B 1900 SYSTEM

TECHNICAL MANUAL VOLUME 3:

THEORY OF OPERATION

Burroughs

FIELD ENGINEERING

F.E. Dist. Code BB

Printed in U.S. America

November 1980

Library Binder 304A

Form 1127396

Burroughs believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document should be addressed directly to Burroughs Corporation, P.O. Box 4040, El Monte, California 91734, Attn: Publications Department, TIO–West.

LIST OF EFFECTIVE PAGES

Page	Issue	Page	Issue
Title	Original	2-30	Blank
ii	Original	2-31	Original
iii	Original	2-32	Blank
iv	Blank	2-33 thru 2-66	Original
v thru xi	Original	3-1	Original
xii	Blank	3-2	Blank
1-1	Original	3-3	Original
1-2	Blank	3-4	Blank
2-1 thru 2-21	Original	3-5 thru 3-11	Original
2-22	Blank	3-12	Blank
2-23 thru 2-27	Original	3-13 thru 3-46	Original
2-28	Blank	4-1 thru 4-33	Original
2-29	Original	4-34	Blank
		A-1 thru A-10	Original

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont)

LIST OF TABLES

LIST OF TABLES (Cont)

SECTION 1
BASIC PRINCIPLES

This FETM includes information on B 1905 and B 1955 single-processor systems as well as the B 1985 dual-processor system.

Material that would ordinarily be included in this section is adequately covered in the B 1900 Central Systems Technical Manual, Volume 1, Operation and Maintenance, Form No. 1127388.

Other material pertinent to B 1900 Central Systems is included in the following three manuals:

1. B 1800 Series Central Systems Technical Manual, Volume 1, Operation and Maintenance; Form No. 1098282.
2. B 1800 Series Central Systems Technical Manual, Volume 3, Theory of Operation; Form No. 1095551.
3. B 1700 I/O Base Technical Manual; Form No. 1053352.

SECTION 2
CIRCUIT OPERATIONAL DETAIL

INTRODUCTION

This section provides detailed descriptions of the circuits and operation of the B 1900 processor, S-Memory, and Host Adapter. The processor description includes discussions of the console interface logic, the system clock, the microinstruction fetch-decode-execute structure, Cache Memory, the processor registers, the 4-bit and 24-bit function boxes, and the I/O and S-Memory interfaces.

Also presented is a discussion of the I/O Distribution Card and signal distribution within the I/O Base. Not described are the circuits and functions of the individual I/O controls, which are described in individual technical manuals.

CONSOLE INTERFACE LOGIC

The console controls represent an extension of the processor control logic that allows manual selection of micro-operator functions. The console provides an operator interface with the processor and permits overall control of system operation, monitoring of processor activity, and maintenance/testing procedures. The majority of actions that can be initiated from the console can also be specified by microinstructions.

The console controls are grouped into two panels: the Operator (Op) panel and the Diagnostic/Maintenance (D/M) panel. In the B 1985 and B 1955, all six push buttons (POWER, INTRPT, CLEAR, START, MODE, REWIND) of the Op panel are in a strip in the top right corner of the cabinet. In the B 1905, the Op panel is distributed: the POWER and INTRPT push buttons are on the left side of the ODT keyboard and the other four push buttons are adjacent to the cassette drive unit, which is behind the front access door and above the D/M panel.

Operator Panel

The Op panel provides basic functional access to the processor. The panel includes six function switches with integral indicators:

Switches	Indicators
POWER	(red light)
INTRPT *	RUN
CLEAR *	
START *	TEST STATE
MODE	NORMAL/MTR
REWIND	BOT

* Duplicated on D/M panel

Functions that are duplicated on the D/M panel are discussed in the subsection titled Duplicate Console Functions.

TEST STATE Indicator

A lighted TEST STATE indicator in the START push button indicates that the processor is in a state other than the normal execution state. Several conditions can enable the TEST STATE indicator:

INTERRUPT switch on D/M panel is up. (CC register bit 3 = 1.)
SINGLE MICRO/NORMAL switch is at SINGLE MICRO.
MICRO SOURCE switch is not at NORMAL.

MODE Push Button, NORMAL/MTR Indicators

The MODE push button allows the operator to select the processor's operating mode, and the indicators NORMAL/MTR show the current mode. In NORMAL mode, processor logic is configured to accept microinstructions from Cache or S-Memory; in MTR mode, the logic is configured to accept microinstructions from cassette tape.

If MODE is pressed while NORMAL is lighted, the MTR mode is entered. Normally-open contacts close and +12V is applied to the MTR lamp. Also, a TRUE level is removed from the NORMAL net. This net is routed through the cassette logic and D/M panel logic where it is ORed with the SINGLE MIC/NORMAL switch. If this switch is at SINGLE MIC, data movement from cassette to processor is inhibited. SINGLE MIC puts the processor into step operation; that is, each time START is pressed, one micro is executed and the processor halts. NORMAL on the SINGLE MIC/NORMAL switch signifies continuous micro execution.

If MODE is pressed with MTR lighted, processor logic is returned to the normal configuration and micros are accepted from Cache or S-Memory, as specified by the MICRO SOURCE switch or the MSSW register.

BOT/RWD Push Button/Indicator

Pressing BOT/RWD provides a signal to rewind the magnetic tape cassette. The BOT indicator is off if the tape is not completely rewound and goes on when the tape reaches the beginning-of-tape marker and stays on until the tape moves off the marker.

Diagnostic/Maintenance Panel

The D/M panel (Figure 2-1) is used primarily for diagnosing system failures, testing circuit functions, and making temporary software changes. The panel is available for use by system operators and programmers as well as field engineers. Figure 2-2 shows how the D/M panel is cabled into the processor.

The D/M panel provides the following functions:

All Systems

24 Console Switches
24 Console Lamps
Push Buttons: HALT, CLEAR, START, LOAD, INC
Toggle Switches: INTERRUPT, SINGLE MIC/NORMAL
Rotary Switches: REGISTER GROUP, REGISTER SELECT, MICRO SOURCE
Indicators: STATE, RUN, ERROR, OVER TEMP

B 1905

Rotary Switch: DDEC (ON LINE/OFF LINE)

B 1955 and B 1985

Rotary Switches: MASTER SELECT, SLAVE, DISPLAY
Push Button Indicator: DDEC (ONLINE/OFFLINE)
Indicators (second set): STATE, RUN, ERROR

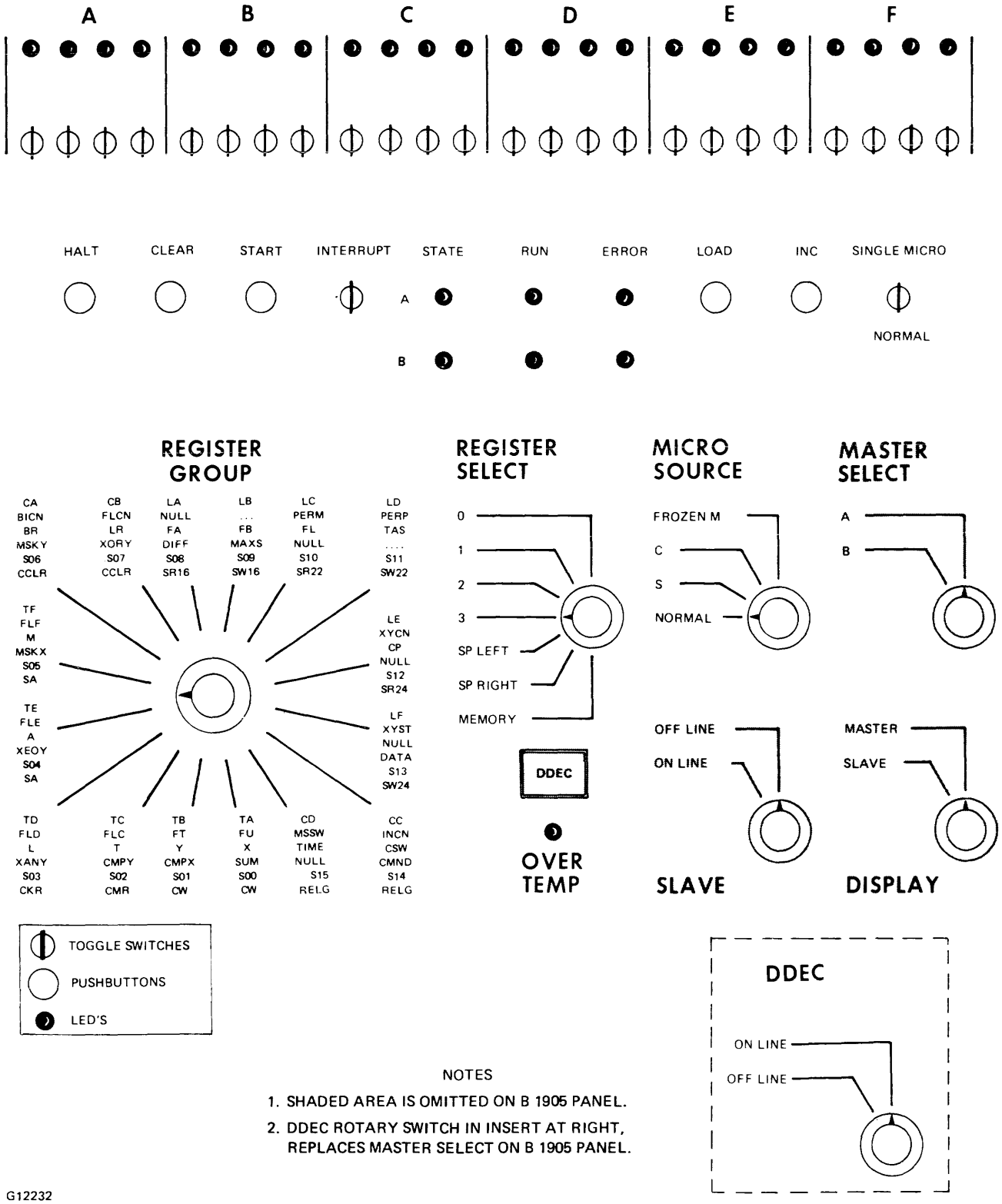


Figure 2-1. B 1900 Diagnostic/Maintenance Panel

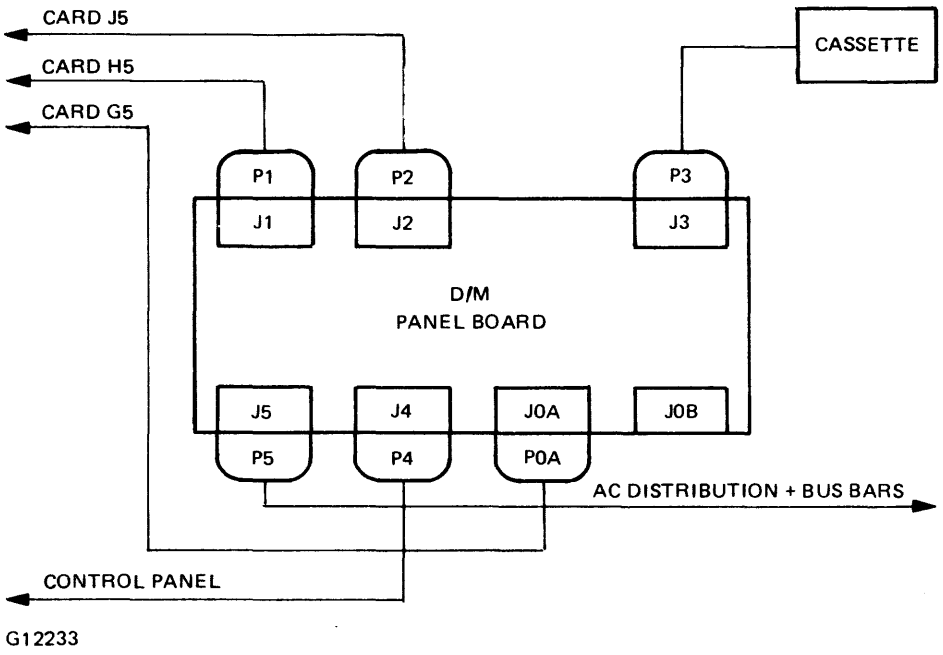


Figure 2-2. D/M Panel Cabling

Console Lamps

The 24 console lamps are light-emitting diodes (LEDs) that can display the contents of selected registers and memory locations when the processor is halted.

The enable signal for the console lamps is a nanoregister output, a signal that allows the contents of the MEX to be latched and routed to the console lamps.

Console Switches

The 24 console switches are 2-position toggles that provide a TRUE output in the up (ON) position. The switches are used to enter data for loading registers or to provide data for console writes to Cache or S-Memory.

Figure 2-3 illustrates the means by which the console switches enable data to the processor through card J. For simplicity, the logic for only one switch is shown. When a console switch is moved to the up (ON) position, a TRUE is applied to the 4-to-1 multiplexer on card J. Select lines and output control lines determine which input is gated to the multiplexor outputs.

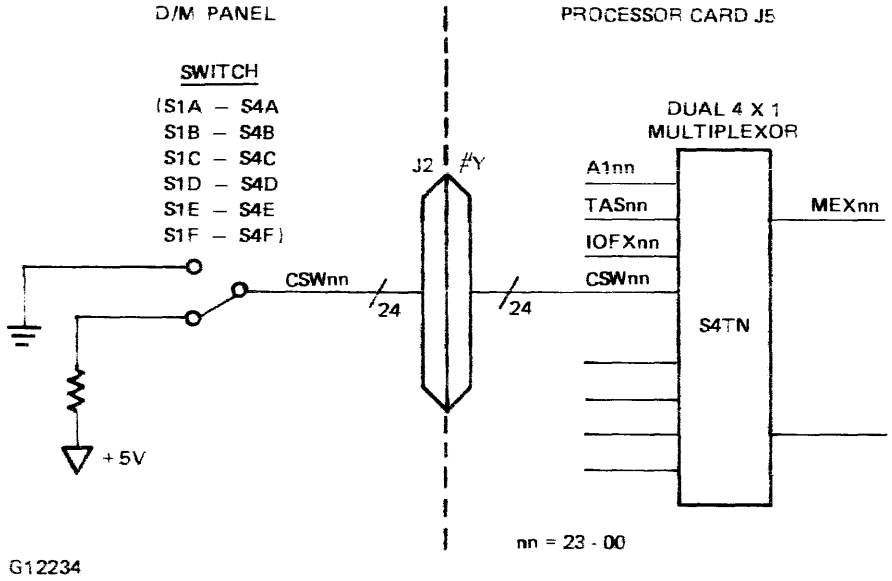


Figure 2-3. Console Switch Logic

REGISTER GROUP and REGISTER SELECT Switches

These two rotary switches are used together to provide selection of a total of 112 (16 X 7) possible conditions. REGISTER GROUP (RG) has 16 functional positions. The position labeled TA is considered position 0; clockwise rotation takes the switch through positions 1 to 15. REGISTER SELECT (RS) has seven positions, 0 through 6 reading down. Table 2-1 lists the specific entities and functions available with each combination of settings of the two switches.

Table 2-1. REGISTER GROUP/REGISTER SELECT Switch Combinations

REGISTER GROUP Switch Position	REGISTER SELECT Switch Position						
	0	1	2	3	SP LEFT	SP RIGHT	MEMORY
0	TA	FU	X	SUM	S00A	S00B	CW
1	TB	FT	Y	CMPX	S01A	S01B	CW
2	TC	FLC	T	CMPY	S02A	S02B	CMR
3	TD	FLD	L	XANY	S03A	S03B	CKR
4	TE	FLE	A	XEOY	S04A	S04B	SA
5	TF	FLF	M	MSKX	S05A	S05B	SA
6	CA	BICN	BF	MSKY	S06A	S06B	SA
7	CB	FLCN	LR	XORY	S07A	S07B	CCLR
8	LA	NULL	FA	DIFF	S08A	S08B	SR16
9	LB	...	FB	MAXS	S09A	S09B	SW16
10	LC	PERM	FL	NULL	S10A	S10B	SR22
11	LD	PERP	TAS	....	S11A	S11B	SW22
12	LE	XYCN	CP	NULL	S12A	S12B	SR24
13	LF	XYST	NULL	DATA	S13A	S13B	SW24
14	CC	INCN	CSW	CMND	S14A	S14B	RELG
15	CD	MSSW	TIME	NULL	S15A	S15B	RELG

All possible combinations of positions 0, 1, 2, and 3 of RS and positions 0-15 of RG define all the addressable registers and allow them to be selected for display or modification. The binary values of the RG/RS combinations also are used in certain micros to address specific registers. While the processor is in HALT, the contents of the particular register selected are displayed in the console lamps and the register is available for loading from the console switches.

Combinations of positions 4 (SP LEFT) and 5 (SP RIGHT) of RS with positions 0-15 of RG address each of the 16 left and 16 right Scratchpad words.

Combinations of position 6 (MEMORY) of RS and positions 0-15 of RG enable certain Cache and S-Memory operations and allow examination of the Error Log (ELOG) register.

Figure 2-4 is a schematic representation of the REGISTER GROUP switch and Figure 2-5 shows the REGISTER SELECT switch logic. Switch positions are interpreted binarily: the four lines from RG provide 0000-1111 and the three lines from RS provide 000-111. The output lines are gated to PROMs for decoding into the various control lines that perform the function or address the register specified by the switches.

These switches are discussed in greater detail in the subsection titled LOAD Push Button.

HALT Push Button

HALT is a momentary push button that is used to halt the system. The processor, upon receipt of the halt signal, completes the micro presently in the execute phase and comes to an orderly halt. Figure 2-6 shows the logic of the HALT push button circuit

In the figure, the switch (S5) is shown in its normal (RUN) position. Pressing HALT applies a LOW to an RS-type flip-flop which causes signal HALT/... to go LOW, setting the Halt Request to halt the processor after execution of the current micro.

STATE Indicator

The STATE indicator, when lit, indicates that bit 03 of the CC register is set. Control of this lamp is dependent entirely on the software, which must programmatically control the setting or re-setting of bit 03.

RUN Indicator

The RUN indicator is used to show that the processor is in the RUN state in NORMAL mode or that the cassette is being driven forward in the MTR mode. The lamp is driven by a term called FDL+RNG1 from card G5. Figure 2-7 illustrates the logic for the RUN indicator.

ERROR Indicator

The ERROR indicator lights as the result of the detection of any one of several errors that turn on one or more bits in the PERM and PERP registers. PERP and PERM bits have the following meanings:

PERP bit	3	Always reset (0)
	2	Cache Key parity error
	1	M-Register parity error
	0	Uncorrectable cassette error
PERM bit	3	Microinstruction time-out
	2	S-Memory field out of physical bounds
	1	S-Memory Error Log register change
	0	Uncorrectable S-Memory error in processor operation

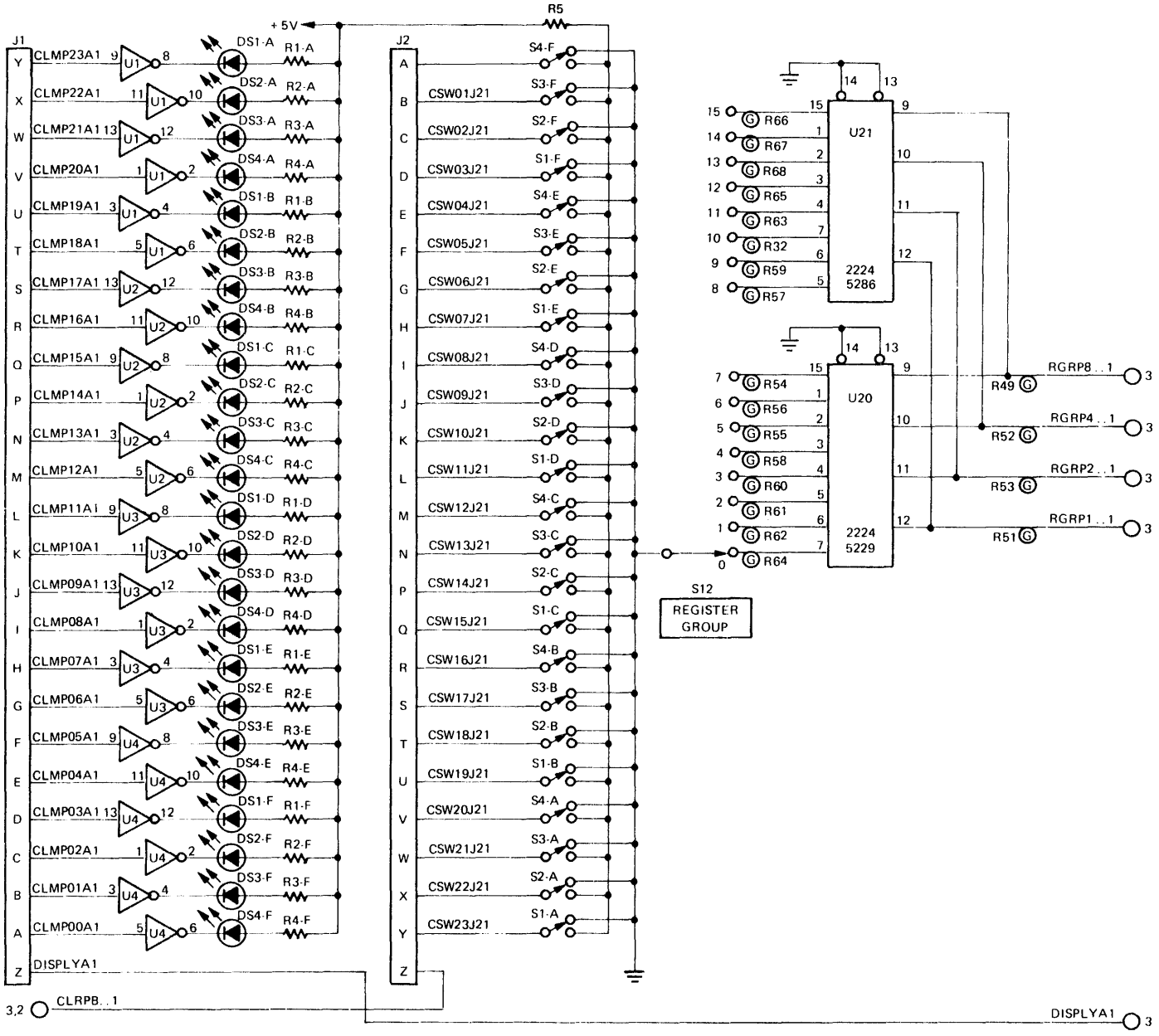


Figure 2-4. REGISTER GROUP Switch Schematic

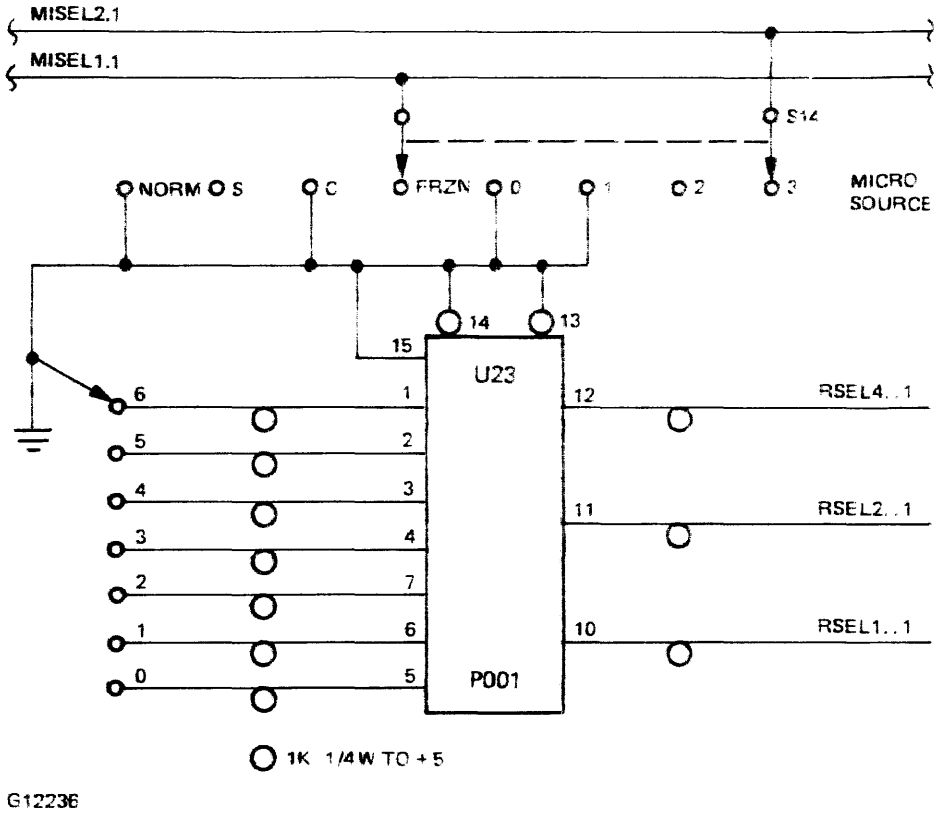


Figure 2-5. REGISTER SELECT Switch Logic

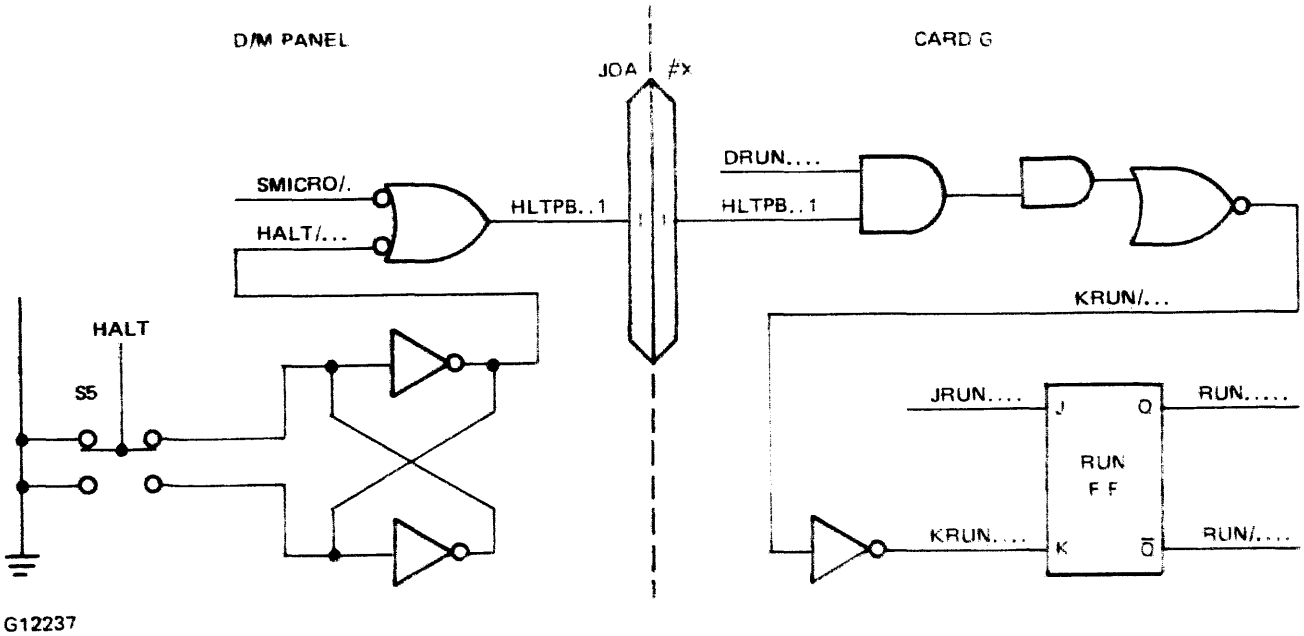


Figure 2-6. HALT Push Button Logic

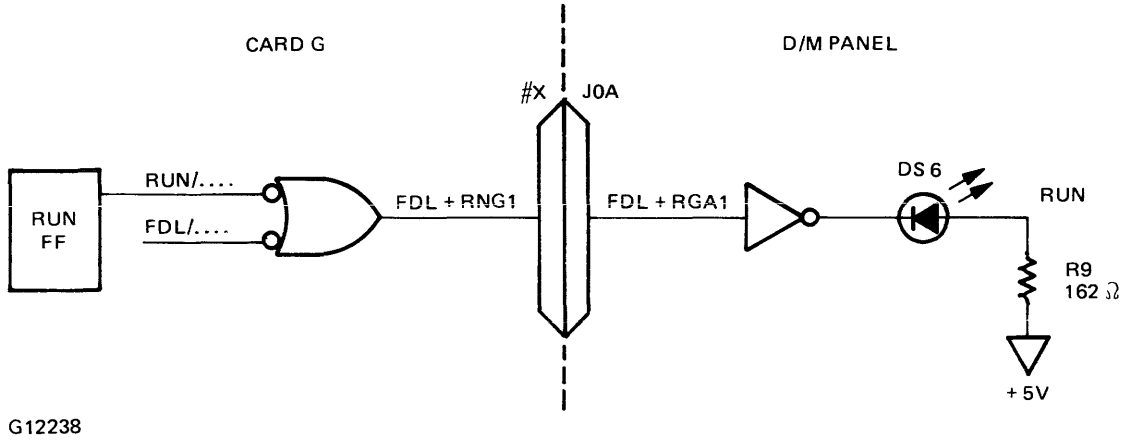


Figure 2-7. RUN Indicator Logic

The processor comes to an immediate halt if PERP bit 2 or 1 or 0 or PERM bit 3 goes TRUE, or if PERM bit 2 goes TRUE on a micro fetch operation. The processor comes to a controlled MCP halt if PERM bit 2 or 0 goes TRUE on a data operation. When PERM bit 1 goes TRUE, the Error Log is updated.

Figure 2-8 shows the ERROR indicator control logic. Note that each of the PERM and PERP register bits is an input to a PROM. The presence of a TRUE level on any of these lines results in the indicator lighting when the processor halts, that is, term RUN..... goes LOW.

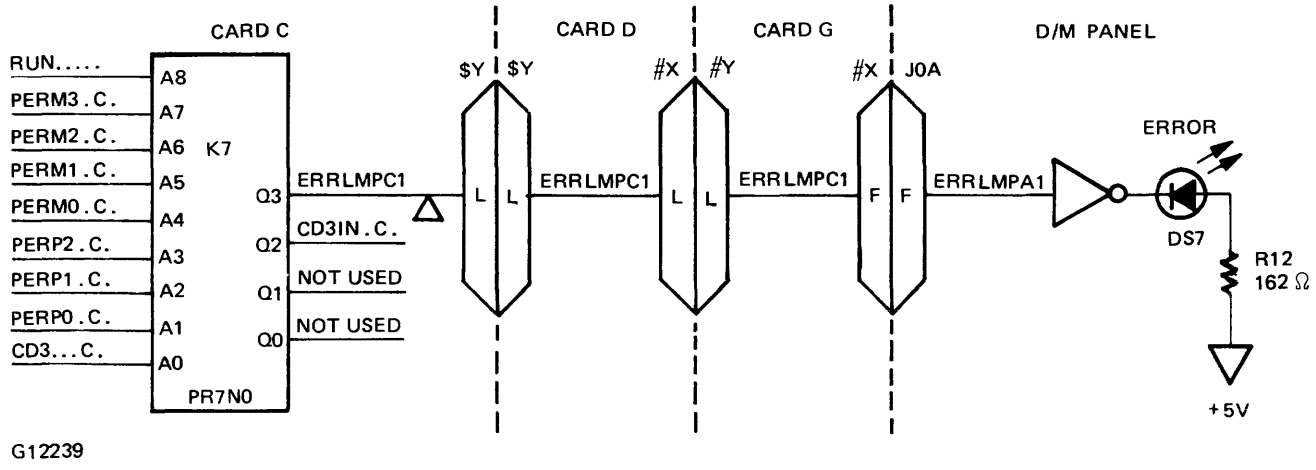


Figure 2-8. ERROR Indicator Logic

LOAD Push Button

LOAD is a momentary-contact push button with one output line. This output is used to determine the proper system reaction, which depends on the settings of the REGISTER GROUP/REGISTER SELECT switches. Pressing LOAD causes the actions specified in Tables 2-2. Table 2-3 describes the contents of the console lamps after S-Memory or Cache locations have been selected by using LOAD. Table 2-4 includes a description of the use of LOAD to clear Cache.

The term LDATAA/1, enabled when LOAD is pressed, serves as an input to the PROMs on card G that are used for decoding D/M switch settings.

Figure 2-9 shows the logic for the LOAD push button.

