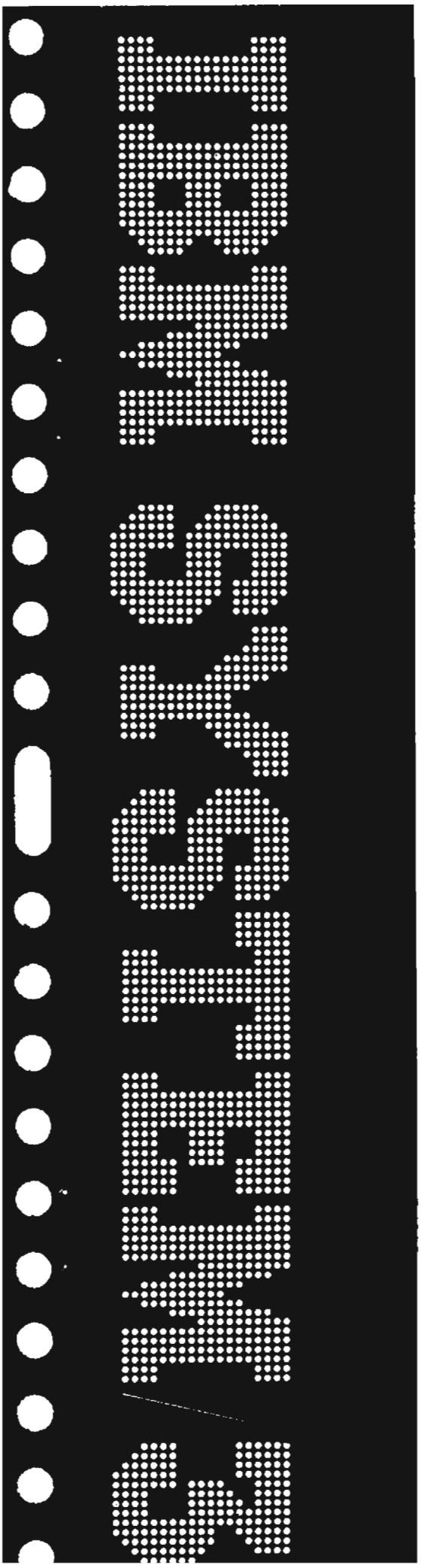


Field Engineering

Theory of Operation

5424 Multi-Function Card Unit Attachment

SY31-0253-0

Field Engineering

Theory of Operation

5424 Multi-Function Card Unit Attachment

Preface

This manual, Form SY31-0253, describes the operation of the 5424 Multi-Function Card Unit Attachment. Use ALDs (automated Logic Diagrams) and the *IBM Field Engineering Maintenance Diagrams, 5424 Multi-Function Card Unit Attachment,* Form SY31-0254 in conjunction with this manual.

This attachment is the interface between the 5424 Multi-Function Card Unit (MFCU) and the 5410 Central Processing Unit (CPU). A knowledge of MFCU operations and CPU operations is necessary to understand MFCU attachment operations.

Other manuals needed to understand and service the 5424 MFCU attachment are:

1. The *IBM 5424 MFCU Attachment, Field Engineering Maintenance Diagrams Manual,* Form SY31-0254.
2. The *IBM 5424 MFCU, Field Engineering Theory of Operations Manual,* Form SY31-0213.
3. The *IBM 5424 MFCU, Field Engineering Maintenance Manual,* Form SY31-0230.
4. The *IBM 5410 Central Processing Unit, Field Engineering Theory of Operations Manual,* Form SY31-0207.
5. The *IBM 5410 Central Processing Unit, Field Engineering Maintenance Diagrams Manual,* Form SY31-0202.
6. The *IBM 5410 Central Processing Unit, Field Engineering Maintenance Manual,* Form SY31-0244.
7. For machine characteristics, refer to the *IBM System/3 Installation Manual-Physical Planning,* Form A21-9084.

First Edition

Some illustrations in this manual have a code number in the lower corner. This is a publishing control number and is not related to the subject matter.

Changes are continually made to the specifications herein; any such change will be reported in subsequent revisions or FE Supplements.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Product Publications, Department 245, Rochester, Minnesota 55901.

Contents

Abbreviations

ALU	Arithmetic and Logical Unit
APL	Advance Program Level
CPU	Central Processing Unit
CS	Cycle Steal
CSR	Cycle Steal Request
DA	Device Address
DBI	Data Bus In
DBO	Data Bus Out
EBCDIC	Extended Binary Coded Decimal Interchange Code
FEMD	Field Engineering Maintenance Diagrams
I-O	Input-Output (I/O)
IPL	Initial Program Load
LIO	Load Input-Output
LSR	Local Store Register
MFCU	Multi-Function Card Unit
MPCAR	MFCU Punch Data Address Register
MPTAR	MFCU Print Data Address Register
MRDAR	MFCU Read Data Address Register
MST	Monolithic System Technology
NPRO	Non-Process Run Out
SAR	Storage Address Register
SIO	Start Input-Output
SIOC	Serial I/O Channel
SLD	Solid Logic Dense
SNS	Sense Input-Output
SS	Single Shot
TIO	Test Input-Output and Branch

The IBM 5424 Multi-Function Card Unit Attachment is the interface between the IBM 5410 Central Processing Unit (CPU) and the IBM 5424 Multi-Function Card Unit (MFCU). The four input/output operations controlled by the attachment are reading, punching, printing, and selective stacking.

In data flow the attachment logic is between the CPU and the MFCU. Attachment logic components are on gate A panel A3 in the CPU.

The attachment circuits are monolithic system technology (MST) logic. All magnet control interface lines between the attachment board and the MFCU electronics board are solid logic dense (SLD) levels. MST conversion occurs at the attachment board. The CB signals are sent directly to the attachment.

The 5424 attachment provides a way for the MFCU to use the facilities of the CPU to communicate with core storage. Data and control information are transferred to and received from the CPU under control of stored program instructions. Also, sense information indicating attachment and MFCU status is transferred to the CPU when a programmed sense instruction is executed. However, some control lines from the MFCU are not under program control. These control lines are used to control operations and indicate conditions requiring operator action.

By means of a fixed cycle steal priority, I/O cycle steal requests are taken between any two CPU cycles to access core storage for fetching or storing data.

Each MFCU station performs a specific function. No main clutch or index is used to control or synchronize the read, punch, print, stack, or combination of these four functions. The MFCU is an asynchronous device, therefore, the attachment synchronizes the data transferred between the MFCU and the CPU, and synchronizes the electrical-mechanical operations. These attachment timing considerations prevent card misfeeds and data transfer errors.

The attachment logic prevents a card from leaving the selected wait station until the card in the transport is advanced to a point where the card ejected from the wait station cannot interfere with the card in the transport. This attachment synchronization allows the MFCU to perform all four functions (on different cards) at the same time; thereby, allowing maximum throughput.

Throughput to and from the MFCU depends upon the MFCU model. The two 5424 MFCU models and speeds are:

- Model A1—reads 250 cards per minute (cpm), punches 60 cpm, and prints (three lines) 60 cpm. (The printer has eight typewheels.)
- Model A2—reads 500 cpm, punches 120 cpm, and prints (three lines) 120 cpm. (The printer has 16 typewheels.)

Printing four lines slows the Model A1 to 48 cpm and the Model A2 to 96 cpm.

DATA FLOW

Commands are received by the 5424 through the signal cables from the attachment circuits (Figure 1-1). These commands, in the form of electrical signals, activate the necessary clutches and magnets. The signals sent to the attachment unit from the MFCU are read data, punch check data, photoelectric card lever signals, and timing signals. The card lever signals indicate card position in the transport. The timing signals are sent to the attachment as magnetic emitter pulses (CB pulses). The attachment uses these pulses to synchronize the electrical-mechanical functions.

Data is sent from core storage to the attachment over data bus out (DBO). Data is sent from the MFCU to the CPU over data bus in (DBI). Refer to FEMD 3-5 for MFCU data flow and control. Diagrams 3-010, 3-015, and 3-020 show data flow and control for the read, punch, and print operations.

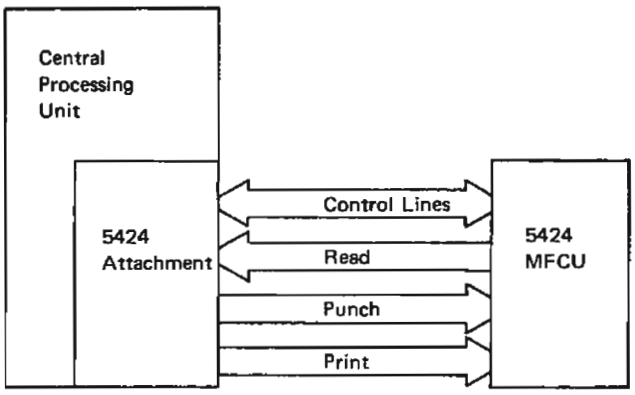


Figure 1-1. 5424 Data Flow

INSTRUCTIONS

- The LIO instruction loads an MFCU read, punch, or print local store register (LSR) located in the CPU.

- The TIO instruction checks for a condition specified in the N field and branches if the condition is met.

- The APL is the same as the TIO instruction except the program advances to the next program level if the condition is met. This instruction is provided primarily for machines with the dual program feature. Without the dual program feature, the APL instruction waits until the condition tested for is not met.

- The SNS instruction stores two sense bytes from the attachment or the MFCU to indicate status and timing conditions. The SNS instruction can also store the contents of the three LSRs.

- The SIO instruction starts the read, punch, print, and/or stack operation(s).

The LIO, TIO, and SNS instructions are one address instructions. These instructions may be three or four bytes long. Byte one contains the op code; byte two contains the Q byte; and bytes three and four contain the storage address. The storage address field can be one or two bytes long. The LIO, TIO and SNS instructions may also use index register one or two.

The APL and SIO instructions are three byte instructions. Byte one contains the op code; byte two contains the Q byte; and byte three contains the SIO control codes. Byte three is not used by the APL instruction.

The CPU sends an I/O instruction to all system I/O devices. Each device, in turn, decodes the device address field. If the device decodes its address, a response must be sent to the CPU to indicate device status.

If the MFCU decodes its address of F (hexadecimal), the MFCU sends condition A, condition B, or conditions A and B to the CPU (Figure 1-2).

A condition A signal indicates a busy condition; therefore, the instruction is rejected. A condition B signal indicates to the CPU that the instruction is accepted by the MFCU.

If both condition A and condition B are sent to the CPU, a parity check occurs and the instruction is not accepted.

An invalid address is indicated if neither condition A nor condition B is sent to the CPU by any device by IQ clock 8 time.

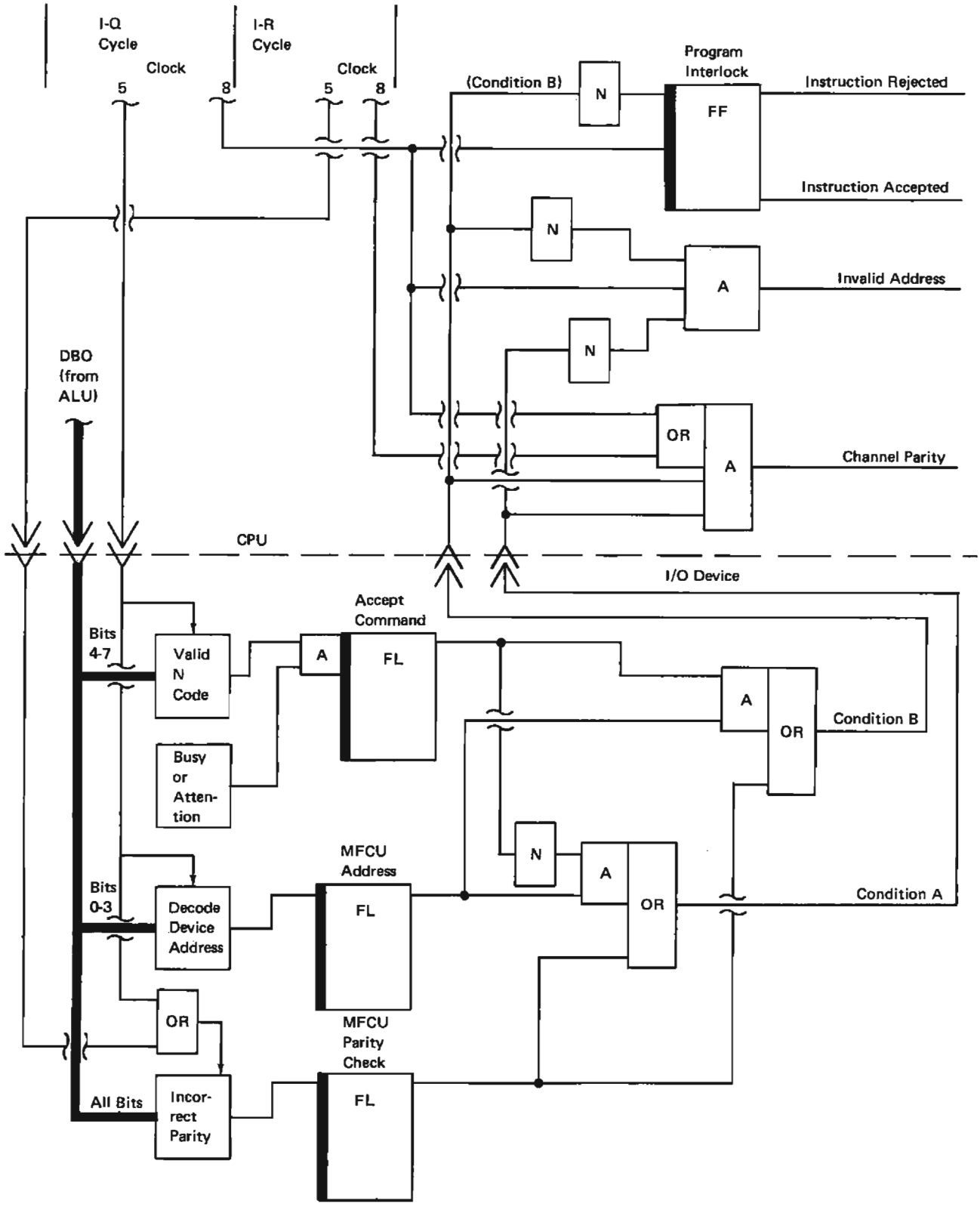


Figure 1-2. SIO–CPU/MFCU Control

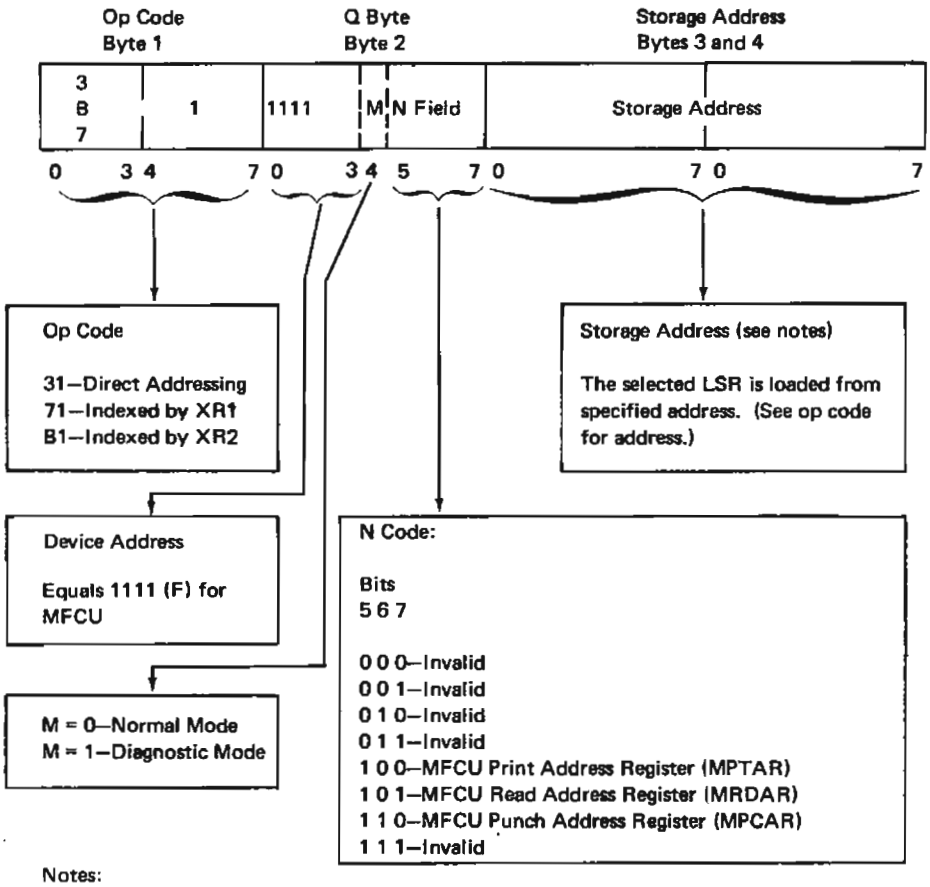
Load I/O (LIO) Instruction

The LIO instruction loads an MFCU read local store register (LSR), punch LSR, or print LSR located in the CPU before the SIO instruction. These LSR registers contain the storage address used during a read, a punch, or a print operation. For read and punch operations, loading the selected LSR low order byte to 00 or 80 hexadecimal (128 decimal) is good practice. This places the LSR at boundaries of 128 bytes. The print address *must be* on boundaries of 256 bytes with the low order byte equal to 00 (hexadecimal).

The read or punch address registers must be loaded before each read or punch operation. The print address register does not require loading before each print operation unless a different area in core storage is required.

Figure 1-3 shows the MFCU LIO instruction format and the control codes for the functions performed.

If bit 4 of the Q byte equals one, a diagnostic read or diagnostic punch operation is performed during the next SIO read or SIO punch command execution. See Chapter 3 for diagnostic read and diagnostic punch operations.



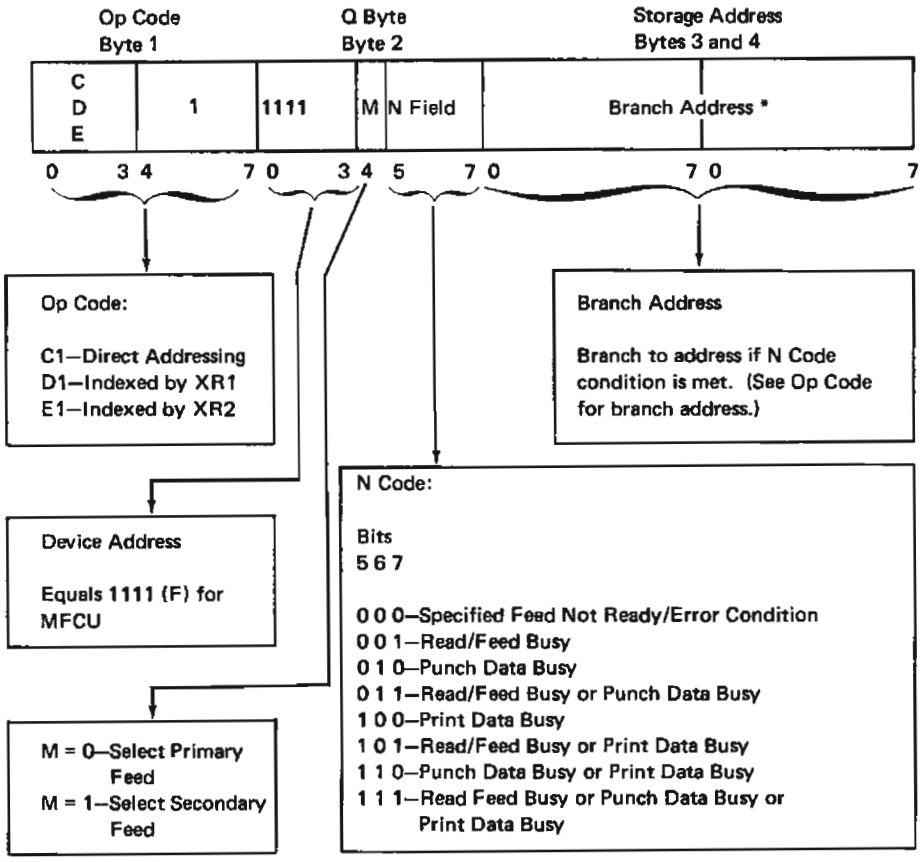
Notes:
1. Storage address can be one byte or two bytes in length.
2. The selected LSR is loaded with two bytes of data from specified address and from the specified address minus one (0401 would load from storage address 0400 and 0401).

Figure 1-3. MFCU LIO Instruction Format

Test I/O and Branch (TIO) Instruction

The TIO instruction checks for the condition indicated in the N field of the Q byte. If the branch condition is met, the instruction branches to the address specified by the branch address byte of the instruction.

Figure 1-4 shows the MFCU TIO instruction format.



* Branch address may be one byte or two bytes long.

Figure 1-4. MFCU APL Instruction Format

Advance Program Level (APL) Instruction

The APL instruction is provided primarily for machines with the dual program feature.

Figure 1-5 shows the APL instruction format and conditions checked. Operation of the APL instruction is the same as the TIO. If the device condition specified by the N field (Q byte bits 5, 6, and 7) exists on machines *with* a dual program feature, the program advances to the next program level. If the device conditions are not met, the operation is equivalent to a no-op and no program advance level occurs.

If the device condition specified by the N field exists on machines *without* the dual program feature, the program continues to execute the APL instruction until the condition no longer exists. The program proceeds to the next sequential instruction. If the Q byte equals zero and the CPU is in dual program mode, an advance program level operation is unconditionally performed. The unconditional advance is equivalent to a no-op.

Sense I/O (SNS) Instruction

The SNS instruction stores two sense bytes from the MFCU attachment or the MFCU representing status conditions in the MFCU attachment and the MFCU. For example, the stored program can examine the status bytes to determine the type of check condition. From the information compiled from the status bytes, the action to be taken is then determined by the program or by the operator. The SNS instruction also stores the contents of the MFCU LSRs.

Figure 1-6 shows the SNS instruction and Figure 1-7 shows the status byte indicators.

