of five, the data byte already in the fourth position would not be stored in the proper byte location; thus, a chain check condition results.

If the 'read CDA doublegating' signal is inactive and the byte counter is other than zero, the DAB must specify doubleword boundaries (DAB equals zero for read or seven for read backward); otherwise a chain check condition results. In otherwords, singlegating has begun, the first four data bytes are in the first four B-register byte locations (read operation), and the bytes previously doublegated to byte positions 4 through 6 are reset; therefore, if the DAB value is other than zero, the first (and possibly other) data bytes would not be stored in the proper byte location.

If the 'A-register full' latch is on, more than eight data bytes have been gated into the channel (B-register has been filled and transferred to the A-register). In this case, the DAB must equal zero for read operations or seven for read backward operations (doubleword boundaries), the count from the second CCW must be 16 or greater, and the byte count register must indicate that no more than eight bytes of the second data doubleword have been gated into the channel; otherwise, a chain check condition is detected.

In the event no data bytes have been gated into the B-register, the DAB may specify any byte boundary within the first data doubleword and a chain check condition will not be detected.

With regard to the count (from the second CCW) versus the count in the byte count register, the CCW count must be greater than the number of bytes already gated into the channel; otherwise a chain check condition is detected. In this case the chain check condition is detected to prevent the storage of data bytes not required by the second CCW. For example, if the count specifies four bytes are to be stored and five bytes have already been gated into the B-register, the fifth data byte would be stored, destroying information in storage already stored at the fifth byte location; thus a chain check condition is detected, and the read CDA operation terminated to prevent storage of unwanted data and destruction of data already in storage.

Assuming that a chain check condition is not detected, the second CCW setup operations continue with the turn-off of the channel clock (clock was on for 'sequence 3' operation) and the turn-off of the 'sequence 2' latch. The channel clock is then turned on by the 'sequence 4' and 'not sequence 3' signals to control the remaining second CCW setup operations. With the clock on, data address register bits 21 through 23 (the DAB) plus the count in the count register are gated to the adder. Subsequently, the DAB plus count sum is latched into the adder and gated back into the count register. The count register now contains a modified count which is used to identify the last data byte to be transferred for the last data doubleword of the second CCW.

When the DAB plus count value is gated to the adder, the DAB is also OR'ed into the byte counter, where it is effectively added to the value existing in the byte counter at the time. With the DAB added to the existing byte count, the byte counter parity bit is corrected (if necessary) to reflect correct parity for the new count value. Since the DAB versus byte count comparison did not previously result in a chain check condition, the new byte count value effectively specifies the B-register byte location into which the next data byte from the control unit will be gated. In otherwords, the new byte count value will cause the next data byte to be gated into the B-register byte position adjacent to the B-register byte location which received the last data byte from the control unit.

After the DAB plus count and DAB to byte counter operations are performed, the DAB bits in the data address register are reset to all zeros, since the second and subsequent data doublewords will be read from the control unit and stored on doubleword boundaries. Prior to resetting the DAB bits, the DAB bits are examined to determine if the reset operation will affect the data address register P2 parity bit. If necessary, the P2 bit is corrected at the P2 latch just prior to the DAB reset.

Following the DAB reset, the 'sequence 4' latch is turned off, causing the 'CDA' latch to turn off. Turn-off of the 'CDA' latch deactivates the 'inhibit service out' signal to complete the second CCW setup operations and (assuming that the first data doubleword is not in the A-register) allow data handling, first data doubleword, second CCW operations to continue (Diagram 5-13).

Data Handling, First Data Doubleword, Second CCW, Continued. With the second CCW setup operations complete, the channel continues to gate data bytes into the B-register on a singlegating basis until the B-register is full (BCL = 0 detected). When the B-register is full, the 'sequence 3' latch is turned on, and the transfer first data doubleword, second CCW, the store first data doubleword, second CCW, the update count and the update data address operations are performed in the manner previously described for the first data doubleword of the first CCW.

Data handling operations for the second data doubleword of the second CCW begin with the turn-off of the 'B-register full' latch during the transfer first data doubleword, second CCW to A-register operation. From this point, channel operations continue as for a normal read operation. For the Diagram 5-13 example, the chain data flag bit in the second CCWis not active. Thus, when the last data doubleword is being stored, the channel enters a sequence 5 ending routine to obtain a status byte from the control unit and clear the I/O interface. Subsequent to the sequence 5 routine, the channel enters an interrupt routine to store the channel status word and gate the unit address register contents and condition code 0 to the CPU interface.

If the chain data flag bit in the second CCW is active, the read CDA operations described in the preceding paragraphs are repeated to fetch a third CCW into the channel.

Detailed Read CDA Operations

Detailed descriptions of the read CDA operations are based upon Diagram 5-14. For simplicity, the detailed read CDA operation is described by subroutine.

Detailed Initial Conditions and Initial Status Cleanup. It is assumed that a successful initial selection routine has been completed, and that the channel has sent a 'release' signal to the CPU interface. Other channel conditions at the start of the read CDA operation are as follows: (1) 'sequence 1' and 'sequence 2' latches on; (2) 'sequence 3', 'sequence 4', and 'sequence 5' latches off; (3) 'status in' signal active; (4) read command is specified by active command register bits 5 or 6 and not bit 7; (5) 'gate command out' (read or write) latch is on; (6) the chain data flag bit in the flag register is active, and the channel clock is on.

Before the actual read CDA operations can begin the channel must clear the status-in condition presented by

the control unit at the end of the initial selection routine (Diagram 5-2). With the 'sequence 1' and 'sequence 2' latches on, the 'status in' signal active, and the status byte in the 'bus in' latches equal to zero, the channel's 'service out' signal is activated by the 'T2' clock signal. The active 'status in' signal gates the 'service out' signal to the control unit. The channel then waits for the control unit to drop its 'status in' signal. When the 'status in' signal drops, the channel turns off the 'sequence 1' latch, deactivates the 'service out' signal, and turns off the channel clock. At this point, the channel's 'sequence 2' latch is on, the 'read' signal is active, the chain data flag bit is active, and the channel is ready to begin handling data for the first data doubleword of the first CCW.

Detailed Data Handling, First Data Doubleword, First CCW. A summary and a detailed description of the data handling, first data doubleword, first CCW operations follow:

- Channel receives 'service in' signal.
- Channel activates 'service out' signal.
- 'Service out' signal activates 'latch bus-in' signal; data byte latched into 'bus in' latches.
- Data byte parity checked at 'bus in' latches by 'sample bus-in' signal.
- 130 ns signal updates byte counter and gates data byte into B-register position specified by byte counter. (Byte counter value changes when 130 ns signal falls.)
- Mark-B register bit activated when byte gated into B-register.
- 'BCL = 0' latch and 'BC = CTB' trigger strobed by signal gating byte into B-register.
- BCL = 0 condition indicates B-register is full.
- BC = CTB condition indicates last byte of CCW gated to B-register.
- If neither BCL = 0 nor BC = CTB condition detected, exchange of 'service in' and 'service out' signals continues.
- Detection of BCL = 0 condition turns on 'BCL = 0' latch.
- 'B-register full' latch turns on.
- 'Service out' signals inhibited.
- 'Sequence 3' latch and clock turn on.

Data handling for the first data doubleword of the first CCW begins when the control unit presents the channel with a 'service in' signal indicating that a data byte is present on the 'bus in' lines (Diagram 5-14). Providing the channel is not performing a sequence 5 operation (due to a device initiated ending, or an ending sequence initiated by a check condition), the channel sends a 'service out' signal to the control unit indicating acceptance of the data byte. With the activation of the 'service out' signal, the 'latch bus-in' signal is activated to latch the data byte into the 'bus in' latches and the 'sample bus-in' signal is activated to check the signal for the 'bus in' data parity. The 'service out' signal also triggers a 130-ns singleshot which produces a signal to update the byte counter and, simultaneously, gate the data byte in the 'bus in' register to the B-register byte position specified by the byte counter. Since the byte counter output is not changed until the fall of the 130-ns signal, the data byte from the 'bus in' latches is gated into the B-register byte position specified by the byte counter prior to the byte count change. (See "Byte Counter Circuits" in Chapter 2.) At the same time the data byte is gated into the B-register the mark-B register bit corresponding to the gated B-register byte position is set to a logic 1 level. A logic 1 level mark bit indicates that data in the corresponding byte position is to be stored in main storage; if the bit is a logic 0 (not set), the corresponding data byte will not be stored. The same signal used to gate the 'bus in' data byte into the B-register also activates a signal to gate the BCL = 0 latch and 'BC = CTB' trigger. If a BCL = 0 condition exists, it indicates that the B-register is full (a data byte has been loaded into the last byte position of the B-register). If a BC = CTB condition exists with the 'last word' trigger on, it indicates that the last data byte specified by the count from the first CCW for the current read CDA operation has been gated into the B-register.

For the present, assume that neither the BCL = 0 nor BC = CTB conditions exist. In this case, the channel awaits the drop of the 'service in' signal from the control unit. When the signal drops, the channel drops its 'service out' signal and waits for the next 'service in' signal from the control unit.

The exchange of 'service in' and 'service out' signals continues until either the BCL = 0 (with the 'last word' trigger off) condition is detected, or the BC = CTB (with the 'last word' trigger on) condition is detected. Assume that the BCL = 0 condition is detected during the gating of a data byte into the B-register. This means, as previously described, that the data byte is being gated into the last byte position of the B-register; i.e., the B-register byte boundary has been reached. Detection of the BCL = 0 condition turns on the 'BCL = 0' latch. In turn, the 'BCL = 0' signal turns on the 'B-register full' latch, after

which the channel is inhibited from responding to further 'service in' signals from the control unit. In the absence of the 'A-register full' signal, the 'BCL = 0' signal generates a 'read and BC equal 0' signal which turns on the 'sequence 3' latch and the channel clock. Turn-on of the 'sequence 3' latch and the channel clock signifies the end of the data handling, first data doubleword, first CCW operation.

Detailed Transfer First Data Doubleword of First CCW to A-Register.

- 'B to A' and 'mark-B to mark-A' latches turn on.
- B-register data transferred to A-register; mark-B register bits transferred to mark-A register.
- 'A-register full' latch turned on by 'read, sequence 3, and T2' signal.
- 'BCL = 0' and 'B-register full' latches are reset.
- B-register is reset.
- 'Not B-register full' signal enables channel to activate 'service out' signal in response to 'service in' signal.
- 'BCU response' signal gates:
  1. Data address register bits 0-20 (+3P) to SAB.
  2. Mark-A register bits 0-7 (+P) to mark lines.
  3. Storage protect register bits 0-3 (+P) to 'storage protection' lines.
  4. 'Store' signal to BCU interface.
- Channel receives 'BCU data request' signal which:
  1. Gates A-register contents to SBI lines.
  2. Causes SBI logic to generate parity bits for A-register bytes.
- SBI generated parity bits:
  1. Gated to SBI lines.
  2. Compared with A-register bits for SBI parity check condition.
- Detected SBI parity check condition turns on 'channel data check' latch; read CDA operation not terminated.
- Channel receives 'accept' signal.
- 'A-register full' latch turns off.

Channel operations to transfer the first data doubleword of the first CCW from the B-register to the A-register begin while the 'sequence 3' latch is being turned on (Diagram 5-14). The same 'read and BC equal 0' signal that turns on the 'sequence 3' latch also causes the 'B to A' latch and the 'mark-B to mark-A' latch to turn on. With the 'B to A' latch on, the data in the B-register is gated into the A-register. Turn-on of the 'mark-B to mark-A' latch gates the mark bits contained in the mark-B register to the mark-A register.

Note: If the 'A-register full' latch is on due to a previous read operation storage request, the A-register full condition indicates that main storage is not storing data at the channel data-transfer rate. In some cases, this may result in a 'status in' signal from the control unit when the channel is finally able to respond to the control unit's 'service in' signal with a 'service out' signal ('service out' signal response cannot occur until the B- to A-register transfer is completed). For example, assume the channel is servicing a high-speed drum which must have a response to its 'service in' signal within a specified period. If the channel is unable to respond within the specified time, the high-speed-drum control unit will, when the channel finally does send a 'service out' signal, present a 'status in' signal indicating that an overrun condition has occurred at the I/O device. However, since this paragraph describes transfer of the first data doubleword of the first CCW, the 'A-register full' latch should be off.

To complete the transfer first data doubleword of first CCW to A register operation, the 'A-register full' latch is turned on and the 'B-register full' latch is turned off by the 'T2' clock signal.

The same signal ('read, sequence 3, and T2') that turns on the 'A-register full' latch, also turns off the 'BCL = 0' and 'B-register full' latches and resets the B-register. Assuming that a second data doubleword is to be transferred for the first CCW, turn-off of the 'B-register full' latch permits the channel to gate data bytes into the B-register for the second data doubleword. At this point, the transfer first data doubleword of first CCW to A-register operations are complete, and further channel operations (store first data doubleword of first CCW and update count, first data doubleword, first CCW) begin with the turn-on of the 'storage request' latch by the 'T0' clock signal and subsequent turn-on of the 'storage cycle' trigger. With the 'storage request' latch on, a 'storage request' signal is sent to the BCU interface for the purpose of requesting a storage cycle to store data contained in the A-register. With the 'storage cycle' trigger on, appropriate channel controls are activated or inhibited in preparation for a data transfer storage cycle.

With the 'storage request' signal active, the channel awaits receipt of the 'BCU response' signal (Diagram 5-14). Upon receipt of the 'BCU response' signal, the channel gates the following to the BCU interface: (1) data address register bits 0 through 20 (+3P) to the SAB; (2) mark-A register bits 0 through 7 (+P) to the mark lines; (3) and

storage protect register bits 0 through 3 (+P) to the 'storage protection' lines. In addition, the 'BCU response' signal AND's with the 'gate store to BCU' signal (ALD MA111) to activate the 'store' signal to the BCU interface. When received in main storage, the data address on the SAB indicates the doubleword location at which the first data doubleword in the A-register is to be stored. Active mark bits specify that corresponding A-register bytes are to be stored at byte locations within the doubleword location specified by the data address. In storage units having storage protection facilities, the storage protect key permits access to the storage location specified by the data address. The 'store' signal indicates that main storage is to store data received via the SBI lines. With the address information and 'store' signal on the appropriate BCU interface lines, the channel awaits the receipt of a 'BCU data request' signal from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'BCU data request' signal is generated within the channel by delaying the 'BCU response' signal approximately 150 ns.

Upon receipt of the 'BCU data request' signal, the data bits in the A-register are gated to the SBI lines and to a parity generator circuit where parity is generated for each of the eight A-register bytes. The generated parity bits are then gated to the SBI lines and are also compared with the eight parity bits contained in the A-register. Approximately 200 ns after the 'BCU data request' signal rises, the result of the parity comparison is strobed to determine if an SBI parity error exists. If a parity error does exist (generated parity bits do not match A-register parity bits), an 'SBI parity check' signal is generated and the 'channel data check' latch is turned on. For a read CDA operation, the channel data check condition does not terminate the operation. However, the channel data check condition is indicated in the channel status word stored at the end of the read CDA operation.

The fall of the 'BCU data request' signal indicates that the A-register data is no longer required on the SBI lines.

The 'accept' signal (which occurs at the end of the 'BCU data request' signal) causes a 350 ns singleshot to fire, which in turn generates a signal to turn off the 'A-register full' latch. Turn-off of the 'A-register full' latch and receipt of the 'BCU advance pulse' signal signifies the end of the store first data doubleword of first CCW operation.

Detailed Update Count, First Data Doubleword, First CCW. A summary and a detailed description of the update count, first data doubleword, first CCW operations follow:

- Count register contents minus eight bytes gated to adder by 'T0' signal.
- 'T3' signal samples count register value for last word conditions (count less than 17).
- 'T4' signal latches adder and gates decremented count value into count register.
- 'T5' signal turns on 'sequence 4' latch.

The count in the count register is updated (decremented eight bytes) for all read CDA storage cycles except for the storage cycle to store the last data doubleword for the first CCW. The last cycle is indicated when the 'last word' trigger is on (Diagram 5-13). Assuming the 'last word' trigger is off, the channel enters an update count operation when, after the clock is turned on, the 'T0' clock signal is activated. With the activation of the 'T0' signal, the count register contents are gated to the adder where eight bytes are subtracted from the count (count minus one value). At 'T3' clock time, channel logic is sampled for a count register value of less than 17 bytes. If the count is less than 17, the 'last word' trigger is turned on to indicate that the next data doubleword transfer to storage is the last data transfer for the first CCW of the read CDA operation. (For simplication, assume that the 'last word' trigger is not turned on during the update count, first data doubleword, first CCW operation.) Note that the count is sampled by the 'T3' clock signal, which is prior to the time ('T4' clock time) that the decremented count is latched into the adder and gated into the count register. Thus, the count is monitored for a value of less than 17 bytes because the storage operation presently in progress is reflected in the count. In otherwords, the last word condition, when present, is detected just before the counter is decremented to reflect a count of eight or less bytes.

After the decremented count is gated back into the count register, the 'T5' clock signal turns on the 'sequence 4' latch which turns off the clock to end the update count, first data doubleword, first CCW operation and begin the update data address, first data doubleword, first CCW operation.

Detailed Update Data Address, First Data Doubleword, First CCW. Entry into the update data address, first data doubleword, first CCW operation for a read CDA operation occurs after completion of the update count, first data doubleword, first CCW operation, providing the 'read CDA' latch is not on (Diagram 5-14). (Turn-on of the 'read CDA' latch indicates that the last data doubleword is in the B-register; in this case, the data address register contents are not updated.)

With the 'sequence 4' latch on, the channel clock off, and the 'A-register full' latch off, the 'sequence 3' latch is turned off and the channel clock turned back on to time the update data address, first data doubleword first CCW operation.

The 'sequence 4' and 'not T6' clock signals are AND'ed to gate the data address register contents to the adder where eight bytes (data address plus one) are added to the data address. At 'T4' clock time, the adder is latched and the updated address is gated back into the data address register. At 'T6' clock time, the 'sequence 4' latch is turned off indicating that the update data address, first data doubleword, first CCW operation is complete. The channel operations described for the first data doubleword of the first CCW (data handling, transfer data doubleword to A-register, update count, and update data address) are identical for all data doublewords for the first CCW, except for the next to the last and the last data doublewords.

Detailed Next to Last Data Doubleword, First CCW, Operations. Channel read CDA operations performed during the next to last data doubleword for the first CCW are identical to those described for the first data doubleword except that the 'last word' trigger is turned on during these operations (Diagram 5-14). Assume that an update data address operation has just been completed, that the 'sequence 4' latch and 'last word' trigger are off, and the count register contains a count of less than 17 bytes. The data handling operations for the next to the last data doubleword of the first CCW began when the 'B-register full' latch was turned off during the previous transfer data doubleword of first CCW to A-register operation. When all eight bytes for the next to the last data doubleword for the first CCW have been gated into the B-register, the BCL = 0 condition is detected causing the 'BCL = 0' latch to turn on. With the 'BCL = 0' latch on, the 'B-register full' latch is turned on. Assuming the 'A-register full' latch is off, the 'sequence 3' latch and channel clock are turned on. The transfer data doubleword, first CCW, to A-register operation and the store data doubleword, first CCW operation are performed as previously described to transfer and store the next to the last data doubleword.

With the 'last word' trigger off, the count register contents minus one (the equivalent of eight bytes) are gated to the adder at 'T0' clock time. At 'T3' clock time, prior to gating the decremented count back into the count register, the count in the count register is sampled for a value of less than 17. Since the next to the last data doubleword is now being stored, the count value is less than 17, and the 'last word' trigger is turned on. Following turn-on of the 'last word' trigger, the decremented count is latched into the adder and gated into the count register at 'T4' clock time.

Following completion of the update count operation, the 'sequence 4' latch is turned on by the coincidence of the 'T5' and 'sequence 3' signals. The channel then performs an update data address operation in the manner previously described for the first data doubleword of the first CCW. At this point, the channel enters operations to gate data bytes into the B-register for the last data doubleword of the first CCW.

Detailed Data Handling, Last Data Doubleword, First CCW. With the 'last word' trigger on and data handling operations for the last data doubleword in progress, the channel samples for a BC = CTB condition each time a data byte is gated into the B-register (Diagram 5-14). Detection of the BC = CTB condition indicates that the last data byte for the last data doubleword of the first CCW is being gated into the B-register. Since the CDA flag bit is active, the 'delay service in' latch is turned on, causing the 'suppress out' signal to the control unit to be raised. The 'suppress out' signal is effective when the control unit being serviced is a buffered I/O device or a start-stop I/O device. With the 'suppress out' signal active, these type devices are prevented from raising a 'service in' signal until the 'suppress out' signal drops.

Detection of the BC = CTB condition turns on the 'BC = CTB' trigger, as during a normal read operation. With the 'BC = CTB' trigger on, the 'BC = CTB' latch is conditioned to turn on, but is held reset by the 'CDA flag' signal. The BC = CTB condition with the 'last word' trigger on also activates the reset signal to the count register, resetting the count to all zeros with good parity. At the same time the count register is reset, the 'B-register full' latch is turned on indicating that the last data doubleword is in the B-register. With the 'A-register full' latch off and the presence of the 'read', 'not CDA latch', 'BC = CTB', 'not T1', 'not sequence 4', and 'not command reject or control check' signals, the 'sequence 3' latch and channel clock are turned on. This completes the data handling operations for the last data doubleword of the first CCW, and the channel begins the read CDA setup operations.

Detailed Read CDA Setup. A summary and a detailed description of the read CDA setup operations follow:

- 'CDA' latch turns on at 'T0' time.
- 'Storage request' latch is turned on.
- 'Storage cycle' trigger turns on.
- With skip flag bit inactive, 'retain storage' latch turns on; if skip flag bit active, 'retain storage' latch remains off.
- When skip flag bit active, 'CCW fetch' and 'storage request' latch turned on to fetch next sequential CCW.

- 'CCW valid' latch turns off.
- 'Last word' trigger turns off.
- B-register contents gated to A-register and mark-B register bits to mark-A register.
- 'B-register full' latch and 'BC = CTB' trigger turns off; B-register is reset.

Read CDA setup operations (Diagram 5-14) are performed to prepare the channel to (1) store the last data doubleword of the first CCW and fetch the second CCW in that order, and (2) prepare the channel for special operations to gate the first data doubleword of the second CCW into the B-register.

With the 'last word' trigger on and the CDA flag bit in the flag register active, the 'CDA' latch is turned on at 'T0' clock time. In turn, the 'CDA' signal turns on the 'storage request' latch and also turns on the 'retain storage' latch, assuming that the skip flag bit in the flag register is inactive. The active 'retain storage' signal insures that the 'storage request' latch remains on until the last data doubleword for the first CCW is stored and the 'storage request' signal for the second CCW has been honored. However, the 'storage request' signal to the BCU interface is disabled upon receipt of the 'BCU data request' signal for the last data doubleword and activated when the 'BCU data request' signal falls and the 'accept' latch is turned off.

Note: If the skip flag bit is on, no data has been stored for the first CCW, even though the channel has received the data bytes specified by the CCW count. At this point in the read CDA setup operations, an active skip flag bit prevents turn-on of the 'retain storage' latch and causes turn-on of the 'CCW fetch' latch and 'storage request' latch to fetch the next sequential CCW from main storage.

With the 'CDA' latch on and the skip flag bit inactive, the byte counter and 'CCW valid' latch are reset by the 'read CDA' signal. Turn-off of the 'CCW valid' latch with the 'CDA' latch on, turns off the 'last word' trigger and inhibits turn-on of the trigger until the second CCW is in the channel. The 'last word' trigger is held in the off position to prevent turn-on of the trigger when the 'sequence 3' latch is turned off.

With the byte counter reset, the 'read and sequence 3' signal active' the 'CDA' latch on, and the 'not BCR bit 4' signal active, the 'read CDA doublegate signal is activated. With this signal active, the channel is conditioned to doublegate data bytes from the control unit into the B-register. (See "Detailed Data Handling, First Data Doubleword, Second CCW".)

During the read CDA setup operations, the last data doubleword of the first CCW is gated from the B-register into the A-register, the mark-B bits are gated into the mark-A register, and the 'B-register full' latch is turned off (11-B2). These operations are accomplished by turning on the 'B to A' latch to transfer the B-register data to the A-register, turning on 'mark-B to mark-A' latch to transfer the mark-B register bits into the mark-A register, and generating a 'read, sequence 3, T2' signal to turn off the 'B-register full' latch and 'BC = CTB' trigger. The same signal which turns off the 'B-register full' latch also turns on the 'A-register full' latch. The B-register is reset at 'T2' clock time in preparation for doublegating data bytes for the first data doubleword of the second CCW into the B-register. Since the operations in progress include storage of the last data doubleword of the first CCW (skip flag bit inactive), a sequence 4 routine to update the data address register is not required and is not performed at this time. In fact, the channel waits with the 'sequence 3' latch and clock on for the 'CCW valid' latch to turn on before entering a sequence 4 routine.

During the read CDA setup operations, (skip flag bit inactive), the 'storage request' latch is turned on at 'T0' clock time. With the 'retain storage' latch on, the 'storage request' latch remains on until the storage request for the second CCW is honored. With the turn-on of the 'storage request' latch, the 'storage cycle' trigger is turned on to complete the read CDA setup operations. The channel now may begin gating data into the B-register while (in the retain storage mode) is performs operations to store the last data doubleword of the first CCW and fetch the second CCW.

Detailed Store Last Data Doubleword and Fetch Second CCW. A summary and a detailed description of the store last data doubleword and fetch second CCW operations follow:

- Channel receives 'BCU response' signal:
  1. Data address register bits 0-20 (+3P) gated to SAB.
  2. Storage protect register bits 0-3 (+P) gated to 'storage protection' lines.
  3. Mark-A register bits gated to mark lines.
  4. 'Store' signal to BCU interface activated.
- Channel turns on 'address valid' latch; 'address valid' signal to BCU interface rises.
- Channel receives 'BCU data request' signal:
  1. A-register bits gated to SBI lines.
  2. SBI parity generator generates B-register parity bits, one for each A-register byte.
  3. 'Storage request' signal to BCU interface degated.

4. SBI generated parity bits and A-register parity bits compared; mismatch turns on 'channel data check' latch.

- 'BCU response' signal falls; 'remember BCU response' latch turns on.
- Channel receives 'accept' signal; 'accept' latch turns on.
- 'CCW fetch' latch turns on.
- 'BCU data request' signal falls; A-register data degated from SBI lines.
- Channel receives 'BCU advance pulse' signal from data store cycle.
- 'Storage cycle complete' signal turns off 'retain storage' latch.
- 'BCU advance pulse' signal turns off 'accept' latch; 'storage request' signal to BCU interface activated.
- Channel receives 'BCU response' signal.
- Command address register contents gated to SAB.
- Channel receives 'accept' signal from CCW fetch cycle.
- 'Storage request' latch turns off.
- Channel receives 'BCU advance pulse' signal.
- Channel gates CCW into appropriate channel registers.
- CCW checked for TIC command; if detected channel fetches new CCW.
- If not TIC command, channel examines SBO bits 37-39 for 0's and 48-63 for not all 0's.
- If zero check successful, 'CCW valid' latch turns on.
- 'CCW fetch' latch turns off.
- Command address register contents incremented eight bytes.

With the 'retain storage' latch on, the channel enters a retain storage routine to store the last data doubleword of the first CCW (skip flag bit inactive), and to fetch the second CCW (Diagram 5-14). Recall that the 'storage request' latch has been turned on, causing the 'storage request' signal to be raised at the BCU interface. With the 'read' signal active, the channel awaits a 'BCU response' signal indicating that the 'storage request' signal has been honored. Upon receipt of the 'BCU response' signal, the channel (1) gates the data address of the last data doubleword (data address register bits 0 through 20 plus 3P) to the SAB; (2) gates the storage protect key (storage protect register bits 0 through 3 plus P) to the 'storage protection' lines; (3) gates the mark bits (mark-A register bits 0 through 7 plus P) to the mark lines; and (4) activates the 'store' signal to the BCU interface indicating that the storage cycle was requested to store data. After a deskewing delay to allow the address information bits to stabilize on the BCU interface lines, the delayed 'BCU response' signal turns on the channel's 'address valid' latch raising the 'address valid' signal to the BCU interface. The 'address valid' signal indicates that the address information (data address, storage protect key and mark bits) and 'store' signal are stable on the appropriate BCU interface lines.

At this point, the channel waits for a 'BCU data request' signal from the BCU interface, indicating that the data to be stored is required on the SBI lines.

Note: For 2860 selector channels used with the System/360, Model 91, the 'BCU data request' signal is generated in the channel by delaying the 'BCU response' signal by approximately 150 ns.

Upon receipt of the 'BCU data request' signal, the 'storage request' signal to the BCU interface is degated and the data bits in the A-register are gated to the SBI lines and to the SBI parity generator. At the SBI parity generator a parity bit is generated for each of the eight A-register bytes. The generated parity bits are then gated to the SBI lines and also compared at SBI logic with the eight parity bits originally contained in the A-register. Approximately 200 ns after the 'BCU data request' signal rises, the result of the parity comparison is strobed to determine if an SBI parity error exists. If a parity error does exist (generated parity bits do not match A-register parity bits), the 'SBI parity check' signal is activated to turn on the 'channel data check' latch. For the read CDA operation, the channel data check condition does not terminate read CDA operations. However, the channel data check bit is active in the channel status word stored at the end of the read CDA operation.

Upon the fall of the 'BCU response' signal, the 'remember BCU response' latch is turned on, and the address information, 'store' signal, and 'address valid' signal to the BCU interface are degated. In the meantime, the channel has either received or is awaiting an 'accept' signal from the BCU interface. Receipt of the 'accept' signal, indicating that the storage cycle to store the last data doubleword for the first CCW has started in main storage, turns on the channel 'accept' latch, and turns off the 'A-register full' latch.

Subsequently, the 'CCW fetch' latch is turned on by the coincidence of the 'remember BCU response' and the 'accept' signals.

Note: Model 2860 Selector Channels used with the System/360, Model 91, supply a 'control word request' signal to the BCU interface with the turn-on of the 'CCW fetch' latch, indicating that the 'storage request' signal is for a control word, rather than a data doubleword. At the same time, the 'pre CDA' signal to the Model 91 is dropped.

With the 'storage request' latch still on (due to the active 'retain storage' signal) receipt of the 'BCU advance pulse' signal for the data store cycle will turn off the 'accept' latch; this activates the 'storage request' signal to the BCU interface. The channel may await receipt of the 'BCU response' signal indicating that the 'storage request' signal for the second CCW has been honored. When the channel receives the 'BCU response' signal, it is AND'ed with the 'CCW fetch' signal to gate the contents of the command address register to the SAB. Recall that the command address register contains the address of the next CCW.

In the meantime, receipt of the 'BCU advance pulse' signal for the last data doubleword of the first CCW indicates that the last data doubleword has been stored. (If the system is unable to honor the 'storage request' signal for the CCW fetch cycle immediately, the 'BCU advance pulse' signal for the last data doubleword may arrive prior to the 'BCU response' signal for the CCW. This will not affect channel operations.)

Upon receipt of the 'BCU advance pulse' signal for the last data doubleword of the first CCW, the channel delays the 'BCU advance pulse' signal. The delayed signal activates the 'storage cycle complete' signal which is AND'ed with the 'CCW fetch' signal to turn off the 'retain storage' latch. When the 'accept' signal for the second CCW storage fetch cycle has been received by the channel, the off condition of the 'retain storage' latch allows turn-off of the 'storage request' latch. Receipt of the 'accept' signal indicates that the storage cycle for the second CCW has begun; in the channel, the 'accept' signal sets the 'accept' latch to turn off the 'storage request' latch and condition channel control logic for operations involving receipt of the second CCW. When the 'BCU response' signal falls, the 'remember BCU response' latch is turned on.

Prior to receipt of the 'accept' signal, the channel receives a 'BCU data request' signal (or in the case of the System/360, Model 91, the channel generates a 'BCU data request' signal). Although data is not gated to the SBI lines during the CCW fetch operation, the 'BCU data request' signal is required for channel operations. For example, the 'BCU data request' signal degates the 'storage request' signal to the BCU interface, and the presence of the 'BCU data request' signal is required for turn-on of the channel 'accept' latch when the 'accept' signal is received.

After receipt of the 'accept' signal, the channel waits for the 'BCU advance pulse' signal. This signal indicates that the second CCW is forthcoming on the SBO lines. The 'BCU advance pulse' signal is delayed, as previously described, until the CCW is on the SBO lines. When the CCW is on the SBO lines, the command bits (SBO bits 0 through 7) are sampled for a TIC command. If a TIC command is indicated, the channel enters a TIC routine during which (1) the CCW is gated into the channel registers, (2) appropriate zero checks are made on CCW bits present on the SBO lines, and (3) a storage cycle is initiated to fetch a new CCW from the storage location specified by the data address bits obtained from the second CCW.

Assuming that a TIC command is not detected, SBO bits 37 through 39 are examined for all 0's (no error) and SBO bits 48 through 63 (count bits) are examined for a not all 0's condition (no error). If an error condition is detected, the channel's 'program check' latch is turned on and the read CDA operation is terminated.

If no errors are detected, the delayed 'BCU advance pulse' signal causes SBO bits 8 through 31 (+3P) to be gated into the data address register, bits 32 through 39 (+P) to be gated into flag register, and bits 48 through 63 (+2P) to be gated into the count register.

After the CCW is gated to the appropriate channel registers, the 'CCW valid' latch is turned on by AND'ing the 'late advance pulse' and the 'CCW fetch' signals. Following turn-on of the 'CCW valid' latch, the 'CCW fetch' latch is turned off by AND'ing the 'not late advance pulse' and 'CCW valid' signals. Since the 'CDA' latch is still on, the turn-on of the 'CCW valid' latch stops the channel from gating further data bytes into the B-register by inhibiting generation of the 'service out' signal. Inhibiting the gating of further data bytes into the B-register permits the channel to sample for a chain check condition during the second CCW setup operations with the B-register and byte counter in a stable condition. In addition, the new DAB is gated into the byte counter, requiring that the byte counter be in a stable condition at the time the DAB is gated.

When the 'CCW fetch' latch is turned on, and the command address register contents gated to the SAB, the command address register contents plus eight bytes are gated to the adder. With the turn-on of the 'CCW valid' latch, the 'latch adder command address' latch is turned on to latch the updated command address into the adder latches. With the 'latch adder command address' latch on, the 'late advance pulse' signal (from the CCW fetch cycle) activates a signal to gate the contents of the adder latches into the command address register. Subsequently, the 'storage cycle complete' signal (activated by the 'late

advance pulse' signal) turns off the 'latch adder command address' latch to complete the update command address register operation. The command address register now contains the address location of the next sequential CCW in the event that a third CCW must be fetched during the read CDA operation. With the command address updated, the 'store last data doubleword of the first CCW and fetch second CCW' operations are complete. At this point, the channel enters the second CCW setup operations to (1) determine if the data bytes gated into the B-register during the data handling for first data doubleword, second CCW operations are compatible with the count and DAB values received from the second CCW, and (2) perform the necessary DAB plus count and DAB to byte counter operations. (See "Detailed Second CCW Setup".)

Detailed Data Handling, First Data Doubleword, Second CCW. A summary and a detailed description of the data handling, first data doubleword, second CCW operations follow:

- 'Read CDA doublegating' signal active.

- If channel receives 'service in' signal(s) before 'CCW valid' latch turns on (read forward):
  1. First data byte gated to B-register positions 0 and 4, second to 1 and 5 and third to 2 and 6, corresponding mark bits set in mark-B register.
  2. Fourth 'service in' signal causes B-register byte positions 4-7 and mark-B register bits 4-7 to be reset.
  3. Fourth through eighth data bytes singlegated to B-register positions 3-7, respectively.
  4. If B-register filled, contents transferred to A-register and singlegating into B-register continues.

- 'CCW valid' latch turns on; activation of 'service out' signals inhibited.
  1. If still active, 'read CDA doublegating' signal drops.
  2. If bytes have been gated into B-register, DAB and count values from new CCW compared with byte counter value to determine if chain check condition exists.

The channel is enabled to start gating data for the first data doubleword of the second CCW (Diagram 8-18, FEDM) when: (1) the 'read CDA doublegating' signal is activated; (2) the 'B-register full' latch is turned off; and (3) in the case of buffered or start-stop I/O devices, when the 'suppress out' signal is dropped ('T6' clock time). Assuming that the control unit sends the channel a 'service in' signal, the first data byte from the control unit is gated into B-register byte positions 0 and 4. The gating of the same data byte into two B-register positions is designated doublegating. If a second and third 'service in' signal are received before the second CCW is received in the channel, the second data byte is gated into B-register byte positions 1 and 5, and the third data byte is gated into byte positions 2 and 6. After the third byte is doublegated, the byte count register is advanced to a count of 4. If the CCW is not yet in the channel ('not CCW valid'), the 'read CDA doublegating' signal is dropped. If a fourth data byte is received from the control unit before the 'CCW valid' latch is turned on, the fourth data byte is gated into the B-register byte 3 position. The 'gate bus in' signal which gates the fourth data byte is AND'ed with the 'read and sequence 3', 'CDA latch', 'BC latch equals 4', and 'not CCW valid' signals to produce reset signals to the last four byte positions of the B-register and the last four bit positions of the mark-B register. In other words, B-register bits 32 through 63 and mark register bits 4 through 7 are reset to prepare these positions to receive the fifth through eighth data bytes from the control unit.

Note: If a read backward CDA operation is in progress, the B-register is loaded in the reverse order to that described for the read CDA operation. That is, the first data byte is doublegated to B-register byte positions 7 and 3, the second byte to 6 and 2, etc. In this case, while the fourth data byte is being gated into B-register byte position 4, bits 0 through 31 of the B-register and bits 0 through 3 of the mark register are reset, so that the fifth through eighth data bytes from the control unit can be gated to B-register byte positions three through zero, respectively.

As long as the 'CCW valid' latch is off, and 'service in' signal are presented to the channel by the control unit, the data bytes will be singlegated into the B-register. For the read CDA operation, the fourth through eighth data bytes from the control unit are gated into B-register byte positions 4 through 7, respectively. Once doublegating has ceased, the DAB in the incoming CCW must equal zero (doubleword boundary) or an overrun condition occurs to turn on the 'chain check' latch and cause termination of the read CDA operation. If the B-register is filled and transferred to the A-register before the 'CCW valid' latch is on, the DAB must be equal to 0 and the count from the CCW must be equal to or greater than 16 bytes, otherwise, an overrun condition is detected and the channel 'chain check' latch is turned on to terminate the read CDA operations. For a detailed description of chain check conditions, refer to "Detailed Second CCW Setup".

Detailed Second CCW Setup.

- 'Sequence 4' latch turns on.

- Chain check detected if:
  1. 'Read CDA doublegating' signal is inactive and DAB is not on doubleword boundaries.

2. 'Read CDA doublegating' signal is active, BCL does not equal zero, and DAB is not on singleword or doubleword boundaries.
3. Both the A-register and B-register are filled with data.
4. CCW count is less than byte count.
5. A-register is full, DAB is on doubleword boundaries and count is less than 16.

- If chain check condition not detected, channel clock turns off.
- 'Sequence 3' latch turns off.
- Channel clock turns on.
- DAB plus count register values gated to adder.
- Sum gated back into count register.
- DAB OR'ed into byte counter.
- Data address register P2 bit changed if DAB bits equal odd number of 1's.
- DAB bits in data address register reset.
- 'Sequence 4' latch turns off.
- 'CDA' latch turns off.
- 'Inhibit service out' signal drops.
- Channel free to gate data bytes into B-register.

The channel performs the second CCW setup operations (Diagram 5-14), after turn-on of the 'CCW valid' latch and update of the command address register. Recall that when the 'CCW valid' latch is on, that 'service out' responses to 'service in' signals are inhibited.

Following turn-on of the 'CCW valid' latch, the 'sequence 4' latch is turned on by AND'ing the 'CCW valid' signal and the 'T6' signal (already active). The channel now enters a sequence 4, read CDA routine to determine whether a chain check condition exists; detection of a chain check condition results in a sequence 5 ending routine to terminate the read CDA operation. If a chain check condition is not detected, the channel must then add the DAB and count, and gate the DAB bits into the byte counter.

After the turn-on of the 'sequence 4' latch, the channel samples the status of the number of data bytes gated into the channel versus the value of the DAB for the second CCW. If the 'read doublegating' signal is inactive, indicated by an active bit 4 from the byte count register, the DAB must be on doubleword boundaries (DAB = 0); otherwise, a chain check condition is detected and the 'chain check' latch is turned on. If the 'read CDA doublegating' signal is not off and the byte count latches equal zero, the channel has not received any data bytes from the control unit. In this case, the DAB may specify any byte boundary within the first data doubleword (DAB equals any value from 0 through 7) and a chain check condition will not be detected.

If the 'read CDA doublegating' signal is active and the byte count latches do not equal zero, the channel has received between one and three data bytes and the DAB must specify either singleword (DAB = 4) or doubleword (DAB = 0) boundaries; otherwise, a chain check condition is detected, and the 'chain check' latch is turned on. If the 'read CDA doublegating' signal is active and the DAB specifies singleword boundaries,bits 0 through 31 of the B-register and bits 0 through 3 of the mark-B register are reset, since only the data in the last four byte positions (bits 32 through 63) of the B-register are to be stored. If the DAB specifies doubleword boundaries, the doublegating operations (when the 'inhibit service out' signal is dropped) are allowed to continue until the third data byte is gated into B-register byte positions 2 and 6. After the bytes are gated, bits 32 through 63 of the B-register and bits 4 through 7 of the mark-B register are reset, and the remaining data bytes comprising the first data doubleword are singlegated into the last five byte positions.

Note: If the operation described above was for a read backward CDA operation, the resets of the B-register and mark-B register bits would be the reverse of that described for the read CDA operation.

Other conditions may also cause a chain check to be detected. Assume that the B-register has been filled (eight bytes have been received) and that the B-register contents are subsequently transferred to the A-register prior to turn-on of the 'CCW valid' latch. If the DAB is not on doubleword boundaries, the 'chain check' latch is immediately turned on. However, assume that the A-register is full and that the DAB specifies doubleword boundaries. In this case, the count from the second CCW is examined for a count equal to or greater than 16 bytes (two doublewords). If the count is less than 16, the 'chain check' latch is turned on. If the count is greater than 16 bytes, the condition of the count in the byte count register is examined for a count of 7 (BCL = 0 condition). If a count of 7 is detected, the 'chain check' latch is turned on; otherwise, the 'chain check' latch remains off and the first data doubleword in the A-register is still valid. (In this case, the possibility of an overrun condition still exists, since the channel may be unable to store the first data doubleword before all bytes of the second data doubleword are gated into the B-register. In summary, if the first data doubleword is transferred to the A-register before the

'CCW valid' latch is on, an overrun condition will not be detected except when the DAB does not equal 0 or when eight bytes of the second data doubleword have been gated into the byte register.

In yet another case, the channel compares the count from the second CCW with the value in the byte counter. If the count value in the byte counter is greater than the CCW count, the 'chain check' latch is turned on, since more data has already been accepted by the channel than required by the CCW.

Assuming that a chain check condition is not detected (and, for simplification, that all data bytes for the first data doubleword of the second CCW are not yet in the B-register), the channel clock is turned off by the 'sequence 3' and 'sequence 4' signals. With the clock off, the 'sequence 4' and 'not T1' signals turn off the 'sequence 3' latch. With the 'sequence 4' latch on, and the 'sequence 3' latch off, the clock is turned on to time the remaining second CCW setup operations. At 'T0' clock time, the DAB (bits 21 through 23 of the data address register) plus the count in the count register are gated to the adder. At the same time, the DAB is OR'ed into the byte count register where it is effectively added to the count existing in the register (see "Byte Counter Circuits" in Chapter 2). If the byte counter parity is affected when the DAB is OR'ed with the existing byte count value, the byte counter circuits correct the parity bit.

At 'T4' clock time, the DAB plus count value is latched into the adder and subsequently gated into count register. The count register now contains a modified count which is used to detect the gating of the last data byte for the last data doubleword of the second CCW. When the adder output is gated into the count register, the DAB in the data address register is reset to all 0's. Prior to the reset operation, the DAB is examined to determine if the reset will affect the P2 parity bit in the data address register. If the bit is affected, it is changed at the 'P2 bit' latch just prior to the DAB reset operation.

At 'T6' clock time, the 'sequence 4' latch is turned off, causing the 'CDA' latch to turn off. With the turn-off of the 'sequence 4' latch and the 'CDA' latch, the 'inhibit service out' level is dropped, and the channel may again respond to 'service in' signals from the control unit. Note that if the DAB of the second CCW equals 0, the channel may respond to 'service in' signals during sequence 4 operations.

If the first data doubleword has previously been transferred to the A-register, (as may have been the case where more than nine data bytes were transferred to the channel by the control unit before the 'CCW valid' latch was turned on) the 'sequence 3' latch is turned on. Turn-on of the 'sequence 3' latch is performed by the 'read and BC equal zero' signal. With the 'sequence 3' latch on, the 'storage request' latch is turned on to request a storage cycle to store the first data doubleword of the second CCW.

Note: For the case described above, the first data doubleword for the second CCW was transferred from the B-register to the A-register upon detection of a 'BCL = 0' signal. This signal enables turn-on of the 'B to A read CDA' latch (ALD LT121). The active output of this latch turns on the 'B to A' latch and 'mark-B to mark-A' latch, causing transfer of the B-register data to the A-register and mark-B register bits to the mark-A register. In this special case, the status of the 'B-register full' and 'A-register full' latches are not changed (both off). When the 'sequence 3' routine begins, the 'A-register full' latch is turned on at 'T2' clock time.

To summarize the completion of the second CCW setup operations, (1) the first data doubleword or second data doubleword handling operations are in progress, and (2) if the first data doubleword has already been transferred to the A-register, a sequence 3 routine is entered to store the first data doubleword of the second CCW while data handling operations for the second data doubleword are in progress.

From this point on, the read CDA operations continue as for a normal read operation. If the second CCW contains an active chain data flag bit, the read CDA operations described to fetch the next CCW are as described for fetching the second CCW.

HALT I/O INSTRUCTION

- 'Halt I/O' signal to channel with interrupt condition pending causes 'release' signal and CC 0 to be sent to CPU interface; interrupt condition remains pending.

- 'Halt I/O' signal to busy channel causes channel to:
  1. Perform I/O interface disconnect and stop channel operation.
  2. Turn on 'interrupt' latch.
  3. Send 'release' signal and CC 2 to CPU interface.
  4. Leave channel with interrupt condition pending.

- 'Halt I/O' signal to free channel (not busy) causes channel to attempt I/O device selection using unit address accompanying 'halt I/O' signal.

- No selection condition results in:
  1. 'Release' signal and CC 3 to CPU interface.
  2. Channel resuming polling operations.

- Wrong device selection (no address compare condition) causes channel to:
  1. Turn on 'interface control check' latch.
  2. Perform I/O interface reset.

- Correct device selection (address compare condition) causes channel to disconnect I/O device from I/O interface.

- Wrong or correct device selection causes channel to:
  1. Store CSW (status bytes only).
  2. Send 'release' signal and CC 1 to CPU interface.
  3. Begin polling operations.

Simplified Halt I/O Operations

The primary purpose of channel halt I/O operations (Figure 3-4) is to terminate a channel operation (channel busy condition) and disconnect the I/O device with which the channel is working from the I/O interface. If the channel is busy when the Halt I/O instruction is received, the channel forces an incorrect length condition, halts channel operations, disconnects the I/O device, and sends a 'release' signal and condition code 2 to the CPU interface. At the end of the halt I/O operation, the channel has an interrupt condition pending.

If the channel has an interrupt condition pending when the 'halt I/O' signal is received, channel operations are limited to sending of a 'release' signal and condition code 0 to the CPU interface. The interrupt condition remains pending in the channel.

If the channel is not busy and does not have an interrupt condition pending when the 'halt I/O' signal is received, the channel attempts to select an I/O device. The device to be selected is designated by the unit address accompanying the 'halt I/O' signal. If no I/O device responds during the attempted selection, the channel sends a 'release' signal and condition code 3 to the CPU interface and resumes normal polling operations. If an I/O device does respond during the attempted selection, the channel compares the address sent to the I/O interface with the one received to determine if the addressed I/O device is the one responding. A successful comparison causes the channel to perform an I/O interface disconnect operation; an unsuccessful comparison (interface control check condition) causes the channel to perform an I/O interface reset operation. In either case, the channel turns on the 'interrupt' latch followed by turn-on of the 'pseudo accept interrupt' latch (to simulate the 'interrupt response' signal from the CPU). The channel then stores the status bytes portion of the CSW at storage location 64. Following the CSW store operation, the channel sends a 'release' signal and condition code 1 to the CPU interface and resumes normal polling operations.

Note that (Figure 3-4) and I/O device selection is attempted only if the channel is not busy and an interrupt condition is not pending; thus, the unit address accompanying the 'halt I/O' signal is used only under these conditions. If the channel is working (busy) with an I/O device, channel halt I/O operations cause the I/O device to stop operations, regardless of the device address. In this case, the unit address accompanying the 'halt I/O' signal is not used by the channel.

Detailed Halt I/O Operations

Channel halt I/O detailed operations are described in the following paragraphs. Descriptions are based upon the detailed flow chart shown on Diagram 5-15.

Halt I/O Channel Available

In order for the channel to perform halt I/O operations, the 'channel available' signal to the CPU interface must be active (Diagram 5-15). If the CPU decodes a Halt I/O instruction to the channel and the 'channel available' signal is inactive, the CPU internally generates condition code 3 and does not select the channel for a halt I/O operation. If the 'channel available' signal is active, the channel receives a 'halt I/O' signal, a 'select channel' signal, and a unit address byte via the CPU interface. The unit address byte is placed on the 'unit address bus-out' lines.

Halt I/O Interrupt In Channel

- 'Interrupt' signal active causes:
  1. Channel to inhibit signals to two condition code lines (CC = 0).
  2. Channel to send 'release' signal to CPU interface.
  3. 'Halt I/O' and 'select channel' signal to drop.
  4. Channel interrupt condition remains pending.

- 'Interrupt' signal inactive causes channel to enter halt I/O channel free or halt I/O channel busy routine, depending upon whether 'path working' signal is active or inactive.

Upon receipt of the 'halt I/O' and 'select channel' signals (Diagram 5-15), channel logic is examined for a polling interrupt operation in progress. If the channel is performing a polling interrupt operation, halt I/O operations are suspended until the channel stacks status back to the I/O device. If a polling interrupt operation is not in progress, the condition of the channel 'interrupt' latch is examined by AND'ing the 'halt I/O' and 'select channel' signals. If

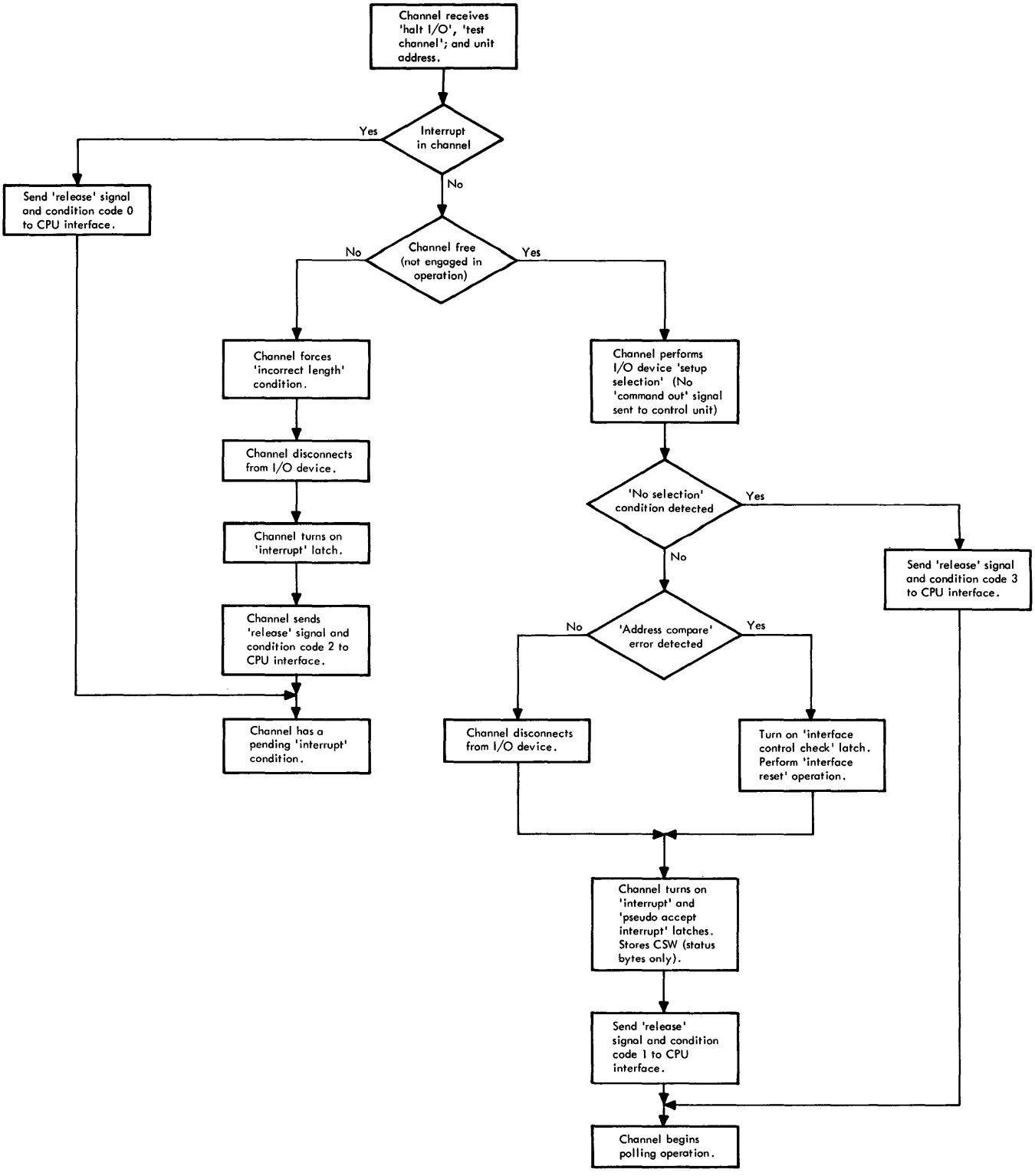


Figure 3-4. Halt I/O Operation, Simplified Flow Chart

the 'interrupt' signal is active, the channel inhibits the activation of the two condition code signals to the CPU interface. This represents condition code 0. Subsequently, the channel activates the 'release' signal to the CPU interface. This causes the 'halt I/O' and 'select channel' signals to the channel to drop; the interrupt condition remains pending in the channel until the channel receives an 'interrupt response' signal or test I/O instruction from the CPU interface.

If the 'interrupt' latch is off, channel halt I/O operations depend upon whether the channel is free (not engaged in an operation other than polling). If the channel is free ('not path working' signal), the channel performs halt I/O, channel free operations. (Refer to "Halt I/O Channel Free".) If the channel is not free ('path working' signal), the channel performs halt I/O, channel busy operations described in the following paragraph.

Halt I/O Channel Busy

- 'Halt I/O channel busy' latch turns on.
- 'Interrupt status' signal activated.
- If 'sequence 2' latch is on, 'incorrect length' latch is turned on.
- If 'sequence 5' latch is off, the latch is turned on.
- If 'incorrect length' latch is on, 'sequence 5' latch is turned on.
- 'Halt I/O address in' signal activated.
- 'Halt I/O reset' latch turns on; 'address out' latch turns off, if on.
- 'Address out' signal forced to I/O interface.
- 'Status in end' signal activated.
- Clock turns on.
- 'Select out' latch turns off ('T4' time).
- Control unit drops 'operational in' signal.
- 'Interrupt' latch turns on.
- 'Sequence 5' and 'sequence 2' latches turn off.
- 'Code 2' latch turns on (CC 2 to CPU interface).
- 'Release' signal sent to CPU interface.
- 'Halt I/O' and 'select channel' signals drop.
- 'Halt I/O channel busy' latch and 'code 2' latch turn off.
- 'Interrupt' latch remains on.

'Halt I/O channel busy' operations (Diagram 5-15) are performed to halt channel operations and disconnect the I/O device with which the channel is working.

Halt I/O channel busy operations are initiated when the 'path working' signal is active and the channel receives the 'halt I/O' and 'select channel' signals. These signals turn on the 'halt I/O channel busy' latch to begin the operation. The 'halt I/O channel busy' signal activates the 'interrupt status' signal. This signal causes the 'incorrect length' latch to turn on if the 'sequence 2' latch is on; i.e., channel is engaged in a read-type or write-type operation which has not ended. If the 'sequence 2' latch is off, the 'interrupt status' signal turns off the 'chain command' latch, if on. (If the latch is on, the channel is engaged in a chain command operation which is terminated by the halt I/O operation.)

If the channel 'sequence 5' latch is off (channel had not entered sequence 5 routine when halt I/O channel busy operation began), the 'sequence 5' latch is turned on by the 'read WLR (incorrect length)' or 'write WLR (incorrect length)' signal, depending upon whether a read or write operation is in progress.

If a storage cycle (fetch or store operation) is in progress, the 'storage cycle' trigger is on. If the trigger is on, halt I/O operations cannot proceed until the storage cycle is complete and the trigger is turned off. With the 'storage cycle' trigger off, the 'halt I/O address in' signal is activated by the 'not storage cycle', 'sequence 1 or 2' and 'not status in' signals.

The 'halt I/O address in' signal: (1) is AND'ed with the 'operational in' signal to turn on the 'halt I/O reset' latch; and (2) turns off the 'address out' latch, if on (with the 'address out' latch off, 'address out' signals to the I/O interface are under control of the 'halt I/O' logic).

The 'halt I/O reset' signal activates the 'address out' signal to the I/O interface and activates the channel 'status in end' signal. Subsequently, the channel clock is turned on by AND'ing the 'address out', 'select out' and 'halt I/O reset' signals.

Approximately 400 ns later, the 'T4' signal turns off the 'select out' latch. With the 'address out' signal still active and the 'select out' signal inactive, the control unit recognizes an interface disconnect operation. Subsequently, the control unit stops operations and drops the 'operational in' signal to the channel. With the fall of the 'operational in' signal, the 'halt I/O reset' latch is turned off. The off condition of the latch causes the 'address out' and 'status in end' signals to drop. At this point, the control unit has been disconnected from the channel.

When the 'T7' clock signal is activated, the channel clock is turned off. This causes the 'not T1' and 'not select out' signals to turn on the 'interrupt' latch. In turn, the 'interrupt' signal: (1) turns off the 'sequence 5' latch; (2) turns off the 'sequence 2' latch; and (3) is AND'ed with the 'halt I/O channel busy' signal to turn on the 'code 2' latch. Turn-on of the 'code 2' latch activates condition code 2 to the CPU interface and triggers a 100-ns single-shot to turn on the 'release' latch. When the '100-ns singleshot' signal falls, the 'release' signal is gated to the CPU interface.

Subsequently, the 'halt I/O' and 'select channel' signals to the channel (via the CPU interface) drop. When these signals fall, the 'halt I/O channel busy' latch and 'code 2' latch turn off. With the 'code 2' latch off, the 'release' latch is turned off to deactivate the 'release' signal to the CPU interface. With the 'interrupt' latch still on, the fall of the 'release' signal completes the 'halt I/O channel busy' operation. The interrupt condition remains pending in the channel until cleared by another operation.

Halt I/O Channel Free

Halt I/O channel free operations (Diagram 5-15) are performed to attempt selection of an I/O device specified by the halt I/O unit address. The operation is performed only if the channel is not busy and an interrupt condition is not pending. Halt I/O channel free operations may result in a no selection condition or selection of an I/O device. For simplification, the channel free, no device selection and channel free, device selection operations are described separately as follows.

Channel Free, No Device Selection. A summary and a detailed description of the channel free, no device selection operation follow:

- 'Select out' latch turns on.
- 'Halt I/O sync' latch turns on.
- 'Halt I/O channel free' latch turns on.
- 'Interrupt status' signal activated.
- Channel receives 'select in' signal.
- 'Select out' latch turns off, 'setup' latch turns on, 'unit address bus-out' bits gated to unit address register.
- Clock turns on.
- Unit address register contents gated to 'bus-out' lines.
- 'Setup reset' signal activated.
- 'Address out' latch turns on.
- 'Select out' latch turns on.
- Channel receives 'select in' signal.
- 'No selection' latch turns on.
- 'Address out' latch turns off; 'no select gated' signal is activated.
- Condition code 3 gated to CPU interface.
- 'Release' signal sent to CPU interface.
- 'Halt I/O' and 'select channel' signals drop.
- 'Halt I/O channel free' latch turns off.
- 'Setup' latch turns off.
- 'No selection' latch turns off.
- Channel resumes polling operation.

Channel free, no device selection operations are initiated with the turn-on of the 'select out' latch with the 'halt I/O', 'select channel' and 'not path working' signals. Subsequently, the 'halt I/O sync' latch is turned on with the 'sync on not select in' signal. The channel free condition is then established in the channel by turn-on of the 'halt I/O channel free' latch with the 'halt I/O sync' signal. The 'halt I/O channel free' signal activates the 'interrupt status' signal to inhibit certain functions that may be performed during other I/O device selection and disconnect operations.

The channel then awaits receipt of the 'select in' signal from the I/O interface. When received, the 'select in' signal: (1) turns off the 'select out' latch; (2) turns on the 'setup' latch; and (3) gates the 'unit address bus-out' bits into the unit address register. The channel now contains the unit address of the I/O device to be selected during the setup operation.

With the 'setup' signal active, the channel clock is turned on to time selection operations and the unit address register contents are gated to the 'bus out' lines. The 'T0' clock signal then activates the 'setup reset' signal, causing selected channel latches and registers to be reset. This leaves the channel in an initial setup condition, prepared to attempt selection of the I/O device.

The 'address out' latch is then turned on by the 'T3' and 'not T4'; this raises the 'address out' signal to the I/O interface indicating to the control unit that a unit address byte is on the 'bus out' lines. After the fall of the 'select in' signal, the 'select out' latch is turned on by the 'T7' clock signal. This raises the 'select out' signal to the I/O interface to determine if the addressed I/O device is available. The channel then waits for either an 'operational in' or a 'select in' signal. An 'operational in' signal indicates that an I/O device has responded to the unit address byte. A 'select in' signal indicates that no device responded; i.e., the addressed device is not connected to the channel or is off line (not available).

Assume that the channel receives a 'select in' signal. The 'select in' signal turns off the 'select out' latch and AND's with the 'address out' signal to turn on the 'no selection' latch. In turn, the 'no selection' latch turns off the 'address out' latch and activates the 'no select gated' signal. The 'no select gated' signal activates condition code three to the CPU interface. After a deskewing delay, the channel sends a 'release' signal to the CPU interface, causing the 'halt I/O' and 'select channel' signals to the channel (via the CPU interface) to drop. With the drop of these signals, the 'halt I/O channel free' latch turns off causing the 'setup' latch to turn off. The 'not setup' signal turns off the 'no selection' latch to end the channel free, no device selection operation. The channel then resumes polling operations and is free to perform other operations.

Channel Free, Device Selection. A summary and a detailed description of the channel free, device selection operation follow.

- 'Operational in' signal received by channel.
- 'Address out' latch turns off.
- 'Address in' signal received by channel.
- Clock turns on.
- Unit address in 'bus in' latches and unit address register contents compared.
- If addresses do not compare:
  1. 'Incorrect selection' signal activated.
  2. 'Interface control check' latch turns on.
  3. 'Machine check' signal activated.
  4. 'Interface control check SS' signal activated.
  5. 'Suppress out' latch turns on; 'select out' latch turns off.
  6. 'Interface reset' signal (6 usec) degates 'operational out' signal; attached I/O device resets.
  7. 'Interrupt' latch turns on.
  8. 'Interface reset' signal falls; 'operational out' signal rises.
- If addresses compare:
  1. 'Setup' latch turns off.
  2. Clock turns off; 'halt I/O address in' signal activated.
  3. 'Sequence 5' latch turns on.
  4. 'Halt I/O reset' latch turns on.
  5. 'Address out' and 'status in end' signals activated.
  6. Clock turns on.
  7. 'Select out' latch turns off.
  8. 'Operational in' signal to channel drops.
  9. 'Halt I/O reset' latch turns off.
  10. 'Address out' and 'status in end' signals deactivated.
  11. Clock turns off.
  12. 'Interrupt' latch turns on.
  13. 'Sequence 5' latch turns off.
- 'Pseudo accept interrupt' latch turns on.
- 'Interrupt storage request' latch turns on.
- 'Storage request' signal to BCU interface activated.
- 'Z-address' latch turns on; 'storage cycle' trigger turns on.
- Channel receives 'BCU response' signal.
- 'Latch Z-address' latch turns on; 'Z-address latch gated' signal activated.
- Mark bits 4, 5, and P gated to mark lines; storage protect P bit to 'storage protection' lines; address 64 decimal to SAB; and 'store' signal to BCU interface 'store' line.
- Channel receives 'BCU data request' signal; 'storage request' signal to BCU interface degated.
- CSW channel and control unit status bytes gated to SBI lines.
- Channel receives 'accept' signal; 'accept' latch turns on and 'interrupt' latch turns off.
- Channel receives 'BCU advance pulse' signal.
- 'Storage cycle' trigger turns off.
- 'Remember interrupt' latch turns off.

- CC 1 gated to CPU interface.
- 'Release' signal gated to CPU interface.
- 'Halt I/O' and 'select channel' signals drop.
- 'Halt I/O channel free' latch turns off.
- 'Pseudo accept interrupt' latch turns off.
- 'Accept' and 'latch Z-address' latches turn off.
- 'Z-address' latch turns off.
- Channel resumes polling.

Channel free, device selection operations are performed by the channel when an attempted I/O device selection results in receipt of an 'operational in' signal from the I/O interface (Diagram 5-15). When the 'operational in' signal is received, it turns off the 'address out' latch. The channel then waits for the control unit to activate the 'address in' signal. Upon receipt of the 'address in' signal the channel clock is turned on to time further device selection operations. The channel then compares the unit address in the 'bus in' latches with the unit address in the unit address register. If the addresses compare, the specified I/O device has responded.

Assume that the addresses do not compare. In this case, the channel performs I/O interface operations to reset the I/O device on the I/O interface. Detection of the data bus compare check condition causes the 'T1' and 'not T3' signal to activate the 'incorrect selection' signal. This signal turns on the 'interface control check' latch which: (1) activates the 'machine check' signal; and (2) activates the 300-ns 'interface control check SS' signal which turns on the 'suppress out' latch. With the 'suppress out' latch on, the 'suppress out' signal to the I/O interface is raised. In addition, the 'interface control check SS' signal turns off the 'select out' latch, causing the 'select out' signal to deactivate at the I/O interface.

When the 300-ns 'interface control check SS' signal drops, the 6-usec 'interface reset' signal is activated by AND'ing the 'suppress out' and 'not interface control check SS' signals. The 'interface reset' signal: (1) degates the 'operational out' signal to the I/O interface causing the attached I/O device to perform a selective reset operation; and (2) turns on the 'interrupt' latch. In turn the 'interrupt' signal AND's with the 'halt I/O channel free' signal to turn off the 'setup' latch.

After the 6-usec 'interface reset' signal deactivates, the 'operational out' signal is again activated to the I/O interface. At this point, the I/O interface reset operation, initiated when the channel detected a data bus compare check condition is complete, and the channel is ready to begin storing the status bytes portion of the CSW (Diagram 5-15). The CSW store operation is described later in this text.

Assume that the channel does not detect a data bus compare check condition when the unit address register contents and the unit address from the I/O interface are compared. In this case, an interface reset operation is not performed; instead, the channel performs an I/O interface disconnect operation.

The disconnect operation begins with the turn-off of the 'setup' latch by the 'T4' clock signal. When the 'setup' latch turns off, the channel clock is turned off and the 'halt I/O address in' signal is activated. Activation of this signal turns on the 'sequence 5' latch which, in turn, causes the 'halt I/O reset' latch to turn on. The 'halt I/O reset' signal activates the 'address out' signal to the I/O interface and activates the 'status in end' signal.

To time the subsequent disconnect operations, the channel clock is then turned on by AND'ing the address out, select out, and halt I/O reset operations. When the 'T4' clock signal is activated, the 'select out' latch is turned off causing the 'select out' signal to the I/O interface to deactivate. With the 'address out' signal active and the 'select out' signal inactive, the control unit recognizes an I/O interface disconnect operation and drops the 'operational in' signal.

The 'not operational in' signal turns off the 'halt I/O reset' latch which deactivates the 'address out' and 'status in end' signals. Subsequently, the 'T7' signal turns off the channel clock. With the clock off, the 'interrupt' latch is turned on by AND'ing the 'not T1' and 'not select out' signals. The 'interrupt' signal turns off the 'sequence 5' latch to complete the I/O interface disconnect operation. The channel is now prepared to begin operations to store the status byte portion of the CSW.

Channel operations to store the CSW begin after an I/O interface reset or I/O interface disconnect operation has been completed (Diagram 5-15). In either case, the CSW store operations begin when the 'interrupt' latch is turned on.

The 'interrupt', 'halt I/O channel free' and 'not interface reset' signals are AND'ed to turn on the 'pseudo accept interrupt' latch. (The 'pseudo accept interrupt' signal simulates receipt of an 'interrupt response' signal from the CPU interface.) The active 'pseudo accept interrupt' signal causes the 'interrupt request storage' latch to turn on. With this latch on, the 'storage request' signal to the BCU interface is activated to request a storage cycle to store the CSW; in addition, the 'interrupt storage request' signal turns on the 'Z-address' latch and the 'storage cycle' trigger. The 'Z-address latch' signal enables AND's at the SAB gating logic for address 64 (decimal), and the 'storage cycle' signal activates and inhibits applicable channel logic for the storage cycle operation.

When the channel receives the 'BCU response' signal, the 'latch Z-address' latch is turned on to activate the 'Z-address latch gated' signal. The 'Z-address latch gated' signal: (1) turns off the 'suppress out' latch, if on as the result of an I/O interface reset operation; (2) is AND'ed with the 'BCU response' signal to gate mark bits 4, 5 and P to the BCU interface mark lines; and (3) is AND'ed with the 'remember interrupt' signal to gate the channel and control unit status bytes to the SBI lines. The 'BCU response' signal is also: (1) AND'ed with the 'Z-address latch' signal at the SAB gating logic to gate address 64 (decimal) to the SAB; and (2) AND'ed with the 'remember interrupt' signal to force the storage protect parity bit to the BCU interface 'storage protection' parity line and to activate the 'store' signal to the BCU interface.

Subsequently the channel receives the 'BCU data request' signal (or, for the Model 91, generates the signal). The 'BCU data request' signal gates the status bytes to be stored to the SBI lines. Also, the 'BCU data request' signal degates the 'storage request' signal to the BCU interface. The channel then awaits receipt of the 'accept' signal from the BCU interface. When received, the 'accept' signal turns on the 'accept' latch and turns off the 'interrupt' latch. The 'accept latch' signal then turns off the 'interrupt storage request' latch.

When the 'BCU advance pulse' signal is received, the channel delays the signal to activate the 'late advance' signal and then the 'storage cycle complete' signal. Activation of the 'late advance pulse' signal turns off the 'storage cycle' trigger. In turn, the 'not storage cycle' signal turns off the 'remember interrupt' signal. (By this time, the 'BCU response' signal has fallen, degating the address information and 'store' signal to the BCU interface.)

Condition code 1 is then gated to the CPU interface by AND'ing the 'not remember interrupt' and 'pseudo accept interrupt' signals. After a deskewing delay of approximately 100 ns, the 'release' signal is activated to the CPU interface lines. This causes the 'halt I/O' and 'select channel' signals to the channel to drop. With these two signals inactive, the 'halt I/O channel free' latch is turned off, causing the 'pseudo accept interrupt' signal to deactivate.

When the 'storage cycle complete' signal is activated ('BCU advance pulse' signal delayed), the 'latch Z-address' latch is turned off. The 'not latch Z-address' signal turns off the 'Z-address' latch to end the 'channel free, device selection' operation. The channel resumes polling operations and is free to perform another operation.

TEST I/O INSTRUCTION

Test I/O channel operation are dependent upon the state of the channel at the time the 'test I/O' and 'select channel' signals and unit address are received via the CPU interface. The channel, upon receipt of these signals, may: (1) be engaged in another operation; (2) have an interrupt condition pending; or (3) be in a not busy condition (not engaged in an operation with no interrupt condition pending). The primary purpose of the test I/O instruction is to clear a pending interrupt condition pertaining to a specified I/O device or to test the condition of a specified I/O device. Whether the primary purpose of the test I/O instruction is accomplished depends upon the channel conditions previously described. Channel test I/O operations are described in the following text at first the simplified flow chart level and then at the detailed flow chart level.

Simplified Test I/O Operations

- Channel busy condition causes channel to send CC = 2 and 'release' signal to CPU interface; test I/O operation complete.

- Interrupt condition in channel causes unit address from CPU interface to be compared with unit address register contents:
  1. If addresses match, CSW is stored; CC = 1 and 'release' signal sent to CPU interface (interrupt cleared).
  2. If addresses do not match, CC = 2 and 'release' signal sent to CPU (interrupt remains pending).
  3. If channel stopped for logout, address comparisons not made; logout operation performed, CSW stored, CC = 1, and 'release' signal generated (interrupt cleared).

- Channel not busy, no interrupt condition in channel; gate test I/O unit address into unit address register.

- Channel attempts device selection.

- If 'select in' signal received (no selection), CC = 3 (device not available) and 'release' signal sent to CPU.

- If 'status in' signal received, control unit busy sequence performed; CSW stored; CC = 1 and 'release' signal sent to CPU interface.

- If 'address in' and 'operational in' signals received, unit address from control unit compared with unit address register contents.
  1. No comparison results in interface control check condition, I/O interface reset, CSW store, and CC = 1 and 'release' signal to CPU interface.
  2. Comparison causes channel to obtain status byte from control unit.

- Channel examines status byte for all zeros:
  1. Status equals 0 condition causes CC = 0 and 'release' signal to be sent to CPU interface.
  2. Not status equals 0 condition causes CSW store and CC = 1 and 'release' signal to be sent to CPU interface.

Channel test I/O operations are initiated when the channel receives the 'test I/O' and 'select channel' signals accompanied by a unit address byte (Diagram 5-16). If the channel is engaged in an operation (channel busy condition), the channel sends condition code 2 and a 'release' signal to the CPU interface. This causes the 'test I/O' and 'select channel' signals to fall and channel test I/O operations are complete. Condition code 2 indicates the busy condition to the CPU.

If the channel is not busy, contains an interrupt condition, and is not stopped pending a logout operation, the channel compares the test I/O unit address (on the 'unit address bus-out' lines) with the interrupt unit address in the unit address register. If the addresses do not compare, the channel sends condition code 2 and a 'release' signal to the CPU interface; the interrupt condition remains pending. In otherwords, the interrupt condition pertains to an I/O device other than that specified by the Test I/O instruction. Condition code 2 to the CPU interface indicates that access to the I/O device specified by the test I/O address is not possible due to a busy (working) condition. The 'release' signal causes the 'test I/O' and 'select channel' signals to drop, and channel test I/O operations are complete.

If the two addresses do compare, the channel clears the interrupt condition by controlling the storage of the CSW (all bytes). When the CSW is stored, the channel sends condition code 1 and the 'release' signal to the CPU interface. Condition code 1 indicates that the CSW has been stored. The 'release' signal causes the 'test I/O' and 'select channel' signals to drop. With the fall of these signals, channel test I/O operations are complete and the channel resumes polling operations.

If the channel is stopped for a logout operation with the 'log wait interrupt' latch on, receipt of the 'test I/O' signal causes the channel to perform the logout operation. Following the logout operation, the channel turns on the 'interrupt' latch, stores the CSW (all bytes), and sends condition code 1 and the 'release' signal to the CPU interface. During the CSW store operation, the interrupt condition is cleared. Condition code 1 indicates that the CSW was stored. The 'release' signal causes the 'test I/O' and 'select channel' signals to drop. This completes channel test I/O operations and the channel resumes polling operations.

If the channel is not busy and an interrupt condition is not pending when the 'test I/O' and 'select channel' signals are received (Diagram 5-16), the channel attempts to select the I/O device specified by the test I/O unit address. With these conditions present, the channel gates the test I/O unit address from the 'unit address bus-out' lines into the unit address register. The channel then attempts to select the I/O device (gates the unit address register contents to the 'bus out' lines and activates the 'address out' and 'select out' signals). If a 'bus-out parity' error is detected after the unit address is gated to the 'bus-out' latches, the channel terminates the test I/O operation with a channel control check condition; the CSW is subsequently stored, condition code 1 and the 'release' signal sent to CPU interface, and the channel resumes polling operations.

If a 'bus out parity' error is not detected, the channel awaits either a 'select in' signal, 'status in' signal, or the 'address in' and 'operational in' signals. If a 'select in' signal is received, the addressed I/O device was not selected (Diagram 5-16). In this case, the channel sends condition code 3 and a 'release' signal to the CPU interface. Condition code 3 indicates that the I/O device addressed by the test I/O instruction is not available (power is off, device is not attached in the system, or a malfunction of some type exists). The 'release' signal causes the 'test I/O' and 'select channel' signals to drop, ending the test I/O operation and allowing the channel to resume polling operations.

If a 'status in' signal is received, rather than a 'select in' signal, the addressed I/O device or control unit is busy. This causes the channel to enter a control unit busy sequence, during which an I/O interface disconnect operation is performed. Subsequently, the channel stores the CSW (all bytes) in main storage, sends condition code 1 and the 'release' signal to the CPU interface (Diagram 5-16). The channel then resumes polling operations.

If the 'address in' and 'operational in' signals are received by the channel, the channel compares the unit address received from the control unit with the unit address register contents. If the addresses compare, the addressed I/O device has been selected. In this case, the channel requests and receives the status byte from the control unit. If the status bits equal 0, the I/O device is available for operation. The status equals 0 condition causes the channel to send condition code 0 and the 'release' signal to the CPU interface. Subsequently, the 'test I/O' and 'select channel' signals drop to end the test I/O operation and permit the channel to resume polling operations.

If the status bits equal other than all 0's, the I/O device is not available for operation. The not status equal 0 condition causes the channel to store the CSW, send condition code 1 and the 'release' signal to the CPU interface, and resume polling operations.

If the correct device was not selected (Diagram 5-16), the 'interface control check' latch is turned on. Providing the LOG ON MACH CHK switch is off, the channel: (1) performs an I/O interface reset operation; (2) stores

the CSW; (3) sends condition code 1 and the 'release' signal to the CPU interface; and (4) resumes polling operations. If the LOG ON MACH CHK switch is on, the channel performs a logout operation, then stores the CSW and sends condition code 1 and the 'release' signal to the CPU interface.

Note that for all test I/O operations involving a CSW store operation, condition code 1 is sent to the I/O interface. However, in each case, the content of the CSW is different, and is available for use by the CPU, if required.

Detailed Test I/O Operations

Channel test I/O operations are described in detail in the following paragraphs. The detailed descriptions are based upon Diagram 5-17. Test I/O operations are described for the following conditions: (1) channel not available; (2) channel busy; (3) interrupt in channel; and (4) channel not busy, no interrupt.

Channel Not Available

When the test I/O instruction is decoded by the CPU, the CPU examines the 'channel available' signal from the channel to which the Test I/O instruction pertains (Diagram 5-17). If the 'channel available' signal is inactive, the CPU internally generates condition code 3 and does not send the test I/O, select channel, and unit address byte to the channel. However if the 'channel available' signal is active, condition code 3 is not internally generated and the multiplex 'test I/O' signal and simplex 'select channel' signal are sent to the channel. In addition, the unit address byte is placed on the 'unit address bus-out' lines to the channels. With these signals and unit address present at the channel, channel test I/O operations are initiated.

Channel Busy Operations

- 'Channel working' signal active.
- 'Test I/O' signal turns on 'code 2' latch; CC = 2 to CPU interface.
- 'Release' latch turns on.
- After 100 ns, 'release' signal gated to CPU interface.
- 'Test I/O' and 'select channel' signals fall.
- 'Code 2' and 'release' latches turn off.
- Channel operations not disrupted.

If the channel is busy ('channel working' signal active) when the 'test I/O' signal is received by the channel, the 'code 2' latch is turned on (Diagram 5-17). This latch is turned on by AND'ing the 'channel working', 'test I/O' and 'select channel' signals. With the latch on, condition code 2 is present on the two condition code lines to the CPU interface. The 'code 2' latch signal also triggers a 100-ns singleshot; the singleshot output turns on the 'release' latch. When the '100-ns SS' signal falls, the 'release' signal is gated to the CPU interface. Subsequently, the 'test I/O' and 'select channel' signals drop to turn off the 'code 2' latch and 'release' latch. Channel operations are not disrupted in the condition described.

If the 'channel working' signal is not active, the described operations are not performed.

Interrupt In Channel

- Clock turns on.
- 'Unit address bus-out' bits compared with unit address register contents.
- No comparison causes following:
  1. 'Code 2' latch turns on ('T3' time).
  2. 100-ns singleshot fires.
  3. 'Release' latch turns on.
  4. 'Release' signal gated to BCU interface when '100-ns singleshot' signal drops.
  5. 'Test I/O' and 'select channel' signals drop.
  6. Clock turns off.
  7. 'Code 2' and 'release' latches turn off.
- Address comparison causes channel to clear interrupt condition.
- 'Pseudo accept interrupt' latch turns on ('T3' time).
- 'Interrupt storage request' latch turns on.
- 'Storage request' signal rises; 'Z-address' latch and 'storage cycle' trigger turn on.
- Channel receives 'BCU response' signal.
- 'Latch Z-address' latch turns on; 'Z-address latch gated' signal activated.
- Address 64 (decimal) gated to SAB; all mark bits to mark lines; P-bit to 'storage protection' lines; 'store' signal to BCU interface.
- Channel receives 'BCU data request' signal.

- CSW (all bytes) gated to SBI lines.
- Channel receives 'accept' signal.
- 'Accept' latch turns on; 'interrupt' latch, clock and 'interrupt storage request' latch turn off.
- Channel receives 'BCU advance pulse' signal.
- 'Storage cycle' trigger turns off; 'remember interrupt' latch turns off.
- Condition code 1 and 'release' signal sent to CPU interface.
- 'Test I/O' and 'select channel' signals drop.
- 'Pseudo accept interrupt' latch turns off.
- 'Latch Z-address' latch turns off; 'Z-address' latch turns off; 'accept' latch turns off.
- Channel resumes polling.

If the channel 'interrupt' latch is on when the 'test I/O' and 'select channel' signals are received, the channel attempts to clear the interrupt condition.

Note: If the channel is stopped awaiting initiation of the logout operation, the 'test I/O' latch is turned on and the logout operation is initiated; in this case, the interrupt condition is cleared at the end of the logout operation by a CSW store operation.

Assume that the channel is not stopped awaiting a logout operation and that the 'interrupt' latch is on. To initiate interrupt in channel operations, the channel clock is turned on by AND'ing the 'test I/O', and 'interrupt waiting' signals. At the same time, the test I/O unit address on the 'unit address bus-out' lines is compared with the contents of the unit address register. The unit address register contains the address of the I/O device for which the channel interrupt condition is pending.

If the unit addresses do not compare, the 'not compare' and 'T3' signals are AND'ed to turn on the 'code 2' latch (Diagram 5-17). With the latch on, condition code 2 is presented to the channel on the I/O interface 'condition code' lines. The 'code 2' signal also triggers a 100 ns single-shot which turns on the 'release' latch. When the '100-ns singleshot' signal falls, the 'release' signal is gated to the CPU interface. This causes the 'test I/O' and 'select channel' signals from the CPU interface to drop. The fall of the 'test I/O' signal turns off the channel clock ('turn on clock direct' signal falls). The 'not select channel' signal turns off the 'code 2' and 'release' latches. The test I/O operation is complete, and the interrupt condition remains pending in the channel.

If the unit addresses do compare (Diagram 5-17), the channel interrupt condition pertains to the I/O device addressed by the Test I/O instruction. To clear the interrupt condition, the channel 'pseudo accept interrupt' latch is turned on by AND'ing the 'not unit address compare check' and 'T3' signals. In the channel, the 'pseudo accept interrupt' signal simulates the 'interrupt response' signal (normally received from the CPU interface in response to the 'interrupt request' signal from the channel). In addition, the 'pseudo accept interrupt' signal inhibits turn-on of the 'test I/O' latch and turns on the 'interrupt storage request' latch. With the 'interrupt storage request' signal active, the 'storage request' signal to the BCU interface is activated and the 'Z-address' latch and 'storage cycle' trigger are turned on. The 'Z-address latch' signal conditions AND's at the SAB gating logic for address 64 decimal (Z-address). The 'storage cycle' trigger activates and inhibits channel logic for the CSW store operation. For example, the 'storage cycle' signal is AND'ed with the 'interrupt' signal to latch the 'remember interrupt' latch.

The channel then awaits receipt of the 'BCU response' signal from the BCU interface. When received, the 'BCU response' signal turns on the 'latch Z-address' latch. The output of this latch is AND'ed with the 'Z-address latch' signal to activate the 'Z-address latch gated' signal. With the 'BCU response', 'Z-address latch' and 'Z-address latch gated' signals active, address information and the 'store' signal are gated to the appropriate BCU interface lines. Address information consists of: (1) address 64 (decimal), gated to the SAB lines by the 'BCU response' and 'Z-address latch' signals; (2) all mark bits, gated to the mark lines by the 'BCU response' and 'Z-address gated' signals; and (3) the storage protect parity bit, gated to the 'storage protection' lines by the 'BCU response' and 'remember interrupt' signals. The 'store' signal is activated by the 'BCU response' and 'remember interrupt' signals.

The channel then awaits receipt of the 'BCU data request' signal from the BCU interface.

Note: Selector Channels used with the System/360, Model 91, generate the 'BCU data request' signal by delaying the 'BCU response' signal approximately 150 ns. When the 'BCU data request' signal is active, the CSW (all bytes) are gated to the SBI lines for storage in main storage location 64 (decimal).