Table 2-2. LOAD Push Button: Console Switch Contents Loading

RS	RG	ID	Pressing LOAD loads switch contents to
0-3	all	** *	The specified register. Exceptions: DATA, CMND
3	13	DATA *	The I/O Bus; RC is issued.
3	14	CMND *	The I/O Bus; CA is issued.
4, 5	all	– *	The Scratchpad location selected.
6	0-1	CW	The Cache location specified by the A register. (Switches contain 16 data bits.)
6	6-7	CCLR	(See Table 2-4.)
6	8-9	SW16	The S-Memory location specified by FA. (Switches contain 16 data bits.)
6	10-11	SW22	The S-Memory location specified by FA. (Switches contain ECC + 16 data bits.)
6	12-13	SW24	The S-Memory location specified by FA. (Switches contain 24 data bits.)
6	14-15	RELG	(See Table 2-3.)

NOTES

Loading takes place when LOAD is pressed with the processor in HALT.
RS = REGISTER SELECT switch
RG = REGISTER GROUP switch
ID = Applicable RG switch label at the setting specified

* The contents of these locations are displayed in the console lamps as long as they are selected.

** The following cannot be loaded: BICN, FLCN, XYCN, XYST, INCN, TAS, CSW, TIME, SUM, CMPX, CMPY, XANY, XEOY, MSKX, MSKY, XORY, DIFF, MAXS, U.

Table 2-3. Displaying Cache, S-Memory, and ELOG After Using LOAD to Load Addresses

RS	RG	ID	Console lamps display
6	2	CMR	The micro plus parity in the Cache location specified by the A register. (Rightmost 17 lamps.)
6	3	CKR	The Cache Key in the Cache location specified by the A register. (Bits 22, 20, 8-0.)
6	8-9	SR16	16 bits (data) of FA-specified S-Memory location.
6	0-11	SR22	22 bits (ECC + data) of FA-specified S-Mem location.
6	12-13	SR24	24 bits (data) of FA-specified S-Memory location.
6	14-15	RELG	ELOG contents. ELOG is cleared; display remains.

NOTES

Display is visible with processor in HALT.

RS = REGISTER SELECT switch
RG = REGISTER GROUP switch
ID = Applicable RG switch label at the setting specified

Table 2-4. Actions With REGISTER SELECT at Position 6

RS	RG	ID	Action
6	4-5	SA	Stop on A. When these settings are selected, if the contents of the A1 register become equal to the contents of the leftmost 20 console switches, the processor halts.
6	6-7	CCLR	Cache CLEAR. With processor in HALT, pressing LOAD clears Cache Memory.

NOTES

RS = REGISTER SELECT switch
RG = REGISTER GROUP switch
ID = Applicable RG switch label at the setting specified

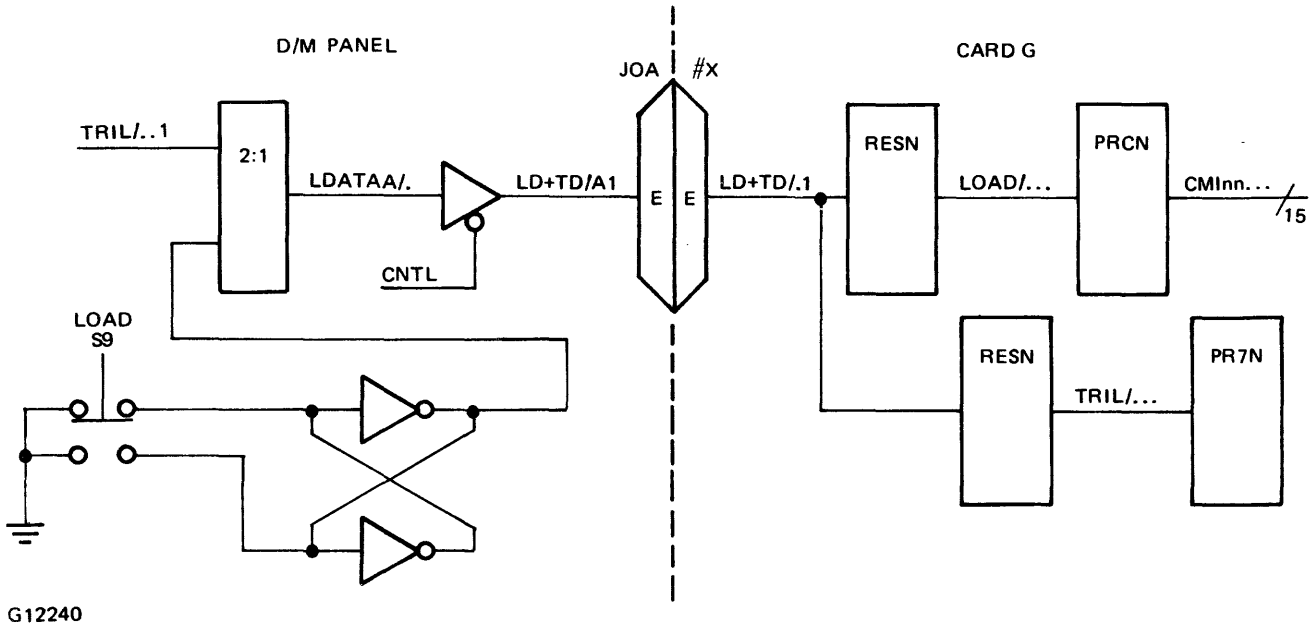


Figure 2-9. LOAD Push Button Logic

INC (Increment) Push Button

INC is a momentary-contact push button used to increment the A register or the FA register. REGISTER SELECT is set to MEMORY and REGISTER GROUP is used to select the register and the increment, as follows:

The A register is incremented by 16 when INC is pressed with the processor in HALT with RG at CW (Cache Write), CMR (Cache Micro Read), or CKR (Cache Key Read).

The FA register is incremented by 16 when INC is pressed with the processor in HALT with RG at SR16 (S-Memory Read 16), SW16 (S-Memory Write 16), SR22 (S-Memory Read 22), or SW22 (S-Memory Write 22).

The FA register is incremented by 24 when INC is pressed with the processor in HALT with RG at SR24 (S-Memory Read 24) or SW24 (S-Memory Write 24).

The logic for INC is shown in Figure 2-10. Interpretation of the REGISTER GROUP/REGISTER SELECT and INC outputs is controlled by the LOAD push button PROMs.

SINGLE MICRO/NORMAL Switch

With this switch in the SINGLE MICRO position, pushing START causes the processor to execute one micro and return to HALT. In NORMAL position, micros are successively executed. These functions are influenced by the setting of the MODE switch on the Op panel. Figure 2-11 shows this interaction.

With MODE at NORMAL, a LOW level is propagated from the Op panel through the net NORMAL. This level is applied to an RS-type flip-flop and the TRUE output is gated with the SINGLE MICRO output to generate the SMICRO/. term. This resets the RUN flip-flop that causes micro instructions to execute one at a time, because KRUN.... does not reset RUN until a micro is completed. When S11 is in the NORMAL position a TRUE level is generated to SMICRO/., which prevents the stepping logic from being enabled.

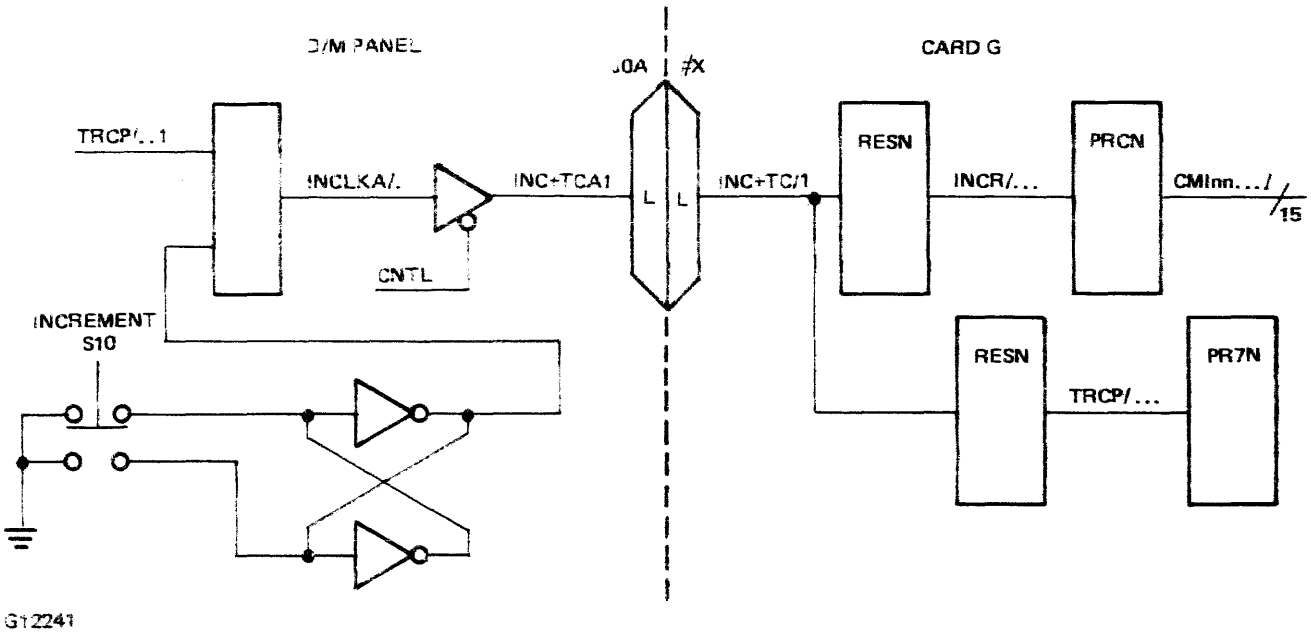


Figure 2-10. INC Push Button Logic

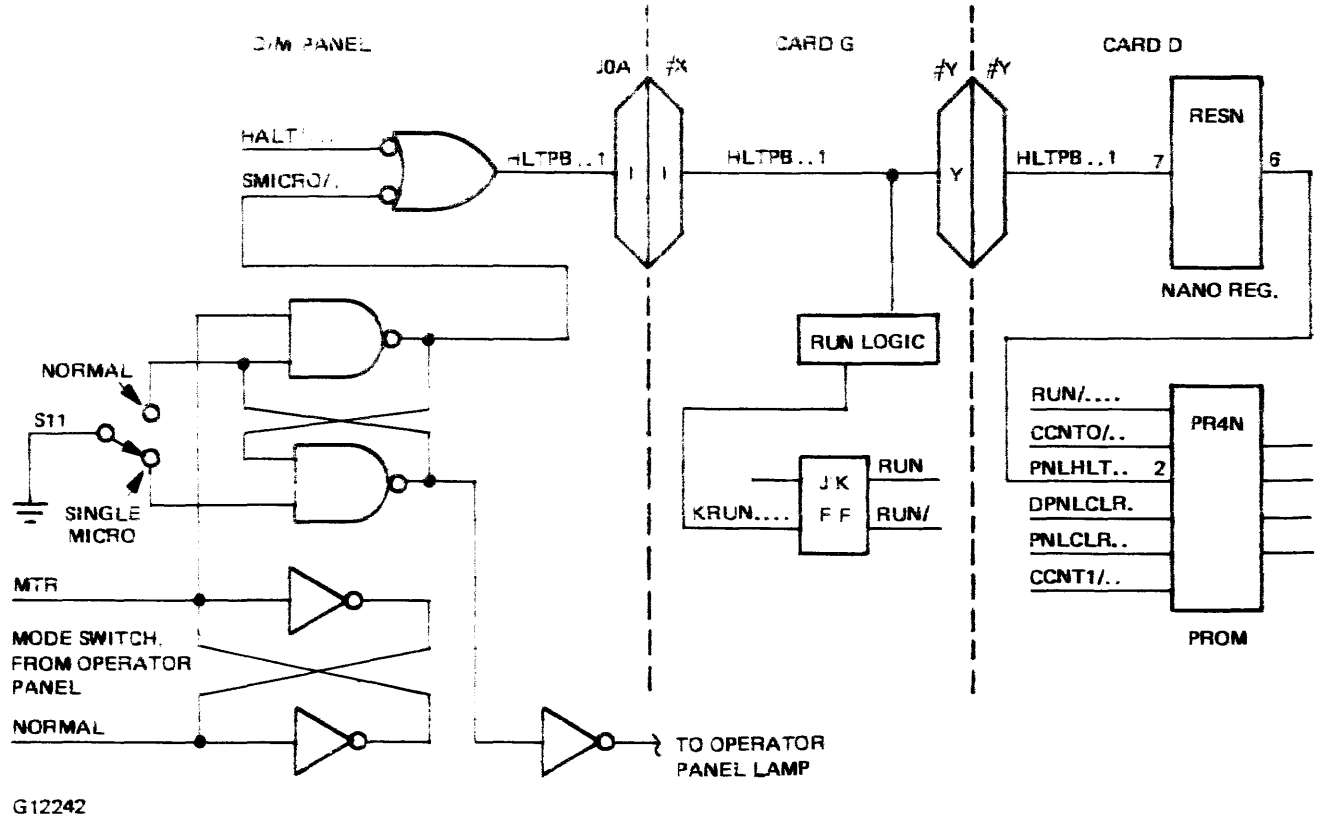


Figure 2-11. SINGLE MICRO/NORMAL Switch Logic

Table 2-5 specifies the interactions. Note that the SINGLE MICRO/NORMAL switch has no effect on the continuous execution of micros accessed from cassette tape.

Table 2-5. Interaction Between SINGLE MIC/NORMAL and MODE Switches

Single Micro	Mode	Action
NORMAL	NORMAL	Processor continuously executes micros from source designated by MICRO SOURCE switch.
SINGLE MICRO	NORMAL	Processor executes one micro each time START is pressed.
NORMAL	MTR	Processor continuously executes micros from cassette tape.
SINGLE MICRO	MTR	Processor continuously executes micros from cassette tape.

MICRO SOURCE Switch

MICRO SOURCE is a 4-position rotary switch that generates two binary encoded lines used by the processor fetch structure to specify the source of micros to the M register. Switch positions, output codes, and meanings are specified in Table 2-6.

The micro source can also be selected programmatically by the settings of bits 1 and 0 of the MSSW register. The interpretation of bits 00, 01, 10, or 11) is identical to the switch code.

The contents of the switch are bit-ORed with the register bits. The results of the logical combinations are shown in Table 2-7.

The MTR position of the MODE switch overrides all settings of the MICRO SOURCE switch.

Table 2-6. MICRO SOURCE Switch Position Codes and Meanings

Position	Code	Source of Micros
NORMAL (0)	00	Cache (S-Memory on misses)
S (1)	01	S-Memory only
C (2)	10	Cache only
FROZEN M (3)	11	M register (micro does not change)

Table 2-7. Results of Bit-ORing MICRO SOURCE and MSSW

	Register			
Switch	**00**	**01**	**10**	**11**
00	00	01	10	11
01	01	01	11	11
10	10	11	10	11
11	11	11	11	11

Over Temperature Indicator

To circulate cooling air, the cabinet is provided with a blower assembly. If a blower malfunctions, the OVER TEMP indicator provides a visual alarm. The OVER TEMP circuit is shown in Figure 2-12.

Control circuits for the indicator are contained within the AC Distribution Box. Blower air is sensed by a switch in the fan assembly. When the air supplied by the fan is sensed as insufficient, the switch closes, energizing relay K3 with 24 volts from transformer T1. The application of power picks K3 which, through its wiper, provides 24 volts to the OVER TEMP lamp and to relay K2. When K2 is picked, the 24 volts is removed from relay K1 and all AC power is removed. To restore power to the system, the POWER push button must be cycled: first OFF, then ON.

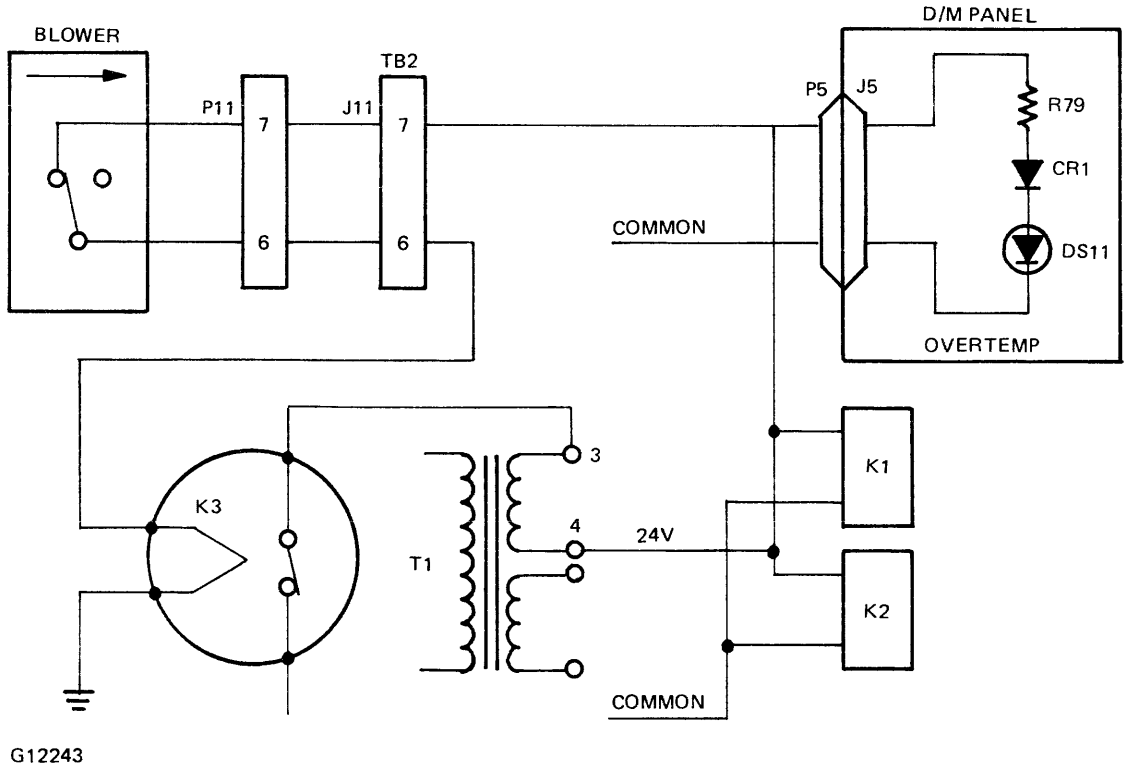


Figure 2-12. OVER TEMP Circuit Logic

DDEC Switch

A disk drive electronics controller may be included in the B 1900 central system cabinet. The DDEC switch, which is a push button in the B 1985 and B 1955 and a rotary switch in the B 1905, provides the same ON LINE/OFF LINE capability here as on a stand-alone controller, permitting the DDEC to be operated locally or under system control.

Switches for Two-Processor Systems

Three switches on the B 1985/B 1955 D/M panel apply specifically to two-processor operation: MASTER SELECT, SLAVE, and DISPLAY. The cable that connects Processor A to the D/M panel runs from J0A on the panel to Processor A, card G. Similarly, a cable from J0B on the panel is connected to Processor B, card G.

MASTER SELECT SWITCH

This two-position rotary switch enables the user to designate either Processor A or Processor B as master. The other processor is automatically designated as slave.

SLAVE SWITCH

This two-position rotary switch allows the slave processor to be placed on or off line.

DISPLAY SWITCH

This two-position rotary switch allows selection of Processor A or Processor B for display in the console lamps.

NOTE

DISPLAY may be switched with the system either in RUN or in HALT. MASTER SELECT switch and SLAVE must only be switched with the system in HALT; switching either with the system in RUN may cause an unrecoverable halt.

Duplicate Console Functions

CLEAR, START, and INTERRUPT are functions that are implemented by pushbuttons or switches.

CLEAR Push Button

CLEAR is a momentary push button that is used to clear the major working registers of the processor. Figure 2-13 shows the logic.

CLEAR is active only while the processor is in the HALT state.

Pressing CLEAR generates a TRUE level that enables CLRPB..1 to come TRUE. This signal propagates through the General Purpose Clear logic that develops the general clear signal PNLCLR*, which is routed to the rest of the processor for the following functions:

1. Enables the ALU on card E to output all zeros to the MEX for loading the registers listed in 2, below.
2. Clears registers M, A, BR, PERP, PERM, MSSW, CC, and CD by loading zeroes.
3. Clears all processor state flip-flops.
4. Clears Cache unless MICRO SOURCE is at C (Cache only).

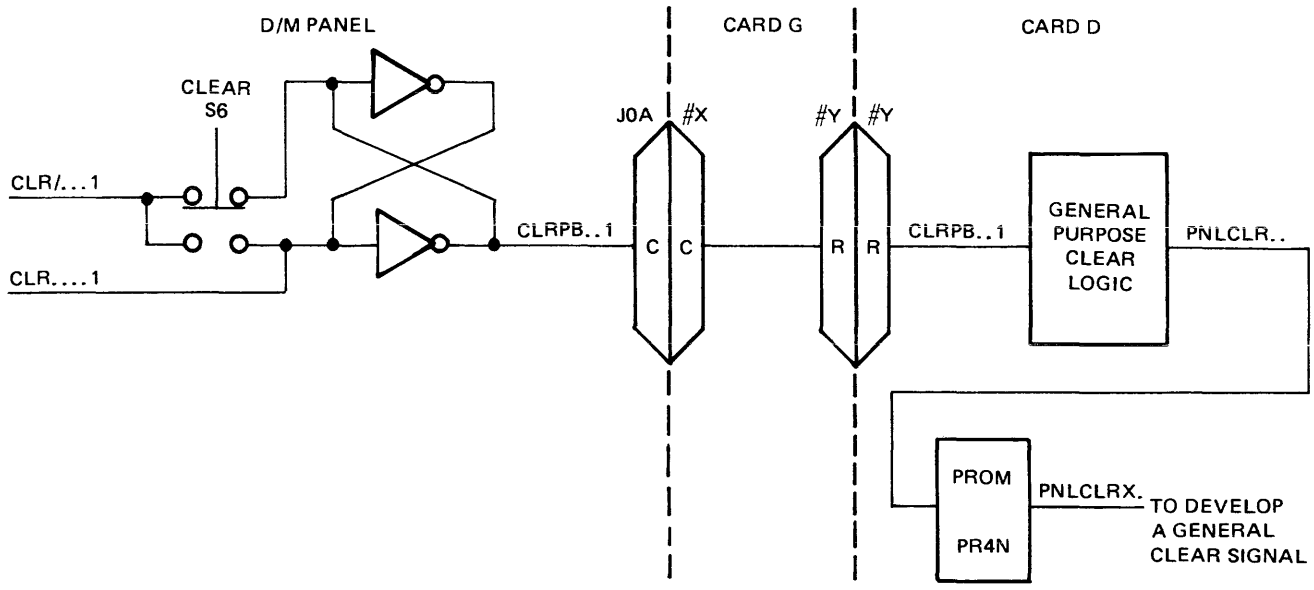


Figure 2-13. CLEAR Push Button Logic

START Push Button

The function of the START push button is to cause the processor to transfer the system from the LOAD/DISPLAY state to the RUN state. In addition, if the processor is in the MTR mode, the START push button initiates a cassette start signal. Figure 2-14 shows the logic for the START push button.

Pressing START applies a TRUE level to the input of an RS-type flip-flop and generates the START/.. term. In the B 1985 system, START/.. is ANDed with STRTAEN. and STRTBEN. to start one of the processors. The gated term is called STARTn.. and is applied to a two-input multiplexor. If FDL+RNG1 is TRUE, then INSTRTA1 becomes TRUE and propagates to the console interface logic which starts the RUN Logic.

INTERRUPT Switch

INTERRUPT, a toggle on the D/M panel and a push button on the Op panel, is used to bring the processor to an orderly, software-controlled halt. This is accomplished by setting bit 0 of the CC register. Software interrogates this bit for a 1 condition, as shown in Figure 2-15.

When INTERRUPT is operated, the INTERSW. signal is generated on the D/M Panel. This signal is ORed with the signal START at the 2-to-1 multiplexor, U18. It is then routed to card G where it is ANDed with the RUN/..G1 signal and sent to card C where it sets CC0.

SYSTEM CLOCK

Card K, the I/O Distribution and Clock card, plugs into the processor backplane. Several clock signals are generated on this card for use by the system. These signals are distributed via backplane connections and frontplane coaxial connectors. There are three versions of this card: one operates at 6 MHz (167-nanosecond cycle) and the other two at 4 MHz (250-nanosecond cycle). Clock outputs are always available when power is applied to the system.

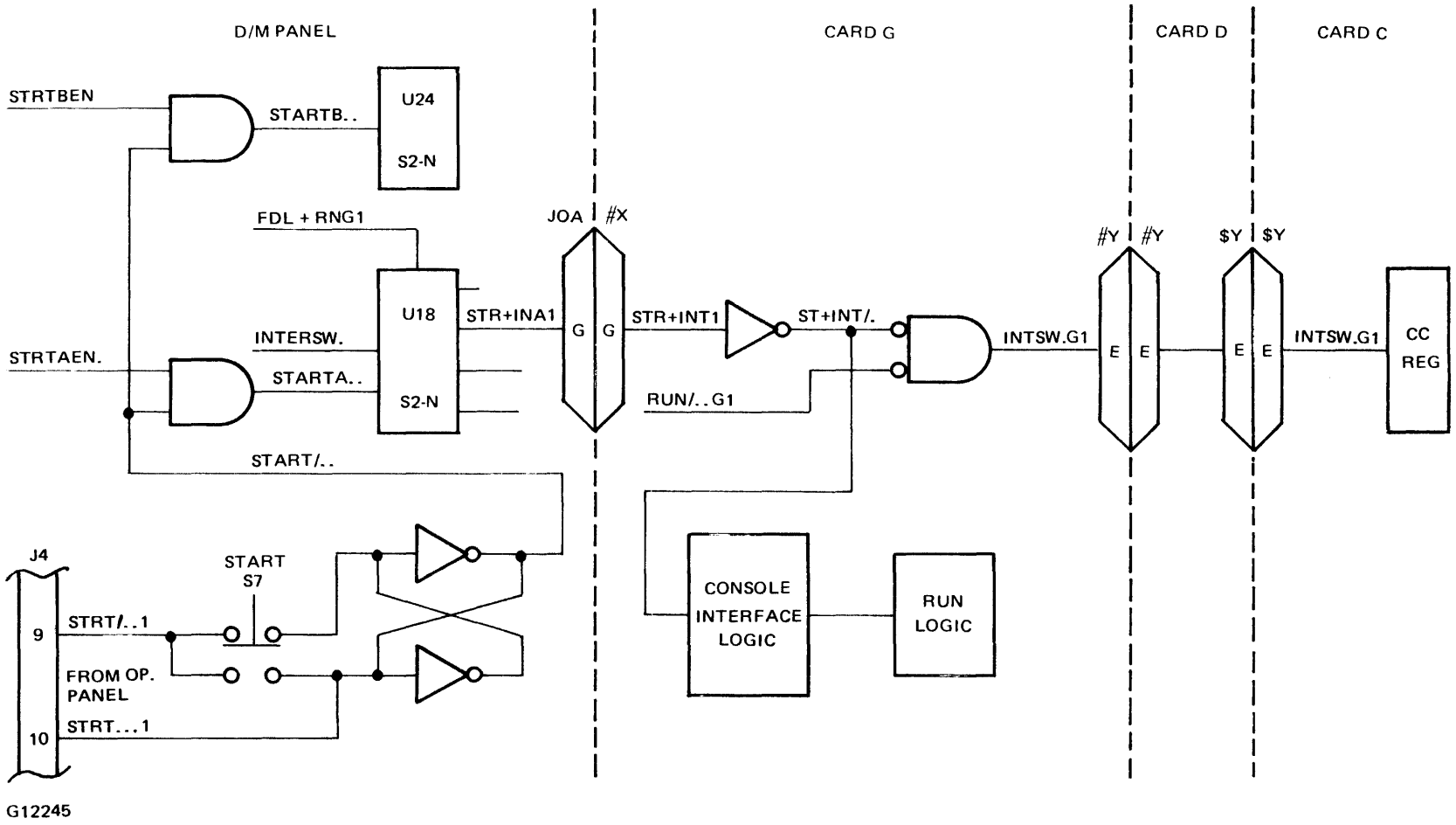


Figure 2-14. START Push Button Logic

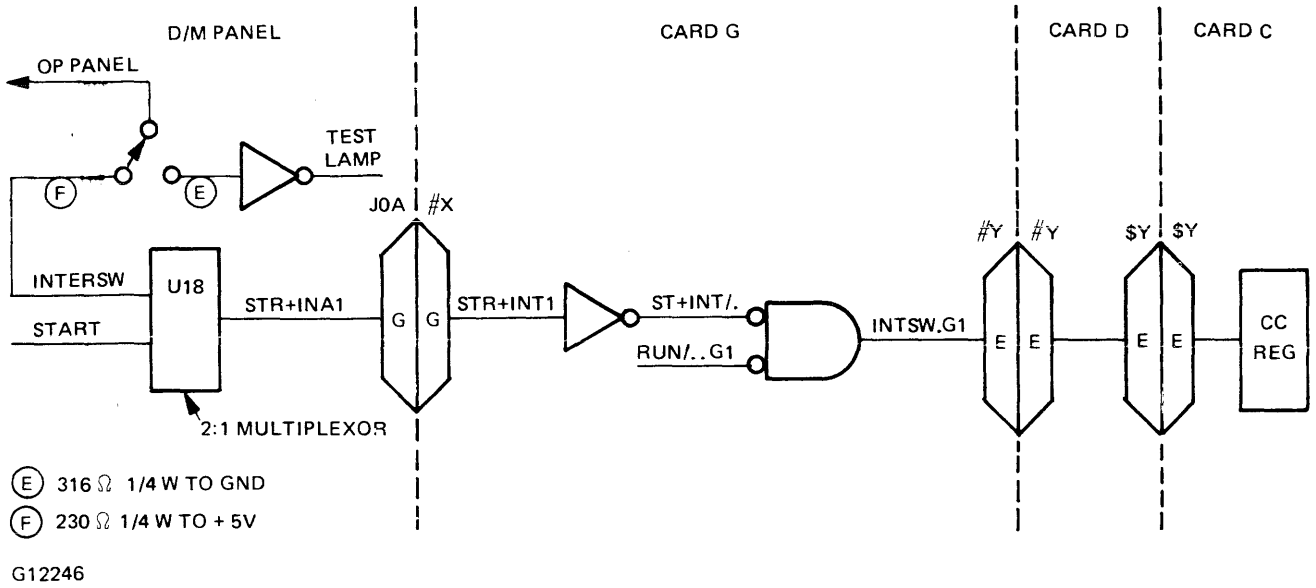


Figure 2-15. INTERRUPT Switch Logic

Figure 2-16 is a functional block diagram of the clock card, Figure 2-17 shows clock generation, calibration and distribution, and Figure 2-18 provides waveforms.

Processor Clocks

Processor clocks, signals DSCPA.KO through DSCPJ.KO, are approximately 33 nanoseconds wide. By shifting the leading edge signal LDEDGE.. and gating it with early reference signal ERLYREF., clocks of different pulse widths can be achieved. The basic processor clock PCLK.... is generated by the same method through the two delay lines, which give a total of 70 nanoseconds of adjustment range. Additional delay lines, one of 100 nanoseconds and two of 20 nanoseconds each, provide the necessary phase delay. Processor clocks are buffered by separate LD4Ns to the backplane.

Cache Memory Clocks

The three Cache Memory clocks are 1) the Scratchpad clock PADS8.KO, 2) the A-Stack clock STKCLKKO, and 3) the Cache Memory clock CA.CLK00. These clocks are adjusted for the correct pulse width by overlapping the ERLYREF. and LDEDGE.. signals and by phase-shifting through two additional delay lines. These clocks are buffered by separate LD4Ns to the backplane.

Memory Base Unit Clocks

The two MBU clocks are SYS7TTL0, and SYS7TTL1. Both are TTL, both are driven by LD4Ns, and both are used with MBU 7/8. SYS7TTL1, used in the B 1985 and the B 1955, is located at #EX at the doghouse on card K. SYS7TTL0, used with the B 1905, is found on the backplane of the card K.

These MBU clocks originate from ERLYRF/.., early reference clock. A 50-nanosecond delay line provides control for the phase shift between the reference clock and the rest of the system.