Special indicator bytes are available as diagnostic aids to be used in troubleshooting the MFCU and the attachment logic. These special indicator bytes (Figure 1-8) are used by the timing analysis program to provide a printed output showing circuit timing relationships in the attachment and the MFCU. To store the special indicator bytes, the N field must equal 000 or 001 (Figure 1-6).

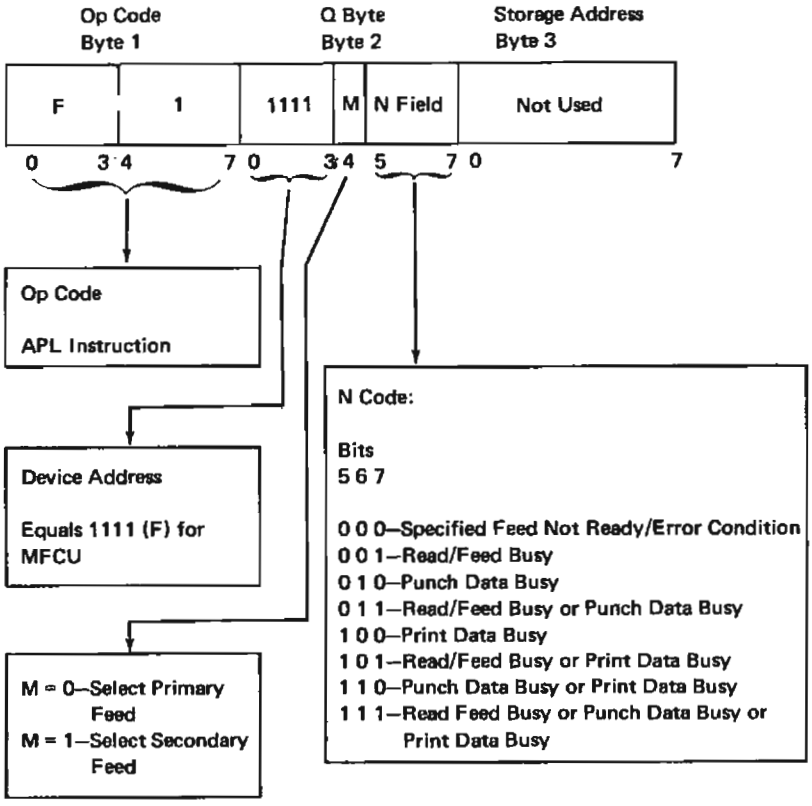


Figure 1-5. MFCU APL Instruction Format

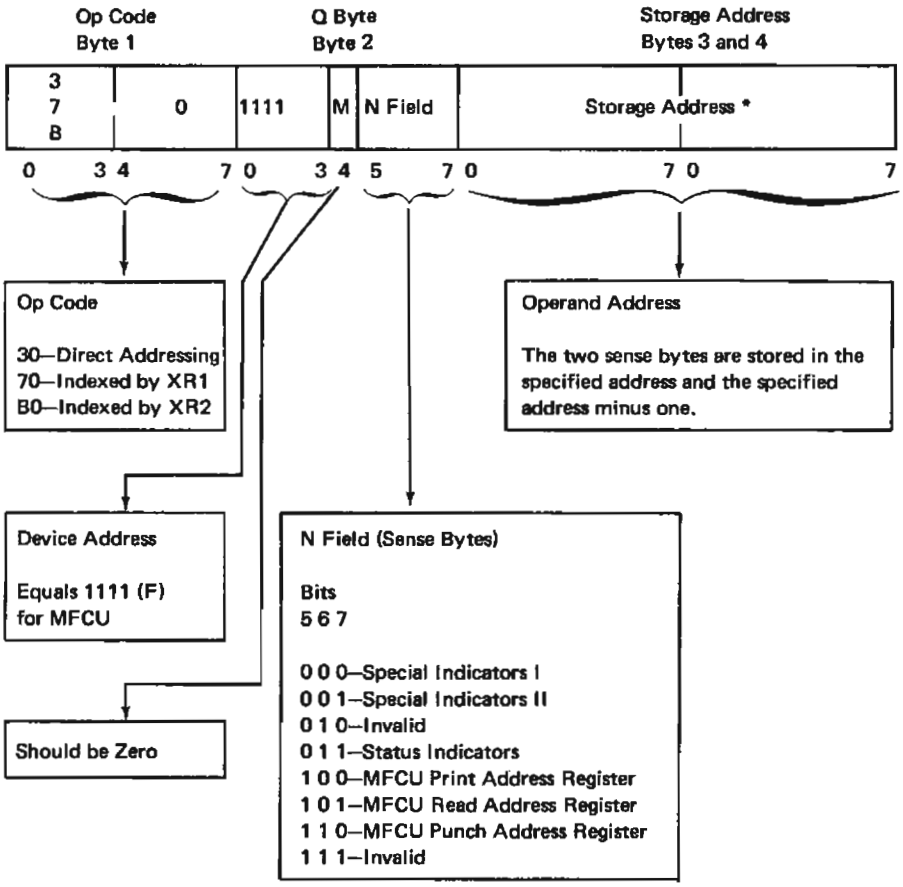


Figure 1-6. MFCU SNS Instruction Format

N Code 011	Byte 1 High Core Address EB1	Byte 2 Low Core Address EB2
0	Read Check	Print Buffer 1 Busy
1	Punch Check	Print Buffer 2 Busy
2	Punch Invalid	Card in Wait 1
3	Print Data Check	Card in Wait 2
4	Print Clutch Check	Reserved
5	Hopper Check	Hopper Cycle not Complete
6	Feed Check	Card in Trans/Counter Bit 2
7	NO-OP	Card in Trans/Counter Bit 1

Figure 1-7. MFCU Status Byte Indicators

N Code 000	Byte 1 High Core Address EB1	Byte 2 Low Core Address EB2
0	Hopper 1 or 2 Magnet	Punch CB
1	Hopper Cell	Punch Strobe
2	Gear Count 1, 3, 5, 7, 9, 11	Punch Magnet One
3	Read Cell One	Spare CE Bit
4	Read Cell 18	Print Time
5	Allow Read	Print Fire CB
6	Hopper CB	Print Magnet One
7	Spare CE Bit	Spare CE Bit

N Code 001	Byte 1 High Core Address EB1	Byte 2 Low Core Address EB2
0	Punch Registration Roll 1 or 2	Corner Kick Magnet
1	Pre-punch Cell	Print Stepper Clutch Magnet Hold
2	Punch Gate Magnet	Post-print Cell
3	Punch Eject Roll Magnet	Print Inject CB
4	Punch Stepper Roll Magnet	Print Kick CB
5	Corner Cell	Print Step CB
6	Punch Stepper CB	Print Allow Punch
7	Spare CE Bit	Spare CE Bit

Note: These indicators are intended for troubleshooting aids only. The special indicator bytes are listed here for reference purposes.

Figure 1-8. MFCU Special Indicators

Start I/O (SIO) Instruction

The SIO instruction command to the MFCU starts an MFCU operation. Figure 1-9 defines the instruction bytes and the control codes.

Three bytes make up the SIO instruction. Byte one (op code) equals F3 (hexadecimal) to indicate an MFCU SIO instruction. Byte two (Q byte) contains the device address (DA), a primary or secondary card feed path modifier (M) bit, and a command code (N field). Byte three (control code) is used to define the following functions:

1. Select the print buffer to be used.
2. Select initial program load (IPL) mode.
3. Select three or four lines of print.
4. Select a stacker pocket.

The functions performed by the MFCU are read, feed, punch, print, and selective stacking. Any combination of these four functions can be specified in the same SIO instruction. For example, a card can be punched, printed, stacked, and a new card read and placed in the vacated wait station. A new card is always fed into the vacated wait station as a result of the SIO instruction. The card may be read or just fed as specified by the SIO instruction N code.

Theory of Reading

- The SIO instruction read command starts the read operation.
- If bits 5, 6, and 7 of the Q byte equals 001, a read operation is performed.
- The M bit selects the primary or secondary feed.
- The reading operation begins after the leading edge of the card covers the read cells.
- Data read from the card is sent to the CPU over DBI and stored in core storage at the address specified by the MFCU read data address register (MRDAR). The MRDAR is loaded during the LIO instruction prior to every SIO read command.

The SIO instruction starts the mechanical operation of the MFCU. The read operation is determined by decoding bits 5, 6, and 7 of the Q byte (equals 001 for a read operation). Bit 4 of the Q byte selects the primary or secondary card path. The card in the selected wait station is ejected and routed to the stacker pocket selected by the current SIO control code.

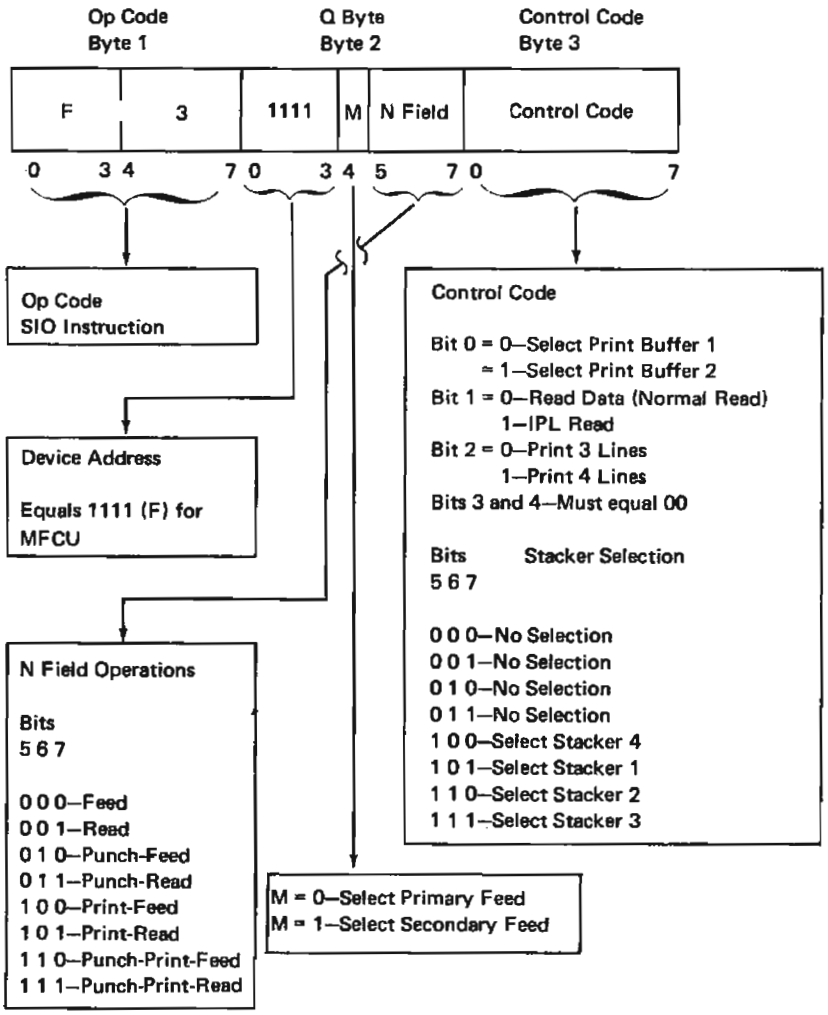


Figure 1-9. MFCU SIO Instruction Format

The attachment unit waits until it receives a hopper CB pulse from the MFCU to select and feed a card from the selected hopper. The hopper magnet is then energized and a card is fed from the selected hopper into the read inject station. The card is then sent to the read station by the read inject feed roll where the card is read.

The read station has 18 photoelectric read cells. When the leading edge of the card covers the read cells, a counter is started to control when the reading of card data is to begin. After the signal to start reading occurs, the counter controls reading and transferring of data to CPU core storage.

The first card column group read by the 18 read cells are columns 1, 33, and 65. A read pulse gates these 18 bits of data to a read register. The next read pulse gates the same group of 18 bits to a read check register. A total of three read pulses per register attempts to gate coincident read cell data. Only one pulse per register is needed to set the data in these registers. The first read cell pulse sets a bit position if a read cell senses a punch in the card. Two additional gated read pulses per register attempt to gate data into the registers. Three attempts per register for each card column group improve reading accuracy if off punching should exist.

The read register bits are then gated to the data bus in (DBI) assembler. Odd parity is assigned to the data byte and the data is then stored in the core storage address specified by the MRDAR. The outputs from these two registers are then compared. If the register outputs do not compare, a read check occurs.

Three data bytes must be sent to the CPU for each group of three columns read. After the first byte is stored, 32 is added to the LSR read address register so that column 33 can be stored in the correct core storage location. The third data byte is stored 32 storage locations above the second

data byte. The LSR address register is then decremented by 63 so that columns 2, 35, and 66 can be read and stored in the correct core storage locations. Data is read serially, three columns at a time, until all card columns are read and stored.

The card being read moves to the selected wait station. The card remains in the wait station until another SIO instruction selects the same wait station. The card that was ejected from the selected wait station is routed to a stacker pocket.

Theory of Punching

- The SIO instruction punch command starts the punch operation.
- An SIO punch command refers to the card in the selected wait station.
- 010 in the Q byte N field indicates a punch-feed operation.
- The M bit selects the primary or secondary card feed.
- Cards are punched serially column by column beginning with columns 1, 33, and 65.
- Data punched in the card generates check signals that are sent to the CPU for checking.

The SIO instruction starts the mechanical operation of the MFCU. The punch operation is determined by decoding bits 5, 6, and 7 of the Q byte (equals 010 for a punch-feed operation). Bit 4 of the Q byte selects the primary or secondary feed path.

To begin the punch operation, the card is fed from the selected wait station. The card is registered against the punch registration gate. Then the punch stepper pressure roll magnet is de-energized, the punch registration gate is de-energized, and the card is incremented and punched column by column through the punch station.

The punch station has 18 punch positions (six punch positions per card tier). The first card columns punched by the 18 punch positions are columns 1, 33, and 65. Three separate data bytes are sent from the CPU to the attachment. CB pulses are used to synchronize the data with the punch magnets.

After the punch data bytes are assembled, the punch magnets are energized if the card is to be punched in a specified position. A 6-bit character is punched into each column of the first column group. As the data is punched in the card, 18 piezo-electric crystals sense the punches that are activated. These check signals are gated into punch check latches located in the MFCU. This check data is assembled into three data bytes and sent to the CPU where the check data from the card is compared with the data that was sent to be punched. Checking occurs just prior to receiving data to be punched in the next card column group.

Three data bytes must be sent to the punch circuitry before a punch cycle is taken. The first data byte (column 1) is sent to the attachment over data bus out (DBO). Thirty two is then added to the LSR punch address register so that data for column 33 can be sent from the correct core storage location to the attachment. Thirty two is again added to the LSR punch data address register for the third data byte for card column 65. After the third data byte is sent to the attachment, the LSR punch data address register is decremented by 64 so that columns 1, 33, and 65 can be checked prior to punching the next column group. After checking is completed, and just before receiving data for the next card column group, the LSR address register is modified so that the next data group can be sent to the attachment for punching columns 2, 34, and 66. In this case 63 is subtracted from the LSR address register.

Data is punched and checked serially, three columns at a time, until all card columns are punched and checked.

After the card is punched, the card is ejected to the corner station. If the SIO instruction indicated a punch-print operation, the card is printed before it is routed to the selected stacker pocket.

A card feed or a card read also occurs as a result of a punch command. The card from the selected hopper is sent to the vacated wait station.

Theory of Printing

- The SIO instruction starts the print operation.
- If bits 5, 6, and 7 of the Q byte equal 100, a print-feed operation is performed.
- The M bit selects the primary or secondary card feed.
- Bit 2 of control code specifies three or four lines of print.
- Printing begins after the card is fed from the corner station and advanced to print line one position.
- The card is printed and fed through the print station in a parallel direction.
- A side motion carriage is shifted to allow printing 32 print positions per print line.

The SIO instruction starts the mechanical operation of the MFCU. The print operation is determined by decoding bits 5, 6, and 7 of the Q byte. Bit 4 selects the primary or secondary card feed path (Figure 1-9).

To begin an operation, the card is ejected from the selected wait station. If a punch operation is not specified, the card passes directly through the punch station to the corner station. From the corner station, the corner kicker registers the card against the print stepper feed roll which is stopped at this time. The stepper clutch magnet then de-energizes and the card moves to print line one position. Line one is now ready to be printed.

CB pulses are sent from the MFCU to the attachment unit to synchronize electrical-mechanical operations. These CB pulses are also used to step counters in the attachment unit to control the print operation.

After the card is positioned at line one, the print operation begins. There are 16 print typewheels (Model A2; 8 typewheels in Model A1) in the MFCU that rotate on a common shaft. A print character counter in the attachment is used to determine the character that is ready to be printed.

The contents of the print character counter are sent to the CPU over DBI during each print cycle steal. The print character count is then compared with the data from each addressed core storage position to determine whether or not the character is to be printed by one or more of the typewheels. The comparison is done by the arithmetic logical unit (ALU). The result of the comparison is sent to the attachment over DBO to the print shift register. As each compare or no compare signal is generated and sent to the print shift register, the data is shifted one shift register position. After 16 (Model A2) comparisons, the print shift register contents are gated to a print register. The print register is gated to print the positions that have been selected as the result of the comparisons. The print fire CB pulse causes the print character counter to be advanced one count. A new group of 16 comparisons (Model A2) is made for the next print character count.

Note that there are 16 comparisons made for each print character counter value. There are 64 characters counted for each revolution of the typewheels. After one typewheel revolution, all characters on the 16 typewheels have been optioned for printing. At the completion of the one typewheel revolution, one-half line is printed (Model A2). The side motion carriage then shifts so that 16 more characters can be printed on line one. Remember that there are 16 comparisons for each character and one typewheel revolution is necessary to option all 64 characters per print position for each line. After the second group of characters is printed, line one printing is completed. (Three side motion carriage shifts are necessary for eight typewheel machines.) The side motion carriage returns the card to the starting position. Also, the print stepper clutch advances the card to print line two for the next group of print cycles.

The printer can print three or four lines per card. If bit 2 of the control code is 1, four lines are printed. After printing is completed, the card is sent to a selected stacker pocket.

During the print operation, a card from the selected card path hopper is fed into the vacated wait station. Reading of the next card takes place if specified by the N code of the print command.

Theory of Stacking

- Stacker selection in the SIO instruction pertains to the card in the wait station.
- Cards from the primary wait station are routed to stacker pocket one unless the SIO control code specifies the stacker pocket.
- Cards from the secondary wait station are routed to stacker pocket four unless the SIO control code specifies the stacker pocket.

The MFCU operations provide selective stacking of cards. Selective stacking is controlled by the program. Stacker pocket selection is indicated by bits 5, 6, and 7 of the control code in the SIO instruction (Figure 1-9). The selection contained in the control code always pertains to the card in the selected wait station. If no stacker control code is indicated, the cards from the primary wait station are routed to stacker pocket one; cards from the secondary wait station are routed to stacker pocket four.

During a non-process-run-out (NPRO) operation, cards from both feeds are routed to stacker pocket one.

MFCU CLOCK

The MFCU clock contains an oscillator divider, a clock, and a clock decoder (Figure 2-1).

The oscillator divider contains three flip flops that increase the 160 nanosecond input signal to a 1.28 microsecond output signal. This 1.28 microsecond output signal is used to step the MFCU clock. The output from the two flip flop MFCU clock is then decoded by the clock decoder.

The clock decoder generates four MFCU clock signals:

1. 'MFCU clock A'
2. 'MFCU clock B'
3. 'MFCU clock C'
4. 'MFCU clock D'

These clock signals are used by the MFCU attachment to synchronize electrical-mechanical operations.

See FEMD 4-005 for a circuit description and a timing chart of the MFCU clock.

DATA BUS OUT REGISTER

Eight data bits and one parity bit are transferred to the MFCU attachment unit by the data bus out (DBO) lines. These data bits are sent to the DBO parity check circuits and to the eight data bus out register latches (Figure 2-2).

The parity check circuits check each data byte sent to the MFCU attachment for odd parity.

Data sent over DBO is set into the DBO register latches by the CPU 'clock 5' signal. The next CPU 'clock 5' signal either gates new data into the register or resets the register.

Outputs from the DBO register are used:

1. During the I-Q cycle to select the MFCU and to define the function. Bits 4, 5, 6, and 7 are sent to the I-Q register.
2. During the I-R cycle to select a print buffer, to indicate an IPL read mode, to indicate the number of lines to be printed (three or four), and to select a stacker pocket.
3. During I/O cycle execution to condition punch magnet and print magnet circuits.

See FEMD 4-010 for a circuit description of the data bus out register and the parity check circuits.

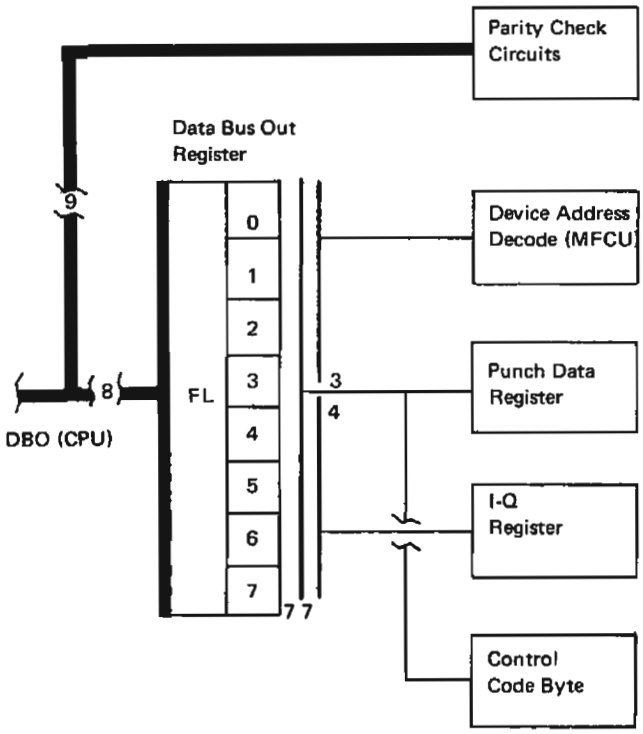


Figure 2-2. Data Bus Out Register

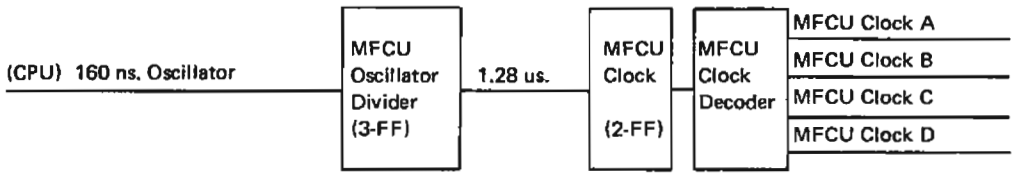


Figure 2-1. MFCU Clock

I-Q REGISTER

The I-Q register is a four bit register that is loaded during the I-Q cycle of an MFCU instruction. During the MFCU instruction I-Q cycle, bits 4, 5, 6, and 7 are gated to the I-Q register by a CPU 'I-Q clock 5C' signal (Figure 2-3).