When the channel receives the 'accept' signal (via the BCU interface): (1) the 'accept' latch is turned on; (2) the 'interrupt' latch is turned off by AND'ing the 'accept' and 'Z-address' latch signals; (3) the clock is turned off; and (4) the 'interrupt storage request' latch is turned

off by the 'accept latch' signal. At this point all information required to store the CSW is available at main storage and main storage has begun the store operation.

The channel then waits for a 'BCU advance pulse' signal indicating that the CSW has been stored. The channel delays the signal to form a 'late advance pulse' signal which turns off the 'storage cycle' trigger. In turn the 'not storage cycle' signal turns off the 'remember interrupt' latch.

The channel then activates condition code 1 (CSW stored) to the CPU interface by AND'ing the 'not remember interrupt' and 'pseudo accept interrupt' signals. The 'condition code 1' signal causes the 'release' signal to be sent to the BCU interface after an approximate 100-ns delay. The 'release' signal causes the 'test I/O' and 'select channel' signals to drop at the CPU interface. With the 'test I/O' signal inactive, the 'pseudo accept interrupt' latch is turned off.

The channel then waits activation of the 'storage cycle complete' signal (activated as a result of delaying the 'BCU advance pulse' signal). When the signal is active, the 'latch Z-address' latch is turned off, causing the 'Z-address' latch to turn off. The channel interrupt condition has been cleared, the CSW stored, condition code 1 sent to the CPU interface, and the CPU released. The channel now resumes polling operations and is free to perform another operation.

Channel Not Busy, No Interrupt

If the channel is not busy (polling) and the 'interrupt' latch is off when the Test I/O instruction is initiated, the channel attempts to select the I/O device to determine its availability. Since several conditions are possible during the attempted selection, channel not busy, no interrupt operations are described in separate paragraphs for the different alternatives. The detailed descriptions in each case are based upon Diagram 5-17.

Test I/O, Initiate Device Selection. A summary and a detailed description of the test I/O, initial device selection operations follow:

- Channel receives 'select in' signal.
- 'Test I/O' latch turns on.
- If 'polling interrupt' latch off, 'select out' latch turns on.
- Channel receives 'select in' signal (if 'address in' and 'operational in' signals received, channel performs 'polling interrupt' routine to stack status).
- 'Select in SS' signal activated; 'select out' latch turns off.
- 'Setup' latch turns on.
- Unit address bus-out bits gated into unit address register (regardless of whether 'polling interrupt' is pending).
- Unit address register contents gated to 'bus out' latches.
- Clock turns on.
- 'Setup reset' signal activated.
- 'Address out' latch turns on.
- Bus-out parity checked.
- No parity error causes following:
  1. 'Select out' latch turns on.
  2. 'Polling interrupt' latch turns off, if on.
  3. Clock turns off.

Test I/O, initiate device selection operation (channel not busy, no interrupt, initiate device selection designation on Diagram 5-17) begins when the 'select in' signal is inactive due to the channel performing a polling operation or due to a polling interrupt conditon in the channel. The 'not select in' signal is AND'ed with the 'not channel working' and 'CPU test I/O' signals to turn on the 'test I/O' latch. The 'test I/O latch' signal conditions channel logic to perform operations required during the attempt to select the I/O device. If the 'polling interrupt request' latch is off, the 'select out' signal is propogating through the attached control units. The channel then awaits either: (1) a 'select in' signal (indicating that no control unit has an interrupt condition pending) or (2) an 'address in' and 'operational in' signal (indicating that a control unit has an interrupt condition pending). In the latter case, the channel turns on the 'polling interrupt latch, gates the unit address from the control unit (via the 'bus in' latches) into the unit address register, and stacks status back to the control unit. The channel 'polling interrupt' latch remains on, but does not affect the test I/O operation (as will be described later in this text).

If the channel receives the 'select in' signal (rather than the 'operational in' signal), the 'select in' and 'not remember operational in' signals AND to activate the 2.2-usec 'select in SS' signal. This signal is delayed from the 'select in' signal received at the channel I/O interface by 2.2-usec. The delayed 'select in' signal turns off the 'select out' latch. From this point, the following channel operations are performed, regardless of whether the 'polling interrupt' latch is on or off.

The 'setup' latch is turned on by the 'select in or remember polling interrupt' signal to condition the channel for the attempted I/O device selection. At the same time, the test I/O unit address is gated into the unit address register by AND'ing the 'select in or remember interrupt' and 'test I/O latch' signals. Whether the unit address register contained a polling interrupt unit address is of no consequence, since the register is reset when the test I/O unit address is gated into the register.

Subsequently, the unit address register contents are gated into the 'bus out' latches (by the 'setup' signal) for presentation to the attached control units. The channel clock is then turned on by the 'setup' and 'not address out' signals. With the clock on, the 'T0' and 'not T1' signals activate the 'setup reset' signal which resets selected latches, triggers, and registers in the channel. At 'T4' clock time, the 'address out' latch is turned on, raising the 'address out' signal to the I/O interface. The unit address in the 'bus out' latches is then checked for correct parity. If parity is correct, the 'no errors' and 'T7' clock signal are AND'ed to turn on the 'select out' latch. With the 'select out' signal active (propogating through the control units), the channel clock is turned off to end the test I/O, initiate device selection operation. The channel may now proceed with the test I/O device response operation (Diagram 5-17). If a 'bus out' parity error is detected (unit address byte parity not odd), the channel enters a test I/O channel control check sequence (Diagram 5-17).

Test I/O Channel Control Check. A summary and a detailed description of the test I/O channel control check operation follow:

- 'Channel control check' latch turns on.
- 'Sequence 5' latch turns on.
- 'Command reject or control check' signal activated.
- Zeros gated to 'bus out' latches.
- Clock turns off.
- 'Interrupt status' signal activated.
- 'Interrupt' latch turns on.
- 'Remember interrupt' signal activated.
- 'Sequence 5' latch turns off.
- 'Address out' latch turns off.
- Enter test I/O, CSW store routine.

Test I/O channel control check operations are initiated when a parity error is detected in the unit address byte contained in the 'bus out' latches. Detection of the bus-out parity check condition turns on the 'channel control check' latch. With this latch on, the channel attempt to select the I/O device is unsuccessful. The 'channel control check' latch activates the 'command reject or control check' signal and turns on the 'sequence 5' latch. With the 'command reject or control check' signal active, all zeros with correct parity (stop command) are gated to the control units via the 'bus out' latches. At 'T7' clock time, the channel clock is turned off by the 'not address in' signal.

The 'control check' signal activates the 'interrupt status' signal which, in turn, activates the 'turn on interrupt end' signal. This signal turns on the 'interrupt' latch in preparation for a test I/O CSW store operation. With the 'interrupt' signal active, the 'sequence 5' latch is turned off and the 'remember interrupt' signal is activated. The 'remember interrupt' signal turns off the 'address out' latch to complete the test I/O channel control check operation. The channel now enters operations to store the CSW. (Refer to "Test I/O, CSW Store".)

Test I/O, No Selection. A summary and a detailed description of the test I/O, no selection operation follow:

- With 'address out' and 'select out' signals active, channel receives 'select in' signal.
- 'No selection' latch turns on.
- Send 'condition code 3' to CPU interface.
- Send 'release' signal to CPU interface.
- 'Test I/O' and 'select channel' signals drop.
- 'Setup' latch and 'test I/O' latch turn off.
- 'No selection' latch turns off.
- Channel resumes polling operations.

A test I/O, no selection operation (Diagram 5-17) is initiated when the 'address out' and 'select out' signals are active and the channel receives a 'select in' signal. When this occurs, the device addressed by the test I/O unit address is not available.

The 'select in' signal is AND'ed with the 'address out' signal to turn on the 'no selection' latch. The resulting 'no selection' signal activates the condition code 3 to the CPU interface, indicating that the I/O device is not available. The 'code 3' signal initiates activation of the 'release' signal to the CPU interface (after an approximate 100-ns delay).

The 'release' signal causes the 'test I/O' and 'select channel' signals to fall. The fall of the 'test I/O' signal turns off the 'setup' and 'test I/O' latches. In turn, the 'not setup' signal turns off the 'no selection' latch and the channel resumes polling operations. This completes the test I/O, no selection operation and, in actuality, completes the test I/O operation.

Test I/O Device Response. A summary and a detailed description of the test I/O, device response operations follow:

- 'Status in' signal or 'address in' and 'operational in' signals received.

- If 'status in' signal received, control unit or I/O device is busy:
  1. 'Control unit busy' latch turns on.
  2. 'Sequence 5' latch turns on.
  3. 'Address out' signal held active.
  4. Clock turns on.
  5. 'Select out' latch turns off.
  6. 'Status in' signal drops.
  7. 'Address out' signal deactivates.
  8. Enter test I/O, CSW store operation.

- If 'operational in' signal received, an I/O device has responded:
  1. 'Address out' latch turns off.
  2. 'Address in gated' signal activated.
  3. Clock turns on.
  4. Unit address register contents compared with unit address in 'bus in' latches.

- If addresses do not compare:
  1. 'Interface control check' latch turns on.
  2. 'Machine check' signal and 'interface reset' signal activated if LOG ON MACH CHK switch off.
  3. If LOG ON MACH CHK switch on, logout operation performed, then test I/O, CSW store operation performed.

- If addresses do compare, test I/O, correct device responded operations performed.

Test I/O, device response operations (Diagram 5-17), are performed by the channel to determine if the addressed I/O device is busy or available. With the 'address out' and 'select out' signals active, the channel awaits either a 'status in' signal or the 'operational in' signal from the control unit. (Assume that the 'select in' signal is not received to cause a test I/O, no selection operation.)

If the 'status in' signal is received, it indicates that the control unit or I/O device is busy. The 'status in' signal is AND'ed with the 'address out' signal to: (1) turn on the 'control unit busy' latch; (2) turn on the 'sequence 5' latch; (3) hold the 'address out' signal active so that the channel can force a disconnect operation and; (4) turn on the channel clock. With the 'control unit busy' signal active, a 'service out' signal response to the 'status in' signal is inhibited.

The 'select out' latch is then turned off by AND'ing the 'T4', 'not T5' and 'not halt I/O busy' signals. With the 'address out' signal active, and the 'select out' signal inactive, the control unit recognizes a disconnect and drops the 'status in' signal. The 'not status in' signal causes the channel to drop the 'address out' signal and enter the test I/O store CSW operation to store the CSW. (Refer to "Test I/O, CSW Store".)

Assume that the channel received the 'operational in' signal, rather than the 'status in' signal (Diagram 5-17). This means that an I/O device has responded to the selection attempt. The 'operational in' signal turns off the 'address out' latch; the 'address in' signal is AND'ed with the 'not start I/O' and 'not CC latch' signals to activate the 'address in gated' signal. The 'operational in' and 'address in gated' signals then turn on the channel clock to provide timing signals for the remaining test I/O, device response operations. With the clock on, the unit address register contents are compared with the unit address (in the 'bus-in' latches) received from the control unit. The comparison is performed to determine if the correct I/O device has been selected.

If the addresses do not compare, the 'data bus compare check' signal is activated and AND'ed with the 'T1' clock signal to turn on the 'interface control check' latch. If the LOG ON MACH CHK switch is on, the channel performs a logout operation, then performs a test I/O, CSW store operation to store the CSW. (Refer to "Test I/O, CSW Store".) If the LOG ON MACH CHK switch is off, the 'interface control check' signal activates the 'machine check' signal and 'interface reset' signal, then performs a test I/O, interface control check operation. (See "Test I/O, Interface Control Check".) Following the test I/O, interface control check operation the channel performs the test I/O, CSW store operation to store the CSW.

If the unit addresses do compare, the channel performs test I/O, correct device responded operations (refer to "Test I/O, Correct Device Responded"). Following this operation the channel performs the test I/O, CSW store operation to store the CSW, or performs the test I/O, device available operation (refer to "Test I/O, Device Available"). Which operation is performed depends upon the bit content of the status byte received during the test I/O, correct device responded operation.

Test I/O, Interface Control Check. A summary and a detailed description of the test I/O, interface control check operation follow:

- Operation results if unit addresses do not compare and LOG ON MACH CHK switch off.
- 'Interface control check SS' signal activated.
- 'Suppress out' latch turns on.
- 'Interface reset' signal activated.
- 'Test I/O, CSW store' operation begins.
- 'Operational out' signal drops.
- 'Operational in' and 'address in' signals drop.
- 'Interface reset' signal drops.
- 'Operational out' signal rises.

'Test I/O, interface control check' operations (Diagram 5-17), are performed when a unit address comparison results in turn-on of the 'interface control check' latch and the LOG ON MACH CHK switch is off. When the 'interface control check' latch is turned on, the latch output triggers a singleshot to activate the 300-ns 'interface control check SS' signal. This signal turns on the 'suppress out' latch and also activates the 6-usec 'interface reset' signal. With the 'suppress out' signal active, activation of the channel 'machine reset' signal is inhibited. This prevents channel latches, triggers and registers from being reset. The 'interface reset' signal causes the channel to begin the test I/O, store CSW operation and also degates the 'operational out' signal to the I/O interface. With the 'operational out' signal inactive, all attached operational control units perform a reset operation. This causes the 'operational in' and 'address in' signals from the control unit to drop. When the 6-usec 'interface reset' signal is deactivated, the 'operational out' signal is gated to the I/O interface. Thus, the test I/O, interface control check operation results in an I/O interface reset and entry into the test I/O, store CSW operation.

Test I/O, Correct Device Responded. A summary and a detailed description of the test I/O, correct device responded operation follow:

- Unit addresses compare.
- 'Setup' latch turns off.
- 0's gated to 'bus out' lines (proceed command).
- Channel activates 'command out' signal.
- 'Address in' signal from control unit drops.
- Channel drops 'command out' signal.
- 'Status in' signal received.
- 'Sequence 5' latch turns on.
- 'Latch status byte' trigger turns on.
- 'Service out' signal activated.
- 'Select out' latch turns off.
- 'Operational in' and 'status in' signal drop.
- 'Service out' signal deactivated.
- 'Status in end' latch turns off.
- Status byte in 'bus in' latches examined for all 0's.
- All 0's status causes channel to perform test I/O, device available operation.
- Not all 0's status causes channel to perform test I/O, CSW store operation.

Test I/O, correct device responded operations (Diagram 5-17) are performed if the unit address register contents and the unit address in the 'bus in' latches compare. The operation is initiated when the 'compare address in' signal AND's with the 'test I/O latch' signal to turn off the 'setup' latch. With the 'setup' latch off, all 0's with correct parity (proceed command) are gated to the 'bus out' latches. Subsequently, the 'T7' clock signal and 'address in gated' signal are AND'ed to activate the 'command out' signal at the I/O interface. The control unit responds to the proceed command by dropping the 'address in' signal to the channel. With the 'address in' signal inactive, the 'address in gated' signal is deactivated. This deactivates the 'command out' signal to the control unit.

The channel then waits for the control unit to place its status byte on the 'bus in' lines and raise the 'status in' signal. When the 'status in' signal is received, it is AND'ed with the 'test I/O latch' signal to turn on the 'sequence 5' latch. With the 'sequence 5' latch on, the 'status in', 'T2' and 'not T3' signals are AND'ed to turn on the 'latch status byte' trigger. The output of this trigger latches the status byte from the control unit into the 'bus in' latches. The 'T3' and 'not T4' signals are AND'ed with the 'status in' signal to turn on the 'status in end' latch; this causes the channel to send a 'service out' signal in response to the

'status in' signal. The 'select out' latch is then turned off by the 'T4' and 'not T5' signals, causing the 'select out' signal to the I/O interface to drop. The control unit responds by dropping the 'operational in' and 'status in' signals. At this point in the operation, the control unit is disconnected from the I/O interface, and the status byte is in the channel 'bus in' latches. With the fall of the 'status in' signal, the 'status in end' latch is turned off. The channel then examines the status byte in the 'bus in' latches for an all 0's condition. If the status bits are all 0's, the channel performs the test I/O, device available operation (refer to "Test I/O, Device Available). If the status bits are not all 0's the channel performs the test I/O, store CSW operation (refer to "Test I/O, Store CSW").

Test I/O, Device Available. A summary and a detailed description of the test I/O, device available operations follow:

- Correct device has been selected and status bits equal all 0's.
- Channel sends CC = 0 to CPU interface.
- Channel activates internal 'accept' signal.
- Channel sends 'release' signal to CPU interface.
- 'Test I/O' and 'select channel' signals drop.
- 'Setup' latch turns off, if on.
- 'Test I/O' latch turns off.
- 'Sequence 5' latch turns off.
- Channel resumes polling; test I/O operation complete.

The test I/O, channel available operation is performed when the channel has selected the I/O device specified by the test I/O unit address, has obtained a status byte of all 0's from the control unit, and has disconnected the control unit from the I/O interface. The operation is initiated when the 'not operational in' and 'not status in' signals inhibit activation of signals on the condition code lines to the CPU interface. The absence of signals on these lines represents condition code 0 to the CPU interface. To release the CPU, the 'status equal zero', 'sequence 5' and 'test I/O latch' signals are AND'ed to activate the 'accept' signal within the channel. This signal activates the 'release' signal to the CPU interface. Subsequently, the 'test I/O' and 'select channel' signals to the channel are dropped. In turn, the 'not test I/O' signal turns off the 'setup' and 'test I/O' latches. With the 'test I/O' latch off, the 'sequence 5' latch is turned off and the channel resumes polling operations. This completes the test I/O operation with the channel free to perform other operations.

Test I/O, CSW Store. A summary and a detailed description of the test I/O, CSW store operations follow:

- Performed as result of channel busy sequence, logout operation, status bits not equal to zero, interface control check condition, or channel control check condition.
- 'Interrupt' latch turns on (if off).
- 'Sequence 5' latch turns on (if off).
- Clock turns off.
- 'Polling interrupt' latch turns off (if on).
- 'Pseudo accept' latch turns on.
- 'Interrupt storage request' latch turns on; 'storage request' signal sent to BCU interface.
- 'Z-address' latch turns on; 'storage cycle' trigger turns on.
- Receive 'BCU response' signal.
- Turn on 'latch Z-address' latch; activate 'Z-address latch gated' signal.
- Turn off 'suppress out' latch, if on.
- Gate address 64 to SAB, P-bit to 'storage protection' lines, all mark bits to mark lines, 'store' signal to BCU interface 'store' line.
- Receive 'BCU data request' signal; gate CSW to SBI lines.
- Receive 'accept' signal.
- Turn on 'accept latch' and turn off 'interrupt' latch.
- Turn off 'interrupt storage request' latch.
- Receive 'BCU advance pulse' signal.
- Turn off 'accept' latch.
- Turn off 'storage cycle' trigger.
- Turn off 'remember interrupt' latch.

- Send CC = 1 and 'release' signal to CPU interface.
- 'Test I/O' and 'select channel' signals drop.
- Turn off 'test I/O' and 'pseudo accept' latches.
- Channel resumes polling; test I/O operation complete.

The test I/O, CSW store operation (Diagram 5-17) is performed to store the CSW in main storage location 64 (decimal). This operation is performed if, during the channel's attempt to select the I/O device specified by the test I/O unit address, any of the following occurs: (1) a channel control check condition is detected with the unit address in the 'bus out' latches ("Test I/O, Channel Control Check"); (2) a control unit busy condition is detected during the device selection attempt ("Test I/O, Device Response"); (3) an interface control check condition is detected when the addresses in the unit address register and 'bus-in' latches are compared ("Test I/O, Device Response" and "Test I/O, Interface Control Check"); and (4) when the correct I/O device is selected, but the ending status byte does not equal all 0's ("Test I/O, Correct Device Responded").

The test I/O CSW store operations begin when the 'interrupt' latch is turned on. This latch may be turned on as a result of any of the conditions previously described. (Refer to ALD's EN113 and EN115.) With the 'interrupt' signal active, the 'sequence 5' latch is turned off and, if on, the 'polling interrupt' latch is turned off. At the same time, the 'interrupt' signal is AND'ed with the 'not sequence 3 or 4' signal to turn off the clock. The 'interrupt' signal also activates the 'remember interrupt' signal. If the test I/O, CSW store operation was initiated by a control unit busy sequence, the 'control unit busy' latch is turned off by the 'remember interrupt' signal.

To simulate receipt of an 'interrupt response' signal from the CPU interface, the 'pseudo accept interrupt' latch is turned on by AND'ing the 'interrupt', 'test I/O latch', and 'not interface reset' signals. In turn, the 'pseudo accept interrupt' signal turns on the 'interrupt storage request' latch; this activates the 'storage request' signal at the BCU interface to request access to main storage. In addition, the 'interrupt storage request' signal turns on the 'Z-address' latch and the 'storage cycle' trigger. The 'Z-address' signal conditions the SAB gating logic for address 64 decimal, while the 'storage cycle' signal activates and inhibits applicable channel logic for the storage operation.

The channel then waits for a 'BCU response' signal from the BCU interface, indicating that the 'storage request' signal has been honored. When received, the 'BCU response' signal (1) turns on the 'latch Z-address' latch, causing the 'Z-address latch gated' signal to activate; (2) turns off the 'suppress out' latch, if on, as a result of a test I/O, interface control check operation; and (3) gates the appropriate address information and 'store' signal to the BCU interface. To gate the address information and 'store' signal, the 'BCU response' signal: (1) AND's with the 'Z-address latch' signal to gate address 64 (decimal) to the SAB; (2) AND's with the 'Z-address latch gated' signal to gate all mark bits to the mark lines; (3) AND's with the 'remember interrupt' signal to force the storage protect parity bit to the storage protection parity line and activate the 'store' signal.

The CSW status bytes are gated to the SBI gating logic by AND'ing the 'Z-address gated' and 'remember interrupt' signals. When the channel receives the 'BCU data request' signal (or in the case of the Model 91, generates the signal), the CSW status bytes are gated to the SBI lines for storage in main storage, and the 'storage request' signal to the BCU interface is degated.

Note: With all mark bits active, the CSW status bytes are stored in their respective storage byte locations and 0's are stored in the other CSW byte locations.

The channel then waits for an 'accept' signal from the BCU interface. When received, the 'accept' signal turns on the 'accept' latch and is AND'ed with the 'Z-address latch' signal to turn off the 'interrupt' latch. With the 'accept' latch on, the 'interrupt storage request' latch is turned off.

The channel then waits for the 'BCU advance pulse' signal, which indicates that the CSW has been stored. When received, the signal resets the 'accept' latch and is delayed to activate the 'late advance pulse' signal. The 'late advance pulse' signal turns off the 'storage cycle' trigger. In turn, the 'not storage cycle' signal turns off the 'remember interrupt' latch. With the 'remember interrupt' latch off, the channel activates condition code 1 to the CPU interface; condition code 1 indicates that the test I/O operation resulted in storage of the CSW. After a deskewing delay, the 'release' signal is gated to the CPU interface. This causes the 'test I/O' and 'select channel' signals to the channel to drop. With the 'test I/O' signal inactive, the 'test I/O' and 'pseudo accept' latches turn off. Subsequently the 'storage cycle complete' signal, (activated by delaying the 'BCU advance pulse' signal) turns off the 'latch Z-address' latch. With the 'latch Z-address' signal inactive, the 'Z-address' latch turns off. The channel then resumes polling operations and is free to perform another operation. With the completion of the test I/O, CSW store operation, channel test I/O operations are completed.

TEST CHANNEL INSTRUCTION

The Test Channel instruction is initiated by the CPU to test the operational state of a specific channel; the instruction does not change the operating status of the channel.

Test channel operations are described at both the simplified and detailed levels in the following text.

Simplified Test Channel Operations

- 'Channel available' signal inactive causes CPU to internally generate condition code 3 and end test channel operation.
- 'Unit address bus out' lines ignored on Test Channel instruction.
- 'Channel available' signal causes channel to receive 'test channel' and 'select channel' signals.
- Interrupt pending condition in channel causes CC = 1 and 'release' signal to be sent to CPU interface.
- Channel busy (working) condition in channel causes CC = 2 and 'release' signal to be sent to CPU interface.
- Channel not busy (available) condition in channel causes CC = 0 and 'release' signal to be sent to CPU interface.

When the CPU decodes a Test Channel instruction (Figure 3-5), the CPU examines the 'channel available' signal from the channel to be tested. If the signal is not active, the CPU internally generates condition code 3 and the 'release' signal to end the operation. In this case, the channel is not available to receive other instructions from the CPU.

If the 'channel available' signal is active, the channel receives a 'test channel' signal and a 'select channel' signal via the CPU interface. The channel does not examine the 'unit address bus out' lines, since only the channel is being tested. These signals examine the channel for an interrupt pending condition. An interrupt pending condition causes the channel to send condition code 1 and the 'release' signal to the CPU interface. The interrupt pending condition is not cleared and the test channel operation ends with the fall of the 'test channel' and 'select channel' signals to the channel via the CPU interface.

If no interrupt condition is pending, the 'test channel' signal examines the channel for a channel busy (channel working) condition. If the channel is engaged in an operation, the channel sends condition code 2 and the 'release' signal to the CPU interface. Channel operations in progress are not altered by the Test Channel instruction. The test channel operation ends when the 'test channel' and 'select channel' signals to the CPU interface drop.

If the channel is not busy (available to perform other operations), the channel sends condtion code 0 and the 'release' signal to the CPU. The test channel operation ends when the 'test channel' and 'select channel' signals to the CPU interface drop.

From the time the channel receives the Test Channel instruction, one of the four condition codes and the 'release' signal previously described are sent to the CPU interface within 0.5 usec. A summary of test channel condition codes and their significance follows:

Condition Code	Significance
0	Channel available for other operation.
1	Interrupt condition pending in channel.
2	Channel busy performing operation.
3	Channel not available for any operation.

Detailed Test Channel Operations

- 'Channel available' signal inactive if:
  1. Channel is in test mode.
  2. Channel is meter disabled.
  3. Channel is not attached to CPU.
  4. Channel power is off.
- 'Not channel available' signal causes CPU to internally generate condition code 3 and end test channel operation.
- 'Channel available' signal permits 'test channel' and 'select channel' signals to be sent to channel.
- 'Program controlled interrupt' or 'interrupt' latches on cause CC = 1 to be sent to CPU interface.
- Following channel conditions (if present) turn on 'code 2' latch (CC = 2 sent to CPU interface):
  1. 'Chain command' latch on.
  2. 'Sequence 5' latch on.
  3. 'Operational in' signal active.
  4. 'Status in' signal active.
- Absence of CC = 1 or CC = 2 conditions activates CC = 0 and internal 'accept' signal.
- If CC = 1 or CC = 2 activates '100-ns singleshot' signal:
  1. 'Release' latch turns on.
  2. '100-ns singleshot' signal falls.
  3. 'Release' signal activated to CPU interface.
- If CC = 0 detected, 'accept' signal activates 'release' signal to CPU interface.
- 'Test channel' and 'select channel' signals drop.
- 'Release' latch turns off (if on) or 'accept' signal falls.

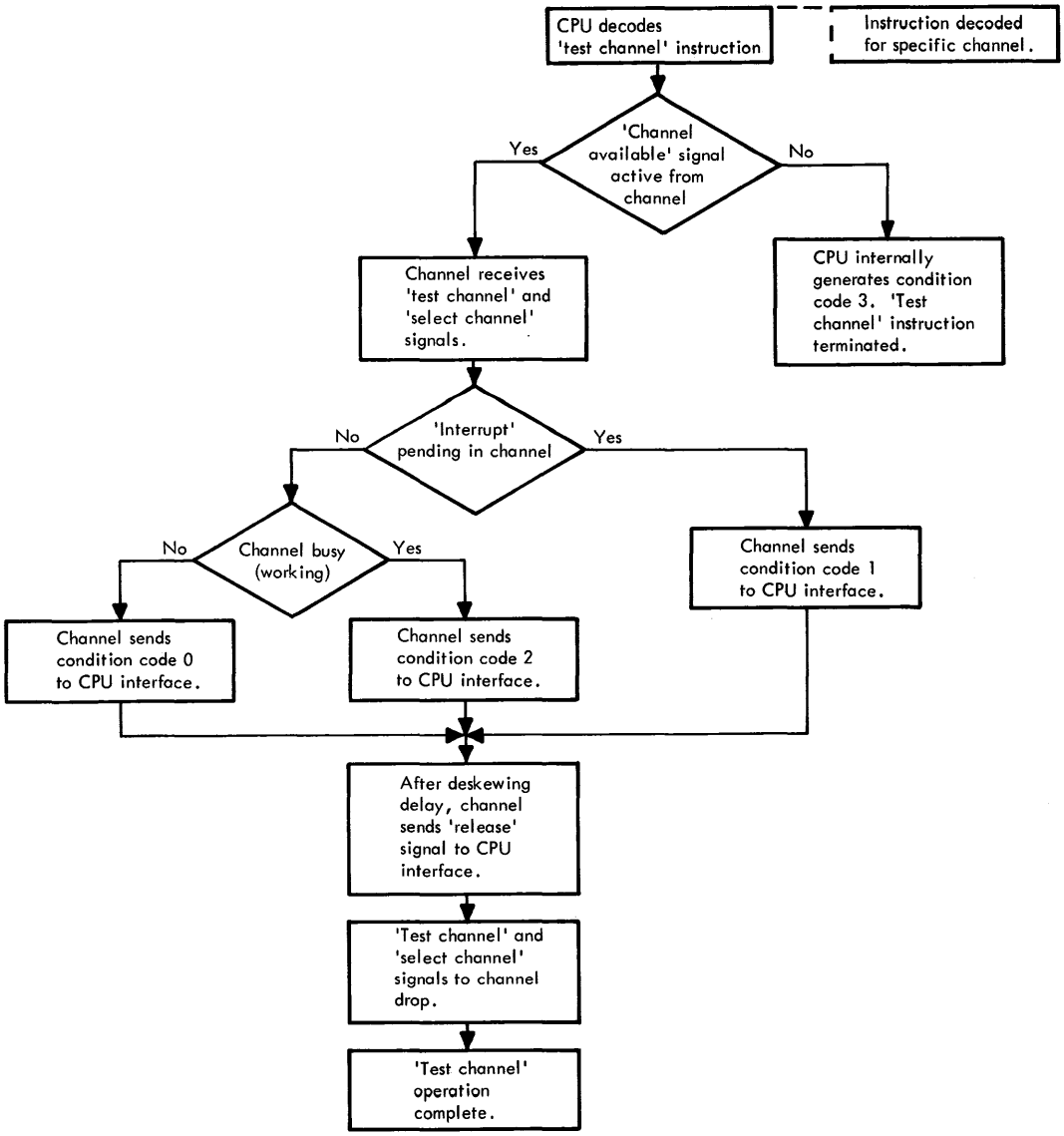


Figure 3-5. Test Channel Operation, Simplified Flow Chart

- 'Release' signal to CPU interface deactivated.
- Condition code signals deactivate.

Detailed test channel operations described in the following text are based upon Diagram 5-18.

When the CPU decodes the Test Channel instruction for a specific channel, the CPU examines the 'channel available' signal from the specified channel. If the signal is inactive the CPU internally generates condition code 3 and the 'release' signal to end the Test Channel instruction. The 'channel available' signal is inactive when: (1) the channel is in the test mode ('channel simulate CPU' signal is active); (2) the channel is meter disabled; (3) the channel is not attached to the CPU; or (4) channel power is off. If the specified 'channel available' signal is active when the Test Channel instruction is decoded, the channel receives a 'select channel' and 'test channel' signal via the CPU interface.

Note: No address is received on the 'unit address bus out' lines to the channel, since the Test Channel instruction does not involve selection of an I/O device. If the channel's 'program controlled interrupt' latch or 'interrupt' latch is on, the active 'interrupt available' signal is AND'ed with the 'test channel' signal to activate condition code 1 to the CPU interface. If the channel is busy (working), the 'test channel' signal is AND'ed with one of the following

channel signals: 'chain command latch'; 'sequence 5 latch'; 'operational in' (from an attached control unit); or 'status in' (from an attached control unit). If any of the four signals are active, the 'code 2' latch turns on to activate condition code 2 to the CPU interface.

If the channel does not contain a pending interrupt condition or is not busy, the 'test channel not simulate'; 'not interrupt available' and 'not path working' signals activate the channel's internal 'accept' signal. In this case, neither of the two condition code lines to the CPU interface is activated, and condition code 0 is present at the CPU interface.

The 'release' signal to the CPU interface is generated by one of two means, depending upon the condition code activated (Diagram 5-18). If condition code 1 or 2 is activated, the '100-ns release' singleshot is triggered when the condition code is activated. The '100-ns release' signal turns on the 'release' latch. When the '100-ns release' signal falls, the 'release' signal is gated to the CPU interface. If condition code 0 is sent to the CPU interface, the internal 'accept' signal activates the 'release' signal to the CPU interface.

With the 'release' signal active at the CPU interface, the 'select channel' and 'test channel' signals to the channel drop. If the 'release' latch is on, the 'not select channel' signal turns off the latch to deactivate the 'release' signal to the CPU interface. If the internal 'accept' signal is active, the 'not test channel' signal deactivates the 'accept' signal causing the 'release' signal to deactivate. Subsequently, the code 1 signal (if active) is deactivated by the 'not test channel' signal. If the 'code 2' latch is on, the latch is turned off by the 'not select channel' signal. This ends the test channel operation. Note that channel operations or conditions were not altered by the test channel operation; i.e., only the condition of the channel was monitored and relayed to the CPU interface by the appropriate condition code.

INITIAL PROGRAM LOAD

The following paragraphs introduce the IPL operation, and describe channel IPL operations at both simplified and detailed levels. For the simplified and detailed channel IPL operations, references are made to positive logic diagrams, flow charts, and timing charts contained in the FEMDM for the 2860 Selector Channel.

IPL Introduction

- IPL operations are controlled by channel to load programs into main storage.
- IPL operation is divided into two routines: IPL loader routine, and program load routine.
- IPL loader routine is standard, controlled by channel hardware, and stores IPL PSW, IPL CCW 1, and IPL CCW 2 at address locations 0, 8 and 16.
- Program load routine is flexible (after IPL CCW 1 is fetched) and may or may not employ command chaining and TIC operations to load a portion or all of the desired program.
- Channel does not release CPU until the program load routine is complete and unit and channel addresses have been stored at address 0 (byte locations 2 and 3).

The IPL operation is controlled by the channel to load a program or set of programs from an external I/O device (such as a card reader or tape) to main storage. The IPL operation is manually initiated at the CPU console. An IPL operation can be considered as consisting of two routines. The first routine, referred to as the IPL loader, is standard for all IPL operations as far as channel operation and main storage allocations are concerned. The second routine of the IPL operation, the loading of the actual program, is flexible in regard to channel operation and storage allocations; the flexibility of the second routine of the IPL operation allows the programmer to designate storage area allocation, as well as the means by which the entire program is to be loaded.

IPL loader information consists of three doublewords designated 'IPL PSW', 'IPL CCW 1' and 'IPL CCW 2' which are stored at main storage addresses 0, 8, and 16, respectively. When the channel is instructed by the CPU to perform an IPL operation, the channel forces CCW information (read chain command 24 bytes) into its control logic and registers specifying that the IPL loader doublewords are to be stored at the address indicated. (See Table 3-1.) Following storage of the IPL loader doublewords, the channel fetches the IPL CCW 1 doubleword from storage location 8 and controls the transfer of command and program information to the main storage locations designated by IPL CCW 1.

After all program information specified by IPL CCW 1 has been transferred to storage, further operations performed by the channel are dependent upon whether the chain command (CC) flag is active in IPL CCW 1, and if active, the command specified by IPL CCW 2.

If the CC flag is inactive in IPL CCW 1, as may be the case when loading an IPL program from tape, the channel enters an IPL ending operation. During the IPL ending operation, the channel stores the channel address and the address of the I/O device involved in the IPL operation in byte locations 2 and 3 of the IPL PSW (at storage

Table 3-1. IPL Routine and Variations

Byte Location, Base 10	Command	Flag	Comments
Channel IPL logic	Read 24 bytes	CC	Initial operations; channel sets command address register to address location 8, data address register to address location 0, and controls storing of three doublewords (IPL loader) at addresses 0, 8, and 16.
0	IPL PSW		Used by the CPU after IPL operations are complete and 'release' signal is received.
8	Read X bytes (IPL CCW 1)	CC (If required)	Fetched by channel after IPL loader is stored. Specifies start storing commands and program data at location Y.
16	TIC or read X bytes (IPL CCW 2)		For TIC, transfer to location Y to obtain new CCW from chain of CCWs. For read command, store program data at location Y.
Y	As programmed	As programmed	Contains additional commands as required for program and contains CPU program.

Note: "X" refers to the byte count of the program to be loaded. "Y" refers to the storage location selected by the programmer to place his CPU program into the main storage.

location 0). Subsequently, the channel sends a 'release' signal to the CPU to end the IPL operation.

If the CC flag is active in IPL CCW 1, as is usually the case when loading an IPL program from a card reader, the channel performs a chain command operation to fetch IPL CCW 2 from storage location 16. If IPL CCW 2 contains a TIC command (usually required when loading an IPL program from a card reader) the channel fetches a new CCW from the command information loaded as a result of IPL CCW 1 read operations; in this case, the address of the new CCW is obtained from IPL CCW 2. The channel then continues to transfer program information to storage and continues to fetch CCW's until a CCW is obtained without an active CC flag, CDA flag, and TIC command. After the program information from this CCW is stored, the channel enters the IPL ending sequence to store the channel and unit addresses and release the CPU.

If IPL CCW 2 is fetched by the channel and specifies a read command, the channel controls the transfer of program information to the storage address specified by IPL CCW 2.

Note: This operation is possible, but is usually not programmed. In this case, further chaining is not possible, since storage locations adjacent to address location 16 (the location of IPL CCW 2) are reserved for CPU control information rather than CCW's. (Recall that, when chaining, the channel fetches CCW's from sequential locations, except when a TIC command is indicated.) After the program data specified by IPL CCW 2 has been read to storage, the channel enters the IPL ending sequence to store the channel and unit addresses and release the CPU.

Note that in all cases described above, the channel does not send a 'release' signal to the CPU until a CCW is obtained by the channel which does not specify command chaining, data chaining, or transfer in channel operations, and until the read operation specified by that CCW has been completed.

Figure 3-6 illustrates the IPL operation from both the channel and system viewpoints. Prior to initiating the IPL operation, the I/O device from which the program is to be read is loaded and made ready; for example, the tape containing the program is loaded on the tape unit or cards containing the program are loaded in the card reader, and the device is readied for on-line operation. At the system console, the UNIT ADDRESS and CHANNEL ADDRESS switches are positioned for the address of the I/O device from which the program is to be read. The IPL operation is then started by pressing the LOAD pushbutton on the system console. When the LOAD pushbutton is pressed, the LOAD indicators on the system console and channel CE panel light and remain lighted until the IPL operation is complete.

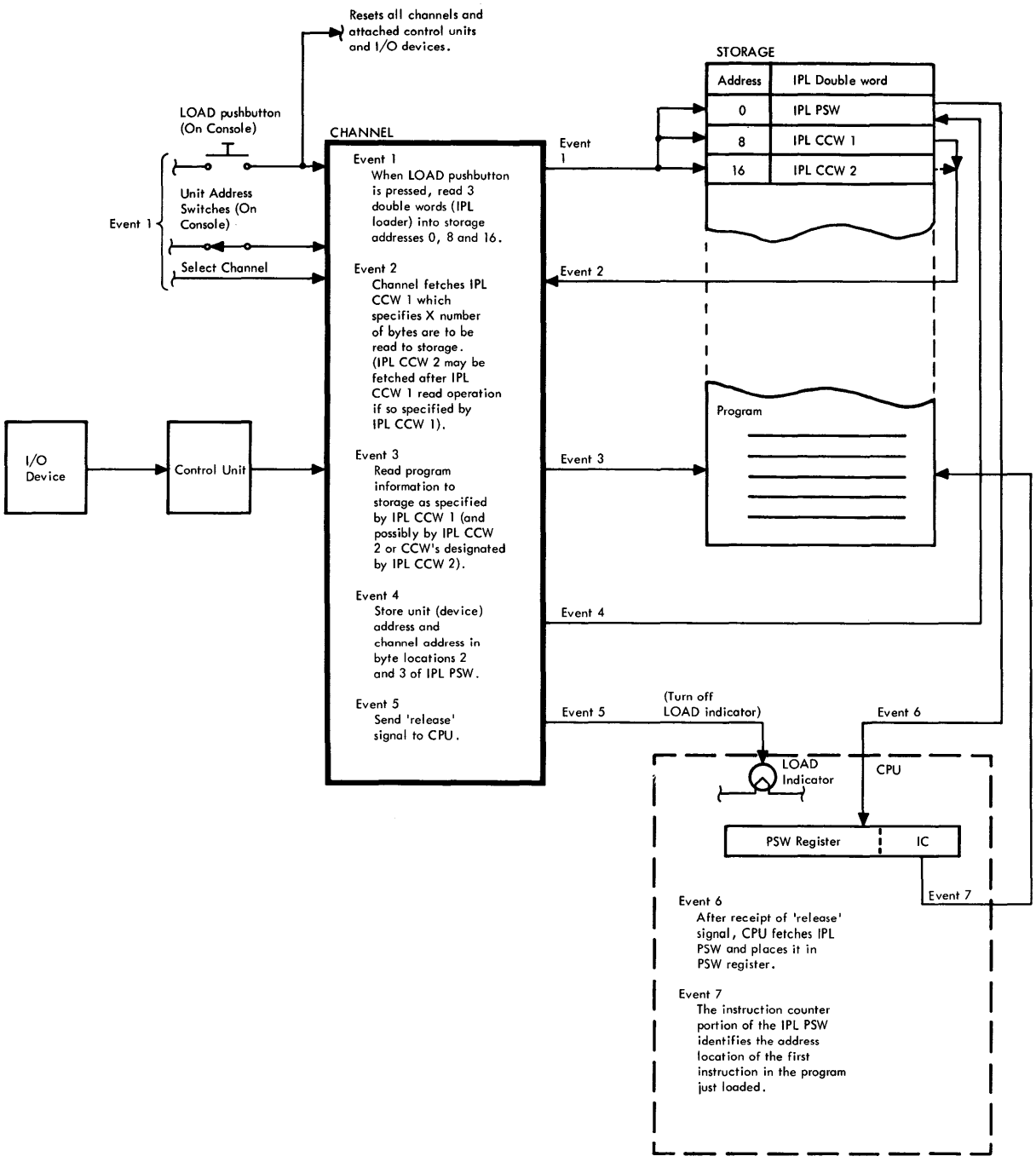


Figure 3-6. IPL Operations, Simplified Block Diagram

When the LOAD pushbutton is pressed (Event 1 in Figure 3-6), all channels in the system receive an IPL pulse which causes all on-line channels and their associated control units and I/O devices to be reset. The channel that is to control the IPL operation also receives a 'select channel' signal. After the selected channel and its control units and I/O devices are reset, the channel selects the I/O device specified by the UNIT ADDRESS switches and forces CCW information into its control logic and registers. To assemble the IPL loader CCW, the channel forces a read command into the command register, a count of 24 into the count register, address 0 in the data address register, address 8 in the command address register, the CC flag on in the flag register, and turns on the 'CCW valid' latch.

Note: Command chaining is forced, rather than data chaining, to insure that IPL operations may be performed with high-speed devices on all models of the System/360.

The channel forced CCW (read 24 bytes) causes the channel to control the transfer of the IPL PSW, IPL CCW 1 and IPL CCW 2 doublewords from the I/O device to main storage locations 0, 8 and 16, respectively (Event 1, Figure 3-6).

After the three doublewords are in storage, the channel performs a chain command operation and fetches IPL CCW 1 from address location 8 (Event 2, Figure 3-6). With IPL CCW 1 in the channel, the channel controls the transfer of program doublewords to the main storage locations specified by IPL CCW 1 (Event 3, Figure 3-6). When all program information specified by IPL CCW 1 has been received from the control unit, the channel examines the CC flag to determine if further program information is to be loaded during IPL operation. If the CC flag is active, the channel performs a chain command operation and fetches IPL CCW 2 from main storage location 16. Normally, IPL CCW 2 will specify a TIC command, causing the channel to fetch a CCW from a main storage location loaded as the result of IPL CCW 1 read operations. If a new CCW is fetched as a result of a TIC operation specified by IPL CCW 2, the channel continues read and chain command operations to control the transfer of program information to storage until a read command CCW with inactive CC and CDA flags is obtained from storage (Events 2 and 3, Figure 3-6).

When the read operations specified by the first read command CCW with inactive CC and CDA flags (this could be IPL CCW 1, IPL CCW 2, or an indeterminate numbered CCW, depending upon the program and the device from which the program is loaded) are complete, the channel stores its address and the address of the I/O device (unit address) in the interruption code portion (byte locations 2 and 3) of the IPL PSW at main storage location 0 (Event 4, Figure 3-6).

Following storage of the channel and unit address, the channel sends a 'release' signal to the CPU indicating that the IPL operation is complete. (Event 5, Figure 3-6). Upon generation of the 'release' signal, the load indicators on the system console and channel CE panel are extinguished. The CPU, after receiving the 'release' signal, fetches the IPL PSW from main storage location 0 and gates the doubleword into its PSW register (Event 6, Figure 3-6). With the instruction counter (IC) portion of the PSW register identifying the address location of the first instruction in the program just loaded (Event 7, Figure 3-6), the CPU may begin processing operations. It is possible (assuming that the entire program was not loaded during the IPL operation) that the first instruction obtained by the CPU may be a Start I/O instruction; in this case, the CPU would initiate a start I/O operation at the channel which just completed the IPL operation to cause the remainder of the program to be read into main storage. While the remainder of the program was being stored, the CPU could begin processing operations with that portion of the program loaded during the IPL operation.

If, during the IPL operation, a check condition occurs or an I/O device selection is unsuccessful, channel operations stop, the CPU remains in a wait state, and the load indicators on the system console and channel CE panel remain lighted. In this case, operator intervention is required to either restart the IPL operation in the manner previously described or determine and correct the cause of the check condition, if necessary.

Basic IPL Channel Operations

- Start IPL requires unit address, 'IPL' signal and 'select channel' signal at channel.
- During IPL initial selection, channel:
  1. Resets control unit, I/O devices and channel.
  2. Forces read 24 bytes, data address 0, CC flag, command address 8 CCW into channel.
  3. Waits 100 ms after the interface reset before starting the I/O device selection.
  4. Selects I/O device specified by unit address.
  5. Clears initial status.
- During IPL loader read operation, channel controls transfer of IPL PSW, IPL CCW 1 and IPL CCW 2 doublewords to main storage addresses 0, 8 and 16.
- For IPL chain command to IPL CCW 1 routine, channel disconnects and reselects I/O device and fetches IPL CCW 1 from storage location 8.
- IPL CCW 1 read operations are performed by channel as for a normal read operation.
- IPL CCW 2 is fetched if IPL CCW 1 contains an active CC Flag.

- IPL CCW 2 usually contains a TIC command causing the channel to fetch another CCW stored during the IPL CCW 1 read operation.

- Channel enters IPL ending condition operation after read operation for first CCW without CC of CDA flags and without TIC command.

- During IPL ending condition operation, channel stores unit and channel addresses (address 0, bytes 2 and 3), disconnects I/O device, and releases CPU.

The basic operations performed by the channel are introduced in flow chart form in Figure 3-7. Channel start IPL operations begin when the channel receives the unit address (eight bits plus P) on the 'unit address bus out' lines, the multiplex 'initial program load' signal, and the simplex 'select channel' signal.

The channel then enters the IPL initial selection routine during which the following operations are performed: (1) the control units and I/O devices attached to the channel are reset; (2) the channel control logic and registers are reset; (3) the channel forces the IPL loader CCW(read 24 bytes beginning at address 0 with the CC flag active and a command address of eight); (4) the channel waits 100 ms after the interface reset; and (5) the channel selects the I/O device specified by the unit address bits. After the I/O device is successfully selected, the channel clears the initial status condition presented during the device selection to complete the IPL initial selection routine.

With the IPL initial selection routine complete, the channel enters the IPL loader read operation. During this operation, the channel controls the transfer of the three IPL loader doublewords (IPL PSW, IPL CCW 1, and IPL CCW 2) from the control unit to main storage. To control the read transfers, the channel assembles each doubleword on a byte-by-byte basis, requests storage cycles to store each doubleword, and monitors the progress of the transfers in the same manner as for a normal read operation.

When the last of the three doublewords is stored, the channel enters the IPL chain command to CCW 1 routine. During this routine the channel disconnects the control unit from the I/O interface, fetches IPL CCW 1 from main storage location 8 and enters it into the appropriate channel registers, reselects the I/O device, and clears the status presented by the control unit during the reselection operation.

With a valid read command CCW in the channel and the I/O device reselected, the channel begins the IPL CCW 1 read operation. During this operation, the channel controls the transfer of command and program doublewords to main storage as for a normal read operation. The amount of command and program data transferred and the storage area to which the data is transferred is determined by the control information contained in IPL CCW 1.

When all program bytes have been received from the control unit, the channel examines the CC flag in the flag register to determine if command chaining is required. If so, the channel disconnects the control unit from the I/O interface, fetches IPL CCW 2 from storage location 16, and reselects the I/O device. When IPL CCW 2 is available at the channel, the channel examines the command bits for a TIC command. If a TIC command is present, the channel fetches another CCW from a main storage address specified by the data address bits in IPL CCW 2. The data address specifies the address location of a CCW stored in main storage during the IPL CCW 1 read operation. If a TIC operation is performed, the channel continues to control the reading of program information to storage, and continues to fetch new CCW's until a CCW without a CC flag, a CDA flag, and a TIC command is obtained.

The channel enters the IPL ending condition operation when a CCW is obtained that does not specify command chaining, data chaining, or a TIC command, and when the read operation specified by this CCW is complete. During the IPL ending condition routine, the channel performs an I/O interface disconnect operation, stores the unit and channel addresses in byte locations 2 and 3 of address 0 (IPL PSW), and generates a 'release' signal to the CPU. With the generation of the 'release' signal, the IPL operation is complete and the channel is free to resume polling operations or receive further I/O instructions.

Simplified IPL Channel Operations

The following paragraphs present an expanded description of the basic IPL channel operations introduced in the preceding paragraph ("Basic IPL Channel Operations"). The descriptions are based upon the simplified flow chart (Diagram 5-19) in the FEDM.

IPL Initial Selection

- IPL initial selection operations are started when 'IPL' and 'select channel' signals are received by channel.

- Channel timeout clock is blocked and interrupt recognition is inhibited.

- I/O interface and channel reset operations are performed.

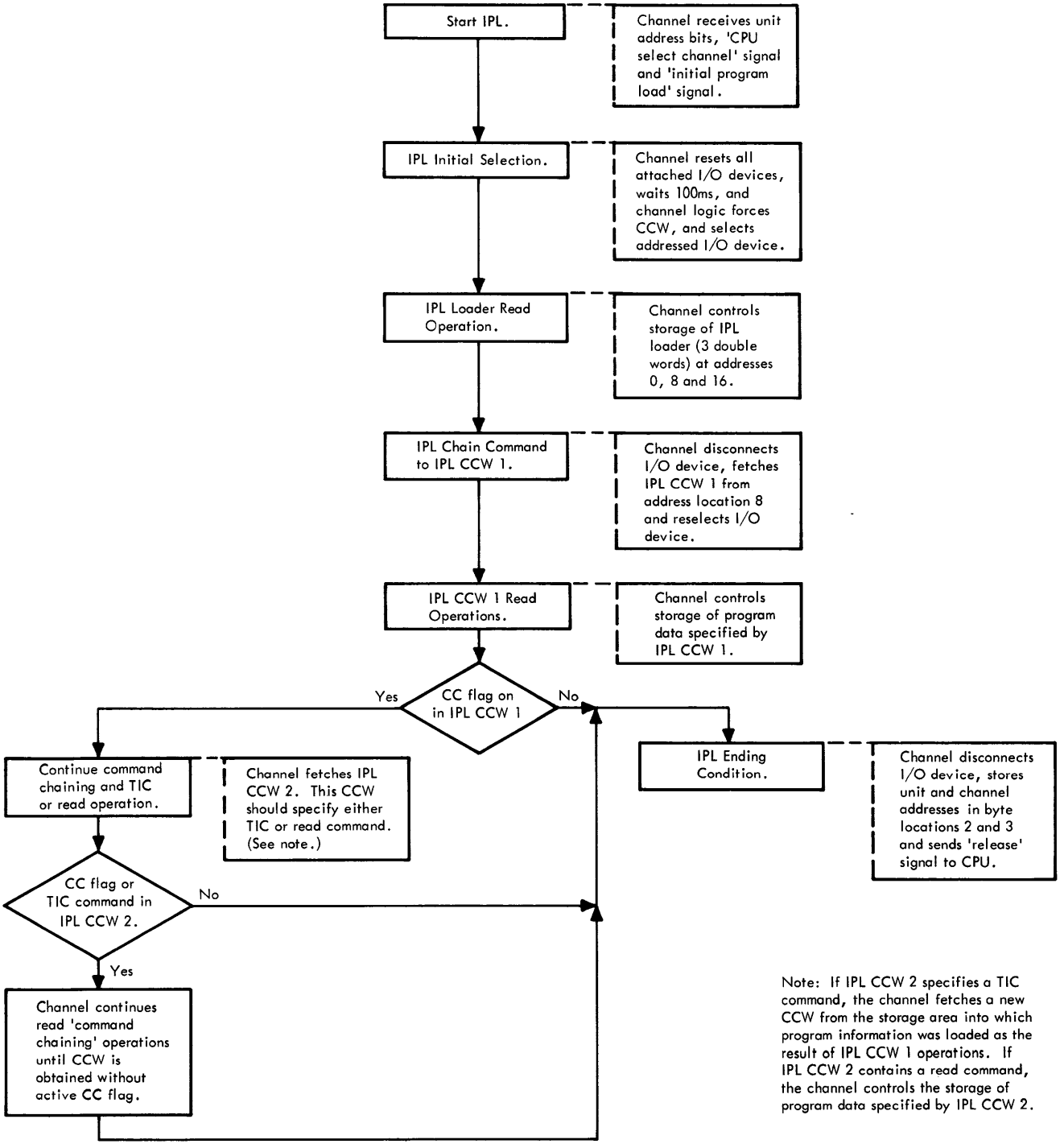


Figure 3-7. IPL Basic Operations, Flow Chart

- After 100 ms, channel forces CCW:
  1. Read (bit 6) to command register.
  2. Address 8 (bits 20) to command address register.
  3. CC flag to flag register.
  4. Address 0 to data address register.
  5. Count of 24, bits 19 and 20 to count register.
  6. 0's to storage protect register.
  7. CCW valid latch on.

- Select I/O device:
  1. Turn on 'setup' latch.
  2. Gate UABO bits to unit address register.
  3. Gate unit address register contents to control unit.
  4. Receive correct unit address from control unit.
  5. Send read command to control unit.
  6. Receive status byte of all 0's from control unit.
  7. Clear status byte from control unit.

Channel IPL initial selection operations are initiated when the channel receives the initial program load and select channel pulses (Diagram 5-19). Upon receipt of these signals, the channel initiates three parallel operations: (1) blocks the channel timeout clock and inhibits channel recognition of interrupts; (2) initiates a channel and I/O interface reset operation; and (3) triggers a 100 ms delay singleshot.

Blocking of the channel timeout clock is necessary, since for other than IPL or FLT operations, the channel timeout circuits detect the absence of a 'release' signal to the CPU approximately 100 ms after the channel has received an instruction. When this condition is detected at the channel, the channel and I/O interface are reset. For the IPL operation, the 'release' signal is not generated within the 100-ms period; thus, the timeout clock is blocked to prevent interference with the IPL operation. Channel detection of an interrupt condition is blocked to prevent interference with the IPL operation due to receipt of an interrupt condition during operation.

The I/O interface reset operation is a 6-usec operation performed to cause a reset operation in all on-line control units and I/O devices attached to the channel. Coincident with the I/O interface reset operation, a 6-usec 'machine reset' signal is generated to reset all channel control logic and registers. The '100-ms singleshot' signal prevents the channel from proceeding with IPL operations and thus insures sufficient time to reset all attached control units and I/O devices.

At the end of the 100-ms period, the channel forces a read 24 bytes into storage locations 0 through 23 CCW. The channel forces the CCW by: (1) activating the CC flag in the flag register; (2) forcing a count of 24 into the count register (activating bits 19 and 20); (3) forcing an address of eight in the command address register (activating bit 20 and deactivating the parity bit); (4) forcing a read command into the command register (activating bit 6); and (5) turning on the 'CCW valid' latch. The data address register and flag registers were reset to 0 with odd parity by the 'machine reset' signal.

While forcing the CCW, the channel gates the unit address from the 'unit address bus out' lines into the channel's unit address register and turns on the 'setup' latch to begin the I/O device selection operation.

During the I/O device selection operation, the channel gates the unit address register contents to the control units on the 'bus out' lines and raises the 'address out' and 'select out' signals. The control unit to which the addressed I/O device is connected responds by raising the 'operational in' signal. This causes the channel to drop the 'address out' signal. The channel then compares the unit address (received when the control unit activates the 'address in' signal) on the 'bus in' lines with the address in its unit address register to ensure that the proper device was selected.

Assuming that the unit addresses compare, the channel sends the 'read command' byte to the control unit (Diagram 5-19) and raises the 'command out' signal. When the control unit drops the 'address in' signal, the channel deactivates the 'command out' signal. The control unit then raises its 'status in' signal. With the 'status in' signal active, the channel examines the status byte to determine if the control unit and I/O device are prepared to execute the read command. If all bits of the status byte are 0's (control unit and I/O device ready to perform the command), the channel raises its 'service out' signal, causing the control unit to remove the status byte from the 'bus in' lines and to drop the 'status in' signal. With the drop of the 'status in' signal, the channel drops its 'service out' signal to complete the IPL initial selection operation. The channel, control unit, and I/O device are now prepared to begin the IPL loader read operations.

IPL Loader Read Operations

- IPL PSW (eight bytes) are loaded into channel B-register on byte-by-byte basis due to exchange of 'service in' and 'service out' signals between control unit and channel.

- IPL PSW transferred to A-register; 'sequence 3' latch turned on.

- IPL PSW stored in main storage location 0.

- Count in count register decremented to value of 16.

- Data address register value incremented to value of eight.

- IPL CCW 1 assembled in B-register on byte-by-byte basis and transferred to A-register.

- IPL CCW 1 stored at main storage location 8.
- 'Last word' trigger turned on.
- Count in count register decremented to value of eight.
- Data address register value incremented to value of 16.
- IPL CCW 2 assembled in B-register.
- 'Sequence 5' latch turned on.
- Transfer IPL CCW 2 to A-register; turn on 'sequence 3' latch.
- Store IPL CCW 2 at main storage location 16.
- Data address register value incremented to value of 24.
- Stop command sent to control unit.

During the IPL loader read operation (Diagram 5-19), the channel controls the transfer of the three IPL loader doublewords from the control unit to main storage. Read operations begin when the control unit activates the 'service in' signal and places the first IPL byte on the 'bus in' lines. The channel responds by raising its 'service out' signal and gating the IPL byte into the channel's B-register. When the control unit receives the 'service out' signal, it removes the IPL byte from the 'bus in' lines and drops the 'service in' signal; this causes the channel to drop its 'service out' signal. The exchange of 'service in' and 'service out' signals continues until eight bytes have been gated into the eight B-register byte positions. When the B-register is full (IPL PSW doubleword assembled in the B-register), the channel transfers the IPL PSW doubleword to the A-register and turns on the 'sequence 3' latch.

At this point in the IPL loader read operation, three parallel operations are started to: (1) store the IPL PSW in main storage; (2) update the count in the count register; and (3) begin assembling the IPL CCW 1 doubleword in the B-register.

To store the IPL PSW doubleword, the channel raises the 'storage request' signal. When the 'BCU response' signal is received, the channel gates the data address register value (address 0) to the SAB, the mark-A register bits (nine active bits) to the mark bus, the storage protect register value (0's with active parity bit) to the 'storage protection' lines, and the 'store' signal to the BCU interface. Subsequently, the 'BCU data request' signal is received by the channel (or in the case of the System/360, Model 91, is generated in the channel 150 ns after receipt of the 'BCU response' signal). The 'BCU data request' signal gates the IPL PSW from the A-register to the SBO lines. When the channel receives a 'BCU advance pulse' signal from the BCU interface, the signal is delayed to generate the 'storage cycle complete' signal. With the generation of this signal the store IPL PSW operation is complete.

While the store IPL PSW operation is in progress, the channel performs an update count operation while the 'sequence 3' latch is on (Diagram 5-19). During the update count operation the channel gates the count register value (24 decimal) minus eight to the adder, latches the adder, and gates the decremented value (16 decimal) back into the count register.

Following the update count operation, the channel enters a 'sequence 4' routine to update the address value in the data address register. During the data address update operation, the channel gates the data address register value (0) plus eight to the adder, latches the adder, and gates the incremented value (eight) back into the data address register.

Meanwhile, the IPL CCW 1 doubleword is being assemblied in the B-register. When the last byte of the doubleword is in the B-register, the IPL CCW 1 doubleword is transferred to the A-register (assuming the store IPL PSW operation is complete).

With the IPL CCW 1 doubleword in the A-register, the channel requests a storage cycle to store the doubleword at main storage address location eight. Channel operations for the store IPL CCW 1 operation are identical to those described for the store IPL PSW operation.

While the store IPL CCW 1 operation is in progress, the channel turns on the 'last word' trigger and performs an update count operation. At the end of the update count operation, the count register contains a new count value of eight. Subsequent to the update count operation, the channel performs a data address update operation to increment the data address register value by eight bytes. At the completion of the data address update operation, the data address register contains an address value of 16.

While the store IPL CCW 1, update count and data address update operations are being performed, the channel is assembling the IPL CCW 2 doubleword into the B-register. When the last byte is received from the control unit, the channel transfers the IPL CCW 2 doubleword to the A-register and turns on the 'sequence 5' latch. With IPL CCW 2 in the A-register, the channel requests a storage cycle to store IPL CCW 2 at main storage address location 16. Channel operations for the store IPL CCW 2 operation are as described for the store IPL PSW operation.

While the store IPL CCW 2 operation is in progress, the channel: (1) performs a data address update operation; and (2) sends a stop command byte to the control unit on the 'bus out' lines if another 'service in' signal is received. Since the chain command flag is active the data address update operation is significant only if the IPL operation is stopped due to a check condition or reselection failure; in this case, the updated data address could serve as an indication of the progress of the IPL operation.

The Stop command (all 0's with the parity bit active) is presented to the control unit in the event the control unit attempts to present another data byte to the channel. Disposition of the Stop command is described in the following paragraph ("IPL Chain Command to IPL CCW 1").

With the store IPL CCW 2, data address update, and stop command operations complete, the channel IPL loader read operations are complete; the channel is now prepared to perform the IPL chain command to IPL CCW 1 operations.

IPL Chain Command to IPL CCW 1

- Channel raises 'command out' signal for stop command if control unit raises 'service in' signal.
- Status byte from control unit with device end bit active turns on 'chain command' latch.
- Channel activates 'service out' signal and drops 'select out' signal, causing control unit to drop 'status in' and 'operational in' signals (I/O interface disconnect).
- Channel reselects I/O device, using unit address still in unit address register.
- Channel fetches IPL CCW 1 from main storage location eight using address in command address register.
- When IPL CCW 1 is available, channel checks for program validity, enters CCW bits in register and turns on 'CCW valid' latch.
- Channel compares unit address from selected I/O device with address in unit address register for correct selection.
- Channel sends IPL CCW 1 read command byte to control unit.
- Channel receives status byte from control unit.
- Channel examines status byte for all 0's (I/O device and control unit ready to perform command.)
- Channel causes control unit to clear status byte.

During the IPL Chain Command to IPL CCW 1 operation the channel disconnects the control unit from the I/O interface, fetches IPL CCW 1 from storage location eight, reselects the I/O device, sends a read command to the control unit, and receives status from the control unit.

The IPL chain command to IPL CCW 1 operation (Diagram 5-19) begins following completion of the previous update data address operation. With the 'sequence 4' latch off, the channel examines the 'service in' line from the control unit. If the 'service in' signal is active (control unit has additional data byte on 'bus in' lines), the channel raises the 'command out' signal signifying that a Stop command is on the 'bus out' lines to the control unit. The control unit responds by dropping the 'service in' signal, causing the channel to drop the 'command out' signal. Subsequently, the control unit raises the 'status in' signal and places its status byte (with the device end bit active) on the 'bus in' lines. The channel detects the device end bit and turns on the 'chain command' latch.

With the turn-on of the 'chain command' latch and activation of the 'service out' signal, the channel performs an I/O interface disconnect operation by dropping the 'select out' signal. This causes the control unit to drop both the 'status in' and 'operational in' signals to complete the disconnect operation.

Following the I/O interface disconnect operation, the channel initiates an I/O device selection operation to reselect the I/O device and initiate a storage request for IPL CCW 1 (Diagram 5-19). During the I/O device selection operation the channel gates the unit address register contents to the control unit on the 'bus-out' lines and, subsequently, raises the 'address out' and 'select out' signals. The channel also sends a 'storage request' signal to the BCU interface, requesting a storage cycle to fetch IPL CCW 1 from main storage.

The control unit responds to the 'address out' and 'select out' signals by raising the 'operational in' signal. With the 'operational in' signal active, the channel removes the unit address byte from the 'bus out' lines and drops the 'address out' signal. With the 'address out' signal inactive, the control unit activates the 'address in' signal to the channel. This completes the I/O device selection operation.

While the I/O device selection operation is in progress, the channel is in the process of fetching IPL CCW 1 from main storage. (Recall that the 'storage request' signal was raised during the I/O device selection operation.) When the channel receives the 'BCU response' signal, it gates the command address register contents (address eight) to the SAB and the storage protect register contents (0's with odd parity) to the 'storage protection' lines. When the channel receives the 'BCU advance pulse' signal, it delays the signal until the IPL CCW 1 doubleword is on the SBO lines. When the CCW is on the SBO lines, the channel checks the CCW for program validity and gates the CCW bits to the appropriate channel registers. The CCW is valid if a TIC command is not indicated, SBO bits 38 through 39 are all 0's and SBO bits 48 through 63 (count bits) are other than all 0's. Assuming the CCW is valid, the 'CCW valid' latch is turned on.

CCW bits on the SBO lines are gated into the channel registers as follows: (1) SBO bits 0 through 7 (+P) to the command register; (2) SBO bits 8 through 37 (+3P) to the

data address register; (3) SBO bits 32 through 36 (+P) to the flag register; and (4) SBO bits 48 through 63 (+2P) to the count register.

During the fetch IPL CCW 1 operation, the channel gates the address value (eight) in the command address register plus eight to the adder, latches the adder, and gates the incremented value (16 decimal) back into the command address register. With the command address updated and the 'CCW valid' latch on, the fetch IPL CCW 1 operations are complete.

Upon completion of both the I/O device selection and fetch IPL CCW 1 operations (Diagram 5-19), the channel continues the IPL chain command to IPL CCW 1 operation by comparing the unit address byte from the control unit with the unit address register contents. If the addresses compare (correct device has been selected), the channel gates the read command from the command register to the control unit on the 'bus out' lines. As the read command byte is being gated to the 'bus out' lines, the channel gates bits 21 through 23 (the DAB) from the data address register to the byte counter to establish the byte location of the first program byte from the control unit; i.e., the DAB, when entered into the byte counter, determines the byte location in the B-register into which the first program byte is to be gated. Simultaneously, the DAB plus the count value in the count register are gated to the adder. The adder is latched and the DAB plus count value is gated into the count register. The DAB plus count operation establishes the byte location in the B-register into which the last program byte specified by IPL CCW 1 is to be gated. Subsequently, the channel raises the 'command out' signal, informing the control unit that the command byte is available. With the 'command out' signal active, the control unit responds by activating the 'address in' signal; this signal causes the channel to drop the 'command out' signal.

The control unit responds by raising the 'status in' signal. This causes the channel to examine the status byte for an all 0's content. If all 0's are detected (control unit and I/O device ready to perform the read command), the channel turns off the 'chain command' latch and raises the 'service out' signal. This causes the control unit to remove the status byte from the 'bus in' lines and to drop the 'status in' signal. Subsequently, the channel drops the 'service out' signal. With the drop of the 'service out' signal, the IPL chain command to IPL CCW 1 operation is complete and the channel is prepared to control the read operation specified by IPL CCW 1.

IPL CCW 1 Read Operations

- Channel controls transfer of program data to main storage as during normal read operation.
- Count from IPL CCW 1 determines number of bytes read.
- Data address from IPL CCW 1 determines storage location at which program information is to be stored.
- IPL CCW 1 may or may not have active CC flag, depending upon program being loaded.

IPL CCW 1 read operations begin after the channel causes the control unit to remove the status byte and when the control unit presents the first program byte to the channel on the 'bus in' lines. Channel operations while controlling the transfer of IPL CCW 1 program data to main storage are as described for a normal read operation (see "Read Operation" in this chapter). The number of program bytes transferred during the read operation is determined by the count value received in IPL CCW 1 and varies from program to program. Likewise, the main storage starting address location at which program information is stored is determined by the data address received in IPL CCW 1.

When the last data byte for IPL CCW 1 has been gated into the channel, the 'sequence 5' latch is turned on and the CC flag in the flag register is examined to determine if a bootstrap or IPL ending condition operation is to be performed. If the CC flag is active, channel IPL operations continue (see "Bootstrap Operation"); if the CC flag is inactive, the channel performs the IPL ending condition operation (see "IPL Ending Condition").

Bootstrap Operations

- Bootstrap operation is performed if IPL CCW 1 CC flag is active and IPL CCW 2 contains TIC command.
- Channel performs chain command operation to:
  1. Disconnect control unit from I/O interface.
  2. Fetch IPL CCW 2 from storage location 16.
  3. Reselect I/O device.
- If IPL CCW 2 contains TIC command (usually the case); the channel fetches new CCW from main storage.
- New CCW is fetched from program data loaded as result of IPL CCW 1 read operations.
- Channel controls read operation specified by CCW and continues to fetch CCW's and control read operations until CCW without CC flag, CDA flag, or TIC command is obtained.

- After read operations for CCW without CC flag, CDA flag, or TIC command are complete, channel performs IPL ending condition operations.

Bootstrap operations (Diagram 5-19), are controlled by the channel to load program information utilizing CCW's stored in main storage as a result of the IPL CCW 1 read operations. For bootstrap operations to be performed, IPL CCW 1 must contain an active CC flag and IPL CCW 2 must contain a TIC command.

At the end of the IPL CCW 1 read operation, the channel performs a chain command operation to: (1) disconnect the control unit from the I/O interface; (2) fetch IPL CCW 2 from main storage address location 16; and (3) reselect the I/O device. Channel operations performed to accomplish the above three operations are described for the IPL chain command to IPL CCW 1 operation, with the exception that command address 16 (decimal) is gated to the SAB, rather than address eight.

When the 'IPL CCW 2' bits are available on the SBO, the channel detects the TIC command and gates the data address bits (SBO bits 8 through 31) into the data address register. With a TIC command indicated, the channel raises the 'storage request' signal. When the 'BCU response' signal is received, the channel gates the data address register contents to the SAB. The address in the data address register specifies the location in main storage of the first of a possible chain of CCW's. In addition, the channel gates the storage protect register contents (0's with the parity bit active) to the 'storage protection' lines. When the channel receives a 'BCU advance pulse' signal, it delays the signal until the new CCW is on the SBO lines. During the TIC operation, the data address register contents plus eight are gated to the adder. After the 'BCU advance pulse' signal is received, the channel latches the adder and gates the incremented address to the command address register; this address will be used to fetch the next CCW, if required. The channel then checks the CCW for program validity and gates the CCW bits to the appropriate channel registers. Channel operations for validity checking, gating the CCW to channel registers and further CCW setup operations are as described for the IPL chain command to IPL CCW 1 operations.

For the bootstrap operation, the channel will continue to control the reading of program information to storage and continue to fetch new CCW's until a CCW is obtained without an active CC flag, CDA flag, or TIC command. When the channel obtains this CCW, it controls the read operation specified by the CCW, then begins the IPL ending condition operation (see "IPL Ending Condition").

If the IPL CCW 1 contains a CC flag and IPL CCW 2 contains a read command (this is rarely the case), the channel fetches IPL CCW 2 from main storage and controls the reading of program data to main storage. (This operation is not termed a bootstrap operation.) At the completion of the IPL CCW 2 read operation, the channel performs the IPL ending condition operation.

IPL Ending Condition Operations

- Channel receives status byte and 'status in' signal with device end bit active.
- Channel activates 'service out' signal and drops 'select out' signal; control unit drops 'status in' and 'operational in' signals (I/O interface disconnect).
- Data address register reset to 0's with good parity.
- 'Storage request' signal is raised.
- Channel blocks Z-address from SAB and A-register bytes 2 and 3 from SBI lines.
- 'BCU response' signal gates data address register contents, 'store' signal, storage protect parity bit, and mark bits 2 and 3 to BCU interface.
- 'BCU data request' signal gates channel address bits (0-2) and unit address register bits (0-7) to SBI lines (21-31).
- Delayed 'BCU advance pulse' signal turns off 'sequence 5' latch and gates 'release' signal to CPU interface.
- Fall of 'select channel' signal turns off 'CCW valid', 'IPL', and 'block release IPL' latches.

The IPL ending condition operation is performed to terminate the IPL operation (Diagram 5-19). During the IPL ending condition operation, the channel disconnects the control unit from the I/O interface, stores the channel and unit address in byte locations 2 and 3 of IPL PSW at storage address 0, and releases the CPU to perform processing operations. When the channel 'sequence 5' latch is turned on and the CC flag in the flag register is inactive, the IPL ending condition operation begins when the control unit raises the 'status in' signal and places its status byte on the 'bus in' lines. When the channel detects the device end bit in the status byte, it activates the 'service out' signal and drops the 'select out' signal, causing the control unit to drop the 'status in' and 'operational in' signals. This disconnects the control unit from the I/O interface.

Subsequently, the channel performs an IPL end condition setup operation. During this operation, the channel: (1) raises the 'storage request' signal to the BCU interface;

(2) resets the data address register to 0's with good parity; (3) blocks gating of the channel's Z-address bits to the SAB and bytes 2 and 3 of the A-register to the SBI lines; and (4) activates IPL ending condition logic.

When the channel receives a 'BCU response' signal, it gates the data address register contents (address 0) to the SAB, forces a master key of all 0's with good parity to the 'storage protection' lines, and forces mark bits 2, 3 and P to the mark lines. Upon receipt of the 'BCU data request' signal (for the Model 91, this signal is generated in the channel by delaying the 'BCU response' signal 150 ns), the channel gates channel address bits 0 through 2 to SBI lines 21 through 23 and unit address register bits 0 through 7 to SBI lines 24 through 31. With only mark bits 2 and 3 active, only the bits placed on SBI lines 21 through 31 are stored in main storage.

After the channel receives the 'BCU advance pulse' signal, it turns off the 'sequence 5' latch and raises the 'release' signal to the CPU interface. Subsequently, the 'select channel' signal falls, causing the channel to turn off the 'CCW valid' latch; the 'IPL' latch, and the 'block IPL release' latch. With these latches off, the IPL ending condition operations are complete; i.e., the IPL operation is complete, and the channel is free to resume polling operations and receive instructions from the CPU.

Detailed IPL Operations

The following paragraphs describe the IPL operation at the detailed level. The descriptions are based upon the detailed flow chart (Diagram 5-20) in the FEDM with references to the simplified IPL control logic diagram (Diagram 5-21) and timing diagrams 5-22, 5-23 and 5-24.

Detailed IPL Initial Selection

Detailed operations performed by the channel during the IPL initial selection operation are described separately in the following paragraphs. Control logic for the IPL initial selection operation is shown in Diagram 5-21. Major timing signals for the IPL initial selection operation are shown in timing chart form in Diagram 5-22.

Start IPL. A summary and a description of the start IPL operations follow:

- Channel receives unit address (eight bits +P), 'initial program load' signal, and 'select channel' signal to begin IPL operation.
- Signals are initiated at CPU by setting CHANNEL and UNIT ADDRESS switches and pressing LOAD pushbutton.
- Unit address identifies desired control unit and I/O device.
- 'Select channel' signal identifies desired channel.
- 'Initial program load' signal identifies operation.

The start IPL routine (Diagram 5-20) initiates the IPL operation at the desired channel. In order for the channel to start an IPL operation, it must receive a unit address (eight bits plus P) on the 'unit address bus-out' lines, a simplex 'select channel' signal, and a multiplex 'initial program load' signal via the CPU interface. The unit address identifies the control unit and I/O device from which the program is to be read, the 'initial program load' signal specifies that an IPL operation is to be performed, while the 'select channel' signal identifies the channel which is to perform the IPL operation. Both the unit address and 'select channel' signals are established at the CPU by setting UNIT and CHANNEL ADDRESS switches for the desired values. The unit address bits, the 'select channel' signal, and the 'initial program load' signal are all presented to the channel after the Load pushbutton on the CPU system console is pressed.

Block Timeout and Interrupt End. A summary and a description of the block timeout and interrupt end signals follow:

- 'Block release IPL' and 'IPL' latches turned on.
- Generation of 'release' signal due to condition code generation blocked.
- 100-ms CPU release timeout clock is blocked; prevents interference with IPL operation.
- 'Interrupt end' signal blocked, preventing channel from recognizing interrupts; prevents interference with IPL operation.

The block timeout and interrupt end routine (Diagrams 5-20 and 5-21), prevents the channel from generating a 'release' signal or recognizing interrupt conditions for the duration of the IPL operation.

When both the 'initial program load' and 'select channel' signals are raised at the channel, the 'block release IPL' latch is turned on. This latch remains on for the duration of the IPL operation, and blocks the generation of the 'release' signal to the CPU interface in the event the channel attempts to generate condition codes 1, 2, or 3. The 'block release IPL' signal also turns on the 'IPL' latch which remains on for the duration of the IPL operation. the 'IPL latch' signal blocks the channel's CPU release timeout clock and prevents generation of the 'interrupt

end' signal. For other than IPL (or FLT) operations, the CPU release timeout clock detects the absence of a 'release' signal to the CPU approximately 100 ms after the channel receives the 'select channel' signal (see Diagram 5-25). If this condition is detected, an interface control check condition is detected. This causes the channel to enter a sequence 5 routine and perform an I/O interface disconnect operation. Since, during the IPL operation, the 'release' signal is not generated within the 100 ms period, the channel CPU release timeout clock is blocked to prevent interference with the IPL operation.

Since generation of the 'interrupt end' signal results in turn-on of the 'interrupt' latch, the 'interrupt end' signal is blocked to prevent the channel from accepting an interrupt condition which would interfere with the IPL operation. If the channel has an interrupt condition pending, the channel is not available to perform other operations until the interrupt is cleared; thus, the 'interrupt end' signal is blocked to prevent this possibility.

Channel and I/O Interface Reset. A summary and a description of the channel end I/O interface reset operations follow:

- 'Interface reset' signal (6 usec) causes reset of all control units and I/O devices.
- 'Suppress out' latch reset.
- 'Machine reset' signal (6 usec) causes reset of channel logic and registers.
- Channel clock turned off.

The channel and I/O interface reset routine (Diagram 5-20) causes a reset of all control units and I/O devices attached to the channel, and causes a reset of all channel control logic and registers. The 'interface reset' signal is generated by AND'ing the 'initial program load' ('IPL channel pulse') signal (Diagram 5-21), with the 'not simulate CPU' signal to produce a 'reset suppress out latch' signal. This signal, in addition to resetting the 'suppress out' latch, triggers a 6-usec singleshot. The resulting 'interface reset' signal turns off the 'operational out' latch to the control units for the 6-usec duration. With the 'operational out' signal inactive, the attached control units and I/O devices initiate a reset operation. In addition, the 'interface reset' signal generates a 6-usec 'machine reset' signal which resets the channel's control logic and registers. Other functions performed by the 'interface reset' signal are: (1) blocking of the 'PCI or log wait interrupt' signal to prevent the generation of an 'interrupt request' signal in the event a program controlled interrupt (PCI) or log wait condition is presented to the channel; (2) blocks gating of the unit address register contents to the 'unit address bus-in' lines to the CPU; and (3) blocks turn-on of the 'pseudo accept interrupt' latch, preventing the channel from generating an 'interrupt request' signal by this means.

The 'machine reset' signal, in addition to performing the channel reset operations previously mentioned, turns on the channel clock and then turns the clock off on the fall of the 'machine reset' signal. Turn-on of the clock is not significant during the channel and I/O interface reset routine; rather, the fall of the 'machine reset' signal is significant in that it triggers the clock turn off singleshots to insure that the clock is off at the beginning of the IPL operation (see Diagram 4-3). With the turn-off of the channel clock, the channel and I/O interface reset operations are complete.

100-ms Delay. To permit sufficient time for completion of the channel and I/O interface reset routine, the channel generates a signal 100 ms in duration (see Diagrams 5-20 and 5-21). This signal is generated when the 'initial program load' ('IPL channel pulse') signal rises to trigger a 100-ms singleshot. While the 'singleshot' signal is active, IPL operations are prevented from progressing; the 100-ms delay is sufficient to insure that all I/O devices have completed the reset operation initiated by the 'interface reset' signal.

Channel Forces CCW. A summary and a description of the channel forces CCW operations follow:

- 'IPL sync' latch turned on; 'select out' latch turned off.
- 'Setup' latch turned on to begin device selection.
- UABO bits gated to unit address register.
- 'IPL control' latch turned on for duration of IPL loader operation.
- 'CCW valid' latch turned on; indicates forced CCW is in channel.
- Count of 24 (bits 19 and 20 turned on) forced into count register.
- Command address of eight (bit 20 turned on, P-bit turned off) forced into command address register.
- Read command (bit 6 turned on, P-bit turned off) forced into command register.
- CC flag forced into flag register; P-bit turned off.
- Storage protect register contains 0's with P-bit active (forced during reset routine).

- Data address register contains 0's with P-bits active (forced during reset routine).

The channel forces CCW routine (Diagram 5-20) forces a read 24 bytes beginning at main storage address 0 CCW into the channel control logic and registers. The routine begins with the fall of the '100-ms delay' signal. The fall of this signal triggers a 300-ns singleshot, the output of which is AND'ed with the 'CPU select channel and not simulate CPU' signal to turn on the 'IPL sync' latch and turn off the 'select out' latch. The turn-off of the 'select out' latch prepares the channel to initiate the I/O device selection routine at the proper time; i.e., the 'select out' latch is turned back on at a specified time to initiate selection of the device.

Turn-on of the 'IPL sync' latch produces a 'turn on setup IPL' signal which turns on the 'setup' latch, the 'IPL control' latch, and the 'gate unit address to register' latch. Turn-on of the 'gate unit address to register' latch causes the unit address on the 'unit address bus out' lines (from the CPU interface) to be gated into the channel's unit address register.

Turn-on of the 'IPL control' latch (which remains on for the duration of the IPL loader operation) enables the channel to force and maintain the 'read 24 bytes' CCW into channel logic. In addition, the 'IPL control' signal blocks incorrect length detection by the channel.

Turn-on of the 'setup' latch turns on the channel clock and initiates the I/O device selection routine (see "I/O Device Selection, Initial Setup").

With the channel clock on, the 'IPL latch', 'IPL control latch', 'setup' and 'T0' and 'not T3' clock signals are AND'ed to activate the 'turn on CCW valid IPL' and 'force read operation' signals. These two signals are approximately 300 ns in duration and force the read 24 bytes CCW into the channel control logic and registers.

The 'turn on CCW valid IPL' signal: (1) turns on the 'CCW valid' latch; (2) turns on the bit 20 latch and turns off the P2 latch in the command address register to force a command address of eight; and (3) turns on the bit 18 and 20 latches in the count register to force a count of 24 bytes. The 'force read operation' signal: (1) turns on the CC flag latch and turns off the P-latch in the flag registers; and (2) turns on the bit 6 latch and turns off the P-bit latch in the command register to force a read command. With the data address register and storage protect registers reset to 0 with odd parity (by the previously generated 'machine reset' signal), a data address of 0 and a master storage protect key are present in these registers. With the preceding operations complete, the channel control logic and registers contain a forced CCW which specifies that 24 bytes (three doublewords) are to be read to main storage beginning at address location 0; the forced CCW also specifies that a chain command operation is to be performed following the reading of the three doublewords, and that the CCW to be fetched during the chain command operation is to be obtained from main storage location eight.

I/O Device Selection, Initial Setup. A summary and a description of the I/O device selection initial setup operations follow:

- Unit address gated from unit address register to control unit on 'bus out' lines.
- 'Address out' signal raised at 'T4' clock time.
- Parity of unit address on 'bus out' lines checked.
- 'Select out' signal is raised.
- 'Operational in' signal received; response to 'select out' signal.
- 'Address out' signal dropped by channel.
- 'Address in' signal received; unit address byte is on 'bus in' lines.
- Parity of unit address on 'bus in' lines checked.
- Unit address register contents compared with unit address on 'bus in' lines; comparison indicates correct unit selected.

The I/O device selection, initial setup routine (Diagram 5-20) is performed to connect the desired control unit and I/O device to the channel via the I/O interface. The routine begins with the gating of the unit address register contents to the control units on the multiplex 'bus out' lines. The gating occurs as a result of the 'setup latch' signal.

Following turn-on of the channel clock (during the channel forces CCW routine), a 100-ns 'setup reset' signal is generated as a result of the 'T0' clock signal. In turn, the 'setup reset' signal generates a 100-ns 'setup or machine reset' signal which resets selected channel control logic. This signal does not affect channel logic activated during the channel forces CCW routine; for the IPL routine, generation of this signal performs redundant reset operations performed during the channel and I/O interface reset routine and is significant at this point only in the sense that it is generated during a normal setup routine and is thus generated for the IPL operation.

Following the 'setup reset' signal, the channel 'address out' latch is turned on by the 'T4' clock signal. This raises the multiplex 'address out' signal to the control units indicating that a unit address byte is on the 'bus out' lines.

With the turn-on of the 'address out' latch, the channel generates a 'sample bus out' signal to determine if the unit address parity is correct (odd). If a parity error is detected, the 'channel control check' latch is set, and channel IPL operations are halted; in this case, manual intervention is required to restart the IPL operation.

If a 'bus out' parity error is not detected, the channel 'select out' latch is turned on by AND'ing the 'address out' signal and the 'T7' clock signal. This raises the 'select out' and 'hold out' signals to the control units, causing the 'select out' signal to propagate through the control units until the signal reaches the addressed control unit. With the turn-on of the 'select out' latch, the channel clock is turned off.