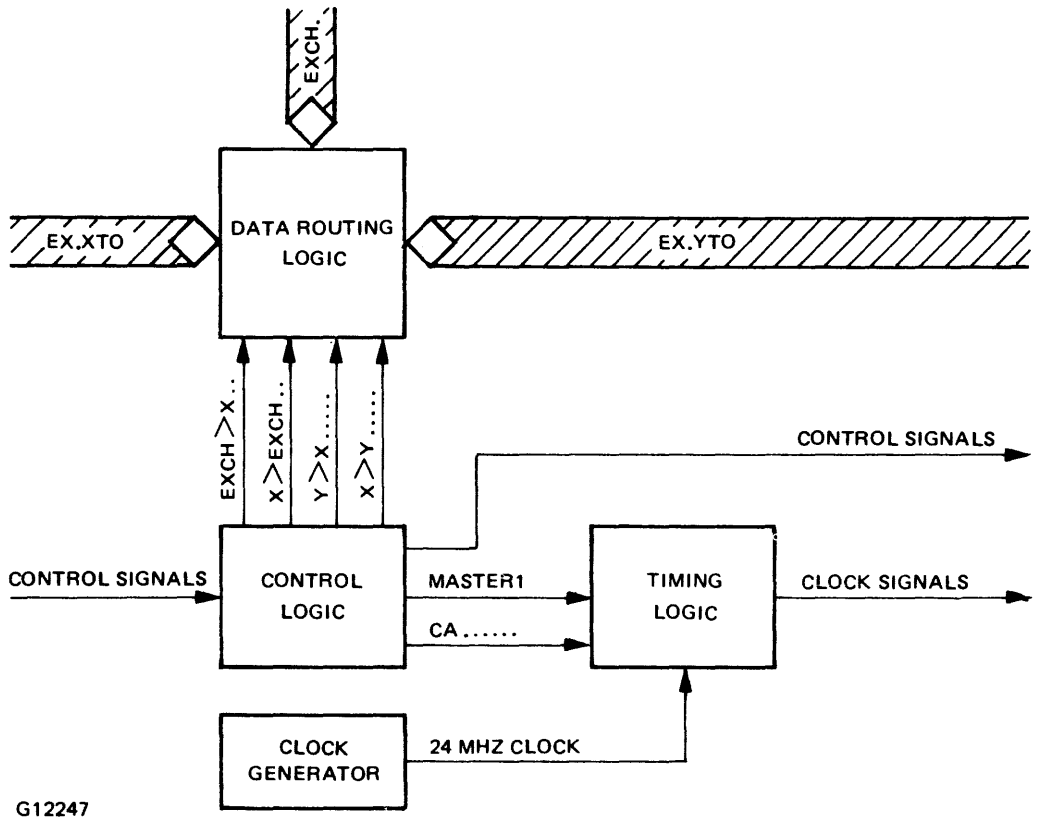


Figure 2-16. Functional Block Diagram of Clock Card (Card K)

System Clocks

Six system clocks are available for use with multiline controls and extensions. These signals, SYS1...1 through SYS6...1, are identical to the MBU clock up to the point of the 50-nanosecond delay. (See Figure 2-16.) All are of the same pulse width as the MBU clocks, and provide a fixed phase delay of about 7 nanoseconds with respect to the reference clock. clock. System clocks are routed to frontplane doghouse connectors #HX, #KX, #PX, #SX, #VX and #YX. These clocks are used with Multiline Controls and Multiline Extensions.

I/O Clocks

The I/O clocks are generated by the I/O Distribution logic. Adjustment of these clocks for correct pulse width and phase shift is similar to that of the processor clocks. The basic clock is fanned out after delay into three separate groups:

SCPMnXXX
: Seven clocks, driven by BG-Ns at the backplane and used for I/O conrols that share the same backplane with the processor.

SCPCnXXX
: Six clocks, routed to frontplane doghouse connections and used for I/O controls in separate backplanes and I/O extensions in daisy-chain configurations.

DSCPKXXX
: A signal that, through additional logic, generates the Command Active signal CA....., the Response Complete signal RC....., and the slow clocks.

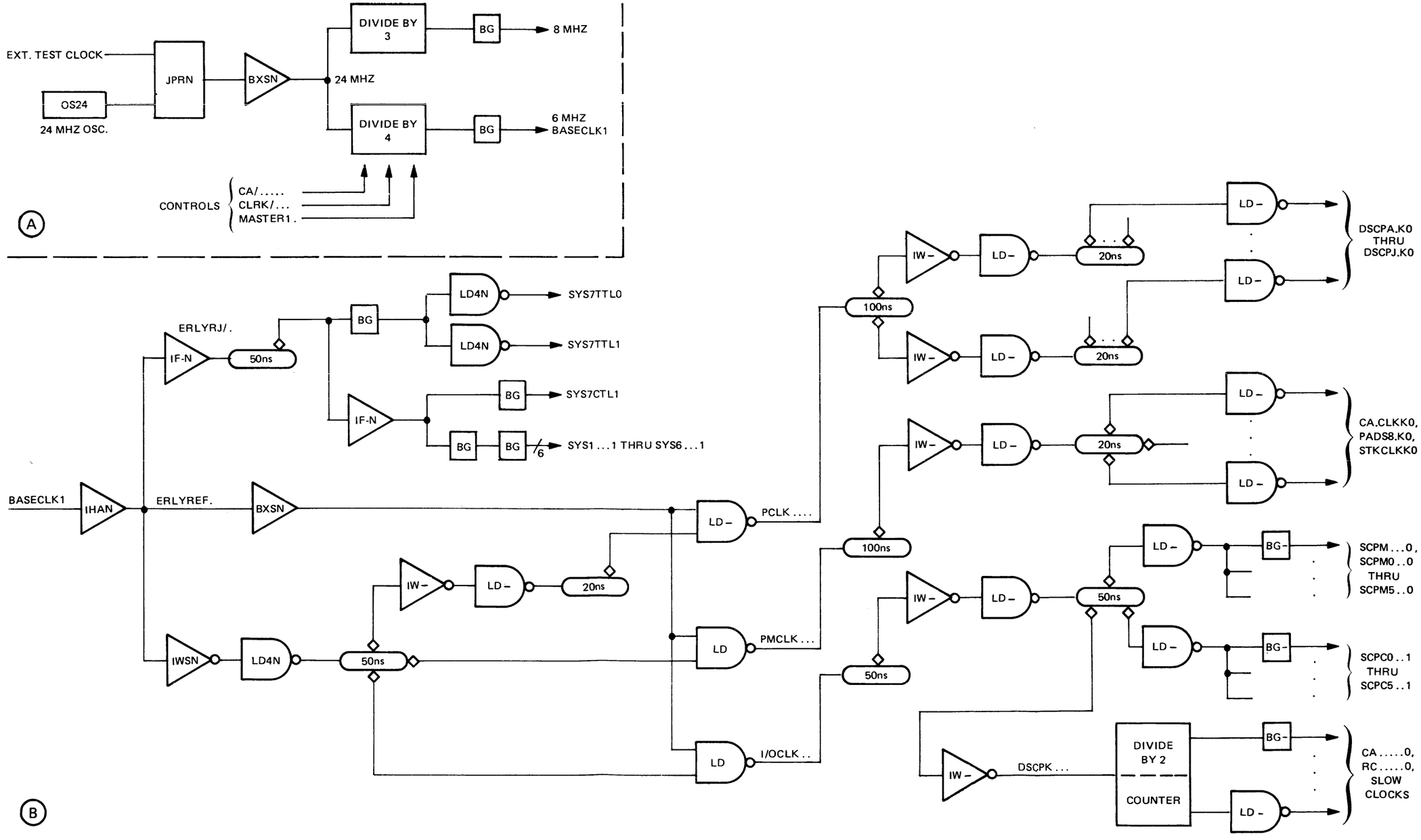


Figure 2-17. Clock Generation (A) and Calibration, Distribution (B)

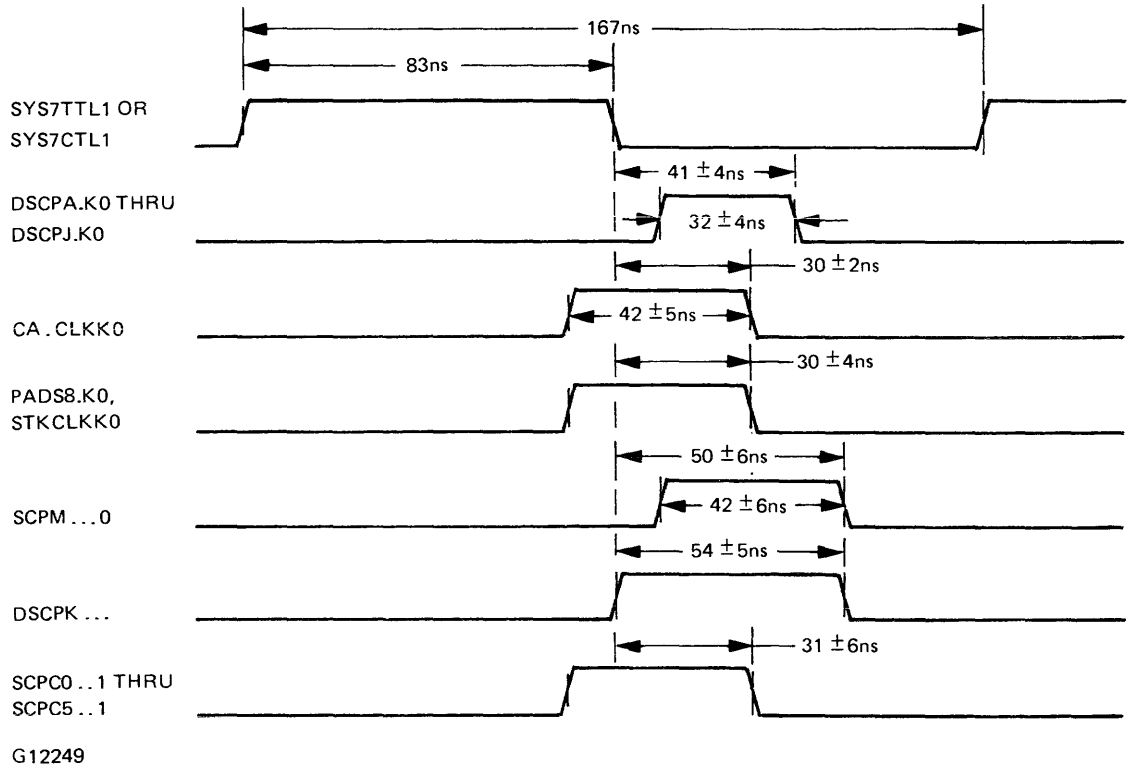


Figure 2-18. Clock Waveforms

MAIN EXCHANGE

The Main Exchange (MEX) is a 24-bit-wide bus used for transferring data from a source such as the I/O exchange, the memory interface, or a working register to a destination. Control of the MEX for data transfers is enabled by the decoding of a microinstruction. Data routing to the MEX is shown in Figure 2-19.

PROCESSOR CARD FUNCTIONS

The following subsections refer to the functions of the nine cards that contain the processing capabilities of the central system. These cards are identified by the letter designations A through J (I is omitted). Logically and in the descriptions that follow, the cards are organized into three groups: card G is the control card for cards H and J, card C is the control card for cards A and B, and card D is the control card for cards F and E.

Figure 2-20 is a block diagram of the B 1900 processor. Functional block diagrams of the cards are presented in the appropriate subsections.

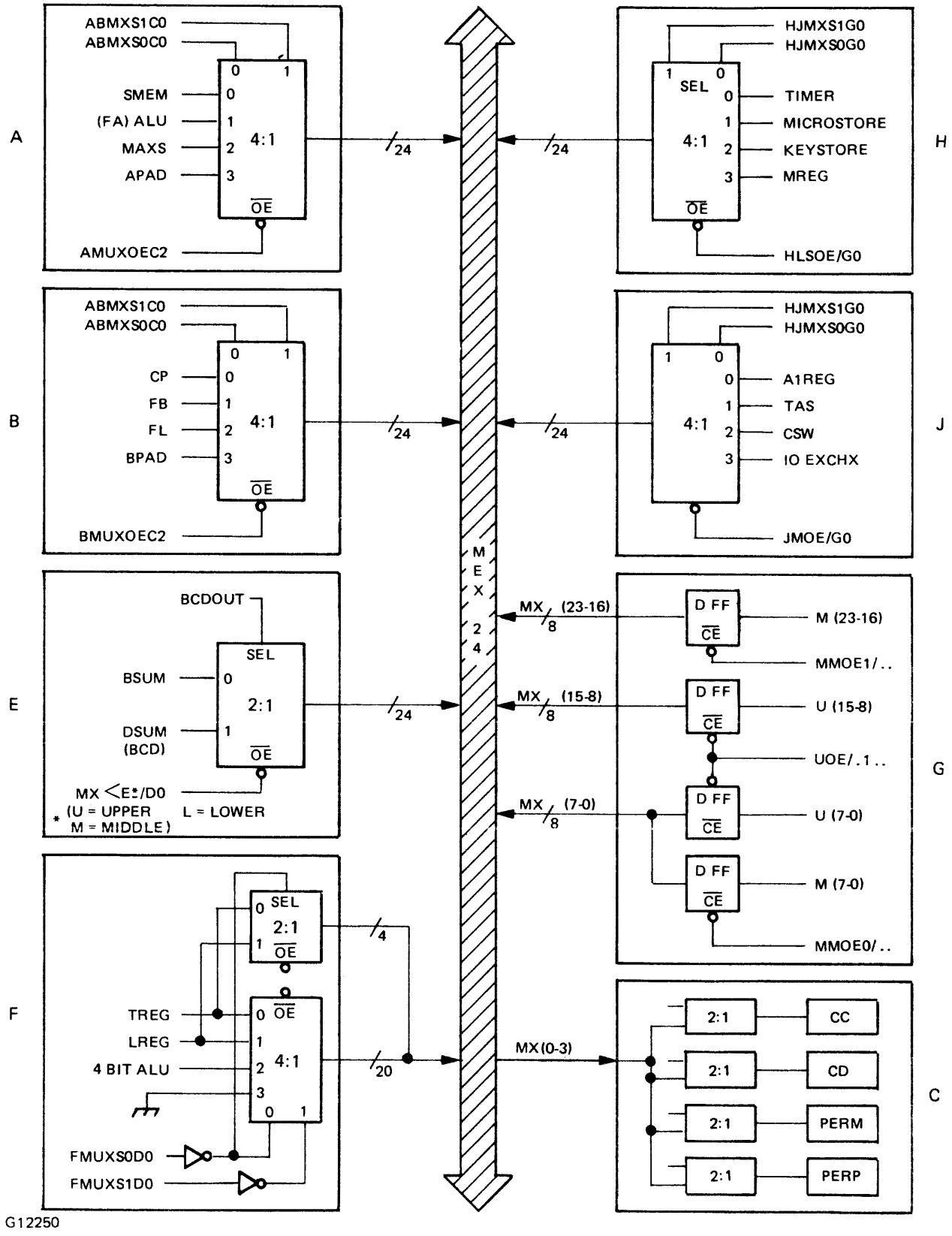
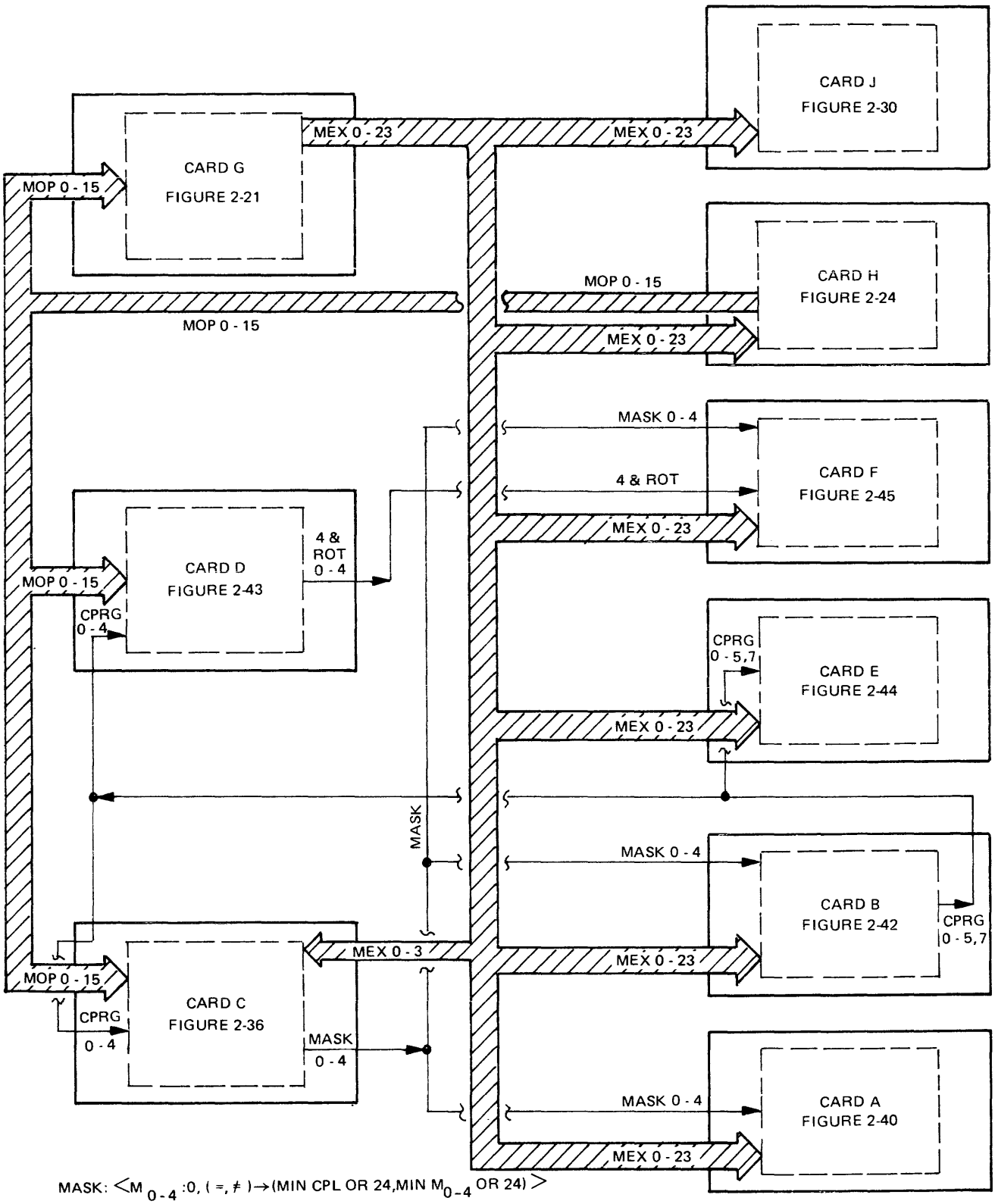


Figure 2-19. Data Routing to the Main Exchange



MASK: $<M_{0-4}:0, (=, \neq) \rightarrow (\text{MIN CPL OR 24}, \text{MIN } M_{0-4} \text{ OR 24})>$

4 & ROT: $<M_{0-4}:0, (=, \neq) \rightarrow (\text{CPL}, M_{0-4}) * 10C + M_{7-11} * 11C + \text{PROM} * 3C + \text{PROM} * 6C>$

G12251

Figure 2-20. Block Diagram, B 1900 Processor

Card Group G, H, J

Card G, the control card, includes the following logical functions:

1. Buffering of the MOP lines from cards G and H and decoding of the micros as they appear on the MOP lines.
2. Generation of control signals from PROMs for cards H and J.
3. Enabling of PROM outputs resulting from the decoding of micros to the nanoregister. The nanoregister is distributed to all three control cards and holds the control states required to fulfill the actions specified by the microinstruction.)
4. Processor RUN/HALT flip-flop and associated logic.
5. Micro sequence pipeline control logic. Fetch, decode, and execute functions are sequenced, depending on processor state and mode.
6. Micro sequence alteration control logic, to deal with micros that change the normal micro sequence.
7. MTR mode control logic. When the processor is in MTR mode, data and the ECC from the cassette is fed bit by bit to the U register. When the micro is completely assembled in U, it is loaded into the M register for decoding and execution, unless the prior micro was a 1C Move from U, which causes the contents of U to be treated as data.
8. Cache Memory fill control logic, including the FHS flip-flop. If the source for micros is NORMAL and the needed micro is missing, an automatic Cache-fill operation takes place under control of this logic.
9. Processor Initialization. GPCLR... (General Processor Clear) occurs 1) when the system is powered up, 2) when CLEAR is pressed with the processor halted, or 3) when HALT and CLEAR are simultaneously pressed with the processor running.
10. Card H and card J control logic.
11. CM and MM register control logic. CM holds the contents of the console switches; MM provides the next 16 bits of a micro if needed.
12. Console Op panel and D/M panel interface logic.
13. Cassette and U register control logic, including mechanism control, state sequencer, single-micro signal generation, and syndrome register and error correction logic.

Card H includes

1. Cache micro and key store and parity generation, the M register, and related logic.
2. Fast-branch decode and address logic.
3. Timer logic.
4. Console lamps register latches and drivers.

Card J includes

1. The A register and related logic. A, the microprogram address register, consists of two parts, A0 and A1.
2. Logic for comparing the console switches with the A1 register.
3. Fast-branch execution circuitry.
4. The A-Stack and the associated TAS (top of A-Stack) register.
5. TTL/CTL and CTL/TTL conversion for the 24-bit path from the MEX to the I/O exchange.

Figures 2-21, 2-24, and 2-30 are functional block diagrams of cards G, H, and J respectively.

Card G

The functions contained on card G (Figure 2-21) are outlined under the subsection titled Card Group G, H, J. Some detail on the micro fetch decode processes and nano generation is provided in the following subsections.

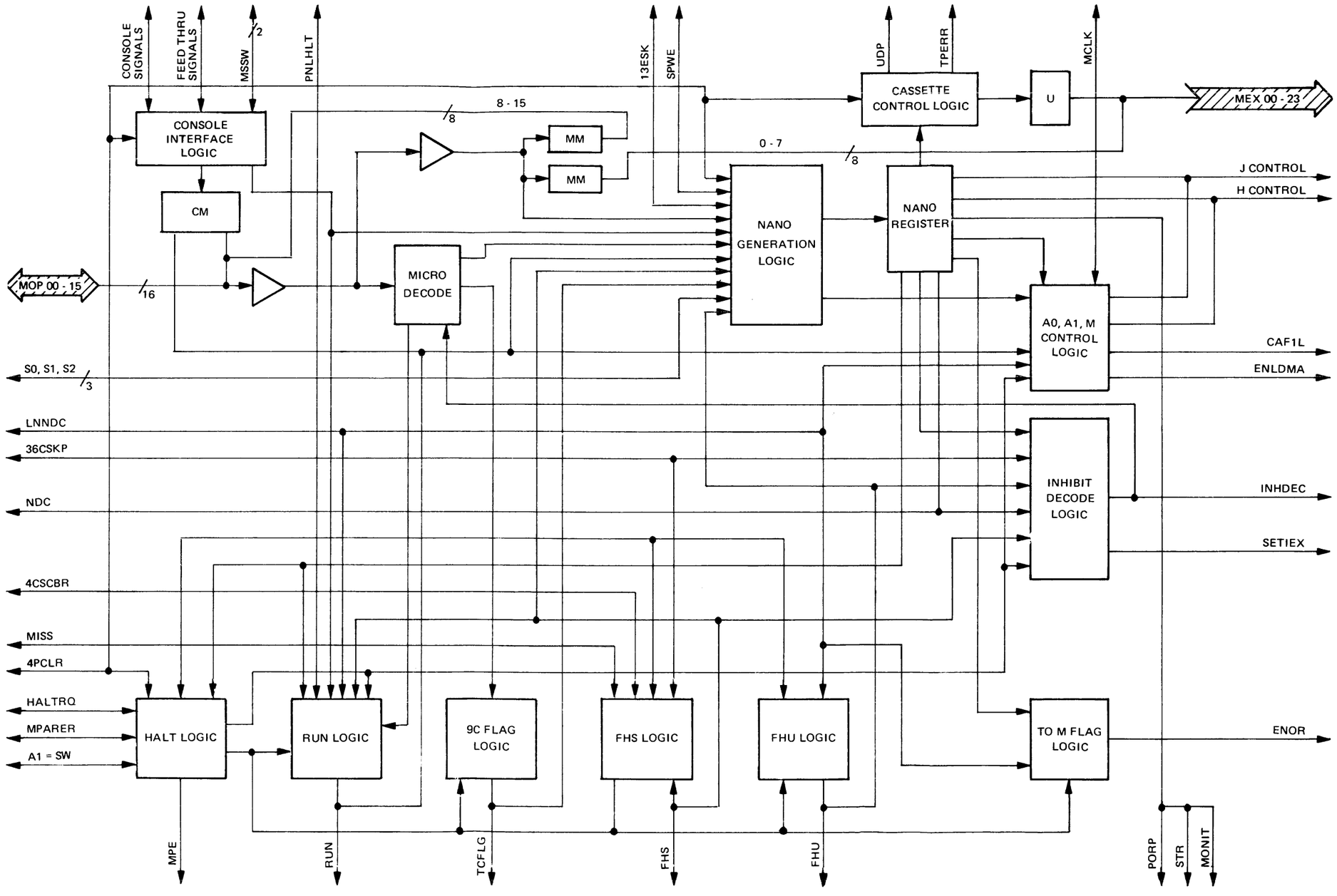


Figure 2-21. Functional Block Diagram of Card G

FETCH STRUCTURE

The fetch structure, distributed among cards G, H, and J, includes Cache Memory, the A and M registers, and the A-Stack. The A register is used to point to a micro for execution, the M register receives the micro before it is decoded, and the A-Stack is used in special cases to store the contents of the A register. Also included as part of this structure is the Top-of-A-Stack (TAS) logic and the A register manipulation logic. The basic function of the fetch structure is to load micros into the M register, from which they are put on the MOP lines and distributed to the control cards for decoding into nanoinstructions and subsequent execution.

Figure 2-22 diagrams the fetch structure.

DECODE STRUCTURE

The basic function of the micro decode logic is to manipulate the MOP lines in such a way that a micro and its variants are converted to address lines. These address lines are routed to the PROMs.

The decode structure provides the transition between the microinstruction and one or more nanoinstructions. The decode phase begins with the output of the micro from the M register, and continues until the nano is available on the output of the Micro Decode PROMs on cards G, C, and D.

The M register outputs are the MOP lines. A 16-bit bus is used to move MOP line data to the three control cards for decoding. The decoded PROM outputs are the nanos that are moved to the nanoregister for execution.

Several signals control the decoding process from the time a micro appears at the M register until the one or more resulting nanoinstructions are loaded into the nanoregister:

NDC.....	Nano Decode Complete
LN.....	Last Nano
LNNDC/..	Last Nano, Nano Decode Complete
DISM....	Disable M Register
INHDEC..	Inhibit Decode

As soon as the M register is loaded, its output is available to the decode logic. A Move 24-bit Literal (9C) micro requires a delay in decoding the next micro in M; therefore, the signal DISM prevents the literal in the M register from being sent the decode logic.

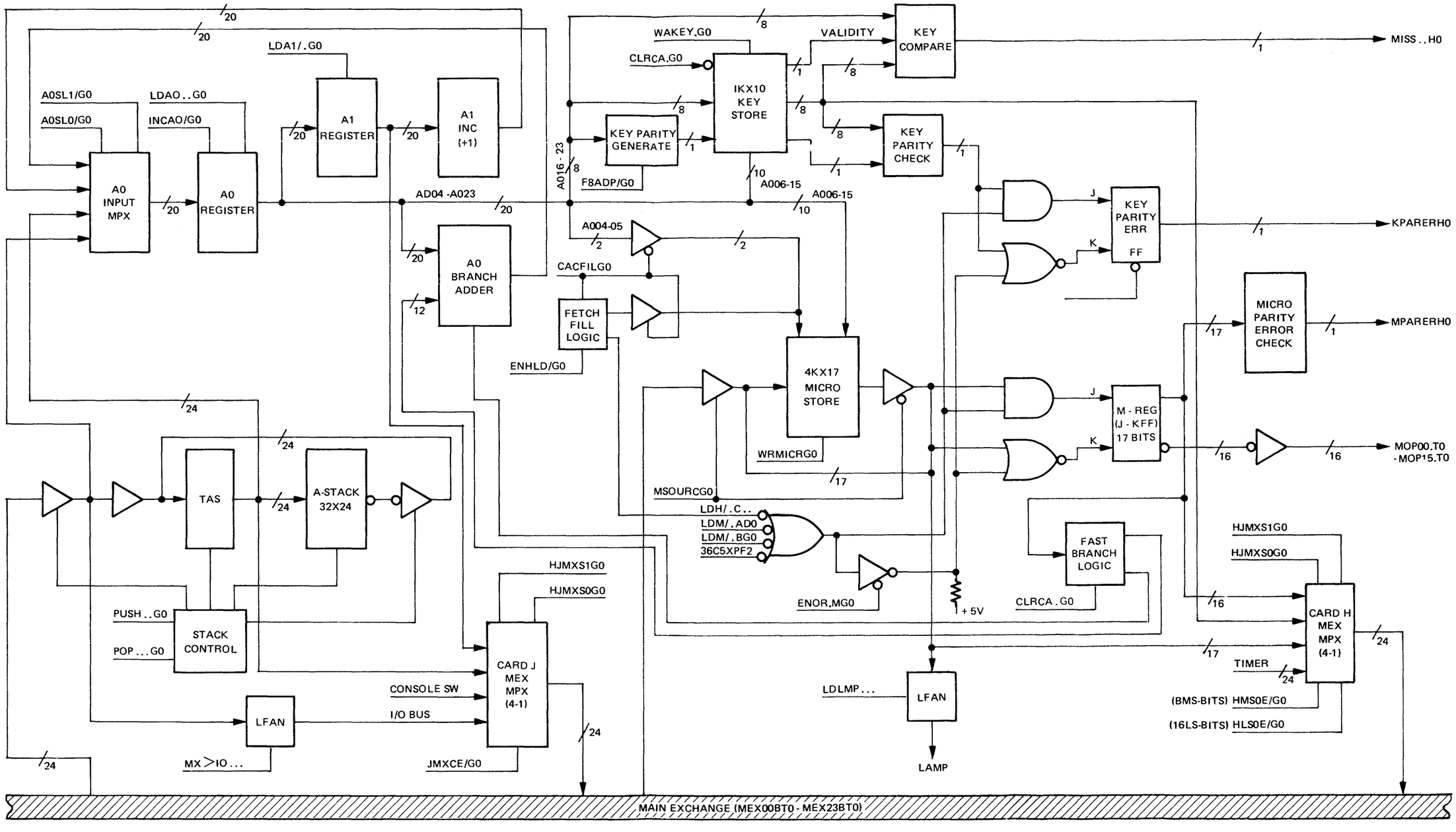
The control signal NDC, when TRUE, indicates that decoding of the present nano is complete. NDC is also used to increment the nano sequence counter.

The control signal LN is gated with NDC to indicate the completion of the micro. LN is TRUE when present sequence number PS#n and last sequence number (LS#n) are equal.

The control signal LNNDC is sent to card G to start the fetch cycle and increment the A0 register.

Buffers for the MOP lines as well as the actual micro decode logic are included on cards G, C, and D. Each of the cards contains the portion of the decode logic and nano register that relates to the functions of the related data cards. Card D contains all the nano sequencing, including the nano-complete logic.

The encode logic generates some control terms from the ANDing or ORing of key MOP lines. The MOP lines are sent to AND gates to generate terms like MD=0 (D subfield of M = 0), which are sent to the decode PROMs.



G12253

Figure 2-22. Fetch Structure

NANO GENERATION LOGIC

The logic implemented to control the generation of the bits that make up the nanoinstruction is in the form of stored logic contained within a PROM network. The binary address formed by the decoded MOP lines is applied to the input lines and a preprogrammed output is derived. The output of the PROM is directed to the nano register. The PROMs use several networks as input, including a sequence counter, the decoded MOP lines, and micro-subset lines. The nano instructions are generated on all three control cards.

NANO SEQUENCE LOGIC

Nano sequence logic controls the number of nanos required to execute a micro. For example, three nanos are required to complete a Bias (3E) micro execution. The MOP lines are sent to four PROMs which generate an LS#n equal to 2, which is one less than the number of nanos required to execute the micro. The PS#n (present is counted up each time NDC is TRUE, and LN is generated when LS#n and PS#m are equal. The sequence counter that generates PS#n is reset by LN*NDC, indicating the end of the micro.

U REGISTER

The U register is a 16-bit register used to accumulate the serial bit input from the cassette tape. The U register is addressable as a source register only.

The register comes into play when the processor is in MTR mode. Micros from the cassette tape are moved bit-serially into U. When 16 bits have been accumulated, the register's contents are automatically moved to the M register for execution.

When a micro that references U as a source is being executed, the contents of U are moved directly to the destination rather than to M. This data is not treated as a micro. If M is the destination, the data from U is bit-ORed with the current contents of M. Following execution of the exception micro, the next 16 bits from U go to M for execution, as before.