The function to be performed by the MFCU is defined by decoding the output of the I-Q register. The functions performed depend upon the programmed I/O instruction. See Chapter 3 for command execution.

See FEMD 4-022 for a description of the I-Q register.

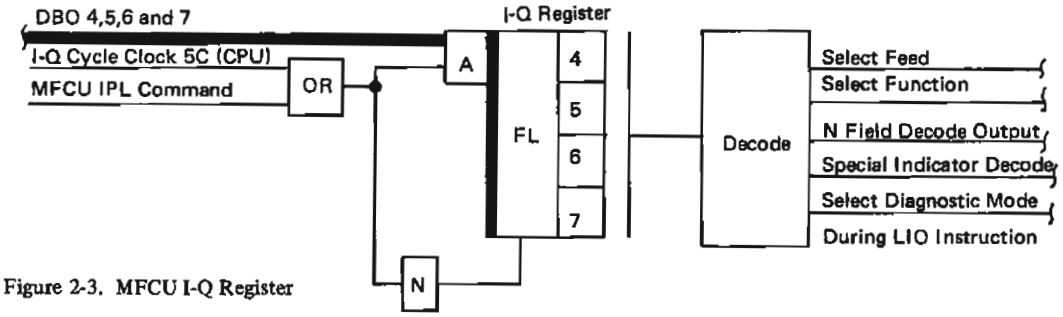


Figure 2-3. MFCU I-Q Register

READ STATION FUNCTIONAL UNITS

Read Gear Emitter Counter and Decoder

The read gear emitter counter is a 10 position binary counter (Figure 2-4). This counter, along with a decoder and other read control circuits, controls reading 96 columns of card data and sends the card data one byte at a time to the CPU.

The counter is normally reset until the leading edge of the card covers the read station photoelectric read cells. After the leading edge covers the read cells, the counter is stepped by read gear emitter CB pulses from the MFCU.

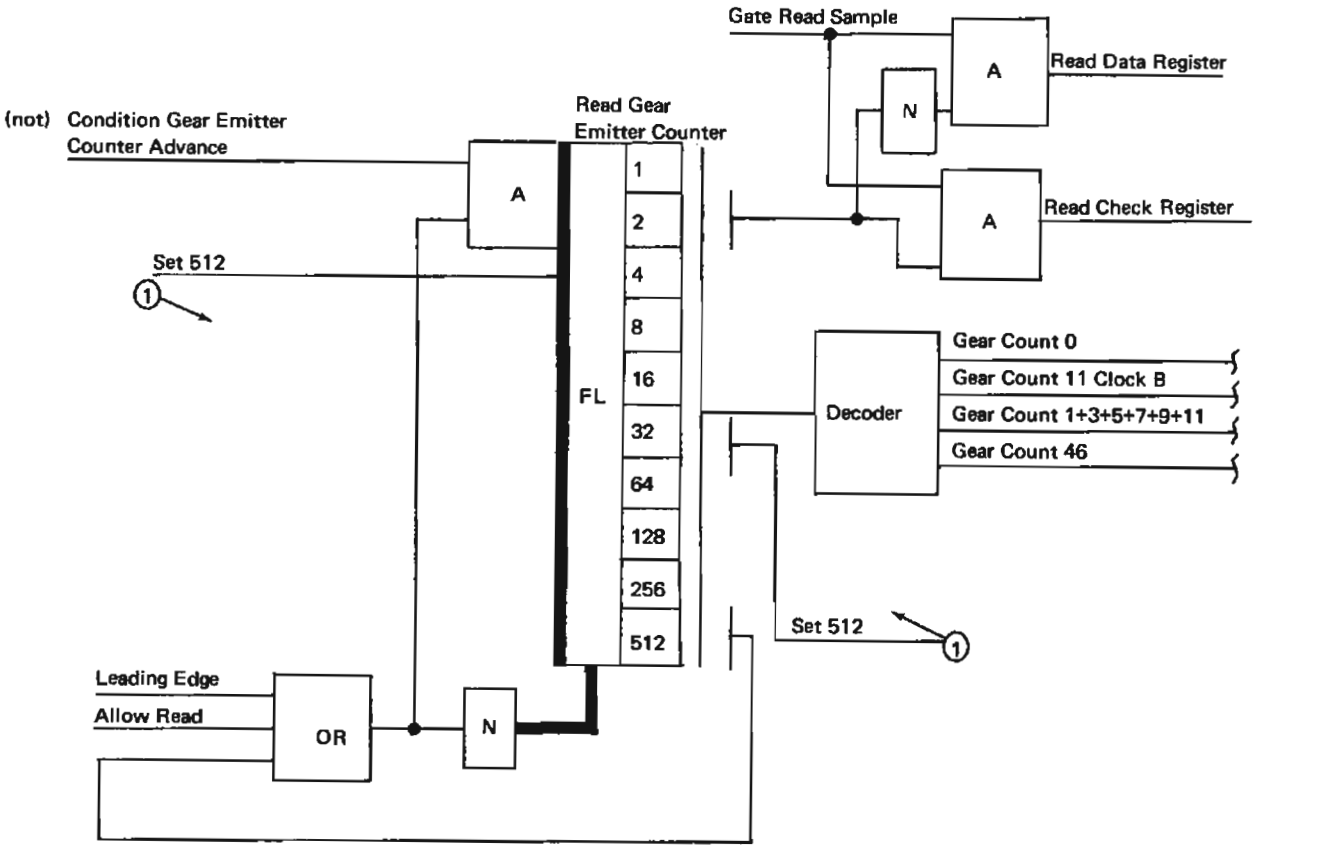


Figure 2-4. Read Gear Emitter Counter and Decoder

The counter then counts from zero to 46, resets and starts counting again. At count 46, card columns 1, 33, and 65 are ready to be read. Counts of 1, 5, and 9 gate data into the read data register (Figure 2-4). Counts 3, 7, and 11 gate data into the read check register. During count 11 read cycle steal request signals are sent to the CPU to start data transfer.

The counter counts consecutively after the reset at count 46. When the counter reaches count 528, the read operation ending sequence completes the read function. For each card column group read, the counter decoder generates counts of 1, 3, 5, 7, 9, and 11. These counts read data from the card and start data transfer to the CPU.

See FEMD 4-027 for a description of the read gear emitter counter and decoder. Refer to Chapter 3 for a read operation description.

Read Register and Read Check Register

The read register and the read check register contain 18 latches each. Data is gated into these registers under control of the read gear emitter counter (Figure 2-5). After 18 bits of data are read from the three card tiers (columns 1, 33, and 65, or 2, 34, and 66 etc.), the data from the read register is sent to the data bus in assembler in three groups (six bits per group). An eight bit byte (with parity) is then sent to the CPU and stored in core storage.

Outputs from the read register and the read check register are compared bit by bit. If a bit from the read register and the corresponding bit from the read check register do not compare, a read check is indicated.

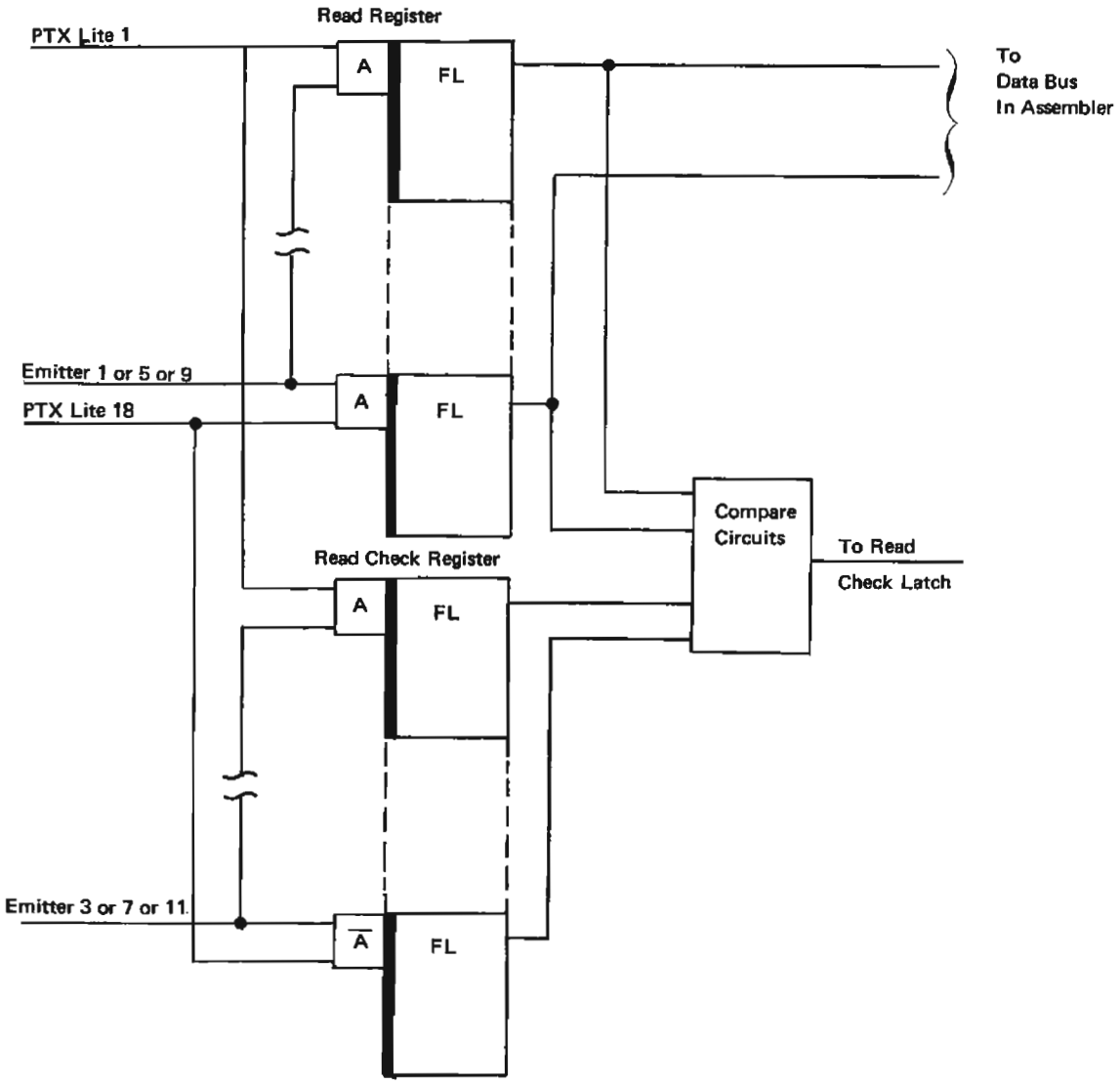


Figure 2-5. Read Register and Read Check Register

A normal read operation reads a 6-bit card code from each tier. The C and D positions are normally zero. Three separate data bytes are sent to the CPU for each card column group.

Tier three positions 8, 4, 2, and 1 are used to make up the C and D positions in tiers 1 and 2 for an initial program load (IPL) operation or a read command in IPL mode. Tier three data sent to the CPU during an IPL operation is usually meaningless.

See FEMD 5-125 for a description of the read register, the read check register, and compare circuits. See Chapter 3 for a description of the read operation and the IPL read operation.

DATA BUS IN ASSEMBLER

The data bus in assembler has eight latches that divide the read register data into three separate data bytes (Figure 2-6). This data is then stored in the CPU core storage.

Data is gated into the data bus in assembler from:

1. The read data register latches and the read check register latches.
2. The sense byte indicators.
3. The punch check data from the punch check crystals and latches located in the MFCU are sent to the CPU. This data is used to compare the data actually punched in the card with the data sent to the attachment from core storage.
4. The print character counter.
5. The LSR modification data.

During a normal read operation, bit positions D and C of the data bus in assembler both equal zero. During IPL operation mode, eight bits for tiers one and two are sent to the CPU. The D and C bits for tier one are the 8 and 4 bits of tier three; the D and C bits for tier two are the 2 and 1 bits of tier three. Data sent to the CPU from tier three is usually meaningless.

See FEMD 4-020 for a description of the data bus in assembler.

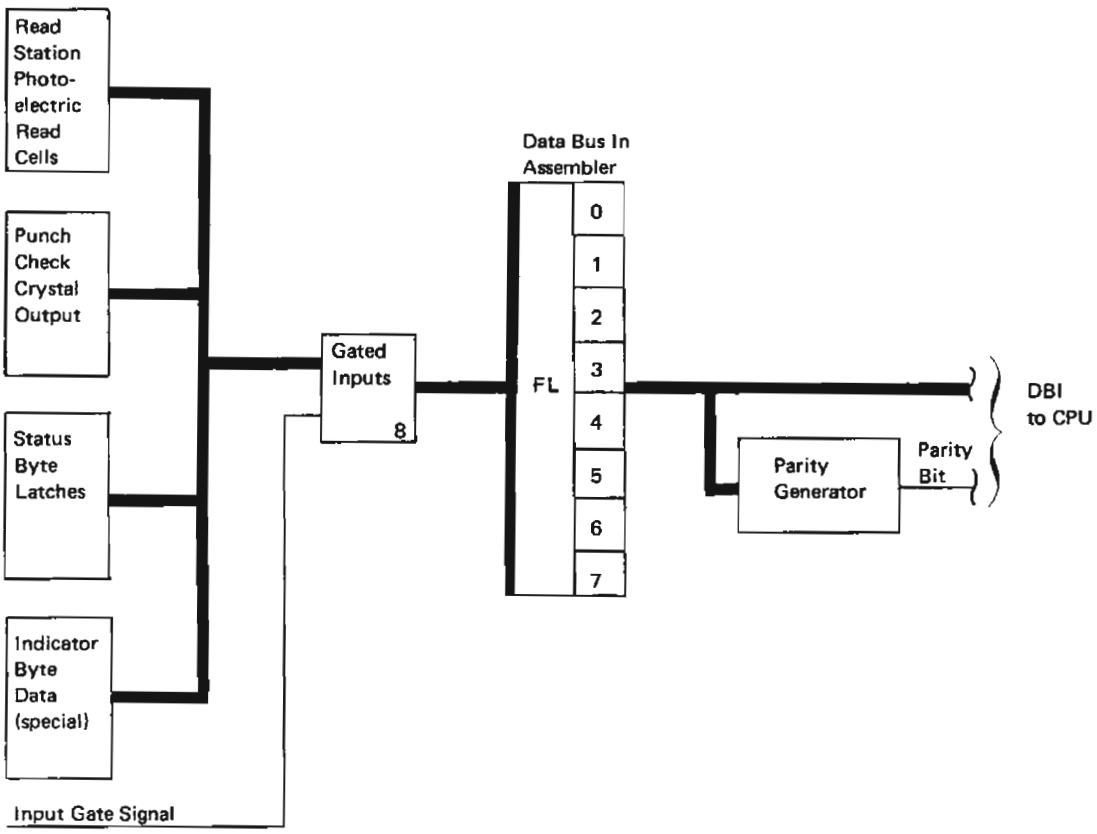


Figure 2-6. Data Bus In Assembler

PUNCH FUNCTIONAL UNITS

Punch Step CB Counter and Decoder

The punch step CB counter is a six stage binary counter that is used to control punching data into a card (Figure 2-7).

This counter is reset before a punch function by an 'MFCU clock D' signal and a punch not ready condition. The counter is stepped during a 'punch step CB' pulse (from the MFCU punch unit) by an MFCU clock B signal.

Outputs from the counter are then decoded and used to control the punch operation. The decode circuits are degated whenever the punch CB counter is being stepped.

See FEMD 4-035 for a description of the punch step CB counter and decoder. Refer to Chapter 3 for a punch operation description.

Punch CB Counter and Decode

This two stage binary punch CB counter and decode activates the punch pick, the punch hold, the punch de-energize, and the punch check circuits (Figure 2-8)

Each 'punch step CB' pulse from the MFCU resets the counter. The counter is stepped by the 'condition punch CB control advance' signal during MFCU clock A time four times per card column group (three tiers) punched.

The outputs from the decoder determine when the punch magnets are to be energized and de-energized. Also, the counter gates the punch magnet hold current.

Checking of a column group that was punched occurs at the beginning of the next card column group to be punched.

See FEMD 4-030 for a description of the punch CB counter and decode circuits.

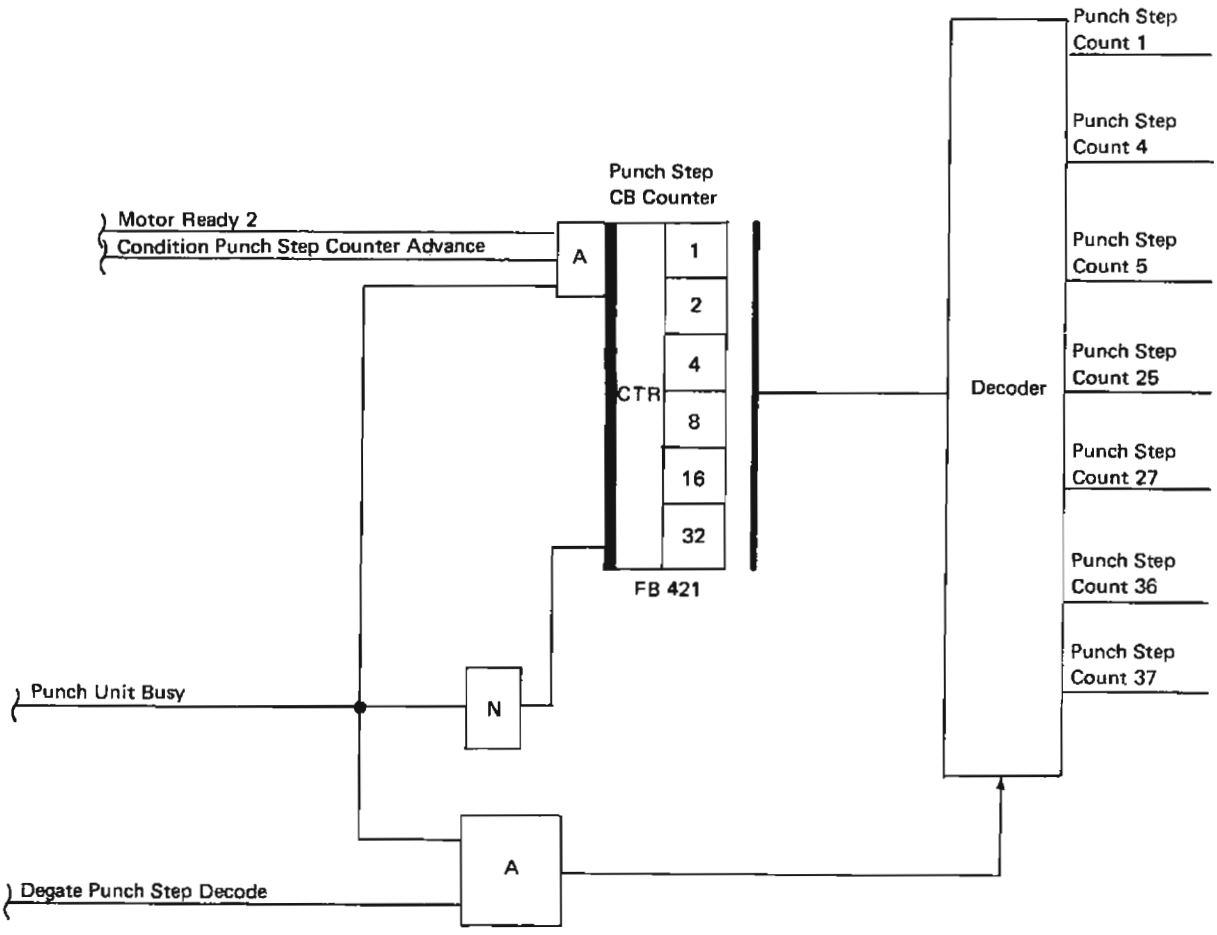
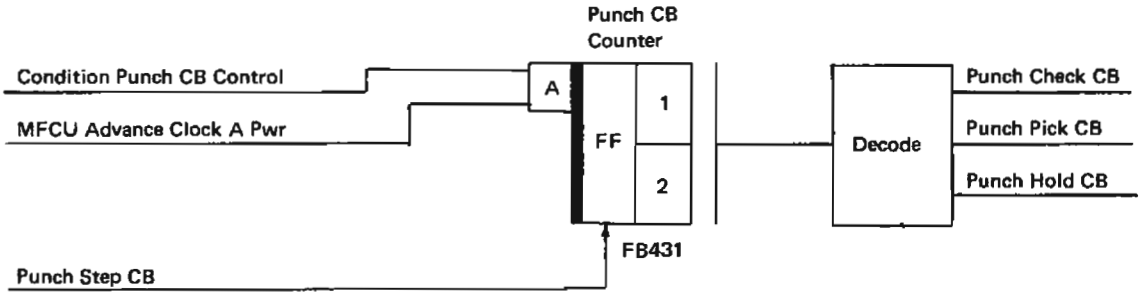


Figure 2-7. Punch Step CB Counter and Decoder



FF 1	FF 2	Action
Off	Off	Punch Check CB
Off	On	Not Used
On	On	Punch Hold CB
On	Off	Punch Pick CB

Figure 2-8. Punch CB Counter and Decode

FEED CHECK TIME BASE COUNTER

The feed check time base counter is a six stage binary counter that counts from zero to 63, resets and begins counting from zero. This counter is stepped every 80 microseconds for model A1 machines (40 microseconds for model A2 machines) by the 'condition gear emitter counter advance' signal (Figure 2-9).