If the 'select out' signal is propagated through all control units, the 'select in' signal to the channel rises, indicating that the addressed I/O device was not selected. In this case, the 'address out' and 'select in' signals are AND'ed to set the channel 'no selection' latch; this halts the IPL operation and requires operator intervention to restart the IPL operation.

Assuming that the control unit with the attached I/O device recognizes the unit address, the propagation of the 'select out' signal is stopped at the control unit ('select in' does not rise) and the control unit raises either the 'status in' or 'operational in' signal. If the 'status in' signal is raised, the channel detects a control unit busy condition and halts the IPL program; manual intervention is required to restart the operation.

If the control unit is not busy, it raises the 'operational in' signal to the channel. The rise of the 'operational in' signal implies that the control unit has recognized the unit address sent by the channel. The rise of the 'operational in' signal turns off the 'address out' latch causing the channel to remove the unit address byte from the 'bus out' lines and to turn off the 'address out' latch. The control unit responds to the fall of the 'address out' signal by activating the 'address in' signal.

The 'address in' signal to the channel is AND'ed with the 'CCW valid' signal to activate the 'address in gated' signal. In turn, this signal is AND'ed with the 'operational in' signal to turn on the channel clock. The 'address in gated' signal is also significant in generating the 'sample bus in' signal; this signal samples the unit address on the 'bus in' lines for correct (odd) parity. If a parity error is detected, the channel's 'interface control check' latch is turned on, and the IPL operation stopped; manual intervention is required to restart the IPL operation.

If no bus-in parity error is detected, the channel compares the unit address register contents with the unit address on the 'bus in' lines. If the addresses compare, the control unit has selected the I/O device requested by the channel; if not, an incorrect selection has been performed and the channel's 'interface control check' latch is turned on. With this latch on, the channel stops the IPL operation and manual intervention is required to restart the IPL operation.

If neither a 'bus in' parity error nor address mismatch is detected by the channel, the I/O device selection routine has been successfully completed. In other words, the correct control unit and I/O device have been connected to the channel via the I/O interface.

Command to Control Unit, Initial Setup. A summary and a description of the command to control unit, initial setup operations follow:

- 'Setup' latch is turned off.
- 'Gate command' latch is turned on.
- Read command byte sent to control unit on 'bus out' lines.
- Parity of read command byte is checked.
- 'Command out' signal is raised by channel.
- Control unit drops 'address in' signal; command accepted.
- Channel drops 'command out' signal.

The command to control unit, initial setup routine (Diagram 5-20), is performed to transfer the read command from the channel's command register to the control unit. The routine begins when the channel's 'setup' latch and clock are turned off by AND'ing the 'address compare' signal and the 'T4' clock signal. With the turn-off of the 'setup' latch, the 'gate command' latch is turned on to activate the 'read' signal. Following turn-on of the 'gate command' latch, the channel turns on the 'gate command to bus out' latch which gates the read command byte to the 'bus out' lines. The 'gate command to bus out' signal is also AND'ed with the 'address in' signal from the control unit to turn on the channel clock. With the read command byte on the 'bus out' lines, the channel generates a 100-ns 'sample bus out' signal with the 'T2' and 'not T3' clock signals. If the read command byte parity is incorrect (even), the 'channel control check' latch is turned on to stop the IPL operation; manual intervention is required to restart the operation. If a parity error is not detected, the 'address in gated' signal and 'T7' clock signal are AND'ed to raise the 'command out' signal to the control unit. The 'command out' signal informs the control unit that the read command is on the 'bus out' lines.

When the control unit accepts the read command, it drops the 'address in' signal, causing the channel to drop

the 'command out' signal. With the drop of the 'command out' signal, the command to control unit, initial setup routine is complete.

Status From Control Unit, Initial Setup. A summary and a description of the status from control unit, initial setup operations follow:

- Status byte and 'status in' signal sent to channel from control unit.
- Channel samples for status byte parity error in 'bus in' lines.

During the status from control unit, initial setup routine (Diagram 5-20), the channel receives the status byte from the control unit and samples the byte for correct parity. After the channel drops the 'command out' signal, it waits for the control unit to send the 'status in' signal and place the status byte on the 'bus in' lines.

The rise of the 'status in' signal turns off the channel clock, turns on the 'sequence 1' latch, and generates a 'status in singleshot' signal. The 'status in singleshot' signal samples the status byte on the 'bus in' lines for correct (odd) parity. If even parity is detected, the 'interface control check' latch is turned on and the IPL operation is stopped; manual intervention is required to restart the program. If odd parity is detected, the status from control unit, initial setup routine is complete.

Status Examination, Initial Setup. A summary and a description of the status examination, initial setup operations follow:

- Channel clock turned on.
- Latch status byte into 'bus in' latches.
- Examine status byte for all 0's; control unit and I/O device ready to perform command.

The status examination, initial setup routine (Diagram 5-20), is performed to determine whether the control unit and I/O device are ready to perform the read command. The routine begins with the turn-on of the channel clock by AND'ing the 'sequence 1' and 'status in' signals. Clock signals 'T0' and 'not T1' are AND'ed to perform two parallel operations: (1) sample the count register for a count of less than nine bytes; and (2) latch the status byte into the 'bus in' latches. The first operation is normally performed to turn on the 'last word' trigger in the event that the total count provided by the CCW is less than nine bytes. Since, for the channel forced CCW, the count is 24, the 'last word' trigger is not turned on when the count register value is sampled. In case 2 above, the status byte is latched into the 'bus in' latches to permit the channel to examine the status byte bits. While latched, the channel examines the status byte for all 0's. If any bit of the status byte is active, the control unit or I/O device is not ready to perform the command and the channel stops the IPL operation; manual intervention is required to restart the IPL operation. If the status byte contains all 0's, the control unit and I/O device are ready to perform the command, and the status examination, initial setup routine is complete.

Clear Initial Status, Initial Setup. A summary and a description of the clear initial status, initial setup operations follow:

- Turn on 'sequence 2' latch, raise 'service out' signal and turn off 'latch status byte' trigger.
- Control unit drops 'status in' signal in response to 'service out' signal.
- 'Service out' signal drops when 'status in' signal drops.
- Turn off 'sequence 1' latch; turn off clock.

The clear initial status, initial setup routine (Diagram 5-20), is performed to cause the control unit to remove the status byte and drop the 'status in' signal prior to executing the read command. With the 'status equal 0' signal active, the channel's 'sequence 2' latch is turned on and the 'latch status byte' trigger is turned off by the 'T2' clock signal. Turn-on of the 'sequence 2' latch is required for the channel to terminate the clear initial status, initial setup routine and to control the IPL loader read operations that follow. Turn-off of the 'latch status byte' trigger unlatches the 'bus in' latches, freeing the 'bus in' lines when the status byte is removed by the control unit.

The 'T2' clock signal is also AND'ed with the 'read' and 'status in' signals to raise the 'service out' signal to the control unit. The 'service out' signal indicates that the channel has accepted the status byte and no longer requires it on the 'bus in' lines. Subsequently, the control unit removes the status byte from the 'bus in' lines and drops the 'status in' signal. The fall of the 'status in' signal: (1) degates the 'service out' signal to the control unit; and (2) is AND'ed with the 'sequence 2' signal to turn off the 'sequence 1' latch. With the turn-off of the 'sequence 1' latch, the channel clock is turned off to end the clear initial status, initial setup routine. The channel is now prepared to control the IPL loader read operations.

Detailed IPL Loader Read Operations

The following paragraphs present a detailed description of the IPL loader read operations performed by the channel. Descriptions are based upon Diagram 5-20 in the FEDM. Refer to the read operation timing chart, (Diagram 5-6) for the major timing signal relationships. Recall that the 'IPL', 'block release IPL' and 'IPL control' latches are on for the entire IPL loader read operation.

Control Unit to Channel Data Handling, IPL PSW. A summary and a description of the operations follow:

- Control unit raises 'service in' signal and places data byte on 'bus in' lines.
- Channel raises 'service out' signal; latches data byte into 'bus in' latches.
- Channel gates data byte to B-register byte 0 position.
- Channel steps byte counter.
- Control unit drops 'service in' signal.
- Channel drops 'service out' signal; unlatches 'bus in' latches.
- Eight exchanges of 'service in' and 'service out' signals cause channel to gate eight data bytes (the IPL PSW) to the eight B-register byte position.
- On last exchange, channel turns on 'byte count equals 0' latch.
- Channel turns on 'B-register' full latch.

The control unit to channel data handling, IPL PSW routine (Diagram 5-20), is performed to assemble eight data bytes, representing the IPL PSW doubleword, into the channel's B-register. The program bytes are obtained from the control unit on a byte-by-byte basis.

The routine begins when the control unit raises the 'service in' signal and places a data byte on the 'bus in' lines to the channel. Assuming a previous error was not detected in the channel, ('sequence 5' latch off), the 'service in' signal is AND'ed with the 'sequence 2' and 'not B-register full' signals to turn on the 'service out' latch. With the 'service out' latch on, the 'service out' signal to the control unit rises. In the channel, the 'service out' signal activates the two 'latch bus in' signals, latching the program byte into the 'bus in' latches. Subsequently, the 'service out' signal triggers two singleshots (ALD CK117) to produce a signal which samples the parity of the program byte in the 'bus in' latches. If parity is incorrect, the 'channel data check' latch is turned on; however, this does not stop the IPL operation, but merely lights the CHANNEL STATUS CHECKS DATA indicator on the CE panel to indicate the condition has been detected.

Following the 'bus in' parity check, the program byte is gated into byte position 0 of the B-register and mark bit 0 in the mark-B register is set. Gating of the byte and setting of the mark bit result when the 'service out single-shot' signal turns on the 'gate bus in to B-register' signal. The latter signal is applied to the byte count encoder logic in the byte counter to activate the 'gate in byte 0' signal. This signal gates the byte to the B-register and sets the mark bit.

The 'gate bus in to B-register' signal also activates the 'change byte count register' signal which steps the byte count register to a count of one. On the fall of the 'change byte count register' signal, the byte count latches in the byte counter are advanced. (The latches remain stable until the fall of the 'change byte count register' signal, since the output of the latches is decoded by the byte count encoder logic, and determine into which B-register byte position the program byte is to be gated.)

The control unit, after receipt of the 'service out' signal removes the program byte from the 'bus in' lines and drops the 'service in' signal. The drop of the 'service in' signal turns off the 'service out' latch causing the 'latch bus-in' signals to fall and the 'service out' signal to the control unit to drop. With the first program byte in B-register position 0, mark bit 0 in the mark-B register set, the byte count register advanced to a count of 1, and the 'service out' latch turned off, the channel waits for the next 'service in' signal from the control unit.

To complete the control unit to channel data handling, IPL PSW routine, the control unit and channel exchange seven more 'service in' and 'service out' signals. During each exchange the channel operations described above are performed, except that the program byte is entered into the next sequential B-register position and the corresponding mark bit is set in the mark-B register.

When the last (eighth) program byte is gated into the B-register, the byte count latches equal 0. The channel detects this condition and turns on the 'byte count equals 0' trigger by AND'ing the detected condition with the 'service out' and 'not sequence 5' signals. Turn-on of the 'byte count equals 0' trigger signifies that the 'B-register' byte boundary has been reached; i.e., a program byte has been gated into B-register byte location 7, thus the IPL PSW doubleword has been fully assembled in the B-register.

The 'byte count equals 0' signal is AND'ed with the 'read' signal to turn on the 'B-register full' latch. With the 'B-register full' latch on, the channel is prevented from responding to further 'service in' signal from the control unit. (The 'B-register full' signal inhibits turn-on of the 'service out' latch.) With the 'B-register full' latch on, the control unit to channel, IPL PSW routine is complete.

Data to A-Register and Start Sequence 3, IPL PSW. A summary and a description of the operations follow:

- B-register doubleword (IPL PSW) transferred to A-register.
- Mark-B register bits transferred to Mark-A register.
- Clock and 'sequence 3' latch turned on.

The data to A-register and start sequence 3, IPL PSW routine (Diagram 5-20) is performed to transfer the IPL PSW to the channel's A-register and to turn on the clock and 'sequence 3' latch.

The contents of the B-register are transferred to the A-register when the 'B-to A-register' latch is turned on by the 'read, BC = 0, and not read CDA A Loaded (Ldd)' signal. This signal is activated by AND'ing the 'not A-register full' and 'BC = 0 or CTB' signals. The 'not A-register full' signal is derived from the off condition of the 'A-register full' latch and indicates that the A-register contains no significant data. The 'BC = 0 or CTB' signal is derived from the on condition of the 'byte count equals 0' trigger.

The 'BC = 0 or CTB' and 'not A-register full' signals are also AND'ed to activate the 'read and BC = 0' signal which turns on the channel clock and the 'sequence 3' latch. With the turn on of the clock and 'sequence 3' latch, the data to A-register and start sequence 3, IPL PSW operation is complete. The channel now performs three parallel routines: B- and A-register full latches, IPL PSW; update count, IPL PSW; and store IPL PSW. Each routine is described separately in the following paragraphs.

B-and A-Register Full Latches, IPL PSW. A summary and a description of the operations follow:

- B-reigster is reset.
- 'Byte count equals 0' trigger is turned off.
- 'B-register full' latch is turned off.
- 'A-register full' latch is turned on.
- Channel starts to assemble IPL CCW 1 in B-register.

The B- and A-register latches, IPL PSW routine (Diagram 5-20) is performed to prepare the channel to assemble the IPL CCW1 doubleword in the B-register and to acknowledge completion of the transfer of the IPL PSW doubleword into the A-register.

The routine consists of AND'ing the 'sequence 3' signal and the 'T2' clock signal to: (1) reset the B-register in preparation for receipt of IPL CCW 1 bytes; (2) turn off the 'byte count equals 0' trigger; (3) turn off the 'B-register full' latch to allow turn-on of the channel 'service out' latch in response to a 'service in' signal; and (4) turn on the 'A-register full' latch to indicate that the IPL PSW doubleword is in the A-register awaiting transfer to main storage. With the 'B-register full' latch off, the channel is prepared to resume the control unit to channel data handling routine to assemble the IPL CCW 1 doubleword in the B-register.

Store IPL PSW. A summary and a description of the operations follow:

- Turn on 'storage request' latch; raise 'storage request' signal to BCU interface.
- Turn on 'storage cycle' trigger; activate channel logic for store cycle.
- Receive 'BCU response' signal:
  1. Gate data address register (address 0) to SAB.
  2. Gate mark-A register (all bits active) to mark bus.
  3. Gate storage protect register (all 0's plus P) to 'storage protection' lines.
  4. Gate 'store' signal to BCU interface.
- Receive 'BCU data request' signal (generated in channel for Model 91); gate A-register (IPL PSW bits) to SBI.
- Receive 'accept' signal from BCU interface; turn on 'accept' latch and turn off 'storage request' and 'A-register full' latches.
- 'BCU response' signal falls; turn on 'remember BCU response' latch.
- 'BCU data request' signal falls.
- Receive 'BCU advance pulse' signal:
  1. Delay approximately 150 ns and turn off 'remember BCU response' and 'accept' latches.
  2. Delay approximately 260 ns and turn off 'storage cycle' latch. (Delay is different for different System/360 models.)

The store IPL PSW routine (Diagram 5-20) is performed to store the IPL PSW doubleword (in the A-register) into main storage location 0. The routine is initiated with the turn-on of the 'storage request' latch by the 'T0' clock signal. With the latch on, the 'storage request' signal to the BCU interface rises and the channel 'storage cycle' trigger is turned on.

With the 'storage cycle' trigger on, the channel awaits receipt of the 'BCU response' signal from the BCU interface. Receipt of the signal indicates that the channel's

'storage request' signal has been honored and causes the channel to gate the required address information and 'store' signal to the BCU interface. Address information gated consists of: (1) the data address register contents (address 0) which are gated to the SAB lines; (2) the mark-A register contents (all eight mark bits active) which are gated to the mark bus; and (3) the storage protect register contents (all 0's with the parity bit active) which are gated to the 'storage protection' lines.

With the address information on the buses and the 'store' signal active, the channel waits for the 'BCU data request' signal from the BCU interface. (For the System/360 Model 91, the channel generates the 'BCU data request' signal approximately 150 ns after receipt of the 'BCU response' signal.) Upon receipt of the 'BCU data request' signal, the contents of the A-register (IPL PSW doubleword) are gated to the SBI lines.

Subsequently, the channel receives an 'accept' signal via the BCU interface which turns on the 'accept' latch and turns off the 'storage request' and 'A-register full' latches. Receipt of the 'accept' signal indicates that main storage has began the storage cycle to store the IPL PSW.

When the 'BCU response' signal drops at the channel, the 'remember BCU response' latch is turned on. With the 'accept latch' and 'remember BCU response latch' signals active, the channel waits for the 'BCU data request' signal to fall, indicating that the A-register contents are no longer required on the SBI lines. When the 'accept' signal is received, the 'A-register full' latch is turned off; turn-off of this latch is required in order for the channel to perform the data address update, IPL PSW routine, (see "Data Address Update, IPL PSW") and to prepare the A-register to receive the IPL CCW 1 doubleword.

To complete the store IPL PSW routine, the channel waits for the 'BCU advance pulse' signal from the BCU interface. When received, the 'BCU advance pulse' signal is delayed approximately 250 to 300 ns, depending upon the System/360 Model with which the channel is used. This delayed 'advance' signal turns off the 'remember BCU response' and 'accept' latches. The 'BCU advance pulse' signal is also delayed to form a 'late advance pulse' signal. This 'late advance' signal is delayed a different amount for different models of the System/360, and used to turn off the 'storage cycle' trigger. The 'late advance' pulse signal is further delayed to generate 'storage cycle complete' signal, which signifies the end of the store IPL PSW routine.

Update Count, IPL PSW. A summary and a description of the operations follow:

- Gate count register contents (24 decimal) minus eight decimal to adder.
- Latch adder; gate adder value (16 decimal) to count register.
- Turn on 'sequence 4' latch.

The update count, IPL PSW routine (Diagram 5-20) is performed to decrement the value in the count register by eight bytes; decrementing the count register is necessary to reflect the transfer of the eight program bytes comprising the IPL PSW to main storage.

The update count, IPL PSW routine is initiated by AND'ing the 'sequence 3' signal and the 'T0' clock signal to gate the count register contents (24 decimal) minus eight to the adder. At clock time T4, the adder is latched and the decremented count value (16 decimal) is gated into the count register. Subsequently, the 'T5' clock signal is AND'ed with the 'sequence 3' and 'not CDA latch' signals to turn on the 'sequence 4' latch. With the decremented count in the count register and the 'sequence 4' latch on, the update count, IPL PSW routine is complete.

Data Address Update, IPL PSW. A summary and a description of the operations follow:

- Data address register contents (address 0) are incremented by eight bytes.
- Routine begins when 'A-register full' latch is turned off.
- Channel clock and 'sequence 3' latch are turned off.
- Turn on clock; gate data address register contents (address 0) plus eight to adder.
- Latch adder; gate incremented data address (eight) to data address register.
- Turn off 'sequence 4' latch.

The data address update, IPL PSW routine (Diagram 5-20) is performed to provide a sequential main storage address at which the next doubleword (IPL CCW 1) is to be stored. During the routine, the data address register contents are incremented eight bytes.

The data address update routine begins when the store IPL PSW routine has progressed to the point where the 'A-register full' latch is turned off (see "Store IPL PSW"). With the latch off, the 'not A-register full' signal is AND'ed with the 'sequence 3' and 'sequence 4' signals to turn off the channel clock. Turn-off of the channel clock produces a 'not T1' clock level which is AND'ed with the 'sequence 4' signal to turn off the 'sequence 3' latch. When the 'sequence 3' latch is turned off, the channel performs two parallel operations: (1) turns on the channel clock by AND'ing the 'not sequence 3', 'sequence 4' and 'not A-register full' signals; and (2) gates the data address register contents (address 0) to the adder by AND'ing the 'sequence 4', 'not sequence 3' and 'not T6' signals. The channel

clock is turned on to time the remainder of the data address update, IPL PSW routine.

With the 'T4' and 'not T6' clock signals, the channel latches the incremented data address into the adder; the 'T4' and 'not T5' clock signals are AND'ed to gate the incremented data address (eight) into the data address register. The register now contains the address at which the next program doubleword (IPL CCW 1) is to be stored.

The data address update, IPL PSW routine is completed when the 'sequence 4' latch is turned off (by AND'ing the 'T6' clock signal with the 'not sequence 3' signal). In the meantime, the control unit to channel data handling, IPL CCW 1 routine is in progress.

Control Unit to Channel Data Handling, IPL CCW 1. A summary and a description of the operation follow:

- Eight program bytes are loaded into B-register to assemble IPL CCW 1.
- Channel operations are identical to control unit to channel data handling, IPL PSW routine.
- Routine ends when 'byte count equals 0' trigger and 'B-register full' latch are turned on.

The control unit to channel data handling, IPL CCW 1 routine (Diagram 5-20) is performed to assemble the IPL CCW 1 doubleword in the B-register. Channel operations for this routine are identical to those described for the control unit to channel data handling, IPL PSW routine. When the eighth exchange of 'service in' and 'service out' signals occurs between the control unit and channel, the byte count latches equal 0 condition is detected to turn on the 'byte count equals 0' trigger, indicating that the eight bytes comprising the IPL CCW 1 doubleword are in the B-register. The 'B-register full' latch is then turned on to end the control unit to channel data handling, IPL CCW 1 routine.

Data to A-Register and Start Sequence 3, IPL CCW 1. A summary of the operation follows:

- Routine is identical to data to A-register and start sequence 3, IPL PSW routine (see Diagram 5-20).
- B-register contents (IPL CCW 1) are gated to A-register.
- Mark-B register bits (eight active bits) are gated to mark-A register.
- Channel clock and 'sequence 3' latch are turned on.

B-and A-Register Full Latches, IPL CCW 1. A summary of the operation follows:

- Routine is identical to B-and A-register full latches, IPL PSW routine (see Diagram 5-20, FEDM).
- B-register is reset.
- 'Byte count equals 0' and 'B-register full' latches are turned off.
- 'A-register full' latch is turned on.
- Channel begins control unit to channel data handling, IPL CCW 2 routine.

Store IPL CCW 1. A summary and description of the operation follow:

- Routine is identical to store IPL PSW routine with the following exceptions:
  1. Data address 8 (rather than 0) is gated from data address register to SAB.
  2. IPL CCW 1 doubleword (rather than the IPL PSW doubleword) is gated from A-register to SBI.

The store IPL CCW 1 routine is performed by the channel to store the IPL CCW 1 doubleword contained in the A-register into main storage location 8. For details of the store IPL CCW 1 routine, refer to Diagram 5-20 and "Store IPL PSW" in this chapter. Note that data address 8 (rather than data address 0) is gated from the data address register to the SAB by the 'BCU response' signal. Also, note that the IPL CCW 1 doubleword (rather than the IPL PSW doubleword) is gated from the A-register to the SBI lines. All other operations are identical for the two routines.

Update Count, IPL CCW 1. A summary and a description of the operation follow:

- Count register value (16 decimal) minus eight is gated to adder.
- 'Last word' trigger is turned on upon detection of count equals 16 or less.
- Adder is latched; decremented count (eight) is gated to count register.

The update count, IPL CCW 1 routine (Diagram 5-20) is performed to decrement the value in the count register by eight bytes and to turn on the 'last word' trigger. Decrementing the count register is necessary to reflect the

transfer of the eight program bytes comprising the IPL CCW 1 doubleword to main storage. Turn-on of the 'last word' trigger enables the channel to detect the presence in the B-register of the last program byte of the IPL loader read operation.

The update count, IPL CCW 1 routine is initiated by AND'ing the 'sequence 3' signal and the 'T0' clock signal to gate the count register contents (16 decimal) minus eight to the adder. At 'T3' and 'not T4' clock times, the count in the counter register (still 16 decimal) is sampled; the 'count equal or less than 16' signal is AND'ed with the two clock signals to turn on the 'last word' trigger.

At clock time T4, the adder is latched and the decremented count value (eight decimal) is gated into the count register. Subsequently, the 'T5' clock signal is AND'ed with the 'sequence 3' and 'not CDA latch' signals to turn on the 'sequence 4' latch. With the decremented count in the count register and the 'sequence 4' latch on, the update count, IPL PSW routine is complete.

Data Address Update, IPL CCW 1. A summary of the operation follows:

- Routine is identical to the data address update, IPL PSW routine, except incremented address value is 16 rather than 8.
- Data address register is incremented to provide main storage address for IPL CCW 2 doubleword.
- Refer to Diagram 5-20 for a flow chart description of the data address update, IPL CCW 1 routine and to "Data Address Update, IPL PSW" for a text description of the routine.

Control Unit to Channel Data Handling, IPL CCW 2. A summary of the operation follows:

- Eight IPL CCW 2 bytes are assembled into B-register on byte-by-byte basis.
- Control unit and channel exchange eight 'service in' and 'service out' signals to gate bytes into eight B-register locations.
- With 'last word' trigger on, channel detects byte count equals count B condition when last byte is accepted.
- 'BC = CTB' trigger and latch are turned on.
- Count register is reset.
- 'Sequence 5' latch is turned on.

The control unit to channel data handling, IPL CCW2 routine (Diagram 5-20) is performed to load the last eight bytes (IPL CCW 1) of the IPL loader read operation into the B-register. The process of gating the eight bytes into the B-register (control unit and channel exchange of 'service in' and 'service out' signals) is described in "Control Unit to Channel Data Handling, IPL PSW". Channel operations for the control unit to channel data handling, IPL CCW 2 routine are, however, altered when the last program byte is gated into the B-register.

Recall that the 'last word' trigger was turned on during the previous update count, IPL CCW 1 routine. With the 'last word' signal active, the channel detects a BC = CTB condition when the eighth byte is gated into the B-register. Detection of the BC = CTB condition indicates that the last byte of the IPL loader read operation is in the channel.

The 'BC = CTB' signal is AND'ed with the 'last word' and 'gate BC = 0 latch' signals to turn on the 'BC = CTB' trigger. The resulting 'BC = CTB' trigger signal sets the 'BC = CTB' latch and is also OR'ed to activate the 'BC = 0 or CTB' signal. The latter signal is then AND'ed with the 'read' signal to turn on the 'B-register full' latch.

The 'BC = CTB latch' signal performs two functions: (1) activates the 'count ingate reset' signal to reset the count register; and (2) is AND'ed with the 'not service in' and 'not CDA' signals to turn on the 'sequence 5' latch. The count register is reset to 0 to indicate that all program bytes required by the channel forced CCW have been received by the channel. The 'sequence 5' latch is turned on to enable the channel to perform the IPL chain command to IPL CCW 1 operations when the IPL loader read operations are complete.

With the reset of the count register and turn-on of the 'sequence 5' latch, the control unit to channel data handling, IPL CCW 2 routine is complete.

Data to A-Register and Turn On Sequence 3, IPL CCW 2. A summary and a detailed description of the operation follow:

- B-register contents (IPL CCW 2) are gated to A-register.
- Mark-B register bits (eight active bits) are gated to mark-A register.
- Clock is turned on.
- 'Sequence 3' latch is turned on (update count operation is not performed).

The data to A-register and turn on sequence 3, IPL CCW 2 routine is performed to transfer the IPL CCW 2 doubleword from the B-register to the A-register, to transfer

mark-B register bits to the mark-A register, and to turn on the channel clock and 'sequence 3' latch. The 'sequence 3' latch is turned on, not to perform an update count routine, but to enable other functions requiring the presence of the 'sequence 3' signal to be performed.

The contents of the B-register are transferred to the A-register when the 'B- to A-register latch is turned on by the 'read, BC = 0 and not read CDA A loaded (Ldd)' signal. This signal is activated by AND'ing the 'not A-register full' and 'BC = 0 or CTB' signals. The 'not A-register full' signal is derived from the off condition of the 'A-register full' latch and indicates that the A-register contains no significant data. The 'BC = 0 or CTB' signal is derived from the on condition of the 'BC = CTB' trigger.

The 'BC = 0 or CTB' and 'not A-register full' signals are also AND'ed to activate the 'read and BC = 0' signal which turns on the channel clock and 'sequence 3' latch. With the clock and 'sequence 3' latch on, the data to A-register and turn on sequence 3, IPL CCW 2 routine is complete. The channel now performs three parallel routines: condition Stop command, store IPL CCW 2, and data address update, IPL CCW 2.

Stop Command. A summary and a description of the operation follow:

- Channel inhibits activation of 'service out' signal.
- All 0's with good parity (stop command) gated to 'bus out' latches.

The stop command routine (Diagram 5-25) is performed to present a stop command byte to the control unit; in the event the control unit presents another data byte to the channel, the channel will (during the IPL chain command to IPL CCW 1 operations) raise the 'command out' signal to the control unit.

The stop command routine consists of (1) inhibiting the activation of the 'service out' signal with the 'sequence 5' signal; and (2) gating the stop command byte (all 0's with parity bit active) to the 'bus out' latches. Gating of the stop command byte is accomplished by AND'ing the 'sequence 5' and 'not service out' signals. At this point, the channel waits for completion of the parallel data address update operation ('sequence 3' and 'sequence 4' latches off) before determining whether to raise the 'command out' signal, thus entering the Stop command into the control unit.

Store IPL CCW 2. A summary and a description of the operation follow:

- Routine is identical with store IPL PSW routine with the following exceptions:
  1. Data address 16 (rather than 0) is gated from the data address register to SAB.
  2. IPL CCW 2 doubleword (rather than the IPL PSW doubleword) is gated from A-register to SBI.

The store IPL CCW 2 routine is performed by the channel to store the IPL CCW 2 doubleword contained in the A-register into main storage location 16. For details of the store IPL CCW 2 routine, refer to Diagram 5-20 and "Store IPL PSW" in this chapter. Note that data address 16 (rather than data address 0) is gated from the data address register to the SAB by the 'BCU response' signal. Also, note that the IPL CCW 2 doubleword (rather than the IPL PSW doubleword) is gated from the A-register to the SBI lines. All other operations are identical for the two routines.

Data Address Update, IPL CCW 2. A summary and a description of the operation follow:

- Reset B-register; turn off 'BC = CTB' trigger and 'B-register full' latch.
- Turn on 'A-register full' latch.
- Turn off 'BC = CTB' latch.
- Wait for turn-off of 'A-register full' latch.
- Turn off clock and 'sequence 3' latch.
- Turn on clock.
- Gate data address register contents (address 16) plus eight to adder.
- Latch adder; gate incremented address (24) to data address register.
- Turn off 'sequence 4' latch.
- Turn off clock.

- Check 'bus-out' parity.
- Turn on 'select out' latch.
- Turn off clock.
- Await turn-on of 'CCW valid' latch (when valid IPL CCW 1 is in channel).

The I/O device selection, CC setup routine (Diagram 5-20) is performed to reconnect the I/O device containing IPL program to the channel. At a specified point in the routine, the channel begins the parallel fetch IPL CCW 1 routine.

The I/O device selection, CC setup routine begins when the 'setup' latch is turned on by the 'turn on setup CC' signal. With the 'setup' latch on, the I/O device address (unit address) is gated from the unit address register to the control unit on the 'bus out' lines and the channel clock is turned on.

Following turn-on of the channel clock, a 100-ns 'setup reset' signal is generated as a result of the 'T0' clock signal. In turn, the 'setup reset' signal generates a 100-ns 'setup or machine reset' signal which resets selected channel control logic. For example, the 'setup or machine reset' signal turns off the 'sequence 2' and 'sequence 5' latches.

Following the 'setup reset' signal, the channel 'address out' latch is turned on by AND'ing the 'T4', 'not T5' and 'not operational in' signals. This raises the multiplex 'address out' signal to the control units indicating that a unit address byte is on the 'bus out' lines. In parallel with the turn-on of the 'address out' latch, the 'T0' clock signal is AND'ed with the 'chain command latch' and 'not T1' signals to: (1) turn off the 'CCW valid' latch; (2) turn on the 'CCW fetch' latch; (3) turn on the 'storage request' latch; and (4) in the case of the System/360, Model 91, activate the 'control word request' signal. With the 'storage request' latch on, the 'storage request' signal to the BCU interface (and for the Model 91, the 'control word request' signal to the BCU interface) is raised to initiate a storage request for IPL CCW 1 in main storage. (See "Fetch IPL CCW 1".) Following turn-off of the 'CCW valid' latch, the 'gate command' latch is turned off, degating the 'read' signal. (The read command is still in the command register.)

When the 'address out' latch is turned on, the channel generates a 'sample bus out' signal to determine if the unit address parity is correct (odd). If a parity error is detected, the 'channel control check' latch is set and channel IPL operations are stopped; manual intervention is required to restart the IPL operation.

If a bus-out parity error is not detected, the channel 'select out' latch is turned on by AND'ing the 'address out' signal and the 'T7' clock signal. This raises the 'select out' and 'hold out' signals to the control units, causing the 'select out' signal to propogate through the control units until the signal reaches the addressed control unit. With the turn-on of the 'select out' latch, the channel clock is turned off.

If the 'select out' signal is propagated through all control units, the 'select in' signal to the channel rises, indicating that the addressed I/O device was not selected. In this case, the 'address out' and 'select in' signals are AND'ed to set the channel 'no selection' latch; this stops the IPL operation and requires manual intervention to restart the IPL operation.

Assuming that the control unit with the attached I/O device recognizes the unit address, the propagation of the 'select out' signal is stopped at the control unit ('select in' does not rise) and the control unit raises either the 'status in' or 'operational in' signal. If the 'status in' signal is raised, the channel detects a control unit busy condition and halts the IPL program; manual intervention is required to restart the operation.

If the control unit is not busy, it raises the 'operational in' signal to the channel, indicating that the control unit has recognized the unit address sent by the channel. The rise of the 'operational in' signal turns off the 'address out' latch, causing the channel to remove the unit address byte from the 'bus out' lines and to turn off the 'address out' latch. The control unit responds by activating the 'address in' signal to the control unit. At this point, the channel waits for the turn-on of the 'CCW valid' latch, which occurs when the valid IPL CCW 1 is in the channel. Thus, with the turn-off of the 'address out' latch, the I/O device selection, CC setup routine is complete.

Fetch IPL CCW 1. A summary and a description of the operation follow:

- Routine performed to fetch IPL CCW 1 from main storage location 8.
- 'Storage request' latch is on (from I/O device selection, CC setup routine).
- 'Storage cycle' latch is turned on.
- Gate command address register value (eight) plus eight to adder.
- Receive 'BCU response' signal.
- Gate command address register value (eight) to SAB.
- 'BCU response' signal drops; degate command address to SAB.
- Receive 'BCU advance pulse' signal; receive IPL CCW 1 on SBO lines.

- Gate SBO bits as follows:
  1. 0 through 7 plus P to command register.
  2. 8 through 31 plus 3P to data address register.
  3. 32 through 36 plus P to flag register.
  4. 48 through 63 plus 2P to count register.

- Sample SBO bits 0 through 7 for TIC command (program check condition, if present).

- Sample SBO bits 37 through 39 for all 0's and SBO bits 48 through 63 for other than all 0's (program check condition exists, if otherwise).

- Turn on 'CCW valid' latch (if no program check condition exists).

- Latch adder; gate incremented command address value (16 decimal) to command address register.

The fetch IPL CCW 1 routine (Diagram 5-20) is performed to fetch IPL CCW 1 from main storage location 8, where it was stored during the IPL loader read operations. IPL CCW 1 is required by the channel to specify the number of program bytes to be transferred to main storage, and the locations at which these bytes are to be stored.

Recall that the 'storage request' latch was turned on during the I/O device selection, CC setup routine. The resulting 'storage request' signal, in addition to being sent to the BCU interface, also turns on the 'storage cycle' latch. The 'storage cycle' signal activates and deactivates applicable channel logic for the duration of the storage cycle. Recall, also, that the 'CCW fetch' latch was turned on during the I/O device selection, CC setup routine. The 'CCW fetch' signal turns on the 'gate command address to adder' latch. In turn, the output of this latch gates the command address register value (eight) plus eight to the adder. However, the value in the command address register is not changed until the end of the fetch IPL CCW 1 routine.

The channel then waits for the 'BCU response' signal from the BCU interface. Receipt of this signal indicates that the 'storage request' signal issued by the channel has been honored. Upon receipt of the 'BCU response' signal, the channel gates the command address register value (eight) to the SAB. After the channel receives an 'accept' signal, main storage uses the command address to fetch IPL CCW 1 from address location 8.

When the 'BCU response' signal falls at the channel, the 'latch adder command address' latch is turned on to latch the command address value (16 decimal) into the adder. Subsequently, the channel awaits receipt of the 'BCU advance pulse' signal from the BCU interface. This signal, for the fetch operation, indicates that the IPL CCW 1 doubleword will be available on the SBO lines in approximately 220 ns. The channel delays the 'BCU advance pulse' signal to form a 'raw advance SS' signal, an 'advance pulse' signal, a 'late advance pulse' signal, and a 'storage cycle complete' signal. Each of these signals is delayed a different amount, and the delay is different for different Models of the System/360. (Refer to FEMM, 2860 Selector Channel for specific timing for each Model.)

When the IPL CCW 1 bits are on the SBO lines, the delayed 'raw advance SS' signal causes the SBO bits to be gated to the channel registers as follows:

1. SBO bits 0 through 7 plus P0 are gated into the command register.
2. SBO bits 8 through 31 plus P1, P2 and P3 are gated into the data address register.
3. SBO bits 32 through 36 plus P4 are gated into the flag register.
4. SBO bits 48 through 63 plus P7 and P8 are gated into the count register.

While the IPL CCW 1 bits are still on the SBO, the delayed 'advance pulse' signal samples the command bits for a TIC command. (A TIC command is indicated if SBO bit 4 equals a logic 1 and SBO bits 5 through 7 equal logic 0's.) Detection of a TIC command turns on the 'TIC' and 'TIC cycle' latches and constitutes a program violation (TIC command in first CCW). In this case, the channel 'program check' latch is turned on and the channel stops the IPL operation. Manual intervention is required to restart the IPL operation. If a TIC command is not detected, the operation continues.

Following the TIC command sample, and while the 'IPL CCW 1' bits are still on the SBO, the delayed 'late advance pulse' signal samples SBO bits 37 through 38 for all 0's, and SBO bits 48 through 63 (count bits) for other than all 0's. If these bits are other than indicated, a program violation exists and the 'program check' latch is turned on to stop the channel IPL operation. Manual intervention is required to restart the IPL operation.

If a program check condition is not detected, the 'CCW valid' latch is turned on and the 'CCW fetch' latch is turned off with the 'not late advance pulse' signal. Turn-on of the 'CCW valid' latch indicates that a properly formatted IPL CCW 1 is in the channel registers.

The 'late advance pulse' signal also gates the incremented command address value (16 decimal) from the adder into the command register. The incremented command address will be used to fetch IPL CCW 2 from main storage if the CC flag or CDA flag is active in IPL CCW 1.

With the 'CCW valid' latch on, the fetch IPL CCW 1 routine is complete, and the channel enters the address byte from control unit, CC setup routine.

Address Byte From Control Unit, CC Setup. A summary and a description of the operation follow:

- Unit address byte received from control unit on 'bus in' lines.
- 'Address in gated' signal is activated by 'CCW valid' and 'address in' signals.
- Unit address byte parity is checked; parity error turns on 'interface control check' latch.
- Unit address register contents compared with unit address on 'bus in' latches; mismatch will turn on 'interface control check' latch and stop IPL operation.

The address byte from control unit, CC setup routine (Diagram 5-20) is performed to determine whether the correct I/O device was selected during the I/O device selection, CC setup routine. With both the I/O device selection, CC setup and fetch IPL CCW 1 routines complete, the 'address in' signal to the channel is AND'ed with the 'CCW valid' signal to activate the 'address in gated' signal. In turn, this signal is AND'ed with the 'operational in' signal to turn on the channel clock. The 'address in gated' signal is also significant in generating the 'sample bus-in' signal; this signal samples the unit address byte on the 'bus in' lines for correct (odd) parity. If a parity error is detected, the channel's 'interface control check' latch is turned on, and the IPL operation stopped; manual intervention is required to restart the IPL operation.

If no bus-in parity error is detected, the channel compares the unit address register contents with the unit address on the 'bus in' lines. If the addresses compare, the control unit has selected the I/O device requested by the channel; if not, an incorrect selection has been performed and the channel's 'interface control check' latch is turned on. With this latch on, the channel stops the IPL operation and manual intervention is required to restart the IPL operation.

If neither a 'bus in' parity error or address mismatch is detected by the channel, the I/O device selection, CC setup routine has been successfully completed. In other words, the correct control unit and I/O device have been connected to the channel via the I/O interface.

Command to Control Unit, CC Setup. A summary and a description of the operation follow:

- 'Setup' latch is turned off.
- Clock is turned on.
- 'Gate command' latch is turned on.
- Parity of read command byte is checked.
- Data address register bits 21-23 (DAB) are gated to byte counter.
- DAB plus count gated to adder and results gated back to count register.
- Reset DAB; correct data address register P2 parity, if necessary.
- 'Command out' signal is raised by channel.
- Control unit drops 'address in' signal.
- Channel drops 'command out' signal.

The command to control unit, CC setup routine (Diagram 5-20) is performed to send the read command in the command register to the control unit, and to perform operations with the count register value and a portion of the data address register value; the count and data address operations are required in order for the channel to control the IPL CCW 1 read operations.

The routine begins when the channel's 'setup' latch and clock are turned off by AND'ing the 'address compare' signal and the 'T4' clock signal. With the turn-off of the 'setup' latch, the 'gate command' latch is turned on to activate the 'read' signal. Following turn-on of the 'gate command' latch, the channel turns on the 'gate command to bus-out' latch which gates the read command byte to the 'bus-out' lines. The 'gate command to bus out' signal is also AND'ed with the 'address in' signal from the control unit to turn on the channel clock. With the read command byte on the 'bus out' lines, the channel generates a 100-ns 'sample bus out' signal with the 'T2' and 'not T3' clock signals. If the read command byte parity is incorrect (even), the 'channel control check' latch is turned on to stop the IPL operation; manual intervention is required to restart the operation. If not, the command to control unit, CC setup routine continues.

With the 'setup' latch off, and the clock on, the 'not setup', 'CC latch' and 'T4' clock signals are AND'ed to gate data address register bits 21 through 23 (the DAB) plus the count in the count register to the adder. At the same time the DAB is gated into the byte count register; with the DAB in the byte count register, the byte counter now specifies the B-register byte location into which the first program byte will be gated when the IPL CCW 1 read operations start. Subsequently, the 'T4' and 'not T6' clock signals are AND'ed to latch the adder and gate the DAB plus count value into the count register. The adjusted count register value permits the channel to determine when the last program byte of the IPL CCW 1 read operation is

presented to the channel, and thus determines the byte location in the B-register into which the last byte is gated. After the DAB has been gated to the byte count register and adder, the DAB bits in the data address register are reset. Just prior to resetting the DAB, the channel examines the DAB bits to determine whether the P2 parity bit in the data address register needs correcting due to the reset. If the status of the P2 latch is to change, the latch status is changed just prior to resetting the DAB.

Assuming that the read command byte has correct parity (bus-out parity error is not detected), the 'address in gated' signal and 'T7' clock signal are AND'ed to raise the 'command out' signal to the control unit. The 'command out' signal informs the control unit that the read command byte is on the 'bus out' lines.

When the control unit accepts the read command byte, it drops the 'address in' signal, causing the channel to drop the 'command out' signal. With the drop of the 'command out' signal, the command to control unit, CC setup routine is complete.

Status From Control Unit, CC Setup. A summary of the operation follows:

- Routine is same as status from control unit, initial setup routine. (See Diagram 5-20 and "Status From Control Unit, Initial Setup" in this chapter.)
- Status byte and 'status in' signal sent to channel from control unit.
- Channel samples for status byte parity error on 'bus in' lines.

Status Examination, CC Setup. A summary of the operation follows:

- Routine is same as status examination, initial setup routine. (See Diagram 5-20 and "Status Examination, Initial Setup" in this chapter.)
- Channel clock is turned on.
- Examine status byte on 'bus in' lines for all 0's. If all 0's, control unit and I/O device are ready to perform command; if not, interrupt status condition stops IPL operation.

Clear Initial Status, CC Setup. A summary of the operation follows:

- Routine is same as clear initial status, initial setup routine. (See Diagram 5-23 and "Clear Initial Status, Initial Setup" in this chapter.)
- Turn on 'sequence 2' latch, raise 'service out' signal, and turn off 'latch status byte' trigger.
- Control unit drops 'status in' signal in response to 'service out' signal.
- 'Service out' signal drops when 'status in' signal drops.
- Turn off 'sequence 1' latch; turn off clock.
- Clear initial status, CC setup routine is complete; this ends the IPL chain command to IPL CCW 1 operations and the channel can begin the IPL CCW 1 read operations.

Detailed IPL CCW 1 Read Operations

The IPL CCW1 read operations are performed by the channel to transfer program bytes specified by IPL CCW 1 from the control unit to main storage. Channel operations are as for a normal read operation. For details of the IPL CCW 1 read operations, refer to Diagram 5-20 and to "IPL CCW 1 Read Operations" and "Read Operation" in this Chapter.

Detailed Bootstrap Operations

The bootstrap operations are performed when IPL CCW 1 contains an active CC or CDA flag and IPL CCW 2 contains a TIC command. The operation is performed to branch the channel to a chain of CCW's in main storage which were stored there during the IPL CCW 1 read operations. For details of the bootstrap operation, refer to Diagram 5-20 and "Bootstrap Operation" in this Chapter.

Detailed IPL Ending Conditions

IPL ending conditions operations are performed by the channel to end the IPL operation. During the IPL ending conditions operations, the channel: (1) obtains a status byte (with the device end bit active) from the control unit; (2) sets up channel logic to store the channel and unit addresses in the IPL PSW main storage location 0; (3) stores the channel and unit addresses; and (4) releases the CPU and clears the channel IPL controls to end the IPL operation. The routines comprising the IPL ending conditions operations are described separately and in detail in the following paragraphs. Descriptions are based upon the detailed IPL flow chart (Diagram 5-20) and the simplified IPL control logic diagram (Diagram 5-21). For the relationship of major timing signals for the IPL ending conditions operations, refer to Diagram 5-24.

Device Status, IPL Ending. A summary and a description of the operation follow:

- Routine begins when read operation is completed for first CCW without active CC flag, CDA flag and TIC command.
- 'Sequence 5' latch is on.
- 'Status in' signal received from control unit.
- Channel clock turned on.
- Device end bit detected by channel.

The device status, IPL ending routine (Diagram 5-20) is performed when the read operations for the first CCW without a CC flag, CDA flag, and TIC command are completed by the channel. The routine is performed to obtain a status byte from the control unit with an active device end bit.

Assume that the 'sequence 5' latch is on (turned on at the end of the read operation). The device status, IPL ending routine begins when the control unit raises the 'status in' signal and places the status byte on the 'bus-in' lines. The 'status in' signal is AND'ed with the 'sequence 5' signal to turn on the channel clock.

If the 'status in' signal is received prior to the turn-on of the 'sequence 5' latch, the 'incorrect length' latch may be turned on. Normally, the suppress incorrect length indication (SILI) flag in the IPL CCW's is on and the channel is prevented from detecting an incorrect length condition. However, if the 'incorrect length' latch should be on, the channel performs a residual count operation and the IPL operation is stopped; manual intervention is required to restart the IPL operations. During a residual count operation, the channel computes the difference between the bytes requested by the count register value and the bytes actually received from the control unit; the difference (residual count) represents the number of bytes not supplied by the control unit. The residual count is computed by gating the two's complement of the value in byte counter, plus the count register value minus 8, to the adder. The adder is then latched and the residual count results are gated to the count register. Since the channel, during the IPL operation, is prevented from generating an 'interrupt request' signal, the channel stops IPL operations after computing the residual count.

Assuming that the 'incorrect length' latch is off, the channel examines the status of the 'control unit busy' latch (the latch is normally not on at this point in the IPL operation). If the 'control unit busy' latch is off, the channel raises the 'service out' signal and examines the status byte for the device end bit. If the device end bit is inactive (status byte may contain only the channel end bit), the control unit drops the 'status in' signal, causing the channel to drop the 'service out' signal. The channel then waits for the control unit to again place the status byte on the 'bus-in' lines and raise the 'status in' signal. The channel responds by raising the 'service out' signal and again examining the status byte for an active device end bit ('bus-in bit 5' signal). Assuming the device end bit is detected, the device status, IPL ending routine is complete.

IPL End Condition Setup. A summary and a description of the operation follow:

- Routine disconnects control unit; conditions channel logic to store channel and unit addresses; initiates storage cycle to store channel and unit address.
- Turn off 'select out' latch.
- Activate 'turn on storage request IPL' signal.
- Control unit drops 'operational in' and 'status in' signals; channel drops 'service out' signal.
- Turn on 'storage request' latch and 'storage cycle' trigger.
- Reset data address register (address 0).
- Turn on 'IPL and condition' latch.
- Turn on 'block Z-address' latch.
- Turn on 'remember IPL end' latch.
- Inhibit gating A-register bytes 2 and 3 to SBI.
- Receive 'BCU response' signal:
  1. Gate 'store' signal to BCU interface.
  2. Gate data address register value (0) to SAB.
  3. Force mark bits 2 and 3 plus P to mark bus.
  4. Force storage protect parity bit to 'storage protection' lines; block all other bits (master storage protect key).
- Receive 'BCU data request' signal (channel generates signal for Model 91) from BCU interface.
- Gate channel address bits 0-2 to SBI lines 21-23, unit address register bits 0-7 to SBI lines 24-31.

The IPL end condition setup routine (Diagram 5-20) is performed to disconnect the control unit from the I/O interface, condition channel logic to store the channel and unit addresses in byte locations 2 and 3 of the IPL

PSW (storage address 0), and initiate the storage cycle to store the addresses. The channel address is wired into channel logic, and the unit address of the I/O device involved in the IPL operation is contained in the unit address register.

The IPL end condition setup routine begins with the turn off of the 'select out' latch. This latch is turned off by AND'ing the 'T4' and 'not T5' clock signals with the 'not CC latch' and 'status in end' latches to activate the 'turn off select out end' signal. When the 'select out' signal falls at the control unit, the control unit responds by dropping the 'status in' and 'operational in' signals. When these signals are dropped, the control unit is electrically disconnected from the channel I/O interface. On the fall of the 'status in' signal, the channel drops the 'service out' signal and resets the 'status in end' latch.

While the 'turn off select out end' signal is still active, it is AND'ed with the 'read' signal to activate the 'turn on storage request IPL' signal (see Diagram 5-21). The 'turn on storage request IPL' signal performs three functions: (1) resets the data address register to 0's with good parity (address 0 is required to store the channel and unit addresses); (2) turns on the 'IPL end condition' latch (Diagram 5-21); and (3) turns on the 'storage request' latch.

Turn-on of the 'storage request' latch raises the 'storage request' signal to the BCU interface, turns on the 'storage cycle' trigger, and turns on the 'block Z-address' latch (Diagram 5-20). The 'storage cycle' signal activates and deactivates applicable channel logic for the duration of the requested storage cycle. The 'block Z-address' signal prevents the channel wired Z-address (64 decimal) from being gated to the SAB.

Meanwhile, the 'IPL end condition' signal turns on the 'remember IPL end' latch and inhibits the gating of A-register bytes 2 and 3 (bits 16 through 31 plus P2 and P3) to the SBI lines. The 'remember IPL end' signal activates one input to the IPL complete AND gate. The A-register bytes are inhibited from SBI byte positions 2 and 3 becaus the channel and unit address bits are to be gated to these SBI byte positions.

The channel now waits for a 'BCU response' signal from the BCU interface. When the signal is received, it gates the 'store' signal to the BCU interface and the data address register contents (address 0) to the SAB. In addition, the 'BCU response' signal is AND'ed with the 'IPL end condition' signal to: (1) force mark bits 2 and 3 plus the P bit to the mark bus (in main storage the two mark bits specify that data is to be stored only in byte locations 2 and 3); and (2) force the storage protect parity bit to the 'storage protection' lines. With the 'IPL end condition' signal inhibiting all other storage protect bits, a master storage protect key of all 0's with good parity is on the 'storage protection' lines. In main storage, the master key allows access to any storage location. With the 'store' signal active and the address information on the appropriate buses, the channel waits for 'BCU data request' signal from the BCU interface. (For the System/360, Model 91, the channel generates the 'BCU data request' signal approximately 150 ns after the 'BCU response' signal rises.) When received, the 'BCU data request' signal gates channel address bits 0-2 (wired into the channel to indicate the address of the channel) to SBI lines 21 through 23 and gates unit address register bits 0 through 7 to SBI lines 24 through 31. With the channel and unit addresses on the SBI lines, the IPL end condition setup routine is complete.

Store Channel and Unit Addresses, IPL End. A summary and a description of the operation follow:

- Routine performed to monitor storage of channel and unit addresses.
- 'Address valid' latch turned on.
- 'BCU response' signal falls.
- Turn on 'remember BCU response' latch.
- 'Accept' signal received from BCU interface.
- 'Accept' latch is turned on; 'storage request' latch is turned off.
- 'BCU advance' signal received from BCU interface.
- Gate 'invalid address check' and 'storage protect check' lines; stop IPL program if check condition present.
- Turn off 'remember BCU response' and 'accept' latches.
- Turn off 'storage cycle' trigger.

The store channel and unit address, IPL end routine (Diagram 5-20) is performed by the channel to monitor the progress of the storage cycle initiated during the IPL end condition setup routine. When the store channel and unit addresses, IPL end routine is successfully completed, the channel and unit addresses have been stored in the interrupt portion (byte locations 2 and 3) of the IPL PSW at main storage address 0.

The store channel and unit addresses, IPL end routine begins when the 'address valid' latch is turned on. This latch is turned on by delaying the 'BCU response' signal approximately 150 ns. With the 'address valid' latch on, the 'address valid' signal to the BCU interface is raised indicating that the address information has stabilized on the BCU interface buses.

The channel waits for the 'BCU response' signal to fall; when the signal falls, the 'remember BCU response' latch is turned on. The channel then waits for an 'accept' signal from the BCU interface, indicating that main storage has started the requested storage cycle. When received, the 'accept' signal turns on the 'accept' latch and turns off the 'storage request' latch.

Subsequently, the channel receives the 'BCU advance' signal. Upon receipt of the 'BCU advance' signal, the channel delays the signal to form an 'raw advance SS' signal, an 'advance pulse' signal, a 'late advance pulse' signal, and a 'storage cycle complete' signal. Each of these signals is progressively delayed, and the delays are different in channels used with different systems. (See FEMM, 2860 Selector Channel, for specific delays.)

The delayed 'raw advance SS' signal gates the 'invalid address check' and 'storage protect check' lines from the BCU interface. If either of the check signals is active, the storage cycle operation is unsuccessful, and the channel stops the IPL operation; manual intervention is required to restart the IPL operation.

The delayed 'advance pulse' signal turns off the 'remember BCU response' and 'accept' latches. Subsequently, the delayed 'late advance pulse' signal turns off the 'storage cycle' trigger. Assuming that neither an invalid address check nor storage protect check condition is detected, the store channel and unit address, IPL end routine is successfully completed.

End IPL Operation. A summary and a description of the operation follow:

- Routine releases CPU and clears channel IPL control logic.
- Turn off 'IPL condition end' latch.
- Activate 'IPL complete' signal.
- Turn off 'sequence 5' latch.
- Gate 'release' signal to CPU interface.
- Inhibit generation of 'start I/O reset' signal.
- 'Select channel' signal drops.
- Turn off 'CCW valid' latch.
- Turn off 'IPL' latch.
- Turn off 'block release IPL' latch.
- IPL operation complete; channel resumes polling and available to receive I/O instructions.

The end IPL operation routine (Diagram 5-20) is performed to generate a 'release' signal to the CPU interface and to clear the IPL control logic. When the routine is completed, the channel resumes polling operations and is free to accept I/O instructions.

The end IPL operation begins when the 'late advance pulse' and 'storage cycle complete' signals are AND'ed to turn off the 'IPL condition end' latch (see Diagram 5-21). Subsequently, the 'IPL complete' signal is activated by AND'ing the 'not IPL end condition', 'not operational-in or status-in', 'not CC latch', 'not interrupt status', and 'remember IPL end condition' signals.

The 'IPL complete' signal performs three parallel operations: (1) turns off the 'sequence 5' latch; (2) gates a 'release' signal to the CPU interface; and (3) blocks generation of the 'start I/O reset' signal. When the CPU receives the 'release' signal, it drops the 'select channel' signal to the channel. In the channel, the 'not select channel' and 'IPL complete' signals are AND'ed to turn off the 'CCW valid' and 'IPL' latches. The 'not select channel' signal also turns off the 'block release IPL' latch (see Diagram 5-15). With the preceding operations complete, the end IPL operation routine is complete. This completes the IPL operation, freeing the channel to resume polling operations and to receive I/O instructions which may be initiated by the CPU.

FAULT LOCATING TEST OPERATIONS

- FLT operations cause channel to read FLT tests to storage; FLT tests are used to test CPU logic, not channel logic.
- FLT operations initiated by CPU.
- Channel IPL logic loads record-1 doublewords into positions 0, 8 and 10 (hex) in main storage.
- IPL logic controls loading of CCW's (record 2) which channel uses to control storage of FLT tests.
- Test blocks are alternately loaded into storage areas designated buffer 1 and buffer 2.
- Channel FLT logic and record 2 CCW's control storage of FLT tests.
- When either buffer is filled, channel sends 'TIC pulse' signal to CPU; CPU may then use tests in that buffer.
- Channel controls loading of alternate buffers by fetching read CDA CCW, controlling transfer of data to buffer, then fetching TIC CCW; TIC CCW points to CCW which channel fetches to control loading of alternate buffer.

- Inter-block gap detected by channel; causes channel to read next test block to buffer 1.

- Unit check condition detected by channel; causes channel to send command to I/O device to reread into storage buffers.

The channel performs FLT operations under CPU control to load FLT tests into main storage. The FLT tests are then used by the CPU to perform FLT tests on CPU logic; FLT tests are not used to test channel logic. In other words, channel FLT operations consist of reading FLT information into storage from an I/O device specified by a unit address received from the CPU interface. Channel FLT control logic enables the channel to communicate to the CPU certain events or check conditions detected by the channel during FLT operations. These events and check conditions enable the CPU to utilize the FLT tests read into storage by the channel or to determine whether the FLT test information is valid for testing purposes. The channel's FLT control logic also enables the channel to respond to FLT control signals from the CPU interface. In general, the FLT control signals allow the CPU to stop and start channel read operations; the stop/start controls cause the channel to perform operations designed to reread a block of tests into storage when the channel or CPU determines that an error was detected when the block was originally read into storage.

FLT tests may be read into storage from a tape, drum or disk. Although the FLT program for each device may be different in certain areas, certain areas of the program are the same for all devices; this is necessary, since a portion of the FLT operation, once started by the CPU, is automatically controlled by the channel's IPL and FLT control logic. Figure 3-8 shows a typical FLT test on tape. Note that records 1 and 2 in the figure contain similar information to that of an IPL tape; in fact, the FLT record 1 and 2 information is read into storage under control of the channel's IPL control logic. The test records shown in Figure 3-8 are then subsequently read into main storage under control of CCW's from record 2, the channel's FLT control logic, and the CPU's FLT control logic.

When the reading of FLT tests begins, the channel alternately stores tests into two areas of storage designated buffer 1 and buffer 2. The starting addresses of the buffer 1 and buffer 2 areas are contained in the CCW's loaded from record 2 and are different for different Models of the System/360. For example, the starting addresses for buffers 1 and 2 in the System/360, Model 2065 are 8000 (hex) and 8080 (hex), respectively; in the Model 2075, 20,000 (hex) and 20,100 (hex), respectively. When either buffer 1 or 2 is filled, the channel sends a 'TIC pulse' signal to the CPU, indicating that the tests in the buffer may be used by the CPU when the CPU finishes with the tests in the alternate buffer.

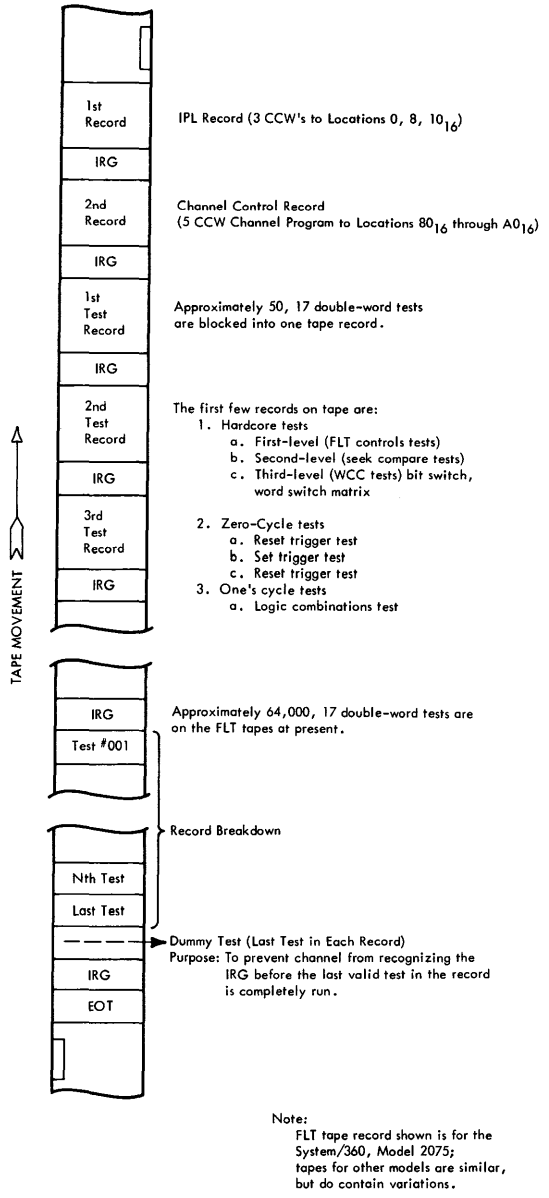
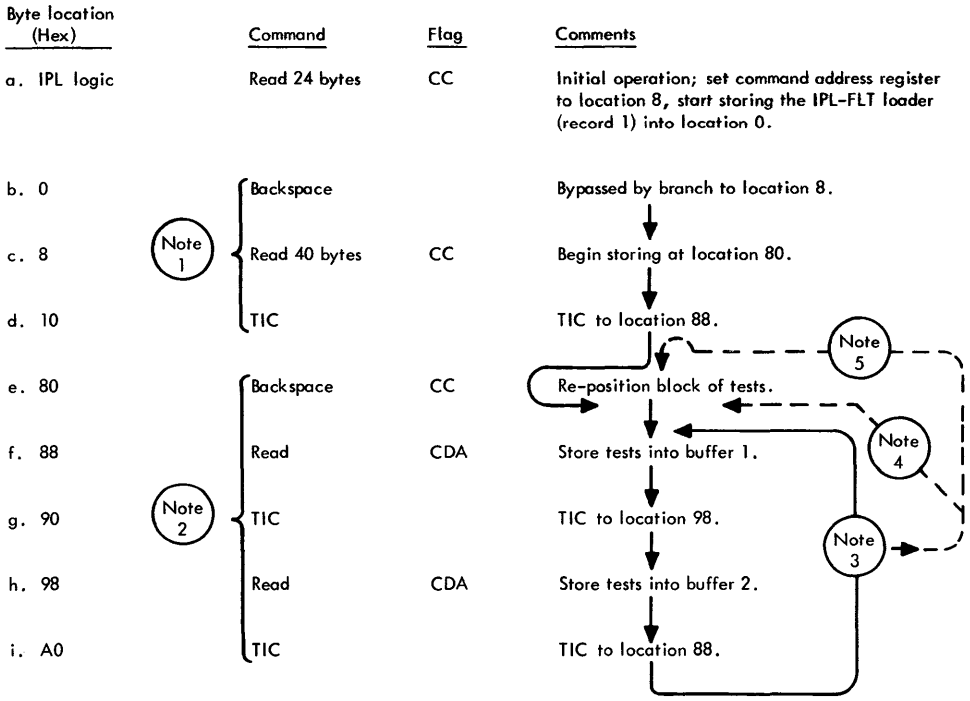


Figure 3-8. Typical FLT Tape Record

Unless stopped by a detected error or a CPU programmed stop, the channel continuously reads blocks of tests into the buffer 1 and buffer 2 storage areas. Refer to Figure 3-9. (This represents a typical FLT operation from a tape unit.) Note commands b. through i. on Figure 3-9. These commands are read into storage under control of the channel's IPL logic. Once the CCW's are in storage locations 80 through A0 (hex), the channel controls the storage of FLT tests into buffer 1 after fetching the CCW from storage location 88 (hex). When buffer 1



Note 1: These three commands are the IPL-FLT loader (record 1), which is used to get the FLT loader (record 2) into storage.

Note 2: These five commands are the FLT loader (record 2), which loads the test records into storage as required and provides a repeat-test-on-error feature.

Note 3: This loop is continuous until the end of a block is reached during the read operation. When the end of the block is sensed, the operation branches to note 4 or note 5.

Note 4: The tape unit providing the FLT reaches an inter-block gap and no unit-check condition exists at the device. A gap pulse is sent to the CPU. Channel branches to location 88 and begins storing into buffer 1 regardless of last buffer in which FLT tests were stored.

Note 5: The tape unit providing the FLT reaches an inter-block gap and a unit-check condition does exist at the device or the device found some condition that ended data transfer. No gap pulse is sent to the CPU. Channel branches to location 80 causing the I/O device to go back to the beginning of the block of tests; channel then controls rereading of the block of tests into buffers 1 and 2.

Figure 3-9. Typical FLT Operation from Tape

is full, the channel performs a read CDA operation and fetches the TIC CCW from storage location 90 (hex). The channel then sends the CPU a 'TIC pulse' signal indicating that buffer 1 is full and performs a TIC operation to fetch the CCW from location 98. After receipt of the CCW, the channel controls the storage of FLT tests into the buffer 2 storage area. When buffer 2 is full, the channel performs a read CDA operation and fetches the TIC CCW from storage location A0. The channel then sends a 'TIC pulse' signal to the CPU and performs a TIC operation to fetch the read CDA CCW from location 88. The channel continues to control the loading of alternate buffers by the CCW looping process described above until all tests are loaded.

Although the number of read CDA and sequential TIC CCW's (f. through i, Figure 3-9) may differ for a given system or I/O device, the principle by which the channel controls loading of alternate buffers is the same.

When an inter-block gap (inter-record gap or IRG in Figure 3-8) is crossed at the I/O device, the channel detects the gap and sends a 'gap pulse' signal to the CPU. This causes the CPU to send a 'stop FLT' signal followed (after a delay) by a 'start FLT' signal. Regardless of the buffer in which the channel was previously storing data, the stop FLT/start FLT sequence causes the channel to fetch the CCW from location 88 and control the loading of the next block of FLT tests into buffer 1 (see Note 4, Figure 3-9).

If the channel detects a unit check condition at the end of an FLT test record, it sends an 'FLT data error' signal to the CPU and fetches the CCW from storage location 80. This CCW contains a command to cause the I/O device to return to the beginning of the record containing the test block with the chain check condition. Subsequently, the channel controls the rereading of the FLT tests into main storage (see Note 5, Figure 3-9).

If the channel detects a control check condition, channel FLT operations are terminated, and the channel sends an 'FLT control check' signal to the CPU; in this case, manual intervention is required to clear the control check condition and restart the FLT operation.

Channel FLT operations are described in the following paragraphs at the simplified and detailed levels. The simplified description is based upon the flow chart in Diagram 5-26. The detailed description is based upon Diagrams 5-20 and 5-27 (detailed flow charts), and Diagram 5-28 (FLT control logic).

Simplified FLT Operations

FLT operations are initiated at the system console (for a description of FLT operation initiation, refer to the appropriate System/360 Model FEMM). When initiated, the channnel receives 'IPL', 'FLT mode', and 'select channel' signals from the CPU interface (Diagram 5-26). In addition, the unit address of the I/O device containing the FLT program is supplied to the channel on the 'unit address bus-out' lines. The 'FLT mode' signal turns on the 'FLT mode' latch to activate the channel's FLT control logic. As during a normal IPL operation, the 'IPL' signal initiates interface reset and machine reset operations to reset the channel logic and attached I/O devices. The IPL logic then forces a CCW in the channel which specifies read 24 bytes into main storage beginning at address 0 and chain command to the CCW at storage location 8.

An initial selection routine is performed by the channel to select the I/O device designated by the unit address. Following successful selection of the device, the channel controls the reading of 24 bytes (record 1) from the device into main storage beginning at address 0. When the 24 bytes (three doublewords) have been stored, the channel performs a chain command operation to disconnect the I/O device, fetch the CCW from storage location 8, and reselect the I/O device. The CCW from storage location 8 specifies a read operation and a subsequent chain command operation to fetch the TIC-CCW at storage location 10 (hex). The bytes to be read (record 2) constitute the CCW's which the channel will use to read FLT tests into main storage. The number of bytes to be read depends upon the FLT format, which is determined primarily by the I/O device from which the FLT tests are to be read. (For example, record 2 from a tape unit usually consists of 40 bytes.) Record 2 bytes are stored in main storage beginning at address location 80 (hex).

After the channel has controlled the transfer of the record 2 bytes to main storage, the channel performs a chain command operation to fetch TIC-CCW 1 from storage location 10 hex (16 decimal). Upon receipt of the CCW, the channel turns off the IPL latch and sends a 'release' signal to the CPU interface.

The channel then performs a TIC operation to fetch the CCW from main storage location 88 (hex). This CCW specifies that a block of FLT tests is to be read into the buffer 1 location in main storage; the CCW also contains an active CDA flag. The buffer 1 storage area is dependent upon the System/360 Model to which the channel is attached. [For example, in the Model 2065, the buffer 1 storage location begins at address 8000 (hex); in the Model 2075, at address 20,000 (hex).]

With the CCW from location 88 established in the channel, the channel controls the transfer of the specified number of bytes into buffer 1, then performs a chain data operation to fetch the next sequential CCW from main storage location 90 (hex). The CCW from location 90 (hex) is a TIC-CCW. When the TIC command (with the 'chain data' latch on) is decoded at the channel, the channel sends a 'TIC pulse' signal to the CPU informing the CPU that buffer 1 is full.

The channel then performs a TIC operation to fetch the next sequential CCW from address location 98 (hex). This CCW specifies a read CDA operation to load FLT tests into the main storage buffer 2 area. (Like the buffer 1 area, the buffer 2 storage location is System/360 Model oriented; e.g., for the 2065, the buffer 2 starting location is address 8080 hex and for the 2075, address 20,100 hex.) The channel controls the storage of the test bytes in the buffer 2 location, then performs a read CDA operation to fetch the next sequential TIC CCW from storage location A0 (hex). When the TIC CCW is received by the channel, the channel again sends the CPU a 'TIC pulse' signal, indicating that buffer 2 is full and that the CPU may utilize the tests in that buffer. The channel then fetches a read CDA CCW either from address location 88 (hex) or from sequential address A8, depending upon the number of CCW's read into storage during the record 2 read operations. Assume the TIC CCW from location A0 specifies that the channel is to fetch the read CDA CCW from location 88 (hex). When the channel receives the CCW, it controls the transfer of the next block of FLT tests into buffer 1 in main storage. Assuming no errors are detected during the FLT operations, the channel continues to alternately load FLT test blocks into buffers 1 and 2 as described in the preceding text. The channel continues to load FLT tests until all tests in the FLT program are loaded. When test loading is complete, the 'FLT mode' signal to the channel is dropped, and the channel is free to perform other I/O operations.

During the FLT operation (Diagram 5-26), the channel detects each inter-record gap on the I/O device program. If no unit check condition is detected in the status byte from the I/O device, the channel sends a 'gap pulse' signal to the CPU; this indicates that the next FLT test block will be loaded into buffer 1, regardless of whether buffer 1 or 2 was being loaded when the gap was detected. The channel then fetches the CCW from address location 88 (hex) and controls the reading of the first FLT test block from the next record into buffer 1. From this point, alternate loading of buffers 1 and 2 is controlled by the channel as previously described.

If a unit check condition exists when the inter-record gap is detected (Diagram 5-26), the channel inhibits generation of the 'gap pulse' signal, and sends the CPU an 'FLT data check' signal. The channel then automatically resets the command address register, turns on the 'chain command' latch, and forces address 80 (hex) into the command address register. With the 'chain command' latch on, the channel fetches CCW 80 (hex) and then sends a return to beginning of record command to the control unit. The I/O device returns to the beginning of the record, and the channel performs another chain command operation to fetch the read CDA CCW from address 88 (hex). From this point, alternate loading of buffers 1 and 2 is controlled by the channel as previously described. If the unit check condition is continually detected, the channel automatically retrys to read the FLT test record into the buffers, as described, a specified number of times (e.g., 32 times when servicing the System/360 Model 2065 and 64 times when servicing the Model 2075). If the specified number of retrys is not successful, channel FLT operations are stopped by the 'stop FLT' signal from the CPU and manual intervention is required.

The CPU, upon recognition of error conditions or a programmed stop condition, can force the channels to control record retrys. If the CPU recognizes a check condition (for example, a channel address check or invalid address check) or a programmed stop condition, the channel receives a 'stop FLT' signal from the CPU. This causes the channel to enter a sequence 5 routine to disconnect the control unit from the I/O interface. The disconnect operation causes the I/O device to stop at the end of the record in which the chain check condition was detected. The channel then forces command address 80 (hex) into the command address register and awaits a 'start FLT' signal from the CPU. Upon receipt of the 'start FLT' signal, the channel performs a chain command operation to reselect the I/O device and fetch the CCW from address location 80 (hex). The fetched CCW rereads the FLT tests from the record into buffers 1 and 2 storage areas as previously described. If a unit check condition (or CPU detected error) is again detected, the channel again performs the operation just described. Depending upon the System/360 Model to which the channel is attached, the stop FLT/start FLT operation will be performed a number of times (e.g., 32 times for the Model 2065; 64 times for the Model 2075) in an attempt to read the FLT tests from the record without error. If all retrys are unsuccessful, the FLT operations are stopped and the CPU indicates a unit data check condition.

If the channel should detect an interface control check, channel control check, storage protect check, program check, chain check, or interrupt condition while the 'start FLT' latch is on, the channel sends the CPU an 'FLT control check' signal and stops the FLT operation. Upon receipt of the 'FLT control check' signal, the CPU stops FLT testing and manual intervention is required to restart the FLT operation.

Detailed FLT Operations

Since the first portion of the FLT operation is identical to the IPL operation, reference is made to Diagram 5-20 and the text describing the detailed IPL portion of the FLT operation. For a detailed description of the IPL portion of the FLT operation, refer to the following paragraphs: "Detailed IPL Initial Selection"; "Detailed IPL Loader Read Operations" (record 1 for FLT operation); "Detailed IPL chain command to IPL CCW 1"; and "Detailed IPL CCW 1 read operations" (record 2 for FLT operation). The operations described by these paragraphs are performed by the channel upon receipt of the 'IPL' and 'select channel' signals along with the unit address byte. On Diagram 5-20 (channel forces CCW), note that an active 'FLT mode' signal references Diagram 5-27, and that at the completion of the IPL CCW 1 read operations Diagram 5-27 is again referenced. In the following paragraphs detailed descriptions of the FLT operations are limited to the operations shown in flow chart form in Diagram 5-27. References are also made to the FLT control logic shown in positive logic form in Diagram 5-28, FEDM and to IPL operations performed concurrent with the operations shown on Diagram 5-20.

FLT Control Initialization

- 'FLT mode' signal turns on 'FLT' latch.
  1. Channel's FLT control logic enabled.
  2. 'Stop' signal blocked to prevent logout operation.
  3. 'PCI or log wait interrupt' signal blocked; prevents PCI or logout routine.
  4. 'Interrupt request' signal due to incorrect length detection blocked.

- Channel forces read 24 bytes, address 0, CC, command address register equals eight CCW; selects I/O device with FLT program.

- Channel controls transfer of FLT record 1 (three doublewords) to storage beginning at address 0.
- Channel performs chain command operation to fetch CCW from location 8.
- Channel controls reading of FLT record 2 (CCW's) beginning at storage location 80.
- Channel performs chain command operation to fetch TIC CCW from storage location 16 (10 hex).

When the channel receives the 'FLT mode' signal (Diagram 5-20), the 'FLT mode' latch is turned on (Diagram 5-27). With the 'FLT mode' latch on, the 'FLT control' logic (Diagram 5-28) is enabled. In addition, the 'FLT mode' signal performs the following functions throughout the FLT operation in the channel's operating logic (see Diagram 5-27): (1) generation of the 'stop' signal is blocked to prevent the channel from attempting a logout operation if the LOG ON MACH CHK switch is on and an interface control check or channel control check condition is detected; (2) generation of the 'PCI or log wait interrupt' signal is blocked to prevent the channel from attempting a PCI or logout interrupt routine; and (3) the 'incorrect length' signal is prevented from causing the channel to send an 'interrupt request' signal to the CPU interface (detection of an incorrect length condition is handled by the FLT control logic). Concurrent with the FLT control logic initialization operations, IPL initial selection operations are performed to select the I/O device specified by the unit address, and to force a CCW into the channel. (Refer to "Detailed IPL Initial Selection".) Like the IPL operation, the channel forced CCW specifies read 24 bytes into main storage beginning at address 0, then chain command to the CCW stored at address location 8. (For the FLT operation, the three doublewords read to locations 0, 8 and 16 decimal are referred to as record 1.) Following the read operations (refer to "Detailed IPL Loader Read Operations"), the channel performs a chain command operation to fetch the CCW from storage location 8 (refer to "Detailed IPL Chain Command to IPL CCW1"). The CCW in location 8 specifies read FLT record 2, consisting of a specified number of CCW's used by the channel to control the loading of FLT tests, beginning at storage address location 80; the CCW also specifies a chain command operation to fetch the TIC CCW from address 8 after FLT record 2 has been read to storage. Upon completion of the read operation to load FLT record 2 (refer to "Detailed IPL CCW 1 Read Operations"), the channel performs another chain command operation to fetch the TIC CCW from storage location 16 decimal (10 hex). When the TIC CCW is received by the channel, operations are performed to release the CPU and terminate the IPL portion of the FLT operation.

Release CPU

- Channel decodes TIC command; turns on 'release FLT' signal.
- 'Release' signal sent to CPU interface.
- 'Select channel' signal from CPU interface drops.
- 'IPL' latch turns off.
- Channel performs TIC operation to fetch read CDA CCW from storage location 88 (hex); used to fill buffer 1.

Upon receipt of the TIC CCW from storage location 16 (decimal), the channel decodes the TIC command (Figure 5-27). The decoded TIC command activates the 'turn on release FLT load' signal (Diagram 5-28) which turns on the 'release FLT' latch. The resulting 'release FLT program load' signal activates the 'release' signal to the CPU interface. Upon receipt of the 'release' signal, the CPU drops the 'select channel' signal (Diagram 5-27), causing the IPL latch to turn off. The subsequent channel FLT operations performed to load the actual FLT tests are now controlled by the CCW's loaded by the FLT record 2 read operation, the channel's FLT control logic, and the FLT control signals from the CPU interface.

The TIC-CCW from storage location 16 (decimal) also causes the channel to perform a TIC operation to fetch the read CDA CCW from storage location 80 (hex). This CCW specifies that the channel is to control the loading of a block of FLT tests into buffer 1 in main storage. The channel continues to control the loading of buffers 1 and 2, alternately, as described in "Simplified FLT Operations". During the course of loading the buffers, the channel monitors the FLT test transfers for errors, for an incorrect length condition, and for inter-record gaps. These monitoring operations and subsequent channel operations are described in the following paragraphs.

FLT Error Detection

- FLT operations stopped by channel when interface control check, channel control check, storage protect check, chain check, program check, or interrupt status condition detected.

- Channel sends CPU 'FLT control check' signal.

- FLT operations altered when channel detects unit check condition.

- Channel sends CPU 'FLT data check' signal; CPU sends channel 'stop FLT' followed by 'start FLT' signal.

- 'Stop/start FLT' signals cause channel to control retry of FLT record in which unit check condition was detected.

While the channel is alternately loading buffers 1 and 2, FLT control logic (Diagram 5-28) monitors channel error detection logic for errors which may alter or stop the FLT operation. Conditions which cause the channel to stop FLT operations are: (1) detection of an interface control check, channel check, storage protect check, chain check, or program check condition; or (2) an interrupt status condition. Any of the check conditions activates the 'command reject or control check' signal (Diagram 5-27). If either the 'interrupt status' or 'command reject or control check' signals are activated, the channel stops the FLT operation and the FLT logic activates the 'FLT control error' (FLT control check) signal to the CPU. The 'FLT control check' signal informs the CPU that the channel has terminated the FLT operation under error conditions.

FLT operations are altered when the channel FLT control logic detects a unit check condition in a status byte from the control unit (Diagram 5-27). When the unit check condition is detected, the 'data in bus bit 6', 'FLT mode', and 'latch status byte' signals are AND'ed (Diagram 5-28) to activate the 'FLT data error' ('FLT data check') signal to the CPU. This signal informs the CPU that the unit check condition has been detected and may cause the CPU to issue a 'stop FLT' signal followed by a 'start FLT' signal. These signals cause the channel to fetch a CCW with a command that causes the I/O device to return to the beginning of the record in which the unit check condition was detected (refer to "Stop/Start FLT").

Incorrect Length and Inter-record Gap

- Incorrect length condition in absence of unit check condition causes channel to send 'gap pulse' signal to CPU.

- 'Gap pulse' signal indicates error-free FLT tests from record have been read to main storage buffers.

- Channel forces address bits 16 and 20 on in command address register (address 88 hex).

- Channel fetches CCW (chain command operation) from address 88 (hex) and begins loading FLT tests from next record into main storage buffer 1.

- Detection of unit check status causes channel to automatically:
  1. Send 'FLT data check' signal to CPU.
  2. Reset command address register.
  3. Turn on 'chain command' latch.
  4. Force address 80 into command address register.
  5. Fetch CCW from address 80 (chain command operation) causing I/O device to return to beginning of record with unit check condition.
  6. Retry loading FLT tests into buffers 1 and 2.

- Continuous detection of unit check condition causes automatic retrys a specified number of times; if retrys are unsuccessful, FLT tests are stopped.

While the channel is loading FLT tests into the main storage buffers, the channel's FLT control logic is monitoring for an incorrect length condition and an inter-record gap condition. Since an incorrect length condition is indicated at the end of each FLT record, the absence of an incorrect length indication permits the channel to continue reading test bytes into main storage buffers 1 and 2, alternately (Diagram 5-27).

When an incorrect length condition is detected, the channel's FLT control logic activates the 'scan not IPL' signal. This signal blocks activation of the 'interrupt status' signal in the event a unit check condition is indicated in the status byte from the control unit. Preventing activation of the 'interrupt status' signal, in turn, prevents the channel from activating the 'interrupt request' signal to the CPU interface. With the 'incorrect length' signal active, the FLT control logic monitors for an active unit check bit in the status byte from the control unit. If the unit check bit is inactive, the channel activates the 'gap pulse' signal to the CPU interface by AND'ing the 'not stop FLT', 'incorrect length' and 'not unit check' signals.

Note: At the time the 'gap pulse' signal is sent, the channel is in a sequence 5 routine.

When the 'gap pulse' signal is sent to the CPU, the channel's FLT control logic AND's the 'gap pulse', 'sequence 5 status in', and 'not IPL' signals. This activates the 'force command address register bit 16' and 'force command address bit 20' signals, causing address 80 (hex) to be entered into the command address register. Subsequently (unless a 'stop FLT' signal is received from the CPU), the 'sequence 2', 'status in end' and 'FLT mode' signals are AND'ed to activate the 'turn on chain command latch' signal. This signal turns on the 'chain command'

latch, allowing the channel to fetch the read CDA CCW from storage location 88. Recall that this CCW specifies read FLT tests into buffer 1. Thus, detection of an inter-record gap in the absence of a unit check condition causes the channel to begin loading FLT tests from the next record into buffer 1, regardless of which buffer was being loaded when the inter-record gap was detected.

The presence of a unit check condition causes the channel to send the CPU an 'FLT data check' signal and subsequently causes activation of the 'reset command address register' signal. This signal resets the command address register to 0. When the 'status in end' signal is activated, the 'chain command' latch is turned on and the 'force command address bit 16' signal is activated. This forces address 80 into the command address register. (Recall that the CCW at address 80 contains a command which causes the I/O device to return to the beginning of the record in which the unit check condition was detected.) With the channel performing a chain command operation, the CCW at address 80 is fetched, the record on the I/O device returned to the start of the record, and the tests on the record reread alternately into buffers 1 and 2. If the unit check condition is continuously detected, the above operations are repeated a specific number of times (depending upon the System/360 Model to which the channel is attached). If the specified number of retrys to complete the test blocks on the record are unsuccessful, automatic retrys (under channel control) terminate and manual intervention is required to determine the cause. Test retrys may also be initiated by an operator at the system console or by the FLT control logic in the CPU. For a description of channel operations under CPU control, refer to "Stop/Start FLT".

Stop/Start FLT

- 'TIC pulse' signal sent to CPU each time channel fills buffer 1 or 2 and fetches next TIC CCW; signal indicates buffer has been filled with FLT tests.
- CPU detected error or test stop condition causes channel to receive 'stop FLT' signal; 'stop FLT' latch is turned on.
- Channel enters sequence 5 routine to disconnect control unit from I/O interface; I/O device stops at end of current record.
- 'Status in end' signal turns on 'chain command' latch.
- 'Start FLT' signal from CPU turns on 'start FLT' latch; 'stop FLT' latch turns off.
- Address 80 (hex) forced into command address register.
- 'Turn on setup chain command' signal is activated.
- Channel performs chain command operation to reselect I/O device and fetch CCW from address 80.
- Command in CCW (address 80 hex) sent to I/O device; device returns to beginning of record.
- Channel alternately loads buffers 1 and 2, beginning with buffer 1 per CCW at address 80 (hex).
- FLT operation ends when CPU drops 'FLT mode' signal.

The stop/start FLT operations permit the CPU FLT control logic to control retrys of test records and permits manual control from the system console of FLT test progression.