Also in MTR mode, when a micro specifying a branch type of action is executed, the contents of the A register may be affected, but the next micro to be executed is the one that moved from cassette tape to U and then to M.

In NORMAL mode, if the U register is addressed following a Cassette Control (2E) micro specifying a cassette halt, the results are undefined. Also in NORMAL mode, the U register may be sourced to load data or a program to registers or to memory.

Figure 2-23 show the logic of the U register.

Card H

Card H (Figure 2-24) includes Cache Memory, the M register, and related logic. The M register, which consists of 17 J-K flip-flops, can be loaded from the Main Exchange (MEX) or from Cache Memory. Its contents can also be ORed with upcoming data by means of control signals.

CACHE MEMORY

Cache is an 8K-byte RAM memory. The data inputs come from the MEX. The address inputs come from the A0 register on card J. Cache-Fill logic signals Cache to accept from S-Memory four words that are written starting at a word 0 location in the specified index.

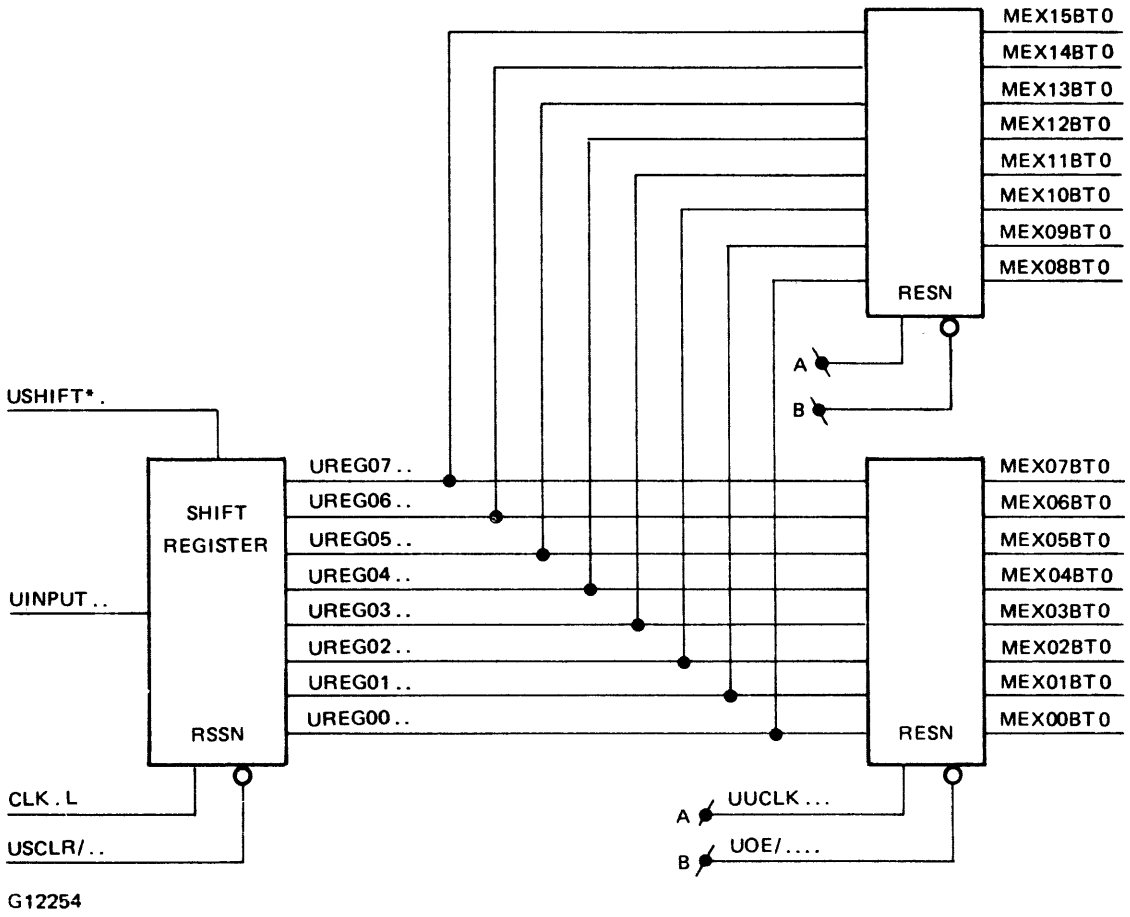


Figure 2-23. U Register Logic

Cache includes keys that are stored in 1K of RAM with address inputs KAD01/J1 through KAD15/J1 from card D. The single data input is the output of the Parity Generator logic. The Key Compare output goes HIGH or LOW depending on whether or not its two inputs match. Key Parity Check and M Parity Check logic insure that data being read has odd parity. Fast Branch Decode & Address logic uses the least significant 12 bits of the current micro in the M register as a displacement value to be added to or subtracted from the address in the A register.

Cache loads and stores micros by means of a hardware-controlled management scheme. When the MICRO SOURCE switch is at NORMAL, all micros are executed from Cache. If a desired micro is not in Cache, it is automatically obtained from S-Memory.

All micros needed for program execution are resident in S-Memory. Their relocation from S-Memory to Cache is an automatic, hardware-managed function that involves the comparison between a portion of the A register and a stored Cache Key field. The comparison indicates presence or absence of the micro. If the micro is absent, a sequence begins that results in the loading of four new micros into Cache.

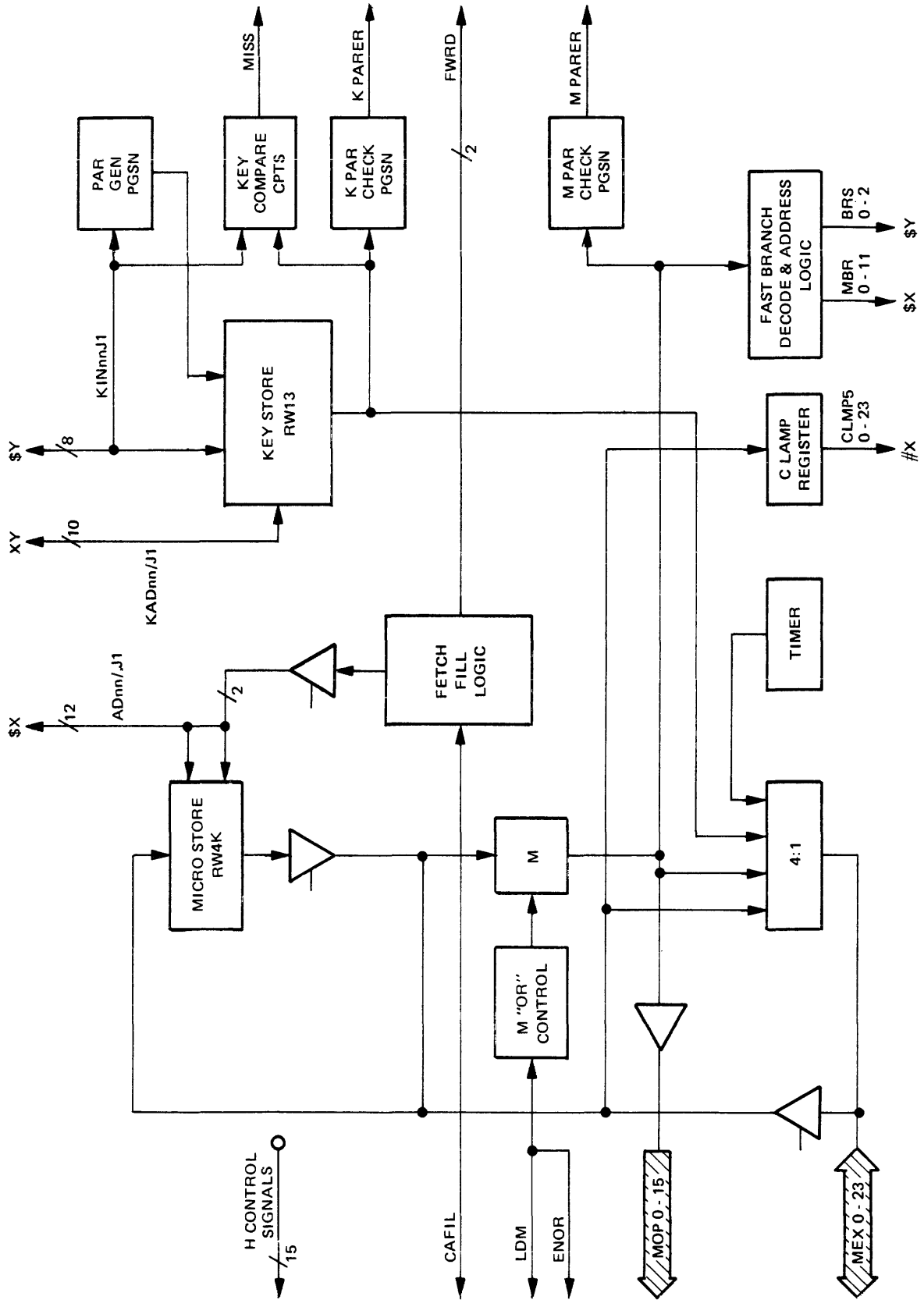


Figure 2-24. Functional Block Diagram of Card H

Figure 2-25 is a block diagram of the essential Cache logic.

Cache Memory Organization

Cache contains 1024 "classes" of four words each for a total of 4096 words (micros). Each class is addressed by an index field in the A register. Figure 2-26 shows the layout of Cache.

The 20 bits of the A register are utilized in addressing Cache as follows: the 2 least significant bits indicate the word number (0-3), the next 10 bits indicate the Index (Cache block), and the 8 most significant bits indicate the Key that is compared at the Index specified. A valid comparison between the Key field in A and the Cache Key at the specified Index signifies a "hit".

The 20 bits of the A register are also related to an S-Memory address, enabling a specific 16-bit word boundary to be found.

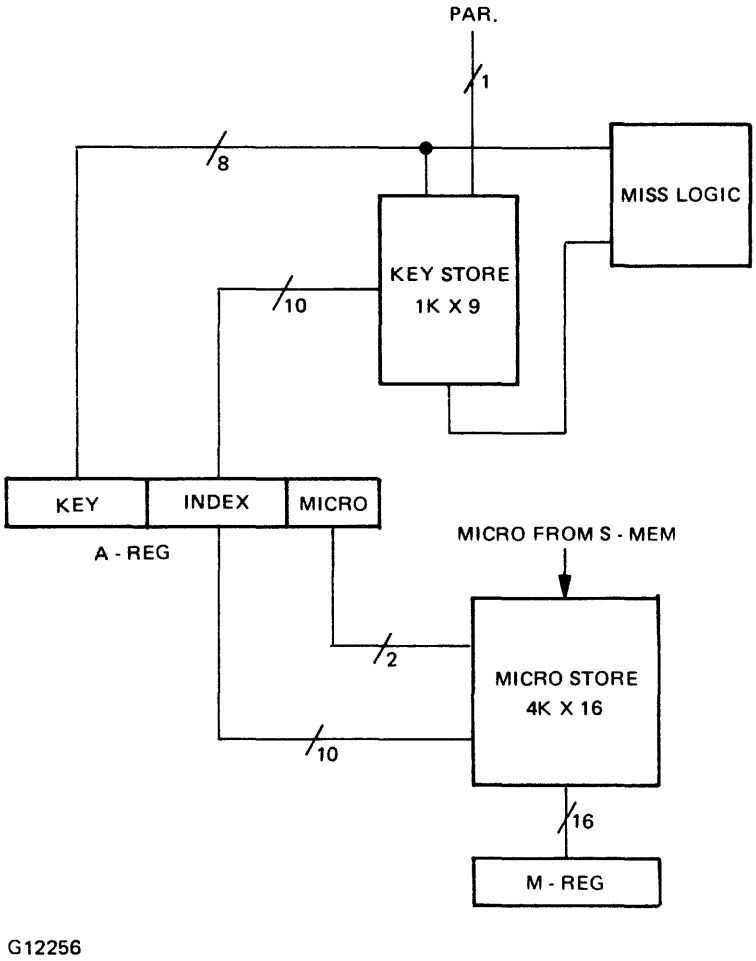
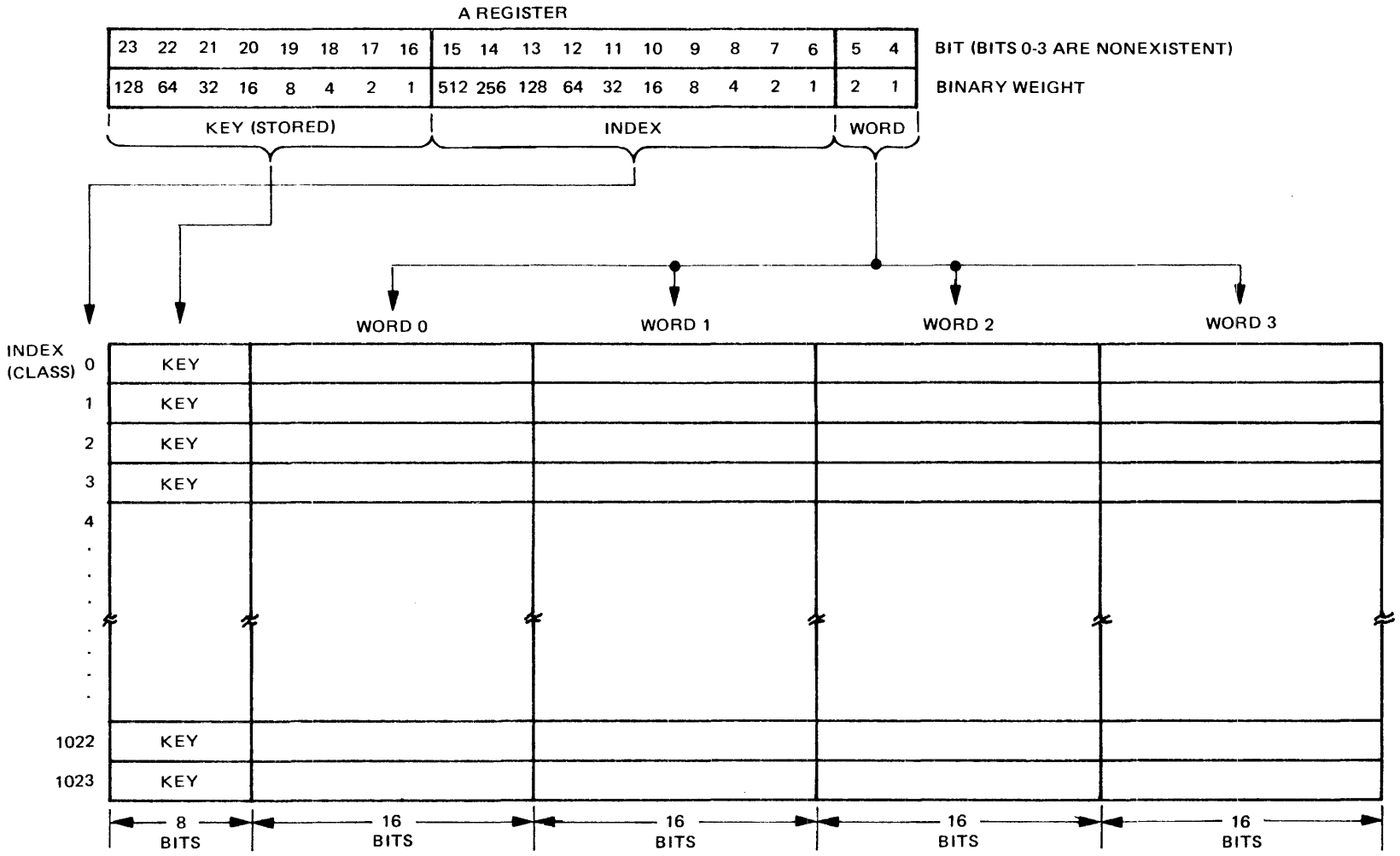


Figure 2-25. Cache Memory Organization



NOTE:

KEY STORAGE IS ACTUALLY 10 BITS IN WIDTH: 8 BITS ADDRESS, 1 VALIDITY BIT AND 1 PARITY BIT. LIKEWISE, MICRO STORAGE IS ACTUALLY 17 BITS WIDE: 16 DATA AND 1 PARITY.

G12218

Figure 2-26. Cache Memory Layout

Implementation

Cache Memory is used in association with the A register, the fields of which contain addressing information for fetching micros from Cache.

NOTE

Although A is a 20-bit register, it is referenced in the discussion that follows as a 24-bit register wherein MSB is bit 23, LSB is bit 4, and bits 3, 2, 1 and 0 are nonexistent.

The Index at which the micro to be accessed is stored is specified by bits 15-6 of the A register. The specific micro to be accessed from the block of four is specified by bits 5 and 4. Bits 23-16 form the Key. One 8-bit Key and a parity bit is associated with every block of four micros.

When Cache is loaded with four micros from S-Memory, bits 23-16 of A are written into the Key storage area of Cache. Conversely, when a micro is needed from Cache, the eight bits of the Key are compared associatively with bits 23-16 of A0. If the Key matches, a hit has occurred, and the micro is fetched to the M register. If there is no match, a miss has occurred and four sequential micros along with the Key portion of the A register are fetched to Cache and inserted at the correct Index.

Figure 2-27 shows the Cache Key compare logic.

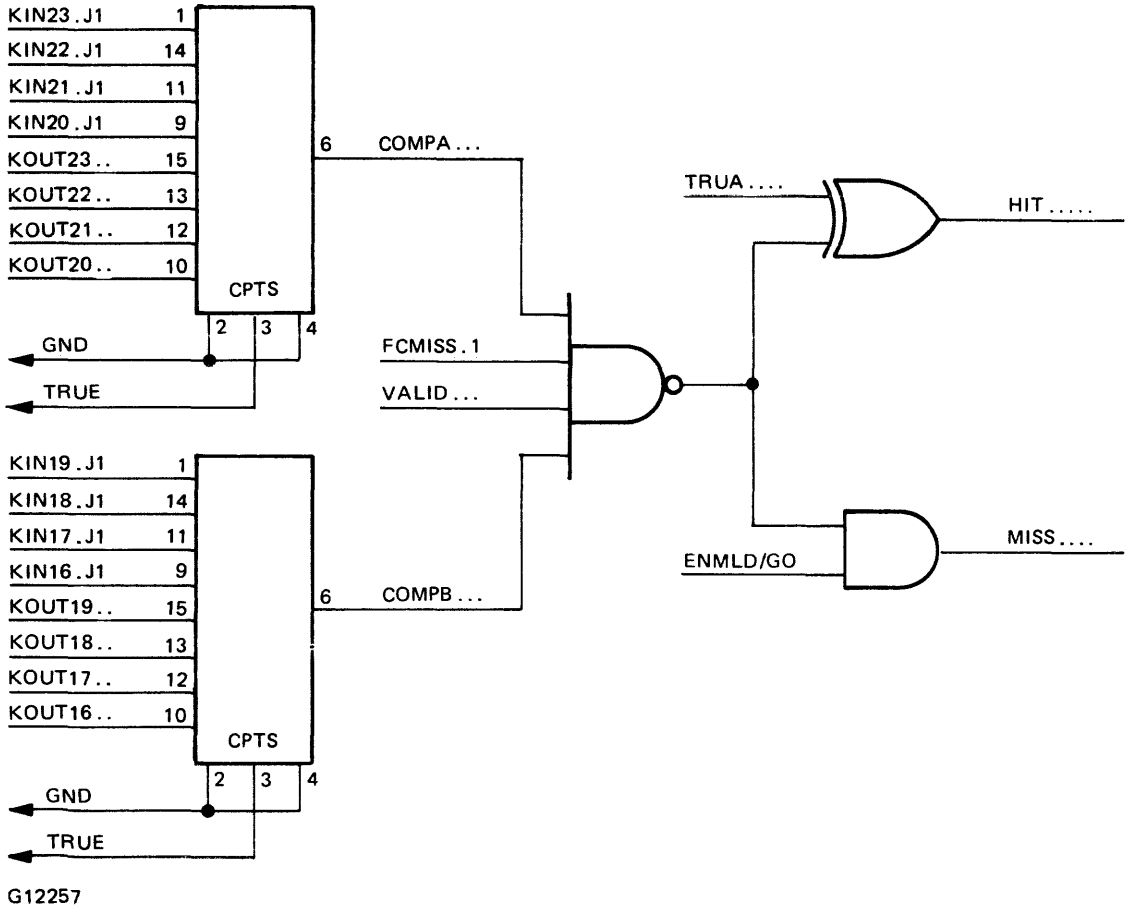


Figure 2-27. Cache Key Compare Logic

Validity

A validity bit is provided with each class. (See Figure 2-26.) If set (1), the class contains valid micros; if reset (0), the block of micros is not valid.

Clear Cache (micro 5F) resets all validity bits. Fetching and loading micros from S-Memory as well as writing micros into Cache from the console sets the validity bit for that block.

Parity

Cache Key storage includes a parity check, which tests for odd parity of the eight Key bits plus the validity bit. In the event of a Key parity error, PERP bit 02 is set, the ERROR lamp goes on, and the processor halts.

Each Cache word also includes a parity bit for a total of 17 bits. Parity is odd; single-bit errors are detected. This logic includes 17 RW4K chips: 16 for data and one for parity.

Cache Key Readout

Figure 2-28 shows the console display that results when the Console Read Cache Key variant of the 7E micro is executed.

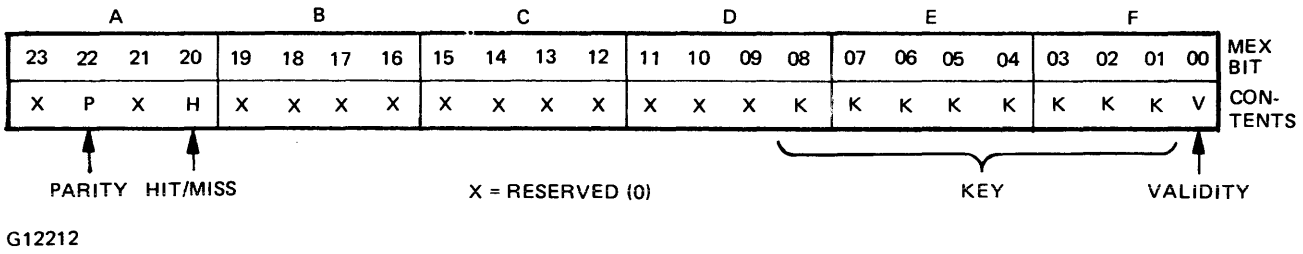


Figure 2-28. Console Lamp Display of Cache Key

M REGISTER

The M register is a 17-bit register of 16 data bits and 1 parity bit used to hold a micro in preparation for decoding by the processor. The contents of this register are decoded to enable the different control signals that result in the operations specified by the micro. Figure 2-29 shows the logic.

The M register is addressable as a source or a sink. When used as a sink, the source data is bit-ORed with the micro just fetched into M.

Bit 16 of M is used for parity checking on each fetch into M from Cache or S-Memory, but parity check is disabled for cassette and console loads into M and for micros that specify moves into M.

The microinstruction output from Cache is applied to the D-set inputs of the M register flip flops, and is loaded when Nano bits ENJ.X... and ENK.X... come TRUE. On a move to M, the bit-OR function is applied and Nano bit ENJ.X... comes TRUE and ENK.X... is FALSE.

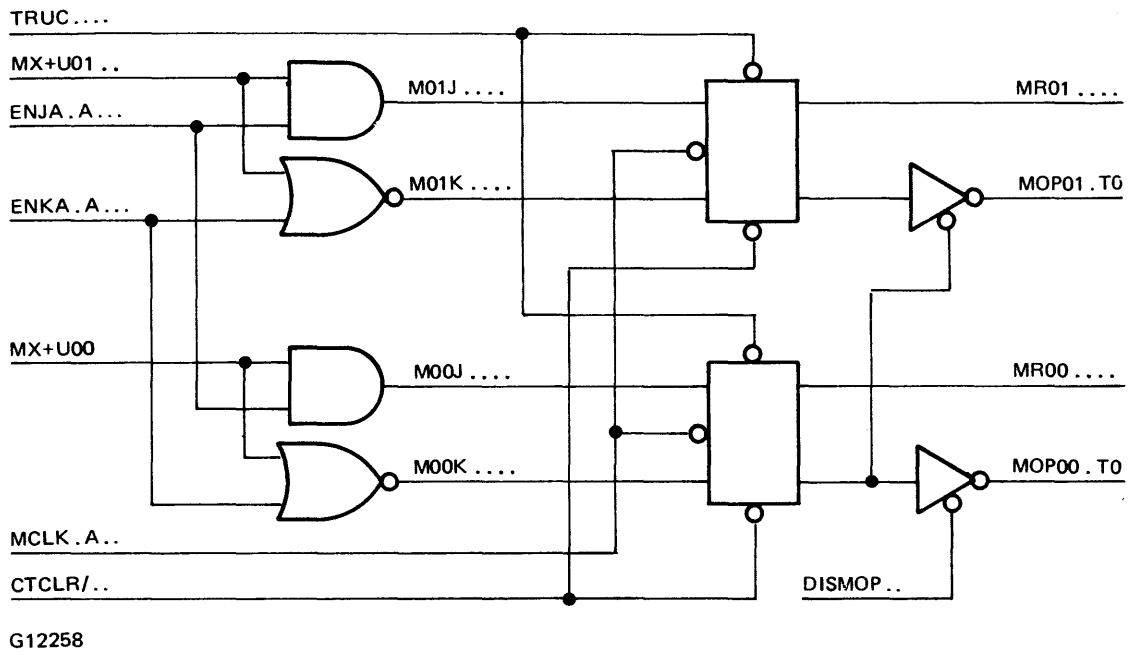


Figure 2-29. M Register Logic

TIME

Time is a 24-bit register that continuously counts every 3 system clocks. It counts up and can wrap around. It is a source-only register and attempting to move data into it causes the register to reset to zero. Time can be accessed by the Register Move (1C) micro and can be reset by three micros: Move 8-bit Literal, Move 24-bit Literal, and Shift/Rotate T, micros 8C, 9C, and 10C, respectively. No other micros affect the Time register.

Card J

Card J primarily consists of the A register, the A-Stack, and related logic. (See Figure 2-30.)

A REGISTER

The A register is actually two registers, A0 and A1, each 20 bits long.

A0, the main receiver of address information for Cache Memory, A0 can be incremented by one word (16 bits). Also, the value of a 4-bit or 12-bit field can be added to or subtracted from A0. The register can receive information from the MEX, the TAS, and the Fast Branch Adder.

The A1 register receives the output of A0. Whatever value resides in A0 is mirrored in A1 at the next system clock; thus, A1 can serve as a holding register for A0 so that Cache address information can be held for one extra clock. The output of the A1 register drives the MEX.

Generally, A0 is used to address Cache Memory in order to fetch micros, and A1 is used to hold this address for the decode phase of that micro. This arrangement assures that the A register receives new Cache address information on every clock, while prior address information remains available to the M register.

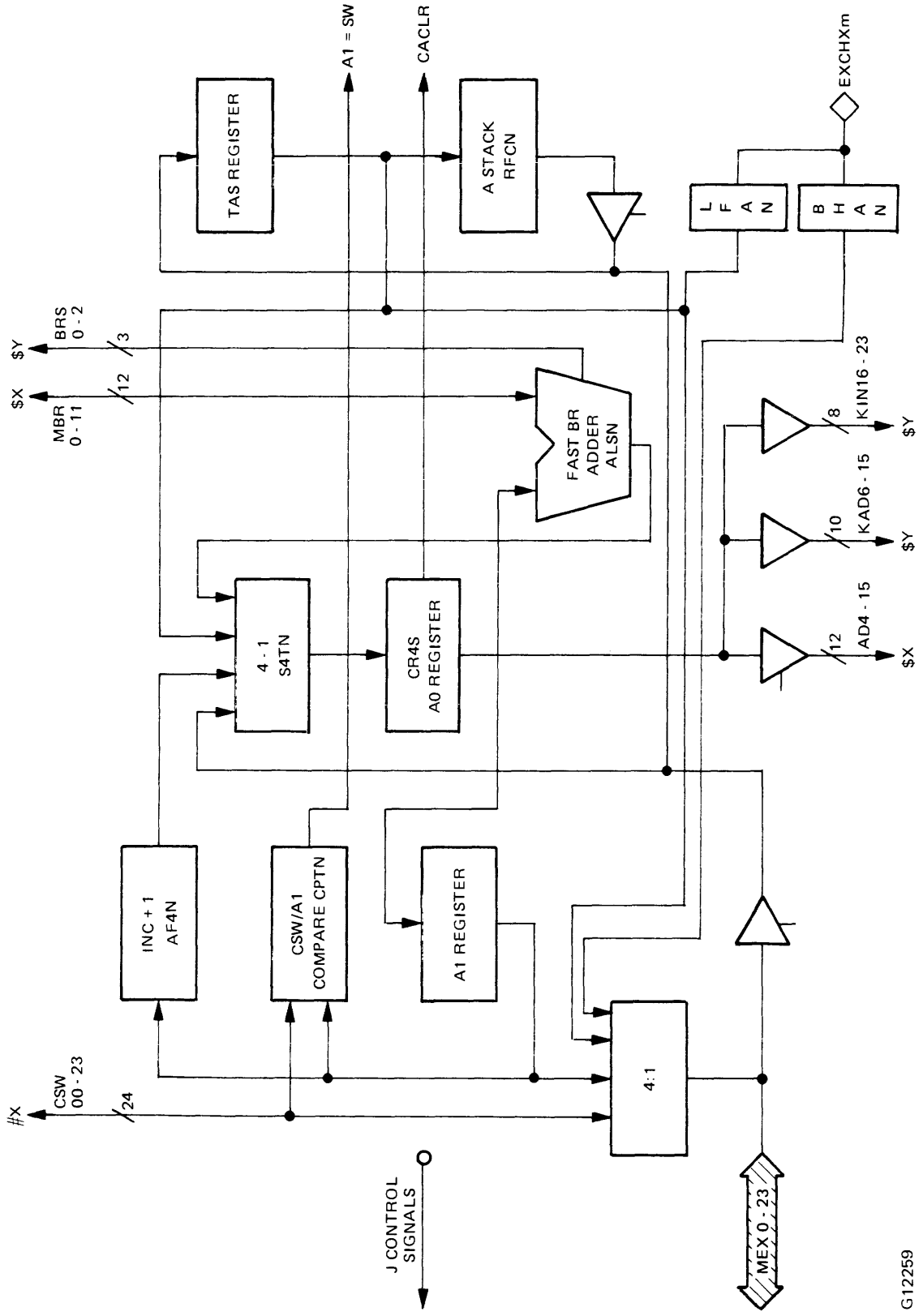


Figure 2-30. Functional Block Diagram of Card J

Figure 2-31 is a block diagram of the A register. Nano bits AOSLO/GO and AOSL1/GO are used as binary encoded inputs to select one of four inputs to A. This is accomplished by applying the four sources as inputs to a 4-to-1 multiplexer and using the A0 select lines to select one source. The A0 register is made up of CR4S chips. When the mode inputs are set for a load, the information from the output of the multiplexer is loaded to the A0 register. The A0+1 increment operation is another function of the CR4S chip in accordance with the input mode lines, control signals that are outputs from the nanoregister.

A-STACK AND TAS REGISTER

The A-Stack is a specialized memory unit that utilizes a last-in, first-out addressing scheme. The stack contains 32 locations, each 24 bits wide, that are addressed by six address lines (STKADn..). The lines are generated by a counter, called the stack pointer, that is automatically incremented by data moves into the stack and decremented by moves out of the stack. The pointer can wrap around in either direction.

Top-of-A-Stack (TAS) is a special location, not part of the body of the stack. TAS is the only portion of the stack accessible as a source or sink by processor hardware. Stack pointer values are all relative to TAS.

RFCN memory chips comprise the body of the A-Stack; three RESN chips comprise TAS.