Counter positions 1 and 2 feed decode circuits to generate three feed check control levels:

1. Feed check control 1.
2. Feed check control 2.
3. Feed check control 3.

These control signals are used along with the feed check time base signals to indicate card progression through the transport. The feed check control levels are used to control and gate various feed check circuits.

The decode circuits are degated whenever the counter is being stepped. Counter positions 4, 8, 16, and 32 feed the feed check time base decode circuits to generate five feed check time base signals. These time base signals are used to drive the various feed check counters (hopper, wait, corner, and stacker) for feed checking.

See FEMD 4-100 for a description of the feed check time base counter and decode circuits.

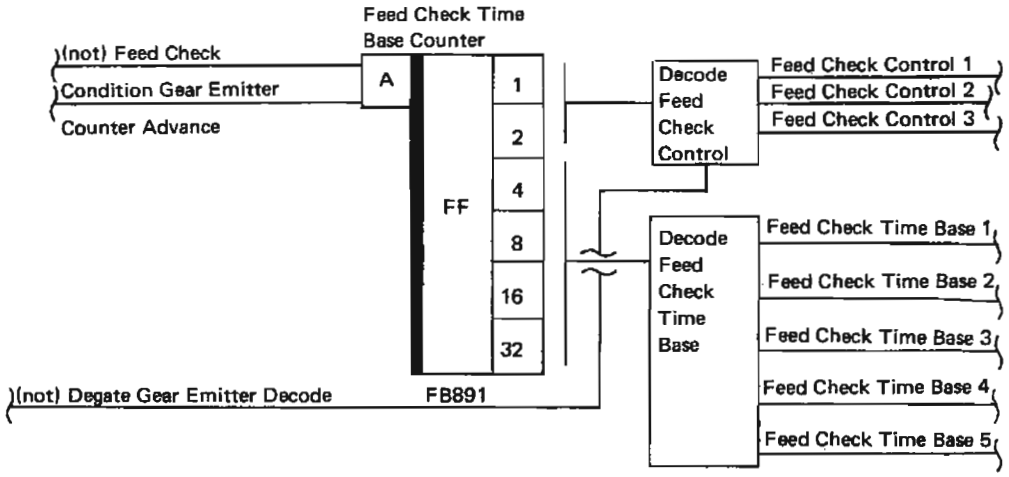
CORNER KICK COUNTER

The corner kick counter is a three stage binary counter that checks card entry into the print station. This counter is held reset until a card covers the corner station card cell. This counter also provides a delay so that the card can be registered in the corner station before energizing the corner kick magnet.

The counter is stepped every 2.56 milliseconds (1.28 milliseconds for model A2 machines) by a 'feed check time base 4' signal (Figure 2-10).

When the counter reaches a count of six, the corner kick and the corner kick aligner magnets are energized. The card then enters the print station side motion feed rolls. The corner kick counter is then reset after the corner kick signals are deactivated.

See FEMD 4-090 for a description of the corner kick counter. See Chapter 3 for a print operation description.



Notes:
1. Model A1 machines—Counter stepped every 80 microseconds.
2. Model A2 machines—Counter stepped every 40 microseconds.

Legend

Feed Check Time Base	FF set for 8 hammer Machines	FF set for 16 hammer Machines	Time in Milliseconds Model 1	Model 2
Feed Check Time Base 1	8	16	1.28	1.28
Feed Check Time Base 2	16	32	2.56	2.56
Feed Check Time Base 3	32	32	5.12	2.56
Feed Check Time Base 4	16	16	2.56	1.28
Feed Check Time Base 5	4	8	.64	.64

Figure 2-9. Feed Check Time Base Counter

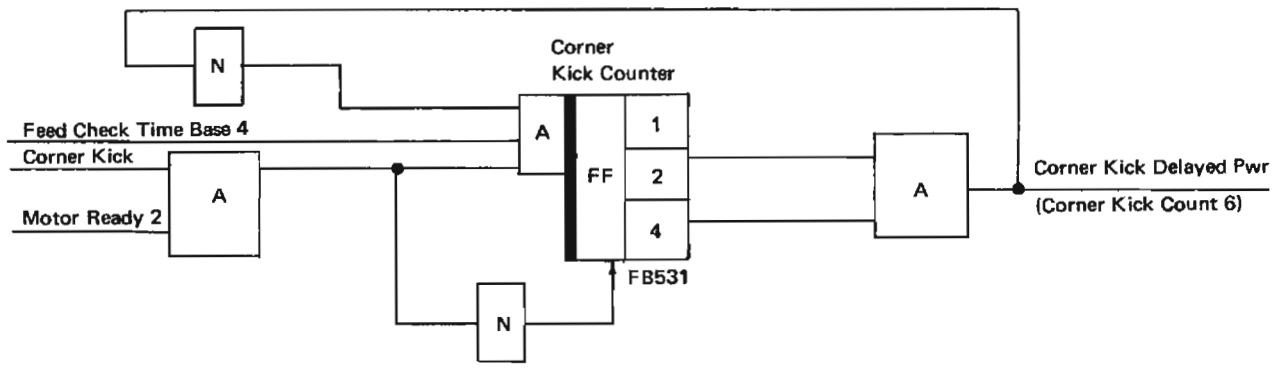


Figure 2-10. Corner Kick Counter

PRINT STATION FUNCTIONAL UNITS

Print Clock Counters

There are two print clock counters that control the print function and synchronize attachment operations with the MFCU mechanical operations. These counters are the low print clock counter and the high print clock counter. Each counter is a three stage counter (Figure 2-11).

The counter control inputs are the MFCU clock pulses, the print kick CB pulses, the print inject CB pulses, and a print mechanical busy signal.

The start conditions of the print clock counters are:

1. High print clock counter equals 111 (binary)
2. Low print clock counter equals 000 (binary).

After the print mechanical busy line is activated, the low print clock counter is stepped by each print kick CB pulse or a print inject CB pulse. The high print clock counter is stepped as a result of activating 'print clocks 4, or 10, or 16, or 22, or 28' line. The low print clock counter is reset each time the high print clock is advanced.

The ouputs of the print clocks are decoded and used by the print control circuits to control the print operation.

See FEMDs 4-055 and 4-060 for a description of the print clock counters. See Chapter 3 for a print operation description.

Print Side Motion Counter

The print side motion counter is a three stage binary counter (Figure 2-12). This counter is used to make sure that the side motion cam in the MFCU is at high dwell before energizing the print side motion cam magnet.

The counter is stepped by each 'feed check time base 2' signal after the 'print clock 4' level from the print clock counters is activated. When the print side motion counter equals six, the side motion magnet latch is set which in turn energizes the print side motion magnet. The counter is then reset at the next 'feed check time base 2' signal. The side motion magnet remains energized until the print operation is completed.

See FEMD 4-085 for a description of the print side motion counter.

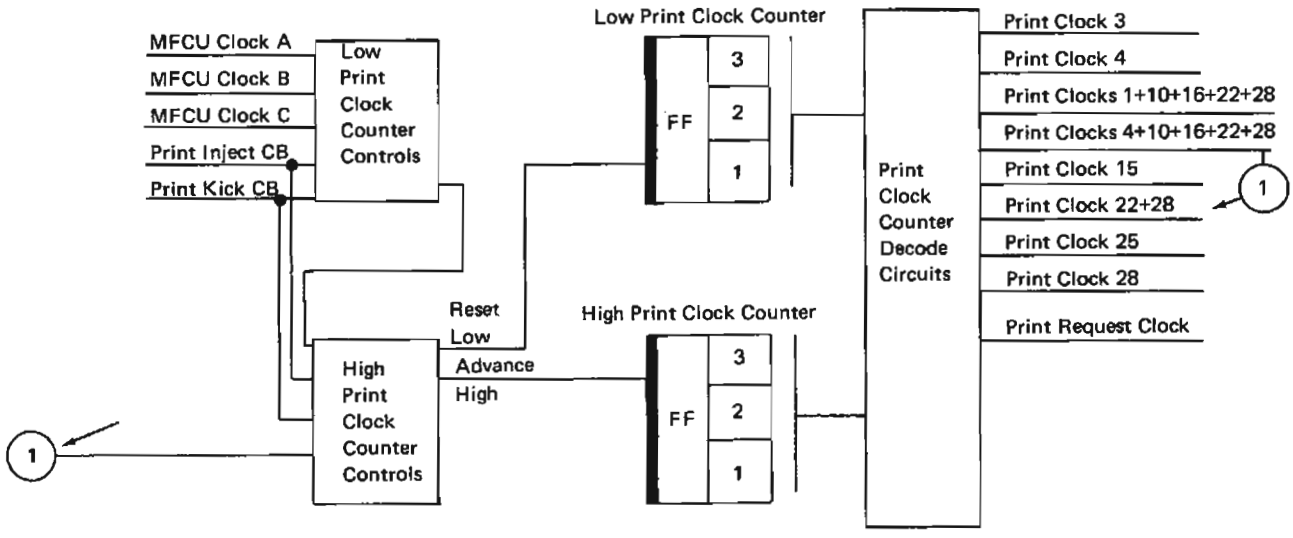


Figure 2-11. Print Clock Counters

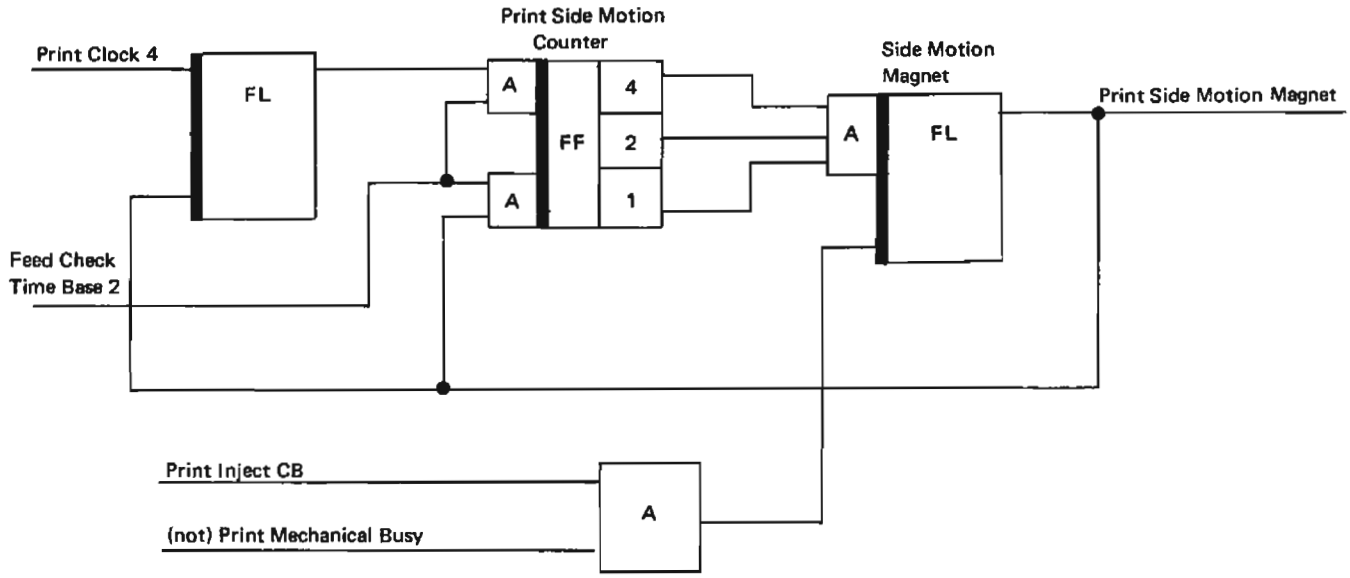


Figure 2-12. Print Side Motion Counter and Control

2

Print Character Counter

The print character counter is a six position binary counter (Figure 2-13). Each count value represents a 6-bit card character code.

The 'print data home CB' pulse from the MFCU results in resetting the print character counter indicating the home position of the print typewheels.

This counter is stepped after the 'print fire CB' pulse by the 'load print data rank 1' signal (Figure 2-13). The outputs from the counter are sent to the print data bus in assembler and then to the CPU where the count value is translated to extended binary coded decimal interchange code (EBCDIC) and then compared against the EBCDIC card image data for each print wheel. Sixteen (model A2) comparisons are made before the next 'print fire CB' pulse; eight comparisons are made on model A1 machines.

The decode counter outputs are used along wtih the 'home CB' signal to check that the counter is operating correctly.

SEE FEMD 4-080 for a description of the print character counter.

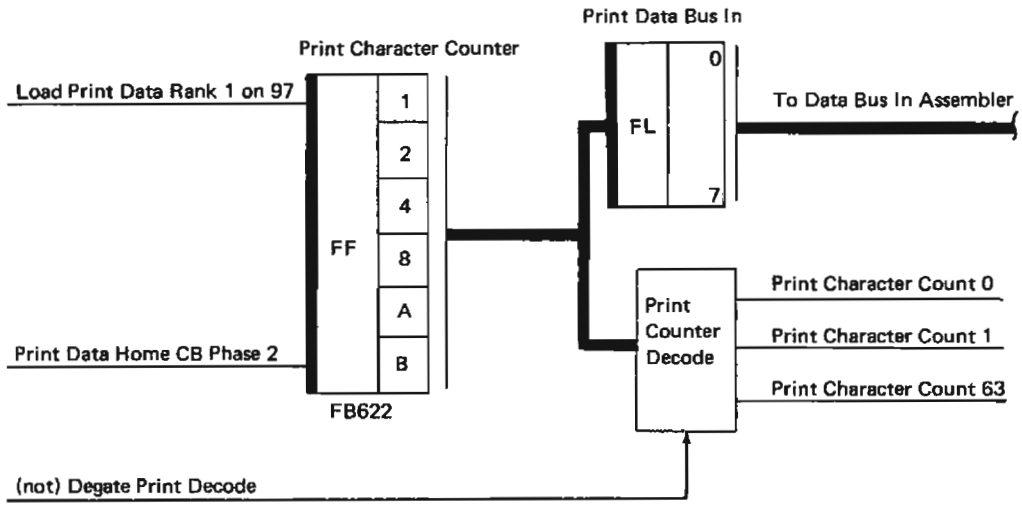


Figure 2-13. Print Character Counter

Print Data Counter

The print data counter is a three stage counter used (1) to control gating to the print data register bank 1, (2) to reset the print shift register and (3) to enable the load print data rank 2 counter.

The print data counter is a modified binary counter that is held reset until the MFCU sends a 'print fire CB' pulse to the attachment. The counter counts from 0 to 1, 2, and 4. A shift of the CPU channel 1 oscillator pulse sets the first counter latch. Setting this latch results in gating data to bank one of the print data register (FEMD 4-074). The load print data rank counter is also enabled by setting the 'gate load print 2' latch (FEMD 4-080).

After two more channel one oscillator pulses, the counter equals 4 resulting in resetting the print shift register. A shift in the next oscillator pulse resets the counter. The counter remains reset until the next 'print fire CB' pulse from the MFCU.

See FEMD 4-074 for a description of the print data counter.

Load Print Data Rank 2 Counter

The load print data rank 2 counter is a four stage binary counter that is used to reset bank 2 of the print data register. This counter is initially set to 0001 (binary).

The 'load print data rank 1' signal from the print data counter sets the 'gate load print 2' latch (FEMD 4-080). The counter is then stepped every 20.48 microseconds. When the load print data rank 2 counter equals count 10, the 'inhibit counter advance' latch sets (Figure 2-14). Setting this latch resets the 'gate load print 2' latch. The 'load print data rank 2' signal then resets bank 2 of the print data register at MFCU clock A time. At MFCU clock D time, the counter is again reset to 0001 (binary).

See FEMD 4-075 for a description of the load print data rank 2 counter. See "Print Unit Operation Timing" discussed under the "Print/Feed Operation" in Chapter 3 for timing considerations utilizing the print data counter and the load print data rank 2 counter.

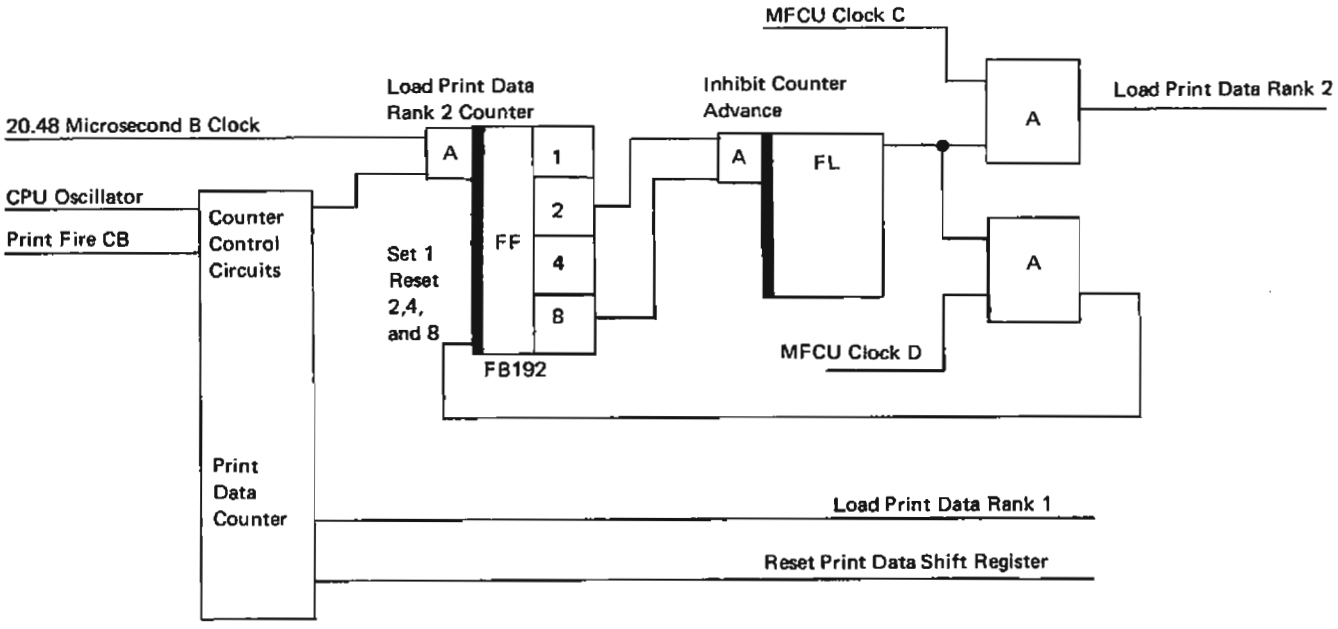


Figure 2-14. Load Print Data Rank 2 Counter

Print Shift Register

The print shift register has 16 bit positions (model A2 machines). On model A1 machines the print shift register has 8 bit positions. The character count that represents a specific character on the 16 (or eight) typewheels is compared with each corresponding character code stored in core storage. Sixteen (or eight) comparisons are made for each character count value. (Note that the same character is ready to be printed by each typewheel for each count value.) As the comparisons are made, the compare result is sent to the attachment over DBO. The 'gated print compare' (or no compare) is gated into the print shift register. Each 'load print sample 2' and 'load print sample 1' signals shifts the data in the shift register one bit position (Figure 2-15).

After 16 (or eight) comparisons are completed, the data is gated into bank 2 of the print data register buffer. From the buffer the data is set directly in the print data register. The set always overrides the reset. The print data register output positions are then gated by the 'print time switch' signal to energize the print magnet hammers. After printing of the first character group, the print data register and the shift register are reset. The logic circuits are ready to receive comparison signals from the next character count sent to the CPU from the print character counter.

See FEMDs 4-065 and 4-067 for a description of the print shift register.