Assume that the 'chain data' latch is on, and that the 'TIC SS' signal to the CPU interface is active (Diagram 5-27). This means that the channel has just controlled the storage of FLT tests to fill buffer 1 or 2. The two signals are AND'ed with the 'FLT mode' signal to activate the 'TIC pulse' signal to the CPU. This signal informs the CPU that the alternate buffer (the buffer not currently being used by the CPU) is full and available for use in FLT testing. Assuming the channel does not receive a 'stop FLT' signal from the CPU, the channel continues loading alternate buffer areas with FLT tests, each time sending a 'TIC pulse' signal to the CPU when a buffer is filled.

The CPU may send the channel a 'stop FLT' signal (Diagram 5-27) as the result of an error condition or a programmed test stop condition recognized by the CPU's FLT control logic. Upon receipt of a 'stop FLT' signal by the channel, the signal is AND'ed with the 'operational in' signal to turn on the 'stop FLT' latch. The resulting 'stop FLT' signal blocks the 'turn on setup chain command' signal to prevent the channel from performing a chain command operation. At the same time, the 'stop FLT' signal is AND'ed with the 'sequence 2' and 'read' signals to force the channel into a sequence 5 routine. During the sequence 5 routine, the channel performs an I/O interface disconnect operation, causing the I/O device to stop at the end of the current FLT record. When the 'status in end' signal is activated during the sequence 5 routine, the 'FLT mode', 'sequence 2', and 'status in end' signals are AND'ed to turn on the 'chain command' latch. This prepares the channel to perform a chain command operation when a 'start FLT' signal is received from the CPU. The channel then remains in a stopped condition until a 'start FLT' signal is received from the CPU. (The 'start FLT' signal

may be activated automatically by the CPU FLT control logic or manually at the system console, depending upon the action which activated the 'stop FLT' signal.) Upon receipt of the 'start FLT' signal, the signal is AND'ed with the 'stop FLT' signal to turn on the 'start FLT' latch. In turn, the 'start FLT latch' signal is AND'ed with the 'not operational in' signal to turn off the 'stop FLT' latch.

With the 'start FLT' latch on, the 'start FLT', 'not command reject write on control', and 'FLT mode and not IPL' signals are AND'ed to activate the 'force command address bit 16' signal and turn off the command address register P2 bit. This forces a command address of 80 (hex) into the command address register. (Recall that the CCW at address 80 contains a command which causes the I/O device to return to the beginning of the current record.)

With the turn-off of the 'stop FLT' latch, the 'not stop FLT' signal activates the 'turn on setup chain command' signal. This initiates a chain command operation to fetch the CCW from address location 80 (hex) and to reselect the I/O device.

Note: During the chain command operation, the 'machine or setup reset' signal is activated; this signal turns off the 'start FLT' latch. When the CCW from address 80 (hex) is in the channel, the channel sends the Return to Beginning of Record command to the I/O device, then performs a chain command operation to fetch the read CDA CCW from address 80. From this point, the channel controls the loading of FLT tests until either another 'stop FLT' signal is received or all FLT tests have been read to storage. If another 'stop FLT' signal is received, the channel repeats the stop/start FLT sequence. If all FLT tests have been loaded, channel FLT operations are completed when the CPU drops the 'FLT mode' signal to the channel.

POLLING INTERRUPT OPERATION

Channel polling interrupt operations are performed to attempt to clear a pending interrupt condition in an attached control unit. The channel polling interrupt operation is described at both the simplified and detailed levels in the following text.

Simplified Polling Interrupt Operation

- Channel is polling ('select out' signal propagating through attached control units).
- 'Address in' and 'operational in' signals received by channel. (Control unit has pending interrupt condition).
- Channel gates unit address from control unit ('bus-in' lines) into unit address register.
- Channel commands control unit to stack status byte. (I/O interface disconnect)
- 'Interrupt request' signal sent to CPU interface.
- If channel receives instruction, instruction performed; status remains stacked.
- If channel receives 'interrupt response' signal, channel attempts to reselect device with stacked status. (Channel forces modified test I/O operation.)
- CSW stored, whether or not device selected and status obtained.
  a. If selection successful, stacked status (interrupt condition) cleared.
  b. If selection not successful, status remains stacked; CSW indicates reason for failure to clear stacked status.

Channel polling interrupt operations (Figure 3-10) are initiated when the channel is propogating the 'select out' signal through the attached control units (polling) and at least one of the control units contains a pending interrupt condition. When the 'select out' signal reaches a control unit with a pending 'interrupt' condition, the control unit places a unit address byte on the 'bus in' lines and activates the 'address in' and 'operational in' signals to the channel. The channel recognizes the polling interrupt request condition by gating the unit address byte from the 'bus in' latches into the unit address register. The channel then performs operations which cause the status byte (polling interrupt condition) to be stacked back to the control unit; i.e., the channel sends a stop command to the control unit causing the 'address in' signal to drop, receives the 'status in' signal from the control unit, and gates a second stop command to the control unit to cause an I/O interface disconnect. At this point, the channel unit address register contains the address of the device with the pending polling interrupt, the status byte is stacked in the control unit, and the control unit is disconnected from the channel.

The channel then raises the 'interrupt request' signal to the CPU interface. If the channel should now receive an instruction via the CPU interface (rather than an 'interrupt response' signal), the channel performs the operation and the status remains stacked at the control unit.

Note: The exception would occur if the channel received a Test I/O instruction specifying the address of the control unit with the pending interrupt.

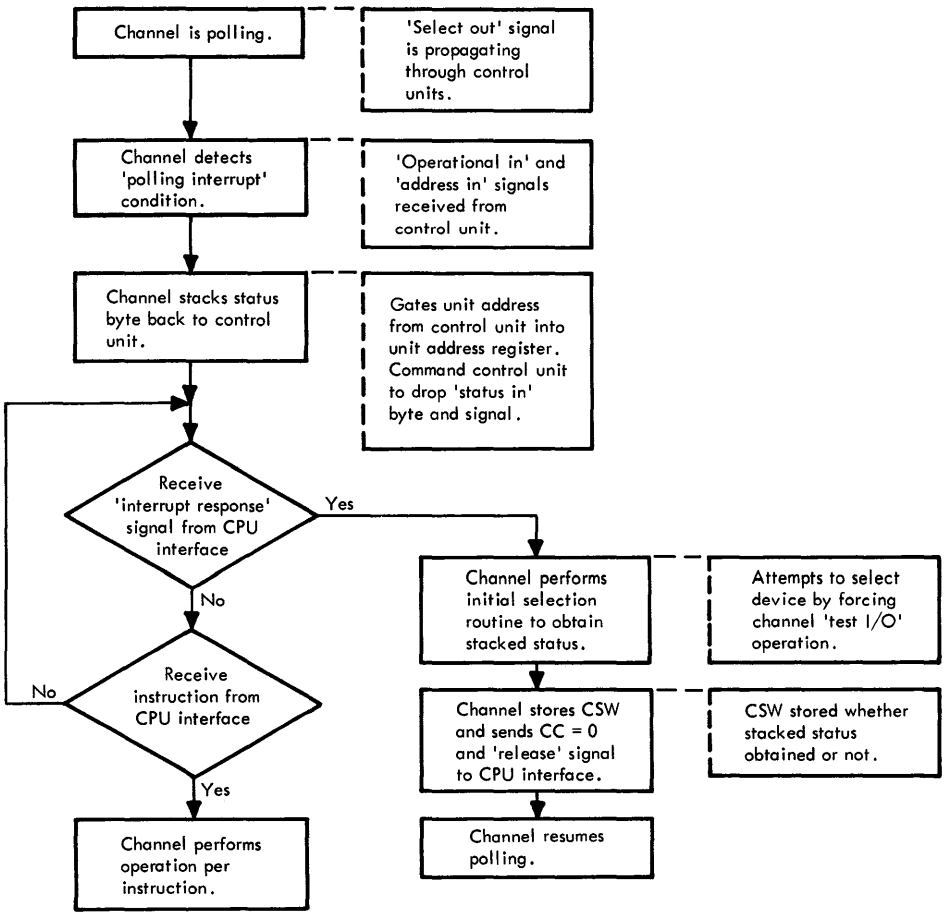


Figure 3-10. Simplified Polling Interrupt Operation Flow Chart

If the channel receives an 'interrupt response' signal from the CPU interface (Figure 3-10) prior to receiving an instruction, the channel forces a modified test I/O routine to reselect the I/O device and obtain the stacked status byte from the control unit. To obtain the stacked status, the channel performs an initial selection routine. If the selection is successful and the stacked status byte is obtained from the control unit, the channel enters a test I/O interrupt routine to store the CSW. After the CSW is stored, the channel sends the 'release' signal to the CPU interface. This causes the channel to clear the modified test I/O control logic and resume polling operations.

With these operations complete, the polling interrupt originally detected by the channel has resulted in clearing the status stacked at the control unit. If the channel attempt to obtain the stacked status is unsuccessful (due to a no selection, control unit busy or detected error condition), the CSW is still stored, indicating the cause of the unsuccessful selection attempt. In this case, the status remains stacked at the control unit and may be cleared by a Test I/O instruction to the channel, or as the result of another channel polling interrupt operation. After the CSW for an unsuccessful selection attempt is stored, the 'release' signal is sent to the CPU interface causing the 'interrupt response' signal to drop. This allows the channel to clear the modified test I/O control logic and resume polling operations.

Detailed Polling Interrupt Operations

Detailed channel operations for the polling interrupt operation are shown in flow chart form on Diagram 5-29, and the polling interrupt control logic is shown on Diagram 5-30. Channel polling interrupt operations are shown on Diagram 5-40 by routine as follows: (1) detect polling interrupt; (2) stack status; (3) initial selection to obtain stacked status; and (4) PIT interrupt routine. Details of each routine are outlined in the following paragraphs.

Detect Polling Interrupt

- Channel is polling.
- Turn on 'select out' latch; 'select out' signal propagates through control units.
- Receive 'address in' and 'operational in' signals from control unit.
- Turn on 'polling interrupt request' latch.

The detect polling interrupt routine (Diagram 5-29) consists of the channel detecting a pending interrupt at a control unit while polling. The channel is polling when the 'select out' signal is propogating through the attached control units. To initiate polling, the 'not select in', 'not polling interrupt', 'not status in end', 'not interface control check' and 'not machine reset' signals are AND'ed to turn on the 'select out' latch. If the channel receives an 'address in' signal and 'operational in' signal while the 'select out' signal is active, a control unit has an interrupt pending. Along with the 'address in' signal, the channel also receives a unit address byte on the 'bus in' lines.

In the channel, the 'address in', 'not setup' and 'not CCW valid' signals are AND'ed to turn on the 'polling interrupt request' latch (Diagram 5-30). If no control unit has a pending interrupt, the channel receives a 'select in' signal which turns off the 'select out' latch. When the 'select in' signal falls, the 'select out' latch is again turned on to resume channel polling operations (Diagram 5-29). Assume that the 'polling interrupt request' latch is turned on with the 'select out' signal active. This ends the detect polling interrupt routine and causes the channel to enter the stack status routine.

Stack Status

- Turn on clock.
- Activate 'polling interrupt reset' signal; activate 'machine or setup reset' signal.
- Gate 0's with correct parity (stop command) to 'bus out' lines.
- Gate unit address from 'bus in' latches into unit address register.
- Activate 'command out' signal to control unit.
- 'Address in' signal drops.
- 'Command out' signal drops.
- Turn off clock.
- Receive 'status in' signal.
- Turn on clock.
- Activate 'command out' signal to control unit (stop command still on 'bus out' lines).
- Turn on 'delay service in' latch.
- Activate 'suppress out' signal to control units.
- Turn on 'polling interrupt' latch.
- 'Status in' and 'operational in' signals from control unit drop (status stacked in control unit and control unit disconnected from channel).
- 'Polling interrupt extended' latch turns on.
- Clock turns off.
- 'Polling interrupt request' latch turns off.
- 'Interrupt request' signal activated to CPU interface.
- Channel awaits 'interrupt response' signal or instruction from CPU interface; polling interrupt unit address is in unit address register, status is stacked and control unit disconnected from I/O interface.

The stack status operation summarized above is shown in flow chart form on Diagram 5-29. Channel signals causing the stack status sequence of events are shown on the flow chart. Diagram 5-30 shows the logic associated with the stack status operation. Upon completion of the stack status operation, the channel awaits either an 'interrupt response' signal or instruction from the CPU interface. If an instruction is received, the channel performs the operation specified by the instruction. If an 'interrupt response' signal is received, the channel forces a modified test I/O operation and enters the initial selection to obtain stacked status routine.

Initial Selection to Obtain Stacked Status

- Receive 'interrupt response' signal.
- Turn on 'setup' latch.
- Turn on 'test I/O' latch.
- Gate unit address register contents (address of device with stacked status) to 'bus out' latches.

- Turn on clock.
- Activate 'setup reset' signal (resets selected channel latches, triggers, and registers).
- Turn on 'address out' latch; 'address out' signal sent to control unit.
- Turn off 'delay service in' latch.
- Deactivate 'suppress out' signal to I/O interface.
- Check unit address byte parity in 'bus out' latches for parity error.
- If parity error detected, turn on 'channel control check' latch.
  1. Block 'service out' signal.
  2. Turn on 'sequence 5' latch.
  3. Activate 'command reject or control check' signal.
  4. Gate 0's to 'bus out' latches (stop command).
  5. Activate 'interrupt status' signal.
  6. Enter PIT CSW store routine. (Stacked status remains pending in control unit.)
- If bus-out parity error not detected proceed with initial selection operation.
- Turn on 'select out' latch.
- Turn off 'polling interrupt' latch.
- Turn off clock.
- If 'select in' signal is received (no device selected) from control unit:
  1. Turn on 'program check' latch.
  2. Block 'service out' signal.
  3. Activate 'command reject or control check' signal.
  4. Gate 0's to 'bus out' latches (stop command).
  5. Activate 'interrupt status' signal.
  6. Enter PIT CSW store routine. (Stacked status remains pending in control unit.)
- If 'status in' signal is received (control unit busy) from control unit:
  1. Turn on 'control unit busy' latch.
  2. Turn on 'sequence 5' latch.
  3. Hold 'address out' signal active.
  4. Block 'service out' signal.
  5. Turn on clock.
  6. Turn off 'select out' latch. (I/O interface disconnect recognized by control unit.)
  7. 'Status in' signal drops.
  8. Enter PIT CSW store routine (stacked status remains pending in control unit.)
- If 'operational in' signal is received from control unit, proceed with initial selection operation.
- Turn off 'address out' latch.
- 'Address in' signal received.
- Activate 'address in gated' signal.
- Turn on clock.
- Compare address received from control unit (in 'bus in' latches) with address in unit address register.
- If unit addresses do not compare (wrong device responded):
  1. Turn on 'interface control check' latch.
  2. Turn on 'sequence 5' latch.
  3. Block 'service out' signal.
  4. Activate 'machine check' and 'interface reset' signals (control units on I/O interface reset).
  5. Perform logout operation if LOG ON MACH CHK switch on.
  6. Enter PIT CSW store routine (stacked status remains pending).
- If unit addresses compare (correct device selected), proceed with initial selection operation.
- Turn off 'setup' latch.
- Gate 0's (stop command) to 'bus out' latches.
- Activate 'command out' signal to control unit.
- Control unit drops 'address in' signal.
- Deactivate 'command out' signal.
- Receive status byte on 'bus in' latches and 'status in' signal from control unit. (Channel has received stacked status).
- Turn on 'sequence 5' latch.
- Turn on 'latch status byte' trigger (status byte latched into 'bus in' latches.)
- Turn on 'status in end' latch.
- Activate 'service out' signal.
- Turn off 'select out' latch; 'select out' signal to control units drops.

- 'Operational in' and 'status in' signals drop (control unit disconnected from I/O interface).
- Turn off 'status in end' latch.
- Initial selection to obtain stacked status operation complete. (Channel has obtained stacked status, cleared interrupt from control unit, and disconnected control unit from I/O interface.)
- Channel enters PIT CSW store operation to store CSW.

The initial selection to obtain stacked status operation summarized above is shown in flow chart form on Diagram 5-29. Channel signals causing the sequence of events are shown on the flow chart. Note that, regardless of whether the stacked status is obtained, that the channel enters the PIT CSW store routine to store the CSW. Thus, the CSW is available to the CPU with the stacked status from the device or an indication (channel status byte) of the reason the stacked status was not obtained.

PIT CSW Store

- Turn on 'interrupt' latch.
- Activate 'remember interrupt' signal.
- Turn off 'sequence 5' latch.
- Turn off 'polling interrupt' latch, if on.
- Turn on 'pseudo accept interrupt' latch.
- Turn on 'interrupt storage request' latch.
- Activate 'storage request' signal to BCU interface. (Request storage cycle to store CSW.)
- Turn on 'Z-address' latch.
- Turn on 'storage cycle' trigger.
- Receive 'BCU response' signal ('storage request' signal honored).
- Turn on 'latch Z-address' latch.
- Activate 'Z-address latch gated' signal.
- Gate address 64 (decimal) to SAB.
- Activate 'store' signal to BCU interface.
- Force 'storage protect parity' bit to 'storage protection' lines (master key of all 0's with correct parity).
- Activate all mark bits to mark lines.
- Receive 'BCU data request' signal.
- Gate CSW (all bytes) to SBI lines.
- Receive 'accept' signal.
- Turn on 'accept' latch; turn off 'interrupt' latch.
- Turn on 'release CPU polling interrupt' latch.
- Turn off 'interrupt storage request' latch.
- Receive 'BCU advance pulse' signal.
- Turn off 'accept' latch.
- Turn off 'remember interrupt' latch.
- Activate 'storage cycle complete' signal.
- Turn off 'latch Z-address' latch.
- Turn off 'Z-address' latch.
- Deactivate 'Z-address latch gated' signal.
- Activate internal channel 'accept' signal (activate 'release' signal to CPU interface).
- Send condition code 0 to CPU interface.
- 'Interrupt response' signal to channel (via CPU interface) drops.
- Turn off 'polling interrupt extended' latch.
- Turn off 'test I/O' latch.
- Turn off 'release CPU polling interrupt' latch.
- Turn off 'pseudo accept interrupt' latch.
- CSW is stored, polling interrupt operation is complete, and channel resumes polling operations. (Channel free to perform other operations.)

The PIT CSW store operation summarized above is shown in flow chart form on Diagram 5-29. 'Polling interrupt' logic associated with the PIT CSW store operation is shown on Diagram 5-30. Note that although all CSW bytes are stored at main storage address location 64, and that only the channel status byte and control unit status byte (if the stacked status byte was obtained by the channel) are significant. The remaining CSW bytes should be stored as all 0's with correct parity.

This chapter describes the existing features and versions (attachment) for the basic 2860 Selector Channel.

ADDRESS PREFIXING FEATURE

- Used with multiprocessor system.
- Provides all data transferred to or from the system with associated ID lines that identify which CPU initiated the operation.
- Feature logic for one channel is contained on that channel gate.
- Feature logic is shown on the 2860 version 003 ALD pages.

The 2860 Address Prefixing Feature allows a system to communicate with a channel when the channel can be controlled by more than one processor. In a multiple-processor system, more than one operation can be in progress simultaneously. Therefore, the identity of the processor that initiates a channel operation must be preserved by the channel throughout the operation. The system can then identify control information and data in storage according to the processor responsible for the operation.

Address prefixing is not under program control. The feature, once installed, functions whenever the specified gating conditions are satisfied. No I/O instruction modification for channel operation is required when the address prefixing feature is installed, nor is the data transfer rate of the 2860 Selector Channel affected.

Address prefixing does not provide facilities for off-line testing, no additional indicators appear on the 2860 CE panel or power control panel, and the 2860 CE panel and power panel controls are not affected.

Addressing Prefixing System Interface

The Address Prefixing Feature adds new lines to the CPU and BCU interfaces of the basic 2860 (Figure 4-1). The lines, designated 'CPU identity' (CPU ID), connect the channel to a unit of the system that provides and accepts channel CPU and BCU interface signals required for proper system operation. Channel operations are not affected by the Address Prefixing Feature. Channel handling of the CPU ID bits added by address prefixing is performed as an automatic function that has no effect on channel sequencing and control operations.

Three incoming multiplex lines are added to the channels CPU interface: 'CPU identity' (ID) line 1, 'CPU ID' line 2, and 'CPU ID' line parity. Signals on the CPU ID lines are valid on either of two conditions:

1. For the duration of the 'select channel' signal received by the channel via the CPU interface.
2. For the duration of the 'interrupt response' signal received by the channel via the CPU interface.

The 'CPU ID' signals identify which processor is making the selection or which processor is responding to an 'interrupt request' signal from the channel. No standard CPU interface lines to the channel are modified by the Address Prefixing Feature.

Three outgoing simplex lines are added to the channels BCU interface: 'data identity' (ID) line 1, 'data ID' line 2, and 'data ID' line parity. Signals on the data ID lines are valid on either of two conditions:

1. From the turn-on of the 'storage request' latch in the channel until the rise of 'accept' signal (received via the BCU interface) at the channel. (The 'accept' signal turns off the 'storage request' latch.)
2. From the turn-on of the 'interrupt storage request' latch at the channel to the turn-on of the 'accept' latch at channel.

The data ID signals identify the channel operation in progress as one initiated by a specific processor. This information is meaningful to the system but not to the channel. No standard BCU interface lines are modified at the channel by the Address Prefixing Feature.

Address Prefixing I/O Interface

The standard channel-to-control-unit I/O interface is not modified in any way by installation of the address prefixing feature. Channel operations across the I/O interface are completely unaffected, and attached control units are not affected by the feature. See Figure 4-1.

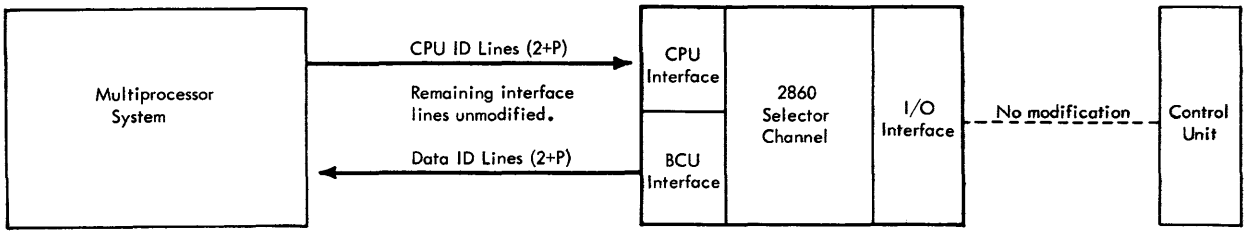


Figure 4-1. Address Prefixing Interfaces

Functional Units

CPU Identity Register

- CPU ID register consists of three polarity holds: ID register 1, ID register 2, and ID register P.
- CPU ID bits from CPU interface are gated into CPU ID register when:
  1. IPL setup operation begins.
  2. At end of start I/O initial selection sequence.
- For start I/O operation, bits are gated to register when:
  1. Status byte equals all 0's with 'sequence 1' latch on.
  2. 'Command immediate' indication (channel-end bit active) received by channel during initial selection sequence.
- 'Accept to CPU start I/O' latch turned on to gate bits into register.
- 'Release' signal and CC = 0 sent to CPU interface when bits are gated into CPU ID register.

The CPU identity (ID) register (Figure 4-2) is a storage element composed of three polarity holds. The register contains two identity bits plus the parity bit. Parity checking and parity generating circuits are not involved for the register. The 'machine reset' signal clears the two bit positions and sets the parity position.

The CPU ID register receives inputs from the CPU ID lines. Signals on the CPU ID lines are gated into the ID register on two conditions: (1) when the setup sequence starts during an IPL operation; and (2) during a start I/O initial selection sequence. For the IPL operation, the CPU ID bits are gated into the polarity holds when the 'turn on setup IPL' signal is activated (Figure 4-2). This signal activates the 'enable latch ID register' signal which, in turn, gates any active CPU ID bits into the respective polarity holds.

For the start I/O initial selection routine, the 'enable latch ID register' gating signal is activated when the channel determines that the I/O device is available for data transfer operations, or when the channel has received a 'command immediate' indication from the control unit (Figure 4-2). In either case, the channel sends the 'release' signal and condition code 0 to the CPU interface while the CPU ID bits are gated into the CPU ID register.

The channel determines that the I/O device is available for data transfer operations when the 'start I/O latch', 'sequence 1', 'not sequence 5', 'status in', 'T2', and 'status equals 0' signals are active. These signals turn on the 'accept to CPU start I/O' latch. In turn the output of the latch ('accept to CPU start I/O') generates the 'release' signal to the CPU interface and is AND'ed with the 'T2' and 'not T4' signals to activate the 'enable latch ID register' signal. This signal gates the CPU ID bits into the polarity holds.

The 'accept to CPU start I/O' latch is also turned on when the channel receives a 'command immediate' indication from the control unit. In this case, the 'start I/O latch' signal is active and the 'chain command latch L1' and 'channel end' bit ('data in bus bit 4') signals are AND'ed to turn on the latch. The 'release' signal is then generated and the 'CPU ID' bits are gated to the polarity holds.

Data ID Control Gates

- For normal read-type or write-type data storage cycle, contents of CPU ID register are gated to data ID lines (1, 2, and P):
  1. CPU ID register contents gated to data ID lines when 'storage request' latch turns on.
  2. Data ID bits degated when 'storage request' latch is turned off by channel receipt of 'accept' signal from BCU interface.
- For logout storage cycle operations, CPU ID bits from CPU interface are gated to BCU interface data ID lines:
  1. 'Inhibit on log' and 'interrupt storage request latch' signals gate bits to data ID lines.

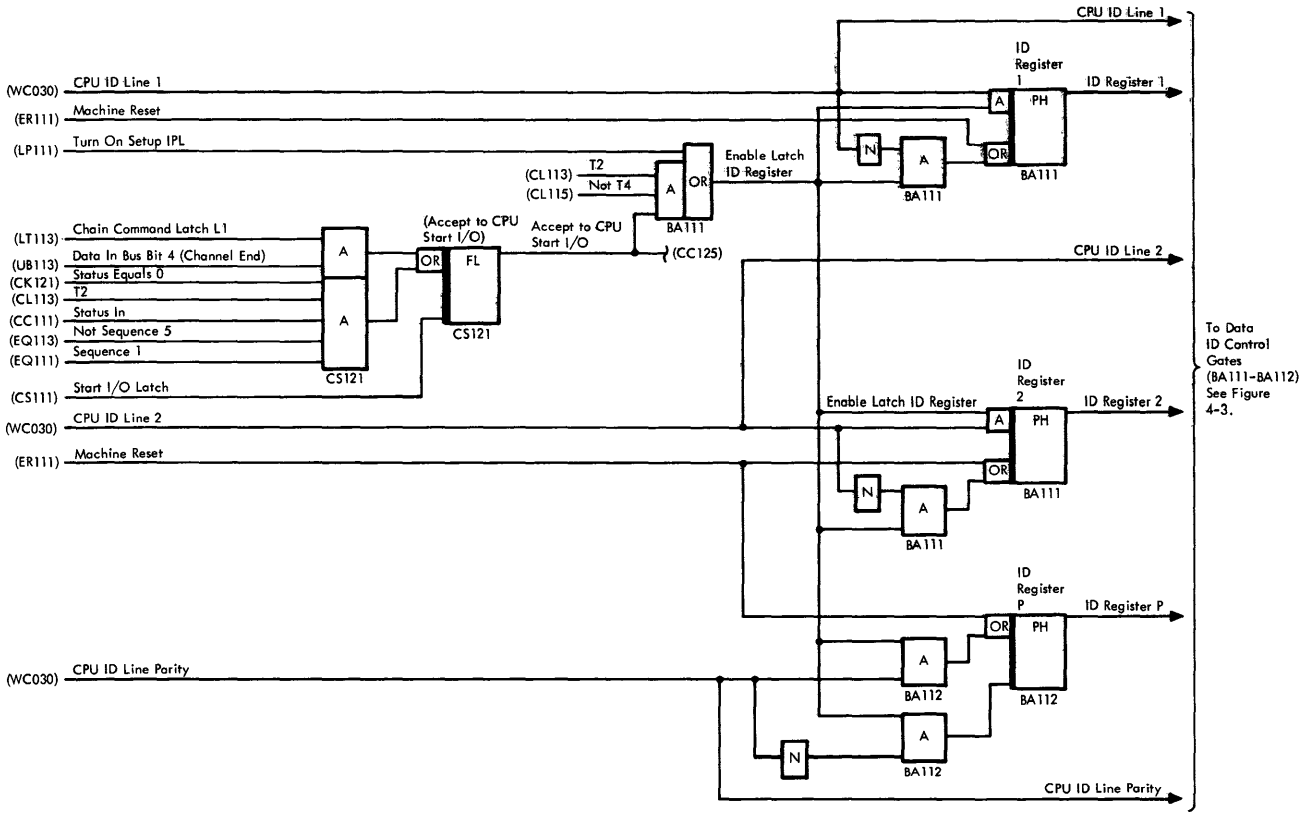


Figure 4-2. CPU Identity Register and Input Control

2. Data ID bits degated when 'interrupt storage request' latch turns off; this latch turns off when 'accept' latch is turned on by channel receipt of 'accept' signal.

- For CSW store operation, CPU ID bits gated to data ID lines:
  1. 'Z address latch' and 'interrupt storage request latch' signals gate bits to data ID lines.
  2. Data ID bits degated when 'interrupt storage request' latch turns off.

- For start I/O CAW and initial CCW fetch cycles, CPU ID bits gated to 'data ID' lines:
  1. 'Remember start I/O' latch and 'storage request latch' signals gate bits to data ID lines.
  2. Data ID bits degated when 'storage request' latch is turned off by 'accept' signal.

The data ID control gates (Figure 4-3) provide the channel with the means of gating identity bits to the BCU interface data ID lines each time the channel raises the 'storage request' signal to the BCU interface. If the channel is not performing a logout operation, storing the CSW or fetching the CAW and initial CCW for a start I/O operation, the bits from the CPU ID register are gated to the 'data ID' lines via the data ID control gates. If the channel is storing log information, fetching the CAW and initial CCW, or storing the CSW, the CPU ID bits supplied to the channel CPU interface are gated to the 'data ID' lines via the data ID control gates. In this case, the contents of the CPU ID register are not changed even though the data ID bits originate at a source other than the register.

The gating of bits to the data ID lines is controlled primarily by the 'storage request latch', 'interrupt storage request latch', 'inhibit on log', and 'remember start I/O latch' signals. For storage cycles not involving a logout operation, start I/O CAW and initial CCW fetch, or a CSW store operation, the contents of the CPU ID register are gated to the data ID lines (1, 2, and P) by the 'storage request latch' signal. The data ID bits remain active until the 'storage request' latch is turned off when the channel receives the 'accept' signal received via the BCU interface.

If the channel is storing log information or the CSW, the CPU ID bits from the CPU interface are gated to the data ID lines (Figure 4-3) by the 'inhibit on log' (logout operation) or 'interrupt storage request latched on' ('CSW store' operation) signal and by the 'interrupt storage

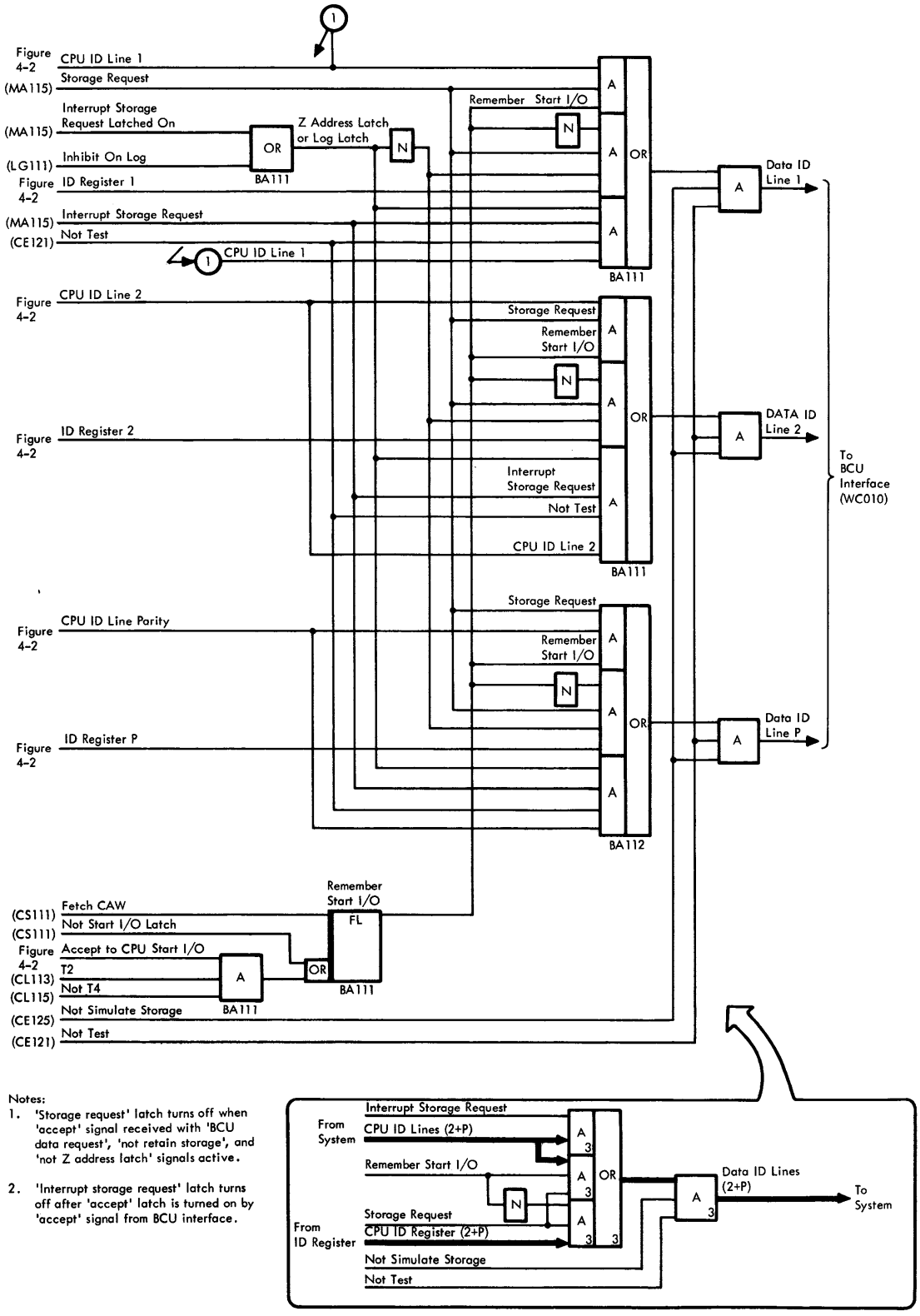


Figure 4-3. Data ID Control Gates

request latch' signal. The data ID bits remain active until the 'interrupt storage request' latch is turned off. This latch is turned off by the 'accept latch' signal (which is activated when the 'accept' latch is turned on by the 'accept' signal received via the BCU interface).

If the channel is fetching the CAW and initial CCW for a start I/O operation, the 'remember start I/O' latch is turned on (Figure 4-3). In this case, the CPU ID bits are gated to the data ID lines by the 'storage request latch' signal. For both the CAW fetch and CCW fetch cycles, the 'storage request latch' is turned off by receipt of the 'accept' signal from the BCU interface; thus, the data ID bits are active while the 'storage request' latch is on for each storage cycle. When the CAW and CCW have been fetched (or the operation terminated by a detected error), the 'remember start I/O' latch is turned off by the 'not start I/O' signal or when the 'release' signal is sent to the CPU interface. (Recall that the CPU ID bits are gated into the CPU ID register when the 'release' signal for the start I/O operation is activated.)

Principles of Operation

The Address Prefixing Feature informs the host system which of its processors is communicating with, or has last communicated with, the channel. Address prefixing supplies data ID bits to the system each time the channel requests a storage cycle. If the channel is storing log information, storing the CSW, or making the initial CAW and CCW fetches on a start I/O instruction, the same ID bits that are sent by the system are returned to the system. When a start I/O initial selection sequence is completed and data transfer is taking place, the identity of the processor which initiated the store operation is stored in the CPU ID register and gated to the system during each storage cycle to access main storage.

Although no parity generation or checking is performed within the channel, the data ID bits are examined by the system for correct (odd) parity. If the system detects a parity error in the data ID bits arriving with a 'storage request' signal from the channel, no storage is selected. A 'storage address check' signal, false 'accept' signal, and false 'BCU advance pulse' signal are sent to the channel, and the operation at the channel ends.

Multiple-CPU control of a channel can produce misleading indications during a completely normal operation. Assume that CPU A has just completed a successful operation with the channel. No interrupt conditions are pending either in the channel or control unit, and no control units or I/O devices are left busy. If CPU B now executes a test I/O instruction with this channel, CPU B receives a 'release' signal with a condition code 0. If CPU A should now issue a start I/O instruction to the channel before CPU B reselects the channel for its intended start I/O instruction, CPU B receives a busy indication (condition code 2) and a 'release' signal on the start I/O attempt, even though the immediately proceding test I/O operation indicated that the channel was available.

HIGH-SPEED DIRECT ACCESS STORAGE PRIORITY FEATURE

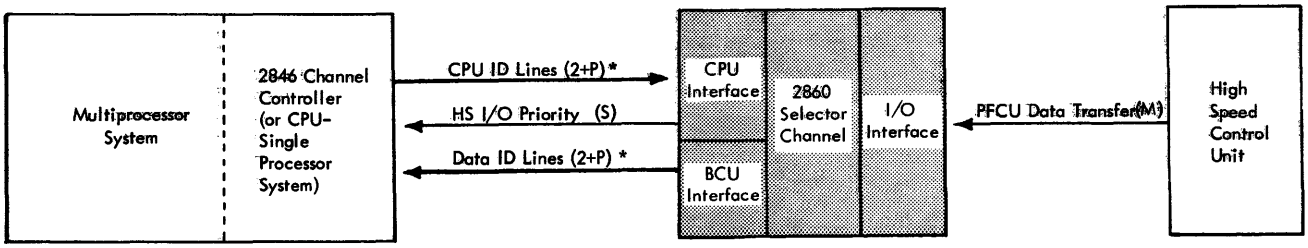
- Currently used only with the System/360 Model 67-2 and Model 85.
- Address Prefixing Feature is currently a preprequisite, on Model 67-2.
- Prevents CPU from selecting main storage when a high-speed I/O device such as the parallel file control unit requires access to main storage; this prevents an overrun condition from developing in the channel due to a storage-access priority conflict.
- Feature logic is shown on 2860 version 004 ALD pages, for Model 67-2 and version 001 and version A18 for Model 85.

Introduction

The High-Speed Direct Access (HSDA) Storage Priority Feature enables the channel to alert system to an imminent 'storage request' signal from a 2860 whenever the 2860 is transferring data between main storage and a high-speed I/O device such as the parallel file control unit (PFCU). Once the system is alerted to the impending storage request, processor accesses to main storage are inhibited to ensure that the channel can access main storage. This reduces the possibility of an overrun developing in the channel or the high-speed I/O device.

The HSDA Storage Priority Feature requires that the Address Prefixing Feature be installed in the channel only when operating with the System/360, Model 67-2. Additionally, all high-speed I/O devices attached to the channel are connected to an I/O interface multiplex line designated 'parallel file control unit (PFCU) data transfer'. Refer to Figure 4-4. The signal used to alert the system of the impending storage request is designated 'high-speed I/O priority', which is a simplex line added to the CPU interface of the channel.

Presence of the HSDA Storage Priority Feature does not affect the basic operation of the 2860.



* Note: These lines are added by the Address Prefixing Feature, which is a prerequisite for the HSDA Storage Priority Feature. All basic 2860 Selector Channel interface lines remain unaltered.

Figure 4-4. High-Speed Direct Access Storage Priority Feature Interface Lines

Principles of Operation

- 'High-speed I/O priority' signal activated in anticipation of channel 'storage request' signal activation to store or fetch data doubleword.
- Value in byte count latches and rate of data transfer by I/O device determines precise time that 'high-speed I/O priority' signal is activated.
- Signal deactivated by 'BCU response' signal.

The High-Speed Direct Access Storage Priority Feature activates the 'high-speed I/O priority' signal to the channel BCU interface by anticipating that the channel will activate the 'storage request' signal to the BCU interface. The feature anticipates activation of the 'storage request' signal by examining the status of the channel byte count latches and the status of signals which gate data bytes (via the I/O interface) into or from the channel's B-register; in addition, the feature is designed to assume that the high-speed I/O devices send and receive data at the rate of about one byte per 800 ns. Once activated, the 'high-speed I/O priority' signal remains active until the channel receives a 'BCU response' signal or detects a 'machine or setup reset' condition.

Interface Lines

The High-Speed Direct Access Storage Priority Feature adds the following lines to the standard channel interfaces:

1. The 'parallel file control unit (PFCU) data transfer' multiplex line available on the standard I/O interface. The 'PFCU data transfer' signal is activated to the channel by a high-speed control unit a minimum of 8 usec before the first 'service in' signal is activated by the control unit; the 'PFCU data transfer' signal remains active until the channel accepts the ending status byte from the control unit. However, the signal may be activated and deactivated during keying operations provided that the signal is always raised at least 8 usec before the next 'service in' signal is activated. The 'PFCU data transfer' signal is not activated by the actual PFCU for data transfer operations of less than 16 bytes; however other high-speed I/O devices are not restricted in this manner.
2. The 'high-speed I/O priority' line is added to the channel's standard BCU interface. When the 'PFCU data transfer' signal is active, the 'high-speed I/O priority' signal is activated approximately 2 to 2.5 usec before the channel activates the 'storage request' signal. The 'high-speed I/O priority' signal is deactivated when the channel receives the 'BCU response' signal (or a machine or setup reset condition is detected by the channel). Because of the method used to activate the 'high-speed I/O priority' signal, the 'storage request' signal may not be activated within the 2 to 2.5 usec period following activation of the 'high-speed I/O priority' signal, the signal remains active until the 'BCU response' signal is finally received or the channel detects a 'machine or setup reset' condition.
3. The CPU ID lines are added to the channel's standard CPU interface, and the data ID lines are added to the channel's standard BCU interface by the Address Prefixing Feature when the channel is operating with the System/360, Model 67-2. (See "Address Prefixing Feature".)

High-Speed I/O Priority Operation

- 'High-speed I/O priority' signal activated for storage cycles only when the following signals are active: (1) 'PFCU data transfer' signal (from I/O interface), (2) 'not simulate storage', (3) 'not sequence 5', and (4) 'not read skip'.

- Activation and deactivation of 'high-speed I/O priority' signal controlled by turn-on and turn-off of 'high-speed priority' latch.

- For read-type operation:
  1. 'High-speed priority' latch turns on when BCL equals 6 and byte is being gated into B-register.
  2. 'High-speed I/O priority' signal activated approximately 2 to 2.5 usec prior to activation of 'storage request' signal.
  3. When B-register is full, channel turns on 'storage request' latch.
  4. 'BCU response' signal turns off 'high-speed priority' latch.

- For write-type operation:
  1. 'High-speed priority' latch turns on when BCL equals 6 or 7, count is greater than 16, and byte is being gated into B-register.
  2. 'High-speed I/O priority' signal activated.
  3. When last byte gated from B-register, channel turns on 'storage request' latch.
  4. 'BCU response' signal turns off 'high-speed priority' latch.

The 'high-speed I/O priority' signal (Figure 4-5) is activated during read-type or write-type channel operations for accesses to main storage. For the signal to be activated, the 'PFCU data transfer' signal from the I/O interface must be active, and the channel must not be engaged in a 'read skip' or test-mode 'simulate storage' operation. Entry of the channel into a sequence-5 routine also prevents activation of the 'high-speed I/O priority' signal.

Assume that the channel is performing a data transfer operation and that a 'read skip' operation is not in progress. The 'not skip and read' signal gates the 'PFCU data transfer' signal to enable one input to the 'priority output' AND. The 'not simulate storage' and 'not sequence 5' signals enable two other inputs to the 'priority output' AND. As long as these conditions are present, activation and deactivation of the 'high-speed I/O priority' signal is controlled by the turn-on and turn-off of the 'high-speed priority' latch. (When on, the latch output satisifies the 'priority output' AND to activate the 'high-speed I/O priority' signal.)

Turn-on of the 'high-speed priority' latch is controlled by: (1) the output of the channels byte count latches; (2) the B-register byte gating signals for the read-type or write-type operations; and (3) for the write operation, the value in the count register.

Assume that the channel is performing a read-type operation. When the 'byte count latch' bits indicate a value of 6 and the 'gate bus in to B-register' signal is active (to gate another byte into the B-register), the 'high-speed priority' latch (Figure 4-5) turns on to activate the 'high-speed I/O priority' signal to the CPU interface. Assuming the channel does not detect an error condition to activate the 'machine or setup reset' signal, the channel completes the loading of the B-register, then turns on the 'storage request' latch. This raises the 'storage request' signal to the BCU interface. When the channel receives the 'BCU response' signal, the 'high-speed priority' latch is turned off (by the 'fast BCU response' signal) to deactivate the 'high-speed I/O priority' signal. Note that the 'high-speed priority' latch is turned on when a data byte is being gated into B-register byte position 5 (BCL = 6). Since the channel must gate two more bytes into the B-register before the 'storage request' latch is turned on, the 'high-speed I/O priority' signal is active at least 2 to 2.5 usec before the 'storage request' signal is activated.

Note: If the high-speed I/O device being serviced transfers data at a rate less than one byte per 800 ns, the 'high-speed I/O priority' signal may be active for a longer period before the 'storage request' signal is activated.

For a write-type operation, the 'high-speed priority' latch is turned on when the value in the byte count latches is 6 or 7 and a byte is being gated from the B-register as a result of the 'change byte count write' signal (Figure 4-5). Turn-on of the 'high-speed priority' latch activates the 'high-speed I/O priority' signal in the manner described for a read-type operation. The latch is turned off by the 'BCU response' signal from the BCU interface to deactivate the 'high-speed priority' signal. If the channel count register contains a count value of 16 or less bytes for the write-type operation, turn-on of the 'high-speed priority' latch is inhibited. In this case, turn-on is inhibited because the channel has already fetched all required data from main storage for the current CCW.

PCI Gating and Interrupt Request Operation

- Gated 'PFCU data transfer' signal:
  1. Prevents PCI condition from causing CSW store operation.
  2. Turns on 'parallel file control unit' latch.

- If 'PFCU data transfer' signal is degated or drops, PCI condition can activate 'interrupt request' signal, provided 'path working' and 'CCW valid' signals are active.

- 'Parallel file control unit latch' signal prevents PCI condition from activating 'interrupt request' signal when only 'CCW valid' signal is active.

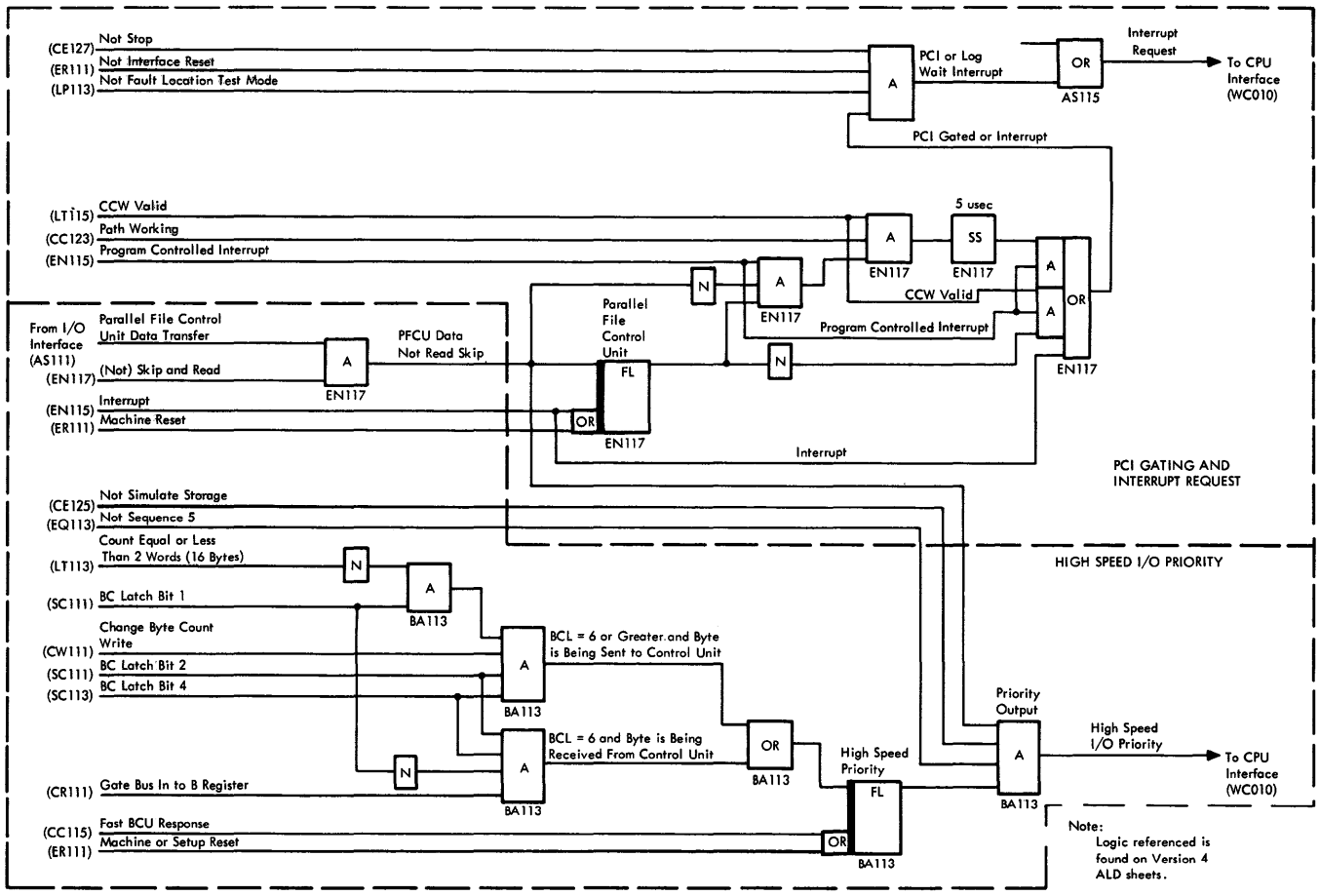


Figure 4-5. High-Speed Direct Access Storage PFCU Logic

The 'PCI gating and interrupt request' logic is affected by the 'high-speed direct access storage priority' feature as described in the following text.

While the 'PFCU data transfer' signal is gated into the channel by the 'not skip and read' signal, PCI CSW storage operations are inhibited (Figure 4-5). The gated 'PFCU data transfer' signal turns on the 'parallel file control unit' latch and also inhibits activation of the 'PCI gated or interrupt' signal when the 'CCW valid', 'program controlled interrupt', and 'path working' signals are active. The active output of the 'parallel file control' latch inhibits activation of the 'PCI gated or interrupt' signal when only the 'CCW valid' and 'program controlled interrupt' signals are active. In either case, inhibiting activation of the 'PCI gated' or 'interrupt' signal prevents a PCI condition from activating an 'interrupt request' signal to the CPU interface while 'high-speed direct access storage' operations are in progress. If the 'PFCU data transfer' signal drops, a PCI condition will activate the 'interrupt request' signal if both the 'path working' and 'CCW valid' signals are active. In this case, a PCI CSW status word may be stored if the channel receives an 'interrupt response' signal. If the 'path working' signal is inactive, the PCI condition is prevented from activating the 'interrupt request' signal by the active output of the 'parallel file control unit' latch. In this case, the PCI condition is prevented from activating the 'interrupt request' signal if a channel chain-command operation is in progress. If the channel enters an interrupt routine or if a machine reset condition is detected, the 'parallel file control unit' latch is turned off. If the latch is turned off by an 'interrupt' signal, the PCI condition is stored in the CSW as a result of an ending CSW-store operation. If the latch is turned off by the 'machine reset' signal, the PCI signal is also deactivated; thus, the PCI condition cannot activate the 'interrupt request' signal.

CHANNEL-TO-CHANNEL ADAPTER FEATURE

Introduction

The channel-to-channel adapter (CCA) is an attachable feature for the 2860 Selector Channel but is not restricted to this application. The CCA can be installed on systems other than those using the 2860 Selector Channel. When installed on a 2860 Selector Channel, the CCA physically appears in channel gate location E1 on one large SLT board.

Principles of Operation

When installed in a 2860 Selector Channel, the CCA feature connects the I/O interface of the 2860 with another System/360 channel I/O interface. Refer to Figure 4-6. The CCA responds to either connected channel as a normal control unit. To the two channels serviced by the CCA, the CCA appears as the logical equivalent of a pair of control units connected back to back. Further, either channel serviced by the CCA appears to be a control unit to the other channel.

The byte-by-byte communication link established by the CCA allows communication between two separate systems through their respective channels (Figure 4-6) or allows storage-to-storage transfer in a single system via two of its own channels (Figure 4-7). In either configuration, CCA and channel operations (once initiated) can continue without CPU control; thus, CCA transfer operations and CPU operations can proceed independently.

References

For a detailed analysis of the CCA operations, refer to the following manuals:

FETMM, *IBM System/360 Channel-to-Channel Adapter, Models 6006 and 6006B,* SY27-2295

FEDM, *IBM System/360 Channel-to-Channel Adapter, Models 6006 and 6006B,* SY27-2296

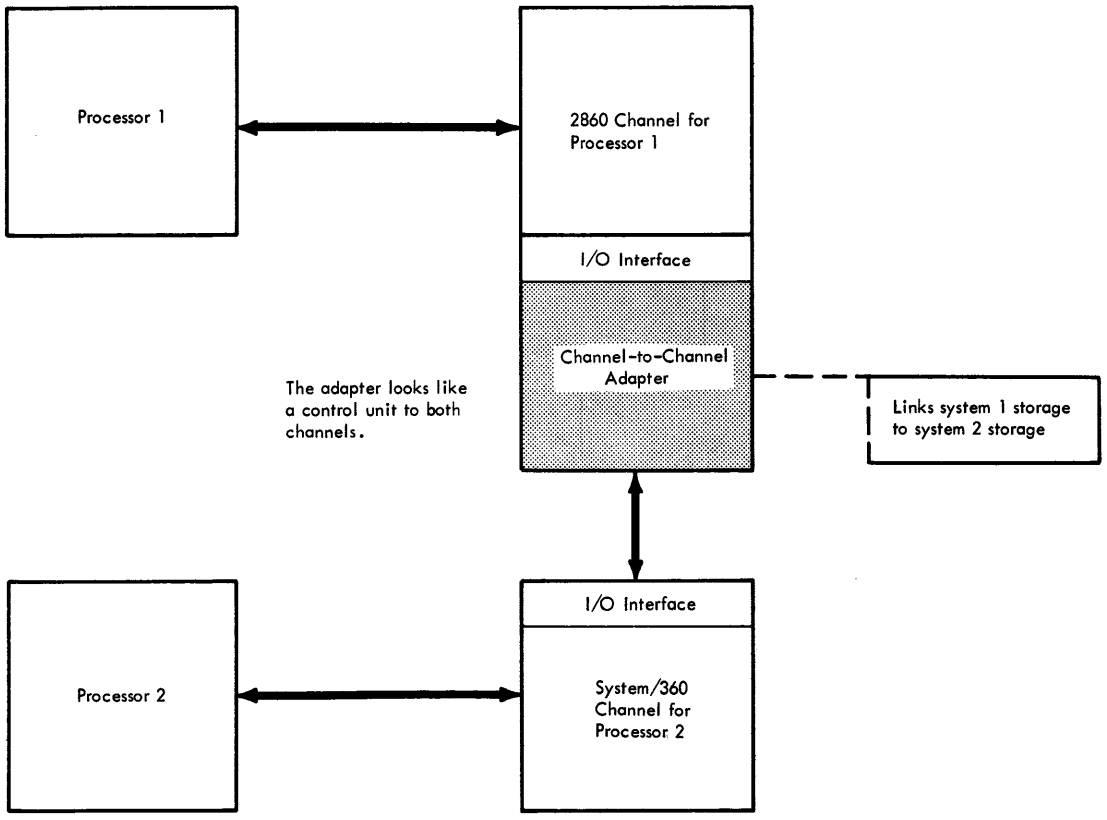


Figure 4-6. Channel-to-Channel Adapter, Intersystem

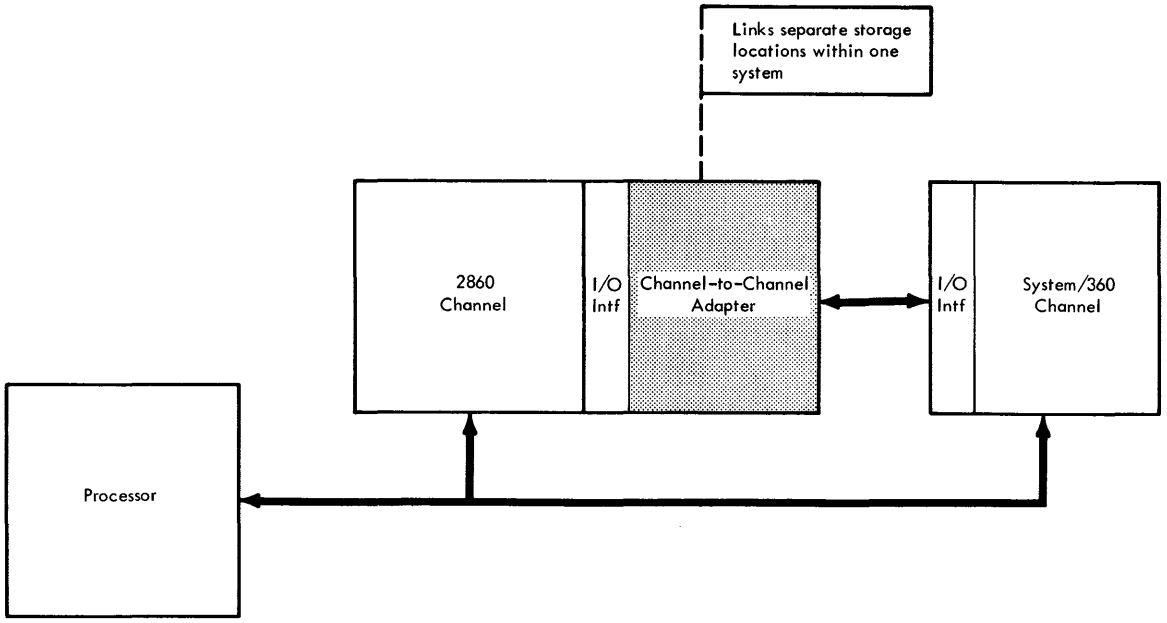


Figure 4-7. Channel-to-Channel Adapter, Intrasystem

STORAGE CHANNEL VERSION

The function of the storage channel is to permit high-speed data transfers between storage locations to proceed concurrently with CPU operations. To the host system, which may be an IBM System/360 Model 65, 67, or 75, the storage channel CPU and BCU interfaces are identical to the 2860 Selector Channel. The storage channel responds to the usual instructions issued to a 2860. The instruction format for the storage channel is the same as for the 2860 Selector Channel but the command format is different.

The storage channel physically displaces one channel gate on the 2860 Selector Channel frame. The gate 3 position in the 2860 Selector Channel frame is always used for the storage channel. The storage channel contains its own CE panel with indicators and manual controls. Storage Channel ALD logic is shown on ALD's designated for Machine 6009.

Data can be transferred to or from any addressable storage in the system, including any combination of high-speed and LCS storage units. Data transfer must begin on singleword or doubleword boundaries, but the amount of data transferred need not be an integral multiple of doublewords. The storage channel transfers one doubleword at a time between storage locations.

For a complete description of the 2860 Storage Channel, refer to FETOM, 2860 Storage Channel, Form Y22-2906, and FEDM, 2860 Storage Channel, Form Y22-2907.

MODEL 91 VERSION

The basic 2860 Selector Channels used with the System/360, Model 91, are modified as shown on the 2860 Version 005 ALD's. The modifications provide operating compatibility between the channel and the Peripheral Storage Control Element (PSCE) of the Model 91. Both the standard BCU and CPU interfaces of the channel are connected to the PSCE. Interface lines added or modified to the standard 2860 Selector Channel for the Model 91 are described in "BCU Interface", Chapter 1. The active-state polarity of channel CPU and BCU interface signals is negative.

Four lines ('Pre CDA', 'control word request', 'channel end', and 'error sample') have been added to the channel's basic BCU interface. A fifth line ('BCU advance pulse', designated 'channel advance' for the Model 91) has been changed to a simplex line. Channel logic to accommodate the added channel functions are shown in Figure 4-8.

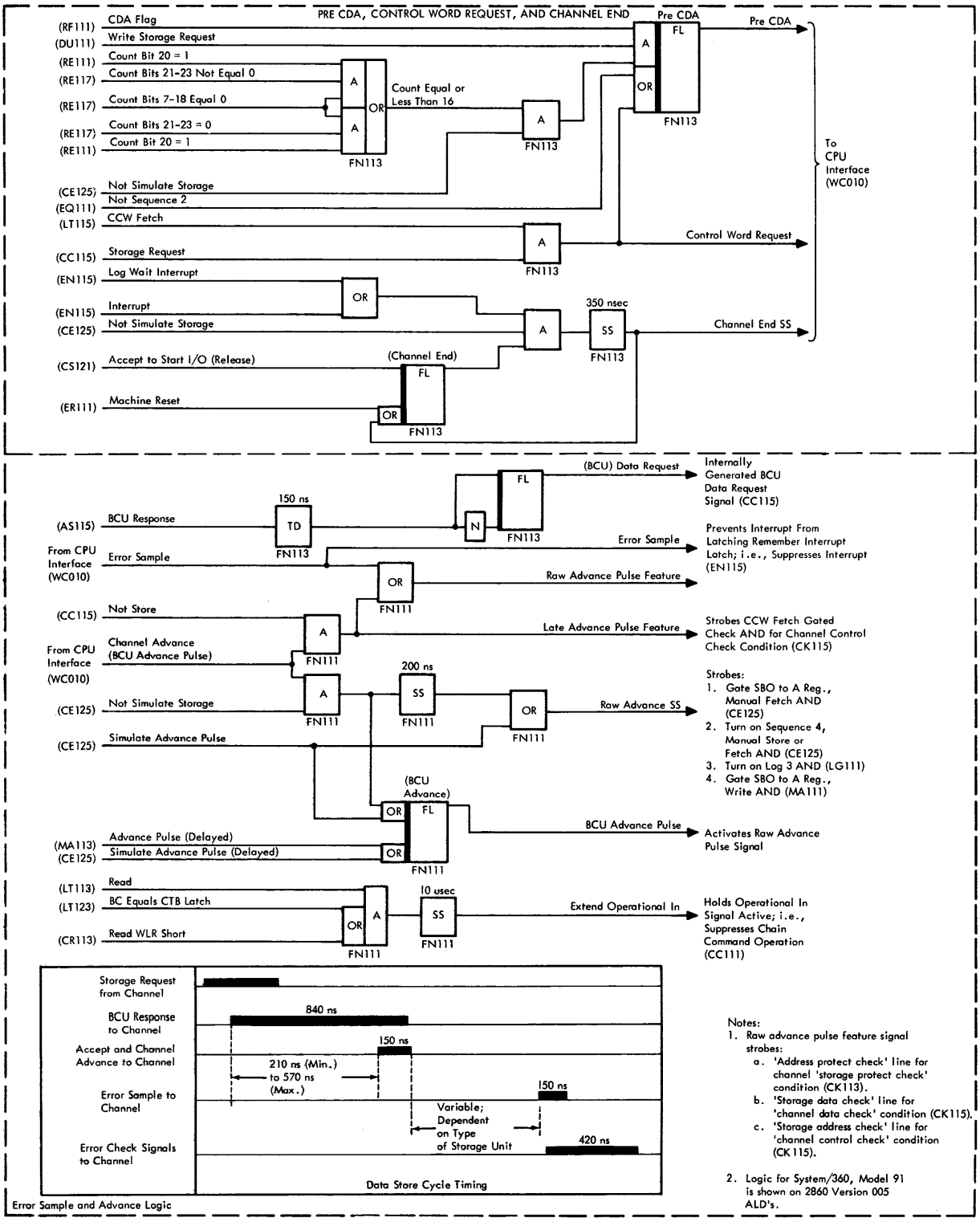


Figure 4-8. Model 91 Version Logic

Pre-CDA, Control Word Request, and Channel End

- 'Pre-CDA' signal is activated for write CDA operation when next to last data doubleword of current CCW is fetched:
  1. 'Pre-CDA' latch turned on when count is equal to or less than 16.
  2. Latch turned off when 'control word request' signal is activated for CCW fetch cycle.

- 'Control word request' signal is activated for CAW and CCW fetch cycles by AND'ing 'CCW fetch' and 'storage request' signals.

- 'Channel end' signal is activated when 'interrupt request' signal is activated to end an operation initiated by start I/O instruction:
  1. 'Channel end' latch turns on at beginning of I/O operation.
  2. 'Interrupt' or 'log wait interrupt' signal AND's with 'channel end latch' signal to trigger singleshot.
  3. Singleshot provides 350-ns 'channel end' signal.
  4. 'Channel end' latch turns off.

Channel activation and timing of the 'pre-CDA' signal (activated during a write CDA operation when the next to last data doubleword of a CDA CCW is fetched) is described in "Write Chain Data Operation", Chapter 3. As shown in Figure 4-8, the 'pre-CDA' signal is activated to the PSCE by AND'ing the 'count equal or less than 16', 'write storage request', and 'CDA flag' signals to turn on the 'pre-CDA' latch. If the write CDA operation progresses normally, the 'pre-CDA' latch remains on until the 'control word request' signal is activated when the channel requests a storage cycle to fetch the CCW. If the write CDA operation is terminated by a detected error prior to activation of the 'control word request' signal, the 'pre-CDA' latch is turned off by the 'not sequence 2' signal. (The 'sequence 2' latch is turned off during the sequence-5 ending routine.)

Channel activation and timing of the 'control word request' signal is described throughout Chapter 3. The signal is activated any time the channel issues a 'storage request' signal to fetch the CAW or a CCW. As shown in Figure 4-8, coincidence of the 'CCW fetch' and 'storage request' signals activates the 'control word request' signal. (The 'CCW fetch' signal is active for a CAW fetch cycle as well as for a CCW fetch cycle.)

The 'channel end' signal (Figure 4-8) is 350 ns and is issued when the channel ends an operation begun by a start I/O instruction. When the start I/O operation begins (CAW and CCW have been fetched, and the I/O device has been successfully selected), the 'accept to CPU start I/O' signal is activated. This signal activates the 'release' signal to the CPU interface and also turns on the channel's 'channel end' latch. This latch remains on (and enables one input to an AND) until the operation is terminated and the 'interrupt request' signal is activated to the channel CPU interface. Since the 'interrupt request' signal may be activated by a channel 'interrupt' or 'log wait interrupt' signal, a second input to the channel-end AND is enabled when either of these signals is activated. Assuming the channel is not simulating storage (a test-mode operation), activation of the 'interrupt' or 'log wait interrupt' signal satisfies the AND to trigger a 350-ns singleshot. The active singleshot output raises the 'channel end' signal (for 350 ns) to the PSCE and turns off the channel's 'channel end' latch. If the channel is simulating storage ('simulate storage' signal active), the 'channel end' signal is not generated, since the channel is off-line to the PSCE. If a channel machine-reset condition occurs while the 'channel end' latch is on, the latch is turned off, preventing generation of the 'channel end' signal.

Error Sample and Advance Logic

- For channel store operations, PSCE sends channel 'error sample' signal.

- 'Error sample' signal (150 ns) occurs during period when 'address protect check' 'storage data check', or 'storage address check' signals are activated to channel (if error is detected by main storage).

- 'Error sample' signal:
  1. Suppresses channel interrupt operations while present.
  2. Activates 'raw advance pulse feature' signal which strobes AND receiving check (error) lines from PSCE.

- 'Raw advance pulse feature' signal causes:
  1. Turn-on of 'storage protect check' latch if 'address protect check' signal is active.
  2. Turn-on of 'channel data check' latch if 'storage data check' signal is active.
  3. Turn-on of 'channel control check' latch if 'storage address check' signal is active.

- When last doubleword of read operation is to be stored, channel activates 10-usec 'extend operational in' signal which holds internal 'operational in' signal active to suppress channel 'chain command' operations; this allows time for channel to receive 'error sample' signal.

- Simplex 'channel advance' signal received by channel for both store and fetch operations:
  1. Signal activates signals normally derived from receipt of 'BCU advance pulse' signal.

2. For fetch operation, check (error) signals (if any) are present at channel when 'channel advance' signal is active.
3. For fetch operation 'channel advance' signal activates 'late advance pulse feature' signal and 'raw advance pulse feature' signal which strobe the AND's receiving the three 'check' lines.

The 'error sample and advance' logic for the 2860 Version 005 (Figure 4-8) permits the PSCE to sample the channel for errors each time the channel performs a store or fetch operation.

To perform the error sampling for a store operation, the PSCE sends the channel a 150-ns 'error sample' signal. This signal is received by the channel after the channel activates the 'storage request' signal and receives the 'accept' and 'channel advance' signals from the PSCE. In addition, the 'error sample' signal is active during a portion of the period that the 'address protect check', 'storage data check', or 'storage address check' signals (if any of the check conditions are detected) are presented to the channel. Refer to the timing diagram shown on Figure 4-8.

In the channel, the 'error sample' signal: (1) prevents the channel from storing a CSW (suppresses channel interrupt operations) until the error sampling is complete; and (2) activates the 'raw advance pulse feature' signal. In turn, the 'raw advance pulse feature' signal strobes the channel AND's receiving the 'address protect check', 'storage data check', and 'storage address check' lines. If a signal is active on any of the three check lines while the 'raw advance pulse feature signal' is active, the appropriate channel latch is turned on as follows:

1. Active 'address protect check' signal causes the 'storage protect check' latch to turn on.
2. Active 'storage data check' signal causes the 'channel data check' latch to turn on.
3. Active 'storage address check' signal causes the 'channel control check' latch to turn on.

For channel operations subsequent to turn-on of any of the three latches, refer to "Error-Checking Conditions" in Chapter 6.

To prevent the channel from entering a chain-command operation until the 'error sample' signal falls, the channel activates the 10-usec 'extend operational in' signal when it is apparent that the channel must request a store cycle (Figure 4-8). The 'extend operational in' signal is activated by the 'read' signal and by the 'BC equals CTB latch' signal (last doubleword of read CCW is assembled in channel) or the 'read WLR short' signal (control unit record shorter than channel count; i.e., last doubleword is in channel). The 'extended operational in' signal holds the 'operational in' signal active within the channel for 10 usec, even though the control unit may drop the I/O interface 'operational in' signal. With the internal 'operational in' signal active, channel chain-command operations are suppressed. The 10-usec period is sufficient to allow the channel to activate the 'storage request' signal and receive the 'error sample' signal for the last doubleword of the CCW. When the 'extended operational in' signal deactivates, the channel may proceed with chain-command operations (if the operation is indicated and not terminated by a detected error).

After a channel 'storage request' signal for a store or fetch operation is honored (by channel receipt of a 'BCU response' signal), the channel receives the 'accept' and 'channel advance' signals (Figure 4-8). The 'channel advance' signal is AND'ed with the 'not simulate storage' signal to: (1) turn on the 'BCU advance' latch; (2) trigger a 200-ns singleshot to activate the 'raw advance SS' signal; and (3) for a fetch operation, to activate the 'late advance pulse feature' signal. Turn-on of the 'BCU advance' latch activates the internal 'BCU advance pulse' signal which is delayed to form the 'raw advance pulse', 'advance pulse', 'late advance pulse', and 'storage cycle complete' signals required for the basic 2860 storage cycle operations. The 'raw advance SS' signal also performs the functions normally performed by this signal for basic 2860 operations (Figure 4-8).

The 'late advance pulse feature' signal strobes the AND receiving the 'CCW fetch gated check' and 'storage data check' signals. If the latter two signals are active, main storage has detected a data parity error in a CCW being fetched by the channel. This causes the 'channel control check' latch to turn on which, in turn, causes the channel to terminate the operation.

For a data fetch operation, the channel receives the 'channel advance' and any active 'address protect check', 'storage data check' or 'storage address check' signals simultaneously. For this reason, the 'raw advance pulse features' signal is activated by the 'channel advance' signal to strobe for the check conditions during a fetch cycle initiated by the channel. (See Figures 4-8 and 4-9.) Thus, for the data fetch operations, the 'channel advance' signal causes the check conditions to be sampled; for the data store operations, the 'error sample' signal causes the check conditions to be sampled.

MODEL 85 VERSION

The 2860 Selector Channels used with the System/360, Model 85, contain minor modifications to accommodate 2860 operation with the Model 85. The 2860 basic CPU and BCU interface lines remain essentially the same, except that one line ('selective channel reset') is added, the timing of some signals is changed, and not all interface lines are used. (See "Interface Lines, Model 85 Version".) Both the BCU and CPU interface lines of the channel are connected

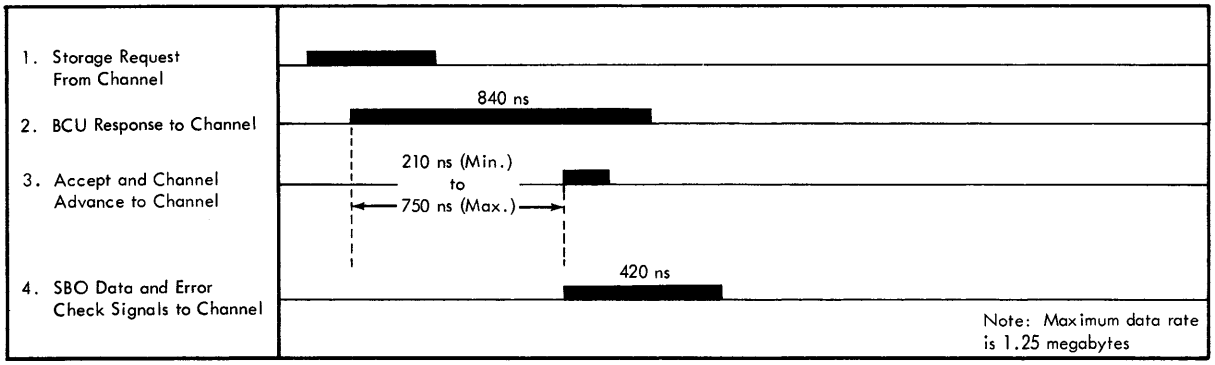


Figure 4-9. Channel Fetch Cycle Timing, Model 91

to the Model 85 Channel Adapter. The channel's I/O interface is not changed. Except for the minor modifications described in "Interface Lines, Model 85 Version" and "Selective Channel Reset", operating characteristics and capabilities of the basic channel are not changed and the basic data rate of the 2860 remains the same. The active-state polarity of signals at the channel's BCU and CPU interface is positive.

The channel used with the Model 85 may or may not contain the high-speed direct access storage priority feature, depending upon whether the channel is servicing a high-speed I/O device. (See "High-Speed Direct Access Storage Priority Feature" in this chapter.) Channel logic for the Model 85 modification is shown on the 2860 Version 008 ALD's. Logic for channels with the High-Speed Direct Access Storage Priority Feature is shown on the 2860 Version 001 and Version A18 ALD's.

Interface Lines, Model 85 Version

The basic channel CPU and BCU interface lines, described in "Channel Interfaces" in Chapter 1, are the same for the Model 85 Version with the following exceptions:

1. Storage Bus Out (SBO): consists of 72 multiplex lines (64 data, 8 parity) from the channel adapter to the channel's BCU interface. The SBO signals rise coincident with the rise of the 'BCU advance pulse' signal and after the 'BCU response' and 'accept' signals have dropped. The SBO signals remain valid until 585 ±25 ns after the rise of the 'BCU advance pulse' signal (timing change in basic 2860 BCU interface signals).
2. BCU Advance Pulse: is a simplex line from the channel adapter to the channel. The signal rises coincident with the data signals on the SBO lines or coincident with a check signal ('storage address check', 'invalid address check', 'storage data check', or 'storage protection check'). The 'BCU advance pulse' signal duration is 240 ± 60 ns. (The basic 2860 line is changed from multiplex to simplex and the signal timing is changed.)
3. Storage Address Check: is a multiplex line to the channel's BCU interface that indicates a parity error was detected in the SAB bits, storage protect key bits, or mark bits. The 'storage address check' signal rises coincident with the rise of the 'BCU advance pulse' signal and after the 'BCU response' and 'accept' signals have dropped. The signal remains active until 585 ±25 ns after the rise of the 'BCU advance pulse' signal (timing change in basic 2860 BCU interface signal).
4. Invalid Storage Address: is a multiplex line to the channel's BCU interface that is activated when the address on the SAB refers to a non-existent storage location. The signal rises coincident with the rise of the 'BCU advance pulse' signal and after the 'BCU response' and 'accept' signals have dropped. The 'invalid storage address' signal remains active until 585 ±25 ns after the rise of the 'BCU advance pulse' signal (timing change in basic 2860 BCU interface signal).
5. Storage Data Check: is a multiplex line to the channel's BCU interface. The 'storage data check' signal is activated when a data parity error is detected in information sent to main storage on a store operation or in information coming from main storage on a fetch operation. The signal rises coincident with the rise of the 'BCU advance pulse' signal and after the 'BCU response' and 'accept' signals drop. The 'storage data check' signal remains active until 585 ±25 ns after the rise of the 'BCU advance pulse' signal (timing change in basic 2860 BCU interface signal).
6. Storage Protection Check: is a multiplex line to the channel's BCU interface. This signal indicates that an attempt was made by the channel to access a protected storage area with an improper storage protect

key. The 'storage protection check' signal rises coincident with the rise of the 'BCU advance pulse' signal and after the 'BCU response' and 'accept' signals drop. The 'storage protection check' signal remains active until 585 ±25 ns after the rise of the 'BCU advance pulse' signal (timing change in basic 2860 BCU interface signal).

7. CDA Priority: basic 2860 BCU interface signal not used in the Model 85.
8. Set LCS Priority: basic 2860 BCU interface signal not used in the Model 85.
9. LCS Priority: basic 2860 BCU interface signal not used in the Model 85.
10. LCS Advance Pulse: basic 2860 BCU interface signal not used in the Model 85.
11. Selective Channel Reset: a simplex line added to the basic 2860 CPU interface from the channel adapter. The 'selective channel reset' signal is approximately 250 ns in duration. When activated, this signal causes the channel to perform a reset operation. If the channel is operating with an I/O device, the 'selective channel reset' signal also causes a 'selective reset' operation at the I/O device.
12. Channel Auto/Manual: basic 2860 CPU interface line not used by the Model 85.
13. Channel Log: basic 2860 CPU interface line not used by the Model 85.
14. Test Light: basic 2860 CPU interface line not used by the Model 85.
15. FLT Mode: basic 2860 CPU interface line not used by the Model 85.
16. Stop FLT: basic 2860 CPU interface line not used by the Model 85.
17. Start FLT: basic 2860 CPU interface line not used by the Model 85.
18. FLT Data Check: basic 2860 CPU interface line not used by the Model 85.
19. FLT Control Check: basic 2860 CPU interface line not used by the Model 85.
20. TIC Pulse: basic 2860 CPU interface line not used by the Model 85.
21. Gap Pulse: basic 2860 CPU interface line not used by the Model 85.

Selective Channel Reset

- Channel receives 200-ns 'selective channel reset' signal.
- 'Remember select channel reset' latch turns on.
- 'Interface control check SS' signal (300 ns) activates.
  1. 'Select out' latch turns off, if on.
  2. 'Suppress out' latch turns on, if off.
  3. 'Remember IF control SS' latch turns on.
- 'Interface reset' signal (6 usec) activates.
  1. 'Operational out' signal degated.
  2. 'Remember IF control SS' latch turns off.
- 'Machine reset' signal (6 usec) activates.
  1. Channel registers are reset.
  2. Channel triggers and latches reset.
- 'Interface reset' and 'machine reset' signals drop.
- 'Operational in' signal activated.
- Channel resumes polling and is available for I/O operations.

Except for timing changes with respect to BCU interface signal from the channel adapter, 2860 Selector Channel operations are modified only by the added 'selective channel reset' signal. (See Figure 4-10.)

When the channel receives the 200-ns 'selective channel reset' signal, the channel performs a machine-reset operation to reset the channel registers, triggers, and latches. In addition, the channel performs a selective-reset operation if operating with an attached I/O device. The selective-reset operation causes the control unit to disconnect from the I/O interface and perform a reset operation.

The 'selective channel reset' signal initiates the channel operations by (1) triggering the 300-ns interface control check singleshot, and (2) turning on the 'remember select channel reset' latch. See Figure 4-10.

The resulting 'interface control check SS' signal initiates the 'selective reset' and 'machine reset' operations by: (1) turning off the 'select out' latch, if on; (2) turning on the 'suppress out' latch if off; and (3) turning on the 'remember IF control SS' latch. If the channel is operating with an I/O device, the rise of the 'suppress out' signal and the fall of the 'select out' signal at the I/O interface indicate to the control unit than an I/O interface disconnect and reset operation is to be performed.

When the 'interface control check SS' signal times out, the 'not interface control check SS' signal is AND'ed with the 'remember IF control SS' and 'not stop' signals to trigger the 6-usec 'interface reset' signal. The 'interface reset' signal degates the 'operational out' signal to the I/O interface, allowing any I/O device operating with the channel to perform the I/O interface disconnect and reset operations.

In addition, the 'interface reset' signal resets the 'remember IF control SS' latch and is AND'ed with the 'remember select channel reset' signal to activate the 6-usec 'machine reset' signal. The 'machine reset' signal resets all channel registers and operating latches and triggers. When the 6-usec 'interface reset' and 'machine reset' signals deactivate, the 'operational out' signal is again activated. The channel is now in a reset condition and free to perform polling operations or to perform another I/O operation, if so instructed.

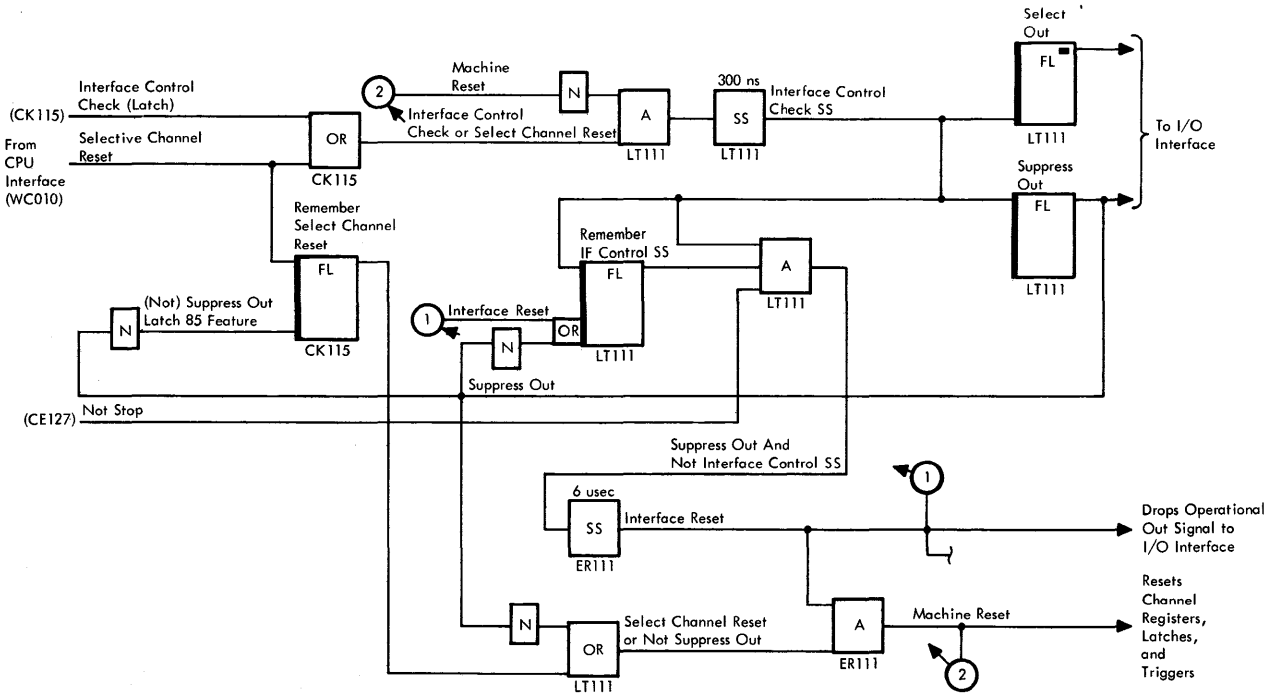


Figure 4-10. Selective Channel Reset Logic, Model 85 Version

In addition to the changes described, 2860 Selector Channels used with the Model 91 also contain 'check and parity generate' logic for the storage protect key in positions 0 through 3 of the storage protect register. This prevents a bad parity key from being sent to storage when bits 4 through 7 of the register do not equal all 0's. In addition, the 'check and parity' logic prevents the channel from detecting a channel control check condition (rather than a chain check condition) when bits 4 through 7 of the CAW are not all 0's on the CAW fetch cycle.

CHANNEL INDIRECT DATA ADDRESS (CIDA) FEATURE

- The CIDA feature allows the 2860B to indirectly address 2,048 (2k) byte blocks of storage.
- Channel operation is altered in CIDA mode.
- The CIDA feature can be installed on any 2860B.

Description

The CIDA feature makes the 2860B compatible with the Relocate feature of the System/370 Models 165II and 168 by changing the function of the Data Address field of the CCW.

Normally, the Data Address field of the CCW is used to access storage for data, but, with the CIDA feature installed, the channel is capable of using the Data Address field (Figure 4-11) as a pointer to a list of words called the Indirect Data Address List (IDAL). Each word of the IDAL contains the address of a 2k-byte data block of storage that can be addressed indirectly by the channel .

The hardware associated with this feature is a physically indistinguishable part of the channel. This feature may be installed on any 2860B Selector Channel.

Functional Characteristics

The functional characteristics for the System/370 2860 Selector Channel (2860B) remains unchanged by the addition of the CIDA feature, with the following exceptions.