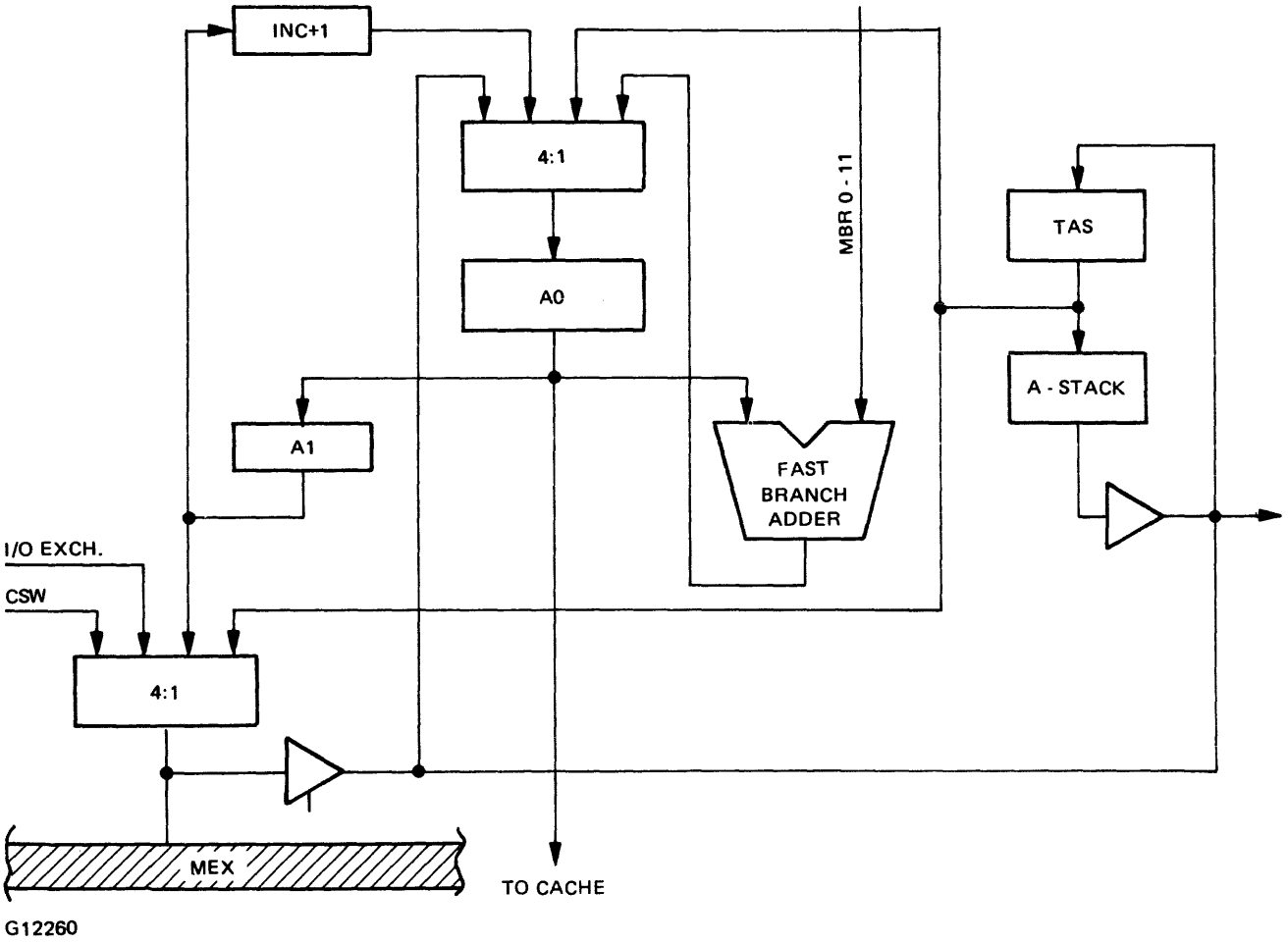


Figure 2-31. A Register Block Diagram

A push is a write into the A-Stack. It is accomplished by simultaneously moving the current contents of TAS to the stack address referenced by the stack pointer plus 1 and moving the contents of the source (the new data) to TAS. The stack pointer is incremented by 1.

A pop is a non-destructive Read of the A-Stack. First, data is read from TAS to the destination register, then the A-Stack location is read to TAS. The stack pointer is then decremented by 1.

Although direct external control of the A-Stack address is not provided, any desired location may be accessed by repeated pops with a Null register as the destination. Pops to Null decrement the pointer without destroying the data stored in the A-Stack.

The following paragraphs step the reader through A-Stack operation. Refer to Figure 2-32.

Pushing

In an initially cleared condition, as shown in Figure 2-32, column 1, TAS contains zeros and the stack pointer is at stack address 00. On the first push, the stack address (SADRn.CO) becomes stack pointer plus 1 (00+1=01). The contents of TAS (zeros) are moved to the stack address, data (A) from the source is moved to TAS, and the pointer is incremented to 01. (See column 2.)

On the next push, the stack address becomes 02 (stack pointer plus one: 01+1=02). Data (A), now in TAS, is moved to this stack address, data (B) from the source is moved to TAS, and the pointer is incremented to 02. (See column 3.)

Subsequent pushes follow the same pattern. On the 32nd push, with the pointer at 31, the stack address is again 00 (31+1) and AE is stored at this address. AF, last source data in the push sequence, is stored in TAS. (See column 5.) On this operation, when the pointer is incremented, it wraps around to 00 as shown in column 6.

Popping

The next operation in the example is a pop. TAS contains data AF, which is moved to the destination. The stack pointer is at 00. On a pop, the stack address and the current pointer are equal; therefore, data at this address is moved to TAS. (See columns 6 and 7.) Also on a pop, the stack pointer is decremented and returns to 31 as shown in column 7.

The final operation in the sequence is another pop. Again, data in TAS (AE) is moved to the destination, data at the address specified by the pointer is moved to TAS, and the pointer is decremented to 30, as shown in column 8.

Logic and Signal Flows

Figure 2-33 shows the A-stack, TAS, and the stack address components. One of the two nano bits (POP..... or POP/....) is applied to the input gates allowing either the stack contents STKnn/.. or the MEX to propagate through. Figures 2-34 and 2-35 illustrate the data paths for a push and a pop operation and the components which each operation affects.

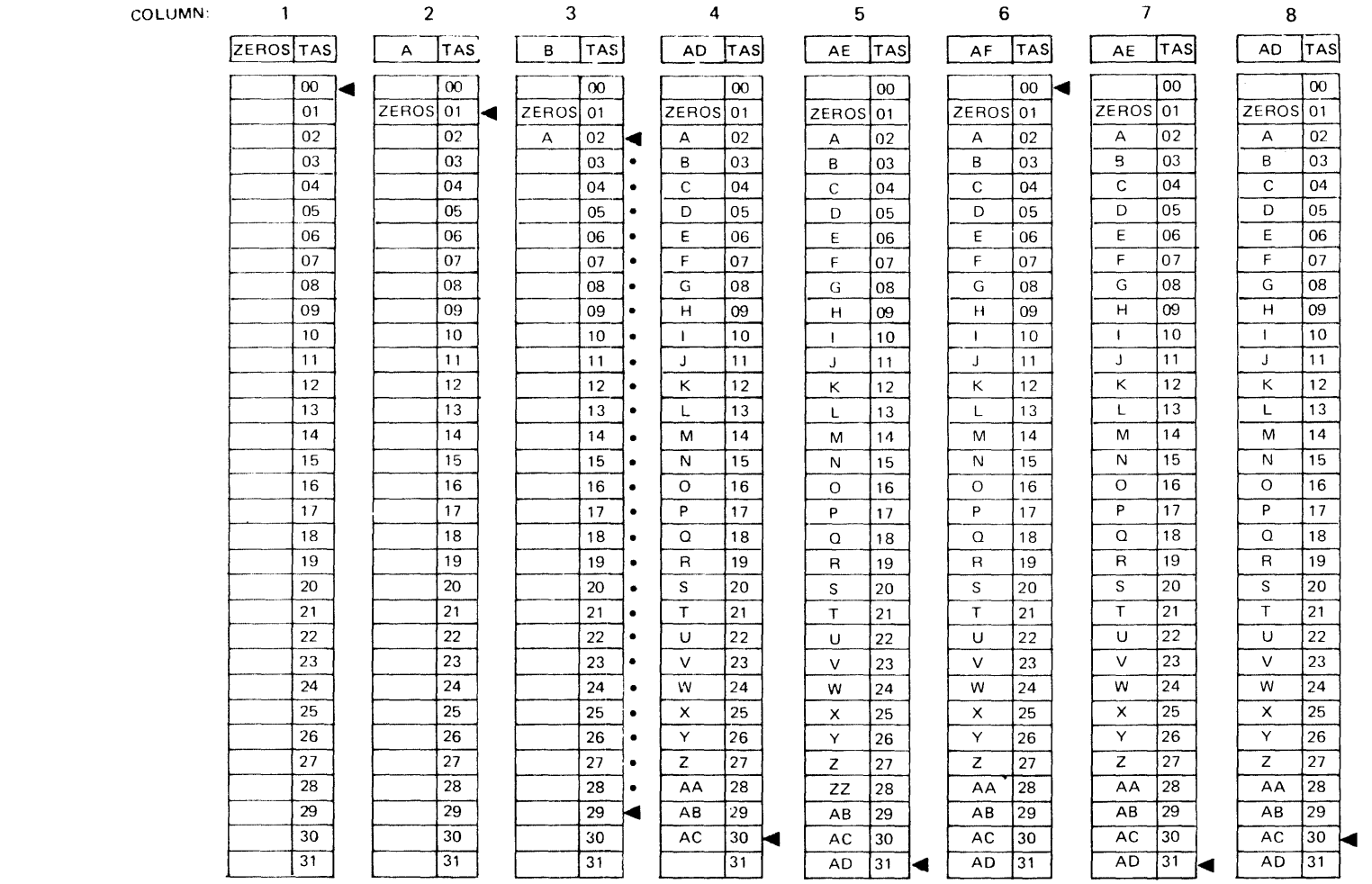


Figure 2-32. Examples of Pushing Into and Popping from A-Stack

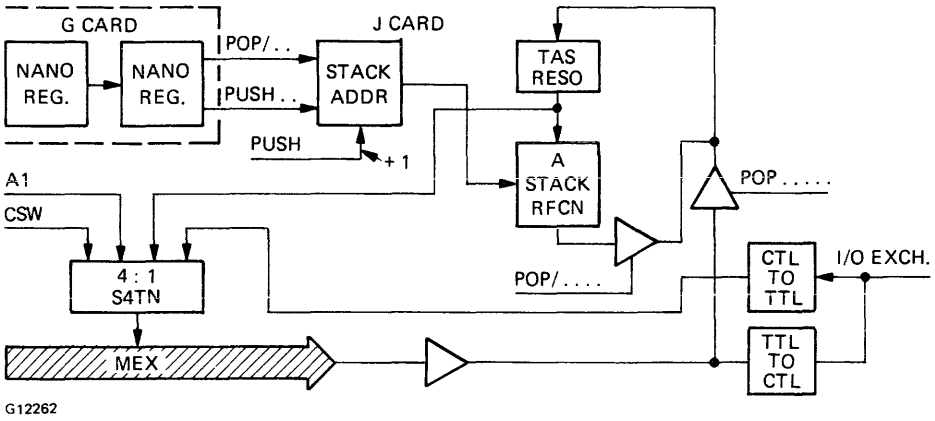


Figure 2-33. A-Stack Block Diagram

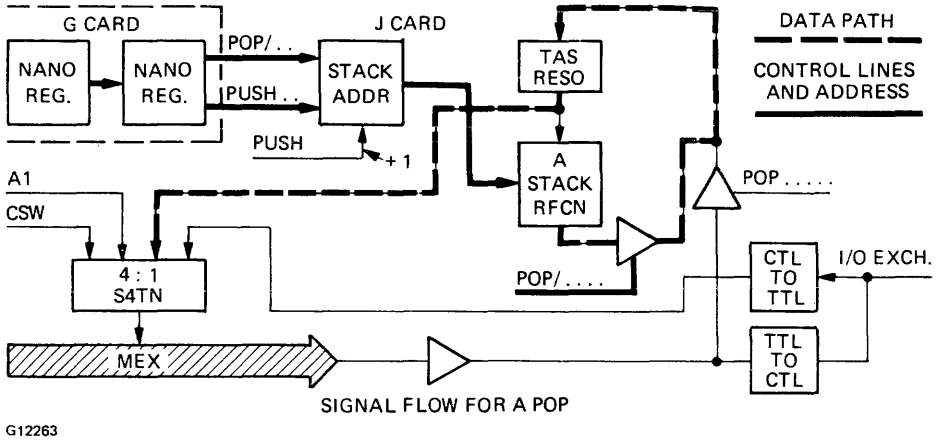


Figure 2-34. Signal Flow for a Pop

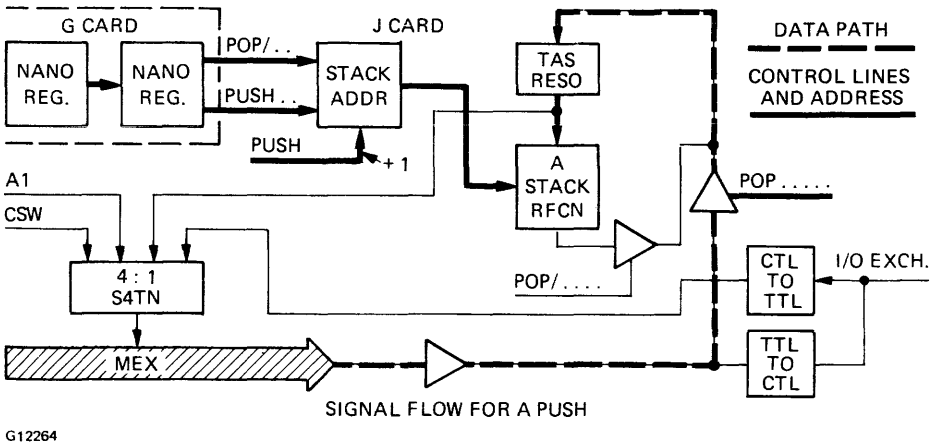


Figure 2-35. Signal Flow for a Push

I/O INTERFACE

Card J contains the CTL/TTL interface for the I/O controls. The Data and Command (CMND) registers are involved in this interface.

Data Register

Data is a 24-bit pseudoregister that can act as a source or as a destination. It is used to transfer data to and from the I/O bus. When it is used as a source, the processor generates the Response Complete (RC) signal to the interface and accepts the 24-bits of data from the bus. When used as a destination, the processor generates the RC signal to the interface and puts the data from the designated source on the bus.

The Register Move (1C) and Scratchpad Move (2C) can use the Data register as source and sink. Shift/Rotate T (10C) can use Data as a sink only. An "I/O spreader" prevents multiple RCs from occurring less than 8 clocks apart.

The Data register, when accessed from the D/M panel, can be used to display selected I/O data in the console lamps. Pressing LOAD with Data selected on the REGISTER GROUP switch generates the required RC signal.

Command Register

Command (CMND) is a 24-bit pseudoregister that can act as a destination only. It is used to transfer commands to devices on the I/O bus. The processor generates the Command Active (CA) signal to the interface and moves the command from the designated source to the bus.

Micros 1C, 2C, and 10C can sink the CMND register. The I/O spreader prevents a CA from following an RC any sooner than 4 clocks.

CMND can also be accessed from the console. Since CMND is a sink only, sourcing CMND will result in a no-operation and, when LOAD is pressed, a CA is transmitted along with the transfer of the console switch contents to the I/O device.

CONSOLE SWITCH REGISTER

The Console Switch (CSW) register, 24-bits, is a source only. When addressed as a source, the contents of the console switches is supplied to the destination.

Card Group C, A, B

This group deals primarily with S-Memory data control, but includes logic and data paths that can be used for general-purpose computing.

Card C (the control card) includes the following functions:

1. Buffering of the MOP lines from cards G and H, and decoding of the micros as they appear on the MOP lines.
2. Generation from PROMs of the control signals for card A and card B.
3. Decoding of Micros to the nanoregister, enabling PROM outputs.
4. Scratchpad address generation and Write control.
5. System status and error information, using the following 4-bit registers: PERM, PERP, CC, CD, and INCN.

6. Micro source information and the MSSW register.
7. 4-bit register source select logic. Any of seven 4-bit registers located on card B may be selected for input to the 4-bit function box located on card F.
8. Error halt detection logic.
9. Mask amount generation logic.
10. Memory Write protection logic when a memory address is outside the limits of memory.

Card A provides a 24-bit bidirectional path for memory data as well as the MBU address. Card A includes the following functions:

1. The FA register for memory addressing and the BR and LR registers for memory address limit checking. All three of these registers also have general-purpose uses.
2. The MAXS register, a jumper chip used to specify the physical size of S-Memory.
3. Scratchpad A, the left 24 bits of Scratchpad memory.
4. The ALU, a 24-bit arithmetic/logic unit with direct inputs from FA, LR, BR, TEMP, and Scratchpad A.
5. A temporary register used by the Bind (4F) and Load Lamps (7F) micros.

Card B includes

1. The CP register.
2. The FB register.
3. Bias and FLCN logic.
4. 4-bit multiplexing logic with inputs from the six 4-bit subregisters of FB and the 4-bit FLCN register.
5. Scratchpad B, the right 24 bits of Scratchpad memory.
6. A 24-bit ALU with inputs from FB and Scratchpad B.

Figures 2-36, 2-39, and 2-41 are functional block diagrams of cards C, A, and B, respectively.

Card C

Card C (Figure 2-36) is the control card of the C, A, B group and includes the functions listed above.

REGISTER SOURCE/SINK CONTROL

The processor's execute structure uses a 24-bit exchange in order to move data to or from the working portion of the processor. The nanoregister generates register source and sink control signals for the manipulation of data to or from a working register.

When it is necessary to source a register, the nanoregister generates all the control terms. On cards A and B, ABMXS0C0 and ABMXS1C0 are sent to the output multiplexors for selecting the register to be output from the card. Then, the output of one card only is enabled to the MEX by the terms AMUX0EC2 for card A or BMUX0EC2 for card B. This process also applies to cards H and J.

Once the data is on the MEX, it is available to all the data cards. When a register is the sink, the nanoregister generates the control signal to enable the data to the register. The data on the MEX is sent through a buffer, then to the multiplexors. The multiplexors are then controlled by the output from the nanoregister. On the next clock, the data is written into the register.

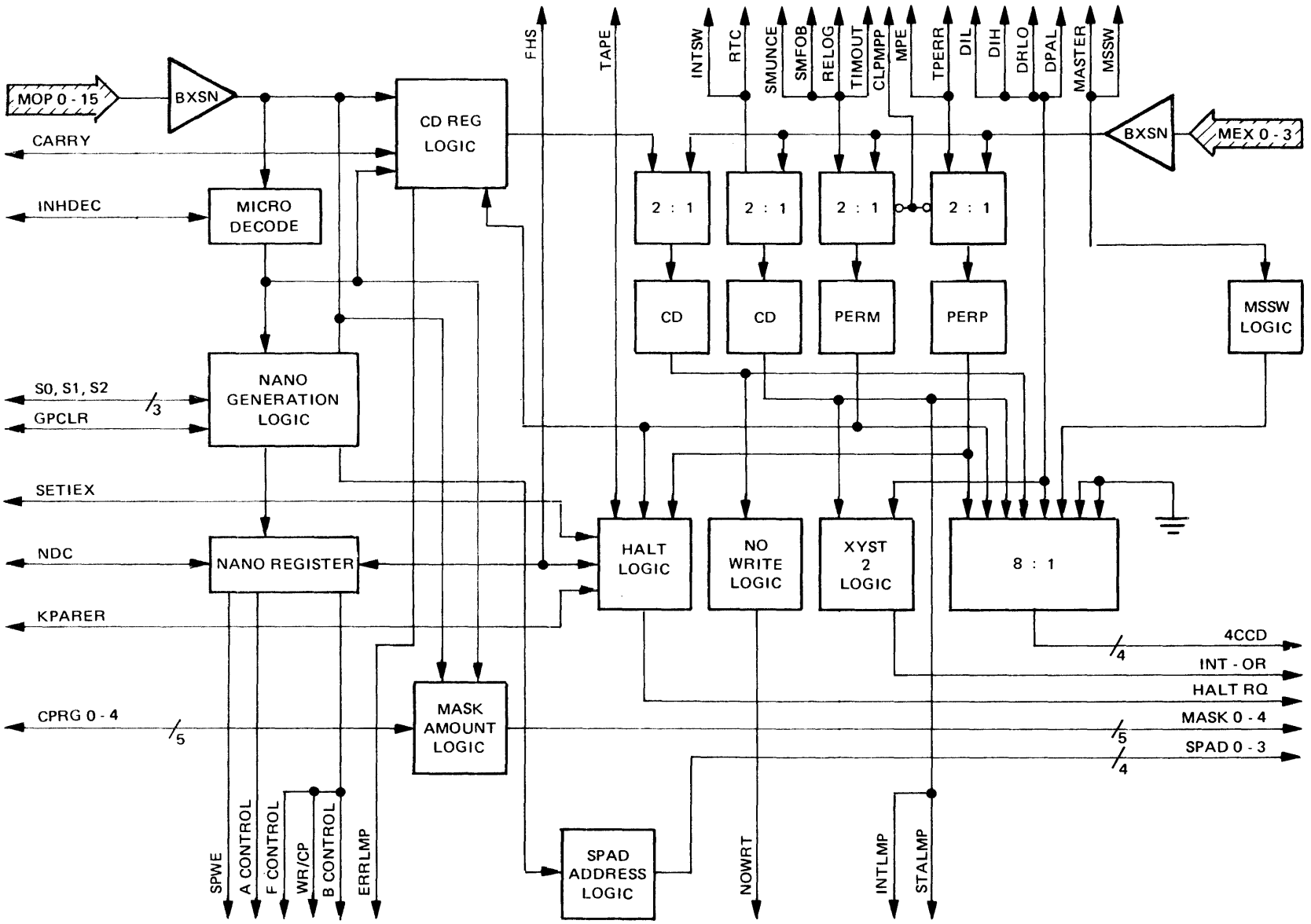


Figure 2-36. Functional Block Diagram of Card C

SCRATCHPAD

Scratchpad, a memory consisting of 16 pairs of 24-bit words, serves to hold field descriptors during the iteration of operands. Also, some of its memory cells may be used to hold S-language stack pointers and other processor registers that are under manipulation. Scratchpad layout is shown in Figure 2-37.

The FU and FL portion of the FB register and corresponding portions (SFU and SFL) of the first cell of Scratchpad are used to set the various conditions of FLCN and the various conditions of the CP register. SFU refers to the four most significant bits of the first cell in Scratchpad B, and SFL refers the 16 least significant bits of the same cell.

Each pair of words is considered as a left (A) word and right (B) word. Together, a pair of words may be called a doublepad word. The left and right halves of Scratchpad are also referred to as APAD and BPAD respectively; they are located on cards A and B.

A Write into Scratchpad is a two-step operation. Each step requires one clock to execute: 1) at the first clock, data is trapped into a latch. 2) During the second clock, data is written into Scratchpad. Another micro can be executed during the second clock if it is not a Read from Scratchpad. Any immediate Read from Scratchpad is delayed by hardware until the preceding Write operation is completed.

Data on the MEX is sent through a 2:1 multiplexor and output as TOPSnnA. The data is then clocked by FRCLKnA. into a holding register. Addressing is from nanoregister control lines called SPADDnCO. These are then decoded to APADDnA. or BPADDnA. to address the Scratchpad. When the data is in the temporary register and the addressing is done at the next clock pulse, the data is written into the selected Scratchpad. The temporary register is used for swap routines in which reads and writes are done during the same micro.

When sourcing the Scratchpads, the addressing is handled the same as the Write operation. Once the addressing is at the Scratchpad memory chip and the term PADWE/A. is TRUE, the Read data is available at the output. Data is output as APADnn.. or BPADnn.. and is available through a 4:1 multiplexor to the MEX.

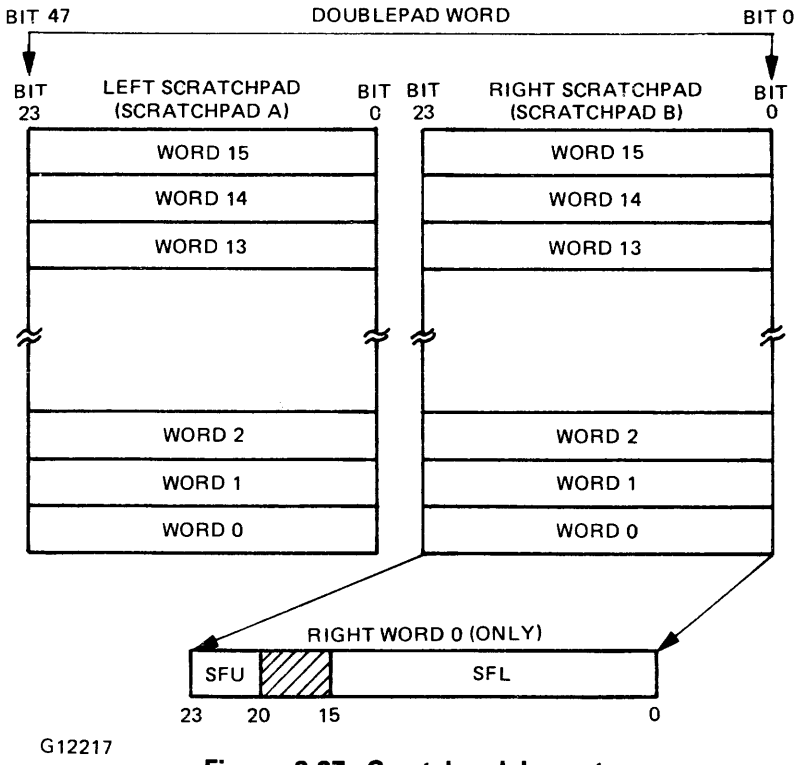


Figure 2-37. Scratchpad Layout

CC AND CD REGISTERS

The two 4-bit registers designated as CC and CD are used for the storage of various processor states and conditions.

CC Register

CC(3)

The control panel state lamp flip-flop, when TRUE, causes the D/M panel STATE lamp to light.

CC(2)

The real-time clock interrupt signal is developed from the system clock in the MBU. The interrupt signal is received and is used to set CC(2) once every 100 milliseconds.

CC(1)

The I/O Bus service request interrupt level is derived from the I/O controls connected to the processor's I/O bus. The level is the result of a service request by one or more controls and is used to set the I/O interrupt bit.

CC(0)

The control panel interrupt level is derived from the ON position of the D/M panel INTERRUPT switch. The level from the switch is used to set the interrupt bit. This flip-flop also drives the lamp behind the INTERRUPT push button on the Op panel.

CD Register

CD(3)

The Memory Error Interrupt flag is set on the next clock if the following conditions as reflected in PERM or PERP registers are TRUE:

M-register parity error
Cache Key parity error
Cassette parity error
Uncorrectable S-Memory processor error
Error log register change
S-Memory field out-of-bounds
Micro time out

The system halts and the ERROR lamp lights.

CD(2-0)

The memory address out-of-bounds signals are derived from logic which compares the contents of the FA register with the contents of the BR and LR registers on all memory accesses. The state of the out-of-bounds override control bit does not prevent setting the of out-of-bounds interrupt bits but does allow the subsequent WRITE operation.

No reaction occurs as a result of any interrupt until the microprogram tests the interrupt bit.

CC and CD register bits can be reset by a micro or by the CLEAR push button. The bit stays reset for at least one clock period following the action regardless of the continued existence of the condition that initially set the bit. Any test micro executed in this clock period finds the bit reset.

MSSW REGISTER

MSSW, the microinstruction source switch register, is special in that 1) the two high-order bits are source only and 2) the output of the two low-order bits is bit-ORed with the MICRO SOURCE switch on the D/M panel. Any value can be loaded into the two low-order bits of this register, but the interpretation is the result of the bit-OR. (See Figure 2-38.)

MSSW can be sourced or sinked as a 4-bit register. All move instructions applicable to a 4-bit register are also applicable to this register. Sinking four bits to this register will result in loss of the two most significant bits.

As a source, the two most significant bits provide the following processor identification information:

MSSW(3)
When TRUE, this bit indicates that the processor designated A has been selected as the slave through the MASTER SELECT switch on the D/M panel.

MSSW(2)
This bit is wired TRUE to signify the B processor in a dual-processor environment.

During RUN, the micro source is controlled by the two low-order bits of MSSW, as follows:

MSSW(1, 0)=00
Micro source is Cache. On a miss, a new block of four micros from S-Memory is routed to Cache. This is the normal mode of micro-sourcing.

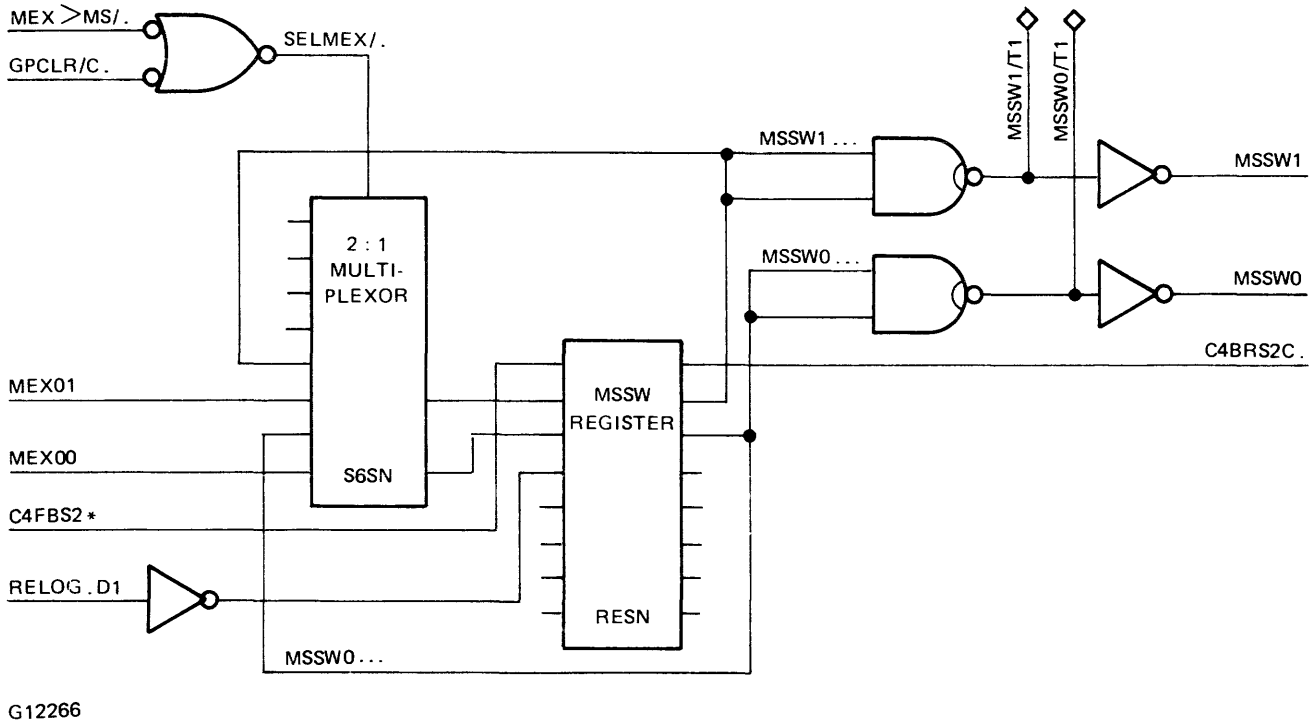


Figure 2-38. MSSW Register Logic

MSSW(1, 0)=01
Micro source is S-Memory only.

MSSW(1, 0)=10
Micro source is Cache only. A miss results in a halt.

MSSW(1, 0)=11
The M register is frozen (does not change); the micro in M is continuously executed.

INCN REGISTER

INCN, a 4-bit pseudoregister addressable as a source only, relates to port operations. Bits 3, 2, 1 and 0 reflect the states of certain interface lines between the processor and the Host Adapter in port-connect systems. In direct-connect systems, which have no Host Adapter, INCN bits 2, 1, and 0 are FALSE.

INCN(3)
Missing port device.

INCN(2)
High-priority port interrupt.

INCN(1)
Port interrupt.

INCN(0)
Port lockout.

PERM REGISTER

PERM, a 4-bit register, indicates that a problem has occurred in S-Memory. PERM register logic is shown in Figure 2-39.

This register is reset whenever the machine is started from a halt state. Also when halted, it is reset when LOAD is pressed with REGISTER SELECT at MEMORY

PERM(3)
If this bit is TRUE a micro has timed-out. The occurrence of a time-out sets the bit and immediately halts the processor.

PERM(2)
If this bit is TRUE on any memory request, either a Read or a write, the memory size as determined by MAXS in the MBU has been exceeded. All zeros are returned for out-of-bounds bits. There is no error-correcting action. During a fetch, a field out-of-bounds error halts the processor.

PERM(1)
This bit is goes TRUE whenever a change occurs in the error-log register (ELOG). ELOG is cleared after it has been read by the processor. Also, ELOG can be changed depending upon the level of the change information. Generally, the first error detected is logged. Then, if there is an error of greater importance, its status is written into ELOG. Three levels of errors represent the possible chain of events:

Case 1: a single-bit error which was corrected, replaced by an uncorrectable error from a non-CPU device, or replaced by an uncorrectable error from a CPU access.