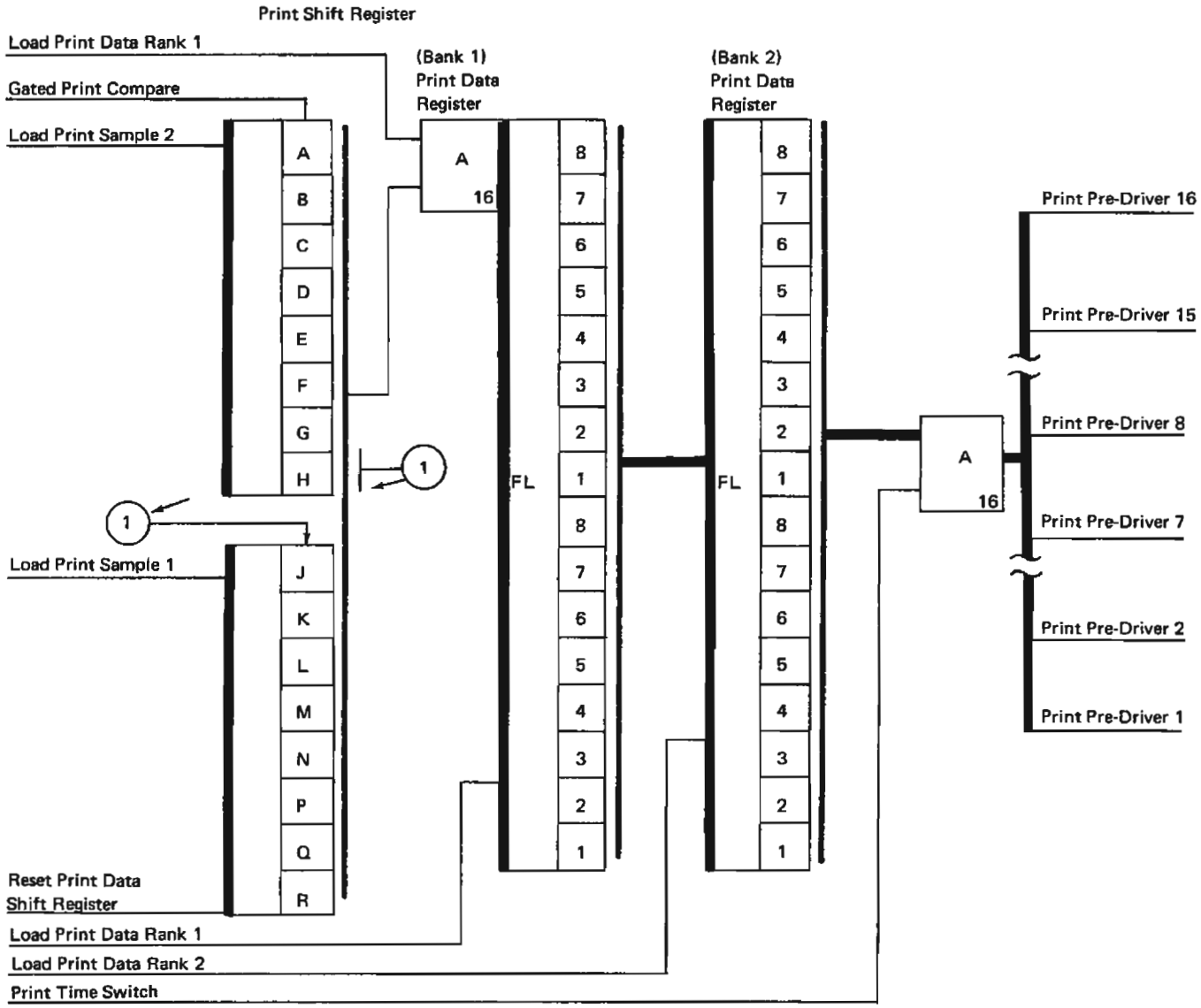


Figure 2-15. Print Shift Register and Print Data Registers

Print Data Registers

Two 16-bit (or two 8-bit) registers make up the print data registers. The print data buffer register receives data from the shift register (Figure 2-15). Data is gated into the print data buffer register by the 'load print data rank 1' signal. Data is transferred (without gating) to the print data register (the set overrides the reset). The outputs from the print data register are gated by the 'print time switch' signal to energize the selected print hammers optioned during the 16 (or eight) character count comparisons. After printing, the print data register is reset by a 'load print data rank 2' signal. See "Print Operation" in Chapter 3 for a description of the print operation.

See FEMDs 4-070 and 4-072 for a description of the print data registers.

Print Cycle Steal Request Counter

The print cycle steal request counter is a seven position binary counter (Figure 2-16). This counter is stepped after each 'print fire CB' pulse by the 'set print CS request' signal (Figure 2-16).

The counter counts to 64; thereby, indicating one complete revolution of the print typewheels. After the count of 64, all characters on all 16 (Model A2 machines) typewheels have been optioned for printing. The next operation may be a carriage shift, or an advance to the next line. Also, the print operation may be complete at this time.

Refer to Chapter 3 for a description of the print operation. See FEMD 4-045 for a description of the print cycle steal request counter.

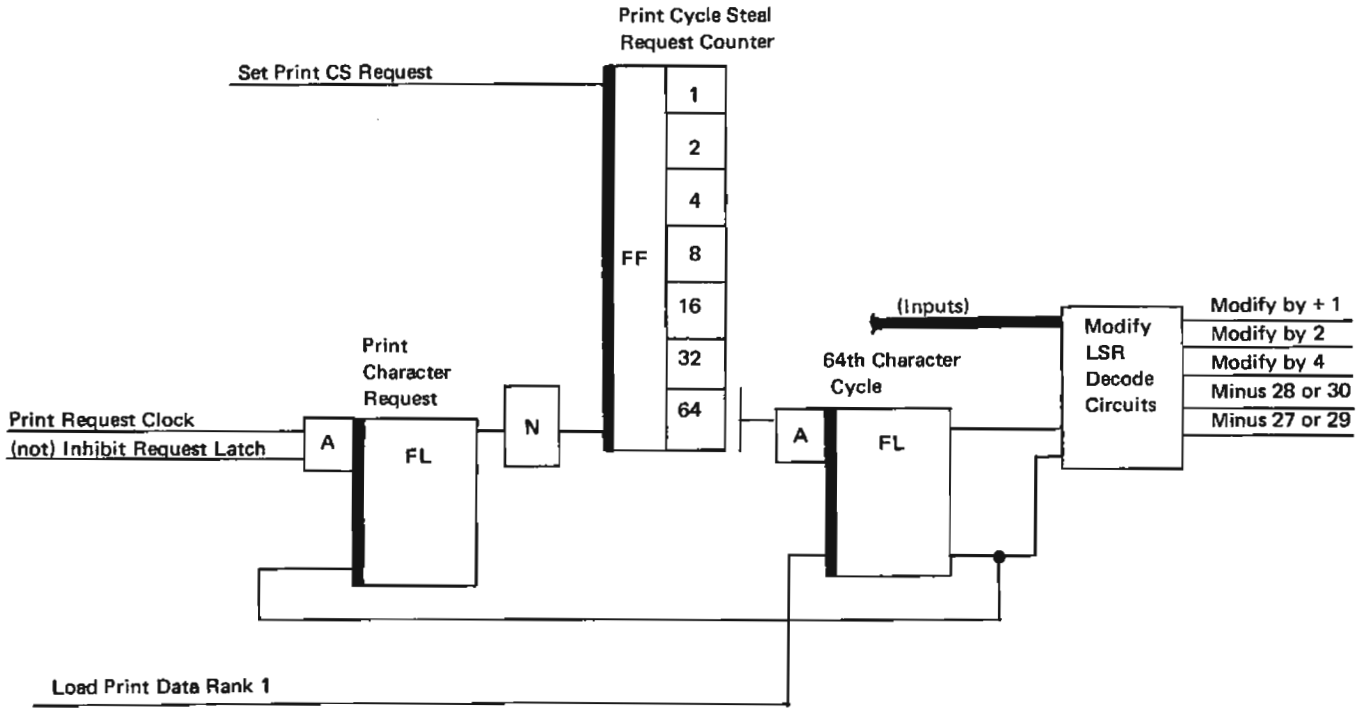


Figure 2-16. Print Cycle Steal Request Counter

Pick Print Stepper Clutch Delay Counter

The pick print stepper clutch delay counter is a four stage binary counter that is used to control the time duration of the pick current to the print stepper clutch magnets. This counter also controls the time between de-energizing and energizing the print stepper clutch magnets while advancing the card to the next line of print.

The counter is normally in a reset status until a print operation begins. The counter steps when stopping the print stepper clutch in its home position at the beginning of the print operation by 'feed check time base 5' signal. Stepping also occurs during print clock 1 or 10 or 16 or 22 or 28 times (Figure 2-17).

At the beginning of a print operation, the print stepper clutch pick and hold magnets are energized. The counter then counts 9 'feed check time base 5' signals. At count 9, the clutch pick magnet is de-energized. The pick magnet is energized 5.12 milliseconds to 5.76 milliseconds.

Whenever a card is advanced to the next line, the print stepper clutch hold magnet de-energizes when the print clock equals 1 (advance to line 1), 10 (advance to line 2), 16 (advance to line 3), or 22 (advance to line 4 or eject the card), or count 28 (eject the card after four lines of print). These print clock counts enable the pick print stepper clutch delay counter. If the card is to be advanced to the next line, the pick and hold magnets are energized when the counter reaches count 9. The counter resets and then starts counting again. Again at count 9, the pick magnet is de-energized. The hold magnet remains energized until the card is to be advanced again.

See FEMD 4-105 for a description of the pick print stepper clutch delay counter. Refer to "Print/Feed Operation" in Chapter 3 for a card advancement description.

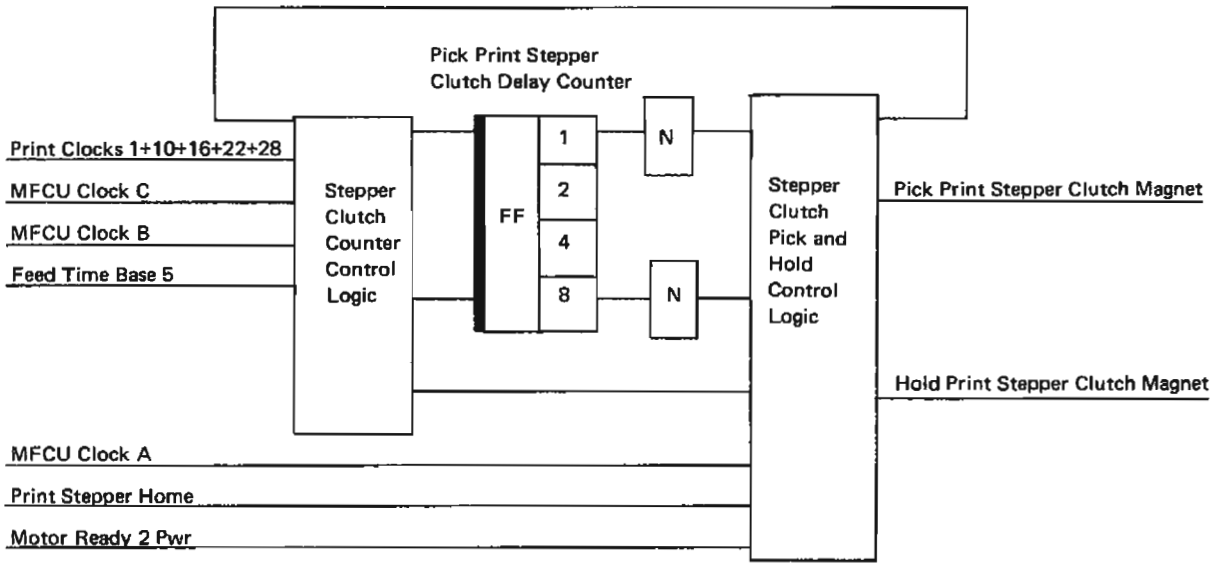


Figure 2-17. Pick Print Stepper Clutch Delay Counter

Print I/O Cycle Counter

The print I/O cycle counter is a five position counter that indicates the 'last print data I/O cycle' (Figure 2-18) for each group of comparisons made by the CPU.

This counter is stepped after each comparison is made by the CPU. After the print character count value for each typewheel has been compared with the character codes in the print area of core storage (16 comparisons for model A2 machines; 8 for model A1), the 'last print data I/O cycle' signal is used to modify the LSR address so that 16 more comparisons can be made for the next print character count. The counter is reset and counting begins for the next group of comparisons.

After a complete revolution of the typewheels, the LSR address is modified accordingly to select the correct core storage address for printing after a carriage shift or to begin printing the next line. Remember that the LSR address is modified after each cycle steal request.

See FEMD 4-050 for a description of the print I/O cycle counter.

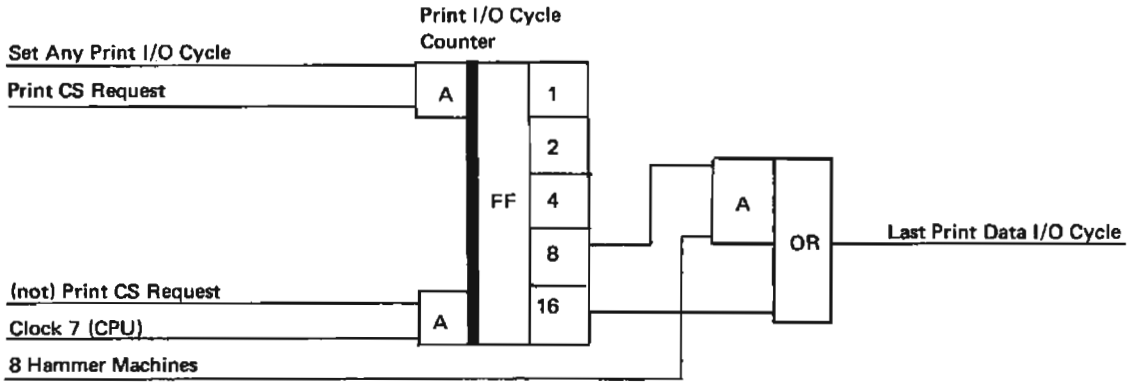


Figure 2-18. Print I/O Cycle Counter

NINE STAGE TIME DELAY COUNTER

The time delay counter is a nine stage counter that is used to turn off the MFCU motor if no MFCU SIO command is given within about 30 seconds after the completion of the last feed command (Figure 2-19).

The counter is stepped by the 'print step CB' pulse sent from the MFCU to the attachment. The nine stage time delay counter is reset and starts counting from zero after each accepted MFCU SIO command. If the counter reaches count 256, no MFCU SIO command was given within 30 seconds. The 'no command time out' signal is generated which results in stopping the MFCU motor.

An MFCU SIO command or an NPRO starts the MFCU motor. As the motor starts MFCU mechanical motion, the nine stage time delay counter is stepped by the 'print kick CB' pulses. At count three, the 'motor ready 1' line is activated. At count 5 the 'motor ready 2' line is activated. MFCU operations start after activating the 'motor ready 2' line. This time delay allows the MFCU motor to reach operating speed before starting card movement.

See FEMD 4-115 for a description of the time delay counter. See "Motor Start-Stop Operation" in Chapter 3.

UP-DOWN COUNTER

The up-down counter is a two stage binary counter that is used to generate the 'wait to stacker clear' signal. This signal allows the motor stop counter to be stepped by each hopper CB pulse.

The up-down counter is reset to three (11 binary) by a power on reset or an NPRO operation (FEMD 4-110). The 'wait to stacker clear' signal is active until the first card is ejected from the wait station. Picking a punch registration pressure roll magnet advances the up-down counter from three to two (10 binary). The 'wait to stacker clear' signal is then deactivated. The up-down counter does not step again until the stacker transport counter equals 21. The stacker transport counter does not begin stepping until the card reaches the post print cell located in the print station. During 'stacker count 21', the up-down counter is stepped three times (FEMD 4-110). The counter equals three after stepping the counter three times. Also during stacker count 21, the 'gate down count' latch was set. Setting this latch results in setting the 'inhibit gate down count' latch. Setting the 'inhibit gate down count' latch results in re-setting the stacker transport counter. The 'gate down count' and the 'inhibit gate down count' latches are also reset thereby preventing further stepping of the up-down counter. The 'wait to stacker clear' signal is again activated.

See FEMD 4-110 for a description of the up-down counter.

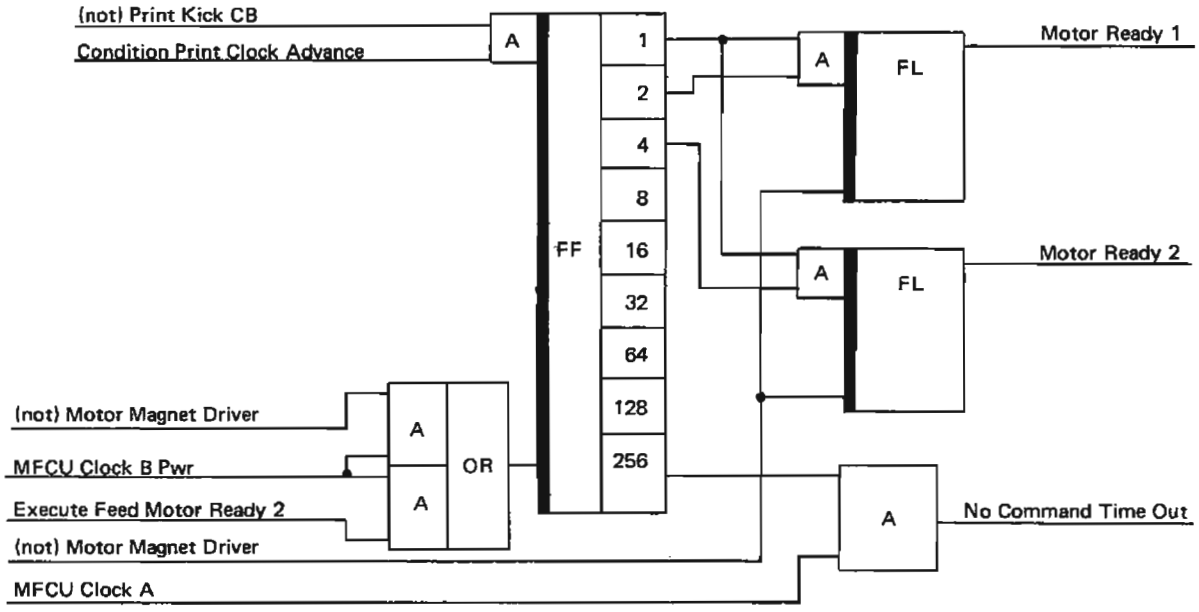


Figure 2-19. Nine Stage Time Delay Counter

MOTOR STOP COUNTER

The motor stop counter is a three stage binary counter that is used to stop the MFCU motor after both hoppers are placed in a not ready status.

This counter is held reset until the 'wait to stacker clear' line is activated. The counter is then stepped by the 'hopper CB leading edge' pulse (Figure 2-20). At count 4, the 'gate stop' line is activated. If both hoppers are not ready, the MFCU motor stops. Pressing the stop key places both hoppers in a not-ready status. This counter along with the up-down counter causes the motor to stop. Without these counters the motor would not stop until a no-command-time-out signal stops the motor. This would take about 30 seconds.

See FEMD 4-095 for a description of the motor stop counter. See "Motor Start-Stop Operations" in Chapter 3.

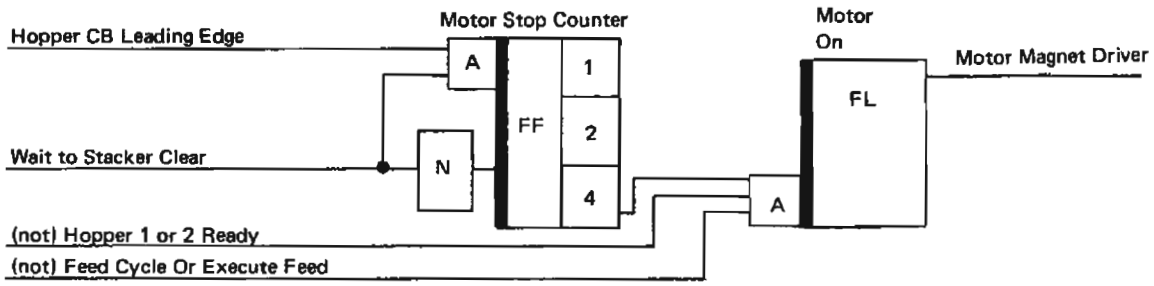


Figure 2-20. Motor Stop Counter

FEED CHECK COUNTERS

The purpose of the various feed check counters is to check card motion as cards move through the transport. There are five feed check counters in the MFCU attachment:

1. 5 stage hopper counter (FEMD 4-120).
2. 5 stage wait counter (FEMD 4-125).
3. 6 stage stacker transport counter. This counter is also used for purposes other than feed checks.
4. 6 stage corner counter (FEMD 4-130).
5. 6 stage feed check time base counter (see "Feed Check Time Base Counter").

Figure 2-21 illustrates the basic principles of operation for one of the MFCU attachment feed check counters. Operation of the first four check counters is basically the same.

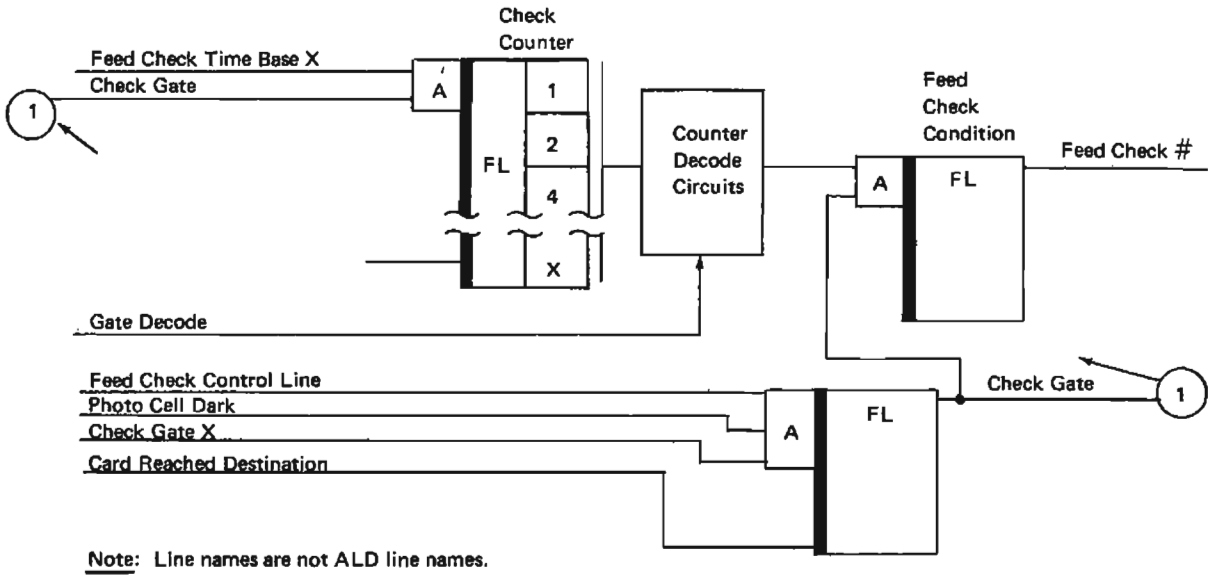


Figure 2-21. Check Counters

A feed check counter is allowed to start counting when the 'check gate' line is activated (Figure 2-21). The counter is driven by the 'feed check time base x' signals. The check gate latch is set to indicate when the counter should start counting. As the card moves through the transport, the counter is stepped by the proper feed check time base pulses from the feed check time base counter. When the card reaches its destination, the check gate latch is reset and a new check gate is turned on resulting in a counter reset.

The card must reach its next check point prior to a specific count decode. If the card did not arrive prior to the specified count to reset the 'check gate latch' (Figure 2-21), a feed check latch is set indicating a feed check condition. The MFCU motor stops whenever a check condition occurs.