1. The CCW is modified by adding the IDA flag (bit 37).
2. An address register is added.
3. The function of the Log On Machine Check switch is changed.
4. Bit 40 is added to Log Word 3.
5. A new Data Address register gating line is added (SBO 40-61).
6. Two CE panel indicators are added.

Indirect Data Address (IDA) Flag

Bit 37 of the CCW is the IDA flag; it was previously required to be zero. CCW's, with and without the CIDA feature, are shown in Figure 4-11. If the IDA flag is equal to a 1, bits 8-31 of the CCW specify the address of the Indirect Data Address List (IDAL). If the IDA flag is equal to a 0, or if the feature is not installed, bits 8-31 specify the data address.

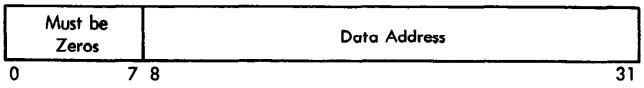
Conditions for program checks remain the same, with the following exceptions.

1. Bit 37 equal to a 1 does not cause a program check.
2. Bits 30 and 31 of the IDAL address must equal 0, or a program check occurs.

Indirect Data Address List (IDAL)

The IDAL (Figure 4-11) is a series of words (addresses) called Indirect Data Address List Words (IDALW's); each word must be on a word boundary. A program check occurs and the operation is terminated if the channel attempts to fetch an IDALW that is not on a word boundary. Each IDALW addresses a 2k-byte data block in main storage. The first word of the list is addressed by bits 8-31 of an IDA CCW; each succeeding IDALW is addressed by using the current IDALW address plus four.

Indirect Data Address List Word (IDALW)



The IDALW is 32 bits long. Bits 8-31, the data address, point to a 2k-byte storage block; bits 0-7 must equal 0, or a program check occurs and the operation is terminated.

Indirect Data Address Register (IDA Reg)

The CIDA feature is designed to allow the channel to indirectly address storage using an IDAL (Figure 4-11). A register is added to hold the IDAL address and any succeeding IDALW addresses; this register is called the IDA register (IDA reg).

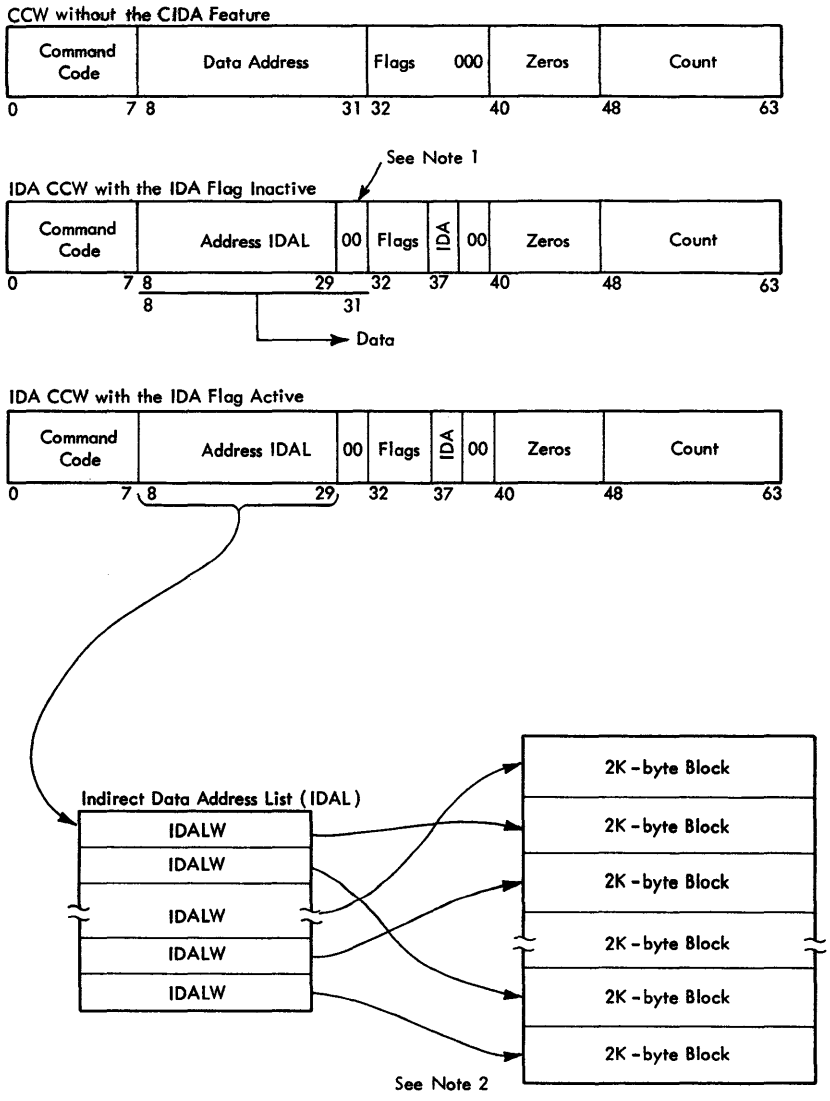
The IDA reg (Figure 4-12) is a 22-bit register; bits 0-20 ingate from the adder and outgate to the data address reg. IDA reg bit 21 is set directly from data address register bit 21. For the first IDALW fetch, IDA reg bit 21 is set to a 1 if data address register bit 21 is equal to a 1, or to a 0 if data address register bit 21 is equal to a 0. IDA reg bit 21 is then alternately set and reset (reset and set) for each succeeding IDALW fetch.

When an IDA CCW is received (Figure 4-11) with bit 37 = 1, the channel gates data address register bits 0-20 unchanged through the adder and into the IDA reg. IDA reg bit 21 is set per bit 21 of the data address register. The data address register is then used to fetch the first IDALW of the IDAL. IDA reg bit 21 determines whether SBO bits 8-31 or bits 40-61 are gated to the data address register.

When data crosses a 2k-byte boundary, another IDALW fetch is required. The current IDALW address is gated from the IDA reg to the data address register and to the adder for updating (+8). The address is returned to the IDA reg and to the data address register if IDA reg bit 21 is a 1; if bit 21 is not a 1, the adder is not outgated and IDA reg bit 21 is set to a 1. The data address register is then used to fetch the next IDALW.

Log On Machine Check Switch (IDA/Data Address Register Select Switch)

The name and function of the log on machine check switch are changed by the CIDA feature. The IDA/data address register select switch selects the register that is displayed in what was previously data address indicators only. With the switch



Notes:
1. Bits 30 and 31 need not equal zero if the IDA flag is inactive.
2. The first IDALW of an IDAL can point to any byte of any 2K-byte block of storage. Succeeding IDALWs must point to the first or last byte (rd bkwd) of any 2K-byte storage block.

Figure 4-11. Indirect Data Addressing

in the up position, the IDA address is displayed; with the switch in the down position, the data is displayed.

Channel logout occurs regardless of the switch position. The channel always stores three log words in locations 130, 138, and 140 (hex) when a machine check is detected.

Log Word 3

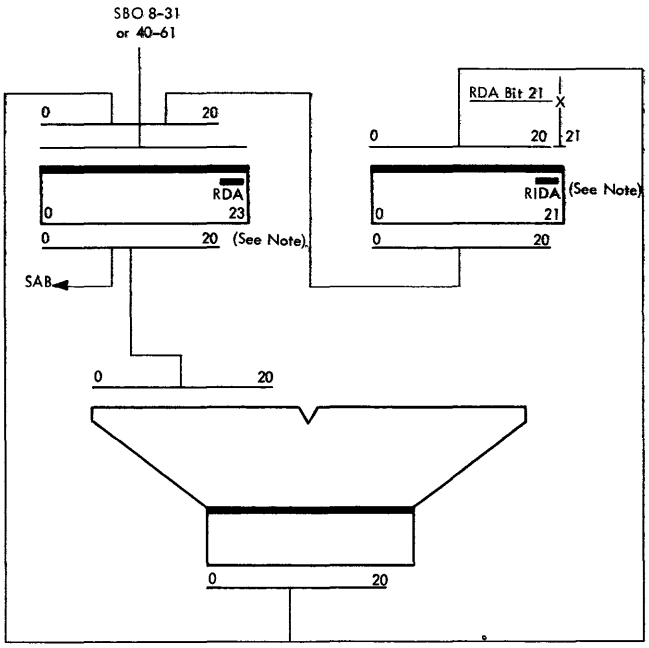
Bit 40 of log word 3, previously unused, is now used to indicate that an IDALW fetch was in progress at the time an error was detected.

New SBO to Data Address Register Gate

An IDA channel operating in IDA mode uses the data address register to receive the IDALW from storage. Because the IDALW can be on a word boundary, a gate from SBO bits 40-61 to the data address register is required. This gate is activated if IDA reg bit 21 is on when an IDALW fetch is made.

CE Panel Indicator Additions

Two CE panel indicators are added by the CIDA feature: the Initial IDA Latch (F17) and the 2k Latch (F18). The



Note: CE panel indicators D1-D27 are used to display the IDA address and the data address registers. With the IDA/data address select switch in the up position, the IDA address register is displayed; with the select switch in the down position, the data address register is displayed.

Figure 4-12. IDA Data Flow

initial IDA latch indicator lights when the channel is fetching the first IDALW. The 2k latch indicator lights when data has crossed a 2k-byte boundary, and the channel is fetching the next IDALW.

Operational Characteristics

Channel operations remain unchanged by the addition of the CIDA feature. With or without the CIDA feature, the 2860 performs I/O operations identically; the only difference is how the channel determines its data address. For channels with the CIDA feature, bit 37 of the CCW (the IDA flag) is sampled. If the IDA flag is active, an IDALW fetch is made. The IDALW supplies the data address for the I/O operation. For channels without the CIDA feature, or for channels with the CIDA feature and the IDA flag inactive, the data address in the CCW is used to access storage for data during I/O operations.

Data processing is the same for either channel unless a 2k-byte data boundary is crossed. If a 2k-byte boundary is crossed and the byte count equals 0, an IDALW fetch is necessary to obtain the address of the next 2k block of data. If a 2k-byte boundary is reached and the byte count equals 0, or if the IDA feature is not installed and the byte count equals 0, the channel stops data transfer and the operation terminates normally.

Write, Read, Control, and Sense Operations

In write, read, control, and sense operations (Figure 4-13), the storage addresses used for data transfers are in ascending order. As data transfer operations proceed, the data may cross a 2k-byte boundary (for example, bits 21-31 of the data address register change from 1's to 0's). When a crossing occurs, the next IDALW points to the first byte of the next 2k-byte block from which data is to be transferred. The second, and all subsequent IDALW data addresses must have bits 21-31 equal to 0's to be on a 2k-byte boundary; if not, a program check occurs and the operation is terminated.

Because write operations prefetch data, and because write operations can be terminated before the prefetched data is used, an IDA data address program check does not occur until the channel attempts to use the data from the IDALW fetch that caused the program check condition.

Read Backward Operations

Storage addressing for read backward operations is in descending order. As data transfer operations proceed, the data may cross a 2k-byte boundary (for example, bits 21-31 of the data address register change from 0's to 1's). When this occurs, the next IDALW points to the end of the next 2k-byte block with which data transfer is to occur. The second IDALW, and all subsequent IDALW's, must have bits 21-31 equal to 1's to be at the end of a 2k-byte boundary; if not, a program check occurs and the operation is terminated.

Performance Characteristics

When the CIDA feature is used, additional delay occurs in the data path to the I/O device. The delay is due to the channel time required to perform the IDA function, and to the additional storage reference required. The delay occurs on any CCW fetch when bit 37 of the new CCW is a 1, or when a 2k-byte block crossing is detected in the data field. The probability of overrun (both service and command) and chaining checks is increased.

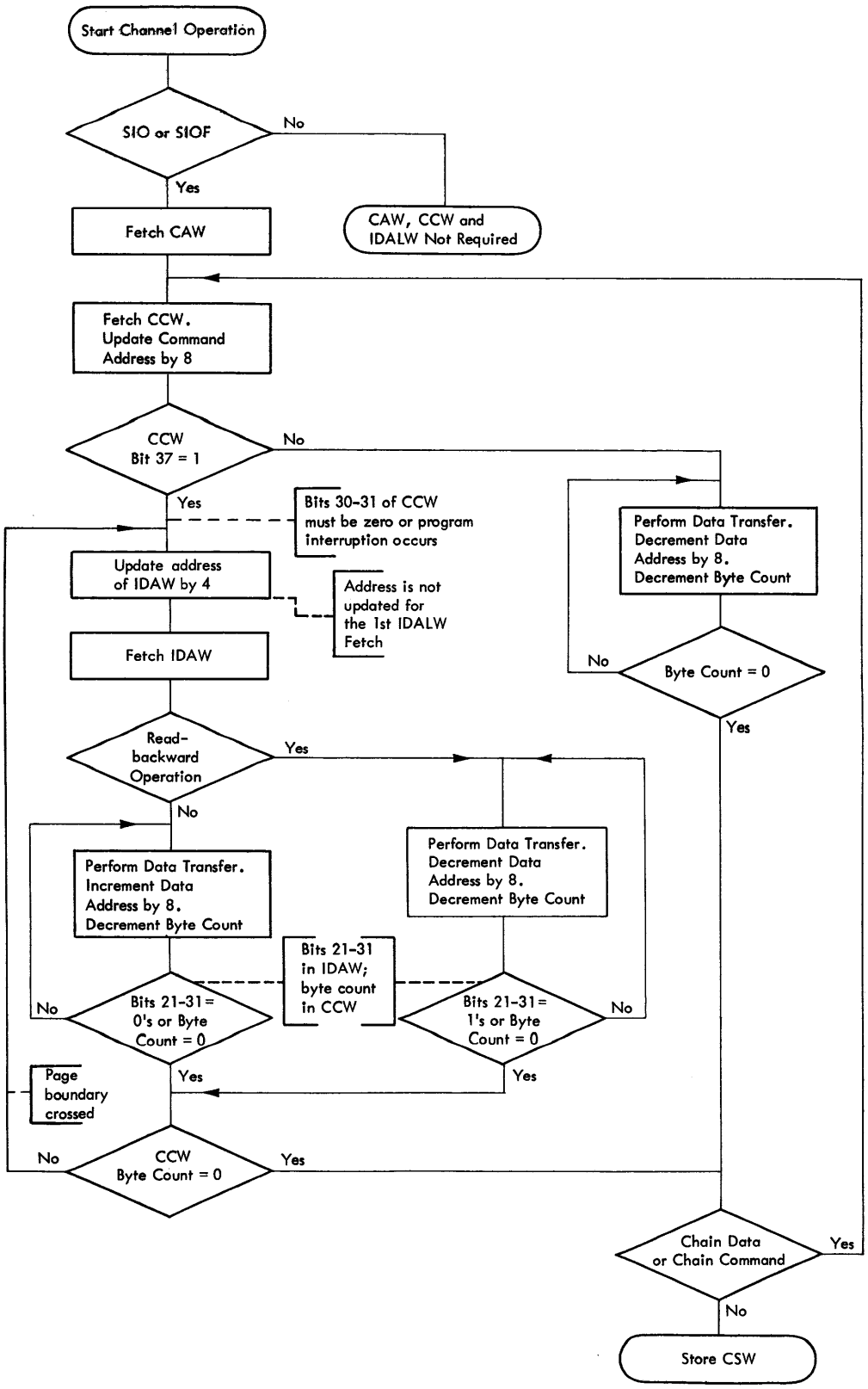


Figure 4-13. IDA Flowchart

- Power is supplied to logic from dc power modules in power compartment.
- Power can be controlled at channel frame or remotely by system power on control.
- Voltages are distributed from output of regulators to each logic gate.

This chapter discusses power supplies, power distribution, and power controls in the 2860 channel. The chapter is divided into the following parts: (1) input power requirements; (2) power components, which discusses components such as power supplies, stepping relays, etc.; (3) power distribution throughout the 2860; (4) power control interface between the 2860 and units connected to it; (5) power-on sequences, which describes the application of power to the 2860; (6) power-off sequences, which describes the removal of power from the 2860; and (7) marginal checking, which is a troubleshooting aid.

Power is supplied to the logic components of the 2860 frame from dc power modules housed in the power compartment between channel gates 2 and 3 (see logic YD911). In a one-channel frame, the power compartment included sequencing controls, a bulk transformer, three regulators, a bias supply, contactors, circuit breakers, and other components as shown on logic YD911. For frames with two or three channels, extra +3V and +6MV supplies are included for each additional channel. Input voltage can be 50 or 60Hz as specified under "Input Power Requirements".

Power can be controlled at either the channel frame or remotely by system power-on control. This is selected by the LOCAL/REMOTE switch on the power control panel. The channel POWER ON pushbutton is only active for local control, but the POWER OFF pushbutton is enabled for both local or remote control.

Power is distributed from the output of the regulators on individual cables to each logic gate. An upper and lower terminal board on each gate feeds the upper and lower horizontal laminar buses. These, in turn, feed the vertical laminar buses which supply the voltages and dc return to the logic.

INPUT POWER REQUIREMENTS

Primary power requirements for each 2860 frame are dependent upon whether 50 or 60 Hz are available and whether a four-wire delta system or a five-wire Y system is used. The options are:

60 Hz (± 1 cps), 3-phase, 4-wire delta, 208V ac or 230V ac (+10, -8%).
50 Hz (± 1 cps), 3-phase, 4-wire delta, 195V ac, 220V ac, or 235V ac (+10, -8%).
50 Hz (± 1 cps), 3-phase, 5-wire Y, 380V ac or 408V ac (-10, +8%).

The amount of power consumed by the frame depends upon number of attached channels, as follows:

	Power Input	Power Factor (Min)	Service Rating (Amp)	Heat Dissipation BTU/hr
Channel/Frame	3.05	0.7	15A	8,250
Channels/Frame	3.65	0.7	15A	10,000
Channels/Frame	4.25	0.7	15A	11,600

POWER COMPONENTS

Logic Power Supplies

A maximum of seven regulators (MPS supplies) in each channel frame supply voltage to the 2860 logic gates. The characteristics of these regulators and their application are:

Part Number	Quantity	Output	Amp
5392280	1 per channel (3 maximum)	+6MC Vdc	32
5247230	1	-3V dc	24
5234378	1 per channel (3 maximum)	+3V dc	32

The 208V ac (hereafter used as an example input) is applied to the regulators through CB's 1 and 5, contactors K15, K16, and K14 (logic YD801) and transformers T2 (logic YD 811). Note that taps are provided on the input of the bulk supply (T2) so that the supply may be used with any of the available input voltages as shown on logic YD811. The operation of each dc logic supply is described in SLT Power Supplies, FEMI, form 223-2799.

Relay Power Supply

A relay power supply (logic YD801) provides control, EPO, and indicator voltage for the 2860. This supply, PN 5351120, provides an unregulated +24V dc output and is rated at 4 amperes. A 208V ac input is provided to the supply through CB5. Fuses F1 and F2 provide overcurrent protection. Note that several taps are provided on the input of the supply so that the supply may be used with any of the available input voltages according to the following scheme:

Input Hertz	Voltages (ac)	Phase	Tap	Common (TB 1-1)
50	235	2	TB1-8	Phase 1
50	195	2	TB1-5	Phase 1
50	220	2	TB1-7	Phase 1
50	408 (Y)	2	TB1-8	Neutral
50	380 (Y)	2	TB1-7	Neutral
60	208	1	TB1-6	Phase 2
60	230	1	TB1-7	Phase 2

The operation of this supply is described in SLT Power Supplies FEMI, form 223-2799.

Convenience Outlets

Each 2860 frame has one duplex convenience outlet that provides 115V ac (60-Hz input) or 220V ac (nominal) (50-Hz input). The 115V ac output is derived from transformer T3 (logic YD801) which is fed 208V ac through CB5 and EPO contactor K15. Note that taps are provided on the input to T3 so that the transformer may be used with any of the available input voltages. Fuses F3 and F4 (10 amp) provide overcurrent protection on the T3 primary. The convenience outlet on 2860 frames with 50-Hz input voltages receives its inputs directly from K15. Fuses F3 and F4 (10 amp) provide overcurrent protection on the lines from the contactor.

Convenience outlet power is available when power is off at the frame, provided input power is applied, CB5 has not tripped, and the EPO contactor is closed.

Cooling

Cooling is provided by circulating room air vertically through the gates. Each of the three channel gates has two single-phase blowers, and a single-phase blower is located below the regulators. Input to the blowers is 208V ac through CB5 and contactors K15 and K16 (logic YD801). Overcurrent protection to all blowers is provided by fuse F5. In addition, each set of gate blowers is individually protected by 0.65 amp, two-pole circuit breakers. Note that the blowers and circuit breakers for gates 2 and 3 are provided only when channels 2 and 3 are provided.

Power and Marginal Checking Controls

The manual controls and indicators associated with power are contained on the power control panel which is mounted on the front of the frame. The operation of these controls and indicators is discussed in Chapter 6 and in this chapter where they affect power sequencing. Marginal checking is provided with both local and remote control. However, it can be done on a frame basis only.

Sequencing Step Switch

The sequencing step switch (Figure 5-1) is a 26-position, ratchet-driven assembly that sequences power on to control units attached to the 2860 when channel power is turned on. The switch (S6) has 26 contacts and rotating wipers which are double-ended and establish contact between common and each of the contacts twice in 360 degrees of rotation.

Switch S6 has three decks of contacts. Deck A (S6A) advances the switch. As the step switch driver, L1, attracts its armature to advance the switch, the driver energizing circuit includes the deck A contact that will be broken by the switch advance. The interrupter contact breakers the driver circuit before the wiper loses contact with the S6A contact. Mechanical inertia and the shorted back-circuit in L1 carry the armature and ratchet-drive forward to the next position. The armature returns, the ratchet enters the next tooth, and the interrupter contact recloses.

The start interlock contact (logic YD891) returns the step switch to start position (26) whenever dc power goes on or off. Thus the step switch starts from the same position when channel sequences control units on through the B and C decks.

Marginal Check Assembly

The marginal check assembly (logic YD871) consists of three ganged motor-driven potentiometers (logic YD831-851) which raise or lower the +6M power supply voltage for performing marginal checking. The potentiometers raise or lower +6V dc at all installed channels simultaneously. (The effect is to vary transistor base bias, revealing faulty collector control.) A cam-operated contact lights an indicator on the system console when the potentiometers are in home position. A variable resistor in each +6V dc regulator enables setting the individual +6V dc potentiometer at home position. Refer to Figure 5-2.

The marginal check motor switch has three positions: center-off, raise, and lower. Either the raise or lower position starts the motor, one in one direction and the other in the opposite direction. The motor, in turn, drives the potentiometers in the indication direction. When the motor circuit opens, the potentiometers are held in that position.

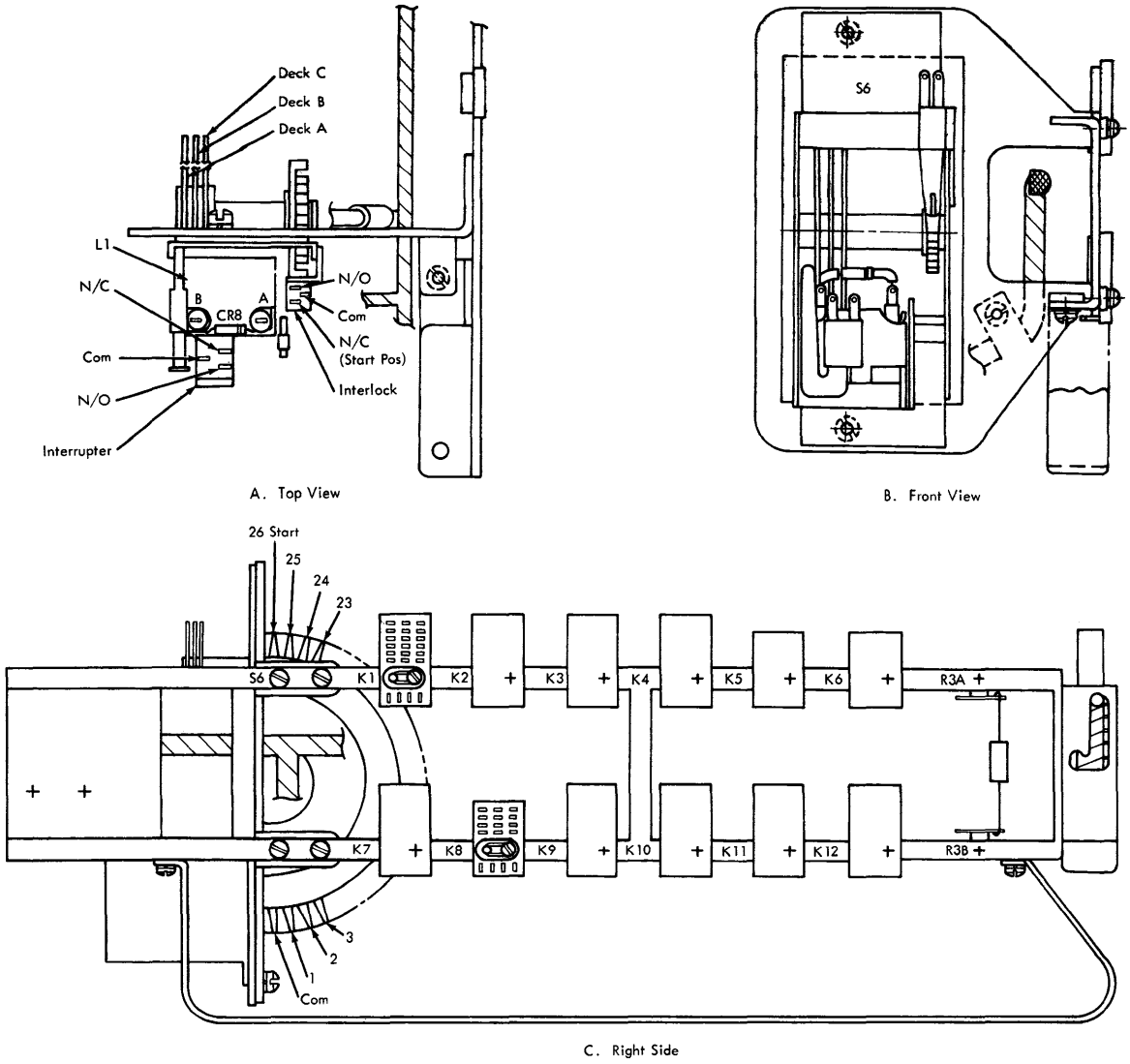


Figure 5-1. Sequencing Step Switch

Power for the marginal check assembly is 28V ac and originates in the system with the 2860 LOCAL/REMOTE switch set to REMOTE or in the 2860 with the switch set to LOCAL.

AC POWER DISTRIBUTION

The primary ac power distribution is shown in Diagram 6-1, FEMDM, for 60-Hz units and in Diagram 6-2, FEMDM, for 50-Hz units.

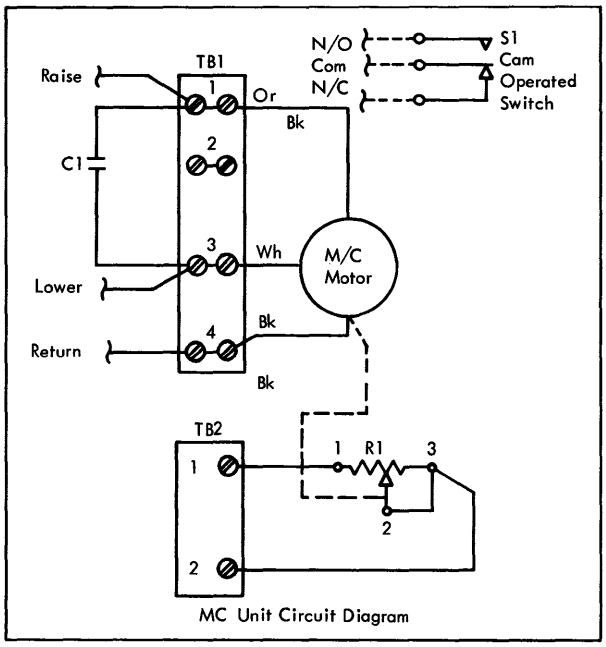
60-Hz Units

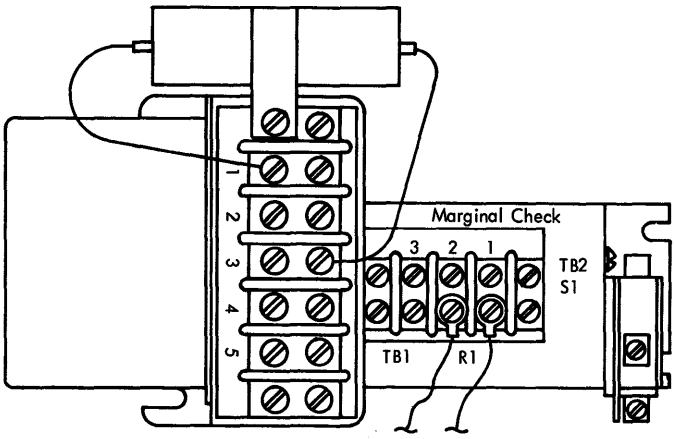
Main power from CB5 is applied to transformer T1 via fuses F1 and F2 (Diagram 6-1). T1 provides +24V dc to the sequencing controls via relay K15 contacts. The output of the 20V source supply is fed to the mid-pac regulators for the voltage sense assemblies.

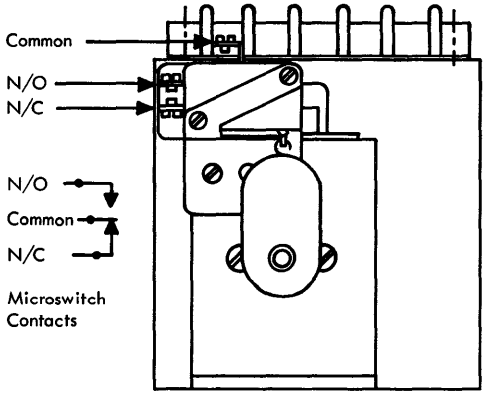
Main power is applied by emergency power-off (EPO) contactor (K15) contacts to T3 via fuses F3 and F4. Transformer T3 provides 115V ac to the convenience outlet.

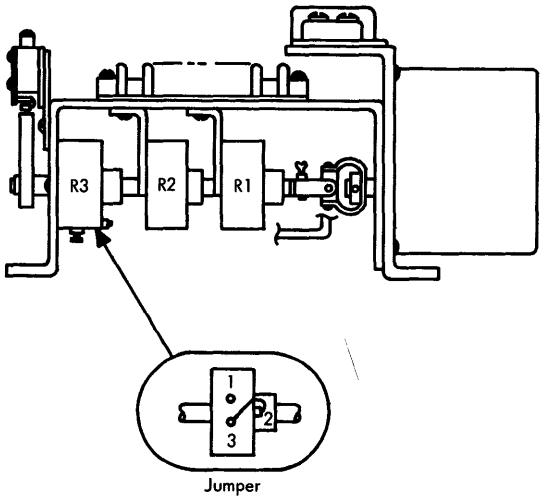
When power is turned on, relay K16 contacts apply power to:

1. The customer meter located on the power control panel.
2. The cooling fans in the power assembly.
3. The gate blowers via CB1. Each gate blower also has its own circuit breaker (CB) for overcurrent protection. Note that Diagram 6-1 shows a 2860 frame with three channels. For frames with less than the maximum number of channels, the gate blowers and associated CB's are not provided.









Note: The potentiometer shaft and the microswitch contacts are shown in the "home" position (nominal voltage output).

Figure 5-2. Marginal Check Assembly.

4. Surge control contactor K14. This contactor prevents full voltage from being applied to the bulk supply during the initial surge of power, thus preventing CB1 from tripping when power is applied.
5. The bulk supply via K14 contacts.

50-Hz Units

In 2860's supplied by 50-Hz power (Diagram 6-2), main power is applied across the three phases and across each phase and neutral, depending on the voltages at the location. Transformer taps and jumpers are adjusted accordingly.

Main power through CB5 is applied to T1 via fuse F1 and a jumper on TB7. Transformer T1 provides 28V ac for the marginal checking circuits and the 20V source supply. In addition it provides +24V to the sequencing controls via K15. The output of the 20V source supply is sent to the voltage sense circuits on the mid-pac regulators.

Main power is applied by EPO contactor (K15) contacts to the convenience outlet via TB7 and fuse F3. The jumper wire on TB7 is connected as shown on Diagram 6-2 for 195V, 220V, and 235V delta inputs. For 380V and 408V wye inputs, the jumper is placed on the opposite terminal connecting the conveience outlet between phase 2 and neutral.

When power is turned on, relay K10 contacts apply power to:

1. The customer meter located on the power control panel.

   Note: The switch block (Diagram 6-2) is shown wired for delta operation. For wye operation, pin 2 is jumpered to pin 5, pin 5 is jumpered to pin 8, pin 2 is jumpered to pin 3, and pin 3 is jumpered to pin 6.

2. The cooling fans in the power assembly.
3. The gate blowers. Each gate blower has its own circuit breaker (CB) for overcurrent protection. Note that Diagram 6-2 shows a 2860 frame with three channels. For frames with less than the maximum number of channels, the gate blowers and associated CB's are not provided.
4. Surge control contactor K14 via CB1 and the switch block. This contactor prevents full voltage from being applied to the bulk supply during the initial surge of power, thus preventing CB1 from tripping when power is first applied.
5. The bulk supply via K14 contacts and the switch block.

DC POWER DISTRIBUTION

Dc voltages are distributed identically in units using 50-Hz and 60-Hz inputs (Diagram 6-1). The bulk supply converts the primary ac input into +6V dc, +3V dc, and -3V dc. These voltages are fed to the mid-pac regulators. Frames containing only one channel have three regulators, one for each voltage. For each additional channel (up to 3), two mid-pac regulators are added to the basic frame; one +6M and one +3V regulator for each additional channel. The output of each regulator is sent to terminal boards mounted on each gate and distributed to four laminar buses (two upper and two lower). From the laminar buses the voltages are routed to the 2860 logic.

POWER CONTROL INTERFACE

Power sequencing and interlocking is provided on at least two levels in systems that include the 2860. The higher level is CPU/system unit (e.g., the 2860); the lower level is system unit (e.g., the 2860)/system sub-unit (e.g., a control unit attached to the 2860). The CPU sequences power up on one system unit at a time, giving each unit a start signal and waiting for that unit's 'sequence complete' signal before stepping to the next unit. Unit power is held on by a circuit through the CPU that is broken when CPU power goes off. Emergency power-off (EPO) is a high-level-to-low-level interlock that removes nearly all power from the entire system under control of a single CPU switch. The EPO switch provides a circuit to each system unit that is in turn passed to subunits. These controls are all part of the power control interface, used in both levels of sequencing and interlocking.

2860/CPU Power Interface

The CPU and the 2860 share the following power control interface lines (Figure 5-3):

1. Unit Source (J25-A): Unregulated 24V dc originating in the 2860 that is controlled by the CPU and returned to the 2860.
2. System EPO (J25-B): 2860 voltage returned to the 2860 by the CPU when no EPO condition exists.
3. Power Pick (J25-R): 2860 voltage returned to the 2860 by the CPU when the CPU is ready to turn on 2860 power.
4. Power Hold (J25-T): 2860 voltage returned to the 2860 by the CPU when CPU power is on; the power-hold voltage is required for the 2860 to maintain a power-on condition.
5. Sequence Source (J25-L): 24V dc sent to the 2860 by the CPU and returned to the CPU when 2860 power is on.
6. Sequence Complete (J25-M): 24V dc from the CPU; this voltage is returned to the CPU by the 2860 when 2860 power is on (including sequencing of control units). This signal advances the system sequence step switch in the CPU to allow power to be applied the next unit.

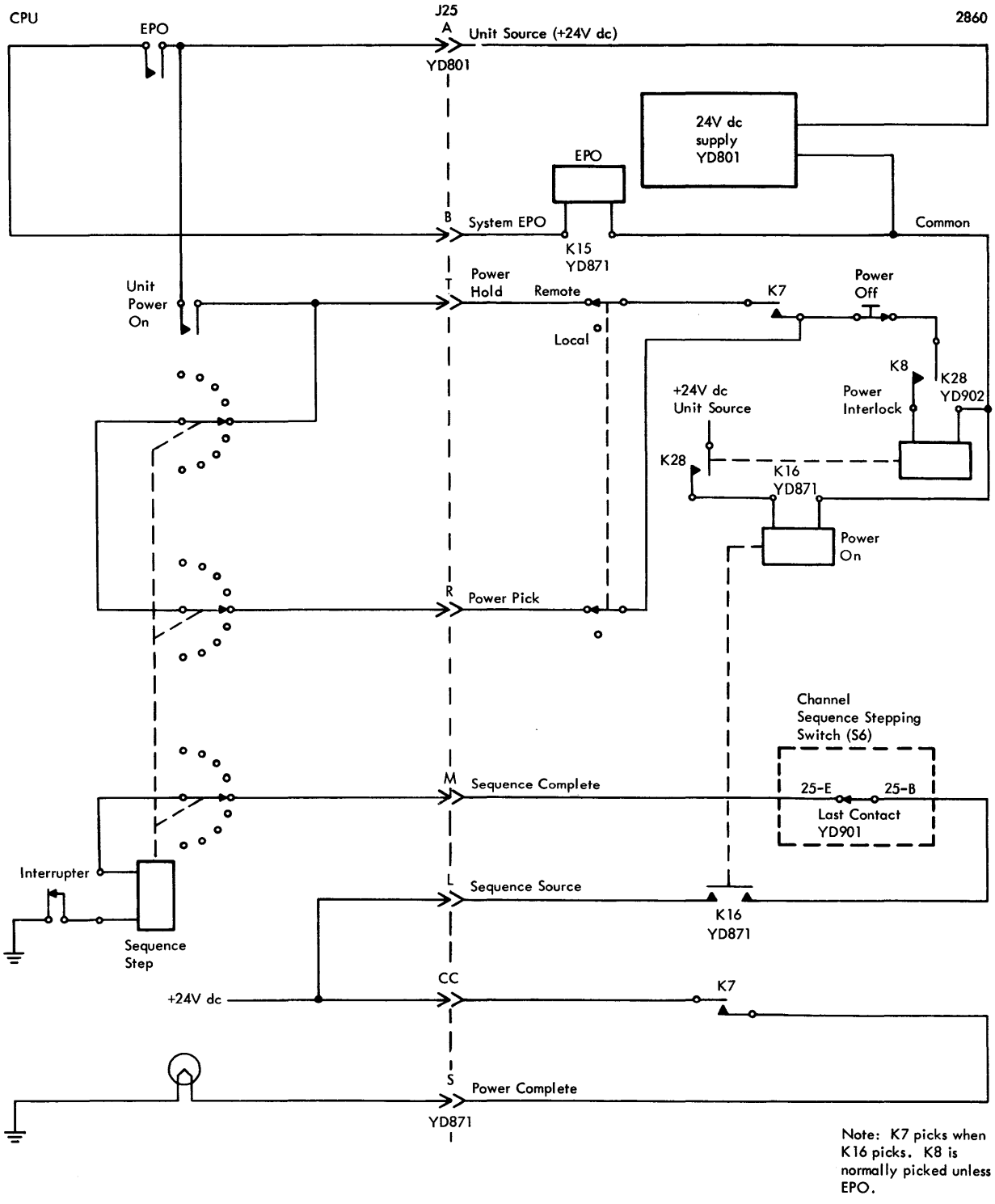


Figure 5-3. 2860/CPU Power Interface

7. Power Complete (J25-S): 24V dc from the CPU; this voltage is returned to the CPU by the 2860 when 2860 power is on to light an indicator on the system control panel.

2860/Control Unit Power Interface

For each control unit attached to the 2860, a power sequence control interface exists which is similar to the interface between the 2860 and the CPU. Power is turned on to each control unit in sequence starting with the control unit attached to jack 1 (J1) and continues up to a maximum of 24 control units (J24). Figure 5-4 shows the interface for a typical control unit. Unused jacks must have pins (3) and (4) jumpered to allow the sequence stepping switch to advance.

The 2860/control unit power interface is as follows (using control unit 1 as an example):

1. Unit Source (J1-1): Unregulated 24V dc that originates in the control unit and is controlled by the 2860. This signal is returned to the control unit.
2. System EPO (J1-2): Control unit voltage that is returned to the control unit by the 2860 when no EPO condition exists.

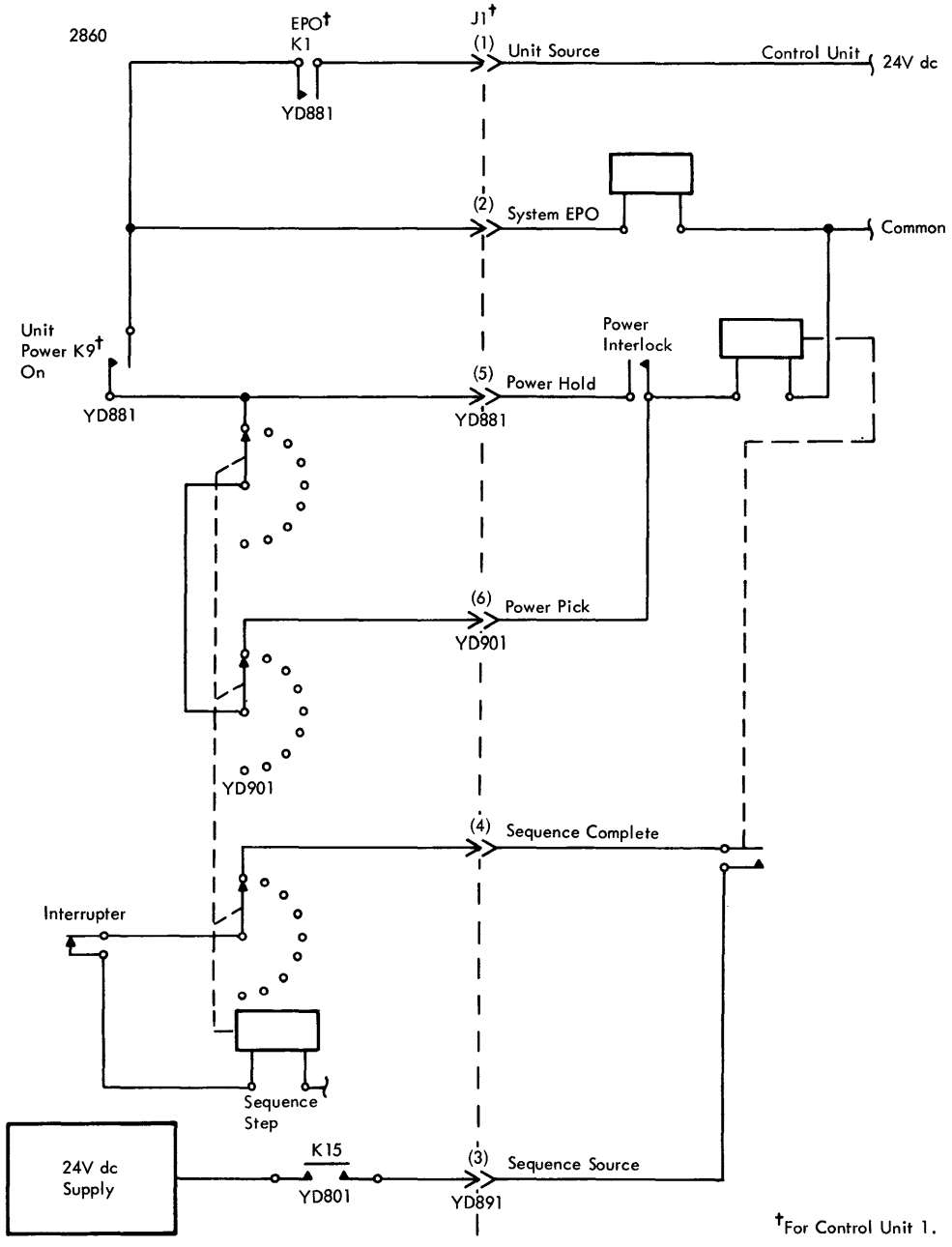


Figure 5-4. 2860/Control Unit Power Interface

3. Power Pick (J1-6): Control unit voltage returned to the control unit by the 2860 when the 2860 is ready to turn on control unit power.
4. Power Hold (J1-5): Control unit voltage returned to control unit by the 2860 when 2860 power is on. It holds control unit power on.
5. Sequence Source (J1-3): 24V dc from the 2860; this voltage is sent to the control unit to be returned to the 2860 when control unit power is on.
6. Sequence Complete (J1-4): 24V dc from the 2860; this voltage is returned to the 2860 by the control unit when control unit power is on. This signal advances the sequence step switch in the 2860 to allow power to be applied to the next control unit. When the last control unit has power up, this signal is sent back to the CPU as the sequence-complete voltage for the 2860.

POWER FAULT PROTECTION

The 2860 logic is protected from electrical damage by devices that turn power off if a power fault occurs. These devices sense for the following conditions:

1. Overcurrent monitored at the output of the mid-pac regulators.
2. Overvoltage monitored at the output of the mid-pac regulator.
3. CB trip monitored at the output of the bulk supply, the input to the blowers, and the input of the bulk supply.
4. Thermal trip monitored at the regulators and the logic gates.

Note: The above conditions, in addition to turning off power, are indicated by lights on the 2860 power panel and by lighting a power check light on the system control panel. They also cause the system power-on indicator to turn from white to red.

5. Loss of regulator voltage monitored at the mid-pac regulators.

Overcurrent Protection

Overcurrent sensing is performed by circuit breakers on the output of each regulator. Any fault that draws excessive current from any regulator trips these CB's and causes the 2860 to drop power. Figure 5-5 shows the overcurrent sense loop.

When an overcurrent condition exists, continuity is broken between the normally closed auxiliary contacts of the circuit breakers in the malfunctioning regulator. Thus, the power-interlock is dropped, causing 2860 power to cycle down. At the same time, the normally open contacts of the regulator sensing circuit are closed, providing +24V dc to light the CB indicator.

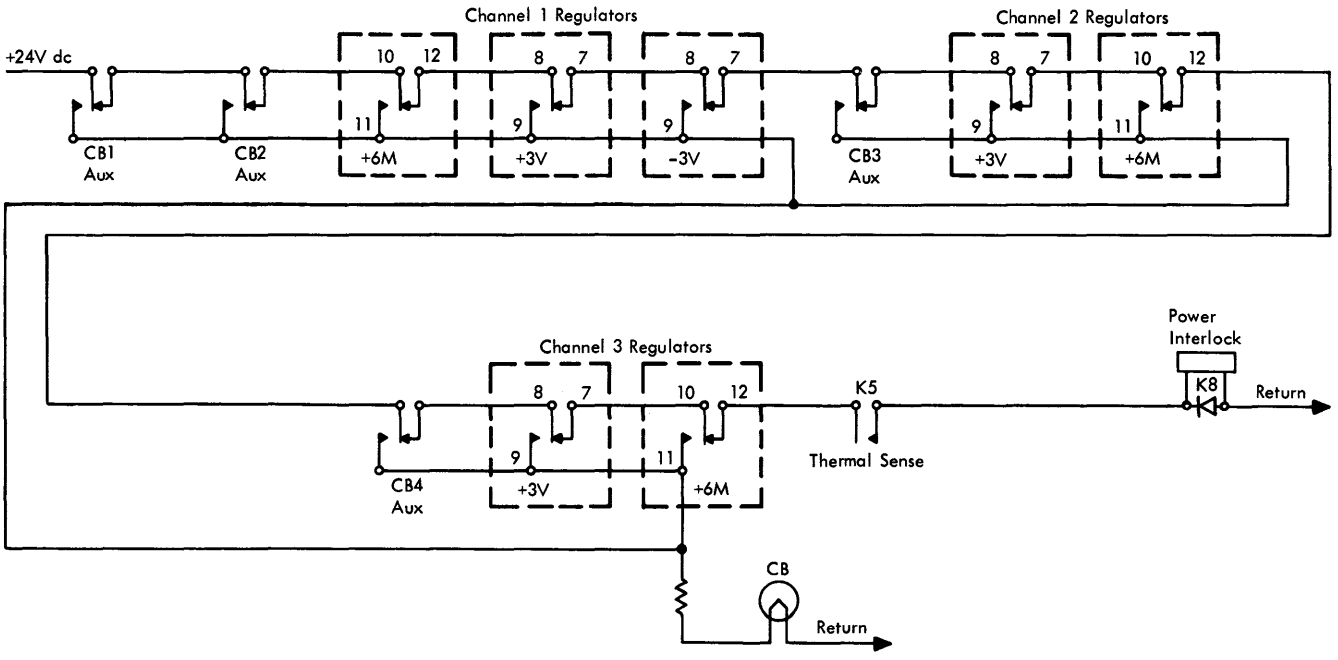


Figure 5-5. Overcurrent Protection Loop

Overvoltage Protection

The overvoltage sensing and protection circuits are internal to each regulator. Any fault that raises the output voltage level of any regulator above the maximum causes 2860 power to drop via the overcurrent sense loop in the same manner as an overcurrent condition. The CB indicator also lights.

Circuit Breaker Protection

If the output of the bulk supplies, the gate blowers, or the input to the bulk transformer draws excessive current, a circuit breaker trips, causing power to drop. CB1 is associated with the bulk supply; if this CB trips, input power is immediately removed from that supply. CB's 2, 3, and 4 are associated with the gate blowers for gates 1, 2, and 3. respectively. The main contacts of each of these CB's immediately disconnect the faulty blower. CB's 1 through 4 also have auxiliary contacts in the overcurrent protection loop which drop power interlock relay K8 and turn on the CB indicator.

Thermal Protection

Thermal protection is provided by placing sensing elements in the hold path of the thermal sense control relay (Figure 5-6). Normally, with power on at the 2860, relays K5 and K6 are both energized. The hold path for K5 is through all the thermal sense switches and its own points. If the temperature in any of the regulators or any logic gate exceeds 134°F, the associated thermal switch opens, causing K5 to drop. When K5 drops, the THERMAL indicator lights and K8, the power interlock relay (Figure 5-5), drops and causes power to cycle down.

When the temperature falls to a normal level, K5 remains de-energized since K6 is picked. Therefore, to restore power, either the 2860 THERMAL RESET or the system THERMAL RESET pushbutton must be depressed. Depressing one of these pushbuttons drops K6. K5 then picks through the normally closed contacts of K6, extinguishing the THERMAL indicator and providing a pick path for K8. With K5 energized, K6 again picks when the THERMAL RESET pushbutton is released and holds through its own contacts and K5's normally open contacts.

Undervoltage Protection

The undervoltage sensing circuits provide fault protection and an indication that the output of each mid-pac regulator is above the minimum voltage level (2.4V). Any fault that lowers the output voltage of any regulator below the minimum level causes 2860 power to drop. The undervoltage relays have contacts in the return path of relay K7 (Figure 5-7) which normally pick at the completion of the power-on sequence. When an undervoltage condition occurs, either K1 or K2 de-energizes, causing K7 to de-energize. K7 points then open, de-energizing K9, the power-on relay.

A line from each positive regulator is fed to an input of an AND (CR 4-11 and R8). The output level of the AND

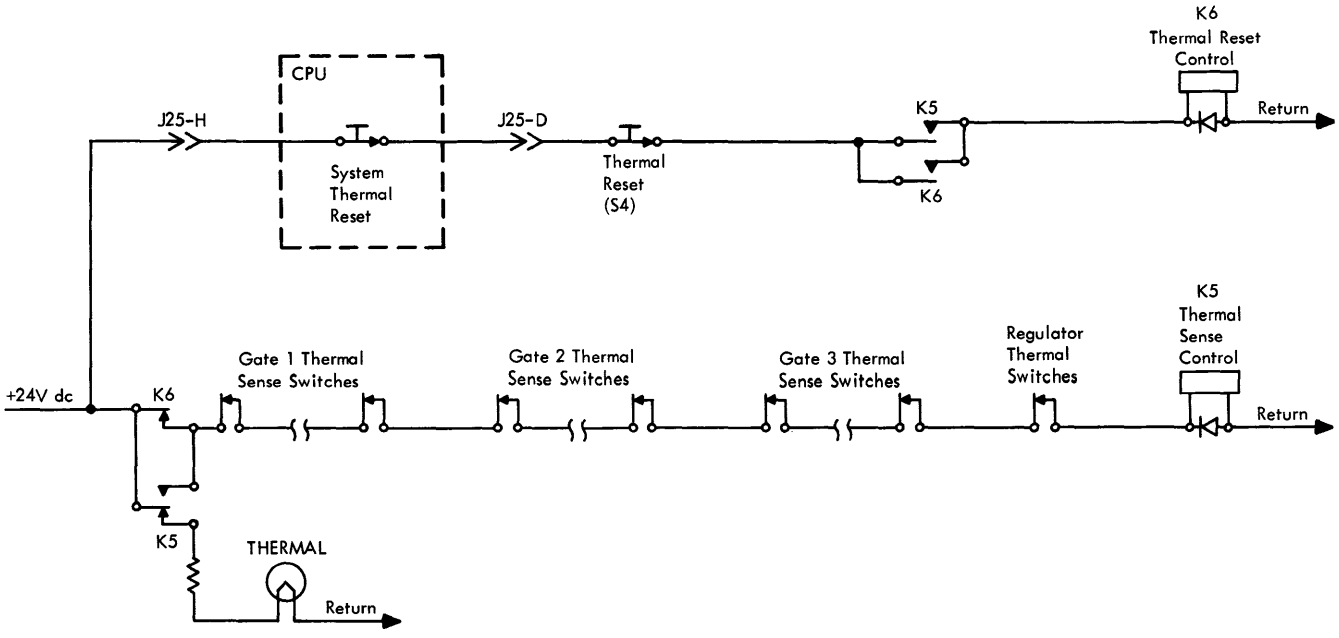


Figure 5-6. Thermal Protection Circuits

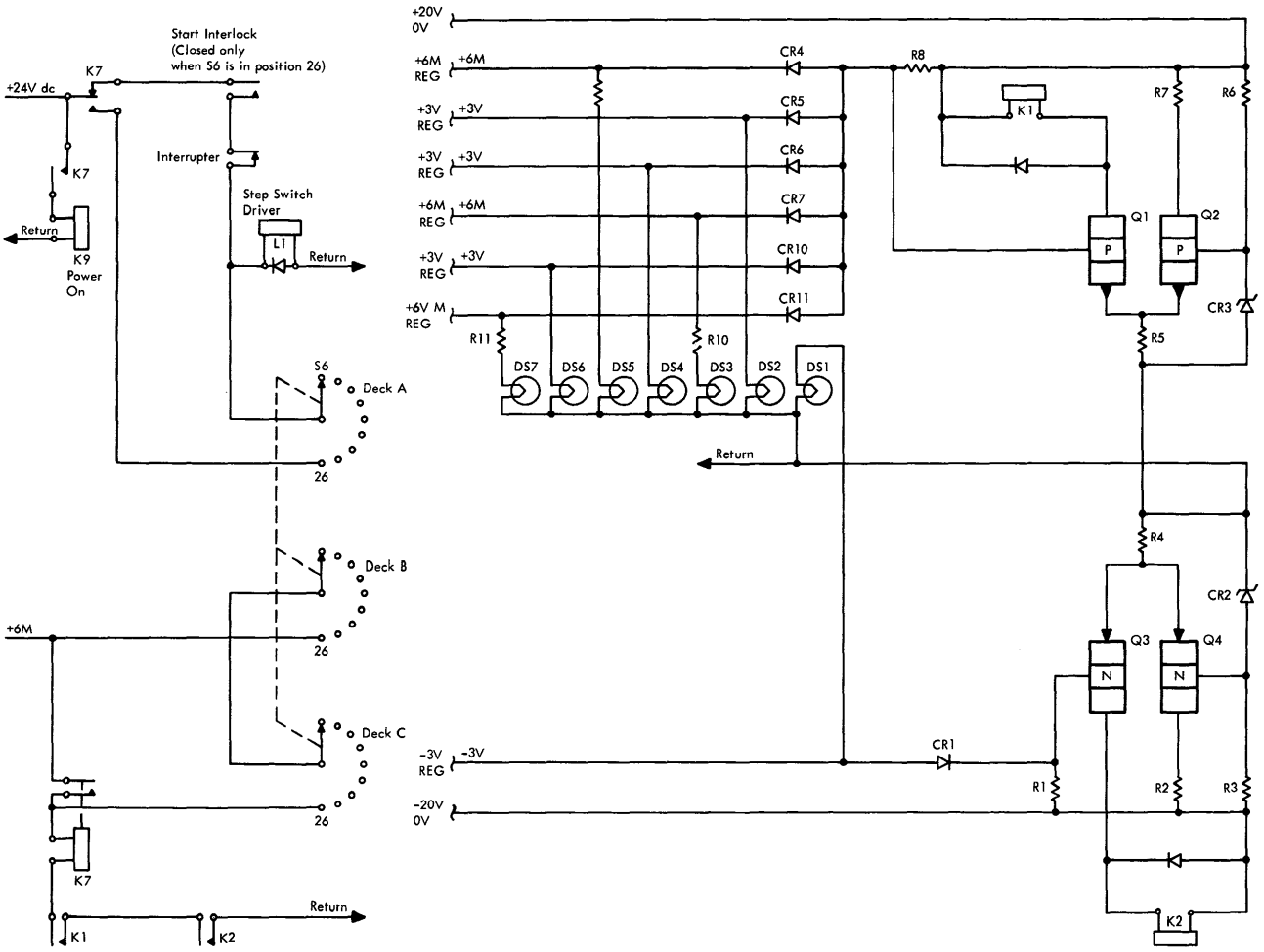


Figure 5-7. Undervoltage Protection Circuit

is determined by the lowest input level; that is, the one nearest zero or ground. Thus, if any supply is off, the output level of the AND is near zero. The output of the AND is transferred directly to the base of transistor Q1, which shares an emitter load resistor with Q2. The base of Q2 is held at 2.4V by Zener diode CR3 and resistor R6. Transistor Q2 now carries the full emitter load current and holds the common emitter circuit to 2.4V. With its emitter held positive with respect to its base, Q1 is biased off. As the output level of the AND raises the base of Q1 to 2.4V, Q1 begins to conduct and starts to share with Q2 the current drawn by the emitter load resistor. As the output level of the AND raises the base of Q1 above 2.4V, Q1 raises the emitter circuit above 2.4V. With its emitter raised to a level positive with respect to its base, Q2 is biased off. Transistor Q1 now carries the full emitter load current; this current is sufficient to pick K1.

A similar circuit, of opposite polarity, checks the negative regulators and picks K2.

POWER CONTROL CIRCUITS AND POWER SEQUENCING

Power control circuits (Diagram 6-3, FEMDM) control the application and removal of power to the 2860 and its attached I/O control units. These circuits sequence power on and off, detect power faults, and communicate power status information to higher-order (CPU) and lower-order (control units) units of the system. The eight main operations executed by power control circuits are:

1. Initial status sequence.
2. Power on from the 2860 power control panel.
3. Remote power on (initiated from the system control panel).
4. Power off from the 2860 power control panel.
5. Remote power off (initiated from the system control panel).
6. Power off due to a CB trip, overcurrent, or thermal condition.

7. Power off due to an undervoltage condition.
8. Emergency power off.

The control circuts apply power only on signal from the system or from the power panel, but may remove power for various reasons: (1) power off from the system, (2) emergency power off from the system, (3) power off from the power panel, (4) overloaded ac circuit or dc supply, or (5) overheated logic gate or power compartment area (sensed by thermal switches). These power control inputs switch 24V dc to energize or de-energize power control relays, which in turn switch 24V dc to energize or de-energize contactors. 24V dc-powered indicators are provided for the power-on, overload (CB trip), and overtemperature (thermal trip) conditions. When all controlled power is removed (exception; CB5, Diagram 6-1, FEMDM), 208V ac is still present in the channel frame and the 28V ac/24V dc sources are still active.

Initial Status Sequence

- With ac power applied to frame and CB5 closed, initial status sequence establishes initial conditions necessary to turn power on.

Before dc power can be applied to the 2860, ac input power must be connected: 28V ac, 24V dc, and 20V dc must be available; and certain control relays must be energized. When these criteria are met, the 2860 power logic is in an initial status. The following sequence establishes an initial status:

1. 50-Hz or 60-Hz input power is available at the input to the frame.
2. To start the sequence that establishes initial status, close CB5.
3. Closing CB5 applies ac input power to EPO contactor K15 contacts and the 24V dc/28V ac power supply (Diagram 6-1).
4. 24V dc is sent to the system EPO contacts via J25 (Figure 5-3).
5. If no EPO condition exists at the system source, the 24V dc is returned to the 2860 and picks the EPO contactor K15 (Diagram 6-3).
6. With K15 energized, 115V ac (on 60-Hz units) is applied to the convenience outlet via transformer T3 (Diagram 6-1).
7. K15 points also apply 24V dc to power control logic (Diagram 6-3). This action applies 24V dc to K5 (via K6 normally closed points and the thermal sense switches), lights the THERMAL indicator on the power panel (via K5 normally closed points), applies 24V dc to K1-4, applies 24V dc to K29, K35, K36, K37, and K30, and applies 24V dc to the sequence-step-switch driver coil if the driver coil is not in the home position (position 26).
8. K5 now energizes unless one of the thermal sense switches is open due to an overtemperature condition or K6 is energized because of a power shutdown caused by a previous overtemperature condition. If the second condition exists (indicated by the THERMAL light staying on), depressing the THERMAL RESET pushbutton on the channel power control panel or the system control panel will drop K6, allowing K5 to pick. K5 now holds through its own normally open contacts and causes the following to occur:

   a. Turns off the THERMAL light.
   b. Energizes K6.
   c. Energizes K8 if no CB is tripped and no overcurrent protection device has tripped. (K8 normally open points will allow power to be brought up to the unit.)

9. K1-K4 energize. These relays allow the unit source voltage from the control units to be fed back to the control units as a system EPO signal (Figure 5-4).
10. K29 and K30 energize. These relays and K31 are only present when the channel-to-channel adapter feature is installed on the channel. Energizing K30 ensures that the bypass relay and receiver relay for the 'select out' signal (Figure 5-8) are de-energized while power is being applied to the 2860 and until the receiver relay has energized. This ensures that any 'select out' signal issued by the other channel connected to the channel-to-channel adapter is propagated while power is being applied to the 2860 from which the channel-to-channel adapter receives its power.
11. At the same time that K29 picks, K35, K36, and K37 pick. These three relays prevent transients from developing on the I/O interface-out lines during a power up or power down on the channel. Figure 5-9 shows how each relay controls the gating for one channel gate; therefore, K36 and K37 are installed only when their respective channel gates are present. Refer to K38 as an example. When it picks, the drive gate line for the OR's shown on ALD AS113 is switched from the +6M supply to ground. This condition holds down the OR circuits so that the OR output stays negative; thus, a false interface signal is prevented from being sent out on the multiplex line to the control units. These relays drop when power is sufficiently stabilized after a power-up sequence.
12. Applying 25V dc to the sequence-step-switch device via K7 normally closed contacts causes the sequence step switch to step to its home position (S26) in preparation for a power-up sequence. This completes the initial status sequence.

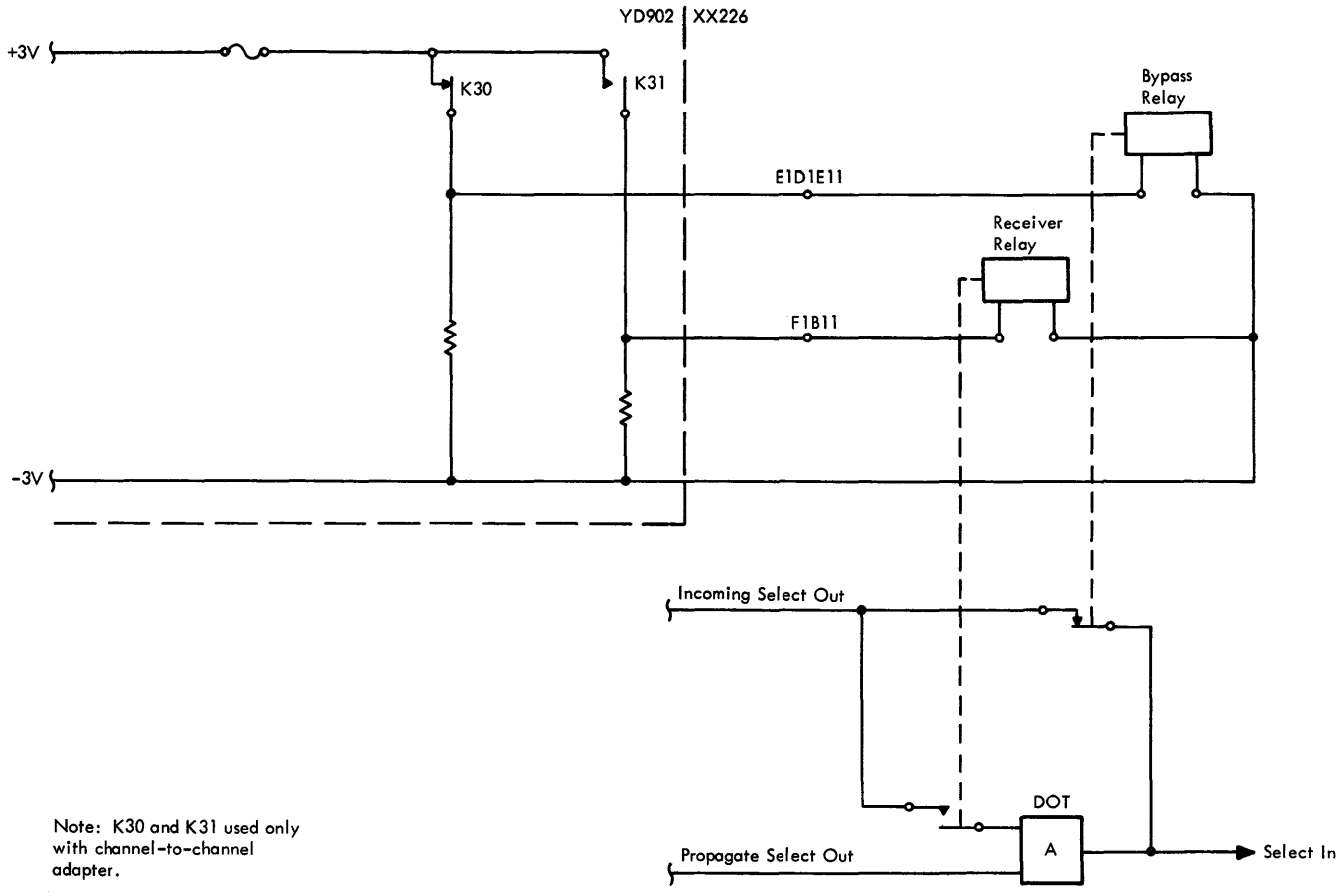


Figure 5-8. Power Sequencing Control of Select Out Propagation (Channel-to-Channel Adapter Feature Only)

Power-On Sequence

- With LOCAL/REMOTE switch in REMOTE, dc power is brought up to 2860 remotely when system power is turned on.
- With LOCAL/REMOTE switch in LOCAL, power is brought up by depressing POWER pushbutton on 2860 power panel.
- K28 picking initiates power-on sequence.

DC power in the 2860 may be brought up in two ways: (1) remotely from the CPU, or (2) locally by depressing the POWER ON pushbutton on the 2860 power panel. The method used to apply power is determined by the position of the LOCAL/REMOTE switch. If the switch is in LOCAL, power can only be applied from the 2860 power panel. If the switch is in REMOTE, power can only be applied from the CPU. In either case the power-on sequence is the same and starts by energizing K28 (Diagram 6-3). The only difference between the two methods is the source of the +24V dc needed to pick and hold K28. If the LOCAL/REMOTE switch is in REMOTE, +24V dc is applied to K28 from the CPU via the 2860/CPU power interface (Figure 5-3). If the LOCAL/REMOTE switch is in LOCAL, depressing the POWER ON pushbutton applies +24V dc from the EPO control in the CPU to K28 (Diagram 6-3). The remainder of the sequence for either power-on method is as follows:

1. K28 energizes via the normally closed points of the POWER ON pushbutton and the normally open points of K8. K8 is the power interlock relay and is energized during the initial status sequence if no power fault exists (see "Initial Status Sequence").
2. K28 picking picks K16, applies +24V dc to K34, which is a 2-second time delay relay, and opens one of the two paths that apply voltage to K30. K30, however, remains energized since K29 is still energized.
3. K16 is the main power contactor and applies 208V ac (Diagram 6-1) to:
   a. Meter power pac
   b. Power supply fans
   c. The gate blowers via CB1-CB4
   d. Surge control contactor K14

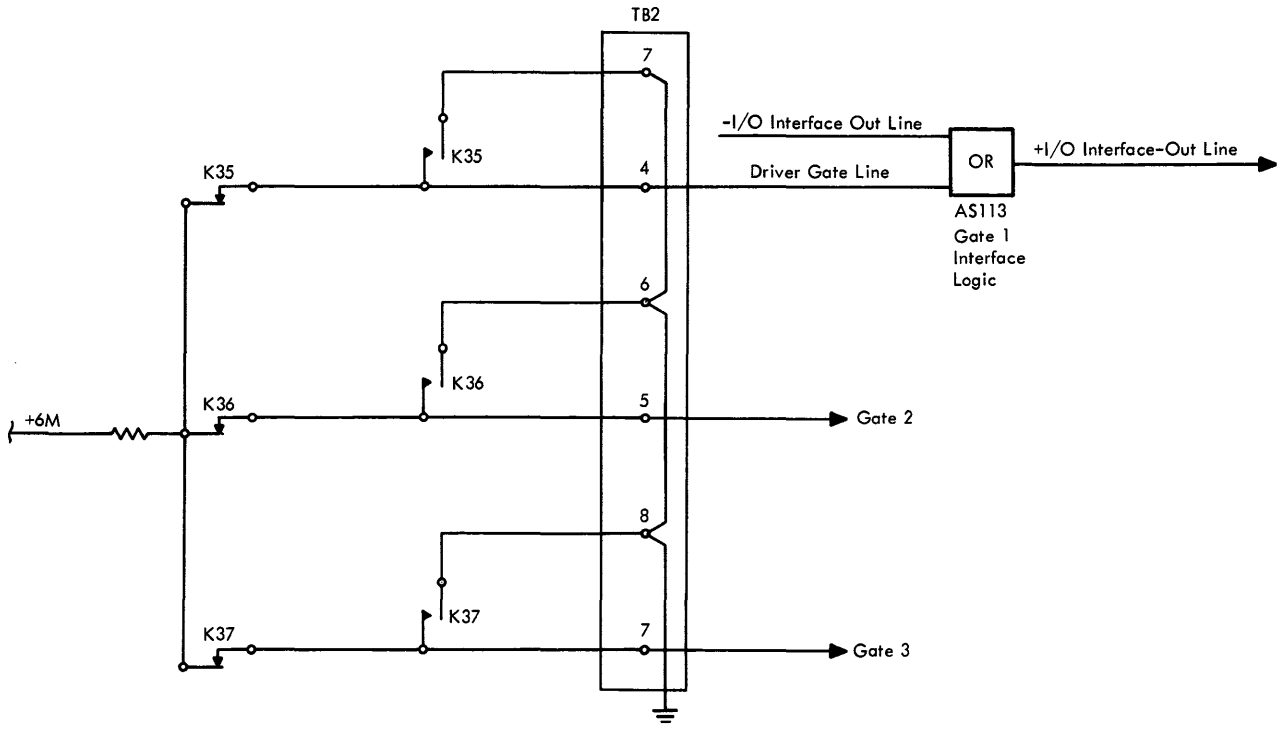


Figure 5-9. Power Sequencing Control of I/O Interface

4. Reduced 208V ac is applied to the bulk power supply via the resistors in parallel with K14 points. This pre-energizes the bulk supply until K14 picks. Full power is not applied at first so that any power surge at the input of the bulk supplies will not affect the mid-pac regulators. When K14 points transfer, 208V ac is applied to the bulk supply transformer. The bulk supply output is the various dc voltages needed for the logic gates. These voltages, however, are unregulated.
5. The mid-pac regulators accept the unregulated dc from the bulk supply and apply regulated dc to the channel gate logic.
6. When the +3V dc regulator comes up, a +3V dc power-on reset signal is sent to the logic circuits to reset all 2860 channel logic (Diagram 6-3). This signal falls when K7 picks.
7. When all the regulators are at their full potential, the undervoltage sense relays (K1 and K2) pick, providing a return for K7 (Figure 5-7).
8. Since the sequence step switch is at its home position (26) because of the initial status sequence, +6V is applied to K7 via position 26 of wafer C, through the wiper to a common contact, down to wafer B, through its wiper, and contact (26). K7 holds through its own normally open contact.
9. With K7 energized, the POWER ON light is turned on, a 'power complete' signal is sent to the CPU to light an indicator, K9–K12 are picked, and the sequence step switch is stepped to position 1. K9–K12 and the sequence step switch are used to sequence power up to the control units. The operation of these circuits is discussed beginning with step 13.
10. Meanwhile, K34, the 2-second time delay relay which allowed the regulators to reach full potential before picking, has by this time energized. Energizing K34 picks K31.
11. Energizing K31 drops K29, K35, K36, and K37. K35, K36, and K37 dropping removes the I/O interface hold-down condition placed on these lines (step 11 of the initial status sequence). K31 points also energize the receiver relay (channel-to-channel adapter feature only), thus allowing the 'select out' signal to be propagated.
12. Dropping K29 drops K30, which in turn energizes the 'select out' bypass relay (channel-to-channel-adapter feature only), thus preventing 'select out' signals from bypassing the channel.
13. Meanwhile, when the sequence step switch is stepped to position 1 (see step 9), control unit sequencing begins. A 'power pick' signal is sent to control unit 1. This signal picks a power-on relay in the control

unit. When control unit power is up, it returns a 'sequence complete' signal to the channel, which steps the sequence step switch to the next position.

14. Power is brought up on the remaining control units in the same manner as control unit 1. Unassigned positions of the stepping switch are wired so that when those positions of the switch are made, the switch steps to next position immediately.
15. When all control units have power on, the sequence step switch has stepped to position 25, causing a 'sequence complete' signal to be sent to the CPU. The 'sequence complete' signal to the CPU (remote power on only) causes the system step switch in that unit to step to the next position, allowing power at other units controlled by the CPU to be brought up.
16. At the end of the sequence, K1–K12, K15, K16, K28, K31, and K34 are picked; K29, K30, K35, K36, and K37 are dropped; and the sequence step switch is in position 25. The POWER ON light is on and control units have power on.

Normal Power-Off Sequence

- Power can be turned off when system power is turned off if LOCAL/REMOTE switch is in REMOTE.
- Depressing POWER OFF pushbutton always cycles down power regardless of LOCAL/REMOTE switch setting.
- Dropping K28 initiates power-off sequence.

DC power may be turned off normally (i.e., not by an emergency power off or a power fault) in two ways: (1) remotely from the CPU, or (2) locally by depressing the POWER OFF pushbutton on the 2860 power panel. Power can only be turned off from a remote unit if the LOCAL/REMOTE switch is in REMOTE. However, the POWER OFF pushbutton will always cycle down regardless of the setting of the LOCAL/REMOTE switch. In either case, the power-off sequence in the 2860 is the same and begins by de-energizing K28 (Diagram 6-3). The remainder of the normal power-off sequence is as follows:

1. K28 dropping picks K30 (select-out bypass relay; channel-to-channel adapter feature) and drops K34.
2. K34 dropping picks K29, K35, K36, and K37 and drops K31.
3. K31 dropping drops K16, the main power-on contactor. This immediately removes 208V ac from the bulk supply, all fans and blowers, and from the meter power pac, and drops K14. In addition, the 'sequence complete' signal to the CPU is suppressed.
4. Since the mid-pac regulators are no longer supplying voltage, the voltage sense relays (K1 and K2) drop.
5. K1 and K2 points open the K7 return path, dropping K7.
6. With K7 de-energized, K9–K12 drop, the POWER ON light is turned off, and the 'power complete' signal is suppressed.
7. K9–K12 relays dropping causes power to cycle down on the attached control units.
8. With K7 dropped, the sequence step switch returns to its home position (26). The sequence ends with K1–K6, K8, K15, K29, K30, K35, and K37 energized, and the sequence step switch in the start position. The POWER ON light is off and control unit power is off. Note that this is the same initial status as after the initial status sequence.

Power-Off Caused by Misperformance

The power control circuits will turn off dc power if any of the following power fault conditions occur:

1. CB1, 2, 3, or 4 trips.
2. An overcurrent or overvoltage condition is detected by the voltage sense circuits in the mid-pac regulators.
3. Any of the thermal sense circuits in the channel gates or the power area detects a temperature in excess of 134°F.
4. The voltage of any mid-pac regulator drops below 2.4V dc.

These sequences differ only slightly from a normal power-off sequence. If a CB trips, a CB trip indicator lights and remains on until the CB is reset. A CB trip will drop K8, which in turn drops K28. The remainder of the power-off sequence is identical to a normal power-off.

If an overcurrent or overvoltage condition is detected by the voltage sense circuits, no indicators are turned on at the power control panel. However, K8 will drop, thus dropping K28. The remainder of the power-off sequence is identical to a normal power-off.

If a thermal sense circuit opens, indicating that the temperature exceeded 134°F in one of the gates or the power area, K5 is dropped and the THERMAL indicator is turned on. Dropping K5 drops K8, which in turn drops K28. The remainder of the power-off sequence is identical to a normal power-off.

If the voltage of a mid-pac regulator drops below 2.4V dc, K1 or K2 in the undervoltage protection circuits drops, and the indicator on the voltage sense assembly turns off. Dropping K1 or K2 drops K7. Dropping K7 drops K9–K12 and K28. With K9–K12 dropped, power is immediately turned off at the control units. The remainder of the power-off sequence is the same as a normal power-off.

Emergency Power-Off

The system emergency power-off switch, when operated, causes the system contact that holds K15 to open. Thus, K15, channel EPO, drops. K1–K4 drop and relay the EPO condition to the control units. All power and power controls drop in the channel and in the attached control units. The 24V dc remain on to repick K15 when the system EPO contact closes again. Notice that the difference between power-off and EPO is that EPO takes power off the convenience outlets and the power control relays.

The sequence is as follows:

1. K15 drops.
2. 208V ac is removed from the bulk supply, all fans and blowers, the meter power pack, and from the convenience outlets. In addition, +24V dc is removed from all power sequence relays.
3. 24V dc, 28V ac, and 20V dc sources remain on.

Emergency power-off ends with no relays energized, the sequence step switch in whatever position it was when the EPO switch was depressed, control unit power off, and no indicators on.

MARGINAL CHECKING

The +6M mid-pac regulator may have its output level varied ±2V from the nominal output. This maintenance aid allows testing of critical circuits with nonstandard voltages as an aid in predicting failures. Applying margins varies logic transistor base bias which should reveal faulty control of transistor collector current.

The status of each channel frame 6M voltage can be monitored at the system control panel or by plugging a meter into jacks on the channel power control panel. When a margin is applied, it is simultaneously applied to all channels in the frame. The 6M levels can be margined on a per frame basis from the system control panel or channel frame.

When the channel frame is in local, margins can be applied only at the channel frame; when in remote, only at the system control panel. In either mode, they can be monitored at both the system control panel and the channel frame. Indication of the 6M regulator in its margin status is sent to the system control panel. Indication of the 6M volt regulator being at nominal (or home) position is also sent to the sytem control panel. Neither indication is shown at the channel frame.

Margins are applied by a motor-driven rheostat controlled by the RAISE/LOWER switch on the power control panel (Figure 5-10). The rheostat is driven by the motor which is energized by either a raise or lower signal. Limits are set by mechanical stops on the rheostat. When a limit is reached, the rheostat stops, but the motor remains energized until the RAISE/LOWER switch is returned to its neutral position. The rheostat is connected to the 6M volt regulators control circuits, causing the 6M volts to raise or lower.

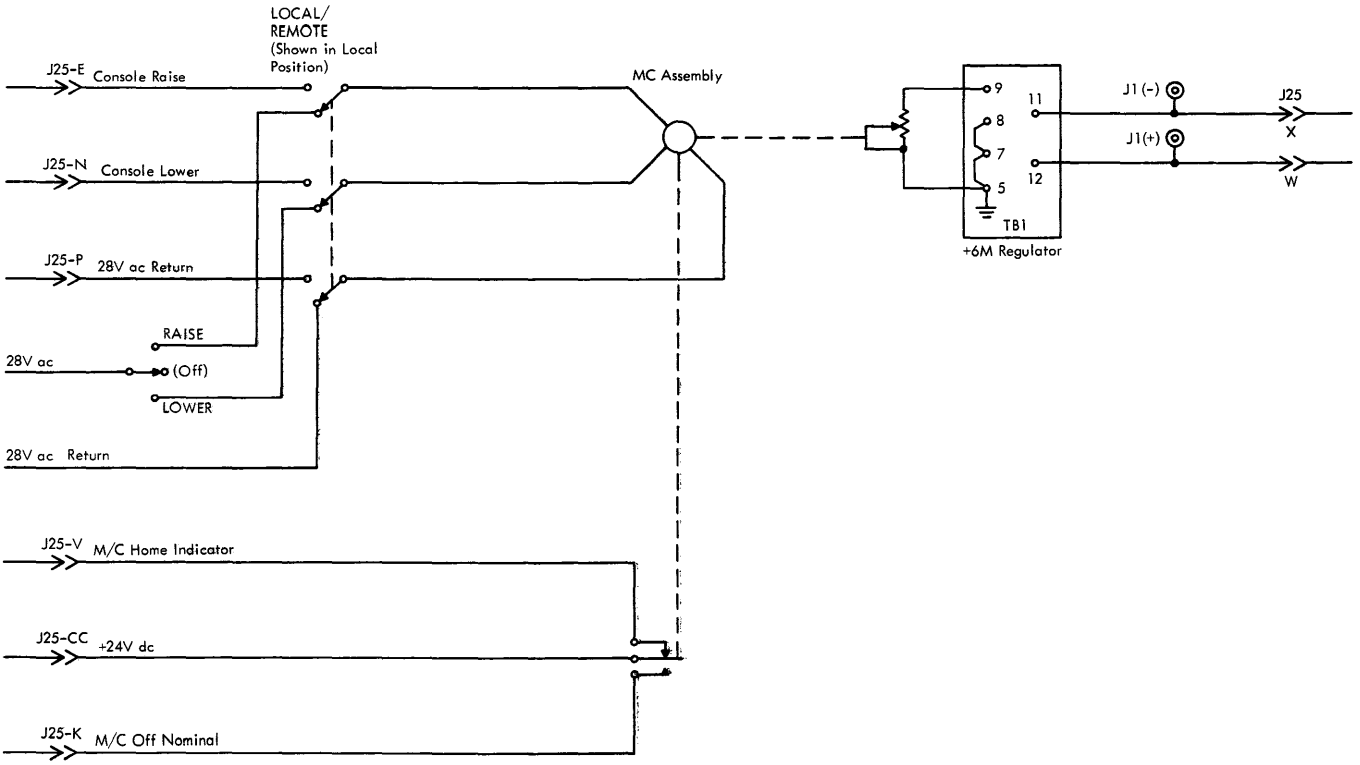


Figure 5-10. Marginal Checking

CHAPTER 6. CONSOLE AND MAINTENANCE FEATURES

This chapter describes the 2860 Selector Channel CE panel, its associated test panel logic, and channel maintenance features. This chapter is divided into two sections. Section 1 describes the principles of operation of the CE panel controls and indicators and their related circuitry. Section 2 describes maintenance features of the channel, including descriptions of the logout, error checking, status byte, and marginal check capabilities of the channel.

SECTION 1. CONSOLE

This section describes (1) the overall principles of operation of the CE panel and associated logic at the block diagram level, (2) the principles of operation of the individual CE panel switches and associated logic at both the block and positive logic diagram levels, and (3) the principles of operation of the CE panel indicators. References are made to figures in this chapter, as well as to diagrams in the FEMDM for the 2860 Selector Channel.

CHANNEL CE PANEL OPERATING PRINCIPLES

- Automatic-mode operation:
  1. CE panel AUTO/TEST switch to AUTO.
  2. Channel operates on line with CPU control.
  3. All CE panel indicators operable.
  4. Only LOG ON MACH CHK and TEST IND switches operable.
  5. When on, LOG ON MACH CHK switch allows logout operation when channel machine check condition is detected.
  6. TEST IND switch may be placed on to test all CE panel indicators except A- and B-register bit indicators (parity bit indicators for these registers are tested.)

- Test mode operation
  1. CE panel AUTO/TEST switch to TEST.
  2. Channel operates off line to CPU control, with CE panel switch settings determining operations performed.
  3. All CE panels indicators and switches operable.

- The following operations normally initiated or controlled by CPU can be simulated at channel in test mode:
  1. Start I/O
  2. Test I/O
  3. Halt I/O
  4. Clear interrupt (interrupt response)

- IPL, test channel, FLT, diagnostic and logout operations cannot be simulated. (LOG ON MACH CHK switch on causes channel operation to stop when channel machine check or channel data check condition is detected.)

- Simulate-storage operation causes channel to simulate signals normally received from BCU interface; BCU interface in and out lines are inhibited at channel.

- Not-simulate-storage operation allows channel to access main storage; BCU interface in and out lines are not inhibited.

- Manual store operation initiates channel store operation; data is gated to A-register from CE panel SBO switches.
  1. For simulate-storage operation, data is not stored in main storage.
  2. For not-simulate-storage operation, data is stored in main storage.

- Manual fetch or start I/O write operation initiates channel fetch operation(s).
  1. For simulate storage operation, data is fetched from CE panel SBO switches.
  2. For not-simulate-storage operation, data is fetched from main storage.

- Simulate-I/O-interface operation causes channel to simulate signals normally received from control units; I/O interface in and out lines are inhibited at channel.

- Not-simulate I/O interface operation allows channel to operate in normal manner with attached control units.

- For simulate-I/O-interface operations, simulate interface register and associated control logic.
  1. Provides means (during device selection) of simulating address from control unit; address register bits gated to bus-out latches, bus-out bits gated to 'simulate interface register control' logic, then to bus-in latches.
  2. For write operations, simulate interface register receives bytes gated to bus-out latches; retains last byte of write operation.
  3. For read operations, static contents of simulate interface register are gated to B-register via bus-in latches.

- For I/O device selection, unit address is gated to unit address register from CE panel UNIT ADDRESS switches.