Case 2: an uncorrectable error from a non-CPU device, replaced by an uncorrectable error from a CPU access.

Case 3: an uncorrectable error from a CPU access. The CPU error could be caused by a Read error before a Write operation.

PERM(0)

If TRUE, this bit indicates that there was an uncorrectable error as a result of a processor access. During a fetch, an uncorrectable processor error halts the processor.

The PERM register can be a 4-bit source or sink for Register Move (1C) micros. Also, the conditions that set the PERM bits also affect CD(3) and the RUN mode of the processor.

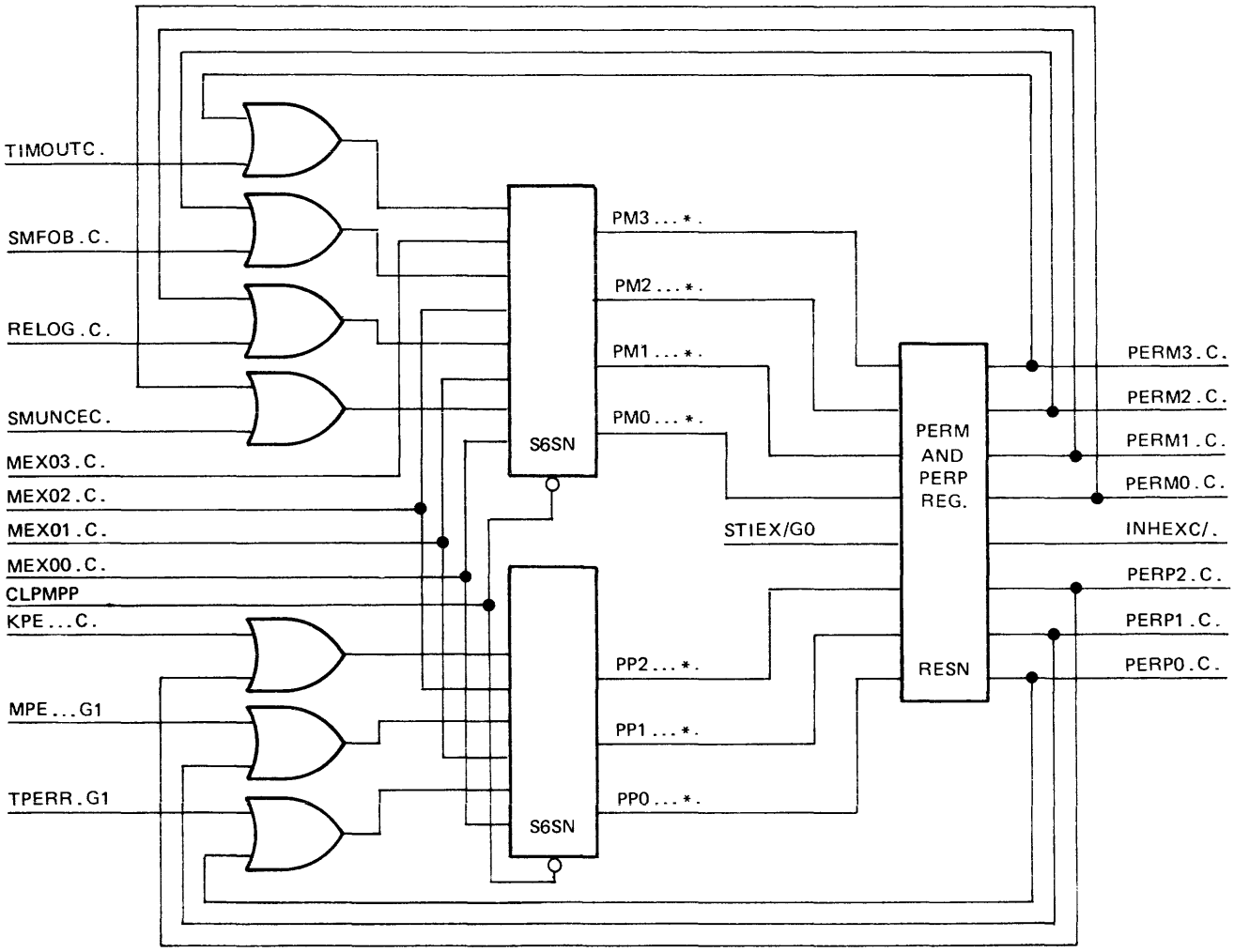


Figure 2-39 PERM and PERP Register Logic

PERP REGISTER

PERP is a 4-bit register that contains the error conditions generated in the processor. These errors are the result of the micro fetch mechanism, cassette tape Read parity error, Cache Key parity error, or a parity error on the micro in the M register. PERP logic is shown in Figure 2-39.

When the processor is halted, pressing START resets PERP. Also in halt, pressing LOAD with REGISTER SELECT at MEMORY resets PERP.

PERP(3) is not used. A zero results whenever PERP is sourced and bit 3 data is lost if sinked to this register.

PERP(2)
: This bit is set whenever there is a parity error on reading the Cache Key store. The parity check is odd parity over the Key (8 bits) and parity (1 bit). The validity bit enables the parity check output for a valid Key and micro in Cache. With Cache enabled, a Cache Key parity error during a fetch halts the processor. When disabled, it sets CD(3) and PERP(2).

PERP(1)
: This bit indicates a parity error on the output of the M register. When loading the M register from the D/M panel or when loading micros from the cassette during MTR mode, parity check is disabled. PERP(1) set causes a processor halt.

PERP(0)
: This bit when TRUE indicates a cassette Read error and causes a processor halt.

Card A Functions

The main function of card A is to interface with the Memory Base Unit (MBU) by providing a 24-bit bidirectional path to and from S-Memory. This path is used by data as well as address and control signals.

Card A also contains logic to check that the S-Memory address falls between the values in the BR and LR registers. A path is provided to get the contents of MAXS to the MEX. The BR and LR registers, as well as the FA register, can be used as general purpose registers; 24-bit paths are provided between the MEX and each of these registers.

Card A includes Scratchpad A, half of the overall Scratchpad described in the preceding subsection, with sixteen 24-bit words and a 24-bit path to and from the MEX.

Figure 2-40 is a block diagram of card A.

BR AND LR REGISTERS

BR, the Base register, and LR, the Limit register, are 24-bit registers used for memory protection and base-relative addressing. Each register is addressable as a source or a sink.

Memory protection is provided by checking the memory address in FA with BR and LR on all 7C and 2D memory cycles. A flag is set in the CD register when any address is outside these bounds.

S-Memory reads are always permitted, whether inside or outside the boundary; but writes and swaps are allowed outside the boundary only if CD(2), the override bit, is TRUE. An S-Memory address equal to BR or LR is considered in bounds. The memory protection is provided only on the initial pointer; protection is not provided on memory bits accessed when field length is greater than one. If the field is outside of the administrative memory, PERM(2) is set.

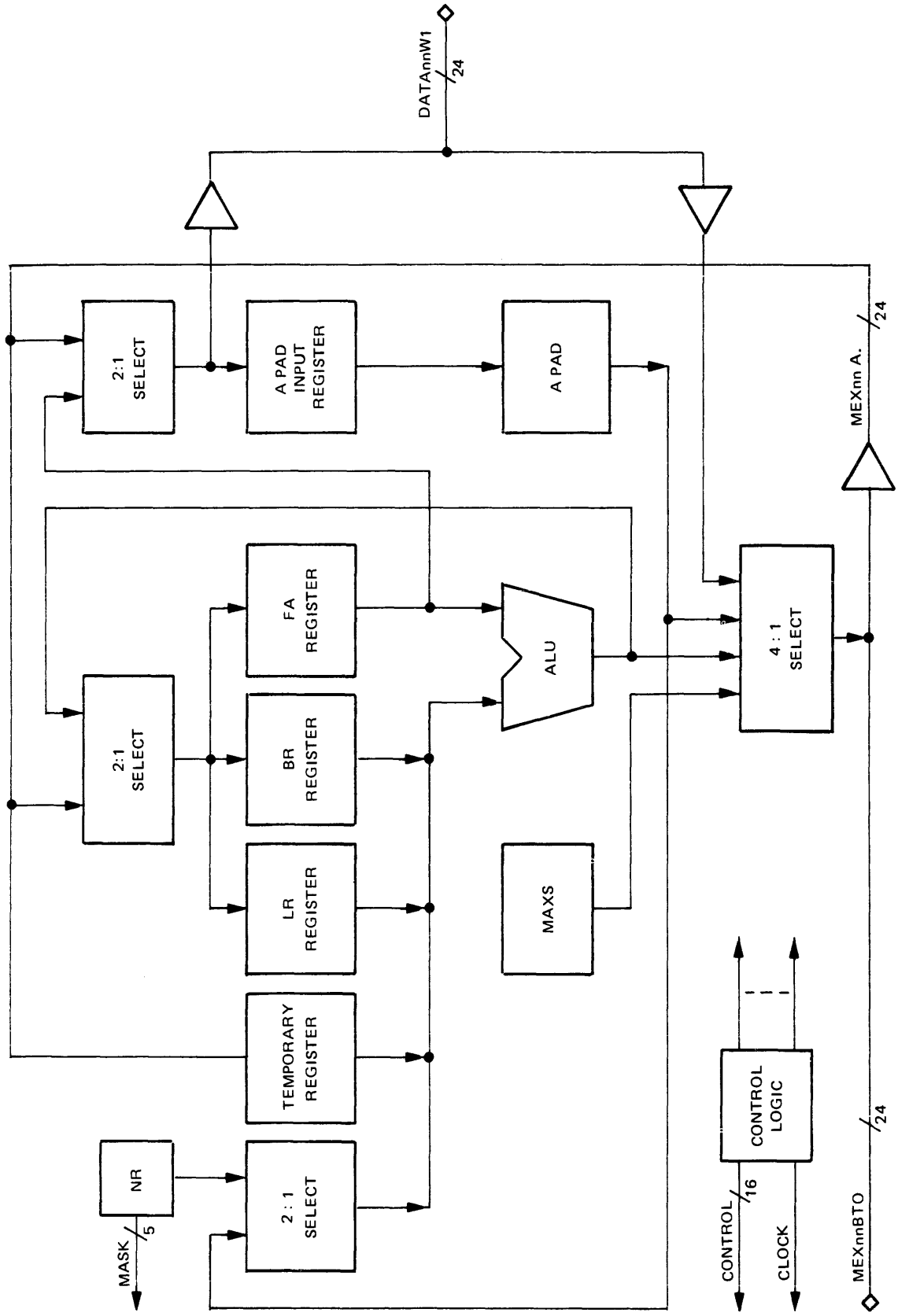


Figure 2-40. Functional Block Diagram of Card A

FA REGISTER

FA, the Field Address register, is a 24-bit register used primarily to hold an absolute bit address for S-Memory. FA can directly address any sequence of bits in S-Memory, starting at any point. The FA register is addressable as a source and as a sink. Refer to Figure 2-41.

To facilitate iteration through a memory field, the FA register can be counted up or down by a literal in a microinstruction or by the value contained in CPL. FA is incremented or decremented by a value in a left (A) Scratchpad word, and can be loaded, stored, and swapped, along with FB, into a double Scratchpad word.

The diagnostic variant of the Read/Write Cache (7E) micro uses FA as the source of the required Cache address.

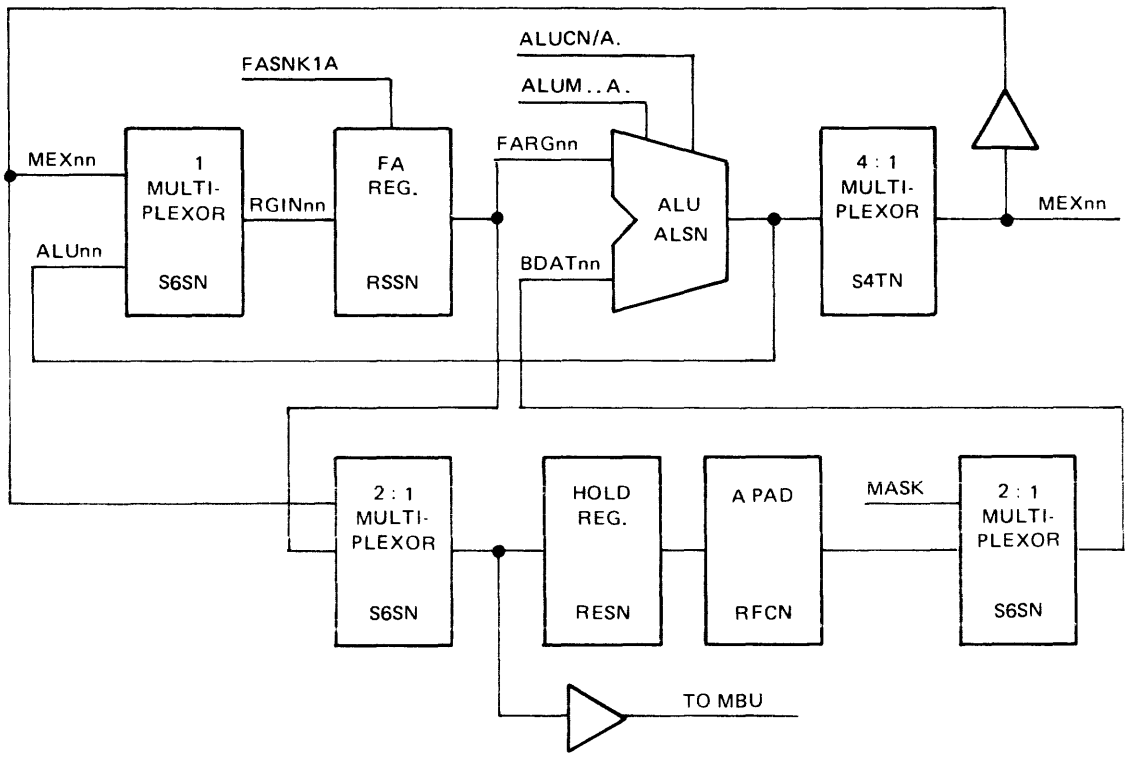


Figure 2-41. FA Register Logic

MAXS REGISTER

MAXS is a 24-bit pseudoregister that can be field adjusted to give the administrative or physical size of the installed S-Memory. MAXS is addressable as a source only and has 16K-byte resolution; that is, its least significant 18 bits are always zeroes.

Card B Functions

Card B includes the FB and CP registers and the B half of Scratchpad. The card contains four blocks that are related to FB, CP and the Scratchpads: bias logic; FLCN logic, ALU, and 4-bit data bus generation logic.

Depending on the instruction BSK1P/B., bias logic loads the CP register with a specific value and indicates whether the final value of CP equals zero.

FLCN logic generates signals indicating the result of a comparison of FL (a subregister of FB) and the corresponding portion of the first word (SFL) of Scratchpad B.

The ALU is used to increment the FL register by the amount specified by the five mask bits.

The 4-bit data bus generation logic comprises a multiplexor whose input select signals, input to card B, drive four lines on the backplane. The inputs to the multiplexor are the outputs of the FB register and the FLCN logic.

Figure 2-42 is a block diagram of card B

FB REGISTER

FB is a 24-bit register with three fields or subregisters: 1) Field Unit (FU), 4 bits; 2) Field Type (FT), 4 bits; 3) Field Length (FL), 16 bits. FL is further subdivided into FLC, FLD, FLE, and FLF, which are four bits each.

FB itself, as well as FL and each of the six 4-bit fields are addresseable as sources and sinks.

FU holds the length of the unit which makes up a field in memory. FT holds information on the field type. FL holds the total length of the field, up to 65,636 bits, and can be adjusted up or down by a literal in a micro or by a CPL value to facilitate iteration through a memory field. Overflow of FL is not detected; its value can go through maximum and wrap around. Underflow is detected; when field value reaches zero it remains at zero and wrap around does not occur.

Because FB is addressable in 4-bit groups, its contents are available for analysis and alteration through the 4-bit function box. The 4-bit Manipulate (3C), Bit Test (4C and 5C), and Skip (6C) micros can operate on the contents of FB. FB can be loaded, stored, or swapped along with FA into a double Scratchpad word.

FU and FL along with corresponding portions of the first cell of Scratchpad are used to set the various conditions of FLCN and the CP register through the Bias (3E) micro.

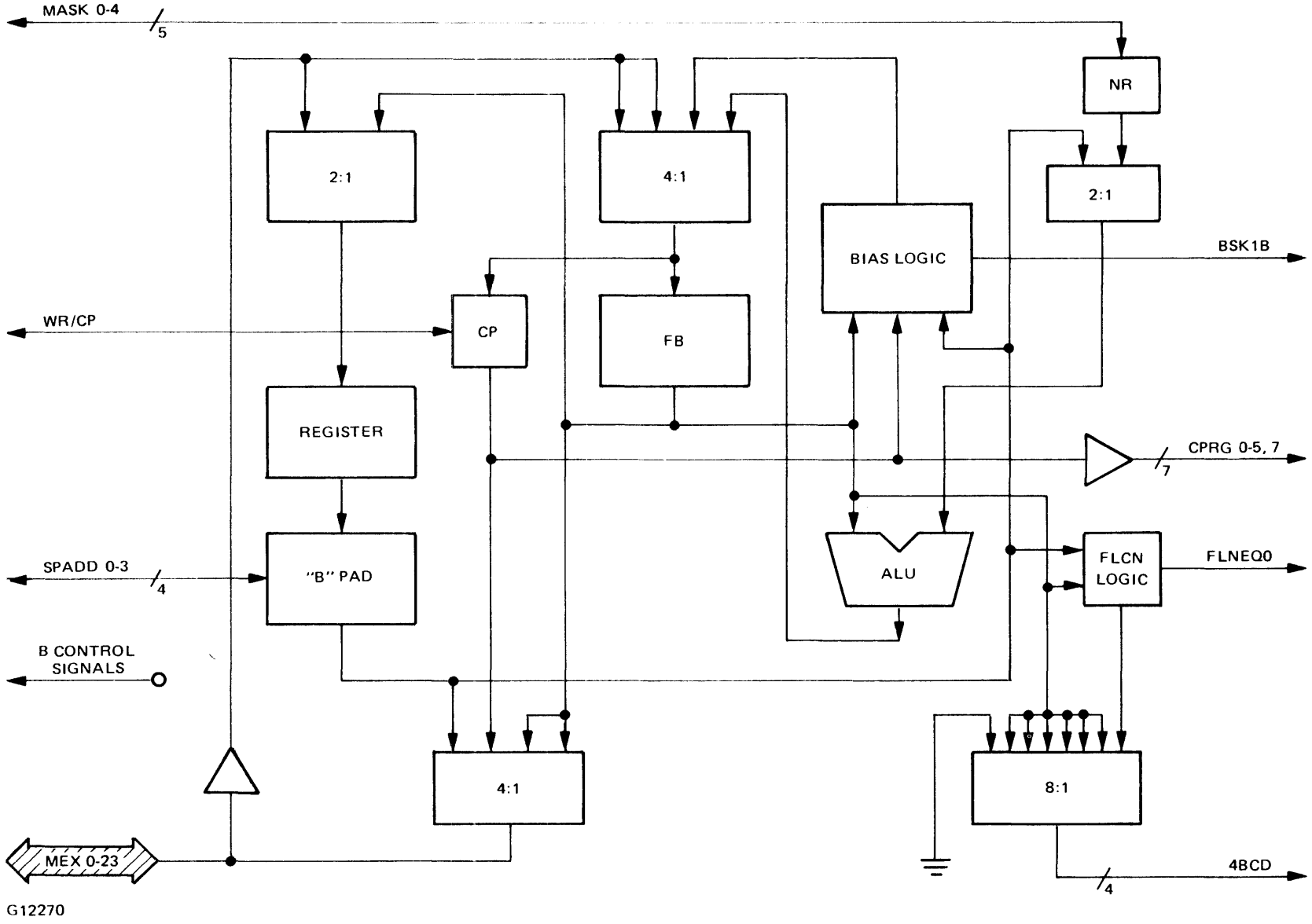


Figure 2-42. Functional Block Diagram of Card B

FLCN REGISTER

FLCN, the Field Length Conditions register, is a 4-bit, source-only pseudoregister that holds the results of comparisons of FL with the corresponding portion (SFU and SFL) of the first word of the right (B) Scratchpad. The bits of FLCN have the following meanings:

FLCN(3) FL = FLCN
FLCN(2) FL > FLCN
FLCN(1) FL < FLCN
FLCN(0) FL neq 0

CP REGISTER

The 8-bit entity called CP is comprised of the one-bit arithmetic unit carry flip-flop CYF, the two-bit unit control for the arithmetic unit CPU, and the five-bit variable data length control CPL. CYF is addressable individually by the Set Carry Flip-Flop (3C) micro, and CPU and CPL may be modified by the Bias (3E) micro. CP as a whole is addressable as a source or a sink.

The high-order bit of CPU has no special meaning; its low-order bit determines the unit of the input field to the ALU, with 0 for binary and 1 for BCD.

The value in CYF can also be read as BICN(2), but it cannot be changed.

Card Group D, E, F

Card D includes the following functions:

1. Buffering of the MOP lines from cards G and H, and decoding of the micros as they appear on the MOP lines.
2. Generation from PROMs of the control signals for cards E and F.
3. Enabling of PROM outputs resulting from the decoding of micros to the nanoregister.
4. Timeout logic and control signals for the clock (K) card.
5. Nano-complete logic, to count the nanoinstruction sequence for the processor. This is required for micros that decode into more than one nano.
6. General Processor Clear (GPCLR) logic.
7. I/O spreader logic for assuring sufficient time for I/O data transfer.
8. The S-Memory control interface.

Card E includes

1. The X and Y registers and related logic.
2. The 24-bit function box: an ALU for the X and Y registers, with BCD sum correction and variable-width output masking under control of CPL.
3. BICN, XYCN, and XYST logic for generating the results of X/Y comparisons.
4. Mask generation and binary to BCD logic.

Card F includes

1. Rotate and masking logic for the T register.
2. The CA, CB, T, and L registers.
3. Skip and branch logic and a nanoregister.
4. A 4-bit function box.

Figures 2-43, 2-44, and 2-45 are functional block diagrams of cards D, E, and F respectively.

Card D

Card D (Figure 2-43) 1) generates control signals for cards E, F, and K and for the MBU, 2) handles the sequencing of micros, and 3) generates the General Processor Clear signal (GPCLR).

Micro Decode logic decodes the MOP lines. Depending on the result, a specific output is driven LOW while the others stay HIGH. A HIGH Inhibit Decode (INHDEC) level forces all outputs to stay high.

Nano Generation logic includes PROMs that generate control signals to enable the decoded micro. The Nano Register latches the output of the nano generation logic.

The Time Out logic is basically a counter. When Nano Execute Complete is low, a value of "1100" is loaded. The counter is incremented by 1 with the 100-ms Real Time Clock from the MBU. If the counter's carry output goes TRUE, a time out is flagged in PERM(3) and the processor halts. Micro time-outs are disabled on moves from the U register.

The I/O Spreader consists of timing logic to create enough spacing between two DATA/.D0 signals. The ENDIO.*. signal indicates that enough time has elapsed.

4-BIT DATA SOURCE/SINK

Moves of the contents of 4-bit registers are handled by making six copies of the 4-bit field and sending the output from the register to a set of four multiplexors. The multiplexors are addressed by Nano Register outputs (terms 4FBMnnD0). The multiplexor output is 4xREGNF., in which x identifies the register. Each set of 4-bit registers is then selected for output to the MEX by a set of multiplexors addressed by 4FBRSnF.. This data, called 4REGn.F., is available to the MEX by the output multiplexors on each data card.

To sink a 4-bit register, MEX data bits 0-3 are sent to a set of six buffers making six copies of the data bits available to the nano register, which selects the appropriate register input from information obtained by the decoding of the micro.

NULL REGISTER

Null, a 24-bit pseudoregister, when addressed as a source provides all zeros to the destination. Null can be sinked to send data "nowhere", useful when popping TAS because the pointer can be moved without changing the contents of the A-Stack or affecting other registers.

Card E

Card E (Figure 2-44) primarily consists of the X and Y registers and related logic.

X AND Y REGISTERS

The X and Y registers are two 24-bit general purpose registers used primarily to hold and to act as the source for the two operands of the 24-bit ALU. Each is addressable as a source and a sink.

Both registers are capable of undergoing all shift/rotate operations together (concatenated) or separately; also, X can be normalized. Y is used to hold data in diagnostic Cache Read/Write (7E) operations and X is used to hold data in diagnostic Cache Write (but not in diagnostic Cache Read) operations. X and Y are compared in the Cassette Control (2E) micro to invoke variant-specified halts. On the Diagnostic Read/Write Memory (11D) micro, Y is the destination for the echo response of the Write data, the address, and the ELOG register.

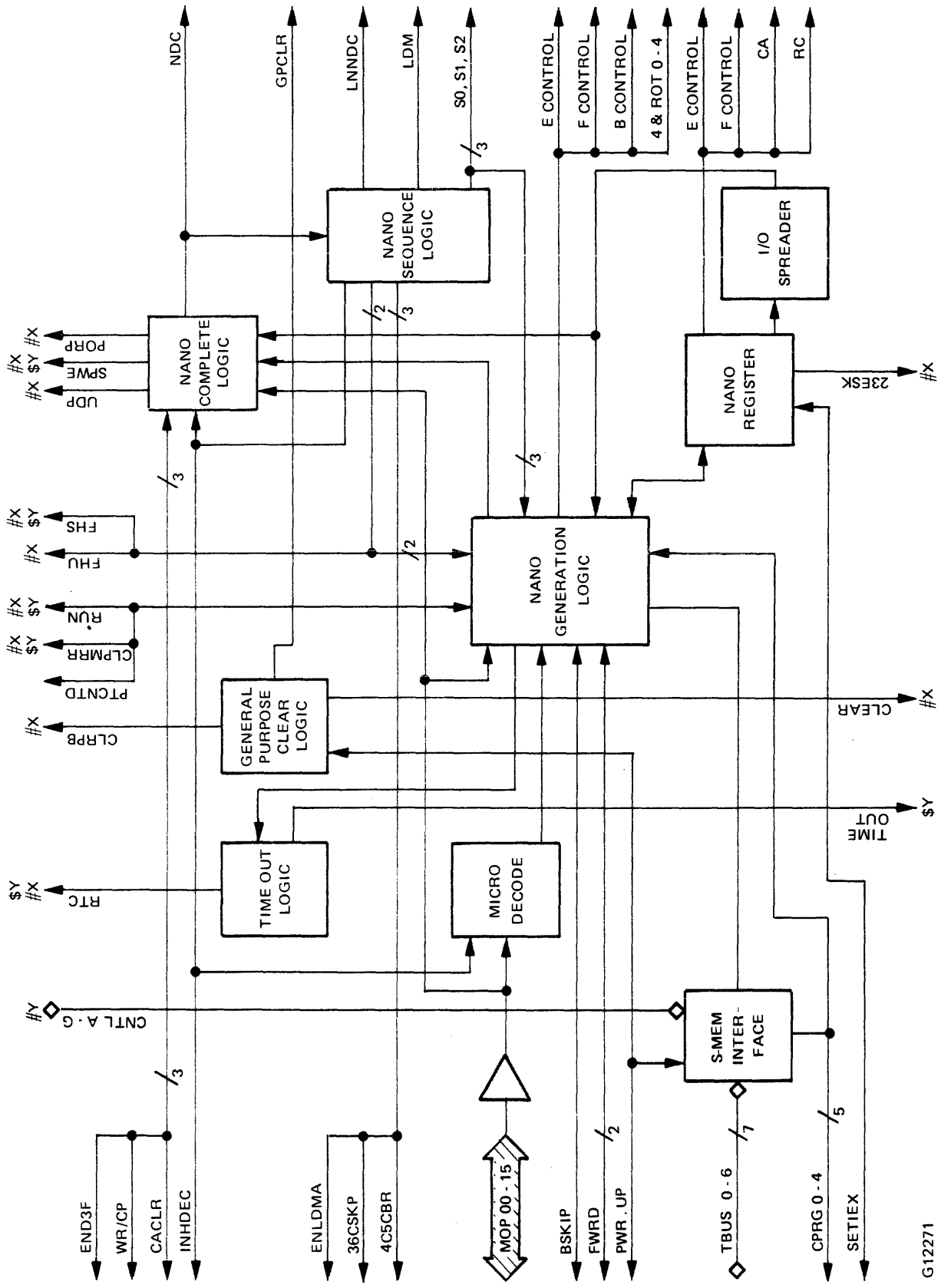


Figure 2-43. Functional Block Diagram of Card D

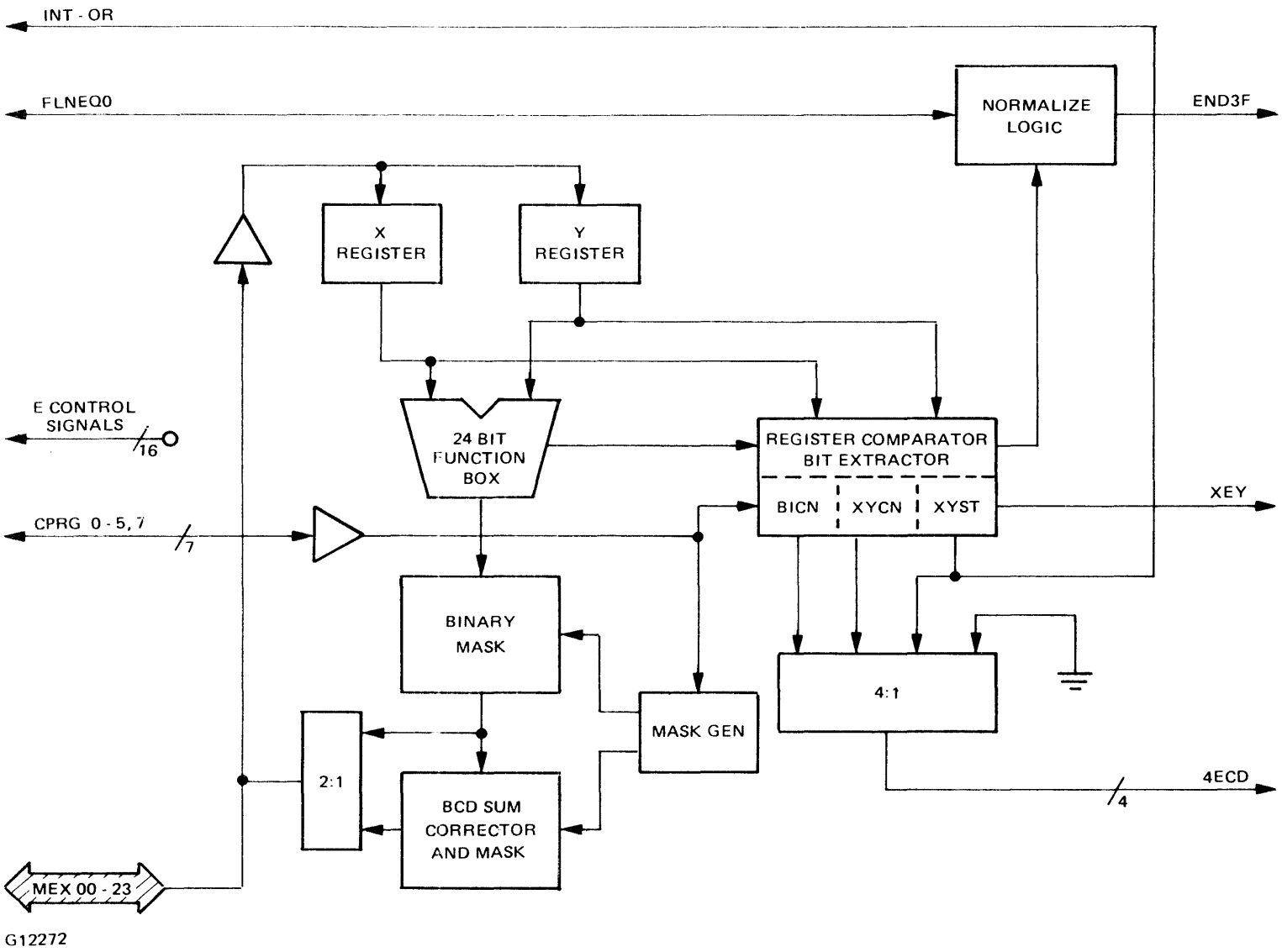


Figure 2-44. Functional Block Diagram of Card E

24-BIT FUNCTION BOX

The 24-Bit Function Box includes a 24-bit ALU and a 24-bit combinatorial unit. Data inputs are the contents of the X and Y registers and the Carry flip-flop (CYF).