The feed check 20 indicator turns on when a gear emitter check or a fire CB check occurs. A gear emitter check turns on check light 20, turns off the ready light, and stops the MFCU motor. A fire CB check turns on check light 20 and de-activates the print time switch to stop printing. The MFCU motor does not stop.

See FEMD 2-005 for MFCU error conditions. Chapter 2 in the *IBM 5424 MFCU Attachment Field Engineering Maintenance Diagrams Manual* develops the cause of the check conditions.

STACKER SELECTION REGISTERS

The stacker selection registers consists of one 3-position register and five 2-position registers (FEMD 4-140).

Bits 5, 6, and 7 of the SIO control code are used to select a stacker pocket. Note that a card sent from the hopper always stops in the wait station. No SIO stacker selection is necessary if the selected wait station is empty. The first card remains in the wait station until the next SIO instruction.

During the SIO I-R cycle, the stacker control code is gated into the wait stacker DBO register. Stacker data is gated in into the punch stacker register when the card is ejected from the wait station. The wait stacker DBO register is then reset by the 'trailing edge or hopper check' signal or by a 'power or NPRO' signal (FEMD 4-140).

As the card moves toward the stacker, the stacker pocket selection data shifts through the stacker registers. The 'stacker count 21' signal gates the data to the stacker magnet register. This data is then decoded thereby energizing a stacker selector magnet.

If not stacker selection is specified by the SIO instruction, cards from the primary feed go to stacker pocket 1; cards from the secondary feed go to stacker pocket 4.

See FEMDs 4-140 and 4-145 for a description of the stacker selection registers.

This chapter describes the MFCU attachment operations. References are made to the *IBM Field Engineering Maintenance Diagrams Manual, 5424 Multi-Function Card Unit Attachment,* Form SY31-0254 (FEMD) to support the descriptions in this chapter.

The types of diagrams in the FEMD are flowcharts, timing charts, and second level diagrams or circuit diagrams. These diagrams illustrate attachment operations. Inputs to the attachment from the MFCU and CPU are shown in the FEMD. Outputs from the attachment to the MFCU and CPU are also shown. These inputs and outputs show the timing relationships among the CPU, the attachment, and the MFCU.

The flowcharts are divided into primary flow and secondary flow. The primary flow states the major objectives accomplished by an operation, a cycle, a request, etc. The primary flow is set off by heavy weight lines from the start of the operation to the completion of the operation. The secondary flow support the objectives stated in the primary flow by showing the sequence and how the function is performed.

The timing charts show when circuits are activated with respect to each other. Timing relationships between the attachment and the MFCU and between the attachment and the CPU are also shown.

The circuit diagrams are drawn in positive logic and show how a logical function is performed. Reference is also made to ALDs.

The three types of diagrams should be used for learning, for recall, and for troubleshooting. This chapter describes attachment operations. Refer to Chapter 2 in this manual and to Chapter 4 in the FEMD for descriptions of the functional units.

LOAD I/O INSTRUCTION OPERATION

- Loads two storage address bytes in the selected LSR from the specified address in the LIO instruction and the specified address minus one.
- Consists of an I-op cycle, I-Q cycle, CPU addressing cycles I-H1, I-L1, or I-X1), and two E-B cycles.
- The I-Q cycle decodes the Q byte of the instruction and sends a condition code to the CPU.
- An LIO instruction must be executed for each LSR address register loaded.
- The LSR address registers are:

  1. MRDAR-MFCU read address register.
  2. MPCAR-MFCU punch address register.
  3. MPTAR-MFCU print address register.

The load I/O instruction is a single address instruction that can be executed only if the MFCU is not busy. If the instruction cannot be executed, 'channel one I/O condition A' is sent from the attachment to the CPU indicating non-acceptance of the LIO command. The CPU loops on the LIO command until command execution can occur. If CPU is in dual program mode, the program advances to the next program level if the MFCU is busy to the LIO instruction.

Three LIO instructions are required to load the three MFCU LSR address registers. (See Figure 1-2 for LIO instruction format.) After the MFCU LSR address register is loaded, a start I/O command can be executed.

I-Q Cycle

During the I-Q cycle, the following events occurs (FEMD 5-005 and FEMD 5-010):

1. The device address is decoded.
2. Bits 4, 5, 6, and 7 are gated into the I-Q register. Bits 5, 6, and 7 are decoded by the N field decoder.
3. The print buffer 1 and 2, read feed, or punch is checked for a busy condition as determined by the N code.
4. The condition code is determined and sent to the CPU.

An LIO instruction and Q byte sent by the CPU are received by all attaching I/O devices. The MFCU attachment decodes the device address (DA) in the Q byte (Q byte bits 0, 1, 2, and 3) to determine if the LIO is for the MFCU. After decoding the N field, the attachment checks to determine if the LSR specified is busy. If the MFCU attachment is selected (DA equals F), and the LSR to be selected is not busy, a 'channel 1 I/O condition code B' is sent to the CPU. This indicates to the CPU that the LIO instruction was accepted.

For normal operations, the M bit equals 0. The N code must equal 4, 5, or 6. Otherwise, an invalid Q code occurs (FEMD 5-005). (Neither condition A nor condition B is sent to the CPU for an invalid Q code.)

The selected LSR is loaded with data from core storage and the attachment is placed in diagnostic mode when the LIO M bit equals one. A diagnostic mode operation requires that the N field equals 5 for diagnostic read mode or 6 for diagnostic punch mode (FEMDs 5-010 and 5-360).

E-B Cycles

Between the I-Q cycle and the EB-1 cycle, the CPU takes I-H1 and I-L1 (direct addressing) or I-X1 (indexing) cycles for core storage addressing purposes.

During the first EB-1 cycle, the channel LSR select lines are sent to the CPU during 'clock 2' time. The first data byte from core storage is gated into the selected LSR low order position at CPU 4D time.

The second data byte is fetched from core storage operand address minus one and loaded into the LSR high order position of the selected LSR register during the second E-B cycle.

The selected LSR is now loaded. The CPU starts the next instruction.

Refer to FEMD 5-005 and FEMD 5-010 for LIO instruction flowchart, timing chart, and circuit diagram.

TEST I/O AND BRANCH INSTRUCTION OPERATION

- Test for conditions specified by N code.
- Branch to address specified in branch address if condition is met.

The test I/O and branch (TIO) instruction is a one address instruction that tests the I/O device for a specified condition and branches to a specified address if the branch condition exists. The device address and M bit are bits 0 through 4 of the Q byte. The N field (bits 5-7) contains the N code specifying the condition being tested. The eight conditions checked for during the I-Q cycle are:

1. N=000–specified feed not ready or device error condition (feed specified by M bit).
2. N=001–read/feed busy.
3. N=010–punch data busy.
4. N=011–read/feed busy or punch data busy.
5. N=100–print data busy.
6. N=101–read/feed busy or print data busy.
7. N=110–punch data busy or print data busy.
8. N=111–read/feed busy or punch data busy or print data busy.

I-Q Cycle

The 'TIO' signal is sent to the attachment during the 'CPU' I-op cycle. The condition to be tested for occurs during the I-Q cycle. If the specified condition does not exist, a 'channel 1 I-O condition B' signal is sent to the CPU (FEMDs 5-015 and 5-025). The program then advances to the next instruction. If the specified condition exists or the specified feed is ready (N equals 0), a 'channel 1 I-O condition A' is sent to the CPU. If condition A exists, the program branches to the address specified by the instruction branch address (Figure 1-4). The program branch occurs during CPU operation end.

Refer to FEMD 5-015, FEMD 5-020, FEMD 5-025 for TIO flowchart, timing chart, and circuit diagram.

ADVANCE PROGRAM LEVEL (APL) INSTRUCTION OPERATION

- The APL operation tests for specified I/O condition.
- The program loops on the APL instruction until condition no longer exists on machines without the dual program feature.
- The program advances to the next program level if condition tested for exists.
- The condition tested for is specified by N code.

The APL instruction is a command instruction used primarily when the dual program feature is installed in the CPU. The dual program level feature provides the ability to execute two independent programs on a time sharing basis. However, for programming compatibility, machines without the dual program accept the APL instruction.

On machines without the dual program feature, the APL instruction tests for the same conditions as a TIO instruction. If the advance condition is met, the attachment activates the 'channel 1 I-O condition A' line. The program loops on the APL instruction until the condition being checked for no longer exists. If the advance condition is not met, the attachment activates the 'channel 1 I-O condition B' line. The program advances to the next sequential instruction.

If the condition tested for is met on machines with the dual program feature installed, the APL instruction advances to the next program level. If the condition tested for is not met, the program continues with the same program level.

I-Q Cycle

The I-Q cycle is similar to that of the TIO instruction. A condition code of A means that the condition being checked for is busy. The program loops on the APL instruction until the condition being check is not busy and a condition code of B is activated.

Because the APL instruction execution is the same as the TIO instruction execution, refer to the TIO diagrams (FEMD 5-015, FEMD 5-020, FEMD 5-025) for flowchart, timing chart, and circuit diagram.

SENSE I/O INSTRUCTION OPERATION

- Move two sense bytes from attachment to core storage.
- Instruction is always executed whether or not the device is busy or needs attention.
- Consists of an I-op cycle, an I-Q cycle, and two E-B cycles.
- I-Q cycle decodes the Q byte and sends a condition code of B back to the CPU.
- Two E-B cycles are taken to assemble the status bytes and to send the status data to the CPU.

The SNS instruction assembles two sense bytes and stores these two bytes in core storage. The program examines these sense bytes to determine MFCU and attachment status.

Status byte data indicates a busy condition, a check condition, or a card location in the transport. (See Figure 1-6 for MFCU status byte indicators.) The diagnostic programs use the MFCU special indicators for troubleshooting. (See Figure 1-7 for MFCU special indicators.)

A SNS instruction may also store one of the three MFCU LSR address registers.

The DA and M fields are located in the Q byte bits 0-4. Bits 5-7 of the Q byte determine the conditions to be sensed. The sense bytes are stored in the specified address and in the specified address minus 1 (Figure 1-5).

The MFCU attachment always accepts an SNS instruction. Therefore, the CPU always receives a condition code of B from the attachment during the I-Q cycle. An N code of 2 or 7 is invalid and causes the CPU to stop with a processor check. An invalid N code sends condition A to the CPU. A parity check sends condition A and condition B to the CPU.

I-Q Cycle

The attachment decodes the DA the same as the other I/O instructions. The M bit is meaningless and should equal 0.

Busy conditions are not checked for during the I-Q cycle of an SNS instruction. If the N field equals 2 or 7, invalid Q code results. Otherwise, the attachment always sends a 'channel 1 I-O Condition B' to the CPU (FEMD 5-045). A parity check results in sending conditions A and B to the CPU. The SNS instruction is not executed.

The N field decode output determines the sense byte that is sent to the CPU (FEMD 5-050). If the N field equals 100, 101, or 110, an MFCU LSR address register is selected and the contents are stored in core storage.

E-B Cycles

Two E-B cycles are necessary to assemble and store the two sense bytes. Data is gated into the data bus in assembler during the E-B cycles. Odd parity is assigned to these sense bytes and stored in core storage.

If the N code equals four or five or six, an LSR is selected at 'clock 2' time during the two E-B cycles. LSR data is stored in storage 1.

Refer to FEMDs 5-030, 5-035, 5-040, 5-045, 5-050, 5-055, and 5-060 for SNS instruction flowcharts, timing charts, and circuit diagrams.

START I/O INSTRUCTION OPERATIONS

- The SIO instruction starts the specified MFCU operation.
- The SIO instruction consists of an I-op cycle, an I-Q cycle, and an I-R cycle.
- The I-Q cycle decodes the Q byte of the instruction and sends a condition code of A or B to the CPU.
- The execute latches are turned on during the I-R cycle.
- The control code part of the instruction is used to:

  1. Select the print buffer address.
  2. Select 8-bit IPL (Initial Program Load) read mode.
  3. Select three or four lines of print.
  4. Select a stacker pocket.

The operations performed by the MFCU are read/feed, punch, print, and selective stacking. Multi-operations can be performed on the card from the specified wait station. The

SIO punch, print, and stack commands always refer to the card in the specified wait station. A card is always read or fed from the specified hopper and stops in the vacated wait station. Therefore, the read/feed operation specified by the SIO instruction refers to the card in the specified hopper. For example, an SIO instruction can specify that the card in the wait station be punched and printed and then stacked in a specified stacker pocket. A new card from the selected hopper is read or only fed and then placed in the vacated wait station.

The SIO instruction starts MFCU operations. A device address of F (hexadecimal) contained in Q bits 0, 1, 2, and 3 selects the MFCU. Bit 4 (M bit) selects the card feed path (wait station and hopper). Bits 5, 6, and 7 (N field) define the operation to be performed.

The control byte of the instruction further defines the device operations (Figure 1-9). This instruction control byte selects the print buffer address, selects 8-bit IPL read mode, selects three or four lines of print, and selects a stacker pocket.

The SIO instruction requires three CPU cycles. During the I-op cycle the CPU sends the SIO instruction to all I/O devices. The I-Q cycle decodes the device address, M bit, and N field. During the I-R cycle, the execute latches are set defining the operation to be performed (such as read, punch, or print). Also, during the I-R cycle controls are set as determined by the instruction control code byte.

Refer to FEMDs 5-065, 5-070, 5-075, and 5-080 for SIO instruction flowchart, timing chart, and circuit diagram.

I-Op and I-Q Cycles

During the I-op cycle the CPU decodes the SIO instruction and sends the 'SIO instruction' to the attachment (FEMDs 5-075 and 5-080). The attachment receives the 'SIO instruction' signal to start attachment operations.

During the I-Q cycle, the Q byte is loaded in the data bus out (DBO) register (FEMD 5-080). The Q byte contains the device address, the M bit, and the command. The attachment unit decodes its address and sends a condition code of A or B to the CPU (FEMDs 5-075 and 5-080).

The CPU sends an SIO instruction signal to all attaching I/O devices. Therefore, each I-O attachment decodes the device address in the Q code. If the device address equals F (hexadecimal), the MFCU responds by sending condition A or condition B to the CPU (FEMD 5-075). If the MFCU is busy or requires attention, condition A is sent to the CPU. The CPU waits (I-R back up) until the MFCU is not busy or no attention required. Condition B indicates acceptance of the SIO instruction. The MFCU then sets up its circuitry to prepare for the I/O operation. No response of condition A or B indicates that the MFCU did not decode its address. Conditions of A and B indicate a parity check.

After the CPU receives condition code B from the attachment, the 'program interlock' latch in the CPU is reset indicating acceptance of the SIO instruction (Figure 1-2). The M and N fields are stored in the I-Q register (FEMD 5-080). This data is then decoded and the command is stored in the MFCU attachment during the CPU I-R cycle to define MFCU operations. The M bit selects the primary or secondary card feed path. The decoded N field determines the operation to be performed. The N field operations that are defined are read/feed, punch, and print.

I-R Cycle

During the I-R cycle, the control byte is sent over DBO to the attachment unit for controlling operations not specified by the Q code. Also, during the I-R cycle the execute latches are set (FEMD 5-080). Outputs from the DBO buffer are decoded to perform functions such as selecting print buffer one or print buffer two (FEMD 5-235), selecting a normal read or an IPL read mode (FEMD 5-320), selecting three or four lines of print (FEMD 5-235), and selecting a stacker pocket (FEMDs 4-140 and 4-145). If no stacker selection is specified, cards from the primary card feed path are sent to stacker pocket one; cards from the secondary card feed path are sent to stacker pocket four.

A no-op command results if a feed check occurred, or if a card is not in the specified wait station, or is not on the way to the specified wait station (FEMD 5-075). On a no-op command, the program continues with the next instruction.

The end of the I-R cycle completes the SIO instruction execution. The attachment can now begin the specified operation. The CPU continues to execute instructions until the MFCU attachment sends a cycle steal request to the CPU.

Read/Feed Operation

- The read operation reads 96 card columns from a card and stores card data in core storage.
- The feed operation is the same as the read operation except that data is not sent to the CPU and the MRDAR is not selected. However, the card is actually read by the attachment circuits.
- The 18 bits read from the three card tiers are sent to the CPU as three separate data bytes.

- The attachment generates a cycle steal request for each data byte transferred.
- The data is checked by comparing outputs from the read data register and the read check register.
- The card being read stops in the selected wait station.

Refer to FEMDs 5-085, 5-090, 5-095, 5-110, 5-120, 5-125 and 5-130 for read/feed flowcharts, timing charts, and circuit diagrams.

The 'execute feed' and the 'execute read' (if read is specified) latches were set during the SIO instruction I-R cycle. The read or feed operation begins by energizing a primary or a secondary hopper magnet (FEMD 5-120). Energizing a hopper magnet feeds a card into the MFCU read inject station. From the read inject station the card is then fed into the read station. After reading the card it is then sent to the wait station. The card remains in the wait station until an SIO instruction selects the same card feed path.

The specified hopper magnet cannot be energized if (FEMD 5-120): a card is in the specified wait station, a card is on the way to the wait station, the 'punch execute' latch is set, or the 'execute print' latch is set. However, if the punch unit is busy and the 'execute punch' latch is set, the clutch can be energized when the 'punch allow hopper' latch is set during a punch operation. If only a read is specified, a hopper magnet is energized soon after the punch registration pressure roll magnet is energized (FEMD 5-120).

For our discussion, assume that no previous SIO instructions were executed and that no card is on the way or in the specified wait station. The specified hopper magnet is energized when the 'hopper CB LE' pulse occurs. The hopper magnet remains energized until the next 'hopper CB LE' pulse occurs. Energizing the clutch magnet causes a card to be fed by the pickerknives into the read inject station. After the card is registered in the read inject station, the read inject feed roll starts the card toward the read station.

The reading of data begins when the leading edge of the card covers the 18 read cells. After the read cells are dark (covered) the 'leading edge' latch is set (FEMD 5-125). The read gear emitter counter then begins counting (FEMD 5-125). When the read gear emitter counter equals 46, the 'start read' and the 'allow read' latches are set (FEMD 5-125). Also, the read gear emitter counter is reset. The first card column group (columns 1, 33, and 65) is now in position to be read.

As the card column group passes over the read cells, the 'Gated read sample' signal gates data into the read data register and the read check register. Three attempts are made to gate data into the read data register on read gear emitter counts of 1, 5, and 9. Three attempts are also made to gate data into the read check register on counts 3, 7, and 11 (FEMD 5-125).

When the read gear emitter counter equals count 11, the 'read CS request' latch is set. Setting this latch indicates that data is read from the card and that cycle steal requests can now begin.

Data from the read data register is sent to the data bus in (DBI) assembler one byte at a time (FEMD 5-130) under control of a read-punch-punch check counter. The attachment sends a cycle steal request (FEMD 5-130) to the CPU for each data byte from a card column group (3 bytes per group, e.g. columns 1, 33, and 65 or columns 2, 34, and 66 etc.). Parity generation circuits assign odd parity to each data byte sent to the CPU.

The data in the two registers are then compared. If the data does not compare, the MFCU read check and the CPU channel I/O check indicators are turned on. A subsequent SNS instruction stores a one in bit 0 of status byte one to indicate a read check. The program may then determine the type of error.

The MRDAR in the CPU selects a storage address for each data byte sent by the attachment. For each data byte transferred, the attachment selects the MRDAR for storing data and for updating the MRDAR. Updating the MRDAR is necessary so that the card column data group will be stored in the correct storage locations.

After the first byte is stored (from tier one), 32 is added to the MRDAR. Thirty-two is again added after the second byte (tier two) is stored. After tier three is stored, the MRDAR is decremented by 63 so that the first byte of the next card column group will be stored in the correct core storage address. Reading continues until all 96 columns (32 card column groups) are read.

Three cycle steal requests are sent to the CPU for each card column group stored. A cycle steal request is sent at clock 4 time to the CPU (FEMD 5-130). The CPU responds by sending a DBO 0 and DBO 4 to the attachment. The MRDAR is then selected by sending a channel LSR select 6 and LSR select 7 to the CPU. The MRDAR contents are transferred to the CPU storage address register to address core storage for storing the data byte sent from the attachment. After the first data byte is transferred, 32 is sent to the CPU via the DBI. This results is adding 32 to the MRDAR so that the second byte in the card column group will be stored 32 locations above the first byte.

At clock 4 time another cycle steal request is sent to the CPU. (Cycle steal requests are sent every CPU clock 4 time

until the CPU responds with DBO 0 and DBO 4 signals.) Byte 2 is stored and 32 is again added to the MRDAR to prepare for the third byte from the card column group.

After the third card column group byte is transferred, the 'last cycle' latch is set (FEMD 5-130). This results in subtracting 63 from the MRDAR so that the first byte of the next card column group will be stored in the correct storage location. Sixty-three is subtracted because 'modify by 32', 'modify by 31', and 'modify minus' signals are activated (FEMD 5-130). The read gear emitter counter continues to count until count 11 is again decoded before generating cycle steal requests for storing the next card column group. By count 11 the next card column group is in the read data register and the read check register.

After 32 card column groups are stored, several read cells are covered. At count 512 the 'allow read' latch is reset (FEMD 5-125). At read gear emitter count 528 the 'trailing edge' latch is set beginning the ending operations (FEMD 5-120). See FEMD 5-090 flowchart for the ending operation sequence. At the end of the operation, a latch is set to indicate a card is in the specified wait station (FEMD 5-120). The card remains in the wait station until the next SIO instruction selects that wait station. An LIO instruction must load the MRDAR before executing another SIO read command.

The feed operation is the same as the read operation except that the MRDAR is not selected and data is not sent to the CPU. Attachment operations are the same for the read and the feed commands. The card is sent to the selected wait station. No LSR selection or data transfer occurs on a card feed operation because no cycle steal requests are generated (FEMD 5-120).

Punch/Feed Operation

- The punch/feed operation punches data in the card from the specified wait station.
- The cards are punched 18 bits at a time.
- Three data bytes are sent to the attachment for each punch cycle.
- The data punched in the card is sensed by punch check crystals for checking purposes.
- The attachment sends three bytes of punch check data to the CPU for checking data that was punched with the data sent to the attachment.
- The card from the selected hopper is sent to the vacated wait station. This card may or may not be read at the read station.