The channel CE panel (Figures 6-1 and 6-1A) and associated logic provides the channel with the capability of monitoring automatic-mode operations and of testing and monitoring test-mode channel operations.

For automatic-mode operation (CE panel AUTO/TEST switch to AUTO), all CE panel indicators are operable and provide visual indications of channel operations; however, all CE panel switches except the LOG ON MACH CHK (log on machine check) switch and the TEST IND (test indicator) switch are inoperable. The LOG ON MACH CHK switch, when on, causes the channel to perform a logout operation if a channel machine check condition is detected during the course of a channel operation. The TEST IND switch, when on, should cause all CE panel indicators except the A-REGISTER and B-REGISTER bit indicators to light (parity indicators for these two registers should light); if an indicator does not light, the indicator or indicator driver may be defective. Placing the TEST IND switch on does not affect channel operation.

For test-mode operations (CE panel AUTO/TEST switch to TEST), all CE panel indicators and switches are operable. In the test mode (Figure 6-2), the channel is off-line to the CPU and, with the available CE panel switches, has the capability of simulating most functions normally initiated or controlled by the CPU. With the 'simulate CPU' (test) level active, the 'CPU operation control' lines to and from the channel are inhibited at the channel's CPU interface. Thus, communication between the CPU and channel is not possible with the channel in the test mode. Pushbuttons (START I/O, TEST I/O, HALT I/O, CLEAR INTERRUPT, LOAD DATA ADDRESS, STORE, and FETCH), the UNIT ADDRESS switches, and the AUTO RST (auto restart) switch on the CE panel (Figure 6-1), along with their associated test logic, provide the channel with the capability of initiating operations normally initiated by the CPU during automatic-mode operations. Operations which cannot be simulated with the channel in the test mode are the IPL, test channel, FLT, and diagnostic operations. Logout operations cannot be performed, either; however, if the LOG ON MACH CHK switch is on during test-mode operations, the channel stops operations upon detection of a channel machine check or channel data check condition.

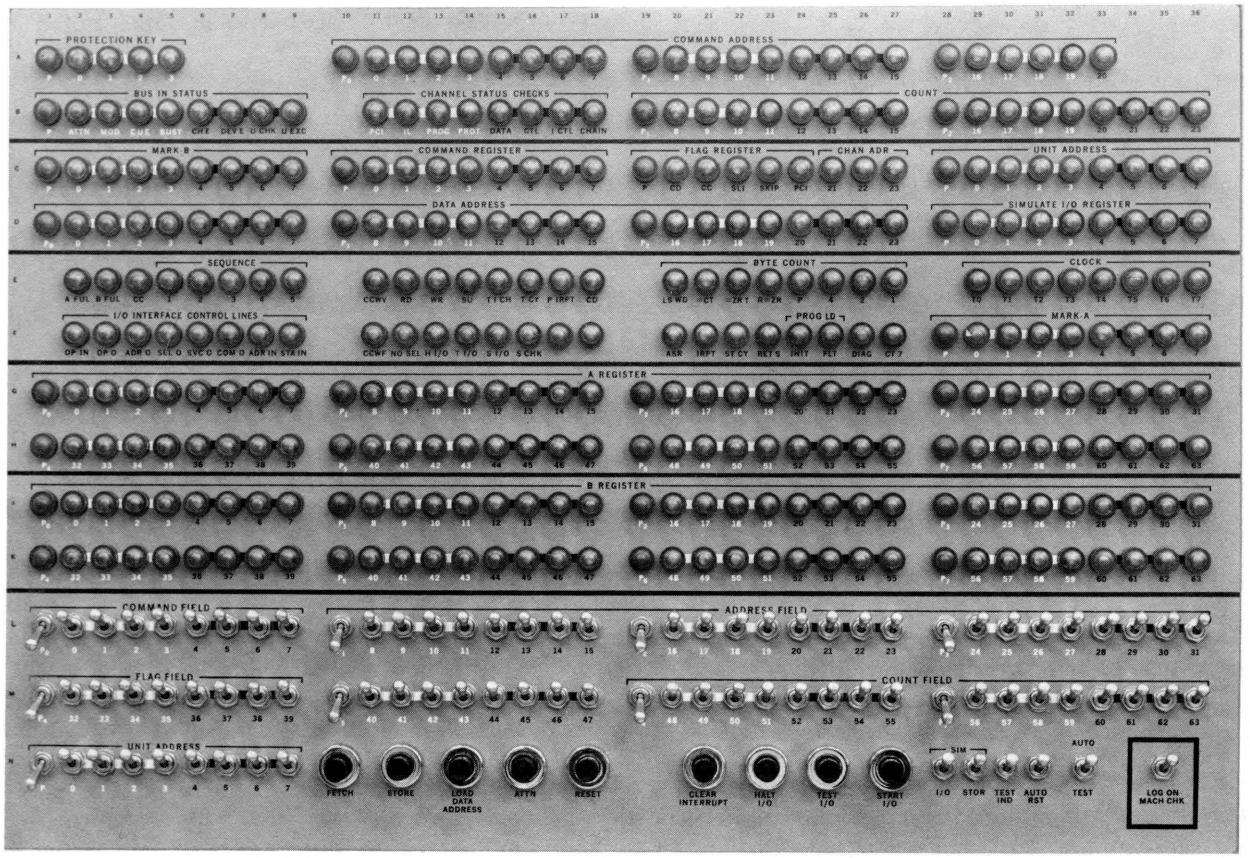


Figure 6-1. 2860 CE Panel

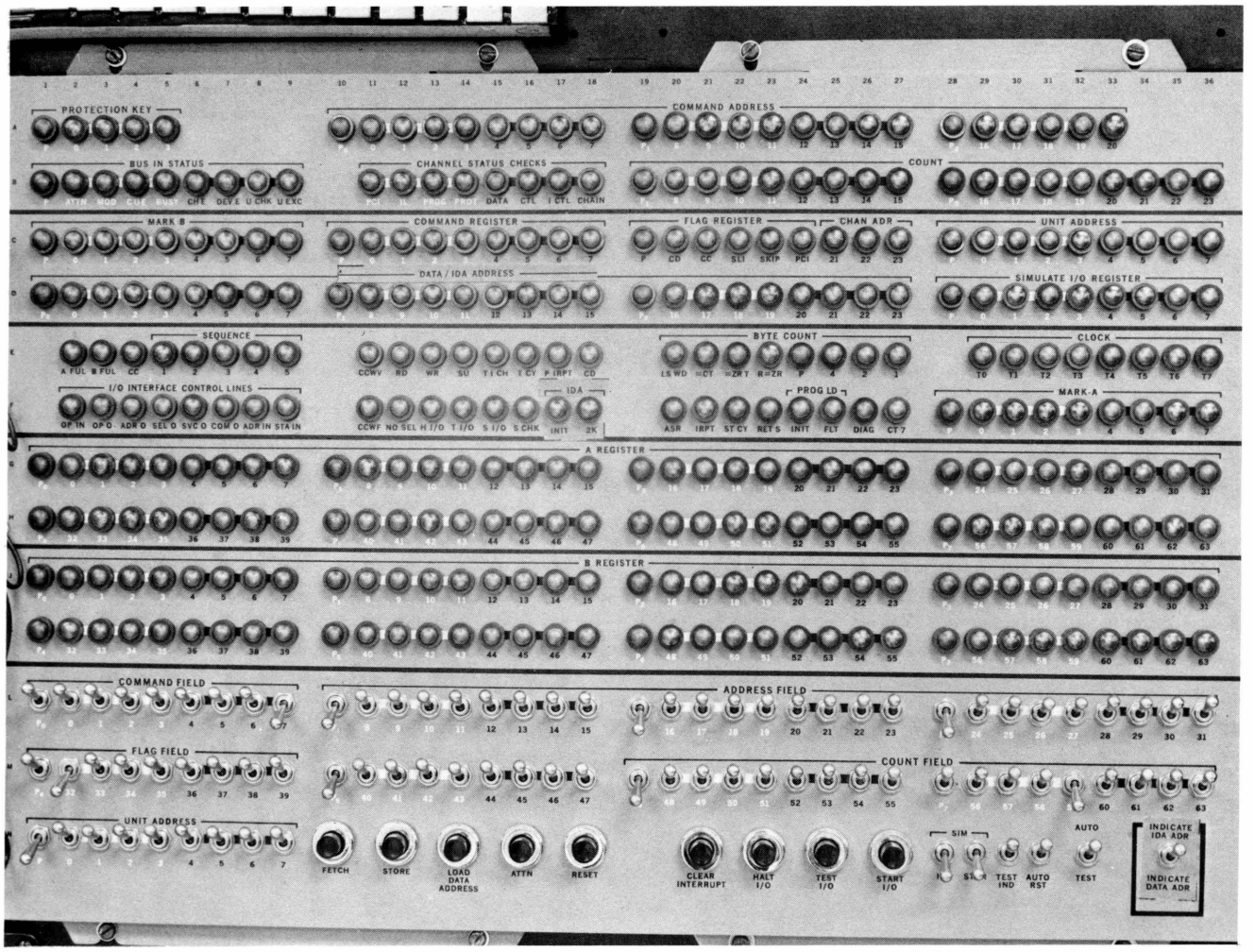


Figure 6-1A. 2860 CE Panel (with CIDA Feature)

When either condition stops channel operations, the channel remains stopped with a log-wait-interrupt condition pending until the condition is cleared by a reset operation. While channel operations are stopped, the CE panel indicators display the condition of channel registers and significant latches. A reset operation to clear the log-wait-interrupt condition occurs when a manual reset (RESET switch) operation is initiated at the CE panel.

Other CE panel switches may be activated to cause the channel to perform operations with or without accessing main storage ("BCU/CPU interface on channel operation control" in Figure 6-2). At the same time other switches

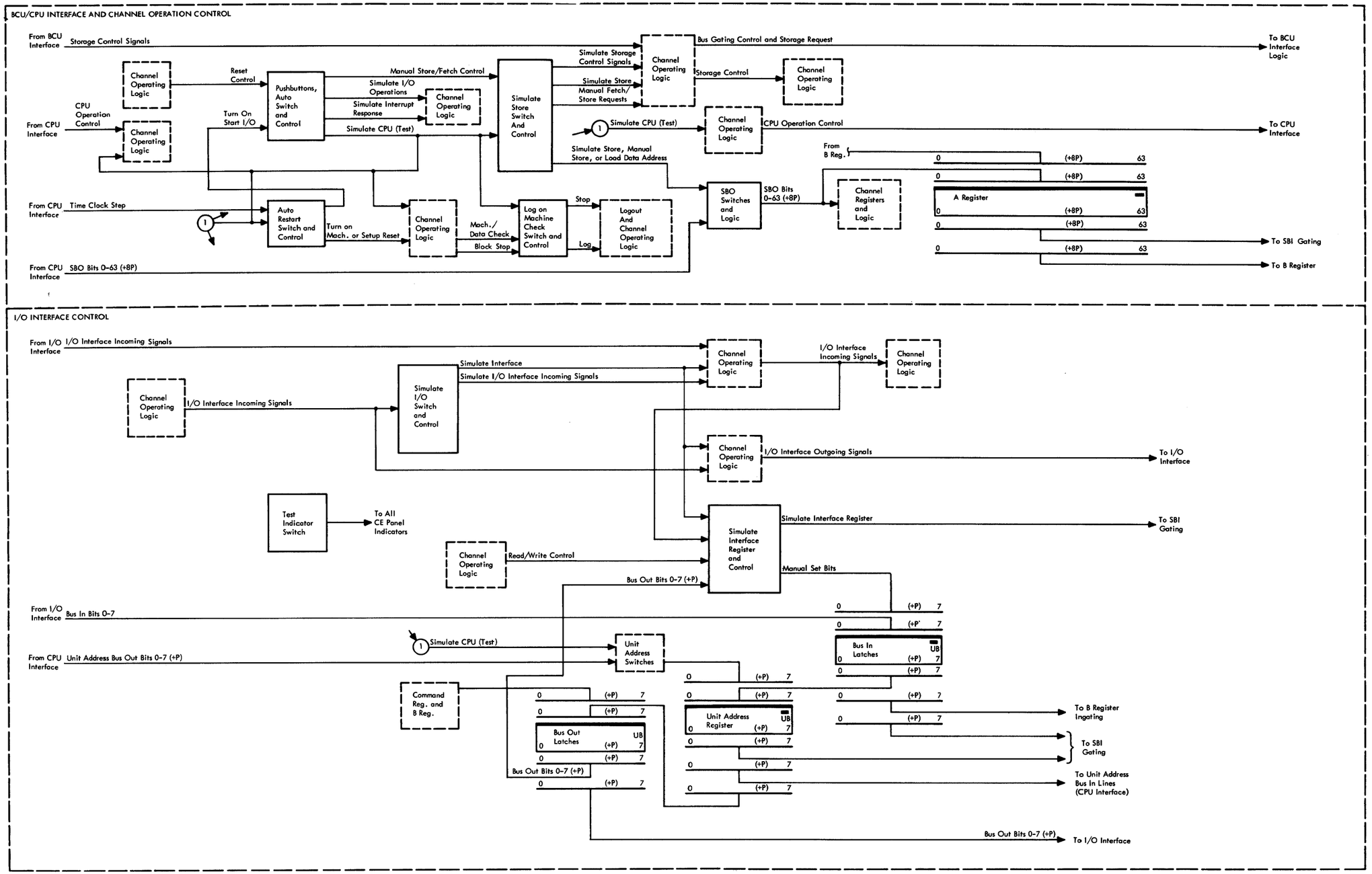


Figure 6-2. CE Switches and Control, Block Diagram

may be activated to cause the channel to perform operations with or without access to control units on the I/O interface ("I/O interface control" in Figure 6-2).

The position of the SIMUL STOR (simulate storage) switch (Figure 6-1) determines whether the channel has access to storage or must simulate storage operations while in the test mode. If the channel is to simulate storage operations (SIMUL STOR switch on), test logic associated with the switch generates 'simulate storage control' signals (Figure 6-2). These signals simulate the 'storage control' signals normally received from the BCU interface during operations when main storage is accessed, but inhibited at the channel by the 'simulate storage' signal. ('Bus gating control' lines and the 'storage request' signal to the BCU interface are also inhibited, preventing the channel from raising BCU interface signals.) In the channel, the 'simulate storage control' signals cause channel operating logic to operate as though 'storage control' signals were received via the BCU interface.

Note: The phrase "channel operating logic" used throughout this text refers to normal channel operation logic, rather than to test logic associated with CE panel switches.

During a simulate storage operation, fetch or store accesses to main storage may be simulated.

Simulate-storage store operations may be performed as a result of a start I/O instruction specifying a read operation or a manual store operation. The start I/O read instruction is initiated by setting the COMMAND FIELD SBO switches for a read command and either pressing the START I/O pushbutton or setting the AUTO RST (auto restart) switch to the on position (Figure 6-1). In the latter case, a start I/O operation is initiated every 16.7 ms by a 'time clock step' signal from the CPU interface (Figure 6-2). This signal, with the AUTO RST switch on, generates a 'turn on machine or setup reset' signal followed by a 'turn on start I/O' signal every 16.7 ms. A manual store operation is initiated by pressing the STORE pushbutton (Figure 6-1) on the CE panel. With the 'simulate storage' level active (Figure 6-2), channel operations differ for the start I/O read operation and manual store operation as follows. For a start I/O read operation, data for simulated storage is gated into the A-register from the B-register. For the manual store operation, data is gated into the A-register from SBO switches 0 through 63 (plus P0 through P7) on the CE panel (Figure 6-1). In either case, the channel operating logic associated with storage cycle store operations responds to the 'simulate storage control' signals generated by 'simulate store switch and control' test logic (Figure 6-2), and data transferred to the A-register is not actually stored.

Simulate-storage fetch operations may be performed as a result of a start I/O write operation or as the result of a manual fetch operation. Start I/O operations are initiated as described for simulate-storage fetch operations, except that a write command is set into the COMMAND FIELD SBO switches. A manual fetch operation is initiated by pressing the FETCH pushbutton on the CE panel (Figure 6-1). In either case, with the 'simulate storage' level active, data is "fetched" from the SBO switches (Figure 6-2) and gated into the A-register. Channel operating logic associated with 'storage cycle' fetch operations responds to the 'simulate storage control' signals generated by the 'simulate store switch and control' test logic.

If the SIMUL STOR switch is off, generation of the 'simulate storage control' signals is inhibited, and the inhibit level is removed from the BCU interface 'storage control' signals, 'bus gating control' signals, and 'storage request' signal (Figure 6-2). In this case, initiation of a start I/O read or manual store operation (as previously described) results in data being stored in main storage; initiation of a start I/O write or manual fetch operation causes data to be fetched from main storage.

The position of the SIMUL I/O switch (Figure 6-1) determines whether the channel has access to attached control units or must simulate I/O interface operations while in the test mode. If the channel is to simulate interface operations (SIMUL I/O switch on), test logic associated with the SIMUL I/O switch generates 'simulate I/O interface incoming' signals as required (Figure 6-2). For simulate interface operations, the 'simulate interface' signal is active and inhibits 'I/O interface incoming' signals at the channel. The 'simulate interface' signal also prevents all I/O interface outgoing' signals (except the 'operational out' signal and bus-out bits) from reaching the I/O interface lines; however, 'I/O interface outgoing' signals are internally generated by the channel at the appropriate times determined by the channel operation in progress. To simulate I/O interface transfer of byte information across the I/O interface, the channel contains 'simulate interface register and control' logic. This logic stores or routes byte information in accordance with the channel operation being performed at a given time.

For channel operations requiring simulated selection of an I/O device (start I/O or test I/O operations), the control portion of the 'simulate interface register and control' logic gates the unit address from the bus-out latches to the bus-in latches (via the 'manual set bits' lines (Figure 6-2) when a 'simulated address in' signal is generated). Prior to this gating operation, the unit address bits are gated into the unit address register from the CE panel UNIT ADDRESS switches. (These switches simulate the unit address bits normally received via the CPU interface 'unit address bus-out' lines.) During the simulated I/O device selection, the channel gates the unit address register contents to the bus-out latches as during normal operation. The outputs of the bus-out latches are then applied to the 'simulate interface register and control' logic and are subsequently gated to the bus-in latches to simulate return of the unit

address to the channel by a control unit. (Note: The bus-out bits are also active on the I/O interface bus-out lines; however, since 'I/O interface outgoing' signals are inhibited, this is of no consequence. The channel may then compare the unit addresses as during a normal I/O device selection routine; i.e., the channel determines that the correct I/O device has been selected.)

For write-type or read-type I/O interface data transfers, the simulate interface register portion (eight bits plus parity) of the 'simulate interface register and control' logic simulates data handling capabilities of a control unit. For write-type operations, data bytes are gated into the simulate interface register from the bus-out latches. For each byte gated to the bus-out latches (from the command register or B-register), the simulate interface register is reset and the new data byte entered into the register. At the end of the write-type operation, the simulate interface register retains the last byte transferred. Except as previously described, simulate I/O write-type operations are controlled by channel operating logic in the same manner as for a normal write-type operation.

For read operations, the byte content of the simulate interface register does not change and is gated to the bus-in latches (via the 'manual set bits' lines) each time a byte transfer is simulated. Since the byte value in the simulate interface register does not change, doublewords assembled in the B-register during the read operation consits of bytes, each of which is identical. Except as previously described, simulate I/O read-type operations are controlled by channel operating logic in the same manner as for a normal read-type operation.

The simulate interface register outputs (Figure 6-2) are sent to the SBI gating logic for transfer to main storage if a logout operation occurs. Since, in the test mode, logout operations are not performed, these outputs to the SBI gating logic are significant only during automatic-mode operation. When a logout operation is performed, bits 0-7 of the simulate interface register are stored in logword 2 and the parity bit in logword 3.

If the channel is to access an attached control unit, the SIMUL I/O switch is set to the off position, deactivating the 'simulate interface' level. This inhibits the generation of 'simulate I/O interface incoming' signals (Figure 6-2) and removes the inhibit level to the 'I/O interface incoming and outgoing' signals. In addition, the 'simulate interface register and control' logic is disabled. Thus, the channel has access to the attached control units while in the test mode. To access a control unit, the UNIT ADDRESS switches are set for the desired device address, and a test-mode start I/O or test I/O operation is initiated. Channel operating logic associated with I/O device selection and I/O byte transfers now operate as during automatic-mode operation, except that the unit address is obtained from the UNIT ADDRESS switches, rather than from the CPU interface 'unit address bus out' lines.

In summary, the CE panel contains switches and indicators. The indicators display the status of channel registers, latches, I/O interface lines, and channel status for both automatic-mode and test-mode operations. Automatic-mode operation allows the channel to operate on-line (under CPU control) while test-mode operation places the channel off-line. (CPU control must be simulated by using CE panel switches.) In test mode, most channel operations initiated by the CPU can be initiated at the CE panel. Further, CE panel switches may be set so that the channel operates by simulating storage, simulating the I/O interface, or simulating both storage and the I/O interface. If storage simulation is not selected, the channel has access to main storage for store and fetch operations. If I/O interface simulation is not selected, the channel may operate with attached control units.

Each CE panel switch and associated logic as well as each indicator is described separately in the following paragraphs.

CE PANEL SWITCHES AND TEST LOGIC

The CE panel switches and associated test logic are described separately in the following paragraphs, with appropriate references to diagrams in the 2860 Selector Channel FEMDM. Diagram 8-1 shows detailed data flow from the CE panel switches and associated logic to channel operating logic. Blocks representing CE panel switches and associated logic are referenced to positive logic diagrams which illustrate the functions of the block in detail.

Pushbuttons, AUTO/TEST Switch and Control Logic

All CE panel pushbuttons, AUTO/TEST switch, and associated control logic are shown on Diagram 8-2. Operating principles of each switch is described separately.

AUTO/TEST Switch

- With AUTO/TEST switch to AUTO, channel is available for system on-line (auto mode) operation.
- With switch to TEST, CPU interface in/out lines are inhibited; channel is off-line (test mode) and unavailable for CPU control.
- For auto-mode operation:
  1. All CE panel switches except LOG ON MACH CHK and TEST IND are inactive.
  2. Channel metering logic is enabled by 'auto mode switch' level.
  3. Channel test logic is disabled.

- Channel changes from auto-mode to test-mode operation (or vice versa) when metering logic is enabled or disabled by halt, wait, or check-stop condition of CPU.

- For test-mode operation:
  1. All CE panel switches are active.
  2. Channel metering logic is disabled by 'not auto mode switch' level.
  3. Channel test logic is enabled.
  4. I/O instructions (start I/O, test I/O, test channel, halt I/O) from CPU interface are inhibited at channel.
  5. Channel's 'operational in' and 'test light' levels to CPU interface drop.
  6. 'Unit address bus out' lines 'IPL' signal and 'master reset' signal from CPU interface are inhibited at channel.
  7. Channel logout operations and logout interrupt condition inhibited.
  8. 'Channel control check' resulting from 'storage protect register parity' check inhibited.
  9. 'Diagnostic' signals from CPU interface inhibited at channel.

The AUTO/TEST switch (Figure 6-1) determines the channel mode of operation (on-line or test). For on-line operation the switch is placed in AUTO; i.e., the channel is available for system operation. When in the AUTO position, the 'auto mode switch' level disables all CE panel switches except the LOG ON MACH CHK (log on machine check) and TEST IND (test indicator) switches (Diagram 8-2). In addition, the 'auto mode switch' level disables the generation of the 'manual pulse SS' signal, normally generated during test operations when a CE panel pushbutton is pressed. Channel metering logic is also enabled by the 'auto mode switch' level, allowing the channel meter to log customer usage time.

Note: When the AUTO/TEST switch is placed in TEST, the channel enters the test mode immediately; however, the metering state of the channel does not change until the CPU is in the halt, wait, or check-stop state. If the CPU does not enter any of the cited states, the AUTO/TEST switch may be placed in TEST, test-mode operations performed, and the channel returned to auto-mode operation without a change in the channel metering state. Normally, the channel ENABLE/DISABLE switch is set to DISABLE when the necessity for test-mode operation is determined. With this switch set to DISABLE, the CPU must still enter the wait, halt, or check-stop state in order for the channel metering state to change; this normally occurs before channel test-mode operations are required.

When the AUTO/TEST switch is placed in TEST (Diagram 8-2), the 'not auto mode switch' level: (1) enables all CE panel switches (the LOG ON MACH CHK and TEST IND switches are not affected by the 'not auto mode switch' level and are active for both auto-mode and test-mode operations); and (2) is OR'ed to produce the 'simulate CPU' and 'test' levels. These levels remove the channel from an on-line condition (auto mode) and place it in the test mode. When the metering logic changes state, a 'block program control' level is generated. This level is also OR'ed to maintain the 'simulate CPU' and 'test' levels when the AUTO/TEST switch is returned to AUTO. When the 'block program control' level is active, the channel metering state must again change to enable the channel to begin auto-mode operations.

Entry into the test mode occurs when the 'test' and 'simulate CPU' signals electrically disconnect the channel from CPU interface control as follows:

1. The channel 'channel available' signal to the CPU interface is dropped, making the channel unavailable for system use. For the System/360, Model 75, a 'test light' signal is activated to turn on a test light on the console.
2. 'Condition code' lines and the 'release' signal to the CPU interface are inhibited.
3. Start I/O, test I/O, halt I/O, and test channel multiplex signals from the CPU interface are inhibited at the channel. (CE panel pushbuttons serve to simulate all but the Test Channel instruction.)
4. 'Unit address bus out' lines from the CPU interface are inhibited at the channel. (CE panel UNIT ADDRESS switches are enabled to simulate the 'unit address bus out' bits.)
5. Channel logout operations (storage of three log words) and recognition of the log wait interrupt condition are inhibited.
6. 'IPL' and 'system reset' signals from the CPU interface are inhibited at the channel, preventing these multiplex signals from initiating channel 'interface reset' and 'machine reset' signals while the channel is in the test mode.
7. Turn-on of the 'channel control check' latch due to detection of a storage-protect-register-parity-check condition is inhibited. (For simulate start I/O operations, a CAW is not fetched from main storage, but zeros are forced into the storage protect register. Thus, 'storage protect register parity' checks are ignored in the channel. If access to main storage is required, main storage may detect a storage protect check condition if bad parity exists.)
8. Diagnostic multiplex signals from the CPU interface are inhibited at the channel. This prevents CPU diagnostic control of the channel while the channel is in the test mode.

The 'simulate CPU' and 'test' signals, in addition to performing the previously described functions, also enable test logic associated with the CE panel LOG ON MACH CHK, AUTO RST (auto restart), SIMUL STOR, and UNIT ADDRESS switches.

In summary, the position of the AUTO/TEST switch determines whether the channel is available for system

operation (auto mode) or test-mode operation. In the AUTO position, all CE panel switches and associated test logic except the LOG ON MACH CHK and TEST IND switches are disabled, and the channel is capable of accepting and responding to instructions and control signals received via the CPU interface. In the TEST position, CPU interface instruction and control signals are inhibited at the channel, and all CE panel switches are activated. Activation of these switches permits the operator to simulate most instructions and control signals normally provided via the CPU interface.

START I/O Pushbutton

- Active when channel in test mode.
- When switch is pressed, 'start I/O switch' signal initiates simulated start I/O operation by:
  1. Causing generation of 170-ns 'manual singleshot' signal.
  2. AND'ing with 'manual singleshot' signal to turn on 'simulate start I/O' trigger.
- 'Simulate start I/O' signal turns on channels 'start I/O' latch.
- Channel operations during simulated start I/O operations depend upon:
  1. Position of SIMUL STOR switch.
  2. Position of SIMUL I/O switch.
- With simulate storage off, channel fetches CAW and CCW from main storage and controls operation per CCW as during normal start I/O operation.
- With simulate storage on, channel does not fetch CAW, but fetches CCW from SBO switches 0-63 (plus P0-P7); channel operations are controlled by CCW values preset into SBO switches.
- Values set into UNIT ADDRESS switches are gated to unit address register.
- With simulate I/O interface off, channel selects I/O device and, if specified by command register bits, controls data transfers as during normal start I/O operation.
- With simulate I/O interface on, channel simulates selection of I/O device and, if specified by command register bits, transfers data via simulate interface register.

The CE panel START I/O pushbutton (Figure 6-1) provides one means of simulating initiation of a start I/O operation at the channel. The doublepole, doublethrow pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in TEST. With the switch in AUTO, the 'auto mode switch' level disables the AND gate to the 'manual pulse' latch. This prevents generation of the 170-ns 'manual pulse SS' signal, which is required before the 'start I/O switch' signal can turn on the 'simulate start I/O' latch.

With the AUTO/TEST switch set to TEST, one input to the manual pulse AND gate (Diagram 8-2) is enabled. When the START I/O pushbutton is pressed, the 'start I/O switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND gate and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'start I/O switch' signal to set the 'simulate start I/O' latch.

The resulting 'simulate start I/O' signal turns on the channel 'start I/O' latch (Diagram 8-1) to initiate a start I/O operation. Channel operations during the start I/O operation depend in part upon the position of the SIMUL STOR and SIMUL I/O switches. If the SIMUL STOR switch is off, the channel accesses main storage to fetch first the channel address word (CAW) and then the channel control word (CCW) as during a normal start I/O operation. (See "Initial Selection Routine.") Once the valid CCW is in the channel registers, further channel operations depend upon the command and other control information received in the CCW.

If the SIMUL STOR switch is on, the channel does not have access to storage during the simulated start I/O operation. In this case, the channel does not fetch a CAW. Instead, it fetches a CCW from the value set into the CE panel SBO switches. (See "Simulate Storage Switch and Control Logic.") When the valid CCW is in the channel, further channel operations depend upon the CCW values set into the SBO switches. If a write command is specified, the SBO switches must be set to satisfy both the requirements of a valid CCW and a data doubleword. This is necessary since both the CCW and the data doublewords for the write operation are fetched from the SBO switches.

If the SIMUL I/O switch is off, the channel selects an attached I/O device during the simulated start I/O operation. During the start I/O initial selection routine, the address of the I/O device to be selected is gated from the CE panel UNIT ADDRESS switches into the channel's unit address register. Otherwise, channel operations performed to select the I/O device and, if necessary, to control the transfer of data across the I/O interface, are as performed during a normal start I/O operation. (See "Initial Selection Routine.")

If the SIMUL I/O switch is on, the channel gates the values in the UNIT ADDRESS switches to the unit address register as previously described. However, the I/O interface is disabled and channel test and operating logic are

used to simulate selection of the I/O device and, if necessary, the transfer of data across the I/O interface. To simulate I/O device selection (Diagram 8-1), the channel gates the values in the UNIT ADDRESS switches to the unit address register, internally generates the 'address out' signal, and gates the unit address register contents of the bus-out latches. Subsequently, the channel simulates generation of the 'address in' signal and gates the unit address bits on the bus-out lines to the bus-in latches. After the unit address comparison by the channel, the command byte is gated to the simulate interface register, and the channel internally generates the 'command out' signal. The channel then simulates the 'status in' signal and examines the bus-in byte for all zeros. If read or write operations are specified by the command, the channel simulates I/O interface data transfers by transferring bytes from the B-register to the simulate interface register during write operations, or by transferring bytes (all of the same values) from the simulate interface register to the B-register during read operations. During the read or write operations, the channel simulates the required 'service in' signals from the I/O interface and internally generates the 'service out' signals. (See "Simulate I/O Switch and Control Logic.")

As during a normal start I/O operation, the channel examines the CC and CDA flags at the appropriate time to determine if chaining operations are required. If so, the channel continues operations by fetching a CCW either from main storage or from the CE panel SBO switches, depending upon the position of the SIMUL STOR switch. If chaining operations are specified while the channel is simulating storage, each CCW fetched during chaining has the same bit values since the CCW is "fetched" from the CE panel SBO switches.

Since CPU interface communication is inhibited while the channel is in the test mode, the channel does not store the channel status word (CSW) upon completion of the simulated start I/O operation. The simulated start I/O operation ends when the channel turns on the 'interrupt' latch. The channel retains the interrupt condition until a new test operation is initiated. (This operation may be a start I/O, test I/O, clear interrupt, or reset operation initiated at the CE panel.)

HALT I/O Pushbutton

- Pressing HALT I/O pushbutton with AUTO/TEST switch to TEST simulates halt I/O operation in channel.
- 'Manual pulse SS' signal AND's with 'halt I/O switch' signal to turn on 'simulate halt I/O' latch.
- 'Simulate halt I/O' signal activates 'halt I/O' signal.
- Channel stops any current operation in progress, then remains in wait state.
- If simulate I/O operation is in progress, channel simulates disconnect operation:
  1. 'Status in' signal is simulated.
  2. Stop command byte is inhibited.
  3. Turn-on of 'incorrect length' latch is inhibited.
  4. Polling interrupt from "halted I/O" device is not simulated.
- If channel is operating with I/O device:
  1. Channel performs normal disconnect operation.
  2. Polling interrupt is initiated by halted I/O device after channel interrupt is cleared.

The HALT I/O pushbutton (Figure 6-1) provides a means of simulating initiation of a halt I/O operation at the channel. The doublepole, doublethrow pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in TEST. With the switch in AUTO, the 'auto mode switch' level prevents generation of the 170-ns 'manual pulse SS' signal, which is required before the 'halt I/O switch' signal can turn on the 'simulate halt I/O' latch.

With the AUTO/TEST switch set to TEST, one input to the manual pulse AND gate (Diagram 8-2) is enabled. When the HALT I/O pushbutton is pressed, the 'halt I/O switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND gate and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'halt I/O switch' signal to set the 'simulate halt I/O' latch.

The resulting 'simulate halt I/O' signal is OR'ed (Diagram 8-1) to activate the 'halt I/O' signal. With the 'halt I/O' signal active, the channel stops any current operation in progress and remains in a wait condition with the 'interrupt' latch on. To perform the Halt I/O instruction, the channel operates as if a Halt I/O instruction had been received from the CPU interface, with the following exceptions.

If the SIMUL I/O switch is on, and the channel is engaged in simulating operations with an I/O device, a simulated I/O interface disconnect operation is performed. This disconnect operation is similar to an actual I/O device disconnect operation, except that the 'status in' signal from the I/O device is simulated by the channel, and the stop command to the bus-out latches is inhibited. In addition, the 'incorrect length' signal, generated when an actual I/O device is disconnected, is inhibited during the simulated I/O interface disconnect operation. The polling interrupt normally attempted by an actual halted I/O device (after the channel interrupt condition is cleared) is not attempted during the simulated I/O interface disconnect operation.

If the SIMUL I/O switch is off and the channel is operating with an I/O device, the channel disconnects the

I/O device. After the channel interrupt is cleared and polling operations are resumed, the halted I/O device attempts a polling interrupt, as during normal operations.

TEST I/O Pushbutton

- Pressing TEST I/O pushbutton with AUTO/TEST switch to TEST simulates test I/O operation in channel.
- 'Manual pulse SS' signal AND's with test I/O switch signal to turn on 'simulate halt I/O' latch.
- 'Simulate test I/O' signal activates 'test I/O' signal.
- If SIMUL I/O switch is off, channel selects actual I/O device per address in UNIT ADDRESS switches; when I/O device status is received, 'interrupt' latch is turned on and channel remains in wait condition, if the status byte is not all zeros.
- If SIMUL I/O switch is on, channel simulates device selection as follows:
  1. Gates UNIT ADDRESS switch values to unit address register.
  2. Internally generates 'address out' signal.
  3. Gates unit address register bits to bus-out lines.
  4. Simulates 'address in' signal; gates unit address on bus-out lines to bus-in lines for address comparison.
  5. Internally generates 'service out' signal; simulated 'address in' signal drops.
  6. Simulates 'status in' signal and I/O device status byte of all zeros on bus-in lines.

The TEST I/O pushbutton (Figure 6-1) enables simulating initiation of a test I/O operation at the channel. The doublepole, doublethrow pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in TEST. With the switch in AUTO, the 'auto mode switch' level prevents generation of the 170-ns manual pulse SS' signal, which is required before the 'test I/O switch' signal can turn on the 'simulate halt I/O' latch.

With the AUTO/TEST switch set to TEST, one input to the 'manual pulse' AND gate (Diagram 8-2) is enabled. When the TEST I/O pushbutton is pressed, the 'test I/O switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND gate and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'test I/O switch' signal to set the 'simulate test I/O' latch.

The resulting 'simulate test I/O' signal is OR'ed (Diagram 8-1) to activate the 'test I/O' signal. With the 'test I/O' signal active, the channel either issues a Test I/O command to an actual I/O unit or internally simulates the test I/O sequence.

If the SIMUL I/O switch is off, the simulated 'test I/O' signal causes the channel to perform a test I/O sequence with an attached I/O device. To test the I/O device, the channel gates the address values from the UNIT ADDRESS switches into the unit address register. Test I/O operations are then performed as for a normal test I/O operation to the point where the channel has an interrupt condition pending; i.e., the channel has selected the specified I/O device and an I/O unit status byte is on the bus-in latches. Since the channel is inhibited from raising the 'interrupt request' signal at the CPU interface, the channel remains in the wait state with the interrupt pending.

If the SIMUL I/O switch is on, the simulated 'test I/O' signal causes the channel to simulate selection of an I/O device. During the simulated I/O device selection routine, the channel gates the values from the UNIT ADDRESS switches into the unit address register. Next, the channel simulates selection of an I/O device by raising the 'address out' signal and gating the unit address register contents to the bus-out lines. Subsequently, the channel simulates generation of the 'address in' signal, causing the unit address on the bus-out lines to be gated to the bus-in lines (via the 'simulate interface register and control' logic). See Diagram 8-1. After the channel determines that the 'unit address' bits on the bus-in lines and the unit address register bits match, it internally generates a 'service in' signal, causing the simulated 'address in' signal to drop. The channel then simulates the 'status in' signal which causes the 'simulate interface register and control' logic to simulate receipt of an I/O device status byte of all zeros (I/O device available and ready for operation). At this point, the channel returns to the idle state and awaits initiation of another I/O operation.

CLEAR INTERRUPT Pushbutton

- CLEAR INTERRUPT pushbutton, when pressed, clears interrupt-in-channel condition and, for not-simulate I/O operation, clears polling-interrupt condition.
- When pushbutton is pressed with AUTO/TEST switch on, the channel:
  1. Generates 'manual pulse SS' signal to turn on 'simulate interrupt response' latch.
  2. Latch output ('simulate accept interrupt') raises 'CPU accept interrupt' signal in channel.
  3. Pending polling-interrupt condition cleared if SIMUL I/O switch off.
     a. Unit address register bits not gated to 'unit address bus out' lines (CPU interface).
     b. Channel forces test I/O routine (activates 'pseudo accept interrupt' signal), selects I/O device with

pending status, receives status byte from control unit, performs a simulated CSW store operation, and enters idle state awaiting receipt of next instruction.

4. Pending interrupt-in-channel condition is cleared as follows:
   a. CSW is in channel.
   b. CSW stored in main storage address location 64 if SIMUL STOR switch is off.
   c. Storage of CSW simulated if SIMUL STOR switch is on. (If pending interrupt condition is result of pressing ATTN (attention) pushbutton, channel operations will hang; reference EC705864).

The CLEAR INTERRUPT pushbutton (Figure 6-1) serves to manually simulate receipt by the channel of an 'interrupt response' signal from the CPU interface. Pressing the CLEAR INTERRUPT switch causes the channel to complete an outstanding polling-interrupt sequence (if the SIMUL I/O switch is off) or to clear an interrupt-in-channel condition. When the interrupt condition is cleared, the channel resumes polling operations (assuming the SIMUL I/O switch is off). For test-mode purposes involving the CLEAR INTERRUPT pushbutton, a polling-interrupt condition is defined as an interrupt condition requiring access to an attached control unit; an interrupt-in-channel condition exists when the channel 'interrupt' latch is on.

The doublepole, doublethrow CLEAR INTERRUPT pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in TEST. When the switch is in AUTO, the 'auto mode switch' level prevents generation of the 170-ns 'manual pulse SS' signal, which is required before the 'interrupt response switch' signal (generated when the CLEAR INTERRUPT pushbutton is pressed) can turn on the 'simulate interrupt response' latch.

With the AUTO/TEST switch set to TEST, one input to the manual pulse AND (Diagram 8-2) is enabled. When the CLEAR INTERRUPT pushbutton is pressed, the 'interrupt response switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'simulate interrupt switch' signal to set the 'simulate interrupt response' latch.

The resulting 'simulate accept interrupt' signal is OR'ed (Diagram 8-1), then AND'ed with the 'not interface reset' and 'not log wait interrupt' signals to raise the 'CPU accept interrupt' signal in the channel. With this signal active, the channel either performs a polling-interrupt sequence (providing a polling-interrupt condition is pending and the SIMUL I/O switch is off) or clears an interrupt-in-channel condition (providing the 'interrupt' latch is on).

The channel performs the polling-interrupt sequence by forcing a test I/O operation. During the forced test I/O operation ('test I/O' latch on), the channel selects the I/O device specified by the address in the channel unit address register. (This address was originally obtained when when the I/O device requested a polling interrupt). When the device is selected, the control unit raises its 'status in' signal and places the status byte on the bus-in lines. The channel responds by raising the 'service out' signal and performing a CSW store operation. During the CSW store operation, the control unit drops the 'status in' signal, the channel drops the 'select out' signal, and the control unit drops the 'operational in' signal. Thus, the channel clears the polling interrupt and is idle awaiting the next I/O instruction.

If an interrupt-in-channel condition exists, pressing the CLEAR INTERRUPT pushbutton (to activate the 'CPU accept interrupt' signal) causes the channel to clear the interrupt in the normal manner (stores the CSW in main storage at address 64 decimal) with the following exceptions. The unit address register contents are not sent to the CPU interface, since the channel 'simulate CPU' signal inhibits the 'unit address bus in' lines. When the SIMUL STOR switch is on, the CSW is not stored, but is lost when the A-register is reset. Since the status bits comprising the CSW are already in the channel, the position of the SIMUL I/O switch is unimportant.

STORE Pushbutton

- When pressed, STORE pushbutton causes manual-store operation.
- 'Manual store' latch is turned on and turns on 'storage request' latch.
- CE panel SBO switch values are gated to channel SBO lines by 'simulate storage or manual store or load data address' signal.
- SBO bits values are gated to A-register by AND'ing 'manual store' and 'storage request'.
- 'Manual store' signal:
  1. Enables all mark bits for gating to mark lines by 'BCU response' signal.
  2. Enables 'storage protect parity' bit for gating to 'storage protection' lines by 'BCU response' signal.
  3. Enables gating of A-register bits to SBI by 'BCU data request' signal.
  4. Inhibits outputs from mark-A register.
  5. Enables data address register bits to be gated to SAB by 'BCU response' signal.
  6. Enables gating of 'store' signal to BCU interface by 'BCU response' signal.

- For not-simulate-storage operation, data in A-register is stored and BCU interface lines enabled by 'manual store' signal are gated to BCU interface at appropriate times.

- For simulate storage operation, data in A-register is not stored, enabled BCU interface lines are not gated; i.e., channel simulates storage cycle.

- 'Raw advance SS' signal (generated by 'BCU advance pulse' signal during 'not simulate storage' or by 'simulate advance pulse' signal during simulate storage) turns on 'sequence 4' latch.

- During sequence 4, data address register value is updated by equivalent of eight bytes.

The STORE pushbutton (Figure 6-1) enables the channel to store one data doubleword directly from the CE panel SBO switches. The channel first gates the values in CE panel SBO switches to the A-register, then gates the data address register contents (loaded previously from the CE panel SBO switches by pressing the LOAD DATA ADDRESS pushbutton) to the SAB. If the SIMUL STOR switch is off, pressing the STORE pushbutton causes data to be stored in main storage. If the SIMUL STOR switch is on, pressing the STORE pushbutton causes a simulated store operation, and data gated from the SBO switches to the A-register is not actually stored.

The doublepole, doublethrow STORE pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in TEST. When the switch is in AUTO, the 'auto mode switch' level prevents generation of the 170-ns 'manual pulse SS' signal, which is required before the 'manual store switch signal (generated when the STORE pushbutton is pressed) can turn on the 'manual store' latch.

With the AUTO/TEST switch set to TEST, one input to the manual pulse AND gate (Diagram 8-2) is enabled by the 'not auto mode switch' level. When the STORE pushbutton is pressed, the 'manual store switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'manual store switch' signal to set the 'manual store' latch.

The resulting 'manual store' signal is OR'ed and then AND'ed with the 'not wrong length' record signal to raise two 'simulate storage or manual store or load data address' signals (Diagram 8-5). One signal gates the CE panel SBO switch values to the channel SBO lines, while the second signal prevents gating to the channel SBO lines from other sources (BCU interface SBO lines and mark-B register).

The 'manual store' signal is also AND'ed with the 'manual pulse SS' signal to generate a 'manual storage request' signal. In turn, the 'manual storage request' signal turns on the channel 'storage request' latch. The resulting 'storage request' signal is AND'ed with the 'manual store' signal, raising the 'gate SBO to A-register' signal. This signal, with the 'simulate CPU' signal active, gates the values in the CE panel SBO switches to the A-register.

To enable address information to be gated to the BCU interface (if the SIMUL STOR switch is off), the 'manual store switch' signal (Diagram 8-5): (1) enables all mark bits for gating to the 'mark bits' bus (by the 'BCU response' signal); (2) enables the 'storage protect parity' bit for gating to the 'storage protection' lines by the 'BCU response' signal; (3) enables the 'store' signal to be gated to the BCU interface by the 'BCU response' signal, and (4) is OR'ed to raise the 'manual fetch or store' signal. The 'manual fetch or store' signal enables the data address register contents to be gated to the SAB by the 'BCU response' signal. In addition, the 'manual fetch or store' signal inhibits the outputs of the mark-A register to prevent interference with the mark bits enabled by the 'manual store' signal at the 'mark bits' gating lines.

To enable gating of the A-register contents to the SBI lines when the 'BCU data request' signal arrives, the 'manual store' signal enables the SBI gates receiving the A-register outputs.

With the 'storage request' latch on and the BCU interface gating enabled as previously described, the channel performs a normal store operation if the SIMUL STORE switch is off, or simulates a store operation if the SIMUL STOR switch is on. If a store operation is simulated, the data address register bits, mark bits, 'storage protect parity' bit, and A-register bits are not gated to their respective BCU interface lines. In either case, the 'raw advance SS' signal is generated either as a result of receiving a 'BCU advance pulse' signal during not-simulate-storage operations, or as the result of the channel generating a 'simulate advance pulse' signal (Diagram 8-5) during simulate storage operations. The 'raw advance SS' signal is AND'ed with the 'manual fetch or store' signal to raise the 'turn on sequence 4' signal. This signal turns on the channel 'sequence 4' latch, causing the channel to update the data address register contents by the equivalent of eight bytes. The updated data address prepares the channel to store data at the next sequential address if another manual store operation is to be performed.

FETCH Pushbutton

- Pressing FETCH pushbutton causes channel to fetch data doubleword, either from main storage or CE panel SBO switches.

- Data fetched from main storage if SIMUL STOR switch is off; from CE panel switches if SIMUL STOR switch is on.

- Pressing FETCH pushbutton (AUTO/TEST switch to TEST) generates 'manual pulse SS', 'any switch on' and 'manual fetch switch' signals to set 'manual fetch' latch.

- 'Manual fetch' signal raises 'manual fetch or store' signal to turn on 'storage request' latch.

- 'Manual fetch or store' signal enables gating of data address register contents to SAB; inhibits mark-A register outputs.

- For not-simulate I/O, manual fetch operation:
  1. 'Storage request' signal raised at BCU interface.
  2. Data address register contents gated to SAB.
  3. Data doubleword received from main storage.
  4. Channel receives 'BCU advance pulse' signal; generates 'raw advance SS' signal.
  5. 'Raw advance SS' signal AND's with 'manual fetch' signal, causing fetched data to be gated to A-register.
  6. 'Raw advance SS' signal AND's with 'manual fetch or store' signal, causing channel to perform data address register update (sequence 4) operation.

- For simulate storage, manual fetch operation:
  1. 'Storage request' signal and data address register contents not gated to BCU interface.
  2. 'Simulate storage switch' level raises 'simulate storage or manual store or load data address' signals; SBO switch values gated to channel SBO lines.
  3. Channel internally simulates storage cycle signals normally received from BCU interface.
  4. 'Simulate advance pulse' signal causes channel to generate 'raw advance SS' signal.
  5. 'Raw advance SS' signal causes channel to gate SBO switch bits to A-register and perform data address update (sequence 4) operation.

The FETCH pushbutton (Figure 6-1) enables the channel to fetch one data doubleword either from the CE panel SBO switches or main storage. The source from which data is fetched depends upon the position of the SIMUL STOR switch. If the SIMUL STOR switch is off, the channel fetches a data doubleword from main storage and enters it into the A-register. If the SIMUL STOR switch is on, the channel fetches a data doubleword from the CE panel SBO switches.

The doublepole, doublethrow FETCH pushbutton (Diagram 8-2) is active only when the AUTO/TEST switch is in AUTO. When the pushbutton is in AUTO, the 'auto mode switch' level prevents generation of the 170-ns 'manual pulse SS' signal, which is required before the 'manual fetch switch' signal (generated when the FETCH pushbutton is pressed) can turn on the 'manual fetch' latch.

With the AUTO/TEST switch set to TEST, one input to the manual pulse AND (Diagram 8-2) is enabled. When the FETCH pushbutton is pressed, the 'manual fetch switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'manual fetch switch' signal to set the 'manual fetch' latch.

The resulting 'manual fetch' signal is OR'ed to raise the 'manual fetch or store' signal (Diagram 8-5). This signal is AND'ed with the 'manual pulse SS' signal to turn on the channel 'storage request' latch. In addition, the 'manual fetch or store' signal enables gating of the data address register contents to the SAB by the 'BCU response' signal and inhibits the output from the mark-A register. (Mark bits are not required for the fetch operation.)

If the SIMUL STOR switch is off, the 'storage request' signal is raised at the BCU interface, and data is fetched from the main storage address contained in the data address register. When the 'BCU advance pulse' signal is received, the channel generates a 'raw advance SS' signal which is active during the time the fetched data is on the SBO lines. This signal is AND'ed with the 'manual fetch' signal (Diagram 8-5) to raise the 'gate SBO to A-register' signal. This signal turns on the channel 'gate SBO to A-register' latch, causing the data to be gated from the SBO lines into the A-register. The 'raw advance SS' signal is also AND'ed with the 'manual fetch or store' signal, causing the 'turn on sequence 4' signal to rise. This signal turns on the channel 'sequence 4' latch, causing the channel to update the data address register contents by the equivalent of eight bytes. The data address update operation is performed if another 'manual fetch' operation is to be performed to a sequential address in main storage.

If the SIMUL STOR switch is on, the channel simulates a storage cycle. The data address register contents are not gated to the BCU interface, and data is fetched from the CE panel SBO switches. For the simulate storage manual fetch operation, the active 'simulate storage switch' level is OR'ed to raise the two 'simulate storage or manual store or load data address' signals (Diagram 8-5). One signal gates the values in the SBO switches to the channel SBO lines, while the second signal inhibits inputs to these lines from the BCU interface SBO lines and mark B-register.

The channel then internally simulates the signals required to perform the fetch operation. (See "Simulate Storage Switch and Control Logic.") When the channel generates the 'simulate advance pulse' signal, the signal causes generation of the 'raw advance SS' signal. This signal causes the SBO switch values on the SBO lines to be

gated into the A-register and initiates a sequence-4 operation in the manner described for a not-simulate storage, manual fetch operation.

LOAD DATA ADDRESS Pushbutton

- LOAD DATA ADDRESS pushbutton used to load values in CE panel ADDRESS FIELD (SBO) switches into data address register.
- Pushbutton active in test mode only.
- In auto mode, generation of 'manual pulse SS' signal is inhibited, disabling LOAD DATA ADDRESS switch.
- In manual mode, pressing LOAD DATA ADDRESS pushbutton generates 'manual pulse SS' signal to turn on 'load data address' latch.
- 'Load data address latch' signal:
  1. Raises 'simulate storage or manual store or load data address' signals to gate value in SBO switches to channel SBO lines.
  2. AND's with 'manual pulse SS' signal to raise 'load data address pulse' signal; this signal causes data address bits on channel SBO lines to be gated into data address register.

The LOAD DATA ADDRESS pushbutton (Figure 6-1) serves to load the values in the CE panel SBO switches into the data address register. The pushbutton is active only when the channel is in the test mode (AUTO/TEST switch set to TEST). Before the LOAD DATA ADDRESS pushbutton is pressed, the desired address is set into the ADDRESS FIELD (SBO) switches on the CE panel.

When the AUTO/TEST switch is in TEST, the 'auto mode switch' level enables one leg of the manual pulse AND. When the LOAD DATA ADDRESS pushbutton is pressed, the 'load data address switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'load data address' signal to set the 'load data address' latch.

The 'load data address latch' signal performs two functions: (1) gates the values in the SBO switches to the channel SBO lines; and (2) causes the channel to gate the data address bit values on the SBO lines (SBO bits 8-31, plus P1, P2, and P3) into the data address register.

To gate the SBO switches, the 'load data address latch' signal is OR'ed (Diagram 8-5) to raise the two 'simulate storage or manual store or load data address' signals. One signal gates the values in the SBO switches to the channel SBO lines; the second signal inhibits inputs to the channel SBO lines from the BCU interface and the mark-B register.

To gate the address bits from the channel SBO lines into the data address register, the 'load data address latch' signal is AND'ed with the 'manual pulse SS' signal (Diagram 8-2) to raise the 'load data address pulse' signal. This signal turns on the 'gate SBO to data address register' latch, causing the SBO bits to be gated into the data address register. With the data address bits in the data address register, the channel is prepared to perform a 'not simulate storage' manual fetch or manual store operation, if initiated at the CE panel.

RESET Pushbutton

- RESET pushbutton used in test-mode operation to reset channel registers and latches (except simulate interface register and 'byte count register equals zero' trigger.)
- If SIMUL I/O switch is off, RESET pushbutton (when pressed) also causes I/O interface reset.
- With AUTO/TEST switch to TEST, pressing RESET pushbutton causes generation of 'manual pulse SS' signal.
- 'Manual pulse SS' signal is AND'ed with 'reset channel switch' signal (from RESET pushbutton).
- Resulting signal generates 'reset suppress out latch', 'interface reset' and 'machine reset' signals.
- 'Reset suppress out latch' signal resets 'suppress out' latch, if on; reset condition of latch is required for channel to generate 'interface reset' and 'machine reset' signals.
- 'Machine reset' signal resets channel registers and latches.
- If SIMUL I/O switch is off, 'interface reset' signal causes channel to drop 'operational out' signal and attached I/O devices are reset; if switch is on, I/O devices are not reset.

The RESET pushbutton (Figure 6-1) is used during test-mode operation to reset all channel registers and triggers (and latches) except the simulate interface (I/O) register and the 'byte count register equals zero' trigger. If the SIMUL I/O switch is off, the channel drops the 'operational out' line, causing all attached I/O units to perform a reset operation; when the 6-usec reset is completed, the channel starts polling operations.

With the AUTO/TEST switch in AUTO, the 'auto mode switch' level inhibits the manual pulse AND gate, preventing generation of the 'manual pulse SS' signal (Diagram 8-2). This prevents generation of the 'reset suppress out latch' and the 'interface reset' and 'machine reset' signals, thus disabling the RESET pushbutton.

When the AUTO/TEST switch is set to TEST, the 'not auto mode switch' signal enables one input to the 'manual pulse' latch. When the doublepole, doublethrow RESET pushbutton is pressed, the 'reset channel switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 170-ns 'manual pulse SS' signal. This signal is AND'ed with the 'reset channel switch' signal to generate the 'suppress out reset', 'interface reset,' and 'machine reset' signals (ALD ER111).

The 'reset suppress out latch' signal resets the 'suppress out' latch, if on. (The 'suppress out' latch must be in a reset condition for the channel to generate the 'machine reset' and interface reset' signals.) The 6-usec 'machine reset' signal resets all channel registers and triggers (and latches) except the simulate interface register and the 'byte count register equals zero' trigger. If the SIMUL I/O switch is off, the channel 'interface reset' signal causes the channel to drop the I/O interface 'operational out' signal for a 6-usec period. During this period, the attached I/O units perform a reset operation. After the 6-usec 'interface reset' signal, the channel starts polling operations.

If the SIMUL I/O switch is on, the channel 'interface reset' signal does not cause the 'operational out' signal to drop, and the attached I/O units are not reset.

ATTN (Attention) Pushbutton

- ATTN pushbutton initiates channel simulation of polling interrupt sequence; sequence ends when simulated status byte with 'device end' bit active is on bus-in lines.
- Pushbutton active when AUTO/TEST switch set to TEST and SIMUL I/O switch set to on.
- When active, pressing pushbutton generates 'manual pulse SS' signal to turn on 'set attention' latch.
- Channel simulates 'select in' signal, causing 'select out' signal to drop.
- 'Simulate attention' latch is set.
- Simulated 'select in' signal drops; internal 'select out' signal rises.
- 'Simulate operational in' latch turns on; internal 'operational in' signal rises.
- 'Simulate address in' latch turns on; internal 'address in' signal rises.
- 'Polling interrupt request' latch turns on.
- Channel clock turns on.
- Zeros forced to bus-out lines.
- 'Bus out to bus in' signal plus 'manual set bit parity bus in' signal gates simulated unit address of zero to bus-in lines.
- Simulated unit address gated into unit address register at T2 clock time.
- 'Command out' signal internally raised at T3 clock time.
- 'Prepare status in' latch turns on.
- 'Command out delayed' signal rises 250 ns after 'command out' signal rises.
- 'Simulate address in' latch turns off; internal 'address in' signal drops.
- 'Simulate status in' latch turns on; 'status in' signal rises.
- 'Manual set bit 5' line to bus-in latches activated to simulate status byte with active 'device end' bit.
- 'Command out' signal is activated.
- 'Status in,' 'select out,' and 'operational in' signals drop.
- Simulated polling interrupt sequence ends with channel in wait state and polling interrupt request pending.

The ATTN pushbutton (Figure 6-1) enables the channel to simulate an I/O interrupt sequence. The simulated sequence ends when the channel simulates receipt of a status byte with the 'device end' bit active. The ATTN switch is active only when the AUTO/TEST switch is set to TEST and the SIMUL I/O switch set to the on position.

With the AUTO/TEST switch set to AUTO, the 'auto mode switch' level inhibits generation of the 'manual pulse SS' signal (Diagram 8-2). This prevents turn-on of the 'set simulate attention' latch (Diagram 8-8), effectively disabling the ATTN pushbutton.

If the SIMUL I/O switch is off, the 'set simulate attention' and 'simulate attention' latches are held in the reset condition by the 'reset simulate attention' level; this prevents the ATTN pushbutton from initiating the

simulated polling interrupt sequence, even though the pushbutton is enabled when the AUTO/TEST switch is set to TEST. Diagram 8-10 shows channel timing for a simulated polling interrupt sequence.

Assume that the SIMUL I/O switch is on, deactivating the 'reset simulate attention' level. With the AUTO/TEST switch set to TEST (Diagram 8-2), one input to the manual pulse AND is enabled. When the ATTN pushbutton is pressed, the 'attention switch' signal from one pole of the pushbutton is OR'ed to enable a second input to the AND. At the same time, a second pole of the pushbutton generates an 'any switch on' signal to satisfy the AND and turn on the 'manual pulse' latch. In turn, the leading edge of the 'manual pulse latch' signal triggers a singleshot to produce a 270-ns 'manual pulse SS' signal. This signal is AND'ed with the 'attention switch' signal to turn on the 'set simulate attention' latch (Diagram 8-8). When the channel 'select out' latch is turned off, due to simulated generation of the 'select in' signal by the channel, the 'set simulate attention latch', 'not select out latch', and 'not select out delayed' signals are AND'ed to turn on the 'simulate attention' latch.

The 'simulate attention' latch disables the 'simulate select in' AND, causing the 'simulate select in' signal to drop; in turn, the channel 'select in' signal drops, allowing the 'select out' latch to turn on. With the 'select out' latch turned on, the 'select out latch', 'simulate attention', and 'not polling interrupt' signals are AND'ed to turn on the 'simulate operational in' and 'simulate address in' latches. Turn-on of these latches raises the 'operational in' and 'address in' signals within the channel. The 'address in' signal then turns on the 'polling interrupt request' latch.

To time a portion of the simulated polling interrupt sequence, the channel clock is turned on. Turn-on is performed by AND'ing the 'polling interrupt request latch' and 'address in' signals to raise the 'turn-on clock direct' signal. The use of clock timing signals is described later in the following text.

During the polling interrupt sequence, the channel simulates receipt of a unit address from the control unit and simulates sending a command byte of all zeros to the control unit. For the channel to accomplish these operations: (1) the 'operational in' signal raises the 'force zeros to bus out' signal; (2) the 'simulate address in' signal raises the 'bus out to bus in' signal; and (3) the 'polling interrupt request' and 'simulate address in' signals are AND'ed to raise the 'manual set parity bus in' signal.

With the 'force zeros to bus out' signal active, all bit values (except the parity bit) on the bus-out lines are at logic zero levels. The parity bit is at a logic one level. (The logic one level on the 'bus out parity' line is significant for checking command byte parity, when the channel simulates sending the command byte to the control unit.) The 'force zeros to bus out' signal remains active during the remaining portion of the simulated polling interrupt sequence.

The 'bus out to bus in' signal (Diagram 8-7) gates the values on the bus-out lines to the bus-in latches via the 'manual set bit 0' through 'manual set bit 7' lines. This gating represents unit address zero from the control unit. The 'bus out parity' bit gated by the 'bus out to bus in' signal is at a logic zero level. (Although the 'bus out parity' bit on the bus-out lines is at a logic one level, the 'bus out parity' bit presented for gating to the bus-in latches is at the logic zero level.) To provide correct parity for the simulated unit address byte (address zero), the 'manual set bit parity bus in' line is raised (Diagram 8-7) by AND'ing the 'polling interrupt request' and 'simulate address in' signals as previously described.

With a simulated unit address of zero in the bus-in latches, the channels 'gate bus in to unit address register' latch is turned on by the clock 'T2' and 'not T3' signals. As a result, the unit address register is reset and the simulated address is gated into the register.

Subsequently, the clock T3 signal internally raises the channel 'command out' signal to simulate sending a command byte of all zeros to the control unit. When the 'command out' signal rises, it is AND'ed with the 'simulate address in' signal to turn on the test logic 'prepare status in' latch (Diagram 8-8). This latch remains on until the channel 'simulate status in' latch is turned on later in the polling interrupt sequence. In addition to turning on the 'prepare status in' latch, the 'command out' signal triggers a test logic 250-ns singleshot. The output of the singleshot turns on the 'command out delayed' latch. When the 250-ns singleshot signal falls, the 'command out delayed latch' signal is AND'ed to produce a 'command out delayed' signal.

The 'command out delayed' signal resets the 'simulate address in' latch, causing the internal 'address in' signal to drop. With the fall of the 'address in' signal, the 'turn on clock direct' signal falls, causing the channel clock to turn off. In turn, the fall of the T3 clock signal causes the 'command out' signal to drop. When the 'command out' signal drops, the test logic 'command out delayed' latch is turned off, causing the 'command out delayed' signal to drop (Diagram 8-8).

With the above operations complete, the channel is prepared to simulate the status-in byte (with the 'device end' bit active) from the control unit. When the 'command out delayed' signal falls, the 'prepare for status in', 'not simulate address in', 'not command out delayed', and 'not service in' signals are AND'ed to turn on the 'simulate status in' latch. With this latch on, the internal 'status in' signal is raised. In addition, the 'simulate status in' signal is AND'ed with the 'simulate interface', 'not test I/O latch', and 'polling interrupt request' signals to raise the 'simulate unit free' signal (Diagram 8-7). The 'simulate unit free' signal is then OR'ed to activate the 'manual set bit 5' line to the bus-in latches. Since bit 5 of the status byte is the 'device end' bit, the channel now has on the bus-in lines a

simulated status byte with the 'device end' bit active. To simulate stacking the status byte back to the control unit, the 'status in' signal (AND'ed with the 'polling interrupt request' signal) turns on the 'command out end' latch (ALD CC117), causing the 'command out' signal to activate. The 'command out' signal generates the 'command out delayed' signal (Diagram 8-8) as previously described to turn off the 'simulate status in' latch. This causes the device-end status byte to be removed from the bus-in latches.

Prior to turn-off of the 'status in' signal, the PIT ('polling interrupt') latch was turned on (ALD AT111) and the I/O interface 'suppress out' signal activated by the 'delay service in' signal (ALD CC121). The resulting 'polling interrupt' signal turns off the 'select out' latch, causing this signal to fall. In turn, the 'select out delayed' signal drops, causing the 'simulate operational in' latch to turn off. This deactivates the internal 'operational in' signal.

At this point, the simulated polling interrupt sequence is complete. The channel has simulated stacking status at the control unit and remains in the wait state until the channel is reset by another operation. At the end of the sequence, the 'suppress out' and 'polling interrupt request' signals are active. In other words, the simulated channel polling interrupt sequence has progressed to the point where the channel has a pending polling interrupt request. Since the 'simulate CPU' signal prevents the channel from raising the 'interrupt request' signal to the CPU interface, the polling interrupt request cannot be cleared except by initiating a channel operation which results in a machine reset operation.

Log On Machine Check Switch and Stop Control Logic

Note: For 2860B Ver 001 (Channel Indirect Data Address feature) machines, see Chapter 4, "Features."

- LOG ON MACH CHK switch is active for both auto and test modes.
- For auto mode, on position of switch permits channel to perform logout sequence when a channel machine check condition is detected.
- For manual mode, on position of switch stops channel operation when channel machine check or channel data check condition is detected.
- With LOG ON MACH CHK switch off, AND's receiving 'machine check' and 'channel data check' signals are disabled.
- For auto-mode operation, channel 'stop' signal is raised by machine check condition (LOG ON MACH CHK switch on) when the following channel conditions are not present:
  1. FLT mode.
  2. CPU interrupt response to other than logout interrupt request.
  3. Program-controlled interrupt.
  4. Interrupt in channel.
- For auto and manual modes, 'stop' signal stops channel operations.
- For auto mode: 'stop' signal, 'stop gated' signal ('stop' and 'not request storage') and 'log on stop switch' level initiate a logout operation.
- At end of logout operation, channel raises 'degate stop' signal, 'stop' signal drops, and channel continues termination of operation in progress before logout operation.
- For manual mode: 'stop' signal is raised by 'machine check' signal subject to same channel conditions in auto mode; 'stop' signal also raised by AND'ing 'channel data check' and 'test' signals.
- For manual mode, 'stop' signal stops channel operations. Logout logic is inhibited by 'test' signal preventing logout operation. Channel remains stopped until reset by another operation.

The LOG ON MACH CHK switch (Figure 6-1) is active for both auto-mode and test-mode operations. However, the function of the switch and associated circuits differs for the two operations.

For auto-mode operation, the LOG ON MACH CHK switch allows the operator to determine whether the channel is to perform a logout sequence when a channel machine check condition is detected. If the LOG ON MACH CHK switch is off (up), the channel will not perform a logout sequence. If the switch is on in the auto mode, a 'channel log' signal is raised at the CPU interface. This signal is used only with the System/360, Model 75, to light an indicator on the system console, indicating that the channel logout function is operable. When the LOG ON MACH CHK switch is on during auto-mode operation, detection of a channel machine check condition stops channel operations, and a logout sequence is performed. (See "Logout".)

If the LOG ON MACH CHK switch is on during test-mode operations, channel operations are stopped when either a channel machine check or channel data check condition is detected. With the channel in a stopped condition, channel static information is displayed on the CE panel indicators. For conditions which cause a machine check or data check, refer to "Error Checking Conditions" in Section 2.

Diagram 8-3 shows the LOG ON MACH CHK switch and associated stop control logic. If the LOG ON MACH CHK switch is off (up) for either auto-mode or manual-mode operations, the 'not log on stop switch' level disables two AND's, one of which receives the channel

'machine check' signal and the second of which receives the 'channel data check' signal. With these AND's disabled, the 'stop' signal and 'stop gated' signal cannot be raised. These signals, along with the 'log on stop switch' level, are required for the logout sequence to be performed during auto-mode operation. The 'stop' signal is required to stop channel operations during manual-mode operations. Thus, with the LOG ON MACH CHK switch off, the channel is allowed to terminate an auto-mode operation without performing a logout sequence upon detecting a channel machine check condition; for manual-mode operations, the channel is allowed to terminate operations without stopping upon detecting a channel machine check or channel data check condition.

Assume the LOG ON MACH CHK switch is on and the channel is operating in the auto mode (AUTO/TEST switch to AUTO). The 'test' signal is inactive, disabling the AND receiving the 'channel data check' signal (Diagram 8-3). This prevents the channel from initiating a logout sequence upon detection of a 'channel data check' condition. However, one input to the AND receiving the channel 'machine check' input is enabled by the 'log on stop switch' level. A second input to the AND ('not block stop') is operation-oriented and determines: (1) whether the channel enters a logout sequence when a machine check condition is detected, and (2) when the channel is to resume operations following a logout sequence. Channel conditions and operations which activate the 'block stop' signal to prevent a logout operation are as follows: (1) fault locating test operation (active 'FLT mode' signal); (2) program-controlled interrupt condition or interrupt condition resulting from other than a detected interface control check ('block stop' signal); (3) an active 'CPU interrupt response' signal to clear an interrupt condition other than the one initiated by a logout operation ('CPU interrupt response' and 'not log wait powered' signals AND'ed); (4) an interrupt condition resulting from a detected interface control check ('interrupt' and 'interface control check SS' signals AND'ed to turn on the 'inhibit stop' latch). If none of these operations or conditions are present in the channel, an active 'machine check' signal is AND'ed with the 'not block stop' and 'log on machine check' signals to raise the 'stop' signal.

The 'stop' signal performs two general functions: stops channel operations from progressing further; and enables the 'log controls and sequencing' logic. The first function inhibits the channel from recognizing interrupt conditions occurring after the 'stop' signal is raised. This is necessary to prevent the 'block stop' signal from being activated. In addition, the 'log wait interrupt' latch is turned on, the channel clock stopped (if on), and the generation of 'interface reset' and 'machine reset' signals are inhibited. The second function conditions the 'log gating controls and sequencing' logic to begin a logout sequence when the 'stop gated' signal rises. The 'stop gated' signal is activated by AND'ing the 'stop' and 'not storage request' signals. Gating the 'stop' signal with the 'not storage request' signal ensures that the logout sequence does not begin while the channel has a pending storage request not involving the logout operation. When the logout operation is complete, the channel raises the 'degate stop' signal, causing the 'block stop' signal to degate the 'machine reset' AND. This causes the 'stop' signal to drop and allows the channel to complete termination of the operation in progress. For a description of the logout operation, refer to "Logout".

For test-mode operations, the 'stop' signal is raised by a channel 'machine check' condition as described for auto-mode operation. However, the 'log controls and sequencing' logic is inhibited by the 'test' signal and a logout operation is not performed; instead, the channel remains in a stopped condition until a channel reset ('machine reset' signal) is initiated. If a channel data check condition is detected, the 'channel data check', 'log on stop switch' and 'test' signals are AND'ed (Diagram 8-3) to raise the 'stop' signal. This stops channel operations, and a logout operation is not performed. When the channel operations stop during test mode as a result of the 'channel data check' signal, the channel remains stopped until reset.

Auto Restart Switch and Control Logic

- AUTO RST switch (for test-mode operations) causes start I/O operation every 16.7 ms.
- On position of AUTO RST enabled by ground level supplies by TEST position of AUTO/TEST switch.
- 'Time clock step' signal (approximately 200-ns duration, 16.7 ms frequency) from CPU interface required for auto-restart operations.
- 'Time clock step' signal AND's with 'test' and 'restart switch' signals to:
  1. AND with the 'not storage cycle' signal to raise the 'turn on machine or setup reset' signal.
  2. Turn on 'test restart' latch (when 'start I/O' latch is off).
- 'Turn on machine or setup reset' signal turns off 'suppress out' latch (if on) and activates 6-usec 'machine reset' and 'interface reset' signals.
- After reset signals fall, 'block start I/O' latch is turned off; 200-ns 'turn on start I/O' signal is generated.
- 'Turn on start I/O' signal turns on 'simulate start I/O' latch; in turn, 'start I/O' latch is turned on.
- 'Start I/O latch' signal turns off 'test restart' signal.

- Start I/O operation performed subject to CE panel switch settings; a new start I/O operation is initiated each 16.7 ms.

The AUTO RST (auto restart) switch (Figure 6-1) enables the channel to initiate a start I/O operation every 16.7 ms. The switch is active only in the test mode, and the channel must be connected to a CPU interface for the auto-restart function to operate. Each start I/O operation, which begins with the generation of a 'machine reset' signal, is subject to the conditions described under "Start I/O Pushbutton", except that: (1) a new start I/O operation is initiated every 16.7 ms and (2) an error-caused stop condition lasts only until the next reset pulse. The channel cannot be stopped for a panel indicator visual readout during auto-restart operations; thus, the auto-restart function serves primarily as a scoping aid for troubleshooting or alignment.

With the AUTO/TEST switch set to AUTO, the on (down) position of AUTO RST switch receives +3V dc from the AUTO/TEST switch to disable the AUTO RST switch (Diagram 8-4). In the TEST position, the AUTO/TEST switch supplies a ground level to the on position of the AUTO RST switch, thus enabling the switch.

Assuming the channel is in the test mode, the AUTO RST switch is on, and the 'time clock step' signal is received via the CPU interface, auto-restart operations are initiated. The 'restart switch' level and 'test' level enables two inputs of an AND. The third input to the AND is the 'time clock step' signal, which is approximately 200 ns in duration and rises every 16.7 ms. When the signal rises, the output of the AND: (1) is AND'ed with the 'not storage cycle' signal to raise the 'turn on machine or setup reset' signal; and (2) turns on the 'test restart' latch (as soon as the 'start I/O' latch is turned off by the 'machine reset' signal. The 'turn on machine or setup reset' signal resets the 'suppress out' latch and activates the 6-usec 'machine reset' and 'interface reset' signals. In addition, the 'turn on machine reset or setup reset' signal (approximately 200-ns duration) turns on the 'block start I/O' latch. The on condition of the 'block start I/O' latch prevents initiation of the start I/O operation until the 6-usec machine reset and interface reset operations are complete.

At the completion of the reset operations, the 'not machine or setup reset' signal turns off the 'block start I/O' latch. With this latch off, the 'not block start I/O', 'test restart', and 'not turn on machine or setup reset' signals are AND'ed to trigger a 200-ns singleshot. The output of the singleshot (a 200-ns 'turn on start I/O' signal) turns on the 'simulate start I/O' latch (Diagram 8-2). From this point, the start I/O operation is as described under "START I/O Pushbutton"; i.e., the channel 'start I/O' latch is turned on and a start I/O operation is performed in the manner determined by other CE panel switch settings.

With the turn-on of the channel 'start I/O' latch, the 'test restart' latch (Diagram 8-4) is turned off. The auto-restart logic is now in an initial condition, ready to initiate a new reset and start I/O operation when the unit 'time clock step' signal arrives. As long as the AUTO/TEST switch remains in TEST and the AUTO RST switch is on, a new start I/O operation is initiated every 16.7 ms in the manner previously described.

Simulate Storage Switch and Control Logic

- SIMUL STOR switch active in the test-mode only; TEST position of AUTO/TEST switch supplies ground to on position of SIMUL STOR switch.

- With SIMUL STOR switch on (test mode), the following BCU interface signals are inhibited by 'simulate storage switch' level:
  1. 'BCU advance pulse' and 'LCS advance'.
  2. "Storage data check', 'storage address check', 'storage invalid address', and 'address protect storage'.
  3. SBI bits, SBO bits, and 'store', 'storage request', and 'CDA priority' signals. (For the System/360 Model 91, the 'pre-CDA' and 'control word request' signals are inhibited.)

- 'BCU response' and 'BCU data request' signals to channel are not inhibited, since these simplex signals are generated only in response to 'storage request' signal.

- Simulated 'BCU response' signal turns off 'storage request' latch rather than 'BCU data request', which is not received or generated.

- Since simulated CAW fetch cycle is not performed for simulate-storage start I/O operation:
  1. CAW program validity checking and storage protect register ingating is inhibited (inhibit 'turn on TIC' signal).
  2. 'Gate CAW address' signal is inhibited.

- 'Simulate storage' test logic generates 'simulate BCU response', 'simulate BCU response SS', and 'simulate advance' signals.

- Test-mode start I/O, manual fetch, or manual store operation turns on 'storage request' latch; after approximately 250 ns, 'simulate BCU response' signal rises to raise internal 'BCU response' signal.

- On fall of 'simulate BCU response', 400-ns 'simulate BCU response SS' signal rises; this turns on 'remember BCU response' latch and activates 'remember BCU response SS' signal.

- On fall of 'simulate BCU responses SS' signal, 'simulate advance pulse' signal is generated; this raises internal 'BCU advance pulse'.

- For not-simulate-storage operations, inhibits are removed from BCU interface signals, simulate-storage logic is inhibited, and channel may access main storage.

The SIMUL STOR (simulate storage) switch, shown in Figure 6-1, is active only during test mode and is used to cause the channel to either simulate main storage or allow channel access to main storage. With the SIMUL STOR switch on (down), the channel inhibits all BCU interface input and output lines and internally generates the 'storage cycle' signals normally received from the BCU interface. If the SIMUL STOR switch is off, the BCU interface input and output lines are not inhibited and the channel has access to main storage. A test-mode simulate storage or not-simulate storage operation (fetch or store storage cycle) is initiated by a simulated start I/O read-type or write-type operation (see "START I/O Switch"), a manual store operation (see "Store Pushbutton"), or a manual fetch operation (see "Fetch Pushbutton").

The on position of the SIMUL STOR switch (Diagram 8-5) is activated by a ground level received via the TEST position of the AUTO/TEST switch. Assuming the SIMUL STOR switch is on (down) and the AUTO/TEST switch is set to TEST, the 'simulate storage switch' level inhibits the following multiplex signals from the BCU interface, preventing signals which may appear on these lines from entering the channel:

1. 'BCU advance pulse' signal
2. 'Storage data check' signal
3. 'Storage address check' signal
4. 'LCS advance' signal
5. 'Storage invalid address' signal
6. 'Address protect storage' signal
7. SBO bits (by OR'ing to raise the 'simulate storage or manual store or load data address' signal).

Internally generated 'storage cycle' signals are prevented from reaching the BCU interface lines by the 'simulate storage switch' level as follows:

1. 'Storage request' signal
2. SBO bits
3. 'CDA priority' signal

The simplex 'BCU response' and 'BCU data request' signals from the BCU interface are not inhibited at the channel. Inhibiting is not required since these signals can be generated by the unit connected to the BCU interface only in response to a 'storage request' signal. (Recall that the 'storage request' signal is prevented from reaching the BCU interface.) Without the 'BCU response' signal from the BCU interface, gating of the mark-A register bits to the mark lines, of the storage protect register bits to the 'storage protection' lines, and of the 'store' signal is not possible. (The outputs of these registers and the 'store' signal are gated by the 'BCU response powered' signal, which is not activated when the channel simulates generation of the 'BCU response' signal. See ALD CC115.)

Other functions performed by the 'simulate storage switch' level are described in the following text. This level enables the simulated 'BCU response' signal to turn off the 'storage request' latch. Normally, the latch is turned off by AND'ing the 'accept' and 'BCU data request' signals from the BCU interface; however, recall that the 'BCU data request' signal is not received from the BCU interface. (The channel does not simulate generation of the 'BCU data request' signal, since it is not necessary for internal simulation of a storage cycle.)

The 'simulate storage switch' level also inhibits activation of the 'turn on TIC' signal (ALD CS113). This signal is normally activated during a start I/O CAW fetch cycle to check the program validity of the CAW and to gate the storage protect bit (contained in the CAW) into the storage protect register. For simulate-storage start I/O operations, the channel does not simulate fetching of a CAW but proceeds to immediately fetch the CCW from the CE panel SBO switches; therefore, the 'turn on TIC' signal is inhibited to prevent an erroneous detection of a program check condition. Since the channel does not simulate fetching of the CAW, the 'simulate storage switch' level also inhibits generation of the 'gate CAW address' signal. For not-simulate-storage start I/O operations, this signal is raised to gate the CAW address to the SAB during the CAW fetch storage cycle.

As previously stated, control logic associated with the SIMUL I/O switch (Diagram 8-5) simulates required BCU interface input signals for simulate storage operations. These signals are: 'simulate BCU response', 'simulate BCU response SS', and 'simulate BCU advance'. To generate these signals, a storage cycle must be initiated as a result of a start I/O read-type or write-type operation, a manual fetch operation, or a manual store operation. When a storage cycle is initiated by any of these operations, the 'storage request' latch is turned on.

The resulting 'storage request' signal is AND'ed with the 'not Z-address latch gated' and 'not simulate BCU response SS' signal to trigger a 250-ns singleshot. The output of the singleshot inhibits the 'simulate BCU response' AND for the 150-ns period and, after a logic time delay, triggers a 350-ns singleshot. The output of the 350-ns singleshot enables the 'simulate BCU interface' AND. When the output of the 150-ns singleshot falls, the AND (which also receives the 'simulate storage switch' level) is satisified and the 'simulate BCU response' signal rises. This signal is OR'ed by channel operating logic (Diagram 8-1) to raise the 'BCU response' signal. In turn the 'BCU response' signal causes the channel to perform internally the operations normally performed upon receipt of a 'BCU response' signals from the 'BCU interface'.

When the 'simulate BCU interface' signal falls, it triggers a 400-ns singleshot (Diagram 8-5) to generate the 'simulate BCU response singleshot' signal. The 'simulate BCU response singleshot' signal turns on the channel operating logic 'remember BCU response' latch and activates the 'remember BCU response SS' signal (Diagram 8-1). For not-simulate-storage operation, turn-on of the 'remember BCU response' latch and activation of the 'remember BCU response SS' signal are accomplished by the 'BCU response' signal received from the BCU interface. For simulate storage operations, this 'BCU response' signal is not raised (ALD CC115).

The 400-ns 'simulate BCU response' signal also disables the 'simulate advance pulse' AND (Diagram 8-5) and, after a logic time delay, triggers a 600-ns singleshot. The output of the 600-ns singleshot enables the 'simulate advance pulse' AND. When the 'simulate BCU response' signal falls, the AND is enabled, and the 'simulate advance pulse' signal rises. The 'simulate advance pulse' signal is OR'ed by channel operating logic (Diagram 8-1) to raise the internal 'advance pulse' signal. This signal causes channel operations normally caused by an 'advance' signal from the BCU interface (i.e., generates the 'raw advance SS', the 'advance' the 'late advance', and the 'storage cycle complete' signals; internal channel functions normally performed by these signals are performed during the simulate storage operation.)

If the SIMUL STOR switch is off during test-mode operations, the 'not simulate storage switch' level removes the inhibit levels from the BCU interface input and output lines and from the affected internal channel logic previously described. In addition, the 'not simulate storage switch' level inhibits the 'simulate BCU response' AND, preventing generation of the 'simulate BCU response', 'simulate BCU response SS' and 'simulate BCU advance' signals. This means that any storage cycle initiated in test mode will cause the channel to access main storage in the normal manner.

Test Clock Logic

- Channel clock is cycled manually by connecting jumper wire with AUTO/TEST switch set to TEST.
- Clock generates T0 through T7 signals; T7 signal raises 'manual turn off clock' signal to turn channel clock off.
- 'Not T0' signal activates 'manual turn on clock' signal; after 150-ns delay clock turns on and repeats cycle. Cycling is continuous until jumper wire is removed.

Provisions for manual cycling of the channel clock are provided by channel test logic (Diagrams 8-5). To initiate the manual cycling, a jumper wire must be connected between ground and a common input to the 'manual turn on' clock and 'manual turn off clock' AND's. The clock can be cycled manually only in test mode. With the 'test' signal active and the jumper wire connected, clock signals T0 through T7 are generated in the normal sequence. When the T7 signal is generated, the 'manual turn off clock' signal rises to turn off the clock. With the drop of the T0 signal, the 'manual turn on clock' signal is raised; however, clock singleshots hold the clock off for 150 ns. At the end of the 150-ns period, the 'manual turn on clock' signal starts the clock, and the T0 through T7 signals are again generated. As long as the 'test' signal is active and the jumper wire connected, the clock continues to cycle as described above. This method of running the clock is used only for testing or adjusting clock delay lines. To perform other test mode operations, the jumper wire must be removed.

Storage Bus-Out Switches and Logic Control

- CE panel SBO switches 0 through 63 plus P0 through P7 simulate CCW's, data doublewords, or data addresses for channel test-mode operations.
- For simulate-storage mode:
  1. Start I/O operation causes channel to fetch CCW from SBO switches; if write-type command is specified, channel fetches data doublewords from SBO switches.
  2. Manual store operation causes channel to gate data doubleword from SBO switches into A-register; A-register data is not stored.
  3. Load data address operation causes channel to gate values in ADDRESS FIELD SBO switches to data address register.
- For not-simulate-storage mode manual-store operation, the channel gates values in the SBO switches to the A-register for storage in main storage; a load data address operation is normally performed first to gate the values in the ADDRESS FIELD SBO switches to the data address register.

The CE panel 'storage bus out' (SBO) switches (0 through 63 plus P0 through P7 shown on Figure 6-1) are used to simulate CCW's, data doublewords, and data addresses for certain test-mode operations. Though the SBO switches are used to simulate both CCW doublewords and data doublewords, they are labeled on the CE panel in groups identifying a CCW format. Switches P0 and 0 through 7 are designated COMMAND FIELD, switches P1, P2, P3, and 0 through 23 are designated ADDRESS FIELD, switches P4 and 32 through 39 are designated FLAG FIELD, and switches P6, P7, and 48 through 63 are designated COUNT

FIELD. Switches P5 and 40 through 47 have no other designation, since these bits are of no significance in the CCW.

The SBO switches (Diagram 8-6) are enabled by +3V dc supplied to the logic one (down) position of all switches when the AUTO/TEST switch is in TEST. Values set into the SBO switches are gated to the corresponding channel SBO lines by the 'simulate storage or manual storage or load data address' signal.