All results from the combinatorial unit are generated immediately and are continuously available to micros. A move to one of the input registers or an alteration of a value in the CP portion of the C register immediately generates a new result that is available to the next micro and accessible by a move of the contents of a result pseudoregister to a destination register or by a test of one of the 4-bit condition registers.

The ALU provides the following outputs in source-only pseudoregisters for the functions of two operands:

AND
OR
EXCLUSIVE-OR
SUM
DIFFERENCE
EQUAL TO
LESS THAN
GREATER THAN
BINARY COMPLEMENT
BINARY MASK

The ALU portion of the 24-Bit Function Box also makes use of CPU, the control for the arithmetic unit, and CPL, the 5-bit variable operand length. CPU and CPL control over ALU results is shown in Table 2-8.

For valid arithmetic operations, the operand length (as specified by CPL) must be an exact multiple of the length of the unit specified by CPU.

The contents of the following registers are immediately available to the micro: SUM, DIFF, XANY, XORY, CMPX, CMPY, MSKX, MSKY, BICN, XYCN, XYST. These registers are described in detail in the B 1900 Field Engineering Technical Manual, Volume 1, Form 1127388.

Table 2-8. CPU and CPL Control over ALU Results

CPU	Unit Type	Possible CPL Values	Data Type
00 or 10	1-bit operands	0 to 24	Binary
01 or 11	4-bit operands	0,4,8,12,16,20,24	BCD

Card F

Card F (Figure 2-45) includes two 4-bit registers CA and CB, two 24-bit registers T and L, and the associated logic.

Shift and rotate logic rotates the output of the T register. This logic is controlled by the five signals 4*ROT0D0 through 4*ROT4D0, all inputs to card F. Mask Generation logic, controlled by signals that are also card F inputs, produce left-justified and right-justified ones. Mask logic performs an AND function between the mask and the output of the shift/rotate logic.

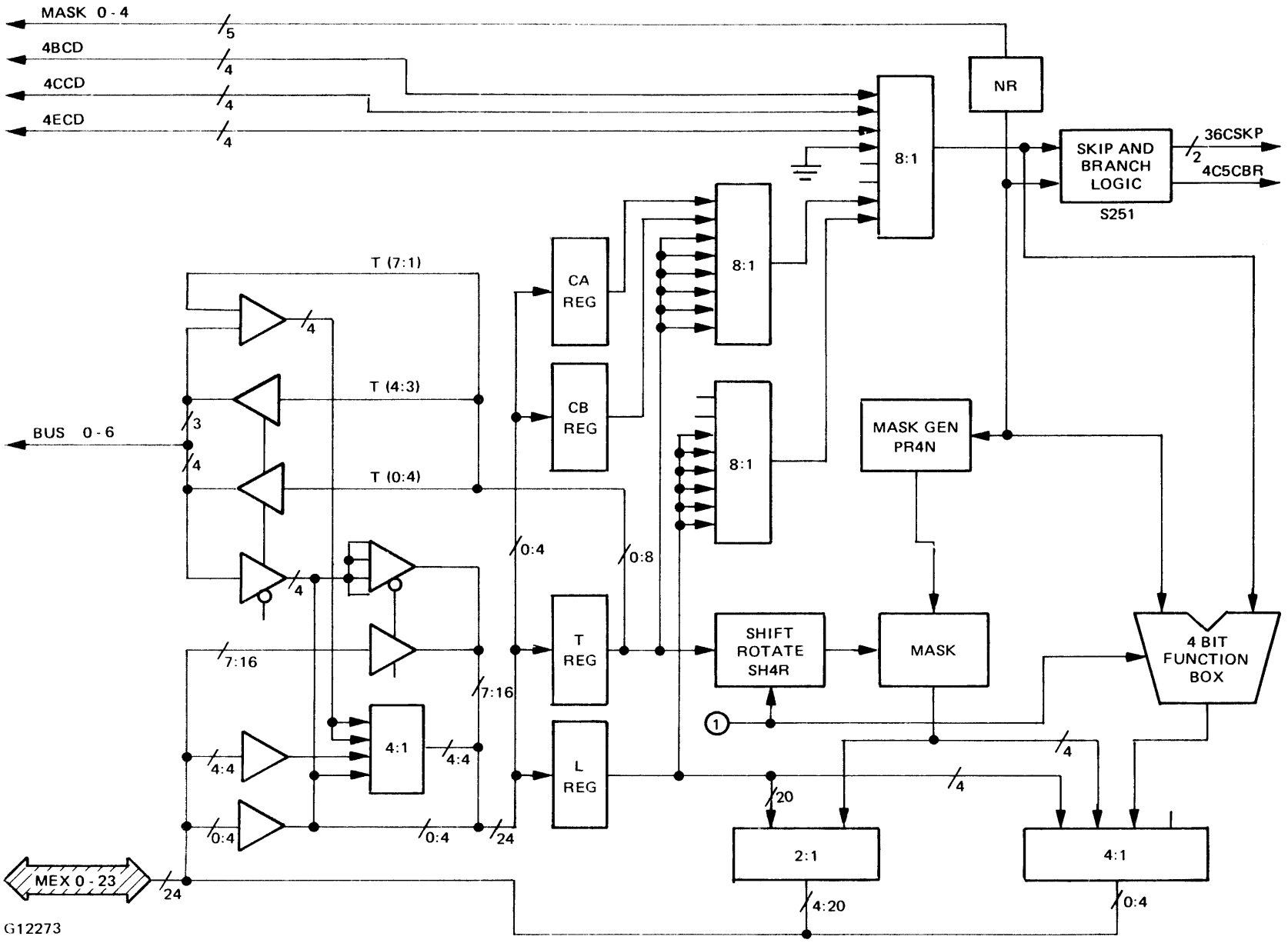


Figure 2-45. Functional Block Diagram of Card F

The 4-Bit Function Box is used for arithmetic and logic functions between literal bits input to card F and any of the 4-bit registers selected by the 8:1 Multiplexor that drives the inputs of the 4-Bit Function Box.

Skip & Branch logic is comprised of two multiplexors whose input select signals are card F inputs. This logic drives three signals on the backplane; 36CSKPE0, 36CSKPF2, and 4C5CBRF2.

The T-bus is used for bidirectional transfer of the least-significant seven bits of the T register to and from this card.

T REGISTER

The T register is a 24-bit general purpose register used primarily for the interpretation of the S-language instructions. T and each 4-bit subregister TA, TB, TC, TD, TE, and TF are addressable as sources and sinks. Since T is addressable in 4-bit groups, its contents are available for analysis and alteration by the 4-Bit Function Box. The micros Manipulate (3C), Skip (6C) and Bit-Test-Branch (4C, 5C) can operate on T data. The Bind (4F) micro moves the sum of L and T to the A register.

The T register is one of four registers (X, Y, L and T) capable of both Read/Write (7C) and diagnostic Write Cache (7E) operations. T is also capable of Shift/Rotate (10C) and Extract (11C) operations.

The T register is also used for port/channel information during a Dispatch (1E) micro. The four low-order (0-3) bits provide channel information with addresses available from @0@ to @F@. Bits 4, 5, and 6 provide port information. The T bus handles the transfer of this data to the Host Adapter via card D.

L REGISTER

The L register is a 24-bit general purpose register used typically to hold logical flags for the micro-program code. The L register and each 4-bit subregister LA, LB, LC, LD, LE, and LF are addressable as sources and sinks.

Dispatch operations use the L register as the source or sink for a 24-bit message (usually an address), which is stored in or fetched from S-Memory location zero.

The Bind micro uses the L register as the source of the 24-bit value to be added to the T register, with the sum going to A register.

Since the L register is addressable in 4-bit groups, its contents are available for analysis and alteration through the 4-Bit Function Box. The Manipulate (3C), Bit Test (4C, 5C), and Skip (6C) micros can operate on L data.

The L register is one of four registers (X, Y, T and L) capable of Read/Write Memory (7C) and diagnostic Cache Write (7E) operations.

CA AND CB REGISTERS

The 4-bit groups designated as CA and CB have no special functional assignment and are available as general purpose 4-bit storage registers. Their contents are available for analysis and alteration via the 4-Bit Function Box, and the Manipulate, Skip and Bit-Test-Branch micros are applicable to these registers.

4-BIT FUNCTION BOX

The 4-bit Function Box (4-bit arithmetic and combinatorial section of the processor) can accept the contents of any of the 4-bit registers and pseudoregisters below as one input. The second input is obtained from the applicable micro.

TA	TB	TC	TD	TE	TF	PERP
LA	LB	LC	LD	LE	LF	
FU	FT	FLC	FLD	FLE	FLF	
CA	CB	CC	CD	INCN	PERM	
BICN	XYCN	XYST	FLCN			

Outputs include the results of most of the commonly used functions between two operands; for example: SET, AND, OR, exclusive-OR, and binary sum and difference. Outputs are directed back to the source register if the source register is not a pseudoregister.

The sum output can be tested for overflow and the difference output can be tested for underflow. Based on the result, a skip of one instruction can be made.

The 4-bit Function Box also provides for the selective testing of one of the bits of a four-bit group and for relative branching based on the result of the test. A skip of one instruction based on the result of testing on a combination of up to four bits in the group is also provided.

BICN, XYCN, XYST, FLCN, and INCN are pseudoregisters and can be sourced. They can be changed only as a result of changing the condition which they reflect.

SECTION 3
CIRCUIT OPERATIONAL DETAIL – B 1900 S-MEMORY

INTRODUCTION

The B 1900 Memory Base Unit (MBU) consists of two logic cards (R6 and S6) and either two to eight storage boards (128KB to 1MB) or two to sixteen storage boards (128KB to 2MB). The MBU provides all of the necessary memory control for storage access and interfaces with a memory-using device (a processor or a multiline control).

Descriptions of the S-Memory storage architecture and memory operations, provided in the B 1900 FETM, Volume 1 (Form No. 1127388), are repeated here for convenience.

S-Memory Storage Architecture

S-Memory storage consists of four randomly-accessible stacks, each 22 bits (one word) wide. The 22-bit word includes 16 bits (2 bytes) for data and six bits for the error correcting code (ECC).

S-Memory organization is shown in Figure 3-1.

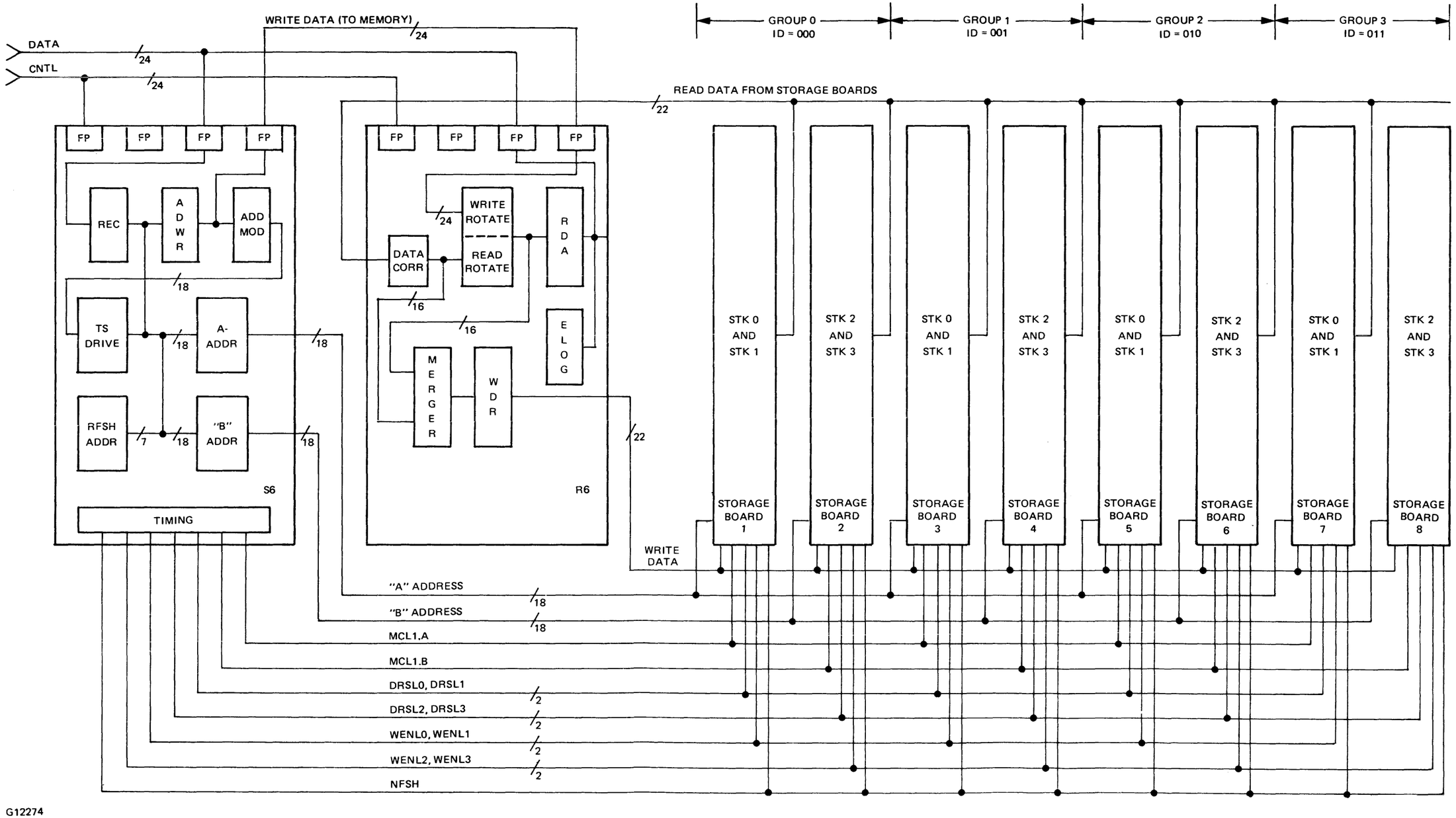


Figure 3-1. B 1900 S-Memory Organization

Error Correction Code

The MBU utilizes a 6-bit error correction code (ECC) to enable detection and correction of all single-bit errors within the 22-bit memory word. Figure 3-2 illustrates the memory word.

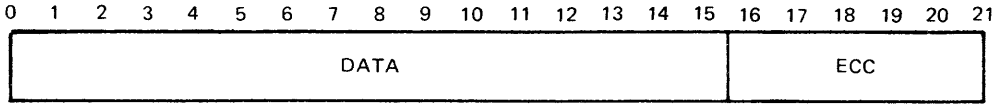


Figure 3-2. 22-Bit Storage Word

The six ECC bits permit the successful detection of the majority of multiple-bit errors. All errors involving two bits are successfully detected. However, certain errors involving three or more bits can be mistaken for single-bit errors and a faulty correction can result. Error detection and correction is accomplished by a 6-bit syndrome. Table 3-1 summarizes ECC generation, Table 3-2 summarizes syndrome generation, and Table 3-3 shows the syndrome patterns for single-bit errors.

Table 3-1. Check Bit Generation

16 = (0 + 1 + 2 + 3 + 4 + 5 + 6 + 7)/
17 = (0 + 1 + 2 + 8 + 9 + 10 + 11 + 12)/
18 = (3 + 4 + 8 + 9 + 10 + 13 + 14 + 15)/
19 = (0 + 3 + 5 + 6 + 8 + 11 + 13 + 14)/
20 = (1 + 5 + 7 + 9 + 11 + 12 + 13 + 15)/
21 = (2 + 4 + 6 + 7 + 10 + 12 + 14 + 15)/

\+ = logical exclusive or
/ = logical not

Table 3-2. Syndrome Generation

S1 = (0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 16)/
S2 = (0 + 1 + 2 + 8 + 9 + 10 + 11 + 12 + 17)/
S3 = (3 + 4 + 8 + 9 + 10 + 13 + 14 + 15 + 18)/
S4 = (0 + 3 + 5 + 6 + 8 + 11 + 13 + 14 + 19)/
S5 = (1 + 5 + 7 + 9 + 11 + 12 + 13 + 15 + 20)/
S6 = (2 + 4 + 6 + 7 + 10 + 12 + 14 + 15 + 21)/

\+ = logical exclusive or
/ = logical not

Table 3-3. Syndrome Patterns for Single-Bit Errors

		Data Bits																ECC Bits					
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
S2	0	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0
S3	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	0
S4	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0
S5	0	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	0	0	1	0
S6	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	0	0	0	1

NOTE

Any syndrome pattern not in the table above represents a multiple-bit error.

MBU Operations

The MBU is capable of performing thirteen separate operations. Twelve of these are commanded by the host system. One, Refresh, is commanded internally by the MBU. The operations are listed in Table 3-4.

Table 3-4. MBU Operations

Micro Bits		Mode Bits	
4/ 3/ 2/ 1/ 0/	REQ/	3/ 2/ 1/ 0/	Operation
Don't Care	1	X X X X	No request
Don't Care	0	1 1 1 1	Reserved
Don't Care	0	1 1 1 0	Refresh (Internal only)
Don't Care	0	1 1 0 1	Micro fetch (four stacks)
Don't Care	0	1 1 0 0	Reserved
Field Length	0	1 0 1 1	Defined field Read (7C)
Field Length	0	1 0 1 0	Defined field Write (7C)
Field Length	0	1 0 0 1	Defined field Swap (2D)
Don't Care	0	1 0 0 0	Reserved
Don't Care	0	0 1 1 1	22-bit Read (11D)
Don't Care	0	0 1 1 0	22-bit Write (11D)
Don't Care	0	0 1 0 1	Echo Write data (11D)
Don't Care	0	0 1 0 0	Echo address (11D)
Don't Care	0	0 0 1 1	Read and clear ELOG (11D)
Don't Care	0	0 0 1 0	No-op (HA diagnostic)
Don't Care	0	0 0 0 1	Dispatch Read (1E)
Don't Care	0	0 0 0 0	Dispatch Write (1E)

NOTES

X = Don't Care.
/ = Not (active low).
The values given for the various signals are those on the interface to the MBU.

Refresh

This MBU-generated command is used to recharge the memory chips on the storage boards. The MBU generates one Refresh request every 16 microseconds. If a memory-using device (user) simultaneously requests a memory cycle, the Refresh cycle is postponed and the requestor is granted the cycle. After a Refresh request is generated by the MBU, the actual Refresh operation can be postponed up to a maximum of 12 microseconds. If a Refresh request is not honored within 12 microseconds, all memory requests from using devices are locked out and a Refresh cycle is performed. The maximum postponement time of 12 microseconds ensures that the Refresh request is honored before another Refresh request is made. Refresh requests are generated every 16 microseconds regardless of any previously postponed Refresh cycles.

Refresh can be turned off and on either programmatically or by a jumper on the backplane. Also, the Echo Address request with 1 in the least significant bit of the FA register disables Refresh.

Refresh may be enabled by the Diagnostic Echo Address request with a "0" in the least significant bit of the register or by clearing the system. Grounding the RFSHEN.0 pin on the backplane of card S6 disables Refresh.

Micro Fetch (Micro-operator Stream Mode)

The MBU responds to a Micro Fetch request by returning, in four successive clocks, four 16-bit data fields plus an odd parity bit for each field. The MBU forces the accesses to start at stack 0, bit 0.

Accesses to the four micros are always in the forward direction. The bit pointer, stack pointer, and field direction sign do not affect this operation. Single-bit errors are corrected and multiple-bit errors are reported.

To ensure proper error handling, the value in micro bits 4 through 0, which the MBU interprets as the transfer width, must not be in the range of 25 through 31 (11001-11111). Transfer width values from 0 to 24 have no effect on this operation.

Defined Field Read

This is the basic memory Read operation. Data of any field length from zero to 24 bits can be accessed at any bit 0 through 15 within any stack 0 through 3 in either the forward or reverse direction. All the above information must be specified by the using device. All single-bit errors are corrected and multiple-bit errors are reported.

Since data can span from one to three stacks, a Read that accesses data from two stacks takes one clock longer than a Read that accesses data from one stack; and a Read that accesses data from three stacks takes one clock longer than a Read that accesses data from two stacks. The one exception is the reverse Read with the bit pointer pointing at bit 0, which is always at least a 2-stack access even though the data may lie entirely within one stack.

Defined Field Write

This is the basic Write operation to memory. As specified by the using device, any field length of data 0-24 bits can be written starting at any bit 0-15, within any stack 0-3, in either the forward or reverse direction.

This is actually a Read-Modify-Write operation. Data is first read out and any single-bit error is corrected. Then, Write data is substituted into the appropriate bit positions to create a new 16-bit data word and a new ECC is generated for this word. Next, the entire 22 bits is written into storage.

A multiple-bit error detected when the data is read from a stack is reported. A new ECC is generated for the new 16-bit data word, and this ECC is inverted before it is written into the storage board to leave an indication in memory that the word may still contain an error. The inverted ECC, even though correct, causes an uncorrectable error to be detected whenever the word is reread.

Transfer width requests of 25 or 26 are truncated to 24 for data manipulation purposes and special conditions are set for the operation. A transfer width of 25 causes correct ECC bits to be written into memory for all stacks accessed, regardless of any error conditions encountered in the read-out phase. A transfer width of 26 causes the correct ECC bits to be inverted before they are written into memory for all stacks accessed regardless of any error conditions encountered in the read-out phase.

Transfer width requests of 27, 30, or 31 are interpreted by the MBU as a transfer width of 26; a transfer width request of 28 is interpreted as 24; and a transfer width request of 29 is interpreted as 25.

If the NOWRITE bit from the using device also comes TRUE for the Write operation, the cycle continues with its normal timing except that all Write enables to the storage boards are disabled and no data is written into memory.

Defined Field Swap

The defined field Swap command causes the MBU to perform a defined field Read immediately followed by a defined field Write in one uninterruptable operation. The segments of this single operation are as specified in the subsections titled Defined Field Read and Defined Field Write.

22-Bit Read (Diagnostic Read)

Diagnostic Read causes the 16 data bits and 6 ECC bits of a memory word to be read. At the MBU interface, the 16 data bits are right justified and the six ECC bits are left justified. An odd parity bit is supplied on the 16 data bits. Single-bit errors are not corrected but single-bit and multiple-bit errors are reported. The parity bit is even for the 16 data bits if any errors are detected in the data.

The data format for the 22-Bit Read operation and the 22-Bit Write operation is shown in Figure 3-3 and discussed in the subsection that follows.

The MBU forces the access to start at bit 0 of the appropriate stack. The access is one stack wide. Therefore, the bit pointer and field direction sign do not affect this operation.

To ensure proper error handling, the transfer width must not be in the range of 25 through 31. Transfer width values of 0 through 24 do not affect this operation.

22-Bit Write (Diagnostic Write)

This operation updates the entire 22 bits (16 data and six ECC) of a memory word with data supplied by the processor. At the MBU interface, the 16 data bits must be right justified and the 6 ECC bits left justified, as shown in Figure 3-3. The parity bit is ignored for 22-Bit Write. There is no error correcting or reporting for this operation.

The MBU forces the Write to occur on a stack boundary starting at bit 0 of the appropriate stack. The operation is not affected by transfer width, by bit pointer, or by field direction sign.

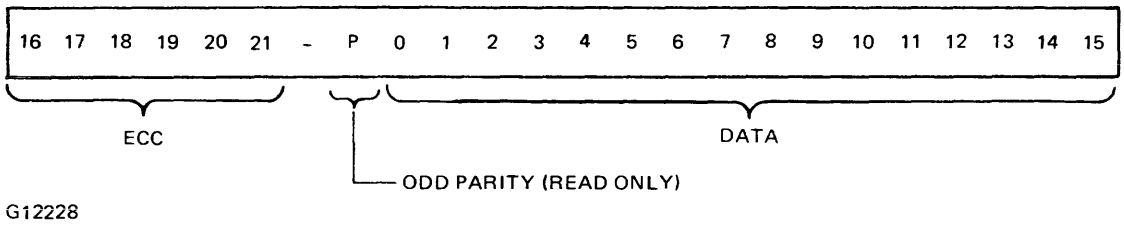


Figure 3-3. Diagnostic Read/Write Data Format

Echo Write Data

This command instructs the MBU to return the Write data sent to it by the processor.

Echo Write Address

This command instructs the MBU to return the address sent to it by the processor, modified either in the forward direction or the reverse direction depending upon the Field Direction Sign (FDS). If FDS = 0, the address returned is ADDR+16; if FDS = 1, the address returned is ADDR-16. ADDR is changed by 16 because modification of the address in the MBU begins with the stack pointer, specified by address bits 4 and 5.

The path for Echo Address is through the Error Log Register; thus, not all bits of the address are returned to the requesting device.

Read and Clear Error Log

This is not an access to the storage boards, but an access to the Error Log Register (ELOG) within the MBU. The MBU returns the contents of the ELOG at the end of the cycle.

The operation is completely defined by the simple request for a Read and Clear of the ELOG; that is, the bit pointer, stack pointer, transfer width, and field direction sign are not required.

No-Op (Host Adapter Diagnostic)

No-Op is executed whenever the processor issues a Host Adapter-3 diagnostic command. The only action taken by the MBU during a No-Op is to supply the processor with the required interface control signals.

Dispatch Read

Dispatch Read is a defined field Read with the following parameters forced by the MBU: address = 0, stack pointer = 0, bit pointer = 0, and transfer width = 24

Data is accessed in the forward direction. All single-bit errors are corrected and all multiple-bit errors are reported.

Transfer width requests of 0 through 24 do not affect this operation; the MBU forces the transfer width to 24. However, transfer width requests of 25 through 31 interfere with proper error handling.

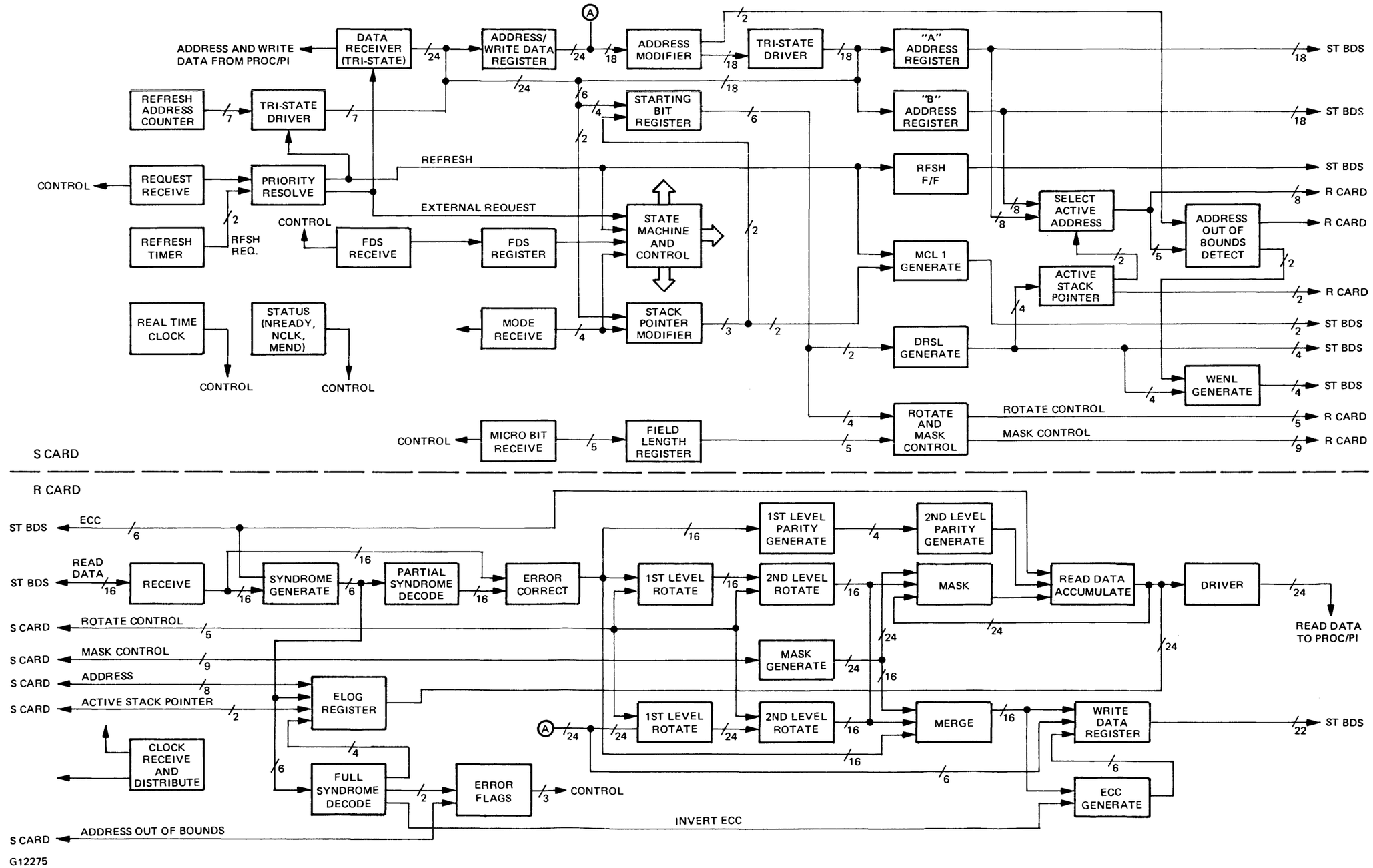


Figure 3-4. B 1900 MBU, Functional Block Diagram

Dispatch Write

Dispatch Write is a defined field Write with the same parameters forced by the MBU as are specified for Dispatch Read, above. All data is accessed in the forward direction.

The operation is always a 24-bit operation, it is unaffected by by other transfer widths that may be specified. However, specified transfer widths of 25 through 31 give the same results as described in the subsection titled Defined Field Write. The effect of a TRUE NOWRITE bit is also discussed there.

MBU FUNCTIONAL DETAIL

Figure 3-4 is a functional block diagram of the logic modules and data paths referred to in this section. The MBU includes both memory control and memory storage, to allow a using device, either a processor or a multiline control, to access as well as store data within the physical boundaries of S-Memory. The MBU performs the data manipulations needed to align the data properly for access, and detects and reports errors that might violate data integrity. The following subsections describe the MBU's functions.

Memory Cycle Initiation

The MBU initiates memory cycles in response to two types of requests: 1) external requests from using devices, signalled by an active state on the memory interface mode lines, and 2) internal requests, either low-priority or high-priority, from the MBU's Refresh timer. The high to low priority sequence for granting memory requests is 1) high-priority Refresh request, 2) external request, and 3) low-priority Refresh request.

Requests for memory cycles are monitored during the MBU's idle state. If there is a conflict between an external request and a low-priority Refresh request while the high-priority Refresh request is inactive, the external request is granted the cycle. If the memory is kept busy with external requests that continually lock out the low-priority Refresh request, the high-priority Refresh request is turned on. The Refresh cycle is then performed, immediately after completion of the current cycle.

The externally requested Refresh cycle and the high-priority Refresh cycle are concatenated; that is, they appear as one continuous cycle to the memory-using device. After the Refresh cycle has been performed, both Refresh request lines are reset.

State Machine

The MBU is controlled by a state machine that is located on card S6. There are 101 valid states in the MBU. These states are depicted in Figure 3-5 for card S6 and Figure 3-6 for card R6.

There are seven state lines used internally on card S6 (STATE0S. to STATE6S.) and used externally on card R6 (STATE0/0 to STATE6/0). These lines determine the precise state that the MBU is in. The state lines on both cards are used to address PROMs that supply most of the control signals used in the MBU. The seven lines give a total of 128 possible states, but 27 of these states (128-101) are not used.

The state of card S6 is expressed by encoding STATE6S. through STATE4S. as an octal digit (000-111 or 0-7) and STATE3S. through STATE0S. as a hex digit (0000-1111 or 0-F). For example, the idle state on card S6 is expressed as 7F (111 1111); on card R6 the idle state is expressed as the complement of 7F (000 0000).

Except for the idle state, which can be many clocks in duration, each state corresponds to one system clock period in a memory cycle. Thus, the number of clocks in any MBU cycle can be determined by examining the "bubbles" in Figures 3-5 and 3-6.