While studying the punch/feed operation, refer to:

1. Punch/feed operation flowcharts—FEMDs 5-135, 5-140, and 5-145.
2. Punch/feed operation timing charts—FEMDs 5-150, 5-155, and 5-160.
3. Punch/feed operation circuit diagrams—FEMDs 5-170, 5-175, 5-180, 5-190, and 5-195.

The SIO instruction starts the punch operation. If the MFCU motor was stopped, the SIO instruction causes the motor to start. Also, as the result of the motor starting, the punch magnets are energized and de-energized several times. The 'rattle punch rattle' signal is available while the motor is reaching full speed (see "Motor Start-Stop Operations"). The rattle-punch-rattle is active after setting the 'motor ready 1' latch and before setting the 'motor ready 2' latch (FEMD 5-180).

The 'execute feed' and the 'execute punch' latches were set during the SIO instruction I-R cycle. Before the punch operation starts, the card from the specified wait station must be registered in the punch station. To register a card in the punch station, the attachment energizes the following in the MFCU (FEMD 3-031):

1. Primary or secondary punch registration pressure roll magnet (FEMD 5-170).
2. Punch eject pressure roll magnet (FEMD 5-170).
3. Punch stepper pressure roll magnet (FEMD 5-170).
4. Punch registration gate magnet (FEMD 5-170).

No card movement occurs until the punch step counter equals count one. The counter is held reset until the card path is clear. For example, a print operation may prevent card movement because the 'punch execute' latch cannot be set until the 'print allow punch execute' line is activated (FEMD 5-170). The 'punch unit busy' latch is set to start the punch operation. The punch step counter is advanced to count one after the next 'punch step CB' pulse occurs.

At count one, the specified punch registration pressure roll magnet energizes (FEMD 5-170). Also, the punch eject pressure roll magnet energizes.

Energizing the specified punch registration pressure roll magnet (FEMDs 3-031 and 5-170) causes the card to move from the wait station toward the punch station. Because a punch command was specified by the SIO instruction, the punch eject pressure roll magnet is energized to prevent sending a card through the punch station. (If a punch operation was not specified, the card would move through the punch station, without stopping, to the corner station because the eject pressure roll magnet and the punch registration gate magnet would be de-energized.)

The punch stepper pressure roll magnet is energized until after the card is registered and after column one is punched. As the card moves to the punch station, the card is held against the card rail. Because the registration gate magnet is energized, the registration gate stops the card at column group one position.

After the card is registered, the primary or secondary punch registration pressure roll magnet and the punch registration gate magnet are de-energized. (The punch eject pressure roll magnet remains energized.) Note that the punch stepper pressure roll magnet is not de-energized at this time (FEMD 5-170). This magnet is de-energized when the punch step counter equals count five. At count five, the punch stepper pressure roll magnet is de-energized. After a punch cycle is complete and the punches are withdrawn from the card, the punch stepper mechanically advances the card to the next card column group.

When the punch step counter reaches count four, the 'punch allow hopper' latch is set. Setting this latch allows a read/feed operation to begin. The card is moved from the specified hopper through the read station and stops in the vacated wait station.

Punch Data Cycle Steal Request

Before punching begins, three data bytes are sent from the CPU to the attachment. The attachment generates a cycle steal request for each data byte received (FEMDs 5-135, 5-155, and 5-175). These three data bytes are gated into the punch data register. This data is then punched in the card.

When the punch step counter reaches count four, the 'punch character request' latch is set. On the next 'punch check CB' signal the 'punch CS request' latch is set (FEMD 5-175). After the 'punch priority' latch sets, the attachment circuits are conditioned to start data transfer operations. Refer to FEMD 5-140 for a data transfer flowchart. Refer to FEMDs 5-155 and 5-175 for punch data cycle steal request timing chart and circuit diagram.

The first 'punch check CB' signal occurs once per revolution of the punch handwheel at 83°. A card column group is also punched for each revolution.

After the attachment circuits are conditioned to transfer data, a cycle steal request signal is sent to the CPU at clock 4 time. The CPU responds by sending DBO 0 and DBO 4 signals to the attachment. The attachment then responds by sending LSR select lines to select the MPCAR containing the storage address of the punch data to be sent to the punch data register. Also, the 'channel translate out' signal is sent to the CPU (FEMD 5-175). The CPU then sends the first data byte to the attachment over the data bus out (DBO) lines (FEMD 5-180). The read-punch-punch check counter controls the gating of data from the CPU into the punch data register (FEMD 5-180).

After the attachment receives the first data byte (tier one), 32 is added to the MPCAR so that the second byte (tier two) can be sent to the punch data register. Because the 'last cycle' latch (FEMD 5-175) is still reset, a modify by 32 is sent to the CPU over DBI. During the punch cycle, the LSR is selected at clock 8 to allow data transfer from the CPU. The LSR is also selected at the next clock 4 time to allow for MPCAR address modification (FEMD 5-175).

Thirty-two is added to the MPCAR after the CPU sends byte two to the attachment. After byte three (tier three) is sent, the MPCAR is decremented by 64. After byte three is sent to the attachment, the 'last cycle' latch is set (FEMD 5-175). Setting this latch causes the 'modify by 64', and the 'channel binary subtract' signals to be sent to the CPU. Subtracting 64 from the LSR restores the MPCAR to the address of the card column group that was punched. The MPCAR is at the correct storage address so that the punched data can be checked. (See FEMDs 5-135, 5-155, and 5-175 for cycle steal request execution.) Parity circuits check each data byte sent to the attachment for odd parity.

The punch CB counter controls the pick and hold signals to the punch magnets. After the punch data register is filled with data, the 'gate punch pick magnets' signal gates the data to the punch magnets. The next punch CB pulse steps the punch CB counter and the 'punch current hold control' is sent to the MFCU to hold the punch magnets until after punching is complete. If the punch magnet picks, a hole is punched in the card for the corresponding bit position.

When the punch step counter equals count five, the punch stepper pressure roll magnet is de-energized. After punching card column group one and after the punches are withdrawn from the card, the punch stepper mechanically advances the card to the next card column group to be punched.

Punching of the card occurs between 122° and 238°. The card is advanced to the next card column group between 266° and 108°. After the card is advanced to the next card column group, the three data check bytes are sent to the CPU for checking before the attachment receives punch data for the next card column group.

Punch Data Checking

Punch check crystals in the MFCU punch unit sense the 18 possible data punches for each card column group. As the punch unit approaches 180°, a selected punch compresses the punch crystal; if a punch is not selected, the crystal is not compressed.

A transistor shunts all the punch check crystals until the 'punch hold current control' line is de-activated (FEMD 5-180). De-activating this control line allows a capacitor to charge for each hole punched in the card during the crystal release cycle. This charge indicates that a hole was punched in the card. The capacitor stores this data until it can be gated to the check latches in the MFCU. There are six latches located in the MFCU for storing the check data; therefore, three sample signals are required to send the check data to the attachment.

Three sample signals are sent to the MFCU when the 'enable punch check' latch is set (FEMD 5-195). The data from the MFCU latches are gated to the data bus in assembler and then to the CPU for checking. The next sample signal turns on the transistor for the tier being checked. If the capacitor is charged the check data storage latch in the MFCU is set.

During punch step count four, the 'punch pick CB' signal sets the 'punch check CS time' latch. This occurs when the punch magnets are picked (FEMD 5-190). After the next 'punch check CB' signal, the 'punch check CS register' latch sets to start cycle steal requests for punch data checking. This occurs just before receiving data for the next card column group that is to be punched. A cycle steal request is generated for each of the three check bytes sent to the CPU. During the punch check cycle the 'enable punch check' latch allows the read-punch-punch check counter to gate data from the MFCU to the attachment DBI assembler (FEMD 5-195).

As each check byte is sent to the CPU, a corresponding data byte from storage is accessed and placed in the CPU B register. Data from the attachment is sent to the CPU A register. These two data bytes are then compared in the ALU. The result should be zero (no DBO bits). If a check condition exists, an 'any DBO bit' sets the 'punch check' latch indicating a punch check condition. However, the punch operation continues.

After the first check byte is sent to the CPU, 32 is added to the MPCAR. Thirty-two is also added after the second byte is compared. After the third check data byte is compared, 63 is subtracted from the MPCAR. The correct address is in the MPCAR for the next sequential card column group that is to be punched.

After the last check byte is sent from the MFCU to the CPU, the 'last cycle' latch is set and the 'punch check CS request' latch is then reset (FEMD 5-190).

The 'punch CS request' latch is then set to begin data cycle steals for punching the next card column group (FEMD 5-180). Data is transferred from the CPU to the attachment and punched in the card. After punching a card column group, the data is checked prior to punching the next card column group. The transfer and checking of data continues until 32 card column groups (96 card columns) are punched.

Ending Operations

During count 36 of the punch step counter, the last card column group is transferred to the punch data register and then punched in the card. After the last data byte is punched, the 'punch character request' latch resets ending data transfer from the CPU (FEMD 5-175).

During count 37, the punch data from the last group punched is checked. Also, the 'punch check CS time' latch is reset. The last punch data check cycle completes the punch operation (FEMD 5-190).

After the card is punched, the punch stepper pressure roll magnet is energized and the punch eject pressure roll magnet is de-energized. The card then advances to the corner station. If a print operation was not specified by the SIO instruction, the card is sent to a stacker pocket (FEMDs 5-145 and 5-150). The punch control circuits are restored to their starting condition.

A card is either read or fed and sent to the vacated wait station during the punch operation (FEMDs 5-120 and 5-170). Also, refer to the "Read/Feed Operation" in this chapter.

Print/Feed Operation

- Print three or four lines in card print area.
- Print character counter contents are sent to the CPU during each data cycle steal request.
- Print character counter value is compared with the core storage print image as addressed by the MFCU print address register (MPTAR) located in the CPU.
- The results of the compare are sent to the print shift register.
- After one complete revolution of the typewheels, eight (model A1) or sixteen (model A2) print hammers were energized unless a blank appears in the core storage print area print position.

- A side motion carriage shifts the card to print the next character group on the line being printed.
- After last line is printed, the card is sent to a stacker pocket.

While studying the print/feed operation, refer to:

1. Print/feed operation flowcharts—FEMDs 5-200, 5-205, 5-210, and 5-215.
2. Print/feed timing charts—FEMDs 5-220, 5-225, and 5-230.
3. Print/feed circuit diagrams—FEMDs 5-235, 5-240, 5-245, 5-250, 5-255, 5-260, and 5-265.

The 'execute feed' and 'execute print' latches were set during the SIO instruction I-R cycle. Bit 0 of the control code selects print buffer address 1 or print buffer address 2 (FEMDs 5-200 and 5-235). Also, the control code selects three or four lines of print (FEMD 5-235).

If the card path is clear, the 'print request' latch is set at the end of an MFCU SIO I-R cycle (FEMD 5-240). Setting this latch conditions the punch/feed circuits to feed a card through the punch station to the corner station by sending the 'print allow punch execute' signal to the punch/feed circuits (FEMDs 5-240 and 5-170). The card moves from the specified wait station to the corner station.

Three objectives must be accomplished before printing starts:

1. Execute a print load cycle steal request (CSR) I-O cycle.
2. Move the card from the corner station and register the card in print station. Also, set print 'mecahnically busy' latch.
3. Advance the card to line one position.

Print Load Cycle Steal Request

While the card is being registered in the print carriage feed rolls, the attachment generates an LSR load cycle steal request to select and load the low order byte of the MPTAR for print buffer 1 or print buffer 2. If print buffer 1 is specified, the MPTAR is set to XX00 to select print buffer 1. If print buffer 2 is specified, the MPTAR is loaded to XX80 (hexadecimal) by activating the 'modify print LSR by 128' (FEMDs 5-200, 5-225, and 5-235).

The print load CSR (cycle steal request) begins by setting the 'LSR load CS request' latch. At CPU clock 2 time, a 'CS priority bit 4' is sent to the CPU (FEMD 5-250). The CPU responds with DBO 1 and DBO 4 signals indicating that the request is granted. At clock 8C time, the 'any MFCU I/O cycle', 'any print I/O cycle', and 'print load LSR I/O cycle' latches are set. The LSR MFCU print address register (MPTAR) is selcted at clock 0 time. If print buffer address 2 was specified, 128 (80 hexadecimal) loaded into MPTAR low order byte. The MPTAR contains the starting core storage address of the data to be printed.

Print Station Card Registration

The card from the wait station moves through the punch station to the corner station. A photoelectric cell senses when the card is registered in the corner station. From the corner station, the card is then registered in the print station.

The first 'print step CB' pulse from the MFCU, after the 'print request' latch is set, turns on the 'print stepper home' latch (FEMD 5-240). The 'print stepper clutch' and the 'print clutch pick' latches are set thereby energizing the print stepper clutch magnet. Energizing this magnet stops the print stepper clutch in its home position. Energizing this magnet also allows the pick print stepper delay counter to be stepped by a shift of the 'feed check time base 5' signal. After the pick print stepper delay counter equals count nine, the 'print clutch pick' latch resets thereby de-energizing the clutch pick magnet. The hold current holds the armature that keeps the clutch in its home position. Also, the pick print stepper delay counter is reset until the card is to be advanced by the print stepper clutch (FEMD 5-240). After the card is registered against the print stepper rolls, the print stepper clutch magnet is de-energized and the card advances to the print line one.

To register a card in the print station, the print stepper clutch magnet, the corner alignment magnet, and the corner kick magnet are energized to kick the card into the MFCU print carriage feed rolls that are now stopped (FEMDs 5-200, 5-220, and 5-240).

After setting the 'print stepper home' latch, the 'print corner kick' latch is set when the card covers the corner station cell and after the next 'print inject CB' pulse from the MFCU (FEMD 5-240). After setting the 'print corner kick' latch, the 'print mechanical busy' latch is set. Also the corner kick counter is conditioned to allow the counter to advance by the 'feed check time base 4' signal. The reason for the corner kick counter is to allow the card to register in the corner station after covering the corner station cell. The corner kick counter counts to six. At count six, the corner kick and the corner alignment magnets are energized. The card is pushed by the corner kicker into the print stepper clutch feed rolls.

As the card is registered in the print station, the corner station cell uncovers resulting in setting the 'card in print station' latch (FEMD 5-240). The card is registered in the

print station and the 'print mechanical busy' indicates a printer busy condition. (See FEMD 5-200 for a flowchart and FEMD 5-220 for a timing chart.)

Advance Card to Line One

The print address register was set and the card was registered in the print stepper rolls during count zero of the print clock. Refer to FEMD 5-205 for a flowchart that shows card advancement. Setting the 'print mechanical busy' latch allows the print clock to be advanced after each 'print pick CB' and 'print inject CB' pulse (FEMD 5-245).

During count one of the print clock, the 'inhibit clutch hold' latch is reset thereby de-energizing the print stepper clutch magnet. The print stepper clutch turns advancing the card to line one. Also, the 'gate print step counter' latch is set allowing the pick print stepper delay counter to step. When the counter reaches count nine, the 'inhibit print clutch hold' latch sets which results in energizing the print stepper clutch pick and hold magnets. Also, at count nine, the 'gate print step counter' latch resets. The pick print stepper delay counter is also reset. Because the 'print clutch pick' latch is set, the 'gate print step counter' latch is set at MFCU clock C time. The pick print stepper clutch delay counter again steps to nine. At count nine, the 'print clutch pick' latch resets de-energizing the pick magnet coil (FEMD 5-240). The hold coil remains energized until the card is to be advanced again. When advancing the card to line one, the hold current is off for 5.12 millisecond to 5.76 milliseconds. The pick current is on for 5.12 milliseconds to 5.76 milliseconds.

No further action occurs until the print counter equals count four. At this time, the 'side motion gate' latch sets allowing the print side motion counter to step every 'feed check time base 2' signal. When the side motion counter equals count 6, the print side motion cam is at high dwell. At this time, the print side motion magnet is energized (FEMD 5-240). This magnet remains energized throughout the print operation. Printing begins when the print counter equals count five.

Print Cycles

The first 'print request clock' signal (print counter equals five) sets the 'print character request' latch (FEMD 5-255). Note that the 'inhibit request' latch was reset by a previous 'condition print clock advance' signal.

Print cycle steal request cannot occur until the attachment receives a 'print fire CB' signal from the MFCU. Because the print data register (bank 1 and bank 2) is reset, no printing occurs until after the next 'print fire CB' signal .

A cycle steal request results in the following (FEMDs 5-210, 5-230, 5-255, 5-260, and 5-265):

1. The contents of the print character counter are sent to the CPU (FEMD 5-260).
2. Contents of core storage are fetched as specified by the MPTAR.
3. Contents of core storage and contents of print character counter are compared in the CPU ALU.
4. A compare is indicated by sending 0's to the attachment over DBO (FEMD 5-265).
5. A compare results in setting the first position of the print shift register. All other bits in this register are shifted one bit position (FEMD 5-265).
6. The MPTAR is increased by 2 (16 hammer machines) or by four (8 hammer machines). See FEMD 5-260.

The first 'print fire CB' after the 'print character request' latch was set results in resetting the print data register bank 1. The print shift register is reset shortly after bank 1 is reset (FEMD 5-265). As a result of the print fire CB pulse, the load print data rank 2 counter is stepped every 20.48 microseconds. About 200 microseconds after the 'print load data rank 1' signal, print data register bank 2 is reset.

The 'load print data rank 1' signal also sets the 'print CS request' latch. Cycle steal requests are sent to the CPU every clock 2 time (FEMD 5-255). The CPU responds with a DBO 1 signal and a DBO 4 signal. During clock 8 time the 'any MFCU I/O cycle', 'any print I/O cycle', and 'print data I/O cycle' latches are set. The channel LSR 7 and LSR 4 signals select the MPTAR at clock 8 time. The data accessed from storage is placed in the CPU A register. After the CPU responds to a cycle steal request, the contents of the print character counter are sent to the CPU A register via DBI (FEMD 5-260). Also, data is fetched from storage and placed in the CPU B register. The CPU A and CPU B register contents are then compared by performing a binary subtract operation (a 'channel binary subtract' is sent to the CPU). The results of the comparison is sent to the attachment over DBO. If 'any DBO bit' is sent to the attachment (FEMD 5-265), a zero is placed in the first position of the print shift register. No bits (except a parity bit) sent over DBO indicates a compare and a one is placed in the print shift register. The bits in the shift register are shifted one bit position after each compare.

At clock 2 time, the MPTAR is selected for address modification. The 'gate address modify' signal sends a 'modify by two' (16 hammer machines) or 'modify by 4' (8 hammer machines) to the CPU via the DBI assembler.

After the first cycle steal request, the character for the first typewheel character was compared with the storage location for that typewheel. Because two (model A2) or four (model A1) was added to the MPTAR, the same character count

for the next typewheel is compared with a different location in core storage. Two (model A2) or four (model A1) is again added to the MPTAR for the third comparison. Sixteen comparisons (model A2) are necessary for each print character count value. These comparisons result in optioning each hammer for a character that is in the print position for a given fire CB pulse. When 16 (model A2) comparisons (8 for model A1) are complete, data from the shift register is gated to the print data register after the next 'print fire CB' pulse by the 'load print data rank 1' signal (FEMD 5-265). Data from bank one of the print data register sets print data register bank two. The 'print time switch' line was activated when the 'print mechanical busy' latch was set. The bits set in bank two select the print hammer magnets to print the selected characters. (See "Print Unit Operation Timing" in this chapter for shift register and print data register operations.)

The print I/O cycle counter counts the number of comparisons made (FEMD 5-255). After 16 comparisons (model A2), the 'last print data I/O cycle' signal resets the 'print CS request' latch. The print CS request counter is stepped at this time to count the characters optioned (FEMD 5-255). To prepare for the next group of comparisons, 30 (model A2) or 28 (model A1) is subtracted from the MPTAR (FEMD 5-260). The next 'print fire CB' pulse advances the print character counter. Sixteen more comparisons are made before energizing the print hammers. Again, the print I/O cycle counter counts the number of typewheels optioned per character. The print CS request counter counts the number of typewheel character groups compared during each revolution of the typewheels (FEMD 5-255). Sixty-four groups of sixteen comparisons are necessary for each half-line printed (model A2). Sixty-four groups of eight comparisons are made for each one-quarter line printed on eight hammer machines.

After one complete revolution of the typewheels, the print CS request counter equals 64. At count 64, the '64th character cycle' latch is set causing the 'print character request' latch to reset. The last data group compared is printed. The first one-half line is completed for 16 hammer machines; one-quarter line is printed for 8 hammer machines.

After printing the first one-half (model A2) line the print carriage shifts so that columns 2, 4, 6, 8, etc. can be printed. To prepare the MPTAR for printing after the carriage shift operation, the MPTAR is decremented by 29 (model A2) or by 27 (model A1). See FEMDs 5-210 and 5-215 flowcharts, 5-230 timing chart, and 5-260 circuit diagram.

Note: Print clock character request times are different for model A1 machines. See FEMD 5-230.

The print clock advances to count 6 and the print carriage shifts so that the second one-half line can be printed. The attachment print control circuits wait until completion of the carriage shift.

The next print request clock occurs when the print clock equals count seven. This results in setting the 'print character request' latch. The starting sequence for the second one-half line is the same as the starting sequence for the first one-half line. See FEMDs 5-210 and 5-215 flowcharts for printing one line of print. After 64 more groups of 16 comparisons, line one is completed. (Eight hammer machines require two more carriage shifts and two more cycles of 64 groups of 16 comparisons.)

After line one printing is completed, one is added to the MPTAR. The MPTAR now contains the starting address for the next line of print. The attachment circuits wait for the print clock to equal count 10. When the print clock equals count 10, the print stepper clutch hold magnet is de-energized. The card advances to print line 2. Circuit operation for moving the card to line two is similar to the operation for moving the card from the home position to line one. The print side motion carriage returns to its original high dwell position. The print circuits are prepared for the next line of print.