For the simulate-storage mode, the SBO switches are the source of all CCW's and data doublewords for fetch storage cycles. If a simulated start I/O operation is initiated while the channel is simulating storage, the channel fetches a CCW from the SBO switches and, if a write-type command is specified, also fetches data doublewords from the SBO switches. In this case, the SBO switches must be set for a properly formatted data doubleword. If a 'manual store' operation is performed while the channel is simulating storage, a data doubleword is gated from the SBO switches into the A-register and represents the data to be "stored". In this case, the data is not actually stored, but is lost when the A-register is reset. If a manual fetch operation is performed while the channel is simulating storage, the data doubleword is "fetched" from the SBO switches and gated into the A-register.

If the channel is not simulating storage, a manual store operation causes the SBO switch values to be gated into the A-register and subsequently stored in main storage. Prior to the manual store operation, a load data address operation may be initiated. In this case, the values in the ADDRESS FIELD switches are gated to the data address register and identify the storage location at which the manual store data is to be stored.

Simulate Interface Register and Control Logic

The simulate interface register and associated 'manual set bits' logic (Diagram 8-7, FEDM) provides the channel with a means of simulating byte transfers across the I/O interface during test-mode simulate interface operations. In addition, the register and logic are used in auto-mode operation during diagnostic control from the CPU interface.

The simulate interface register consists of nine latches (designated bit 0 through bit 7 and parity bit). Inputs to these latches are the outputs of the 'bus out' latches (data out bus buts 0 through 7 and P).

Simulate I/O Write

- For simulate I/O write-type operations, data bytes are transferred via bus-out latches to simulate interface register; as each byte is gated into register, previous byte is reset.
- 'Write', 'service out latch' and 'simulate service in' signals are AND'ed to generate 'ingate reset' and 'gate bus out to bus in' latch.
- Last byte of write operation is retained by simulate interface register.

During simulate interface write-type operations, the data bytes normally transferred to the control unit are gated into the simulate interface register. For each byte transfer, the 'write', 'service out latch', and 'simulate service in' signals are AND'ed (Diagram 8-7) to generate an 'ingate reset' signal and turn on the 'gate bus out to simulate interface register' latch. The 'ingate reset' signal unlatches the simulate interface register latches, while the 'gate bus out to simulate interface register' signal gates the 'data out bus' bits into the register. When the 'ingate reset' signal falls, turn-off of the 'gate bus out to simulate interface register' latch is delayed by a logic delay; this delay permits the simulate interface register to latch the 'data out bus' bits into the register latches. During the write-type operation, the simulate interface register is reset each time a byte is transferred to the bus-out latches, and the new byte is entered into the register. Thus each byte stored in the register is lost when the next byte is ingated. The last byte transferred during the write-type operation is retained by the simulate storage register.

Simulate I/O Read

- Simulate interface register outputs are continuously gated to 'simulate interface' AND's in bus-in latches for read and read CDA operations.
- AND's at bus-in latches enabled by 'simulate interface' signal when:
  1. Channel is in test mode and simulating I/O interface.
  2. Channel is in auto mode and under diagnostic control of CPU.
- 'Simulate read gate' signal gates simulate interface register outputs through 'manual set bits' gates.
- Signal activated for 'read, not CDA' by AND'ing 'not latch status byte trigger', 'read', 'sequence 2', 'not sequence 5', 'not sequence 1', and 'not BC equals CTB trigger' signals.
- For 'read CDA', 'CDA latch' signal prevents 'BC equals CTB trigger' signal from dropping 'simulate read gate' signal.
- Simulate interface register signals gated to SBI for storage during auto-mode logout operations.

During a read operation (both auto and test modes), the outputs of the simulate interface register are continuously gated to 'simulate interface' AND's in the bus-in latches (via the 'manual set bits' gates) until the last data byte of a read or read CDA operation is gated into the channel B-register. Although the simulate interface register outputs are gated to the bus-in latches for both test-mode and auto-mode operations, the 'simulate interface' AND's at the bus-in latches are only enabled when: (1) the channel is in test mode and the SIMUL I/O switch is on, or (2) the channel is in auto mode and under diagnostic control of signals from the CPU interface. For auto mode read-type operations not involving diagnostic control, the simulate interface register bits are inhibited at the bus-in latches, even though they are gated through the 'manual set bits' gates.

The simulate interface register outputs are gated to the bus-in latches by the 'simulate read gate' signal. This signal is activated during read operations not involving data chaining by AND'ing the 'not latch status byte trigger', 'read', 'sequence 2', 'not sequence 5', 'not sequence 1', and 'not BC equals CTB trigger' signals. For read operations involving data chaining, turn-on of the CDA latch (turned on before the 'BC equals CTB' trigger is turned on), prevents the 'BC equal CTB trigger' signal from dropping the 'simulate read gate' signal. In other words, the simulate interface register contents are continuously gated to the bus-in latches until the last data byte of a 'read, not CDA' operation is gated to the B-register or until the last data byte of a 'read CDA' operation is gated to the B-register. If, during the read-type operation, a detected error causes the channel to enter a sequence 5 routine, the 'simulate read gate' signal is dropped to degate the 'manual set bits' gates.

Simulate interface register outputs are also supplied to the SBI gating logic. During an auto-mode logout operation, simulate bits 0-7 are stored in logword 2 (bit positions 56-63) and the simulate parity bit in logword 3 (bit position 63).

Simulate I/O Unit Address Bytes

- For test-mode simulate interface operations, 'manual set bits' gates provide means of simulating unit address bytes from I/O device.

- For start I/O or test I/O initial selection:
  1. Unit address gated from UNIT ADDRESS switches to unit address register.
  2. Unit address register contents gated to bus-out latches.
  3. Bus-out bits ('data out bus' bits) gated through 'manual set bits' gates when 'simulate address in' signal is generated.
  4. 'Manual set bits' 0-7 plus P applied to bus-in latch AND's; simulates unit address from I/O device.

- For simulated polling interrupt sequence, unit address 0 is simulated by AND'ing 'simulate unit address' and 'polling interrupt' request to activate' manual set parity bit' signal; other 'manual set bits' gates are disabled.

In addition to providing gating for the simulate interface register outputs, the 'manual set bits' gates (during test-mode simulate storage operations) enable the channel to simulate a unit address byte.

A unit address byte is simulated by the channel for simulated test I/O and start I/O operations or a simulated polling interrupt sequence when the channel SIMUL I/O switch is on. For a simulated start I/O or test I/O operation, the channel simulates selection of an I/O device by first gating the contents of the UNIT ADDRESS switches into the unit address register. During the I/O device selection, the channel gates the unit address register contents to the bus-out lines ('data out bus' bits on Diagram 8-7), then simulates the 'address in' signal from the control unit. The 'simulate address in' signal raises the 'bus out to bus in' signal. This signal gates the 'data out bus' bits from the bus-out latches to the 'manual set bits' 0 through 7 plus parity lines. These lines are applied to AND's enabled by the 'simulate interface' signal at the bus-in latches. Thus, the channel has simulated receipt of a unit address from a control unit normally received during an initial selection routine.

For a simulated polling interrupt sequence (initiated by pressing the ATTN pushbutton with the SIMUL I/O switch on), the channel simulates receipt of unit address 0 from the control unit. During the polling interrupt sequence, the channel simulates the 'address in' signal normally received from the control unit, and turns on the 'polling interrupt request' latch. While the 'simulate address in' signal is active, the channel bus-out latches contain all zeros. At the 'manual set bits gates' logic (Diagram 8-7), the 'simulate address in' signal raises the 'bus out to bus in' signal as during a simulated test I/O or start I/O operation. However, since all zeros are on the 'data out bus bits' lines (including the parity bit line), all 'manual set bits' lines remain at the logic zero level when the 'bus out to bus in' signal is raised. To provide a unit address with correct parity, the 'simulate address in' and 'polling interrupt request' signals are AND'ed to raise the 'manual set parity bus in' line. With a unit address of zero (with correct

parity) in the bus-in latches, the channel gates the unit address into the unit address register, as during a normal polling interrupt sequence.

Simulate I/O Status Bytes

- For test-mode simulate interface operation, channel simulates status byte for:
  1. Start I/O initial selection rotuine (all zeros with correct parity).
  2. Sequence 5 routine (device-end status byte with correct parity).
  3. Simulated polling interrupt sequence (device-end status byte with correct parity).

- For start I/O initial selection routine:
  1. Status byte is generated when channel generates 'simulate status in' signal.
  2. 'Simulate status in' signal is AND'ed to raise 'manual set parity bit' signal; other 'manual set bits' gates are degated.
  3. 'All zeros' status byte (manual set bits 0 through 7 and logic one P) gated through bus-in latches by 'simulate interface' signal.

- For polling interrupt sequence:
  1. 'Simulate address in' signal is AND'ed to raise 'simulate unit free' signal.
  2. 'Simulate unit free' signal is OR'ed to raise 'manual set bit 5' line ('device end' bit); other 'manual set bit' lines are degated.
  3. Device-end status byte gated through bus-in latches by 'simulate interface' signal.

During test-mode simulate interface operations, a status byte is simulated by the channel for a simulated start I/O initial selection routine, a sequence 5 routine or a polling interrupt sequence. For the start I/O initial selection routine, a status byte of all zeros with correct parity is simulated. For the sequence 5 routine and polling interrupt sequence, a status byte with the 'device end' bit active is simulated.

During the simulated start I/O routine, the channel performs the initial selection routine, simulating selection of the I/O device as though a control unit was being accessed. At the point in the initial selection routine where the initial status byte is simulated, the channel simulates the 'status in' signal from the I/O device. The 'simulate status in' signal (Diagram 8-7), enables one input of a 'manual set parity bus in' AND gate which also received the 'polling interrupt request' and 'simulate unit free' signals as inputs. To enable the AND, the two latter signals must be inactive. Since the 'polling interrupt request' latch is not on during the start I/O initial selection routine, the 'not polling interrupt request' signal enables a second input to the AND. With the 'start I/O latch' signal active and the 'polling interrupt request' signal inactive, the 'simulate unit free' AND is disabled; thus, the 'not simulate unit free' signal satisfies the 'manual set parity bus in' AND to raise the 'manual set parity bus in' signal. Since all other 'manual set bits' gates are disabled, the bus-in latches receive an initial status byte of all zeros with correct parity.

For a polling interrupt sequence, only the 'manual set bit 5' AND ('device end' bit) is enabled when the channel generates the 'simulate status in' signal. To generate the 'device end' status byte, the 'simulate unit free' signal (Diagram 8-7) is raised by AND'ing the 'simulate status in', 'simulate interface', 'not test I/O latch', and 'polling interrupt request' signals. In turn, the 'simulate unit free' signal is OR'ed to raise the 'manual set bit 5' signal. Since all other 'manual set bits gates' are degated, the 'manual set bits 0 through 7' outputs represent a status byte with only the 'device end' bit active. The 'manual set bits' status byte is gated through the bus-in latches by the 'simulate interface' signal. Since the bus-in latches are not latched for this operation, the 'simulate unit free' signal is sent to the bus-in latches where it activates the 'data in bus bit 5' signal to the 'bus in parity checking' logic.

The simulated device end status byte for a sequence 5 routine is generated in the manner described for a polling interrupt sequence, with the following exception: the 'simulate unit free' AND is enabled by the 'sequence 5 ungated' and 'not start I/O latch' signals, rather than the 'polling interrupt request' signal.

Diagnostic Simulate I/O

- Simulate interface register and 'manual set bits' gates are used in auto mode under diagnostic control by CPU.

- 'Simulate interface' signal is active.

- 'Simulate read gate' signal gates simulate interface register contents to bus-in latches.

- When 'diagnose reverse data parity' signal is activated, parity bit from simulate interface register is reversed to insert incorrect parity.

- Bus-in parity checking logic is inhibited by 'diagnose reverse data parity' signal; SBI parity checking logic should detect 'channel data check' condition.

With the channel in auto mode, the simulate interface register and 'manual set bits' gates are used when the channel is performing diagnostic operations under control of the CPU 'maintenance control word' (MCW). For a complete description of the diagnostic operations, refer

to "Channel Diagnostics" in Section 2 of this Chapter. The following text is limited to a description of the 'diagnose reverse data parity' signal (Diagram 8-2) and its function at the simulate interface register and 'manual set bits gates' logic.

During CPU diagnostic testing, the channel 'simulate interface' signal is activated (Diagram 8-8) by the 'diagnose simulate interface' signal to enable the channel to simulate read and write byte transfers without access to a control unit. Under CPU diagnostic control, the channel is instructed by a Start I/O instruction to perform read or write operations. When a read operation is performed, the 'simulate read gate' signal is raised (Diagram 8-7) and data in the simulate interface register is gated to the bus-in latches via the 'manual set bits' gates. When the 'diagnose reverse data parity' signal is raised, the 'simulate parity' bit from the simulate interface register is reversed, causing a data bit with incorrect parity to be gated through the bus-in latches to the B-register. (The simulate interface register should, for this operation, contain a byte with correct parity from a previous write operation.) Since the purpose of the 'diagnose reverse data parity' signal is to check the channel SBI parity checking logic, the 'diagnose reverse data parity' signal also inhibits the bus-in latch parity checking logic (ALD CK117). This prevents detection of a channel data check condition due to the bus-in latch parity check, and permits the SBI parity check logic to detect the incorrect parity.

Simulate I/O Switch and Control Logic

- SIMUL I/O switch (when on) and associated logic simulate I/O operations when channel is in test mode.
- Active 'simulate interface' signal:
  1. Inhibits incoming I/O interface signals at channel.
  2. Inhibits outgoing I/O interface signals, except 'operation out' signal and bus-out lines.
  3. Enables simulate-interface logic.
  4. Inhibits incorrect-length detection logic.
- Simulated start I/O operation initiates simulated I/O device selection routine.
- Channel turns on 'address out' latch, causing generation of 'address out delayed' signal.
- Channel turns on 'select out' latch, causing generation of 'select out delayed' signal.
- 'Simulate operational in' latch turns on; 'simulate address in' latch turns on.
- 'CCW valid' latch turns on; 'command out' latch turns on; 'prepare status in' latch turns on; 'prepare for status in' signal is activated.
- 'Command out delayed' signal is generated; 'simulate address out' latch turns off.
- 'Command out' signal falls; 'command out delayed' signal falls.
- 'Simulate status in' latch turns on; simulated status byte of all zeros gated to bus-in latches.
- For write or read operations, first 'service out' signal activates 'service out delayed' signal.
- 'Simulate status in' latch turns off.
- 'Response delay' latch turns on; 250 ns later 'simulate service in' latch turns on; 'response delay' latch turns off.
- For write, 'simulate service in' signal enables gating of data byte to simulate interface register; for read, 'simulate service in' signal allows channel to gate byte from simulated interface register to B-register.
- 'Service out' latch turns on; 'service out delayed' signal is activated; simulate service in' latch turns off; 'service out' latch turns off.
- Above sequence continues until last byte is transferred.
- 'Simulate status in' latch turns on; channel enters sequence 5 ending routine; 'simulate status in' latch causes channel to simulate receipt of 'device end' status byte.
- 'Select out' latch turns off; 'select out delayed' signal is deactivated; 'simulate operational in' latch turns off; 'simulate status in' latch turns off.

The SIMUL I/O switch (Figure 6-1) and associated control logic enables the channel to simulate operations with an I/O device. The switch is active only in the test mode. When the switch is on in the test mode, the channel inhibits all incoming and outgoing I/O interface signals (except the 'operational out' signal and bus-out lines). In addition, for any simulated channel operation requiring access to a control unit, the channel simulates generation of incoming I/O signals as though they had been received from an attached I/O device.

Many of the test circuits associated with the SIMUL I/O switch was previously described under "Attention Pushbutton" and "Simulate Interface Register and Control Logic". This paragraph describes the SIMUL I/O switch and associated logic operation for a simulated start I/O routine. The description is based primarily upon positive logic Diagram 8-8. Diagram 8-9 shows the timing for a simulated interface operation initiated by a simulated Start I/O instruction. Timing is shown for the initial selection routine and for service-in/service-out exchange for a read-type or write-type data transfer. For ease of identification, timing bars shown for signals generated by normal channel operating logic and test logic associated with the SIMUL I/O switch are different.

The on position of the SIMUL I/O switch (Diagram 8-8) is activated (receives a ground level) when the AUTO/ TEST switch is in TEST. With the SIMUL I/O switch on, the 'simulate I/O switch' signal is OR'ed to raise the 'simulate interface' signal. This signal enables the associated test logic shown on Diagram 8-8. In addition, the 'simulate interface' signal: (1) enables generation of the 'simulate unit free' signal at the 'simulate interface register and control' logic (Diagram 8-7); (2) inhibits the I/O interface 'select in', 'service in', 'operational in', 'status in', and 'address in' signals from entering the channel; (3) inhibits the internally generated 'select out' and 'address out' signals from reaching the I/O interface and raises the 'operational out' signal to the I/O interface; (4) inhibits generation of the 'incorrect length' signal (residual count operations are not performed during simulate interface operations); (5) enables the 'manual set bits' gates so that the channel can simulate read byte transfers, unit address bytes, and status bytes from the control unit; and (6) prevents the channel from gating a 'stop command' byte (all zeros) to the I/O interface during a sequence 5 routine. routine.

For a start I/O operation, the 'simulate start I/O' latch is turned on. (Refer to "START I/O Pushbutton".) When the 'start I/O' initial selection routine reaches the point where the channel turns on the 'address out' latch ('not operational in' signal and clock signal T4 active), the 'address out latch' signal causes the test logic to generate an 'address out delayed' signal. This signal prevents generation of the 'simulate select in' signal and, later in the sequence, enables turn-on of the 'simulate operational in' signal. The 'address out delayed' signal (Diagram 8-8) is generated as follows. The 'address out latch' signal triggers a 250-ns singleshot. The output of the singleshot turns on the 'address out delayed' latch and inhibits an AND to prevent the 'address out delayed' signal from activating. When the 250-ns singleshot signal times out, the 'address out delayed' signal activates and remains active until the 'address out latch' signal falls.

When the channel turns on its 'select out' latch (at T7 clock time), the test logic 150-ns select out delayed single-shot is triggered (Diagram 8-8). After 150-ns, the 'select out delayed' signal is activated; this signal is AND'ed with the 'address out delayed' signal to turn on the 'simulate operational in' latch. With the 'simulate operational in' signal active, the reset to the 'simulate address in' latch is removed, and the 'not PIT request', 'setup, not operational in', and 'simulate operational in' signals are AND'ed to turn on the 'simulate address in' latch. At the 'manual set bits' gates (Diagram 8-7), this causes the unit address on the bus-out lines to be gated to the bus-in latches.

As during a normal start I/O selection routine, the channel obtains a valid CCW, turns on the 'CCW valid' latch and channel clock and, subsequently, turns on the 'command out' latch. At the 'simulate I/O switch' logic (Diagram 8-8), the 'command out' and 'simulate address in' signals are AND'ed to turn on the 'prepare status in' latch. The output of this latch is AND'ed with the 'simulate operational in' signal to activate the 'prepare for status in' signal. With this signal active, the test logic is conditioned to simulate generation of the 'status in' signal later in the simulated I/O device selection routine.

The 'command out' signal also triggers a 250-ns 'command out delayed' singleshot. When the 250-ns singleshot times out, the 'command out delayed' signal turns off the 'simulate address in' latch to degate the simulated unit address from the bus-in latches and drop the internal 'address in' signal.

When the channel deactivates the 'command out' signal, the 'command out delayed' signal drops. As a result, the 'not command out delayed', 'not simulate address in', 'not simulate service in', and 'prepare for status in' signals are AND'ed to turn on the 'simulate status in' latch. The output of this latch activates the internal 'status in' signal and gates a status byte of all zeros with correct parity through the 'manual set bits' logic to the bus-in latches.

Following activation of the 'status in' signal, the channel turns on the 'sequence 1' latch and clock, then latches the all-zeros status byte into the bus-in latches (as during normal operation). At T2 clock time the channel turns on the 'sequence 2' latch. If a write operation is to be performed, the channel fetches two data doublewords (if specified by the count) before attempting to clear the simulated initial status byte. If a read command is indicated, the channel proceeds to clear the simulated initial status byte after the 'sequence 2' latch is turned on. In either case, the channel clears the initial status byte by turning on the 'service out' latch. In the test logic (Diagram 8-8), the 'service out latch' signal triggers a 150-ns 'service out delayed' singleshot. The singleshot signal turns on the 'service out delayed' latch. When the singleshot times out, the 'not singleshot', 'service out delayed latch', 'simulate interface', and 'not status in

end' signals are AND'ed to activate the 'service out delayed' signal.

The 'service out delayed' signal resets the 'simulate status in' latch. This causes the internal 'status in' signal to deactivate and degates the 'manual set bits' gates to remove the all-zeros status byte to the bus-in latches. In addition, the 'service out delayed' signal is AND'ed with the 'simulate operational in' signal to trigger a 250-ns singleshot. The singleshot output turns on the 'response delay' latch and prevents turn-on of the 'simulate service in' latch.

When the 250-ns singleshot times out, the 'not single-shot', 'response delay latch', 'sequence 2', and 'read or write' signals are AND'ed to turn on the 'simulate service in' latch. The resulting 'simulate service in' signal turns off the 'response delay' latch and raises the internal 'service in' signal. In turn, the 'service in' signal turns on the 'service out' latch. If a write operation is specified, the 'simulate service in' signal enables a data byte to be gated from the B-register (via the bus-out latches) into the simulate interface register. For a read operation, the 'service out latch' signal latches a data byte (gated from the simulate interface register) into the bus-in latches. As during normal read operations, the data byte is subsequently gated into a B-register byte location.

To continue the read or write operation, the 'service out latch' signal again generates the 'service out delayed' signal as previously described. The 'service out delayed' signal turns off the 'simulate service in' latch, causing the internal 'service in' signal to deactivate and, in turn, causing the 'service out' latch to turn off. The 'service out delayed' signal is also AND'ed with the 'operational in' signal to trigger the 250-ns singleshot, which turns on the 'response delay' latch.

Assuming that further byte transfers are to be simulated, the 'simulate service in' latch is again turned on as previously described and the 'service in/service out' exchange is again simulated.

For a read operation, turn-on of the 'BC=CTB' latch, (with the 'CDA latch' off) indicates that the last byte of the start I/O operation has been read into the B-register. For this condition, the 'BC=CTB latch', 'read', and 'not CDA latch' signals are AND'ed by the test logic to prevent turn-on of the 'simulate service in' latch. The output of this AND is also AND'ed with the 'response delay latch', 'read', 'not 250-ns singleshot', and 'sequence 2' signals to turn on the 'simulate status in' latch. (See Diagram 8-8). At this point in the simulated start I/O operation, the channel enters a sequence-5 ending routine. Turn-on of the 'simulate status in' latch during the sequence 5 routine causes the channel to simulate receipt of a device-end status byte. (See "Simulate Interface Register and Control Logic.")

During the sequence 5 routine, the channel operating logic performs a simulated I/O interface disconnect operation. During the disconnect operation, the channel turns off the operating logic 'select out' latch. This causes the test logic 'select out delay' latch to turn off and, consequently, the gated 'select out delayed' signal to be deactivated. The 'not select out delay' level turns off the test logic 'simulate operational in' latch. In turn, the 'not simulate operational in' signal, turns off the 'simulate status in' latch to complete the simulated start I/O operation.

For a write operation, the 'simulate service in' latch is turned on by the 'service out latch' signal (as previously described) after the last byte has been gated to the B-register. The 'simulate service in' signal activates the internal 'service in' signal. With the 'sequence 5' latch on, the channel internal 'command out' signal is activated to simulate sending a stop command byte to the I/O interface. In the simulate I/O test logic (Diagram 8-8), the 'command out' signal is AND'ed with the 'simulate service in' signal to turn on the 'prepare status in' latch. The 'prepare status in latch' signal is then AND'ed with the 'operational in' signal. In the meantime, the 'command out delayed' signal is generated as previously described, and turns off the 'simulate service in' latch. This deactivates the internal 'service in' signal which, in turn, causes the 'command out' and 'command out delayed' signals to deactivate.

To simulate the device-end status byte, the 'prepare status in', 'not simulate service in', 'not simulate address in', and 'not command out delayed' signals are AND'ed to turn on the 'simulate status in' latch. From this point, the completion of the I/O interface disconnect operation is as described for the read operation.

If the channel enters a sequence 5 routine upon detecting a check condition, the channel simulates the device-end status byte and performs a simulated I/O interface disconnect operation as described for the write operation.

UNIT ADDRESS Switches

- Nine UNIT ADDRESS switches (0 through 7 plus P) active in test mode.
- 'Simulate CPU' signal gates values in unit address switches to channel 'unit address bus bit' lines; inhibits CPU interface 'unit address bus out' bits.
- Switches used for simulated start I/O and test I/O operations.

The nine UNIT ADDRESS switches (Figure 6-1), active only in test mode, enables the channel to simulate the unit address byte normally received from the CPU interface 'unit address bus out' lines.

The logic one (down) positions of the UNIT ADDRESS switches are activated by +3V dc when the AUTO/TEST switch is in TEST. The TEST position also activates the 'simulate CPU' level. This level enables the AND's

receiving the 'unit address switch 0 through 7 and parity' levels. At the same time, the 'simulate CPU' level inhibits the AND's receiving the 'unit address bus out' bits from the CPU interface. Thus, the values in the UNIT ADDRESS switches are present on the channel internal 'unit address bus bit 0 through bit 7 and parity' lines at all times when the channel is in test mode.

The UNIT ADDRESS switches provide the unit address bits for simulated start I/O and test I/O operations. If the channel is not simulating I/O interface operations, the switches should be set for the address of an I/O device attached to the channel; otherwise an incorrect-selection condition will result.

When the channel is simulating I/O interface operations, the UNIT ADDRESS switches may be set for any value. For both simulate and not simulate interface operations, correct parity should be set into the switches. If not, a bus-out parity check condition will result in a channel control check condition.

Test Indicators Switch

- TEST IND switch is active for both test and auto modes.
- When switch is on, all CE panel indicators (except the A REGISTER 0-63 and B REGISTER 0-63 indicators) should light.

The TEST IND (test indicators) switch, shown in Figure 6-1, is used to test all CE panel indicators except the A REGISTER and B REGISTER bit indicators. The parity indicators (P0-P7) for both registers are checked when the TEST IND switch is on. The TEST IND switch is active for both auto-mode and test-mode operations.

The on (down) position of the TEST IND switch (Figure 6-3) receives +6V dc when the channel is in either auto-mode or test-mode operation. When the switch is on, the 'indicator test' level is activated and OR'ed through all applicable indicator drivers to light the CE panel indicators. Figure 6-3 shows the OR indicator driver for the test panel P IRPT (polling interrupt) indicator as well as the indicator. During channel operations, the P IRPT indicator normally lights when the 'polling interrupt' latch is on. With the 'indicator test' level active, the lamp should light; otherwise the indicator or indicator driver may be defective. The 'indicator test' signal turns on the other CE panel indicators in the manner described for the P IRPT indicator. Except when testing the indicators, the TEST IND switch should be off to allow the indicators to show the progress of channel operations.

CE PANEL TEST INDICATORS

The CE panel indicators present a visual indication of the channel's operating condition and are for use primarily as a troubleshooting aide. Each indicator lights when the function indicated by the indicator is active in the channel. When the TEST IND switch is on, all indicators light except the A REGISTER bit 0-63 and B REGISTER bit 0-63 indicators.

Each indicator has an associated indicator driver configuration which receives an 'operating signal' input. Figure 6-4 shows three logic configurations of indicator drivers used to drive the CE panel indicators. The type 1 configuration is used for A REGISTER and B REGISTER bit 0-63 indicators, where only the operating bit signal drives the indicator. The type 2 configuration is used primarily for the DATA ADDRESS, SIMULATE I/O REGISTER, BUS IN STATUS, and UNIT ADDRESS indicators. The type 3 configuration is used for the remaining indicators. Note that both the type 2 and type 3 configurations have the 'indicator test' signal as an input as well as the operating logic output. The CE panel indicators are described in the following paragraphs.

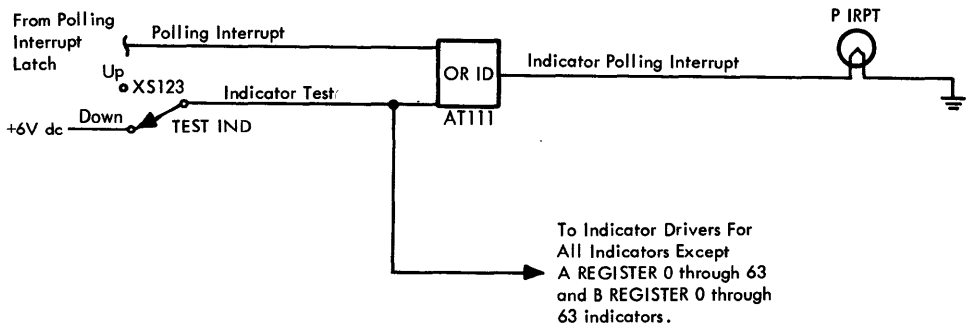


Figure 6-3. Test Indicator Switch

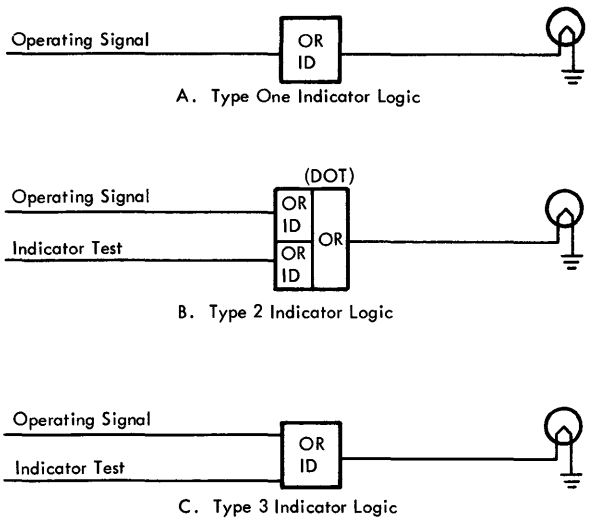


Figure 6-4. Indicator Drivers Logic

PROTECTION KEY Indicators

PROTECTION KEY indicators P and 0-3 (Figure 6-1) display the storage protection key contained in the storage protect register. Although the storage protect register (ALD RP111) consists of 9 latches (0-7 plus parity), the storage protection key consists only of bits 0-3 plus the parity bit. For a valid storage protection key, bits 4-7 must be all zeros. Therefore, only the storage protection key bits are displayed on the CE panel PROTECTION KEY indicators. The storage protect key is obtained from the CAW and entered into the storage protect register when a start I/O operation is initiated.

All PROTECTION KEY indicators should light when the TEST IND switch is on.

COMMAND ADDRESS Indicators

COMMAND ADDRESS indicators 0-20 plus P0, P1, and P2 (Figure 6-1) display the contents of the channel command address register. The command address register (ALD's RC111-RC115) is used during start I/O operations and contains the storage address plus eight bytes (one doubleword) of the CCW currently in the channel. The register contents are used to fetch a new CCW if chaining is specified by the current CCW. If chaining is specified, the COMMAND ADDRESS indicators "ripple" as the channel updates the command address for each CCW fetched from storage.

All COMMAND ADDRESS indicators should light when the TEST IND switch is turned on.

BUS IN STATUS Indicators

The BUS IN STATUS indicators (Figure 6-1) display the contents of the bus-in latches (ALD's UB111 - UB115). Address byte, data byte, and status byte transfers through the bus-out latches are displayed on the BUS IN STATUS indicators. At the end of an operation, the ending status byte is displayed and the panel labels correspond to the status bits. Indicator labels are P (parity bit). ATTN (attention bit), MOD (modifier bit), CUE ('control unit end' bit), BUSY (control unit busy bit), CH E ('channel end' bit), DEV E ('device end' bit), U CHK ('unit check' bit), and U EXC ('unit exception' bit).

All BUS IN STATUS indicators should light when the TEST IND switch is turned on.

CHANNEL STATUS CHECKS Indicators

The CHANNEL STATUS CHECKS indicators (Figure 6-1) display channel check conditions detected as a result of channel operations. The individual indicators are as follows: PCI (program-controlled interrupt), IL (incorrect length), PROG (program check), DATA (data check), CTL (channel control check), I CTL (interface control check), CHAIN ('chain' check). All CHANNEL STATUS CHECKS indicators should light when the IND TEST switch is on. The principle of each indicator is described separately in the following paragraphs.

Program-Controlled Interrupt Indicator

The PCI (program-controlled interrupt) indicator lights when the 'PCI status' latch is on (ALD EN115). The PCI condition is not a true channel-check condition. The 'PCI status' latch is turned on as a result of an active PCI flag in the flag register; this causes the channel to send an 'interrupt request' signal to the CPU interface as soon as channel operations permit. Until the program-controlled interrupt is cleared, the PCI indicator remains lit.

Incorrect Length Indicator

The IL (incorrect length) indicator lights when the IL latch (ALD CK111) is on. This latch is turned on when an operation ends and the number of bytes specified by the count register value do not agree with the number of bytes actually transferred. The IL indicator remains on until the residual count is computed, the interrupt routine is completed, and a 'machine or setup' reset is generated.

Program Indicator

The PROG (program) indicator lights when the channel detects a programming error during a channel operation. (ALD CK113). Programming errors may be detected as the result of an improperly formatted CAW or CCW, a TIC command in the first CCW of a start I/O operation, two consecutive CCW's with TIC commands during chaining operations, an 'invalid address check' signal received via the BCU interface, and a polling-interrupt-request sequence resulting in an unsuccessful device-selection attempt.

Protect Indicator

The PROT (protect) indicator lights when the channel accesses main storage and sends an incorrect storage protect key to the 'storage protection key' lines (CK113). In main storage the storage protect key from the channel is compared with a key present in main storage. If the two keys do not compare, the channel receives a 'storage protection check' signal via the BCU interface. This signal sets the channel 'storage protect check' latch, causing the PROT indicator to light. (A 'storage protect key' of all zeros allows access to any storage area; no key comparison is performed in main storage and no 'storage protect check' signal should be generated.)

DATA Indicator

The DATA indicator lights when the channel detects a parity error in the data transferred during a channel operation and turns on the 'channel data check' latch (ALD CK115). The channel data check condition is detected as a result of: (1) a data parity check condition detected by main storage as a result of the channel accessing main storage; (2) an SBI parity check condition detected when the channel is performing a store operation; (3) a bus-in parity check condition detected in a data byte at the bus-in latches; (4) a bus-out parity check condition detected in a data byte at the bus-out latches.

Control Indicator

The CTL (control) indicator lights when the channel detects a check condition affecting channel controls and turns on the 'channel control check' latch (ALD CK115). This latch may be turned on by: (1) detection of a 'bus out parity check' during a unit address byte or command byte transfer through the bus-out latches; (2) an adder parity check condition detected during a data address, command address, or count adder operation; (3) a storage address check or storage protect register parity check condition detected when the channel accesses main storage; (4) a byte count parity check condition detected when the byte count is advanced; (5) a flag register parity check condition, detected when the count register contents and DAB are added during CCW setup operations; or (6) a storage data check condition detected in main storage during a CCW fetch.

Interface Control Indicator

The I CTL (interface control) indicator lights when an invalid signal or invalid signal sequence is detected at the I/O interface to turn on the 'interface control check' latch (ALD CK115). This latch can be turned on by: (1) a no-response ('service in' signal rises) condition when the channel attempts to select a device; (2) a bus-in control parity check condition detected as a result of a 'data in bus parity' check strobed by the 'service out SS' signal; (3) selection of an incorrect I/O device by the channel and detected when the two unit addresses do not compare during the selection routine; (4) the I/O device selection routine not being completed within the 100 ns time allowed; and (5) the channel 'operational in' latch and 'select out' latch being on, but the 'operational in' signal from the I/O interface falling.

CHAIN Indicator

The CHAIN indicator lights when an I/O device presents the channel with data faster than the channel can handle it (ALD CK115). This may occur during a read-not chain operation when the 'last word trigger' is on and count register bits 21-23 equal 0, or during read CDA operations when the DAB in the chained CCW is not compatible with the number of bytes already received by the channel while the CCW was being fetched.

Note: Recall that the simulate (I/O) interface register simulates I/O device transfers at the maximum transfer capability of the channel. Thus, when chaining data during simulate I/O operations, a chaining check (overrun) occurs if the chained CCW is programmed on byte boundaries or if the specified count is less than five.

COUNT Indicators

COUNT indicators P1 and 8–15 and the CT7 indicator (Figure 6-1) display the contents of the count register (ALD RE111–RE115). This register (except the count 7 position) is loaded from a CCW during a start I/O initial setup routine. Note that indicators P1 and 8–15 are located at positions B19–B36 on the CE panel while indicator CT7 (count 7) is located at position F27. The

count 7 indicator lights when a DAB plus count operation (during start I/O setup operations) results in a carryout from high-order position 8 of the normal 16-position count field. The CT7 position of the register does not receive a gated bit during the count-from-CCW bit gating.

After the DAB plus count operation is performed, the COUNT indicators "ripple" as the count register is updated during data doubleword transfer operations. After the start I/O operation is complete, the COUNT indicators indicate either (1) a count register content of all zeros with correct parity if a normal ending occurred, or (2) a residual count in the count register if the number of bytes specified by the count are not actually transferred. All COUNT indicators should light when the TEST IND switch is on.

MARK-B Indicators

MARK-B indicators P and 0–7 (Figure 6-1) display the contents of the mark-B register (ALD's RM115–RM117). For read operations, each position of the mark-B register is set, position by position, as the register is loaded byte by byte. For store operations, the mark-B register contents are transferred into the mark-A register and sent to the BCU to indicate which byte positions in the doubleword are to be placed in main storage.

On write CDA operations, the channel transfers the five low-order count bits to the mark-B register while the new CCW is being fetched. In this case, the mark-B register assumes the function of the count register. All MARK-B indicators should light when the TEST IND switch is on.

COMMAND REGISTER Indicators

COMMAND REGISTER indicators P and 0-7 display the contents of the command register (ALD's RG111–RG113). The command is gated into the command register during a start I/O operation from the CCW. If the channel is performing chain command operations, the indicators will "ripple" as each new command is entered into the command register from the chained CCW. Otherwise, the register contents remain the same throughout the start I/O operation. All COMMAND REGISTER indicators should light when the TEST IND switch is on.

FLAG REGISTER Indicators

FLAG REGISTER indicators P, CD, CC, SLI, SKIP, and PCI (Figure 6-1) display the flag positions of the flag register (ALD RF111). The flag register (nine positions) contents are obtained from the CCW during a start I/O operation. Since the three low-order bits of the flag register should be zeros and are used only for parity checking, these positions are not displayed on indicators. The P indicator lights when the flag register parity bit is active, the CD indicator when the chain data flag is active, the CC indicator when the chain command flag is active, the SLI flag when the suppress-incorrect-length flag is active, the SKIP indicator when the skip flag is active, and the PCI indicator when the program-controlled-interrupt flag is active. These flags determine channel operations to be performed during or after completion of channel operations specified by the CCW from which the flags were obtained. For example, an active CD flag specifies that a new CCW is to be fetched to continue data transfers controlled by the channel because of the current CCW. The CC flag specifies that a new CCW with possibly a different command be obtained by the channel when the command specified by the current CCW is completed. (If both the CD and CC flags are active, the CD flag has precedence.) The SLI flag prevents the channel from recognizing that all data bytes specified by the count have not been transferred, and is useful when the data count versus device record is known to be incompatible, or where an incorrect length is of no importance to the program. The skip flag causes the channel (during read-type operations) to skip a block of data specified by the CCW and is usually used in conjunction with chaining operation; in actuality, the skipped data is transferred from the control unit to the channel, but the data is not stored in main storage. The PCI flag causes the channel to attempt an interrupt request, depending upon the channel operation in progress. All FLAG REGISTER indicators should light when the TEST IND switch is on.

CHANNEL ADDRESS Indicators

CHAN ADR indicators 21, 22, and 23 (Figure 6-1) display the channel address wired into the channel logic (ALD RU115) in binary form. The address indicators will always show the same address unless the channel address is rewired. The channel address bits are available to the SBI for logout operations.

UNIT ADDRESS Indicators

UNIT ADDRESS indicators P and 0–7 (Figure 6-1) display the contents of the unit address register. An address is entered into the unit address register as the result of a start I/O or test I/O instruction, or as the result of unit address from a control unit attempting a polling interrupt sequence. All UNIT ADDRESS indicators should light when the TEST IND switch is on.

DATA ADDRESS Indicators

DATA ADDRESS indicators 0–23, P0, P1, and P2 (Figure 6-1) display the contents of the data address register (ALD RU115). For a start I/O operation the data address of the first CCW is gated into the data address register from the CAW. When the first CCW is fetched, the beginning main storage address location at which the channel will access data doublewords during read-type or write-type operations is gated into the register. As data doublewords are transferred throughout the read-type or write-type operations, the DATA ADDRESS indicators "ripple" as the data address register is updated. All DATA ADDRESS register indicators should light when the TEST IND switch is on.

SIMULATE I/O REGISTER Indicators

SIMULATE I/O REGISTER indicators P and 0-7 (Figure 6-1) display the contents of the simulate I/O interface register (ALD's RS111-RS113). This register contains the last byte transferred by the channel during a test-mode simulate interface write operation. The register contents are changed only during a test-mode write-type operation; the register contents are not affected by channel reset operations except as a result of entry of a new data byte into the register. All SIMULATE I/O REGISTER indicators should light when the TEST IND switch is on.

A-Register Full Indicator

The A FUL (A-register full) indicator (Figure 6-1) lights when the 'A-register full' latch (ALD LT121) is turned on. The latch is turned on during a write-type operation when a data doubleword is gated from the SBO into the A-register; for read-type operations, the latch is turned on when the channel transfers a doubleword from the B-register to the A-register. The A FUL indicator should light when the TEST IND switch is on.

B-Register Full Indicator

The B FUL (B-register full) indicator (Figure 6-1) lights when the 'B-register full' latch (ALD LT121) turns on. This latch is turned on during a write-type operation when the channel transfers the A-register contents into the B-register; on read-type operations, the latch is turned on when the channel assembles a complete doubleword in the B-register or enters the last byte of the read-type operation into the B-register. The B FUL indicator should light when the TEST IND switch is on.

CC Indicator

The CC (chain command) indicator (Figure 6-1) lights when the 'chain command' latch (ALD LT113) is turned on when the channel is chaining to fetch the next CCW. The latch is turned on only if no errors are detected during operations specified by the current CCW. The CC indicator lights when the TEST IND switch is on.

SEQUENCE Indicators

SEQUENCE indicators 1–5 (Figure 6-1) light when their respective sequence latch turns on. All SEQUENCE indicators should light when the TEST IND switch is on.

The SEQUENCE 1 indicator lights (ALD EQ111) after the initial status byte is obtained from a control unit for a channel operation during which read-type or write-type data is to be transferred. The indicator turns off when the initial status byte is cleared.

The SEQUENCE 2 indicator lights (EQ111) while the channel is controlling data transfers. The latch is turned on before the start of read-type or write-type transfers and turned off after the transfers are completed.

The SEQUENCE 3 indicator (ALD EQ111) lights when the channel is updating the count register while storing or fetching a data doubleword. For a write CDA operation, the SEQUENCE 3 indicator lights before the channel fetches the first data doubleword of the next CCW; in this case, the 'sequence 3' latch is turned on to enable turn-on of the 'last word' trigger, and an update-count operation is not performed.

The SEQUENCE 4 indicator (ALD EQ113) lights when the channel is updating the contents of the data address register as a result of storing or fetching a data doubleword.

The SEQUENCE 5 indicator (ALD EQ113) lights when (1) the channel is ending an operation and entering an interrupt routine to store the CSW, or (2) while the channel is continuing a chain command operation by fetching a new CCW, disconnecting the control unit from the I/O interface, and reselecting the I/O device.

CCW Valid Indicator

The CCWV (CCW valid) indicator (ALD LT115) lights when the 'CCW valid' latch is turned on. This latch is turned on when a properly formatted CCW is in the channe The channel may obtain a valid CCW from main storage or the SBO switches as a result of a start I/O operation, or may force a valid CCW during IPL or FLT operations. The CCWV indicator should light when the TEST IND switch is on.

Read Indicator

The RD (read) indicator (Figure 6-1) lights when the channel decodes a read command (ALD LT113) in the command register (bit 5 or 6 active) with a valid CCW in the channel. The read command specifies that the channel is to control the transfer of data from an I/O device to storage. The RD indicator should light when the TEST IND switch is on.

Write Indicator

The WR (write) indicator (Figure 6-1) lights when the channel decodes a write command (ALD LT113) in the command register (bit 7 active) with a valid CCW in the channel. The write command specifies that the channel is to control the transfer of data from an I/O device to storage. The WR indicator should light when the TEST IND switch is on.

Setup Indicator

The SU (setup) indicator (Figure 6-1) lights when the 'setup' latch (ALD CS113) is turned on. This latch is turned on during channel selection of an I/O device and fetching of the CCW for a start I/O and chain command operation. In addition, the latch is turned on for device selection during IPL, FLT, and test I/O and halt I/O operations. The SU indicator should light when the TEST IND switch is on.

Transfer in Channel Indicator

The T I CH (transfer in channel) indicator (Figure 6-1) lights when the TIC latch (CX111) is turned on. This latch is turned on under two different circumstances. During a start I/O operation, the latch is turned on during a CAW fetch so that the data address received in the CAW may be used to branch to the first CCW of the operation. The second case is when the channel is performing start I/O chaining operations and receives a CCW with a TIC command in the command field. In this case, the channel uses the data address from the CCW with the TIC command to branch to a new chain of CCW's. The T I CH indicator should light when the TEST IND switch is on.

TIC Cycle Indicator

The T CY (TIC cycle) indicator (Figure 6-1) lights when the 'TIC cycle' latch (ALD CX111) is turned on. This latch is on while the channel is actually fetching the CCW specified by the CAW or CCW TIC. The T CY indicator should light when the TEST IND switch is on.

Polling Interrupt Indicator

The P IRPT (polling interrupt) indicator (Figure 6-1) lights when the 'polling interrupt' latch (AT111) is turned on. This latch is turned on after a control unit with an outstanding status condition tries to present that status to the channel. The latch remains on until the requested interrupt has been honored or until a new instruction has been received at the channel. The P IRPT indicator should light when the TEST IND switch is on.

Chain Data Indicator

The CD (chain data) indicator lights when the 'chain data' latch (ALD LT111) is on. This latch is on when the channel is fetching the next CCW in the chain while using the CCW already in the channel to complete the data-handling specified by that CCW. The latch is turned off when the new CCW is available at the channel. The CD indicator should light when the TEST IND switch is on.

BYTE COUNT Indicators

BYTE COUNT indicators LS WD, = CT, = ZRT, R = ZR, P, 4, 2, and 1 (Figure 6-1) show the byte count register values and the overall progress of byte and doubleword transfers. The indicators are described in separate paragraphs below.

Last Word Indicator

The LS WD indicator lights when the 'last word' trigger is turned on (ALD LT113). This trigger is turned on during channel data transfer operations when the count register is decremented to a value of one doubleword or less. The trigger may be turned on during a start I/O setup routine if the initial 'count plus DAB' operation results in a count of one doubleword or less. The 'last word' trigger enables the channel logic to detect the transfer of the last data bytes. The LS WD indicator should light when the TEST IND switch is turned on.

Equal Count Indicator

The = CT (equal count) indicator lights when the 'byte count equals count B' trigger turns on (ALD LT123). This trigger turns on when the byte count latches and the

three low-order bits of the counter are equal and indicates that the last data byte has been transferred. The trigger turns off after the channel enters a sequence-5 ending routine. The = CT indicator should light when the TEST IND switch is turned on.

Zero Trigger Indicator

The = ZR T (zero trigger) indicator lights when the 'byte count latch equals zero' trigger is turned on. This trigger (ALD LT123) is turned on when the B-register is filled or emptied by a byte gated into or out of the register. The = ZR T indicator should light when the TEST IND switch is turned on.

Register Equals Zero Indicator

The R = ZR indicator lights when the 'byte count register equals zero' trigger is turned on. This trigger (ALD LT121) is turned on during a read or write operation and indicates that the channel is beginning to assemble or disassemble a new doubleword in the B-register; i.e., a doubleword boundary has been crossed during data transfer operations. The R = ZR indicator should light when the TEST IND switch is on.

BYTE COUNT P, 4, 2, 1, Indicators

BYTE COUNT P, 4, 2, 1 indicators display the binary contents of the byte count register. The register (ALD's SC111–SC113) values specify the B-register byte location for byte transfer into or from the register. During transfer operations, the P, 4, 2, and 1 indicators "ripple" as the byte count is changed for each byte transfer. These indicators should light when the TEST IND switch is on.

Clock Indicators

CLOCK indicators T0–T7 (Figure 6-1) correspond to the clock signals produced by the channel clock. Each indicator lights as the corresponding clock signal is activated and remains on until the clock is turned off. The CLOCK indicators should light when the TEST IND switch is on.

I/O INTERFACE CONTROL LINES Indicators

The I/O INTERFACE CONTROL LINES indicators (OP IN, OP O, ADR O, SEL O, SVC O, COM O, ADR IN, STA IN), shown in Figure 6-1, light when the associated I/O interface signals are active. All these indicators should light when the TEST IND switch is on.

The OP IN (operational-in) indicator (ALD CC111) lights when an I/O unit is selected by the channel and raises its 'operational in' signal.

The OP O (operational out) indicator (ALD CC121) lights when the 'operational out' signal from the channel is active. This signal is active at all times when the channel is operable except when an I/O interface reset operation is performed. (This may be due to system reset or channel reset caused by a operation initiated at the channel.)

The ADR O (address out) indicator (ALD LT111) lights when the channel activates the 'address out' signal to the control units; this signal indicates that a unit address byte is on the bus-out lines for decoding by the control units.

The SEL O (select out) indicator (ALD CC121) lights when the channel activates the 'select out' signal to the I/O device. This signal is activated when the channel is performing polling operations and is ready to accept outstanding status from a control unit, or because the channel has selected a specific I/O device. In the first case, the SEL O indicator may be turned on and off (select-out and select-in sequence) but may appear to be lit continuously. In the second case, the indicator will be turned off only if the selection attempt is unsuccessful. If not turned off, the indicator should glow continuously and appear brighter than when the channel is polling.

The SVC O (service out) indicator (ALD CC117) lights when the channel activates the 'service out' signal to the I/O interface. This signal is activated when the channel accepts a status byte or data byte from the bus-in lines. The channel also activates the 'service out' signal when providing a data byte to the bus-out lines in response to a 'service in' signal from the I/O interface.

The COM O (command out) indicator (ALD CC117) lights when the channel places a command byte on the bus-out lines and raises the 'command out' signal. The channel raises the 'command out' signal in response to an 'address in', 'status in', or 'service in' signal (depending on the progress of the operation) from the control unit.

The ADR IN (address-in) indicator (ALD CC111) lights when the 'address in' signal from the I/O interface is activated. This signal is activated when the control unit places the unit address byte of a selected I/O device on the bus-in lines.

The STA IN (status-in) indicator (ALD CC111) lights when the 'status in' signal from the I/O interface is activated. This signal is activated when the control unit places a status byte on the bus-in lines.

CCW Fetch Indicator

The CCWF indicator (Figure 6-1) lights when the 'CCW fetch' latch (ALD LT115) is turned on. This latch is

turned on each time the channel fetches a CAW or CCW. CAW and CCW fetches are made during initial setup operations; CCW fetches are also made during chaining operations. The CCWF indicator should light when the TEST IND switch is on.

No Selection Indicator

The NO SEL indicator (Figure 6-1) lights when the 'no selection' latch is turned on (ALD CC125). This latch is turned on when the channel attempts to select an I/O device and not control unit responds to the attempted selection. The NO SEL indicator should light when the TEST IND switch is on.

Halt I/O Indicator

The H I/O (halt I/O) indicator (Figure 6-1) lights when the 'CPU halt I/O' and 'CPU select channel' signals (ALD HT111) are active at the channel, or the 'simulate halt I/O' signal is active. In either case, the channel halts any operation in progress at the earliest possible time. The H I/O indicator should light when the TEST IND switch is on.

Test I/O Indicator

The T I/O indicator (Figure 6-1) lights when the 'test I/O' latch (ALD CT111) is on. This latch may be turned on by: (1) the 'test I/O' and 'CPU select channel' signals from the CPU interface; (2) by the 'polling interrupt extended' signal when the channel attempts to reselect an I/O device to complete a polling interrupt routine; or (3) by the 'simulate test I/O' signal when the channel is in test mode. The T I/O indicator should light when the TEST IND switch is on.

Start I/O Indicator

The S I/O indicator (Figure 6-1) lights when the 'start I/O' latch (ALD CS111) is turned on. This latch is turned on by: (1) 'CPU start-I/O' and 'CPU select channel' signals from the CPU interface; (2) by the 'time clock step' signal from the CPU interface when the channel is in the test mode; or (3) by the 'simulate start I/O' signal when the channel is in the test mode. The S I/O indicator should light when the TEST IND switch is on.

Storage Check Indicator

The S CHK indicator (Figure 6-1) is turned on when the 'storage check' latch (ALD CK115) is turned on. This latch is turned on when the storage unit detects a storage address check or storage data check condition and activates the corresponding signals to the channel. The S CHK indicator should light when the TEST IND switch is on.

Initial IDA Indicator (2860B Ver 001–CIDA Feature)

The initial IDA indicator (INIT IDA) lights when the channel receives a CCW with bit 37 = 1. The INIT IDA latch is reset after the first IDALW fetch is made (ALD DA111). The INIT IDA indicator lights when the test indicator (TEST IND) switch is on.

2k IDA Indicator (2860B Ver 001–CIDA Feature)

The 2k IDA indicator lights when the channel detects a 2k-byte boundary crossing. This latch (ALD DA111) is reset after the IDALW is fetched. The 2k IDA indicator lights when the test indicator (TEST IND) switch is on.

Auxiliary Storage Request Indicator

The ASR (auxiliary storage request) indicator (Figure 6-1) lights when the LCS (large capacity storage) latch is turned on. This latch (ALD CC116) is turned on when the channel requests a storage cycle from an LCS unit. The ASR latch and indicator turn off when the storage request is honored ('LCS advance pulse' signal is received by the channel). The ASR indicator should light when the TEST IND switch is on.

Interrupt Indicator

The IRPT (interrupt) indicator (Figure 6-1) lights when the 'interrupt' latch is turned on. This latch (ALD EN115) is turned on when an interrupt condition (other than a polling interrupt) is in the channel. The latch and indicator turn off when the interrupt condition is accepted or otherwise cleared. The IRPT indicator should light when the TEST IND switch is on.

Storage Cycle Indicator

The ST CY indicator (Figure 6-1) lights when the 'storage cycle' latch is turned on. This latch (ALD MA115) turns on when the channel activates the 'storage request' signal to the BCU interface (by turning on the 'storage request' or 'interrupt storage request' latch). The latch and indicator remain on until the storage request has been honored ('late advance pulse' signal is generated). The ST CY indicator should light when the TEST IND switch is on.

Retain Storage Indicator

The RET S indicator (Figure 6-1) turns on when the 'retain storage' latch is turned on. This latch (ALD LT115) turns on to prevent turn-off of the 'storage request' and 'storage cycle' latches during two storage requests of both read and write 'chain data' operations. Although, internally, the 'retain storage' signal prevents turn-off of these latches, the 'storage request' signal to the BCU interface is dropped during the 'data request' signal from the BCU interface.

Note: For 2860 Selector Channels used with the System/360, Model 91, the 'data request' signal is generated internally by delaying the 'BCU response' signal approximately 150 ns.

Program Load INIT Indicator

The PROG LD INIT indicator (Figure 6-1) lights when the channel IPL latch is on. This latch (ALD LP111) is turned on when the channel begins an IPL (initial program load) operation, and remains on until the CPU assumes control of the new program just loaded from external storage. The PROG LD INIT indicator should light when the TEST IND switch is on.

Program Load FLT Indicator

The PROG LD FLT (program load FLT) indicator (Figure 6-1) lights when the channel 'FLT mode' latch is turned on. This latch (ALD LP113) is turned on at the beginning of an FLT operation and remains on as long as the channel is operating in the FLT mode. The PROG LD FLT indicator should light when the TEST IND switch is on.

Diagnostic Indicator

The DIAG indicator (Figure 6-1) lights when the 'diagnostic simulate interface' latch is turned on. This latch (ALD CC113) is on when the channel is in auto mode and the channel is under diagnostic control of the CPU maintenance control word (MCW). The latch is turned on by AND'ing the 'diagnostic select channel' and 'setup' signals and remains on until both signals are deactivated. The DIAG indicator should light when the TEST IND switch is turned on.

Mark-A Indicators

MARK-A indicators P and 0-7 (Figure 6-1) display the contents of the mark-A register. This register (ALD's RM111 through RM113) is loaded from the mark-B register at the same time the contents of the B-register are transferred to the A-register on a store operation. Active mark bits indicate which bytes of the doubleword in the A-register are to be stored. All MARK-A indicators should light when the TEST IND switch is on.

A-REGISTER Indicators

A-REGISTER indicators 0–63 and P0–P7 (Figure 6-1) display the contents of the A-register. This register (ALD's RA111–RA137) is a buffer register between the channel B-register and main storage during data doubleword transfer operations. At the end of an operation, the A-register normally contains the last data doubleword stored in storage (for a store operation) or the last data doubleword fetched from storage (for a fetch operation). Only the A-REGISTER P0–P7 indicators should light when the TEST IND switch is on.

B-REGISTER Indicators

B-REGISTER indicators 0–63 and P0–P7 (Figure 6-1) display the contents of the B-register. This register (ALD's RB111–RB137) is used to assemble or disassemble data doublewords transferred between the channel and control unit on read-type or write-type operations, respectively. On a read-type operation, the channel assembles a doubleword in the B-register on a byte-by-byte basis; on a write-type operation, the channel disassembles the doubleword on the same basis. Only the B-REGISTER P0–P7 indicators light when the TEST IND switch is on.

POWER CONTROL PANEL

The power control panel (Figure 6-5) controls and indicators affect all channels installed within a frame. The operating principles of the power controls and indicators are described in Chapter 5. The following paragraphs describe the function of these switches and indicators and the metering logic operating principles associated with the customer usage meter and the ENABLE/DISABLE switch.

LOCAL/REMOTE Switch

The LOCAL/REMOTE switch (Figure 6-5) determines whether power-on sequencing and marginal checking can be performed at the channel power control panel (LOCAL position or at the system console (REMOTE position).

POWER ON Pushbutton

The POWER ON pushbutton (Figure 6-5) is active only when the LOCAL/REMOTE switch is set to LOCAL. Depressing the POWER ON pushbutton causes the channel to bring up power to all self-contained channel power supplies as well as to all attached control units that are not off-line. The POWER ON pushbutton is back-lighted by indicators that light when the power-on sequence is complete at the channel frame. Depressing the POWER ON pushbutton when the LOCAL/REMOTE switch is set to REMOTE has no effect.

POWER OFF Pushbutton

The POWER OFF pushbutton (Figure 6-5) is active with the LOCAL/REMOTE switch set to either position.

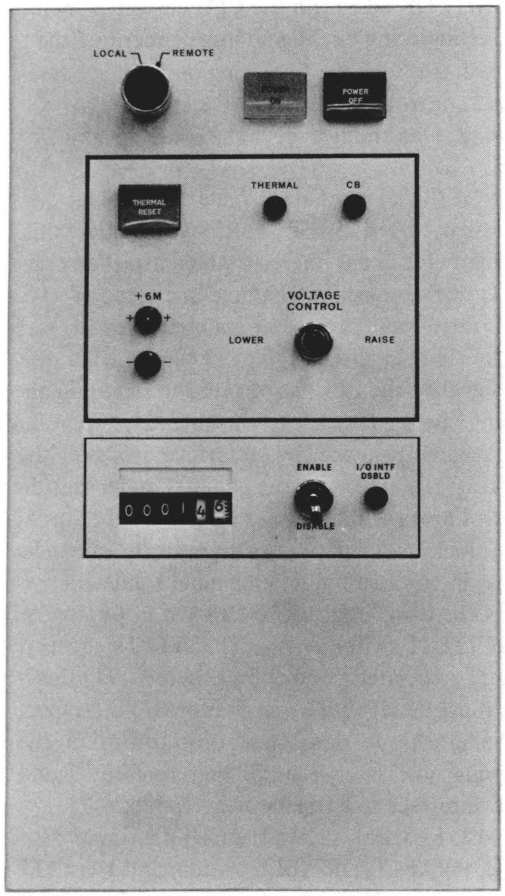


Figure 6-5. 2860 Power Control Panel

Depressing the POWER OFF switch causes the channel frame to drop power internally, as well as at all attached control units that are on-line. Since switching transients due to power-off sequencing affect system operation, the POWER OFF switch should be pressed only in an emergency unless processor operation is stopped.

THERMAL RESET Pushbutton

The THERMAL RESET pushbutton (Figure 6-5) resets a thermal condition which caused a channel frame to drop power. If a thermal switch opens in one of the channel gates or in the channel frame power section, power cannot be restored until the THERMAL RESET pushbutton is depressed.

THERMAL Indicator

The THERMAL (Figure 6-5) lights when channel frame power drops because of a thermal fault. No indication is given as to which thermal switch opened.

CB Indicator

The CB indicator (Figure 6-5) lights when channel frame power drops because of a tripped circuit breaker (CB). Power cannot be restored until the tripped CB is located and manually reset.

Marginal Voltage Test Jacks

The +6M jacks (+ and -), shown in Figure 6-5, serve to monitor the +6M Vdc marginal voltage at the 2860 without having to open the 2860 covers and gates.

VOLTAGE CONTROL Switch

The VOLTAGE CONTROL switch (Figure 6-5) raises or lowers the +6M Vdc marginal voltage when the LOCAL/REMOTE switch is set to LOCAL. Positioning the VOLTAGE CONTROL switch to RAISE causes the marginal voltage to rise above its nominal +6V dc. Positioning the switch to LOWER causes the voltage to drop. An external voltmeter must be plugged into the +6M marginal voltage test jacks as a monitor; the marginal check limits of +5V dc and +7V dc must not be exceeded.

Metering Switch, Indicator and Associated Logic

- Channel metering logic interlocked with system metering.
- Channel metering state cannot change until CPU enters halt, wait, or check-stop state. ('Clock out' signal from CPU interface drops.)
- ENABLE/DISABLE switch must be set to ENABLE, AUTO/TEST switch to AUTO, 'metering out' signal from CPU interface active, and channel metering logic enabled for channel usage meter to run.
- 'Metering out' signal to I/O interface permits control unit or I/O device meters to run if meter-enabled.
- 'Metering in' signal from I/O device is active when control unit meter is running; if channel is meter-enabled, 'metering out' signal to CPU is active, holding 'metering in' signal from CPU interface active.
- Channel 'metering sync' latch must be on for channel to be meter-enabled.
- With ENABLE/DISABLE switch to DISABLE or AUTO/TEST switch to TEST and both the 'channel working' and I/O interface 'metering in' signal inactive, the fall of 'clock out' inhibits channel-metering logic.

The channel frame metering logic consists of the ENABLE/DISABLE switch, the I/O INTF DSBLD indicator, and the associated metering logic. The channel metering logic circuits are logically interlocked with the CPU and control units metering logic so that the metering state of the channel frame and associated control units can be changed only if the CPU is in a halt, wait, or check-stop state.

Figure 6-6 shows the system metering logic interlocking the CPU, channel, and control units. The 'clock out' signal to the channels (and control units via the channel) must be inactive for the channel or control unit metering state to change: e.g., if the channel ENABLE/DISABLE switch is set to DISABLE or the AUTO/TEST switch to TEST, the 'clock out' signal must drop before the channel metering logic can stop the channel usage meter. With the usage meter not running, the ENABLE/DISABLE switch set to ENABLE, and the AUTO/TEST switch to AUTO, the 'clock stop' signal must drop before the channel metering logic can run the channel usage meter.

The channel also receives a 'metering out' signal via the CPU interface. This signal is active when the 'CPU clock out' signal is active or the 'metering in' signal from the channel is active. If the channel is meter-enabled, the active 'metering out' signal enables the channel clock to run, and also supplies a multiplex 'metering out' signal to the control units. An active 'metering out' signal enables the control unit or I/O device meter to run, providing the unit or device metering logic is meter-enabled.

If an attached control units meter is running, the control unit activates the 'metering in' signal to the channel via the I/O interface. If the channel is meter-enabled ('metering sync' latch set to activate 'not block program control' signal), the I/O interface 'metering in' signal activates the 'metering in' signal to the CPU interface. (If the channel is meter-enabled, the 'channel working' signal also activates the 'metering in' signal.) In the CPU, this signal is effectively OR'ed to run the CPU customer meter and activate the 'metering out' signal independent of the 'clock out' signal. In turn, the 'metering out' signal causes the channel and control unit (or I/O device) meter to continue running, even though the 'clock out' signal may drop.

In summary, if the channel usage meter is running, the 'metering out' signal from the CPU interface is active; this also permits the meter in any meter-enabled attached control unit or I/O device to run. An active 'metering in' signal from a control unit to a meter-enabled channel frame keeps the channel and CPU meters running; i.e., the 'metering in' signal to the CPU is activated to maintain an active 'metering out' signal back to the channel. When the 'metering out' signal from the CPU interface drops (or the 'channel working' signal drops), the channel meter and the attached control unit (or I/O device) meters will be turned off when the 'clock out' signal drops to reset the channels 'metering sync' latch (assuming the channel frame ENABLE/DISABLE switch is set to DISABLE, or the channel AUTO/TEST switch is set to TEST). To again run the channel and associated control unit meters: (1) the two switches must be set to ENABLE and AUTO (respectively); (2) the 'clock out' signal must drop to turn on the channel metering sync' latch; (3) the 'metering out' signal from the CPU interface must again be activated.

Diagram 8-12, FEMDM, shows the channel frame metering logic, ENABLE/DISABLE switch, and I/O INTF DSBLD indicator in positive logic form. Assume that the 'metering sync' latch is on and the 'metering out' signal from the CPU interface is active. The 'not block program control' signal and 'metering out' ('CPU clock running') signals are AND'ed with the 'not sync reset' signal to activate the I/O interface 'metering out' signal. The active

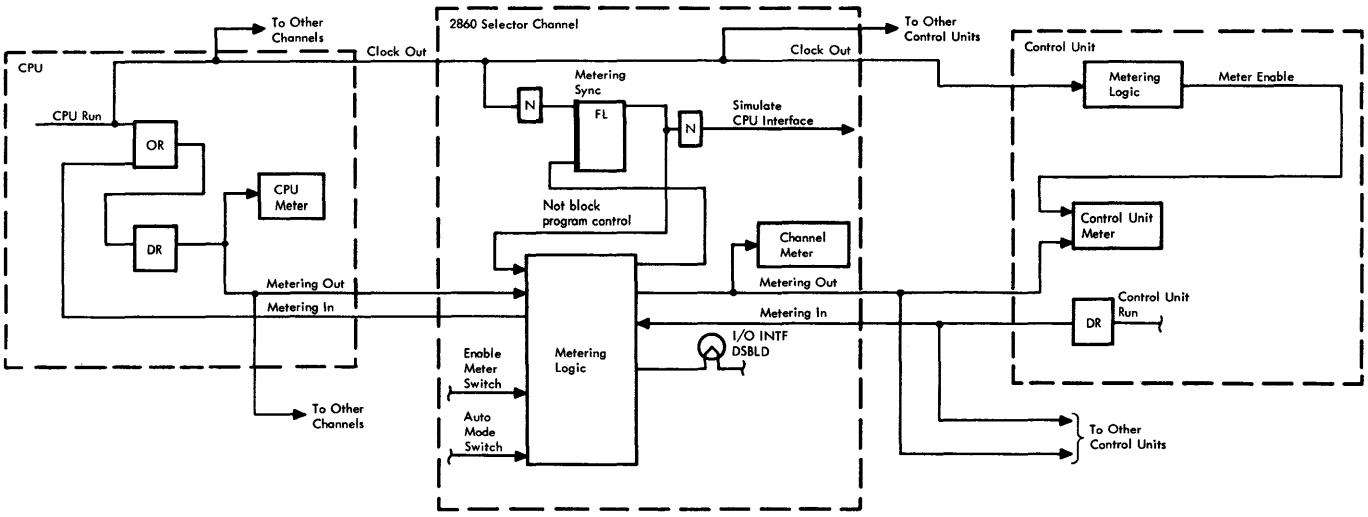


Figure 6-6. Channel/System Metering

'metering out' signal is routed to the control units via the I/O interface and also runs the channel usage meter ('run clock' signal) on the channel power panel. With the 'not block program control' and 'run internal clock multiplex' ('run clock') signal active, the I/O INTF DSBLD indicator is off.

Assume that, subsequently, either the ENABLE/DISABLE switch is set to DISABLE or the AUTO/TEST switch is set to TEST. This deactivates one input to the 'metering sync' latch reset OR and activates one input to the 'sync reset' AND. When both the channel 'channel working' signal and the 'metering in' signal from the I/O interface drop, the 'metering in' ('intermediate unit run CPU clock') to the CPU interface is disabled; with this signal disabled, a second input to the 'sync reset' AND is enabled. When the channel enters the halt, wait, or check stop state, the 'clock out' ('on clock stopped') signal from the CPU interface deactivates to completely enable the 'sync reset' AND. The 'sync reset' output inhibits the 'metering out' AND to deactivate the second input to 'metering sync' latch reset OR. With the OR completely disabled, the 'metering sync' latch is reset to produce a 'block program control' signal. This signal inhibits the 'metering in' and 'metering out' AND's, preventing the channel usage meter and attached I/O device meters from running. In addition, the 'block program control' signal activates the 'simulate CPU' signal, which inhibits the CPU interface input and output signals to and from the channel. Under these conditions, the channel is off-line for system operation.

Note: When the CPU is stopped, entry of any one channel of the 2860 Model 2 or 3 into the test mode (AUTO/TEST switch to TEST) causes the I/O INTF DSBLD indicator to light. This is not a malfunction.

To change the channel metering state to a run condition: (1) the AUTO/TEST switch must be set to AUTO and the ENABLE/DISABLE switch to ENABLE to remove the reset level from the 'metering sync' latch; and (2) the 'clock out' signal must fall ('on clock stopped' signal) to turn on the 'metering sync' latch. This produces the 'not block program control' signal which enables one input to the 'metering in' and 'metering out' AND's. With the 'not sync reset' signal enabling a second input to the 'metering out' AND, an active 'metering out' signal from the CPU interface satisifies the AND to run the channel clock and enable the control units or I/O devices clock to run as previously described.

SECTION 2. MAINTENANCE FEATURES

This section describes the function and operation of all channel maintenance operations, including logout, error checking conditions, diagnostics, channel status byte, and control unit status byte. Each operation or facility is described separately; where feasible, FEMDM diagrams are referenced.

LOGOUT OPERATIONS

- Logout operation stores three doublewords in main storage when the channel detects a machine-check condition.
- Doublewords stored at addresses 304, 312, and 320 (decimal).
- Channel operating status at time of machine-check detection not altered by logout operation.
- CPU is not released for other operations during logout operations.

The channel logout logic enables the channel to store three doublewords in main storage when the channel detects a machine-check condition during auto-mode operation. Each doubleword contains information pertinent to the condition of the channel at the time the machine-check condition is detected. During the logout sequence, the three doublewords are stored at main storage addresses 304, 312, and 320 (decimal) where they may be later accessed by the CPU for recording on a printout device, if so programmed. The printout may then be used as a maintenance aid in determining the error that caused the machine-check condition.

For a logout sequence to begin during auto-mode operation, the machine-check condition must be detected while the channel is engaged in a start I/O or test I/O operation. During logout sequence the channel operating status at the time the machine-check condition was detected is not altered, and the CPU is not released for other operations. When logout sequence is completed, the channel ends the operations in the normal manner; i.e., performs a sequence 5 routine followed by an interrupt routine to store the CSW, send the appropriate condition code and unit address to the CPU, and release the CPU.

Diagrams 8-13 through 8-15, FEMDM, are, respectively, a simplified flowchart, detailed flowchart, and positive logic diagram of the logout control logic. In the following paragraphs, the logout sequence is described at the simplified and then at the detailed levels. The simplified description is based upon Diagram 8-13; the detailed description upon Diagrams 8-14 and 8-15.

Simplified Logout Operations

- Start I/O or test I/O operation is in progress; interface control check or channel control check condition is detected.
- 'Machine check' signal is activated.
- LOG ON MACH CHK switch on, AUTO/TEST switch to AUTO, block-stop conditions not present; 'stop' signal activates to stop channel operations.
- 'Interrupt request' signal raised to CPU interface.
- Store log word 1 operation:
  1. 'Log 1' trigger turns on; 'log' trigger turns on.
  2. SAB gates enabled for address 304 (decimal): mark-A register bits enabled; storage protect parity bit enabled.
  3. Log word 1 bits gated to SBI gating logic.
  4. 'Storage request' signal activated to BCU interface.
  5. 'BCU response' signal gates address, marks, storage protect key, and 'store' signal to BCU interface.
  6. 'BCU data request' signal gates log word 1 bits to SBI gating logic.
  7. 'BCU advance pulse' signal received.
- Store log word 2 operation:
  1. 'Log 2' trigger turns on; 'log 1' trigger turns off.
  2. SAB gates enabled for address 312.
  3. Log word 2 bits gated to SBI gating logic.
  4. Remainder of store log word 2 operation same as with store log word 1 operation.
- Store log word 3 operation:
  1. 'Log 3' trigger turns on; 'log 2' trigger turns off.
  2. SAB gates enabled for address 320 (decimal).
  3. Log word 3 bits gated to SBI gating logic.
  4. Remainder of store log word 3 operation same as store log word 1 operation.
- End-logout operation:
  1. 'Log 3' trigger turns off; 'log stop' latch turns on.
  2. 'Stop' signal drops; 'interrupt request' signal drops.
  3. Channel operations resume; sequence 5 routine entered followed by interrupt routine.

Assume that the channel is engaged in a start I/O or test I/O operation (Diagram 8-13) and that it detects an interface control check or channel control check condition. Either check condition activates the channel 'machine check' signal. (See Table 6-1 for error conditions which raise the 'machine check' line.) With the LOG ON MACH

Table 6-1 Machine Check Conditions Causing Logout

Channel Control Check		Interface Control Check	
Check	Cause	Check	Cause
1. Adder	Predicted parity and actual adder output parity do not agree during command address, data address, or count update operation.	1. Bus-in parity	Status or address byte on bus-in lines has wrong parity.
2. Byte counter	Predicted parity and actual parity of updated byte counter do not agree.	2. Address compare	Address of I/O device selected by channel does not agree with address specified by unit address register contents.
3. Bus-out parity	Control byte on bus-out lines has wrong parity.	3. No response	Control unit did not respond to 'address out' signal during reselection sequence for a chain command operation.
4. Flag parity	Flag register parity is wrong.	4. Channel time-out	Channel did not activate 'release' signal within 100.05 ms of receipt of 'select channel' signal (IPL-FLT excluded).
5. CCW parity	Main storage detects parity error in CCW being fetched by channel.	5. Incorrect I/O Interface	'Operational in' signal from control unit drops before channel drops 'select out' signal.
6. Storage address	Main storage detects parity error in address on SAB, in mark bits, or in storage protect key.		
7. Storage protection parity	Channel detects parity error in storage protection key obtained from storage during CAW fetch.		

CHK switch on, the AUTO/TEST switch to AUTO, and no block-stop conditions present in the channel, the channel activates the 'stop' signal. (Block-stop conditions which prevent entry into the logout sequence are: (1) 'FLT mode' signal active; (2) active 'CPU interrupt response' signal for other than the logout sequence in the channel; and (3) the channel 'interrupt' or 'PCI' latch on.) Following activation of the 'stop' signal, the channel raises the 'interrupt request' signal to the CPU interface, and enters the store log word 1 routine.

During the store log word 1 routine (Diagram 8-13), the channel 'log 1' trigger is turned on, followed by turn-on of the 'log' trigger. Together, the 'log 1' and 'log trigger' signals enable the SAB gates required to produce address 304. The 'log trigger' signal also activates all mark-A register bits and forces the 'storage protect parity' bit. The SAB bits, mark-A register bits, and 'storage protect parity' bit are gated to the appropriate BCU interface lines when the channel receives the 'BCU response' signal later in the sequence. Log word 1 information is gated to the SBI gating logic by the 'log 1' signal and is gated to the SBI lines when the channel receives the 'BCU data request' signal later in the sequence. For the contents of log word 1, refer to Table 6-2.

After the 'log 1' trigger is turned on, the channel raises the 'storage request' signal to the BCU interface. When the 'BCU response' signal is received, address 304, all mark bits, the 'store' signal, and a storage protect key of all zeros are gated to storage on the appropriate BCU interface lines. When the 'BCU data request' signal is received (or in the case of the Model 91, generated by the channel), log word 1 is gated to the SBI lines for storage at address 304 (decimal). Receipt by the channel of the 'BCU advance pulse' signal indicates that log word 1 has been stored; the 'BCU advance pulse' signal is delayed in the channel, then used to start the store log word 2 operations.

Table 6-2 Log Words Content

Log Word 1		Log Word 2		Log Word 3	
Log Bits	SBI Lines	Log Bits	SBI Lines	Log Bits	SBI Lines
1. Storage protect register bits 0-3	0 - 3	1. Mark-B register bits 0-7	0 - 7	1. 'A-reg, full' latch	0
2. All zeros	4 - 7	2. Command register bits 0-7	8 - 15	2. 'B-reg full' latch	1
3. Command address register bits 0-20	8 - 31	3. Flag register bits 0-4	16 - 20	3. 'Chain command' latch	2
4. Bus-in latches bits 0-7	32 - 39	4. Channel address bits (wired, 3 bits)	21 - 23	4. 'Sequence 1' through 'sequence 5' triggers	3-7
5. 'Channel status' latch bits (8)	40 - 47	5. Unit address register bits 0-7	24 - 31	5. 'CCW valid' trigger	8
6. Count register bits 8-23	48 - 63	6. Data address register bits 0-23	32-55	6. 'Read' signal	9
		7. Simulate interface register bits 0-7	56 - 63	7. 'Write' signal	10
				8. 'Setup' latch	11
				9. 'TIC' latch	12
				10. 'TIC cycle' trigger	13
				11. 'Polling interrupt request' latch	14
				12. 'Chain data address' latch	15
				13. 'Last word' trigger	16
				14. 'BC equals CTB' trigger	17
				15. 'BCL equals zero' latch	18
				16. 'BCR equals zero' latch	19
				17. BCR P, 4, 2, 1 bits	20 - 23
				18. Clock latches T0 - T7	24 - 31
				19. 'Operational in' signal	32
				20. 'Address out' latch	34

Table 6-2. Log Words Content (Cont)

Log Word 1		Log Word 2		Log Word 3	
Log Bits	SBI Lines	Log Bits	SBI Lines	Log Bits	SBI Lines
				21. 'Select out' latch	35
				22. 'Service out' latch	36
				23. 'Command out' latch	37
				24. 'Start I/O' latch	38
				25. 'Halt I/O channel busy' latch	39
				26. 'CPU storage check on log' latch	45
				27. Storage protect reg P-bit	48
				28. Command address reg P0, P1 and P2 bits	49 - 51
				29. Bus-in latches parity bit	52
				30. Count reg P0, P1 and P2 bits	53 - 55
				31. Unit address reg parity bit	56
				32. Data address reg P0, P1 and P2 bits	57 - 59
				33. Mark-B reg parity bit	60
				34. Command reg parity bit	61
				35. Flag reg parity bit	62
				36. Simulate interface reg parity bit	63

The store log word 2 operations (Diagram 8-13) begin when the delayed 'BCU advance pulse' signal turns on the 'log 2' trigger and the 'log 2' signal turns off the 'log 1' trigger. The sequence for the store log word 2 operations is the same as that for the store log word 1 operation with two exceptions: (1) the 'log 2' and 'log trigger' signals enable the SAB gates required to produce address 312 (decimal) and (2) the 'log 2' signal gates log word 2 information to the SBI gates. (For the contents of log word 2, refer to Table 6-2.) Receipt by the channel of the 'BCU advance pulse' signal indicates the log word 2 has been stored; the 'BCU advance pulse' signal is delayed in the channel, then used to start the store log word 3 operations.

The store log word 3 operations (Diagram 8-13) begin when the delayed 'BCU advance pulse' signal turns on the 'log 3' trigger, and the 'log 3' signals turns off the 'log 2' trigger. The sequence for the store log word 3 operations is the same as that for the store log word 1 operation with two exceptions: (1) the 'log 3' and 'log trigger' signals enable the SAB gates required to produce address 320

(decimal); and (2) the 'log 3' signal gates 'log word 3' information to the SBI gates. (For the contents of log word 3, refer to Table 6-2.) Receipt by the channel of the 'BCU advance pulse' signal indicates that log word 3 has been stored; the 'BCU advance pulse' signal is delayed in the channel, then used to begin the end-logout operation.

The end-logout operation begins when the delayed 'BCU advance pulse' signal turns off the 'log 3' trigger, and the 'not log 3' signal turns on the 'log stop' latch. With the 'log stop' latch on, the 'stop' signal is degated, the 'interrupt request' signal to the CPU interface is dropped, and the channel 'sequence 5' latch is turned on (if not already on).

With the 'stop' signal inactive, channel operations resume to end the operation because of the machine check condition which originally initiated the logout operations. The actual channel operations performed during the sequence 5 routine depend upon the condition of the channel at the time the machine check condition was detected.

When the sequence 5 operations are complete, the channel enters an interrupt routine. During this routine, the channel activates the 'interrupt request' signal to the CPU interface. Upon receipt of an 'interrupt response' signal, the channel stores the CSW at main storage address 64 (decimal) and sends the proper condition code and the unit address register contents to the CPU interface. Upon completion of the CSW store operation, the channel deactivates the 'interrupt request' signal and resumes polling operations; i.e., the channel is prepared to perform other I/O operations.

Detailed Logout Operations

Detailed logout operations are shown in flowchart form in Diagram 8-14. Diagram 8-15 shows the logout control logic in positive logic form. For simplicity, the routines performed during logout operation are described separately in the following paragraphs.

Logout on Machine Check

- LOG ON MACH CHK switch on, AUTO/TEST switch to AUTO, 'channel control check' or 'interface control check' detected; 'machine check' signal activated.
- 'Interface control check' signal turns on 'suppress out' latch and causes 6-usec 'interface reset' signal; attached I/O devices are reset.

Assume that: (1) the LOG ON MACH CHK switch is on (Diagram 8-3); (2) the 'start I/O' latch or 'test I/O' latch is on (Diagram 8-15); and (3) an interface control check or channel control check condition has been detected by the channel (Diagram 8-14).

An interface control check condition turns on the channel 'interface control check' latch. The active output of the latch then raises the channel 'machine check' signal and also triggers a 300-ns singleshot to produce an 'interface control check SS' signal. The 'interface control check SS' signal turns on the 'suppress out' latch. The resulting 'suppress out' signal is sent to the I/O interface to halt operations in buffered I/O devices, and prevents generation of the 'machine reset' signal within the channel. When the 'interface control check SS' signal falls, the channel generates a 6-usec 'interface reset' signal; this causes the 'operational out' signal to the I/O interface to fall and, thus, causes the attached control units to perform a reset operation.

A detected channel control check condition (Diagram 8-14) turns on the 'channel control check' latch. The active output of this latch activates the 'machine check' signal.

Stop Channel Operations, Activate Interrupt Request Signal

- If LOG ON MACH CHK switch is off, logout operation is not performed; normal sequence 5 ends operation due to 'machine check' signal.
- LOG ON MACH CHK switch on and AUTO/TEST to TEST; channel operations stop and no logout performed.
- LOG ON MACH CHK switch on and AUTO/TEST switch to AUTO; 'machine check' signal raises 'stop' signal (assuming to block-stop conditions are present).
- 'Stop' signal is on, 'log wait interrupt' latch turns on, 'interrupt request' signal activated; channel clock prevented from stepping, turning on or turning off.
- Initiation of channel operations other than logout are inhibited.
- 'Storage request' latch must be off before log operations can begin.

If the LOG ON MACH CHK switch is off, a logout operation is not performed. Instead, the channel enters a normal sequence 5 routine. In this case, an 'interrupt request' signal is sent to the CPU interface to store the CSW and present the proper condition code and unit address register contents to the CPU interface.

If the LOG ON MACH CHK switch is on and the AUTO/TEST switch is in TEST, the 'machine check' signal raises the 'stop' signal, halting channel operations. However, since the channel is in the test mode, the logout operation is not performed and the channel remains stopped until reset. If the AUTO/TEST switch is in AUTO and no block-stop

conditions are present in the channel, the channel 'stop' signal is activated. Block-stop conditions are those conditions which prevent channel operations from stopping upon detection of a machine check condition, and include: (1) FLT mode operation, (2) an active 'CPU interrupt response' signal in channel at the time the machine check condition is detected, (3) the channel has an interrupt pending ('interrupt' latch is on), or (4) a program-controlled interrupt is in the channel (PCI latch on).

With the 'stop' signal active, block-stop conditions which may be received by the channel after the 'stop' signal is activated are inhibited. In addition, the 'stop' signal performs the following functions: (1) inhibits generation of the 'interface reset' and 'machine reset' signals to prevent a change in channel status; (2) turns on the 'log wait interrupt' latch to activate the 'interrupt request' signal to the CPU interface; (3) prevents the channel clock from stopping, turning on, or turning off, depending upon clock status at the time of the machine-check condition was detected. (If the clock is stepping, the 'stop' signal may or may not prevent the clock from generating the next clock timing signal; this depends upon the time factor involved in activating the 'stop' signal as well as the time elapsed since generation of the current clock signal.); and (4) latches the 'service out' or 'command out' latch if either of these signals is active at the time the 'stop' signal is activated.

Turn-on of the 'log wait interrupt' latch (Diagram 8-15), in addition to activating the 'interrupt request' signal: (1) inhibits gating of the unit address register contents to the CPU interface 'unit address bus out' lines; (2) prevents initiaition of channel operations should a 'start I/O'; 'test I/O', or 'halt I/O' signal be received by the channels; and (3) prevents generation of a 'CPU release' signal to the CPU interface. The 'log wait interrupt' latch remains on until logout operation is complete.

If the channel is engaged in accessing storage when the 'stop' signal is activated (the 'storage request' latch is on), the logout operation cannot proceed until the storage cycle is complete (Diagram 8-14). If the 'storage request' latch is off, the 'stop gated' signal is activated and the channel begins the store log word 1 operation.

Store Log Word 1

- 'Stop gated' signal plus 'start I/O latch', 'test I/O latch', or 'interrupt response' signal turns on 'log 1' latch.
- 'Log 1' signal enables SAB P2, 18, and 19 gates, and gates log word 1 bit to SB1 gating logic.
- 'Interrupt storage request' latch turns on; 'storage request' signal to BCU interface rises.
- 'On log' trigger turns on:
  1. Inhibits 'Z address' latch turn-on.
  2. Inhibits 'storage data check' and 'storage address check' signals.
  3. Activates all mark-A register bits.
  4. Activates 'force storage protect parity' signal.
  5. Enables SAB P0 and 15 gates.
- Channel receives 'BCU response' signal; address 304, mark bits, storage protect key, and 'store' signal gated to BCU interface.
- Channel receives 'BCU data request' signal; log word 1 bits gated to SBI lines.
- Channel receives 'accept' signal; 'accept' latch turns on; 'interrupt storage request' latch turns off.
- Channel receives 'BCU advance pulse' signal.

The store log word 1 operations (Diagram 8-14) begin immediately after the 'stop gated' signal is activated if the 'test I/O' or 'start I/O' latches are on. If neither of these latches is on, the channel awaits receipt of the 'interrupt response' signal from the CPU interface before continuing the store log word 1 operation. When the 'start I/O', 'test I/O', or 'interrupt response' signal turns on the 'log 1' latch, the resulting 'log 1' signal enables the SAB P2, 18, and 19 gates; these enabled gates form a portion of address 304 (decimal) at which log word 1 is to be stored. The 'log 1' signal also gates the log word 1 contents to the SBI gating logic and triggers a 110-ns singleshot to produce a signal which turns on the 'interrupt storage request' latch. With this latch on, the 'storage request' signal to the BCU interface is activated. The 'log 1' signal also activates the 'log 1 or log 2' signal which turns on the 'log' trigger. The 'on log' output of the 'log' trigger performs the following functions: (1) inhibits turn on of the 'Z address' latch to prevent address 64 from being gated to the SAB; (2) inhibits the 'storage data check' and 'storage address check' signals (from the BCU interface) in the channel to prevent the channel from recognizing these conditions should they occur; (3) activates all mark-A register bits so that all bytes of the log words will be stored; (4) activates the 'force storage protect parity' signal so that a storage protect key of all zeros with correct parity can be gated to storage; and (5) enables the SAB P0 and 15 gates to form the remainder of address 304 (these gates remain enabled for all three log words to form a portion of addresses 304, 312, and 320.)

When the channel receives the 'BCU response' signal, enabled SAB gates P0, P2, 15, 18, and 19 are gated, and address 304 is placed on the BCU interface SAB lines. In addition, the mark-A register bits are gated to the mark lines, a storage protect key of all zeros with correct parity to the

storage protection lines and the 'store' signal is activated. When the channel receives the 'BCU data request' signal, the log word 1 bits are gated to the BCU interface SBI lines for placement in main storage.

The channel then awaits the 'accept' signal from the BCU interface, indicating that the storage cycle has began. The 'accept' signal turns on the 'accept' latch, and turns off the 'interrupt storage request' latch to deactivate the BCU interface 'storage request' signal. The channel subsequently receives the 'BCU advance pulse' signal, indicating that log word 1 has been stored. With the receipt of the 'BCU advance pulse' signal, the store log word 1 operation is complete.

Store Log Word 2

- 'Late advance pulse' signal turns on 'log 2' trigger.
- 'Log 1' trigger turns off.
- 'Log 2' signal gates log word 2 bits to SBI gating logic, enables SAB 20 gate, and turns on 'interrupt storage request' latch; 'storage request' signal rises.
- 'Log 1 or 2' signal enables SAB 18 and 19 gates.
- Channel receives 'BCU response' signal:
  1. Address 312 (decimal) gated to SAB.
  2. Mark bits, all-zeros storage protect key, and 'store' signal are gated to BCU interface.
- Channel receives 'BCU data request' signal; log word 2 bits gated to SBI lines.
- Remainder of operation identical to store log word 1 operation.

The store log word 2 operations begin when the 'late advance pulse' signal (delayed from the 'BCU advance pulse' signal) turns on the 'log 2' trigger and turns off the 'accept' latch. The 'log 2' signal turns off the 'log 1' trigger to degate the log word 1 bits from the SBI gating logic and disable the SAB P2 gate. The 'log 2' signal gates the log word 2 bits (Table 6-2) to the SBI gating logic, raises the 'log 1 or 2' signal to enable the SAB 18 and 19 gates, and enables the SAB 20 gate. The 'log 2' signal is also OR'ed to turn on the 'interrupt storage request' latch. As in store log word 1 operation, the channel receives a 'BCU response' signal. This signal gates address 312 (decimal) to the SAB lines, activates the 'address valid', and gates the mark bits, storage protect key, and 'store' signal to the BCU interface. Subsequently the 'BCU data request' signal gates the log word 2 bits to the SBI for storage in main storage. The remainder of the store log word 2 operation is as described for the store log word 1 operation.

Store Log Word 3

- 'Advance pulse' signal turns on 'log 3' trigger.
- 'Log 2' trigger turns off.
- 'Log 3' signal enables SAB 17 gate and gates log word 1 bits to SBI gating logic.
- 'Interrupt storage request' latch turns on; 'storage request' signal to BCU interface rises.
- Channel receives 'BCU response' signal:
  1. Address 320 (decimal) gated to SAB.
  2. Mark bits, all-zeros storage protect key, and 'store' signal gated to BCU interface.
- Channel receives 'BCU data request' signal; log word 2 bits gated to SBI lines.
- Remainder of operation identical to store log word 1 operation.

The store log word 3 operation (Diagram 8-14) begins when the 'advance pulse' signal (delayed from the 'BCU advance pulse' signal) turns on the 'log 3' trigger and turns off the 'accept' latch. The 'log 3' signal then turns off the 'log 2' trigger to disable the SAB 18, 19, and 20 gates and degate the log word 2 bits from the SBI gating logic. In addition, the 'log 3' signal: (1) is OR'ed to turn on the 'interrupt storage request' latch, thus raising the 'storage request' signal to the BCU interface; (2) gates the log word 3 bits (Table 6-2) to the SBI gating logic; and (3) enables the SAB 17 gate.

When the channel receives the 'BCU response' signal, address 320 decimal is gated to the SAB lines, the mark bits are gated to the mark lines, the storage protect key is gated to the storage protection lines, and the 'store' signal is gated to the store line.

Subsequently, the 'BCU data request' signal gates log word 3 to the SBI lines for storage in main storage. The remainder of the store log word 3 operation is as described for log word 1 operation.

End Logout

- 'Raw advance SS' signal turns off 'accept' latch and 'log 3' trigger.
- 'Storage cycle complete' signal turns on 'log stop' latch.
- 'Log' trigger turns off.

- 'Stop' signal drops.
- 'Sequence 5' latch turns on (if not already on).
- 'Interrupt status' signal activates.
- 'Interrupt' latch turns on.
- Channel resumes operations to end operation in progress when machine check condition is detected.
- Sequence 5 routine followed by interrupt routine to store CSW and send condition code, unit address, and 'release' signal to CPU interface.
- Channel polling; ready to perform new I/O operation.

The end-logout operations begin when the 'raw advance SS' signal (delayed from the BCU advance pulse' signal) turns off the 'accept' latch and the 'log 3' trigger. The channel then activates the 'storage cycle complete' signal (delayed from the 'BCU advance pulse' signal) to turn on the 'log stop' latch. The resulting 'degate stop' signal turns off the 'log' trigger, deactivating the SAB P0 and 15 gates, the mark-A register bits, and the 'force storage protect parity' signal. In addition, the 'degate stop' signal causes the 'stop' signal to deactivate, allowing the channel to resume operations to end the operation that was in progress when the machine check condition was detected. The 'degate stop' signal is also supplied to the 'sequence 5' latch to turn this latch on if the latch is off, and activates the 'interrupt status' signal.

The 'interrupt status' signal turns off the 'chain command' latch (if on) and inhibits the 'suppress out fast' and 'on jump command sequence 5' signal. In addition, the 'interrupt status' signal enables activation of the 'turn on interrupt end' signal which turns on the 'interrupt' latch. With the channel 'sequence 5' latch on, operations during the sequence 5 routine depend upon the operating condition of the channel at the time the machine check condition was detected. When the sequence 5 routine is complete, the channel enters the interrupt routine. Upon receipt of an 'interrupt response' signal from the CPU interface, the channel stores the CSW at main storage location 64, and sends the appropriate condition code to the CPU interface along with the unit address register contents. With the CSW stored, the channel resumes polling operations and is ready to perform further I/O operations.

CHANNEL DIAGNOSTICS

- CPU diagnose instructions control channel diagnostic operations.
- Diagnose instructions:
  1. Allow channel to store bytes with incorrect parity in main storage.
  2. Check the channel parity checking logic.
- MCW bits 0-2 (with 7 on) select specific channel for diagnostic operations ('diagnose select channel' signal).
- MCW bit 3 activates 'reverse byte count parity' signal.
- MCW bit 4 activates 'reverse byte count parity' signal.
- MCW bit 5 activates 'block storage data check' signal.

Channel diagnostic operations are controlled by the CPU diagnose instructions in those System/360 models having diagnostic capabilities. Diagnostic control signals from the CPU interface control the channel diagnostic operations. The channel diagnostic control signals perfrom two functions: (1) allow the channel to store bytes in main storage with incorrect parity, and (2) test the channel parity checking logic.

Figure 6-7 illustrates CPU diagnostic control of the channel. The CPU diagnose instruction performs dynamic tests on the selector channel by activating bits in the maintenance control word (MCW) register and by issuing start I/O instructions to perform specific channel operations. The MCW bits in the register can be activated or changed only by the diagnose instruction. Diagnostic lines to the channel are under direct control of the MCW. Once a diagnostic control line becomes active, the line remains active until another diagnose instruction is executed to change the MCW bits.

The diagnostic lines from the CPU to the channel (Figure 6-7), the MCW derivation of each, and the function performed by each in the channel are as follows. One of six simplex 'diagnose select channel' signals is activated to select a specific channel. The specific 'diagnose select channel' line activated is determined by the bit values of MCW bits 0–2; MCW bit 7 must be active for the CPU to decode MCW bits 0-2. In the channel, the 'diagnose select channel' line gates other diagnostic control signals into the channel, inhibits access to the control units on the I/O interface, and enables the simulate I/O test logic. With the simulate I/O test logic enabled, start I/O operations initiated by the diagnose instruction use the simulate interface register and simulate I/O test logic to perform the diagnostic operations.

The multiplex 'reverse data parity' signal is activated when MCW bit 3 is active. In the channel, the 'reverse data parity' signal reverses the parity bit from the interface control register during read operations and inhibits the bus-in parity checking logic.

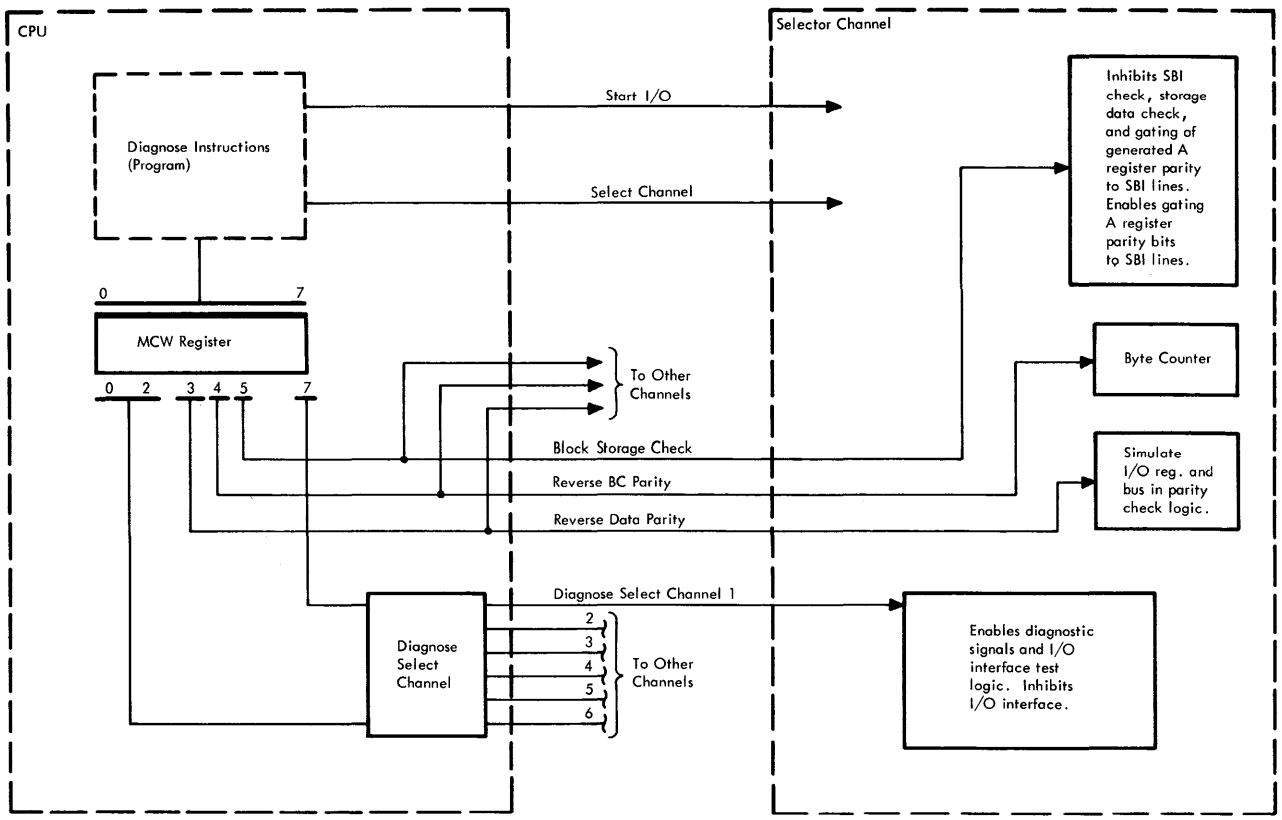


Figure 6-7. Channel/CPU Diagnostic Control

The 'reverse byte count parity' signal is active when MCW bit 4 is active. In the channel, the 'reverse byte count parity' signal reverses the byte-count parity bit during byte count update operations. When update is complete, the byte counter contains even (incorrect) parity. MCW bits 3 and 5 must be off for the 'reverse byte count' signal to perform the above functions.

The 'block storage data check' signal is active when MCW bit 5 is active. In the channel, the 'block storage data check' signal inhibits the 'SBI parity check' logic and the 'storage data check' signal from main storage. This permits the channel to store bytes from the A-register with incorrect parity into main storage during read operations.

If the 'reverse data parity', 'block storage data check' and 'reverse byte count parity' signals are active simultaneously, channel adder parity checks are inhibited during a command address, data address, or count update operation, and the byte count parity is not affected by the 'reverse byte count parity' line. Usually, this situation is used during the diagnose instruction when a CCW is fetched by the channel with bad parity in the data address field; the bad parity data address is then updated and used to access main storage. In this case, the channel should receive a 'storage address check' signal which turns on the 'channel control check' latch to indicate storage detection of the forced error.

Diagnose instruction operations and the results of the operation are printed out for use by the CE as a maintenance aid in determining whether the expected results of the operation were obtained. If the results are not as expected, the printout should aid the CE in isolating malfunctions to specific areas of the channel, CPU, or main storage. Channel tests performed by the diagnose instruction are described in the following paragraphs. Diagram 8-15, FEMDM is referenced throughout the discussion.

Enable Diagnose Operations

- 'Diagnose select channel' signal enables channel to perform diagnostic operations.
- 'Diagnose simulate interface' latch turns on to enable simulate I/O test logic and disable I/O interface.
- 'Diagnose select channel' signal is deactivated when diagnostic operations are complete.

Channel diagnostic operations are enabled by an active 'diagnose select channel' signal (Diagram 8-15) from the CPU interface. This signal is AND'ed with the 'not simulate CPU' level (channel is on-line to the CPU). The 'diagnose select channel' AND output enables the AND's receiving the 'block storage check', 'reverse data parity', and 'reverse byte count parity' lines. In addition, the AND'ed 'diagnose select channel' signal is further AND'ed with the 'setup' and 'not operational in' or 'status in' signals to turn on the 'diagnose simulate interface' latch. The 'diagnose simulate interface' signal is OR'ed to activate the 'simulate interface' signal. The latter activates the simulate I/O test logic and inhibits I/O interface input and output signals. This permits the channel to use the simulate interface register to perform I/O operations without access to the attached control units. For a description of the simulate interface register and associated test logic operations, refer to Section 1 of this chapter. When the diagnostic operations are complete, the 'diagnose select channel' line is dropped, and the 'diagnose simulate interface' latch is turned off.

Channel Diagnostic Operations

Operations performed by the channel under diagnostic control are described separately below. For details of the diagnose instruction implementation and diagnose instruction variations not described in the following paragraphs, refer to the appropriate programming document for the System/360 Model in question.

Store Incorrect Parity Bytes

- 'Start I/O' instruction specifies write operation.
- Correct parity bytes transferred to simulate interface register.
- 'Block storage data check' and 'reverse data parity' signals activated.
- Start I/O instruction specifies write operation.
- Parity from simulate interface register is reversed; bus-in parity check disabled.
- SBI parity checks disabled; A-register bits gated to SBI; 'storage data check' signal blocked in channel.
- Incorrect parity bytes are stored in storage.

Incorrect parity bytes are loaded into main storage by the channel under diagnostic control to test error-detection capabilities of main storage and the CPU. Under diagnostic control, the channel is instructed by a Start I/O instruction to perform a write operation. During the write operation, correct parity bytes are transferred into the simulate interface register, and the last byte written is retained by the register. The 'block storage data check' and 'reverse data parity' signals are then activated (Diagram 8-16), and the channel is instructed (via a Start I/O instruction) to perform a read operation.

The 'reverse data parity' signal is applied to the output logic associated with the simulate interface register and reverses the parity bit from the register each time a byte is transferred to the bus-in lines. To prevent the channel bus-in parity checking logic from detecting incorrect parity during the transfer, the 'reverse data parity' signal inhibits the bus-in parity checking logic. During the read operations, bytes are assembled in the B-register and transferred to the A-register as during a normal read operation.

The 'block storage data check' signal performs three principal functions: (1) inhibits channel recognition of an SBI parity check condition which would normally be detected to turn on the 'channel data check' latch when incorrect parity bytes are gated from the A-register to the SBI logic; (2) enables gating of the A-register parity bits to the SBI lines for storage, rather than the parity bits generated by the SBI parity generating logic; and (3) prevents the 'storage data check' signal, generated when main storage detects the incorrect parity bytes, from setting the 'channel data check' latch. With the channel logic conditioned by the 'block storage data check' and 'reverse data parity' signals as described, the channel can store incorrect parity bytes without interrupting channel operations because of a detected parity check error.

Storage Protection Parity Check Test

- 'Block storage data check' signal blocks 'storage data request' signal in channel.
- Start I/O instruction causes CAW fetch with incorrect parity in storage protect key.
- Key bits gated to storage protect register; register parity check logic detects parity error.
- 'Channel control check' latch is set.

The channel storage protect register parity-checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives an active 'block storage data check' signal and a Start I/O instruction. The CAW fetched by the channel for the start I/O operation contains a storage protect key with incorrect parity.

Since main storage will activate the 'storage data check' line when the incorrect parity CAW is fetched, the 'block storage data check' signal (Diagram 8-15) prevents the 'storage data check' signal from turning on the 'channel control check' latch. When the storage protect key is gated into the storage protect register, the register storage protection parity check logic detects the incorrect parity and turns on the 'channel control check' latch. This causes the channel to enter a sequence 5 routine followed by an interrupt routine to store the CSW. The 'channel control check' bit should be on in the CSW.

Flag Register Parity Check Test

- 'Block storage data check' signal blocks 'storage data check' signal in channel.
- Start I/O instruction causes CCW fetch with incorrect parity in flag field.
- Flag bits gated to flag registers; when sampled, register parity check logic detects incorrect parity.
- 'Channel control check' latch is turned on.

The channel flag register parity-checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives an active 'block storage data check' signal and a Start I/O instruction. The CCW fetched by the channel for the start I/O operation contains a flag field with incorrect parity. Since main storage will activate the 'storage data check' signal when the incorrect parity CCW is fetched, the 'block storage data check' signal (Diagram 8-15) prevents the 'storage data check' signal from turning on the 'storage data check' latch. When the flag bits are gated into the flag register and later sampled, the incorrect parity is detected, and the 'channel control check' latch is turned on. The channel enters a sequence 5 routine followed by an interrupt routine to store the CSW.

Adder Parity Check Test

- 'Block storage data check' signal blocks 'storage data check' signal at channel.
- Start I/O instruction causes CCW fetch with incorrect parity in count or data address field.
- When count or data address is updated, adder parity checking logic detects parity error.
- 'Channel control check' latch is turned on.
- Three Start I/O instructions are necessary to examine each eight-bit adder group separately.

The channel adder parity-checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives an active 'block storage data check' signal and a Start I/O instruction. The CCW fetched by the channel for the start I/O operation contains incorrect parity in the count or data address field. Since main storage will activate the 'storage data check' signal when the incorrect parity CCW is fetched, the 'block storage data check' signal (Diagram 8-15) prevents the 'storage data check' signal from turning on the 'channel control check' latch. When the count or data address is updated by the adder, the adder parity checking circuits detect the incorrect parity and turn on the 'channel control check' latch. The channel then enters a sequence 5 routine followed by an interrupt routine to store the CSW with the 'channel control check' bit on in the channel status byte. Since the adder parity-check logic examines three 8-bit groups individually, three separate Start I/O instructions are required if each byte is to be tested individually.

Bus-Out Parity Check Test

- 'Block storage data check' signal blocks 'storage data check' signal in channel.
- Start I/O instruction causes write operation with incorrect parity in bytes.
- Bus-out parity checking logic detects incorrect parity as bytes are transferred through bus-out latches.
- 'Channel data check' latch turned on.

The channel bus-out parity checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives an active 'block storage data check' signal and a Start I/O instruction specifying a write operation. At least the last data byte of the write operation must contain incorrect parity. Since main storage will activiate the 'storage data check' signal when incorrect data parity is detected, the 'block storage data check' signal (Diagram 8-15) prevents the 'storage data check' signal from turning on the 'channel data check' latch. During the write operation, the bus-out parity checking logic should detect the incorrect parity and turn on the 'channel data check' latch. Upon completion of the write operation, the channel enters a sequence 5 routine, followed by an interrupt routine to store the CSW with the 'channel data check' bit on. The simulate interface register should also contain a byte with incorrect parity.

Bus-In Parity Check Test

- 'Block storage data check' signal blocks 'storage data check' signal and 'SBI parity check' signal at channel.
- Start I/O instruction causes write operation with bad parity bytes; last bad parity byte retained by simulate interface register.
- Start I/O instruction causes read operation; bus-in parity checking logic should detect incorrect parity in byte from simulate interface register.
- 'Channel data check' latch is turned on.

The channel bus-in parity checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives an active 'block storage data check' signal and a Start I/O instruction specifying a write operation. Data bytes received from main storage must have incorrect parity. Since main storage will activate the 'storage data check' signal when incorrect data parity is detected, the 'block storage data check' signal (Diagram 8-15) prevents the 'storage data check' signal from turning on the 'channel data check' latch. Since the write operation is followed by a Start I/O instruction specifying a read operation, the 'block storage data check' signal also prevents the 'SBI parity check' signal from turning on the 'channel data check' latch. During the read operation, the incorrect parity byte in the simulate interface register is transferred through the bus-in latches; the bus-in parity checking logic should detect the incorrect parity and turn on the 'channel data check' latch. At the end of the read operation, the channel enters the sequence 5 routine followed by an interrupt routine to store the CSW with the 'channel data check' bit on.

Storage Bus-In Parity Check Test

- Start I/O instruction causes write operation with correct parity bytes; last byte retained by simulate interface register.
- 'Reverse data parity' signal reverses parity bit in byte from simulate interface register and inhibits bus-in parity checking logic.
- Start I/O instruction causes read operation.
- SBI parity-checking logic detects parity error as byte is stored.
- The eight SBI parity checking circuits can be individually tested by eight Start I/O instructions specifying read operation, specifying count of 1 byte, and varying three low-order data address register bits (DAB) to specify a different byte each time.
- 'Channel data check' latch turns on for each of the eight read operations.

The channel storage bus-in (SBI) parity checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives a Start I/O instruction specifying a write operation. Data fetched from storage must have correct parity, and the last byte written is stored in the simulate interface register. The channel then receives an active 'reverse data parity' signal (Diagram 8-15) and a Start I/O instruction, causing a CCW to be fetched which specifies a read operation and a count field of one byte. During the read operation, the 'reverse data parity' signal reverses the parity bit of the byte transferred to the B-register and inhibits the bus-in parity checking logic. When the byte is transferred to the A-register and gated to the SBI, the SBI parity-checking logic detects the incorrect parity and generates an SBI parity check signal to turn on the 'channel data check' latch. The channel then enters a sequence 5 routine followed by an interrupt rotuine to store the CSW with the 'channel data check' bit on. Since there are eight SBI bytes, the parity-checking logic for each byte can be tested by repeating the described start I/O read operation eight times; this is accomplished by specifying a count of one and varying the three low-order bits (DAB) of the data address register to specify individual bytes 0–7. For each start I/O operation, only one byte with incorrect parity is tested; further, a different byte is tested for each operation.

Storage Address Check Test

- 'Reverse data parity', 'block storage data check', and 'reverse byte count parity' signals inhibit adder parity checks.
- 'Start I/O' signal causes CCW fetch with incorrect parity data address.
- Channel updates data address; incorrect parity maintained.
- Data address register contents gated to SAB on storage access.

- Storage detects address parity error; sends channel 'storage address check' signal.

- 'Channel control check' latch is turned on.

The diagnose instruction may be used to test the address parity checking circuits in main storage. To perform this test, the channel receives the 'reverse data parity', 'block storage data check', and 'reverse byte count parity' signals. In addition, the channel receives a Start I/O instruction, causing the channel to fetch a CCW with incorrect parity in the data address field. The three active diagnostic signals (Diagram 8-15) are AND'ed to inhibit the 'adder parity check' signal; this allows the channel to update and gate the data address to main storage without the 'channel control check' latch being turned on by a detected adder parity check condition. (With the 'adder parity check' signal inhibited, the 'reverse byte count parity' signal is inhibited from reversing the byte count parity bit.) When address parity checking logic in main storage detects the incorrect parity, the 'storage address check' signal to the channel is activated and the 'channel control check' latch is turned on. The channel then enters a sequence 5 routine, followed by an interrupt routine to store the CSW with the 'channel control check' bit on.

Byte Count Parity Check Test

- Start I/O instruction causes CCW fetch specifying read or write operation.

- 'Reverse byte count parity' signal causes incorrect byte count parity to be generated.

- Byte count parity-checking logic detects parity error.

- 'Channel control check' latch is turned on.

The channel byte count parity-checking logic can be tested under diagnostic instruction control. To perform the test, the channel receives a Start I/O instruction, causing the channel to fetch a properly formatted CCW specifying a read or write operation. The channel then receives an active 'reverse byte correct parity' signal (Diagram 8-15), which causes even (incorrect) byte counter parity to be generated during the change-byte-count operation. The byte count parity-check logic should detect the parity error and turn on the 'channel control check' latch. The channel then enters a sequence 5 routine, followed by an interrupt routine to store the CSW with the 'channel control check' bit on.

ERROR-CHECKING CONDITIONS

- Seven check conditions detected by channel:
  1. channel control check
  2. interface control check
  3. channel data check
  4. storage protect check
  5. chain check
  6. program check
  7. unit check

- For initial selection routine, channel control check, interface control check, program check, and unit check conditions can be detected:
  1. LOG ON MACH CHK switch on, auto mode, and channel control check or interface control check condition cause logout, then CSW store, CC code, unit address, and 'release' signal to CPU interface.
  2. LOG ON MACH CHK switch off, auto mode, and any of the four check conditions cause CSW store, and CC1 and 'release' signal to CPU interface.

- For channel operation, any of the seven check conditions can be detected:
  1. LOG ON MACH CHK switch on, auto mode, and channel control check cause 'interrupt request' signal to activate, followed by logout operations, then CSW store and CC, unit address, and 'release' signal to CPU interface.
  2. LOG ON MACH CHK switch off, auto mode, and any of the seven check conditions (except data check) activate 'interrupt request' signal to CPU interface; when 'interrupt response' signal is received, CSW is stored, and 'release' signal is sent to CPU interface.

- Data check condition allows read or write operation to complete, but breaks command chaining if indicated.

Channel check conditions identify the operational area in which an error is detected during a channel operation. In most cases, the detected check condition immediately terminates the operation in progress. The channel detects seven different check conditions: (1) channel control check, which in general is activated when the channel detects an error within the channels control logic; (2) interface control check, which is activated when an invalid I/O interface signal or sequence is detected; (3) channel data check, which is activated as a result of a data parity error during channel controlled data transfers; (4) storage protect check, which is activated by a storage protect key mismatch in main storage (channel may have sent an improper storage

protect key to storage for the storage location being accessed); (5) chain check, which is activated during read or read chain data operations when the I/O device presents data faster than the channel can handle it; (6) program check, which is activated when the channel detects a programming error during channel operations; and (7) unit check, which originates in the I/O device or control unit and is presented to the channel in the control unit status byte.

Channel operations upon detection of a check condition are shown in simplified form in Figure 6-8. Operations performed depend upon whether the channel is performing an initial selection routine or has completed the routine and is engaged in an operation. If the channel is performing an initial selection routine, only four check conditions are possible: channel control check, interface control check, program control check, and unit check. If a channel control check or interface control check condition is detected with the LOG ON MACH CHK switch on (channel in auto-mode operation), channel operations are stopped to reflect the channel conditions at the time the check condition was detected. Since the CPU has not yet been released by the channel, the channel immediately enters a logout operation to store three log words in main storage. (See "Logout Operation.") Upon completion of the logout operation, the channel stores the CSW at storage location 64 and sends a condition code, unit address, and 'release' signal to the CPU.

Assume the LOG ON MACH CHK switch is off, and a channel control check, interface control check, program check, or unit check condition is detected during the initial selection routine. Since the channel has not yet released the CPU, the channel immediately enters a sequence to store the two CSW bytes at main storage location 64. Following the store CSW sequence, the channel sends condition code 1 to the CPU interface, followed by a 'release' signal.

If the channel has completed the initial selection routine and is engaged in a channel operation, any of the seven check conditions (Figure 6-8) can occur. If a channel control check or interface control check condition is detected with the LOG ON MACH CHK switch on (channel in auto-mode operation), channel operations are stopped, an 'interrupt request' signal is sent to the CPU interface, and the channel performs a logout operation. (See "Logout Operation".) Following the logout operation, the channel stores the CSW at address location 64, and sends the appropriate condition code, unit address, and 'release' signal to the CPU interface.

Assume that the LOG ON MACH CHK switch is off, and the channel is in auto mode and performing an operation. Excepting the data check condition, detection of any check condition causes the channel to immediately end the operation in progress. The channel ends the operation by entering a sequence 5 routine, activating the 'interrupt request' signal to the CPU interface, and waiting for an 'interrupt response' signal from the CPU interface. If a data check condition is detected during a read or write operation, the operation continues to completion, and the data check bit is active when the CSW is stored during normal ending operations. If a chain command operation is indicated when the data check condition is detected, command-chaining operations are terminated at the end of the read or write operation in which the data check condition was detected.

Each of the check conditions is described separately in the following paragraphs, with appropriate reference to FEMDM diagrams.

Channel Control Check

- Turn-on of 'channel control check' latch activates channel 'machine check' signal.

- Logout operation performed if LOG ON MACH CHK switch is on and channel is in auto mode.

- 'Channel control check' latch turned on by:
  1. Flag register parity check
  2. Storage protect register parity check
  3. Byte count parity check
  4. Bus-out parity check on command or address byte
  5. Adder parity check
  6. Storage address check
  7. Storage data check on a CAW or CCW fetch

- 'Machine check' signal prevents turn-off of channel clock (if on), prevents turn-on of 'sequence 1' latch, resets 'IPL sync' latch (if on), and activates 'command reject or control check' signal.

- 'Command reject or control check' signal inhibits and enables channel signals to allow channel to perform logout operation (if performed) and terminate current channel operation.

- 'Channel control check' signal applied to SBI gate bit 45 for log word 1 storage (if performed) and CSW storage.

- When CSW is stored during initial selection routine, condition code 1, unit address, and 'release' signal are sent to CPU interface; otherwise only 'release' signal is sent.

The channel control check condition is detected as a result of channel control parity errors (Diagram 2-1). Detection of the check condition turns on the 'channel control check' latch; in turn, the 'channel control check' signal raises the channel 'machine check' signal. Together, the 'machine check' and 'channel control check' signals

cause the channel to terminate the operation in progress. Depending upon the positions of the LOG ON MACH CHK and the AUTO/TEST switches, the channel may or may not perform a logout operation before performing the normal sequence-5 ending routine.

Error conditions which turn on the 'channel control check' latch are: (1) a flag register parity check; (2) a storage protect register parity check; (3) a byte count parity check; (4) a bus-out parity check with an address or command byte on the bus-out lines; (5) an adder parity check; (6) a storage address check detected by storage as a result of bad parity in the address sent by the channel; and (7) a storage data check detected by storage during a CAW or CCW fetch operation. Diagram 2-1 illustrates the logic and signals which generate each of the seven conditions which will turn on the 'channel control check' latch. Note that detection of a storage address check or storage data check condition also turns on the 'storage check' latch. (Refer to "Channel Data Check.")

Assume that the 'channel control check' latch is turned on and the channel is in auto mode. The 'machine check' signals enables the stop logic associated with the LOG ON MACH CHK switch. If the switch is on, and no block-stop conditions are present in the channel, the channel performs a logout operation. (Refer to "Logout Operation.") If the switch is off, no logout operation is performed. In addition, the 'machine check' signal: (1) inhibits the 'read or write and status in' signal to prevent turn-off of the channel clock and turn-on of the 'sequence 1' latch; (2) resets the 'IPL sync' trigger if on during an IPL operation; and (3) activates the 'command reject or control check' signal. The 'command reject or control check' signal inhibits and activates channel signals associated with the termination of the operation in progress when the 'channel control check' signal was detected. For details of the functions performed by the 'command reject or control check' signal, refer to Diagram 2-1. Exact signals enabled or disabled depend upon the point in an operation at which the 'channel control check' signal is detected. In general, the 'command reject or control check' signal permits the channel to end the operation in progress by entering a sequence 5 routine followed by an interrupt routine to store the CSW in main storage and send a 'release' signal to the CPU interface. If the check condition is detected during the initial selection routine, condition code 1 and the unit address are also sent to the CPU interface.

The output of the 'channel control check' latch also activates the 'command reject or control check' signal. (This is redundant to activation of the signal by the 'machine check' signal.) In addition, the 'channel control check' signal is applied to SBI gate bit 45 for storage in main storage location 304 during store log word 1 operations (if performed) or main storage location 64 during the CSW store operation. The 'channel control check' signal also inhibits generation of the 'gate SBO to A-register write' signal to prevent gating of a data doubleword, if previously requested, from storage into the A-register.

Interface Control Check

- Turn-on of 'interface control check' latch activates 'machine check' signal.
- Logout operation performed if LOG ON MACH CHK switch is on and channel is in auto mode.
- 'Interface control check' latch turned on by:
  1. Incorrect selection (wrong I/O device selected).
  2. No response (device not reselected during chain command operation).
  3. Bus-in control parity check (parity error in address or status byte on bus-in lines).
  4. 'Release' signal not generated within 100.05 ms after receipt of 'CPU select channel' signal.
  5. 'Operational in' signal from control unit drops before channel drops 'select out' signal. (Control unit becomes unavailable while operating with channel.)
- 'Machine check' signal performs same functions as when 'channel control check' signal is detected.
- 'Interface control check' signal applied to SBI gate bit 46 for log word 1 storage (if performed) and CSW storage.

The interface control check condition is detected when an error is detected during a sequence or operation involving the I/O interface (Diagram 2-1, FEDM). Detection of the check condition turns on the 'interface control check' latch; in turn, the 'interface control check' signal raises the channel 'machine check' signal. Together, the 'machine check' and 'interface control check' signals cause the channel to terminate the operation in progress. Depending upon the positions of the LOG ON MACH CHK and AUTO/TEST switches, the channel may or may not perform a logout operation before performing the normal sequence 5 ending routine.

Error conditions which turn on the 'interface control check' latch are as follows (Diagram 2-1):

1. An incorrect selection condition which occurs during an initial selection routine or chain command operation when the unit address byte received from the I/O interface does not match the unit address register contents.
2. A no-response condition which occurs during chain command operations when the attempt to reselect the I/O device is unsuccessful ('select in' signal activates).

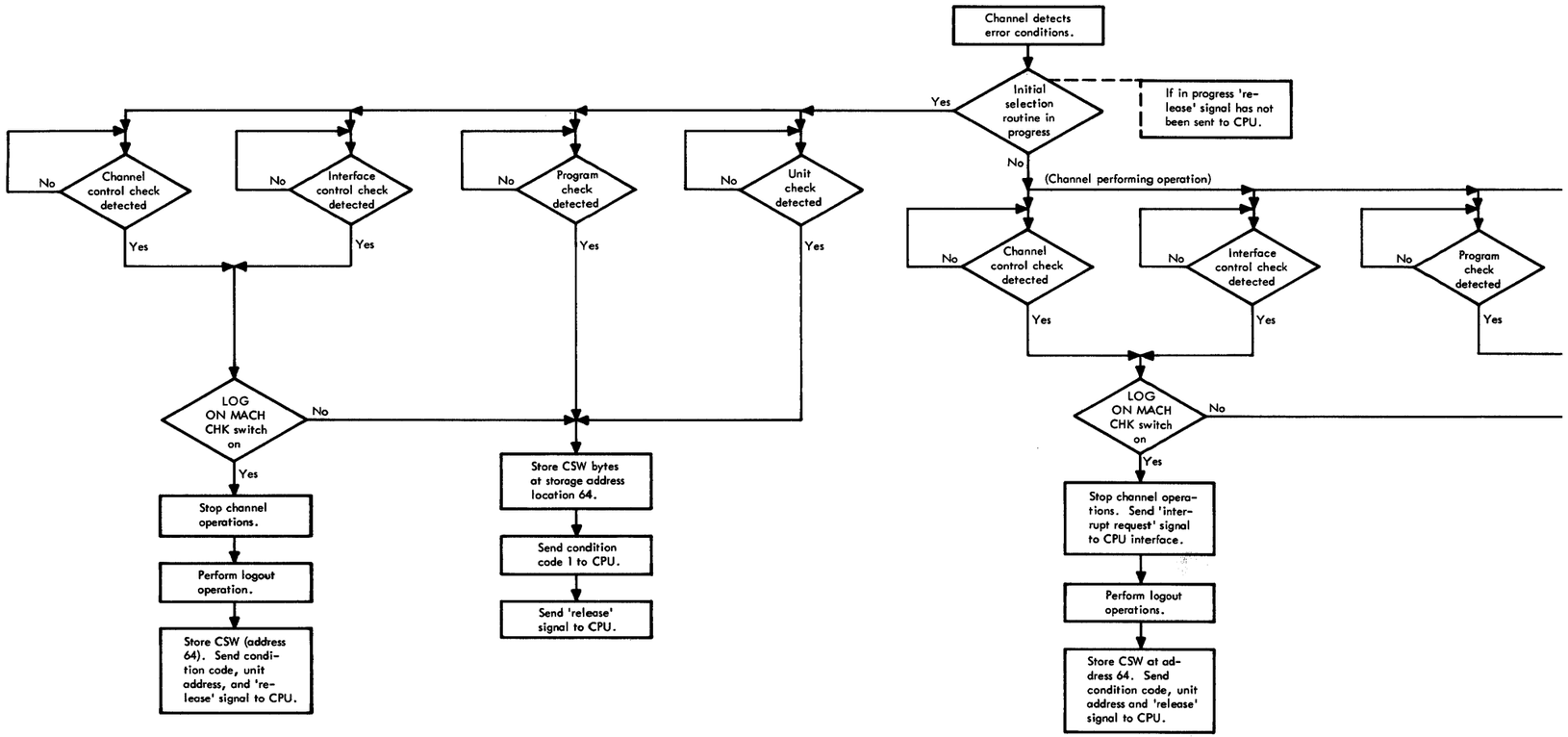
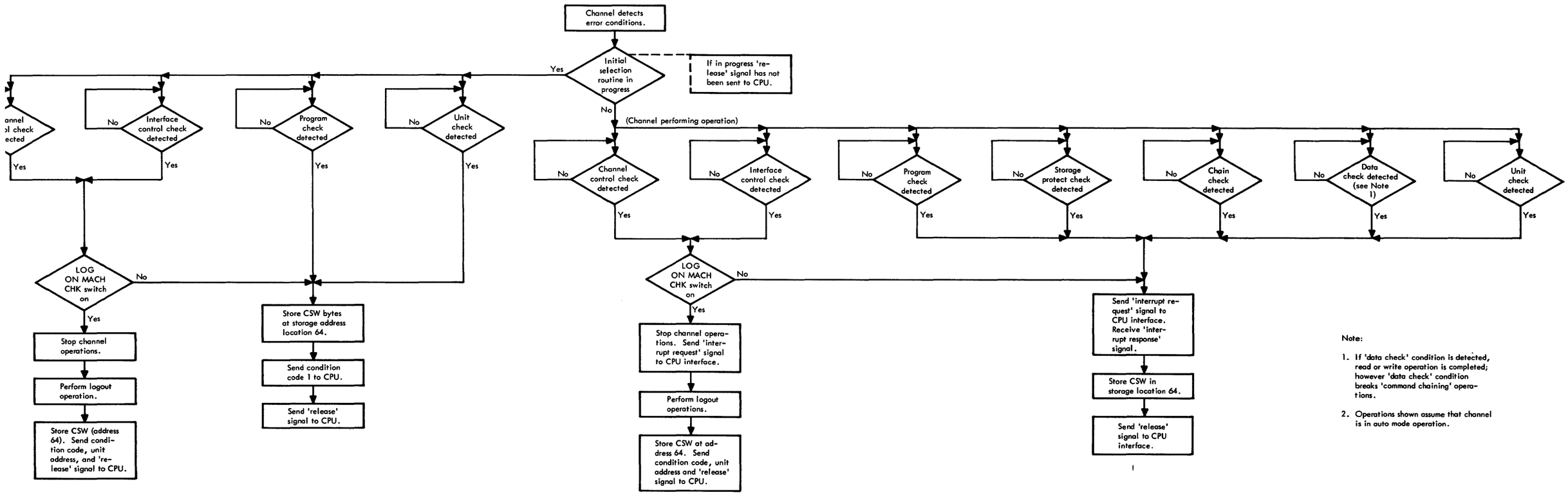


Figure 6-8. Channel Error Checking Operations



Note:

1. If 'data check' condition is detected, read or write operation is completed; however 'data check' condition breaks 'command chaining' operations.
2. Operations shown assume that channel is in auto mode operation.

annel Error Checking Operations

3. During an initial selection sequence, when the channel does not send a 'release' signal to the CPU interface within 100.05 ms after receipt of the 'CPU select channel' signal. (Time delay times out; this check condition is blocked for IPL and FLT operations.)
4. The channel detects a bus-in parity check with an address or status byte on the bus-in lines.
5. The 'operational in' signal from the control unit drops before the channel turns off the 'select out' latch.

In condition 5 above, the control unit becomes unavailable for operation while performing an operation with the channel; thus, the 'interface control check' latch is turned on. Note that (Diagram 2-1), when the 'operational in' and 'select out' signals are active, the 'remember operational in' latch is turned on. Should the 'operational in' signal drop, the 'remember operational in' latch remains on and inhibits the 'select in' logic within the channel. This prevents the 'select in' signal, presented to the channel as a result of the 'operational in' signal dropping, from turning off the 'select out' latch. This allows the channel time to set the 'interface control check' latch. The 'interface control check' signal activates the 'machine check' signal, which in turn, performs the functions described in "Channel Control Check" to terminate the operation in progress. In addition, the 'interface control check' signal: (1) turns off the 'select out' latch, if on; (2) inhibits activation of the 'turn-on select out setup' signal to prevent turn-on of the 'select out' latch; (3) enables turn-off of the 'service out end' latch, if on; and (4) is presented to SBI gate bit 46 for storage at address location 304 during a logout operation (if performed) or for storage at address location 64 during the CSW store operation.

Channel Data Check

- Channel data check condition allows read or write operation to complete, but stops chain command operation if 'chain command' flag is active.

- In test mode with LOG ON MACH CHK switch on, channel data check condition stops channel operations; no logout operation is performed.

- 'Channel data check' latch turned on by:
  1. SBI parity check when bad parity data is stored.
  2. Bus-in data check when bad parity data byte is on bus-in lines.
  3. Bus-out data check when bad parity data byte is on bus-out lines.
  4. Storage data check when storage detects bad parity in data stored or fetched by channel.

- 'Storage check' latch turned on by 'storage data check' or storage address check' signals: latch output is available at SBI gate 45 for storage in log word 3 of logout operation.

- 'Channel data check' signal prevents turn-on of 'chain command' latch and enables turn-on of 'interrupt end' latch.

- 'Channel data check' signal applied to SBI channel status gate bit 44 for CSW store.

A channel data check condition is detected as a result of a data parity error detected while the channel is transferring data (Diagram 2-2). Detection of a data check condition turns on the 'channel data check' latch. During auto-mode operations, turn-on of the 'channel data check' latch does not interfere with the read or write operation in progress. However, if a chain command operation is indicated following the read or write operation, the 'channel data check' signal inhibits the chain command operation. Thus, a channel data check condition allows the read or write operation (including chain data operations) to continue to completion; at the end of the operation, the 'channel data check' bit is stored in the CSW. During test-mode operations, detection of a channel data check condition stops channel operations when the condition is detected if the LOG ON MACH CHK switch is on. (Refer to "Log on Machine Check Switch and Stop Control.") A logout operation is not performed and the channel remains stopped until reset. If the LOG ON MACH CHK switch is off, a detected channel data check condition affects the channel in the manner described for auto-mode operations.

Check conditions which turn on the 'channel data check' latch (Diagram 2-2) are: (1) an SBI parity check condition detected when A-register data is gated to the SBI lines during a read operation; (2) a bus-in data check condition when a data byte with bad parity is on the bus-in lines; (3) a bus-out data check condition detected when a data byte with bad parity is on the bus-in lines; and (4) a storage data check condition detected in main storage when data stored or fetched by the channel has a parity error. The storage data check condition, in addition to turning on the 'channel data check' latch, also turns on the 'storage check' latch (Diagram 2-2). This latch is also turned on by the 'storage address or storage data check' signal that turns on the 'channel control check' latch. The output of the 'storage check' latch is applied to SBI gate bit 45 for storage in storage location 320 (decimal) during a store log word 3 sequence of a logout operation.

When the 'channel data check' latch is turned on, the resulting latch output performs the following functions.

If the channel is in test mode and the LOG ON MACH CHK switch is on, the 'channel data check' signal enables activation of the 'stop' signal. With the 'stop' signal active, channel operations are stopped to reflect the condition of the channel at the time the channel data check condition was detected. If the LOG ON MACH CHK switch is off, or the channel is in the auto mode, the 'storage data check' signal will not activate the 'stop' signal.

The 'channel data check' signal also activates the 'interrupt status' signal to prevent turn-on of the 'chain command' latch if the chain command flag in the flag register is active. This prevents a chain command operation from being performed, and allows the read or write operation, during which the channel data check condition was detected, to be terminated in the normal manner. Activation of the 'turn on interrupt end' signal when the 'sequence 5' latch is turned on is also enabled by the 'channel data check' signal. For the CSW store operation performed when the read or write operation ends, the 'channel data check' signal is applied to SBI channel status gate bit 44.

Storage Protect Check

- Storage protect check detected in main storage when storage protect key from channel does not match key in main storage.
- Key mismatch causes storage to send 'protect address check' signal to channel.
- Channel 'storage protect check' latch turns on.
- 'Command reject or control check' signal is activated to terminate operation and cause CSW to be stored.
- 'Storage protect check' signal is applied to SBI channel status gate bit 43 for storage in CSW.

The storage protect check condition is detected when the channel accesses main storage and the storage protect key sent by the channel does not match the accessed storage protect key in main storage. (This check condition is possible only when the channel is accessing a storage unit which has the storage protection feature installed.) The key in main storage is accessed from storage by specified bits of the address on the SAB. If the two keys do not match, the channel has attempted to access a protected area of storage, and main storage activates the 'protect address check' signal. (If the channel sends a master storage protect key of all zeros with correct parity, no key matching is performed and the 'protect address check' signal remains inactive.)

When the channel receives a 'protect address check' signal (Diagram 2-3), the 'raw advance pulse' signal (delayed from the 'BCU advance pulse' signal) AND's to turn on the 'storage protect check' latch. The latch output is OR'ed to activate the 'command reject' signal. In turn, the 'command reject' signal activates the 'command reject or control check' signal (Diagram 2-1). This signal activates and inhibits specific channel logic, causing the channel to enter a sequence 5 routine followed by an interrupt routine to store the CSW and send a 'release' signal to the CPU. The 'storage protect check' signal (Diagram 2-3) is applied to SBI channel status gate bit 43 for storage as part of the CSW.

Chain Check

- 'Chain check' latch is turned on when control unit supplies channel with data during read or read CDA operations faster than channel can handle it.
- 'Chain check' condition during read operation results when BC-equals-CTB condition cannot be detected on last word data assembly.
- During read CDA operations, chain check condition results when channel is fetching new CCW and:
  1. Channel has double-gated at least one doubleword and DAB in new CCW specifies byte boundaries.
  2. B-register contains more bytes than count in new CCW specifies (count less than eight bytes; BC greater than CTB).
  3. A-register contains eight bytes, and count in new CCW is less than eight.
  4. A-register contains eight bytes; B-register contains more bytes than count in new CCW specifies (count less than 16 bytes; BC greater than CTB).
  5. A- and B-registers are full and new CCW is not established in channel.
  6. A-register contains eight bytes, and DAB in new CCW is not on doubleword boundaries.
  7. A-register empty; B-register contains more than three and less than eight bytes; DAB in new CCW is not on doubleword boundaries.
- Turn-on of 'chain check' latch terminates operation; causes sequence 5 routine and CSW store.
- 'Chain check' signal stored in CSW via SBI gate bit 47.

A chain check condition is detected during a read or read chain data (CDA) operation when the control unit supplies data to the channel faster than the channel can handle it. Detection of a chain check condition results in termination of the operation in progress and storage of the CSW with the 'chain check' bit on. Any detected chain check condition results in turn-on of the 'chain check' latch (Diagram 2-4).

For the read operation (not CDA), a chain check condition results when the 'last word' trigger is on, the byte count latches equal zero, and count register bits 21 through 23 do not equal zero. This means that the last doubleword is being gated into the B-register, that the last byte position of the B-register contains a data byte, and that the CTB in the count register is greater than the byte count. (Recall that the BC-equals-CTB condition is required to determine the last B-register byte position of the read operation.) Since the B-register is not capable of accepting further data bytes and the 'last word' signal prevents the channel from assembling another doubleword, the BC-equals-CTB condition cannot be detected. Thus a chain check condition exists, and the 'chain check' latch is turned on to indicate that the channel cannot handle the data from the I/O device.

For read CDA operations, some conditions can occur which cause a chain check condition. Recall that, for a portion of a read CDA operation, the channel may be receiving data from a control unit without control by a CCW; during this portion of the operation, the channel is also fetching a CCW which will specify control of any data bytes received by the channel. For the following description of read CDA chain check conditions, refer to Diagram 2-4 unless otherwise specified.

Assume that the 'byte count register equals zero' trigger is on. This means that the B-register and the three low-order bits in the data address field of the new CCW must specify either singleword or doubleword boundaries ('read DAB 4 or read backward DAB 3' or 'read DAB 0 or read backward DAB 7' signals, respectively). If single or doubleword boundaries are not specified, the 'read CDA overrun' signal is activated, OR'ed, then AND'ed to activate the 'turn on chain check' signal. This signal turns on the 'chain check' latch.

Another chain check condition results when the new CCW contains a count of less than one word (eight bytes) and the byte count is greater than count B (CTB). Recall that CTB is the result of adding the DAB to the low-order count bits. Since only one store operation is specified, and the B-register already contains more bytes than specified by the CTB value, a BC-equal-CTB condition (required to determine the location of the last byte of a read operation) cannot be obtained. Thus, a chain check condition results and the 'turn on chain check' signal is activated to turn on the 'chain check' latch.

Assume that the channel has received eight bytes which have already been transferred into the A-register. Further, assume that the new CCW specifies a count of less than one word (eight bytes). This means that the channel has already accepted more bytes than the count has specified; thus a chain check condition exists. The 'turn on chain check' signal is activated to turn on the 'chain check' latch.

Assume that the channel has accepted eight bytes which have been transferred to the A-register and that a number of other bytes have been gated into the B-register. Further, assume that the new CCW specifies a count of less than two words (16 bytes) and that the byte count is greater than CTB. Although two transfers to storage may be specified, the second transfer would consist of less than eight bytes. Since the number of bytes gated into the B-register (byte count) is greater than CTB, the BC-equals-CTB condition cannot be obtained. Thus, a chain check condition results and the 'turn on chain check' signal is activated to turn on the 'chain check' latch.

Assume that the channel has gated eight bytes into the A-register ('A-register full' and 'read CDA sequence 3 reset' latches on) and is gating a byte into the last position of the B-register. Further, assume that the channel has not received a valid CCW and that the CDA latch is still on. Since both the A- and B-registers are full and a valid CCW has not been established in the channel, a chain check condition exists and the 'turn on chain check' signal is activated to turn on the 'chain check' latch.

Assume that the channel has received at least eight bytes from the control unit and that eight bytes have been transferred to the A-register. Further, assume that the DAB received in the new CCW is not on doubleword boundaries ('Read DAB 0 or read backward DAB 7' signal inactive.) Since a doubleword is in the A-register and the DAB does not specify doubleword boundaries, storage of the eight A-register bytes would be meaningless. Thus, a chain check condition exists, and the 'overrun' signal (Diagram 2-4) is activated to turn on the 'chain check' latch.

Assume that the channel has received at least three but no more than eight bytes from the control unit ('byte count register equals 4' signal) and that the DAB in the new CCW is not on doubleword boundaries ('not read DAB 0 or read backward DAB 7'). This means that the channel has ceased doublegating operations, and that the number of bytes received by the channel has exceeded singleword boundaries; i.e., the DAB must specify doubleword boundaries for the channel to handle the data already received. Since the DAB is not on doubleword boundaries, a chain check condition exists and the 'overrun' signal is activated to turn on the 'chain check' latch.

With the 'chain check' latch on, the 'command reject' signal (Diagram 2-3) is activated. In turn, the 'command reject' signal activates the 'command reject or control check' signal (Diagram 2-1), to inhibit and activate specified controls in the channel. With the 'command reject or control check' signal active, the channel enters a sequence 5 routine and sends an 'interrupt request' signal to the CPU interface. When the channel receives an 'interrupt response' signal, an interrupt routine is performed to store the CSW. The output of the 'chain check' latch (Diagram 2-4) is also sent to SBI channel status gate bit 47 for storage at location 64 when the CSW is stored.

Program Check

- 'Program check' latch is turned on when channel detects programming error.
- 'Storage invalid address' signal activates 'program check' signal when channel attempts to address a non-existent storage location.
- Two consecutive CCWs with TIC commands turn on 'program check' latch.
- A non-TIC CCW with bits 37–39 not equal to zero, or count field bits 48–63 equal to zero, turns on 'program check' latch.
- A CAW with bits 4–7 or 29–31 not equal to zero turns on 'program check' latch.
- A TIC CCW with data address field bits 29–31 not equal to zero (not doubleword boundaries) turns on 'program check' latch.
- A CCW obtained during an initial selection routine with command register bits 4-7 equal to zero is an invalid operation and turns on 'program check' latch.
- Unsuccessful selection of an I/O device during a polling interrupt sequence turns on 'program check' latch.
- Active 'program check' signal:
  1. Activates 'command reject or control check' signal to initiate sequence 5 routine and store CSW.
  2. Inhibits "CCW valid and no errors' and 'gate A to SBI' signals.
  3. Is applied to SBI channel status gate bit 42 for CSW store.

A program check condition is detected when the channel detects a programming error during a channel operation. Detection of a program check condition results in termination of the operation in progress and storage of the CSW with the 'program check' bit on. Any detected program check condition turns on the 'program check' latch (Diagram 2-5). Program check conditions are detected as the result of the following program error conditions.

Assume the channel is accessing main storage to store or fetch a doubleword and gates an invalid address to the SAB. The address may be invalid if the storage location specified by the address does not exist in the system. When the invalid address is detected (in the BCU for System/360 Models 60, 65, 67-1, 75, and 85, in the Channel Controller for the Model 67-2, and in the PSCE for the Model 91), the channel receives a 'storage invalid address' signal. This signal is AND'ed (Diagram 2-5) with the 'CCW valid' and 'raw advance pulse' signals to turn on the 'storage invalid address' latch. The output of this latch is OR'ed to activate the 'program check' signal. An active 'incorrect length' signal during a write operation or a 'machine or setup reset' signal turns off the 'storage invalid address' latch or prevents the latch from latching; however, the 'storage invalid address' output is still active for the duration of the 'raw advance pulse' signal.

An invalid address may also be detected when the channel is fetching a CCW. If an invalid address is gated to the SAB, the channel receives the 'storage invalid address' signal which is AND'ed with the 'CCW fetch gated check' and 'raw advance pulse' signals (Diagram 2-5) to turn on the 'program check' latch. The latch output is then OR'ed to activate the 'program check' signal.

A program check condition exists when two consecutive CCW's contain a Transfer in Channel (TIC) command. Two consecutive TIC operations are illegal. In the channel this condition is detected by AND'ing the 'TIC cycle trigger' signal (activated when the first TIC command is detected) with the 'turn on TIC latch' signal (activated when the second TIC command is detected). AND'ing these two signals turns on the 'program check' latch to activate the 'program check' signal.

A program check condition exists when a non-TIC CCW is fetched and bits 37–39 are not all zeros and bits 48–63 (count field bits) are equal to zero. A logic 1 in bits 37–39 or a count of zero constitutes an improperly programmed CCW. In the channel, the above bits are gated by the 'gate CCW error' signal (Diagram 2-5) If an error is detected, the condition is AND'ed enabling the 'late advance pulse' and 'not TIC' signals to activate the 'non-TIC CCW error' signal, which turns on the 'program check' latch.

On a CAW fetch operation, a program check condition exists if CAW bits 4–7 or 29–31 are not equal to zero. Bits 4–7, though not part of the storage protect key, are gated to the storage protect register and must be all zeros or a program violation exists. Bits 29–31 are part of the data address field used to fetch the CCW. Since CCW's are fetched only on doubleword boundaries, bits 29–31 must be all zeros (specifying a doubleword boundary address) or a program violation exists. In the channel, the CAW bits in question are gated by the 'turn on TIC' signal (activated for the CCW fetch operation that follows). If the bits are not all zeros, the 'CAW error' signal is activated to turn on the 'program check' latch.

When a CCW with a legal TIC command is obtained by the channel, data address field bits 29–31 must equal zero to specify doubleword boundaries for the CCW fetch operation that follows. If not, a program check condition exists. In the channel, SBO bits 29–31 are gated by the 'TIC operation' signal. If the bits are not all zeros, the 'TIC CCW error' signal is activated to turn on the 'program check' latch.

When the channel obtains a CCW during the initial selection routine, command register bits 4–7 must not be all zeros. These bits specify the channel operation to be performed, and thus must be other than all zeros for the channel to initiate an operation. If these bits are all zeros, the 'invalid operation' signal is activated and AND'ed with the 'CCW valid', 'setup', and 'storage cycle complete' signals (Diagram 2-5) to turn on the 'program check' latch.

A program check condition also results when the channel performs a polling interrupt sequence and is unable to select an I/O device. During the polling interrupt sequence, the channel sends the unit address byte, 'address out' signal, and the 'select out' signal to the I/O interface. If an I/O device is not selected, the channel receives a 'select in' signal. In the channel this signal is AND'ed with the 'PIT extended,' 'setup', 'not operational in or status in' and 'not command reject or control check' signals (Diagram 2-5) to turn on the 'program check' latch. The 'program check' latch is reset when a 'write CDA wrong length record' or 'machine or setup reset' signal is generated.

With the 'program check' signal active, the 'command reject' signal (Diagram 2-3) is activated. In turn, the 'command reject' signal activates the 'command reject or control check' signal (Diagram 2-1) to inhibit and activate specified controls in the channel. With the 'command reject or control check' signal active, the channel enters a sequence 5 routine and sends an 'interrupt request' signal to the CPU interface. When the channel receives an 'interrupt response' signal, an interrupt routine is performed to store the CSW.

The 'program check' signal also inhibits generation of the 'CCW valid and no errors' signal. This prevents: (1) turn-on of the clock; (2) an address comparison (performed during I/O device selection); (3) generation of the 'no selection gated' signal which normally activates condition code 3; and (4) the 'write CDA store data fetch' signal which normally turns on the 'storage request' latch. These four functions (or signals) are inhibited to prevent the possibility of the indicated operations from occurring after detection of the program check condition.

In addition, the 'gate A to SBI' signal is inhibited by the 'program check' signal to prevent gating of data to the SBI if the channel had previously requested a store operation. The 'program check' signal is applied to SBI channel status gate bit 42 and is stored as part of the CSW.

Unit Check

- Unit check condition detected by control unit; presented to channel with status byte (bus-in bit 6 active).

- Initial selection routine detection of unit check causes CSW store and operation termination; condition code 1 and 'release' signal sent to CPU.

- Normal operation detection of unit check results in normal sequence 5 operation; CSW is stored and 'release' signal sent to CPU. Chain command operations are terminated, if chain command flag active.

- Polling interrupt sequence clears device-end status byte when unit check condition is detected after channel-end status byte is received by channel.
  1. Device-end byte stacked back to control unit.
  2. Channel forces test I/O routine.
  3. Unit address register contents sent to 'unit address bus in' lines.
  4. CSW stored at address 64; 'release' signal sent to CPU interface.

Unit check conditions are detected by the control unit and presented to the channel in the control unit status byte with bus-in bit 6 active. In the control unit, the cause of the unit check condition is entered in the "sense" information. The channel may be affected by the unit check condition during an initial selection routine, during normal channel read-type or write-type operations, or upon normal completion of an operation when the channel receives a status byte with the channel-end bit active and the device-end bit inactive. In the last case, the unit check condition is detected after the channel-end status byte is presented to the channel; the unit check condition is then presented to the channel during a polling interrupt sequence.

If a unit check condition is detected during the initial selection sequence, the channel receives a 'status in' signal in response to the 'address out' signal, and only the 'unit check' bit in the status byte is active. In the channel, the 'address out' and 'status in' signals are AND'ed (ALD CC111) to turn on the 'sequence 5' latch. The 'unit check' bit activates the 'interrupt status' signal (ALD CK121), enabling turn-on of the 'sequence interrupt end' latch. The channel then proceeds to store the CSW and send condition code 1 and a 'release' signal to the CPU.

If the unit check condition is detected during normal operation, the channel receives a 'status in' signal with the 'unit check', 'channel end', and possibly the 'device end' bit active. In this case, the channel enters a normal sequence 5 routine and the 'interrupt status' signal is activated by the active 'unit check' bit (bus-in bit 6). With the 'interrupt status' signal active, 'chain command' operations are inhibited if the chain command flag is active. During the sequence 5 routine, the channel sends an 'interrupt request' signal to the CPU interface, and, upon receipt of the 'interrupt response' signal, stores the CSW at address location 64 and sends a 'release' signal to the CPU interface.

Assume the channel receives a status byte with only the 'channel end' bit active, and that a unit check condition is detected in the control unit prior to presenting the status byte with the device end bit active. When the device end

status byte is presented, the channel performs a polling interrupt sequence. During the sequence, the channel stacks the status back to the control unit, then forces a test I/O routine to clear the polling interrupt. During the test I/O routine, the status byte is obtained from the control unit. The channel then sends an 'interrupt response' signal, and gates the unit address register contents to the 'unit address bus in' lines; in addition, the channel stores the CSW ('device end' bit on) at address location 64 and sends a 'release' signal to the CPU interface.

CSW STATUS BYTES

Bytes comprising the channel status word (CSW) are assembled in the channel and stored in main storage location 64 at the end of each channel operation. If a logout operation is performed, the CSW is stored in main storage location 304 and then, after the logout operation is complete, is stored in main storage location 64. The CSW bytes are assembled in the channel to provide program information describing the condition of the channel at the end of an operation; the CSW is stored regardless of whether the operation is completed, or is prematurely terminated due to a detected error. Once stored, the CSW is available until altered by storage of another CSW or until cleared under program control.

The CSW assembled in the channel (Diagram 2-6, FEMDM) consists of the contents of the flag register, the contents of the command address register, the unit status byte, the channel status byte, and the contents of the count register. Each is described separately in the following paragraphs; the descriptions are based upon Diagram 2-6.

Storage Protect Register Byte

- Storage protection key byte consists of storage protect register bits 0-3 plus zeros in SBI bit positions 4-7.
- Storage protect key originally obtained from CAW fetched by channel.
- Parity for storage protection key byte generated by SBI parity generator after byte is gated by 'gate command address register to SBI' signal.
- 'BCU response' signal gates byte to SBI 0-7 plus P0 lines.

The storage protect register byte consists of the storage protection key from bits positions 0-3 and all zeros in byte positions 4-7. The storage protect register contains the storage protection key obtained when the CAW was fetched by the channel during a start I/O initial selection routine. When the CSW is stored, the storage protect register bits are gated to SBI lines 0-3 by the 'gate command address register to SBI' signal (Diagram 2-6). Parity for the storage protect key byte is generated by the SBI parity generator. Upon receipt of the 'BCU response' signal, supplied to the channel as a result of a storage request to store the CSW, the storage protect key byte is gated to SBI positions 0-7 plus P0.

Command Address Register Bytes

- CSW command address field bytes consist of command address register bits 0-20 to SBI 8-28 lines and zeros to SBI 29-31 lines.
- Command address field usually contains storage address eight bytes higher than last CCW used by channel.
- Parity for the three command address field bytes are generated by SBI parity generator after bytes are gated by 'gate command address register to SBI' signal.
- 'BCU response' signal gates byte to SBI 8-31 plus P1, P2, and P3 lines.

The three command address field bytes (Diagram 2-6), are assembled in the channel by gating command address register bits 0-20 to SBI bit positions 8-28 and logic zeros (doubleword boundaries) to SBI positions 29–31. The command address register normally contains a command address eight bytes higher than the last (or current) CCW used by the channel. However, if the channel detects an error during a CAW fetch or during a CCW fetch before the register is updated, the address value may be zero or the address of the current CCW.

For a store log 1 or CSW store operation, the command address field bytes are gated through the SBI gates by the 'gate command address register to SBI' signal. The SBI parity generating logic generates SBI parity bits P1, P2, and P3 for the command address field. Upon receipt of the 'BCU response' signal (for the store log 1 or CSW store operation), the command address field bytes (bits 8-31 plus P1, P2, and P3) are gated to the SBI lines for storage.

Unit Status Byte

- Status byte from control unit (via bus-in latches) comprises fifth byte of CSW; byte is referred to as unit status byte.
- For store logword 1 or CSW store operations, bus-in bits 0-7 are gated through SBI bit gates 32-39 by 'gate control unit to SBI' signal.

- P4 parity bit generated by SBI parity generator logic.
- 'BCU response' signal gates 'unit status byte to SBI lines' for storage.
- Unit status bytes are designated 'attention', 'status modifier', 'control unit end', 'busy', 'channel end', 'device end', 'unit check', and 'unit exception'.

The status bits received from the control unit via the channels bus-in latches (Diagram 2-6), comprise the fifth byte of the CSW, and are referred to as the unit status byte. Bits of this byte provide status information pertaining to the control unit or I/O device with which the channel is operating. When the CSW is stored, during a store logword 1 or CSW store operation, the unit status byte (bus-in bits 0-7) is gated to SBI bit gates 32-39 by the 'gate control unit status to SBI' signal. After gating, the SBI parity generator generates the parity bit (SBI P4) for the unit status byte. Subsequently, the 'BCU response' signal gates the unit status byte to SBI lines 32-39 and P4 for storage.

In addition to providing information for the CSW store, certain bits of the 'unit status' byte affect channel operations in progress. A description of each of the eight unit status bits and their significance to channel operations are presented separately in the following paragraphs. Unit status bits 0-7 are respectively designated 'attention', 'status modifier', 'control unit end', 'busy', 'channel end', 'device end', 'unit check', and 'unit exception'.

Attention Bit

The 'attention' signal (bus-in bit 0) is generated by the I/O device when an asynchronous indication occurs in the I/O device. Presentation of the 'attention' signal to the channel is under control of the I/O device and is normally not presented to the channel until operations in progress at the control unit and I/O device are complete. The 'attention' signal may or may not be presented in the status byte with the 'device end' signal. Depending upon the I/O device, the 'attention' signal may or may not suppress command chaining operations.

When the unit status byte with the 'attention' bit active is presented to the channel, the channel either stacks the status byte back to the control unit or requests an interrupt to store the CSW, depending upon the operating status of the channel at the time the status byte is presented. When the status byte with the active 'attention' bit is stored, it occupies bit position 32 of the CSW.

Status Modifier Bit

- 'Status modifier' and 'busy' bits cause control-unit-busy sequence; detected during start I/O or test I/O initial selection routine, control-unit-busy causes CSW store operation.
- 'Status modifier' and 'device end' bits active during chain command operation causes channel to jump one CCW in the sequence and fetch a CCW from a storage location eight bits higher; if chaining is not involved, CSW is stored.
- 'Status modifier' alone indicates I/O device cannot respond to Test I/O command; CSW is stored.

The 'status modifier' bit (bus-in bit 1) is supplied by the control unit in three different situations: (1) with the 'busy' bit active; (2) with the 'device end' bit active, and (3) with only the 'status modifier' bit active. When received in the unit status byte with the 'busy' bit active, a control-unit-busy condition exists. This occurs when the channel performs a start I/O or test I/O initial selection routine to a busy I/O device. When this occurs, the control unit responds to the 'address out' signal from the channel with a 'status in' signal; in addition the control unit places the unit status byte with the 'status modifier' and 'busy' bits active on the bus-in lines. With both the 'address out' and 'status in' signals active, the channel enters a sequence 5 routine, followed by an interrupt routine to store the CSW containing the 'status modifier' bit (CSW position 33) and 'busy' bit (CSW position 35).

When the 'status modifier' bit is received with the 'device end' bit during a chain command operation (Figure 6-9), the channel performs a jump command address sequence. During the chain command sequence 5 routine, the channel updates the command address register contents by the equivalent of eight bytes, then uses the updated address to fetch the next CCW. In other words, the next CCW is not fetched from a sequential address location, but from a location eight bytes higher. If a chain command operation is not in progress when both the 'status modifier and 'device end' bits are received, the jump operation is not performed and the bits are stored with the CSW as during a normal ending sequence.

If the 'status modifier' bit is supplied alone, the I/O device cannot provide a status response to a Test I/O command. In this case, the channel stores the 'status modifier' bit on the CSW.

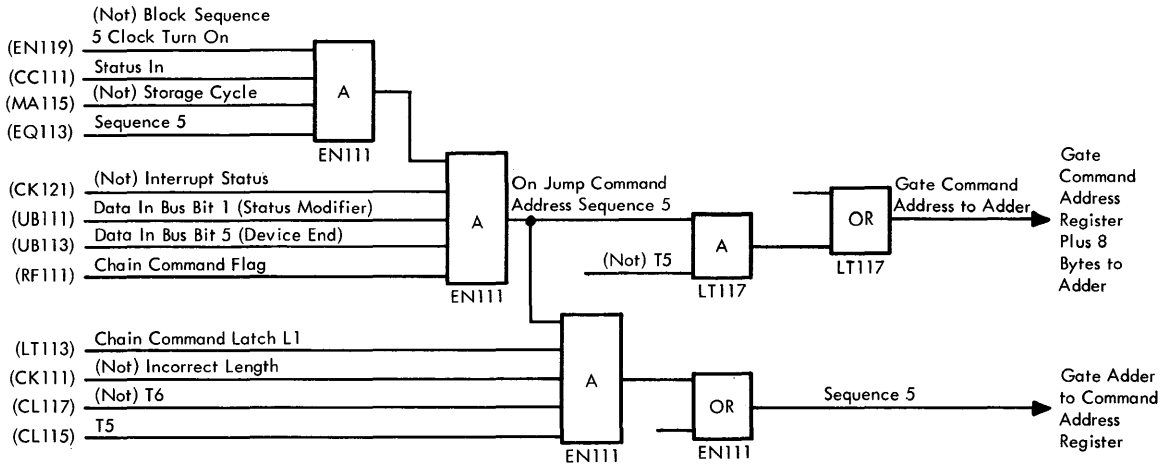


Figure 6-9. Device End and Status Modifier, Chain Command

Control Unit End Bit

The control unit sends the 'control unit end' bit (bus-in bit 2) in the unit status byte to indicate that a control unit is now free for use by the program. The 'control unit end' bit is sent by the control unit only if the control unit has responded to a previous command with a control-unit-busy condition. (Refer to "Busy Bit.") The 'control unit end' bit, when received in the unit status byte by the channel, causes an interrupt and is stored in bit position 34 of the CSW. The unit status byte containing this bit may be stacked back to the control unit by the channel and cleared by a polling interrupt sequence.

Busy Bit

The 'busy' bit (bus-in bit 3), when accompanied by the 'status modifier' bit, is interpreted by the channel as a control-unit-busy condition. (Refer to "Status Modifier Bit.") The 'busy' bit, unaccompanied by a 'status modifier' bit, indicates that the I/O device cannot accept a new command either because it is performing a previously initiated operation or because it contains an interrupt condition. The 'busy' bit is off in any CSW that is formed by the Test I/O instruction unless a previously initiated I/O operation is being performed. When stored, the 'busy' bit occupies position 35 in the CSW.

Channel End Bit

The 'channel end' bit (bus-in bit 4) is sent to the channel in the unit status byte when the portion of an I/O operation involving transfer of data or control information between the I/O device and the channel is completed. When the channel receives the unit status byte with the 'channel end' bit, an interrupt is requested to store the CSW at main storage location 64. The 'channel-end' bit is stored in the CSW in bit position 36 and indicates to the program that the channel is now free to perform another operation.

Device End Bit

A control unit sends a unit status byte with the 'device end' (bus-in bit 5) active to the channel when the I/O device completes all device operations associated with an I/O operation. If the 'chain command' bit in the flag register is active, the 'device end' bit causes the channel to perform a chain command operation. Unless the 'channel end' and 'device end' bits are presented to the channel simultaneously, the unit status byte containing the 'device end' bit is stacked back to the control unit and cleared with a polling interrupt routine. If both the 'channel end' and 'device end' bits are presented to the channel together (assuming the chain command flag is inactive), status is not stacked, and the channel requests an interrupt to store the CSW as during a normal ending sequence. The 'device end' bit is stored in position 37 of the CSW. A control unit with a stacked 'device end' bit does not make a control unit "busy" to a new instruction with a different I/O device address.

Unit Check Bit

The control unit sends a 'unit check' bit (bus-in bit 6) when the I/O device or control unit detects an unusual condition which can be reflected by the sense bits in the control unit or device. The 'unit check' bit can be stacked

back to the control unit and cleared with a polling interrupt sequence; if the 'unit check' bit is accompanied by the 'channel end' and/or 'device end' bits, channel operations end, even though a chain data or chain command flag may be active in the flag register. The 'unit check' bit is stored in CSW bit position 38.

Unit Exception Bit

The 'unit exception' bit (bus-in bit 7) is presented to the channel when an I/O device detects a condition that usually does not occur. The 'unit exception' bit has only one meaning for a specific command performed by a specific I/O device. If the unit-exception condition is detected during an initial selection sequence, the 'unit exception' bit is sent to the channel in the initial status byte; the channel terminates the operation and stores the CSW with the 'unit exception' bit in position 39 of the CSW. If the unit-exception condition is detected while the I/O device is performing an operation, the 'unit exception' bit may be accompanied by the 'channel end', 'control unit end', or 'device end' bit, depending upon when the condition is detected. Thus, the channel may perform a normal interrupt routine to store the CSW or stack the unit status byte back to the control unit by a polling interrupt sequence, depending upon the operating status of the channel at the time.

Channel Status Byte

- Channel status byte reflects channel conditions at end of operation.
- Eight channel status byte bits gated to SBI gate bit positions 40-47 by 'gate channel status to SBI' signal.
- SBI parity generating logic generates SBI parity bit P5 for channel status byte.
- 'BCU response' signal for store log word 1 or store CSW operation gates channel status byte to SBI lines for storage.

The channel status byte (Diagram 2-6) comprises the sixth byte of the CSW. Bits of this byte provide information to indicate channel conditions at the end of the preceding channel operation. When the CSW is stored during a store log word 1 or CSW store operation, the channel status byte is gated to SBI bit gates 40-47 by the 'gate channel status to SBI' signal. After gating, the SBI parity generator generates the parity bit (SBI P5) for the channel status byte. Subsequently, the 'BCU response' signal gates the unit status byte to SBI lines 40-47 and P5 for storage.

The eight channel status bits are described separately in the following paragraphs. These bits (SBI lines 40-47) are: 'program controlled interrupt status', 'incorrect length', 'program check', 'storage protect check', 'channel data check', 'channel control check', 'interface control check', and 'chain check'.

Program Controlled Interrupt Status Bit

The 'program controlled interrupt status' bit (CSW position 40) is active when the 'program controlled interrupt status' latch is turned on (Diagram 2-6). This latch is turned on after the channel fetches a CCW with the PCI flag active. The interrupt routine initiated by the active PCI flag is performed as soon as possible after the CCW is fetched; however, the interrupt routine may be delayed an unpredictable time if channel interrupts are masked in the CPU or if other system activity prevents performance of the routine. Although detection of the PCI flag causes the channel to request an interrupt, the current I/O operation in progress is not disrupted. (The current I/O operation may be temporarily delayed when the requested interrupt results in storage of the CSW; however, the delay will result only if the CSW store operation conflicts with a store or fetch operation during the current I/O operation.) When the PCI interrupt request is honored, the CSW reflects the status of the channel at the time the CSW is stored. The 'program controlled interrupt status' (PCI status) bit is stored in position 40 of the CSW.

Incorrect Length Bit

The incorrect length (IL) condition (wrong length record condition) is detected by the channel any time the data-byte block length on the I/O device and the count received in the CCW do not agree. For example, if the CCW count for a read operation from a tape unit is 100 bytes and the data-byte block length on the tape unit is only 50 bytes, the operation ends with the IL latch on after 50 bytes are read and an inter-block gap is reached. The suppress incorrect length (SLI) flag in the flag register must be off to allow presentation of the IL condition to the program except on a chain-data operation. (The IL condition is presented to the program by calculating a residual count, placing this count in the count registers, and storing the count register contents as part of the CSW.) Detection of the IL condition suppresses chain command operations (if applicable) and causes the channel to request an interrupt to store the CSW. The IL bit is stored in CSW bit position 41 (Diagram 2-6).

Program Check Bit

The 'program check' bit is stored in bit position 42 of the CSW (Diagram 2-6). For a description of the conditions causing a program check condition and channel operations that subsequently result, refer to "Program Check".

Storage Protection Check Bit

The 'storage protection check' bit is stored in bit position 43 of the CSW (Diagram 2-6). For a description of the condition causing a storage protect check condition and channel operations that subsequently result, refer to "Storage Protect Check".

Channel Data Check Bit

The 'channel data check' bit is stored in bit position 44 of the CSW (Diagram 2-6). For a description of the conditions causing a channel data check condition and channel operations that subsequently result, refer to "Channel Data Check".

Channel Control Check Bit

The 'channel control check' bit is stored in bit position 45 of the CSW (Diagram 2-6). For a description of the conditions causing a channel control check condition and channel operations that subsequently result, refer to "Channel Control Check".

Interface Control Check Bit

The 'interface control check' bit is stored in bit position 46 of the CSW (Diagram 2-6). For a description of the conditions causing an interface control check condition and channel operations that subsequently result, refer to "Interface Control Check".

Chain Check Bit

The 'chain check' bit is stored in bit position 47 of the CSW (Diagram 2-6). For a description of the conditions causing a chain check condition and channel operations that subsequently result, refer to "Chain Check".

Count Register Bytes

- Count register bits 8-23 gated through SBI gate bits 48-63 by 'gate count register to SBI' signal.
- SBI parity generator logic generates parity bits P6 and P7 for the two count field bytes.
- 'BCU response' signal for store log word 1 or CSW store operation gates count field bits 48-63 plus P6 and P7 to SBI lines for storage.

Count register bits 8-23 comprise the two remaining bytes of the CSW (Diagram 2-6, FEDM). Depending upon the conditions under which the channel operation was terminated, the count register contents may or may not be meaningful. For example, if the operation is terminated normally without error conditions, the count should be zero. If an incorrect length condition was detected, the count could reflect the residual count (providing the SLI flag is off). If an operation is terminated due to a detected error, the count is an indication of the operation's progress before the error was detected.

Count register bits 8-23 are gated to SBI gates 48-63, respectively, by the 'gate count register to SBI' signal. After gating, the SBI parity generator logic generates parity bits P6 and P7 for the two count field bytes. When the 'BCU response' signal (for a store log word 1 or CSW store operation) is received by the channel count field, bits 48-63 plus P6 and P7 are gated to the SBI lines for storage.

Data Transfer Rate . . . . . .1.3 megabytes/second (maximum)

Power

Primary power options: 60 ± 1 Hz, 208/230V +10 to -8%, 3-phase 4-wire delta.
50 ± 1 Hz, 195/220/235V +10 to -8%, 3-phase 4-wire delta.
50 ± 1 Hz, 380/408V +10 to -8%, 3-phase 5-wire wye.

	Power Input	Power Factor	Service Rating	Heat Dissipation
Model 1	3.05 KVA	0.7 (min)	15 amperes	8,200 BTU/hour
Model 2	3.65 KVA	0.7 (min)	15 amperes	10,000 BTU/hour
Model 3	4.25 KVA	0.7 (min)	15 amperes	11,600 BTU/hour

Convenience Outlets:

60 Hz machines - two pair per frame, 115V ac output, 20 ampere physical rating, convenience power fused at 10 amperes in the primary of the step-down transformer.

50 Hz machines - two pair per frame, 220V ac output, 15 ampere physical rating, convenience power fused at 5 amperes in the load side of the auto-transformer.

Cooling . . . . . .Forced room air; blower fans on each installed channel gate and frame power section.

Environmental Limits

	Not Operating	Operating
Temperature	50 - 110°F	60 - 90°F
Rel Humidity	8 - 80%	8 - 80%
Max. Altitude	7,000 ft	7,000 ft

Frame Size and Weight (Based on Model 3)

	Width	Length	Height	Volume	Floor area	Weight
With covers	32.25 in.	73.50 in.	70.75 in.	89 ft^3	15.1 ft^2	2000 lbs
Without covers	29 in.	60 in.	70 in.	70 ft^3	12.1 ft^2	1800 lbs

Figure A-1. 2860 Unit Characteristics

Appendix B. Special Circuits

There are no special circuits for the 2860 Selector Channel.

Appendix C. ALD Contents By Prefix

Figure C-1 provides alphabetic access to general areas of the 2860 automated logic diagrams. This figure is primarily a CE aid to recall.

		MA	Storage Controls
AS	I/O and CPU Interfaces	MB	Storage Bus
AT	Polling Interrupt	MT	Metering
BA	Prefix ID Register (Feature)	RA, RB	Register A, Register B
CC	Interface Lines	RC	Command Address Register
CE	Manual Controls	RD	Data Address Register
CK	Status	RE	Count Register
CL	Clock	RF	Flag Register
CR	Read	RG	Command Register
CS	Setup	RM	Mark Registers
CT	Test I/O	RP	Storage Protect Register
CW	Write	RS	Simulate I/O
CX	Channel Transfer	RU	Unit Address Register
DU	Data Address Update	RW	Register Ingate Powering
EN	Interrupt	SA	Adder
EQ	Sequences	SC	Byte Counter
ER	Reset	UB	Unit Data Bus (In and Out)
FN	Model 91	WA	I/O Interface Lines
		WC	Processor Interface Lines
HT	Halt I/O	XR, XS, XV, XW	Indicators, Switches, Wiring, Board Cables
LG	Log Controls	YD	Power (YA in early models)
LP	IPL and FLT	ZZ	Socket Lists
LT	Triggers		

Figure C-1. 2860 ALD Contents by Prefix

2860 Selector Channel
FETOM

Order No. SY27-2220-1

Cut or Fold Along Line

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

If you desire a reply by the group that prepared this manual, include your name and address.

From

NAME _____ OFFICE/DEPT NO. _____

CITY/STATE _____ ZIP CODE _____ DATE _____

- How did you use this publication?

  ☐ As a reference source ☐ As a classroom text ☐ As a self-study text

We would appreciate your comments; please give section or figure titles where appropriate.

- What sections or figures were particularly useful or understandable to you?

- What sections or figures could be improved?

- What sections or figures require additional information?

- Any other comments?

- How do you rate this manual?

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along Line

Reader's Comment Form

Fold

Fold

FIRST CLASS
PERMIT NO. 419
POUGHKEEPSIE, N.Y.

Business Reply Mail
No postage stamp necessary if mailed in U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department B97
P.O. Box 390
Poughkeepsie, New York 12602

Fold

Fold

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

System Maintenance Library

cut here

SY27-2220-1

IBM 2860 Selector Channel Printed in U.S.A. SY27-2220-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)