Print cycle steals and data transfer are the same as described in printing line one. The carriage shifts for each revolution of the typewheels. After one typewheel revolution, every character was compared with core storage location for the positions being printed.

Ending Operations

If four lines are to be printed, the card is advanced and one more line is printed. The 'last print clock' signal is inhibited until print clock 28 (FEMDs 5-215 and 5-245).

When printing three lines, the 'last print clock' signal occurs at print clock 22 time (FEMD 5-245). The last print clock resets the 'print stepper clutch' latch. This de-energizes the print stepper clutch hold magnet and the card is sent to a stacker pocket. The 'card in print station' latch and the 'print mechanical busy' latch are reset (FEMDs 5-215 and 5-240).

Print Unit Operation Timing

A characteristic of the print unit is that the print magnets must be energized about 800 microseconds for each character printed. Data is available at the inputs to the print register buffer bank one for about 600 microseconds (FEMD 5-265).

The 'load print data rank 1' signal gates the print data shift register contents to bank one of the print data register. The data from bank one is forced directly to bank two of the print data register. A set level from bank one always overrides the reset level to bank two. Because the set overrides the reset, the first 'load data rank 1' (reset) signal does not reset the bank two. The same data is still in bank one because bank one has not been reset. The next comparison group is always different from the first comparison group.

The next comparison group is loaded into bank one 600 microseconds after the first comparison group. Bank two now contains the result of both comparison groups. The first comparison group has energized the selected print magnet for 600 microseconds when the second group begins to energize the selected magnets. The bank two data from the first comparison group is reset 200 microseconds later resulting in energizing the print magnets for a total of 800 microseconds. Because data from two comparison groups are in bank two for 200 microseconds, the end result is energizing the print magnets for 800 microseconds.

STACKER SELECTION

- The wait stacker DBO bits 5, 6, and 7 are loaded during the SIO instruction I-R cycle. The control code (bits 5-7) of the SIO instruction specifies the stacker pocket.

- If no stacker pocket is specified, cards from the primary card feed path are sent to stacker pocket 1; cards from the secondary card feed path are sent to stacker pocket 4.

- Stacker data is shifted through the stacker selection registers as the card moves through the MFCU transport.

During the SIO instruction I-R cycle bits 5, 6, and 7 of the instruction control code are gated to the wait stacker register (FEMD 4-140). These bits indicate the stacker pocket for the card in the wait station during SIO instruction execution. The stacker pocket selection codes for bits 5, 6, and 7 are:

1. 000–no selection.
2. 001–no selection.
3. 010–no selection.
4. 011–no selection.
5. 100–stacker pocket 4.
6. 101–stacker pocket 1.
7. 110–stacker pocket 2.
8. 111–stacker pocket 3.

If no stacker pocket selection is specified, the cards from the primary card feed path are sent to stacker pocket 1; cards from the secondary card feed path are sent to stacker pocket 4.

As the card moves along the MFCU transport, the stacker pocket data is shifted in sequence through the stacker selection registers (FEMD 4-140). Data is gated to the punch stacker register 6 and 7 when the 'punch registration pressure roll 1 or 2 SS' signal occurs. This indicates that the card is moving from the wait station.

Because the specified wait station was vacated, the wait stacker DBO register is available for new stacker data during the next SIO instruction command.

After the card moves through the punch station, the corner station cell is covered. When the card leaves the corner station, uncovering the corner station cell gates the stacker data to the corner station register.

After the card leaves the corner station and is registered in the print station, the data is shifted to the print stacker register. Decoding stacker counter count 1 and count 21 gates the stacker data to the stacker register and then to the stacker magnet register. The selected stacker magnet energizes causing the card to go to the selected stacker pocket.

If DBO 5 bit is not on and if the primary card feed path is selected, the 'punch stacker 7' latch is set resulting in selecting stacker pocket 1 magnet; if the secondary card feed path is specified, no stacker magnets are energized and the card goes to stacker pocket 4 (FEMD 4-140).

INITIAL PROGRAM LOAD (IPL)

- An IPL operation reads program instructions into core storage.

- An IPL operation is started by pressing the CPU program load key or by executing an IPL SIO read instruction. Bit 1 of the SIO control code must equal 1.

- An SIO IPL can select either hopper.

- The program load key selects only the primary hopper.

- The IPL operation performs a read operation with the 8 and 4 bits in tier 3 becoming the D and C bits for tier 1; and the 2 and 1 bits of tier 3 becoming the D and C bits for tier 2. Tier 3 data stored in storage is usually meaningless.

When studying the IPL operation, refer to:

1. IPL Operation Flowchart–FEMD 5-315.
2. IPL Operation Circuit Diagrams–FEMDs 5-320 and 5-325.
3. "Read/Feed Operation" in this chapter and to FEMDs 5-085, 5-090, 5-095, 5-110, 5-120, 5-125, and 5-130 for flowcharts, timing charts, and circuit diagrams.

The IPL operation assembles 8-bit bytes of program instructions from a card and stores the program in core storage.

Because a 6-bit card code is normally used, DBI bit positions 0 and 1 equal 00 during a normal read operation. However, during an IPL operation, 8-bit bytes are read and stored in core storage. These bytes are the program instruction codes. Because an 8-bit code is necessary, bits 8, 4, 2, and 1 of tier 3 are used as the D and C bits for tiers 1 and 2.

During an IPL operation, bits 8 and 2 from tier 3 become the D bit for tiers 1 and 2; bits 4 and 1 from tier 3 become the C bits for tiers 1 and 2 (FEMD 5-320).

Bits 8 and 4 from tier 3 are gated as D and C bits when 'gate read data byte 1' gates the first byte to the DBI assembler (FEMDs 5-130 and 5-320). Bits 2 and 1 from tier 3 are gated as D and C bits to the DBI assembler by the 'gate read data byte 2' signal.

SIO IPL Operation

An SIO read instruction starts the read operation because Q byte bit 7 of the I-Q register is set. The 'execute primary' and the 'execute read' latches are set during the I-R cycle (FEMD 5-080). If bit 4 of the I-Q register is set, the secondary hopper is selected. Because bit one of the SIO instruction control code equals 1, the I-R cycle 'SIO command sample' signal sets the 'execute IPL' latch (FEMD 5-320). A normal read operation is executed except that bits 8, 4, 2, and 1 are ORed with the D and C positions and gated to the DBI assembler (FEMD 5-320).

Program Load Key Operation

Pressing the program load key on the CPU operator panel starts the IPL operation. The CPU sends a 'system or power reset' and 'data bus out 7' signals to the attachment (FEMD 5-325). When the program load key is pressed and released, the 'IPL command stored' latch is set by a CPU 'clock 5' pulse. The MRDAR LSR select levels are sent to the CPU. Selecting the MRDAR at this time results in resetting the MRDAR to 0000 by a system reset. The 'MFCU IPL command' signal results in setting I-Q register bit 7 and 'execute IPL' latch. The 'execute primary' and the 'execute read' latches are set after setting the I-Q register bit (FEMD 5-320) because the 'IPL command' latch was set.

An IPL read operation is then executed the same as the normal read, except that bits 8, 4, 2, and 1 are ORed with the D and C positions and gated to the DBI assembler (FEMD 5-320). After reading of the card is completed, 'channel 1 I-O condition B' is sent to the CPU (FEMD 5-325). Sending condition code B allows the CPU to start instruction execution beginning from core storage location 0000.

NON-PROCESS-RUNOUT (NPRO) OPERATION

- Clears cards from the MFCU transport and sends cards to stacker pocket one.
- Resets feed check indicators.
- Both card feed paths must be *not ready* and one or both hoppers must be empty.
- The NPRO key must be pressed two times to clear both card feed paths.
- Cards must be manually removed from all card lever cells.

When studying the NPRO Operation, refer to:

1. NPRO Operation Flowchart—FEMDs 5-275.
2. NPRO Operation Circuit Diagrams—FEMD 5-280 and 5-290.

The NPRO operation clears cards from the 5424 MFCU primary or seconary card feed paths. Cards are sent, from both card feed paths, to stacker pocket one. An NPRO operation also resets any feed check indicators that may be on.

For the NPRO key to be active, both card feed paths must be be *not ready* and one or both hoppers must be empty. Any cards that cover the photo cell sensors must be manually removed.

The MFCU motor must be stopped before pressing the NPR NPRO key. If both hoppers are not ready, one or both hoppers are empty, and no cards are in the card feed transport, pressing the NPRO key causes the 'execute NPRO time' latch to set on the next MFCU clock A signal (FEMD 5-280). Setting the 'execute NPRO time' latch starts the MFCU motor and energizes stacker pocket one magnet. Setting this latch also resets the 'NPRO SS' latch generating a reset signal until the next 'MFCU clock C' pulse. The 'NPRO SS' latch is normally set.

When the 'NPRO SS' latch is set, the 'NPRO select' latch selects punch register pressure roll one or two (FEMD 5-290). The table on FEMD 5-280 shows hopper status and 'NPRO select' latch status after an NPRO reset.

The card is ejected from the selected wait station through the punch station to the corner station. From the corner station, the card goes through the print station to stacker pocket one. (See FEMDs 4-140 and 5-280 for stacker pocket one selection.)

When the nine stage time delay counter equals count 8, the 'execute NPRO time' latch resets stopping the MFCU motor and de-energizing stacker selector one magnet.

The NPRO key must be pressed the second time to empty the other card feed path.

MOTOR START-STOP OPERATION

- An SIO instruction starts the MFCU motor.
- An NPRO operation starts the MFCU motor.
- A feed check condition, a power-on reset, a no command time out, a not ready condition, and an interlock cover open stops the MFCU motor (FEMD 5-295).

Refer to motor control circuits shown on FEMDs 5-295, 5-297, and 5-300 when studying motor start-stop operations.

When power is applied to the system, a power-on-reset resets the 'motor on' latch. When an accepted MFCU SIO instruction is executed, the 'motor on' latch is set, which in turn starts the MFCU motor (FEMD 5-295).

The nine stage time delay counter outputs are used to allow the MFCU motor to reach operating speed before the SIO operation starts (FEMD 5-300). This counter is reset each time the 'execute feed' latch is set. Counting then begins again from zero. The 'motor ready one' latch is set when the nine stage time delay counter reaches three. Between count three and count five, the 'rattle-punch-rattle' signal operates the punch magnets about 15 times before feeding a card. This operation occurs regardless of the operation to be performed. Operating the punch magnets helps prevent sticking armatures because of dust or oil.

If the nine stage time delay counter reaches a count of 256, a 'no command timeout' signal resets the 'motor on' latch at 'MFCU clock A' time. The MFCU motors then stop. The counter reaches 256 to stop the motor because no SIO instruction selected the MFCU within the 30 second time limit.

An MFCU SIO instruction or an NPRO operation restarts the MFCU motor (FEMD 5-295).

The up-down counter indicates when the wait to stacker transport is clear of cards. The 'wait to stacker clear' signal allows the gate motor stop counter to be stepped by the 'hopper CB leading edge' pulses. Whenever the gate motor stop counter equals count 4, a 'gate stop' signal occurs (FEMD 5-297). The 'gate stop' signal is used to stop the MFCU motor after both hoppers go to a not ready status and after the SIO instruction command is completed. Both hoppers go to a not ready status when the following occurs:

1. The stop key was pressed.
2. Both hoppers empty during program execution.
3. An interlocked cover is opened.
4. A stacker full condition occurred.
5. A hopper check or feed check occurred.

These controls allow for an early motor stop; otherwise, the 'no command timeout' is used to stop the motor. This requires about 30 seconds.

DIAGNOSTIC READ-PUNCH OPERATIONS

- A diagnostic read stores the read check register 128 bytes above the read data.
- A diagnostic punch stores the punch check data 128 bytes above the original punch data.

To place the attachment in diagnostic read or diagnostic punch mode, the LIO Q byte bit four must equal one. The LIO Q byte N field must also select an MRDAR or an MPCAR. An SIO read or SIO punch must then follow the LIO instruction to perform the diagnostic operation.

During the LIO instruction execution, the LSR registers are loaded with the specified address the same as any LIO instruction. In addition,the 'read diagnostic' latch or 'punch diagnostic' latch is set during the LIO insturction E-B cycles (FEMDs 5-010 and 5-360). If the N field equals five, a diagnostic read is specified by setting the 'read diagnostic' latch. If the N field equals six, the 'punch diagnostic' latch is set indicating a diagnostic punch operation. An SIO instruction operation allows the diagnostic latches to modify the specified LSR so that check data will be stored 128 bytes above the original data is storage.

The read-punch-punch check counter stepping is inhibited at 'clock 6C' time during a diagnostic read or a diagnostic punch operation. This allows time for the check byte to be stored in storage. The check byte is then stored 128 bytes above the original data. In order to store the check byte, 128 is added to the LSR. A normal read or a normal punch operation is performed. The only basic difference between a normal read and a diagnostic read or between a normal punch and a diagnostic punch operation is that the check data is stored in core storage. Data is checked during the diagnostic operations.

At the end of the specified SIO operation, the 'read diagnostic' latch is reset by the 'trailing edge or hopper check' signal; the 'punch diagnostic' latch is reset by the 'end punch cycle' signal.

Diagnostic Read

When studying the diagnostic read operation, refer to:

1. "Read/Feed Operation" in this chapter.
2. Diagnostic read flowcharts—FEMDs 5-330, 5-335, 5-340, and 5-345.
3. Diagnostic read timing charts—FEMDs 5-350 and 5-355.
4. Diagnostic read circuit diagrams—FEMDs 5-360, 5-365, and 5-370.

Card feeding from the hopper to the read station is the same as for a normal read/feed operation. Reading of data starts when the leading edge of the card covers the 18 read cells. After the read cells are covered, the 'leading edge' latch is set (FEMD 5-125). The read gear emitter counter begins counting. When the read gear emitter counter equals count 46, the 'start read' and the 'allow read' latches are set (FEMD 5-125). Also, the read gear emitter counter is reset. The first card column group (columns 1, 33, and 65) is now in position to be read.

As the card column group passes over the read cells, the 'gated read sample' signals gate data into the read data register and the read check register. Three attempts are made to gate data into the read data register on read gear emitter counts of 1, 5, and 9. Three attempts are also made to gate data into the read check register on counts 3, 5, and 11 (FEMD 5-125).

When the gear counter equals 11, the 'read CS request' latch is set (FEMD 5-365). Setting this latch indicates that data has has been read from the card column group and that cycle steal requests can now begin. The 'read priority' latch is then set and the next clock 4 sends a 'CS priority bit 4' signal to the CPU. The CPU responds by sending a DBO 0 signal and a DBO 4 signal to the attachment. As a result of this response, the 'any MFCU I/O' latch and the 'read I/O cycle' latch are set (FEMD 5-365). Setting these latches result in selecting the MRDAR and activating control lines to the CPU.

The diagnostic read operation requires two cycle steal requests per card tier. During the CPU cycle resulting from the first cycle steal request, data from the read data register is stored in core storage. During the second cycle steal request, the data from the read check register is stored 128 bytes above the first data byte location. The two cycle steal requests are called (1) first diagnostic read-punch I/O cycle and (2) second diagnostic read-punch I/O cycle (FEMD 5-360). These cycles result because the 'read diagnostic' latch was set by a previous LIO instruction.

During normal read operations, the read-punch-punch check counter is stepped at clock 6C time and at clock 7 time. Therefore, the 'gate read check byte' signal is blocked during 'clock 0 thru 1' time. Because the 'clock 6C' signal is inhibited during diagnostic operations, bit 1 of the read-punch-punch check counter can then be used to control check byte data transfer and address modification after the check byte transfer.

The first data byte from the read data register is sent to the DBI assembler under control of the read-punch-punch check counter (FEMD 3-370). Odd parity is assigned to the data byte and the data is then stored in core storage as specified by the MRDAR.

After the first read data byte is sent to the CPU, 128 is added to the MRDAR. Updating the MRDAR occurs before stepping the read-punch-punch check counter (FEMDs 5-360 and 5-365). The first data byte transfer occurred during clock 0 thru 1 time. Data from the read data register and the read check register are then compared at clock 2 thru 3 time (FEMD 5-370). A cycle steal request occurred at clock 4 of a previous CPU cycle (FEMD 5-350).

The read-punch-punch check counter is advanced at clock 7 time. Because bit 1 of the counter is set, the 'gate read check byte' signal sends the first read check byte (tier 1) to the DBI assembler during the next clock 0 thru 1 time. During address modification of the second diagnostic read-punch I/O cycle, 96 is subtracted from the MRDAR because the 'modify by 64' and 'modify by 32' signals are sent to the CPU along with the 'channel binary subtract' signal (FEMDs 5-340 and 5-365). This prepares the LSR to gate tier two data into the correct storage address. Read checking is again performed.

The read-punch-punch check counter is advanced. Another cycle steal request is sent to the CPU. After the CPU responds to this cycle steal request, the second data byte (tier 2) is stored. The next cycle steal request results in storing the read check byte (tier 2). Read checks are performed after each byte transfer. After the second check byte from tier 2 is transferred, 96 is subtracted from the MRDAR to prepare the LSR for data byte transfer from tier 3.

The data byte from tier 3 is then sent to the CPU. During the transfer of the check byte from tier 3, the 'last cycle' latch is set. The 'read CS request' latch is reset which inhibits cycle steal requests until after the next card column group is read. Setting the 'last cycle' latch which results in activating the 'modify by 128', 'modify by 31', 'modify by 32', and 'channel binary subtract' lines (FEMD 5-365). This results

in subtracting 191 from the MRDAR. The MRDAR now contains the address for the next card column group to be stored.

The card moves to the next card column group. When the read gear emitter counter again equals count 11, three more data bytes and three check bytes are stored in core storage. Data transfers and address modifications occur until all 32 card column groups are stored. Storage now contains data from the read data register and data from the read check register.

The ending operation is the same as for a normal read/feed. The 'read diagnostic' latch is reset after the 'trailing edge' latch sets (FEMDs 5-345 and 5-360). A hopper check also resets this latch. The card is then sent to the wait station and the read circuits are restored to their starting conditions (FEMD 5-345).

Diagnostic Punch

When studying the diagnostic punch operation, refer to:

1. "Punch/Feed Operation" in this chapter.
2. Diagnostic punch flowcharts—FEMDs 5-380, 5-385, and 5-390.
3. Diagnostic punch timing charts—FEMDs 5-395 and 5-400.
4. Diagnostic punch circuit diagrams—FEMDs 5-405, 5-410, and 5-415.

The card is registered in the punch station the same as for a normal punch feed operation. After the card is registered, the punch control circuits wait for the 'punch pick CB' pulse after the punch step counter equals count 4. The 'punch CB CS time' latch is then set. The next 'punch check CB' sets the 'punch check CS request' latch.

Three data bytes are sent to the attachment the same as described in "Punch/Feed Operation" in this chapter. Data is punched in the card and data is stored in the MFCU punch check circuits.

Because the 'punch diagnostic' latch is set, the read-punch-punch check counter steps at clock 7 time only. This results in sending the data check bytes to the CPU twice. The first data byte sent to the CPU is compared against the contents of the original data punched in the card. The second data check byte sent to the CPU is stored 128 bytes above the original data.

When the read-punch-punch check counter equals count 1 with bit 1 off, the attachment circuits perform a punch check cycle because the 'first diagnostic read-punch check I/O cycle' line is active. When bit one of the read-punch-punch check counter is set, the second diagnostic read-punch check I/O cycle circuits are activated (FEMD 5-360). Because a diagnostic punch operation is specified, 128 is added to the MPCAR during the first diagnostic read-punch I/O cycle.

During the second read-punch check I/O cycle, the punch check data is sent to the CPU core storage. This data is stored 128 bytes above the original punch data. The 'channel store data' line is active during the second diagnostic read-punch check I/O cycle. Also during the second diagnostic punch check I/O cycle, 96 is subtracted from the MPCAR (FEMDs 5-340 and 5-410). The MPCAR address now specifies the storage location for the second byte (tier 2). The second byte is checked and stored in core storage the same as the first check byte.

During the second diagnostic punch check I/O cycle of tier 3 (byte 2) transfer, the last cycle latch is set. Setting this latch results in subtracting 191 from the MPCAR. The 'modify by 128', 'modify by 32', 'modify by 31', and 'modify minus' signals are sent to the CPU for this address modification.

The card moves to the next card column group. Data is punched, checked, and the punch check data stored in storage.

While the last check byte is being stored in core storage, the 'last cycle' latch is set. The 'punch check CS request' latch is then reset (FEMD 5-405).

The punch control circuits then wait for punch step count 37. After the punch step counter equals 37 the 'punch check CS time' latch is reset. The punch stepper pressure roll magnet is energized and the punch eject pressure roll magnet is de-energized. The card then advances to the corner station. The card is then sent to a stacker pocket. The punch controls are then restored to their starting condition.

A card is either read or fed and sent to the vacated wait station during the diagnostic punch operation (FEMDs 5-120 and 5-170). Also, refer to the "Read/Feed Operation" in this chapter.

There are no features applicable to the MFCU attachment.

Chapter 5. Power Supply and Control

Power is applied to the attachment circuits from the CPU power supplies.

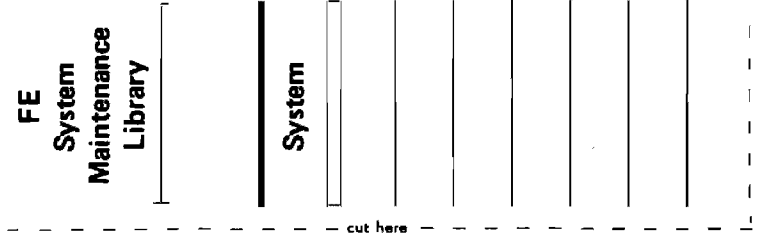
Refer to *IBM 5424 Multi-Function Card Unit Field Engineering Theory of Operation,* Form SY31-0213 for MFCU console and maintenance features.

Appendix A. Machine Characteristics

Not applicable.

Appendix B. Special Circuits

There are no special circuits in the MFCU attachment circuits.



SY31-0253-0

IBM

International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N.Y. 10601

IBM System/3 Printed in USA SY31-0253-0