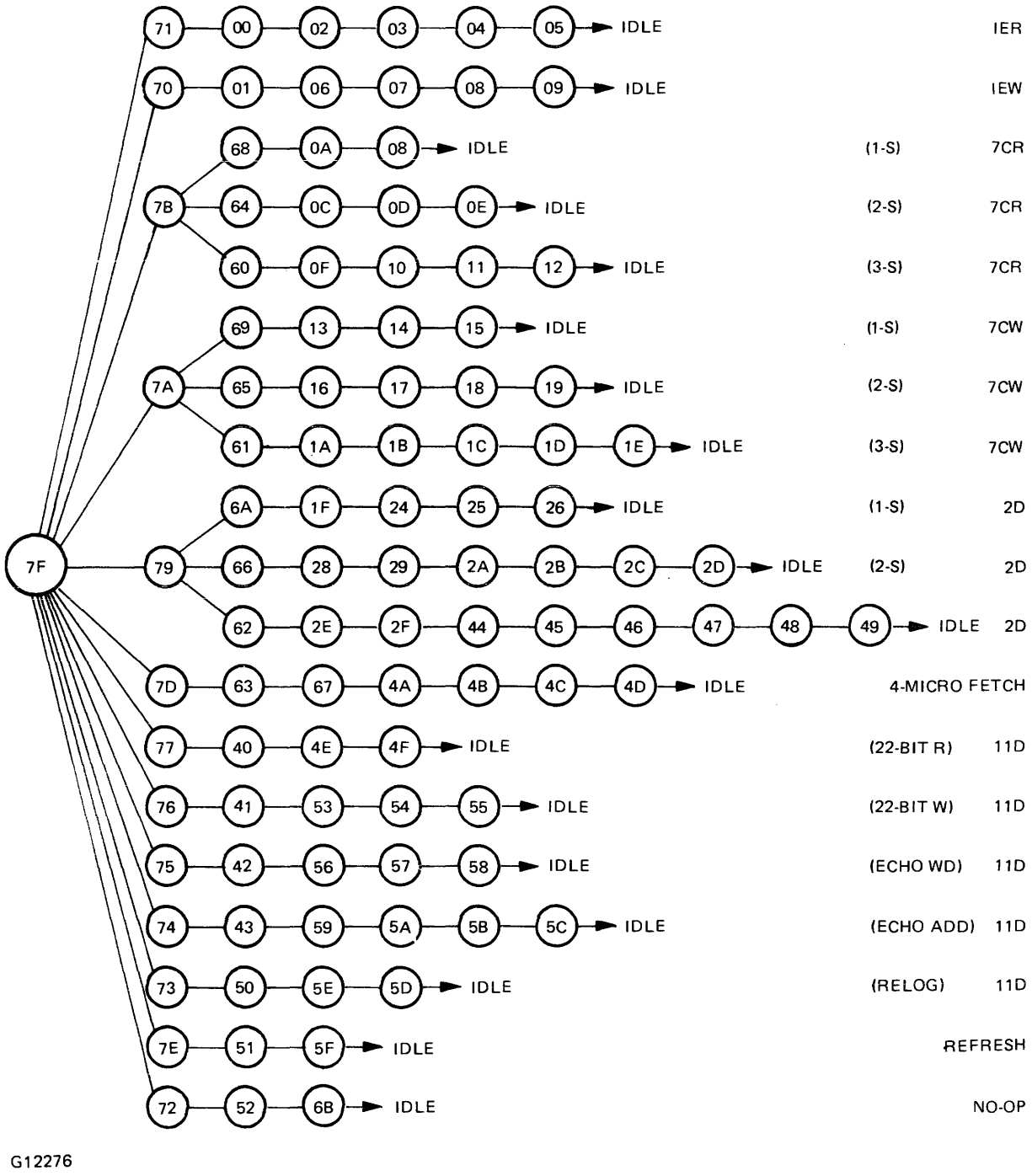
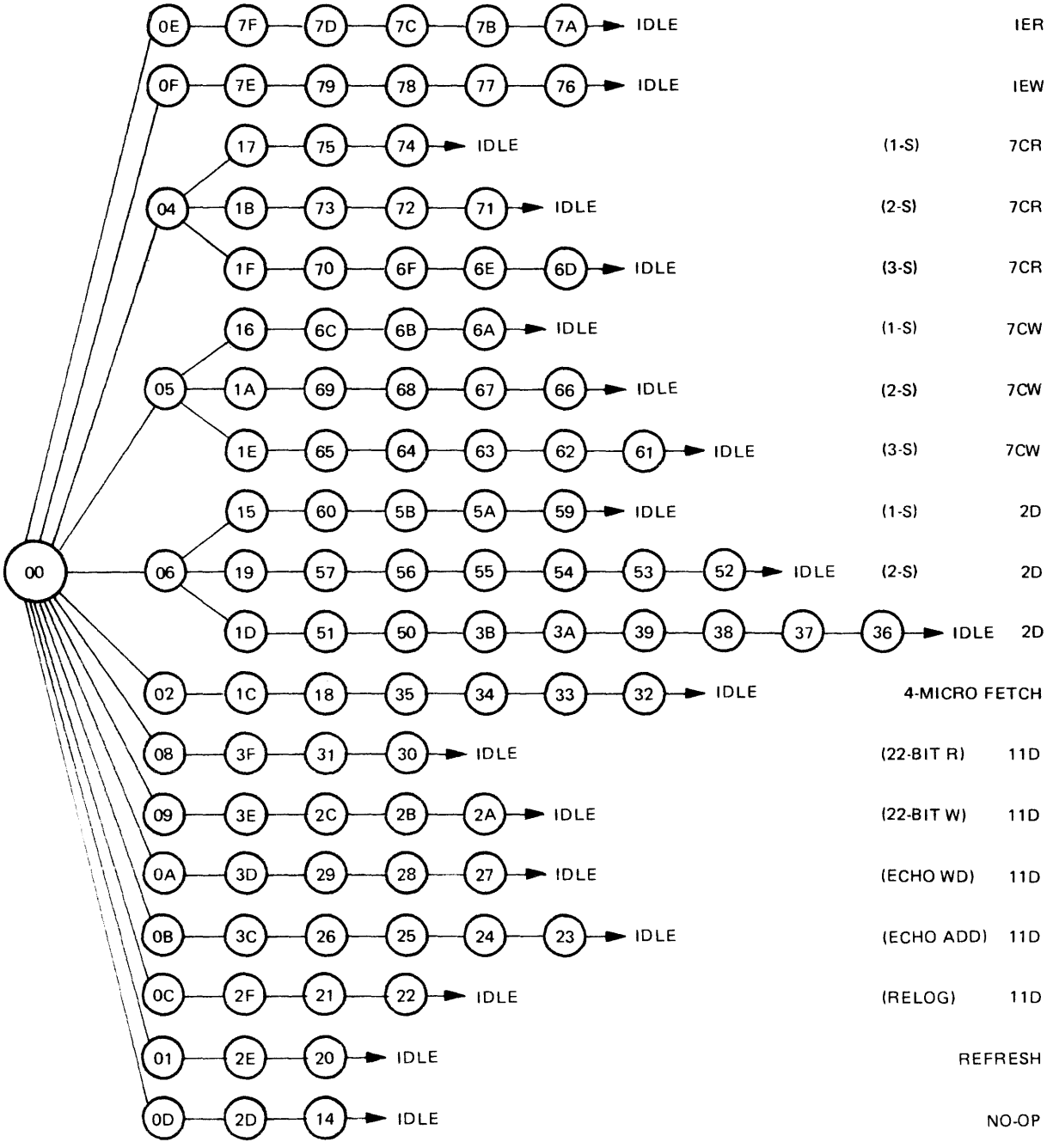


Figure 3-5. State Machine Flow, Card S6



G12277

Figure 3-6. State Machine Flow, Card R6

Figures 3-5 and 3-6 also illustrate the clock-period naming convention used in the MBU. The idle state is referred to as T0 and subsequent clock periods are designated T1, T2, T3, and so on.

For example, a 2-stack Read goes through a total of six states and therefore is six system clocks in duration. If the state lines were monitored during this Read, their contents would sequence as shown in Table 3-5.

Table 3-5. State Sequence for a 2-Stack Read

Card S6, STATEnS n=	6 5 4 3 2 1 0		Clock
	1 1 1 1 1 1 1	= 7F	T0 (Idle)
	1 1 1 1 0 1 1	= 7B	T1
	1 1 0 0 1 0 0	= 64	T2
	0 0 0 1 1 0 0	= 0C	T3
	0 0 0 1 1 0 1	= 0D	T4
	0 0 0 1 1 1 0	= 0E	T5

Card R6, STATEn/0 n=	6 5 4 3 2 1 0		Clock
	0 0 0 0 0 0 0	= 00	T0 (Idle)
	0 0 0 0 1 0 0	= 04	T1
	0 0 1 1 0 1 1	= 1B	T2
	1 1 1 0 0 1 1	= 73	T3
	1 1 1 0 0 1 0	= 72	T4
	1 1 1 0 0 0 1	= 71	T5

The type of cycle that the MBU is to perform is determined by wire-ANDing the mode bits (refer to Table 3-4) into the Next-State lines (NSTAT0S.through NSTAT6S.) on card S6. During the idle state (7F), the Next-State PROMs also generate a value of 7F for the next state, which is still an idle state. This means that if no request is received, the state machine idles by looping on state 7F.

When a defined field Read is initiated, the mode bits (M3/ M2/ M1/ M0/) equal 1011 (hex B) which is wire-ANDed with the 1111 (hex F) on NSTAT3S., NSTAT2S., NSTAT1S., and NSTAT0S. to produce 1011 in the lowest four bit positions of Next State. Thus, at the beginning of the second clock period, the state machine advances to state 7B, as shown in Figure 3-5.

If the operation is not a defined field Read, defined field Write, or defined field Swap, the Next-State PROMs control the state sequence starting with the second clock period. If the operation is one of those three, the Fire PROM is used during the second clock period to determine the number of stacks (1, 2, or 3) required by the operation.

For example, in the second clock period of a defined field Read, the state value is 7B. With an input of 7B, the next-state PROMs output is 110 1100 (hex 6C).

The outputs of the Fire PROM are wire-ANDed into the NSTAT3S. and NSTAT2S. lines. After examining the field direction sign, starting bit, and field length, the Fire PROM emits 10 (NSTAT3S.=1, NSTAT2S.=0) for a 1-stack operation, 01 for a 2-stack operation, or 00 for a 3-stack operation.

Thus, for a 2-stack Read, the 110 1100 (hex 6C) from the Next-State PROMs is modified by the 01 (on NSTAT3S. and NSTAT2S.) from the Fire PROM to produce 110 0100 (hex 64) on next state. Therefore, at the beginning of the third clock period of the cycle, the state machine advances to state 64, which uniquely identifies a 2-stack Read. From that point, the Next-State PROMs control the state sequence for the remainder of the cycle.

If memory accesses are required, the access of the first stack starts during T1. This is accomplished by issuing MCL1 to the appropriate stack. There are two MCL1 signals on the MBU backplane: MCL1.A.0 and MCL1.B.0. When active, MLC1.A.0 allows accesses to stacks 0 and 1 and MCL1.B.0 allows accesses to stacks 2 and 3. Therefore, it is possible that a 2-stack access can involve both MCL1's. The timing diagrams in Figures 3-7 and 3-8 illustrate the differences in timing for the two MCL1's, according to the number of stacks involved, the starting stack, and the forward or reverse access direction.

Figure 3-7 illustrates the timing for a 2-stack forward Swap and Figure 3-8 illustrates the timing for a 2-stack reverse Swap. A Swap operation involves both Read and Write timing.

Tables 3-6 and 3-7 denote signal descriptions as they appear on the cable.

Read data is returned to card R6 only after a DRSL (Data Read Select) has been issued to the storage boards by the S6 card. There is a separate DRSL for each of the four stacks. The four DRSLs (DRSL0..0, DRSL1..0, DRSL2..0 and DRSL3..0) are mutually exclusive; that is, only one can be active at a given time. During the time DRSL is active, the data from the selected stack is processed and stored in the appropriate register. On a Read operation the data is corrected, rotated, masked, and stored in the Read Data Accumulator (RDA). On a Write operation, the Read data is corrected, merged with the Write data according to the mask, and stored in the Write Data Register (WDR). A Write Enable (WENL) is then issued to the same stack that was just read. The MBU has separate Read data and Write data busses so data can be read and processed from one stack at the same time that different data is being written to another stack.

The calculations to determine the mask and rotation values are performed in the clock period preceding the one in which they are actually used.

On a Read operation, the final clock of the cycle is used to return the Read data to the requestor through the DATA frontplane cable. The MCL1s are returned to the inactive state during this period. On a write, the MCL1s are returned to the inactive state after the last stack has been written into.

Table 3-6. Data Cable Signal Description

Interface Signal	Time Period	Description
DATA23W1 thru DATA00W1	T0	DATA23-06: These 18 bits are mapped into address bits 17-0 respectively on the A and B address busses to the storage boards. DATA05-04: These two bits are the stack pointer bits that point at one of four memory stacks. DATA03-00: These four bits are the bit pointer field of the starting address.
	T2	DATA23-00: These are Write data bits
	T3 and subsequent clocks	DATA23-00: These 24 bits are the data being sent from the MBU to the processor.

Table 3-7. Control Cable Signal Descriptions

Interface Signal	Time Period	Description
MB.0A/1 MB.1.B/1 MB2.C/1 MB.3.D/1 MB.4.E/1	T0	These five bits represent the field length for defined field operations (7C Read, 7C Write, and 2D Swap.)
MB.5.F/1	T0	Field Direction Sign (FDS).
MD.0.G/1	T0	LSB of the the four mode lines.
MD.1.H/1	T0	Next LSB of the four mode lines.
	T2	NOWRITE bit; disables all Write enables to storage boards for the remainder of the current cycle
	T3 and subsequent clocks	This is MPCYCLE bit that, when high, indicates to the MBU that the processor requested the current cycle.
MD.2.I/1	T0	Second MSB of the four mode lines.
	T4 and subsequent clocks	AOUT/ – When low, indicates that the MBU has detected an address out-of-bounds.
MD3.J/1	T0, T4 and subsequent clocks	MSB of the four mode lines. UERR/; When low, indicates that an uncorrectable error has been detected.
MRQ.GK/1	All clock periods	Global memory request; signals MBU that an external request is coming during the present clock period.
MEND.L.1	All clock periods	Memory end; TRUE (high) for one clock period during each cycle. Always precedes Memory Ready by two clocks.
EMCK.M.1	All clock periods	Early memory clock; precedes by two
RLOG.N/1	All clock periods	Read Error Log; indicates to Processor that ELOG contents have changed.
CLRB.P/1	All clock periods	A LOW on this line clears the MBU. The LOW is present for at least 3 clocks.
EN.A.Q.1	All clock periods	A HIGH on this line allows the processor to drive the interface cables.
MRDY.S.1	All clock periods	Memory ready is LOW during meory cycles.
RTC..U/1	All clock periods	Real Time Clock is HIGH for one clock period every 0.1 seconds.
PORT.V/1	All clock periods	LOW indicates that HA-3 is present. (A level; no timing is involved)

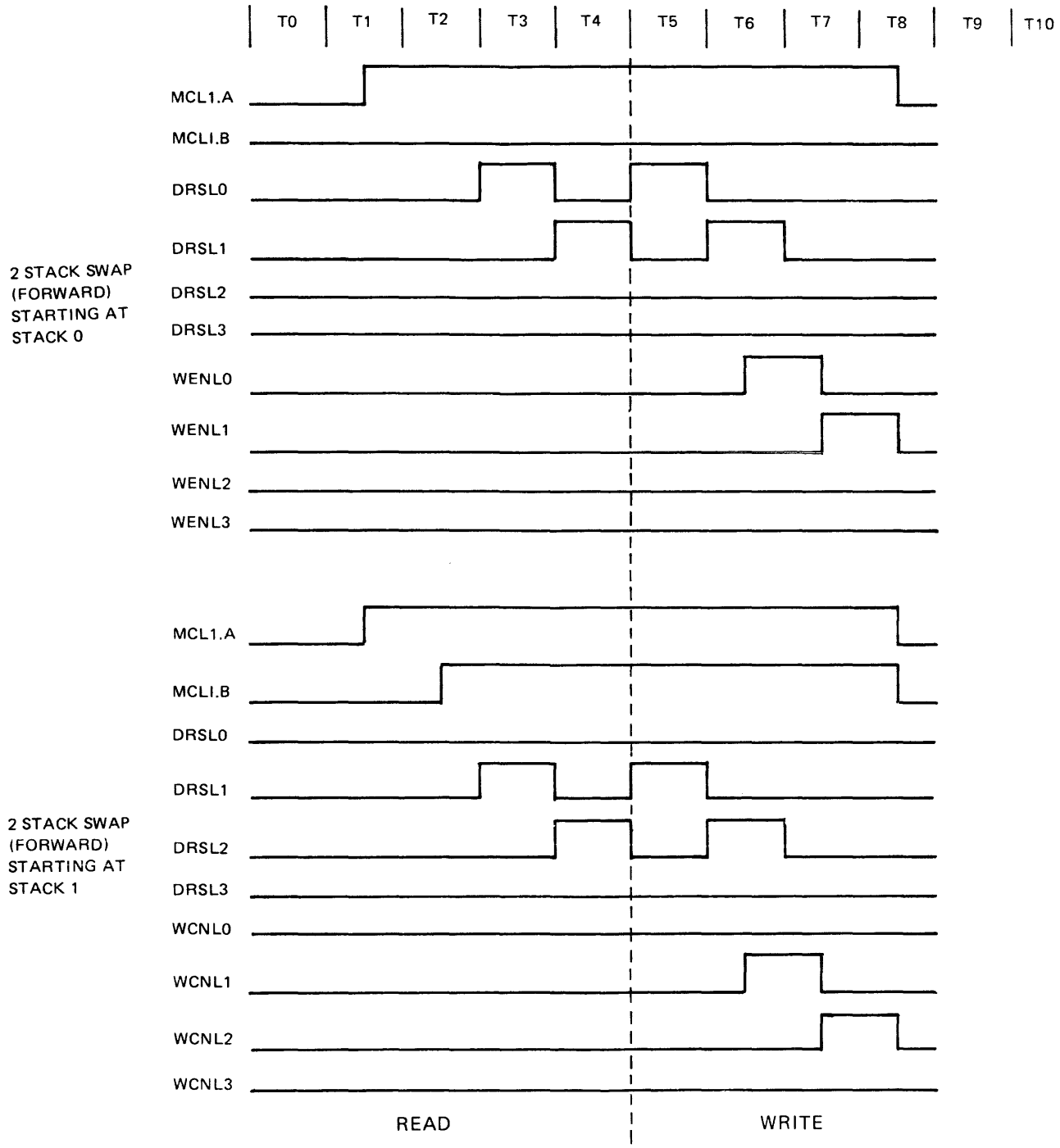


Figure 3-7. Swap Timing (2-Stack Forward)

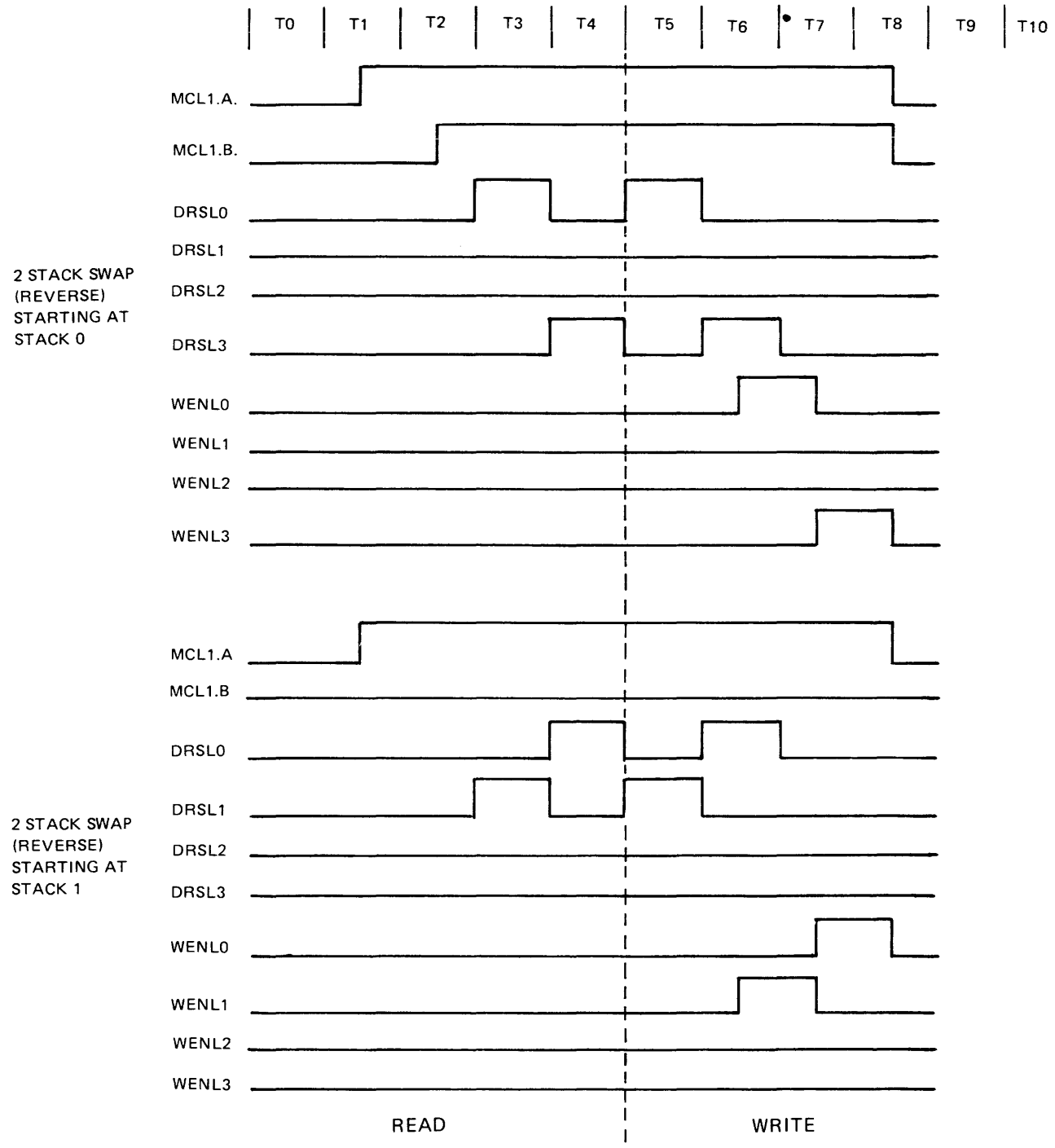


Figure 3-8. Swap Timing (2-Stack Reverse)

Address Loading and Modification

Two address registers are located on card S6: Address Register A, which holds the A address, and Address Register B, which holds the B address. The outputs of the registers (ADDRA000-170 and ADDRB000-170) connect directly to the address busses on the storage boards. Address Register A addresses stacks 0 and 1, and Address Register B addresses stacks 2 and 3.

During the request clock period (T0) of a non-Refresh memory cycle, the address sent to the MBU is presented to the inputs of both address registers and is clocked into both at the end of T0. Thus, during T1, all four stacks in the memory receive the same address.

Also, at the end of T0, the address is clocked into the Address and Write Data Register (AWDR). During T1, the address in the AWDR is routed through the address modifier and a decision is made to update either the A Address Register or the B Address Register. The update, if necessary, occurs at the end of T1.

If the request is a Refresh request, the address contained in the Refresh Address Register is presented to both the A Address Register and the B Address Register during T0. At the end of T0, the Refresh address is clocked onto the A and B address busses where it remains constant throughout the Refresh cycle; that is, no modification is made to the contents of either Address Register.

Address Register Characteristics

The A Address register and the B Address registers are each 18 bits wide. During T0 of a non-Refresh cycle, the 18 bits presented to these registers represent the 18 most significant bits of the 24-bit address received by the MBU. The least significant four bits of this address comprise the bit pointer; the remaining two bits are the stack pointer. These six bits are not sent to the storage boards.

During T1 of a non-Refresh cycle, the 18 bits presented to the A and B Address Registers are the most significant 18 bits of the 24-bit address received by the MBU as modified by the address modifier.

The Refresh Address Register is an 8-bit counter that contains the address of the next internal storage chip row to be refreshed. With 16K RAM devices, only the lower seven bits of the address are used for the Refresh. All storage boards in the MBU are refreshed simultaneously.

Address Modification

The Address Modifier is an 18-bit adder/subtractor that is used during T1 to modify the 18 most significant bits of the address stored in the Address and Write Data register (AWDR). Field direction sign (FDS) = 0 signifies a forward access, which is positive and increments the value. FDS = 1 signifies a reverse access, which is negative and decrements the value. Thus, the Address Modifier provides the address of the next higher or lower word within the four-stack group.

If the entire 24-bit address stored in the AWDR is denoted A, an increment of the 18 most-significant bits of A actually corresponds to adding 64 to A, and a decrement of these bits corresponds to subtracting 64 from A.

Table 3-8 illustrates two cases: 1) a forward access starting in stack 2, and 2) a reverse access starting in stack 0.

Table 3-8. Address Modification

	Stack 0 (16 bits)	Stack 1 (16 bits)	Stack 2 (16 bits)	Stack 3 (16 bits)
ADDR A+64	→			
ADDR A	←		→	→
ADDR A-64				←

→ Forward Access
← Reverse Access

The outputs of the Address Modifier are gated through tri-state Octal Buffers (BXSNs) to the inputs of the A Address and B Address registers. While the address is being modified, a decision is made by a PROM on card S6 to update either the A Address or B Address register or to update neither by clock enables (ENACLK.. and ENBCLK..).

The A Address register is loaded with the modified address at the end of T1 if ALL of the following conditions are TRUE:

1. The current operation is a defined field operation, that is, a Read/Write Memory (7C), a Swap (2D), or a Diagnostic Read/Write Memory (11D) with echo modified address specified by the variant.
2. The first stack to be accessed is either stack 2 or stack 3. (In the address received by the MBU, this is specified by bit 5 = 1.)
3. FDS = 0, indicating a forward access.

The B Address register is loaded with the modified address at the end of T1 if all the following conditions are TRUE:

1. The current operation is a defined field operation, that is, a Read/Write Memory (7C), a Swap (2D), or a Diagnostic Read/Write Memory (11D) with echo modified address specified by the variant.
2. The first stack to be accessed is either stack 0 or stack 1. (In the address received by the MBU, this is specified by bit 5 = 0.) received by the MBU being equal to zero.
3. FDS = 1, indicating a reverse access.

At the time of loading, the state machine is not aware of the number stacks to be accessed; it is only aware of a condition that possibly requires address modification. Hence, a modified address may be loaded into the A or B Address register but not used in an access. An example of this is a 2-stack forward access beginning in stack 2. Even though the only stacks accessed are stacks 2 and 3, the A Address register is loaded, at the end of T1, with the incremented address.

Address Out-of-Bounds

When Data Read Selects (DRSLs) are issued to the storage boards by card S6, a comparison is done on card S6 between the address being used for the access and the address that corresponds to the amount of physical memory present in the MBU. The memory size is stored in a jumper chip called MAXS on card S6. This jumper can be set for a minimum of 64K bytes and a maximum of 2048K bytes (2MB) bytes in 64K-byte increments. If the address being used for the access is outside the bounds of the physical memory present in the MBU, an Address Out Of Bounds Error (AOB) is reported.

The use of DRSLs in the address comparison ensures that an out-of-bounds address only causes an error to be reported if it is used for a data access. This is important because a modified address that is out of bounds can be loaded into the A or B Address Registers; but if it is not used in an access, AOB must not be reported.

Read Data

Memory Read data enters the MBU data path at card R6. From the storage board there is a 22-bit information path (16 data; 6 ECC) to the R6 card. The read data enters immediately into the Read Data Corrector. If the current operation is a Read, the corrected Read data passes through the Read Data Rotator and the Masker, and into the Read Data Accumulator.

If the current operation is a Write, the corrected Read data goes to the Write Data Merger where it is selectively merged with the Write data to form a new 16-bit word to be written to memory.

Read Data Selection

Read data is accessed from the storage board through the DRSL signal approximately 300 nanoseconds after the MCL1 to that stack. This means that the first Read data appears at card R6 during T3.

The Read data from the first stack accessed is processed and arrives at either the Read Data Accumulator or the Write Data Register before the end of T3. Data read from other stacks in subsequent clock periods are processed in the same manner.

Syndrome Generation

The syndrome is a code that indicates whether there are errors in the Read data. In the case of a single-bit error, the syndrome is encoded such that the exact bit in error can be determined. The syndrome is generated on card R6 using Parity Generators (PGSNs). (The syndrome code is given in Table 3-3 of the earlier subsection titled Error Correction Code.)

A single 1 bit with five 0 bits in the syndrome indicates a single-bit error within the six ECC bits. If there is a single-bit error among the 16 data bits, a syndrome with three ones and three zeros is generated. A syndrome of all zeros indicates no errors in any of the 22 bits.

A syndrome that does not correspond to the pattern for no errors or single-bit error indicates a multiple-bit error in the Read data.

Syndrome Decode and Error Correction

In order to determine whether there are any errors in the Read data, the syndrome must be decoded. The syndrome is decoded in two different logic sections of card R6 as follows:

1. Partial Syndrome Decode logic, used in the error correction logic. This is called partial because, in the interest of speed, not all of the 64 different syndrome patterns are decoded. The decode circuitry is implemented with single-level decoder chips (DC3Ss). A syndrome for a multiple-bit error presented to the Partial Syndrome Decoder might look like a single bit error and cause a bit to be complemented. This is of no consequence because the data already contains multiple errors.
2. Full Syndrome Decode logic, implemented with a PROM and used to update the ELOG register. This logic is also used to control inversion of the ECC when a multiple-bit error is detected during a Write operation.

The Partial Syndrome Decoder can be disabled so that, regardless of the value of the syndrome, no correction is made to the Read data. At the same time that it is disabled; however, the Full Syndrome Decoder can still check for errors, which permits error detection without error correction, used in the 22-bit Diagnostic Read Memory (11D) operation.

Error Correction

Error correction is accomplished on card R6. There are 16 active-low signals (CORR00R.-CORR15R.) generated by the partial syndrome decoder that feed the error correction logic along with the 16 Read data bits. Only errors in the 16 data bits are corrected; that is, if a single-bit error is detected in one of the ECC bits, no attempt is made to correct it. When one of the 16 lines from the partial syndrome decoder goes low, it indicates that that bit is in error and must be corrected.

Read data is received from the storage boards by inverting tri-state devices (BXSNs) on card R6. The Read data is presented to the error correction logic in its inverted state. The output of the error correction logic, however, is the corrected Read data in its positive (non-inverted) state. If the Read data bit is not in error, it must be inverted back to its positive state as it passes through the error correction logic. If the Read data bit is in error, it is left in its inverted state as it passes through the correction logic and is thus corrected.

The correction logic is implemented with one exclusive-OR (XOR) gate per bit. The correction mechanism is shown in Table 3-9.

Concerning error correction, the signals CORRnnR. are the active-LOW signals from the Partial Syndrome Decoder, the M.RDnnR. signals are the inverted data from the storage boards, and the C.RDnnR. signals are the output of the correction logic.

Table 3-9. Error Correction Truth Table

CORRnnR.*	M.RDnnR.*	C.RDnnR.	COMMENT
0	0	0	Data incorrect, do not invert
0	1	1	Data incorrect, do not invert
1	0	1	Data correct, invert
1	1	0	Data correct, invert

* Active Low Signals

Read Data Rotation

Read data must be rotated in the MBU to support bit addressability. For example, if the processor requests the memory to perform a single-bit read, the single-bit could be in any of the 16 data bit positions in the memory word. It is the function of the MBU to rotate and mask the data so that the single bit requested by the processor is right justified on the DATA interface and that the 23 most significant bits are zeroes.

All Read data being returned to a user passes through the Read Data Rotator, a 16-input, 16-output right rotator composed of two levels of 4-bit shifters (SH4Rs) and capable of a right rotate of from 0 to 15 positions. The Read Data Rotator is located on card R6 while the control logic for generating the rotate amount (ROT.01S0-ROT.16S0) is located on card S6.

Read Data Masking

After the Read data has passed through the rotator, the data passes through the Read Data Masker. The Read Data Masker consists of eight dual 4-to-1 multiplexors (S4SNs) and supports bit addressability by allowing only the bits relevant to the request to be loaded into the RDA

The masking is done on a bit-by-bit basis by conditioning a mask line (MASK00R.-MASK23R.). A zero on a mask line causes the bit presently stored in the corresponding RDA location to be reloaded into the same RDA bit location. A one on a mask line causes the corresponding RDA bit to be loaded with corrected Read data.

Read Data Accumulator

The Read Data Accumulator is a 24-bit register that holds Read data that is to be sent to the user. The RDA consists of four Octal-D flip-flops (RESNs), three of which are loaded with data from the Read Data Masker (RDA.00..-RDA.15..) while the fourth is loaded with the ECC bits (DR16..X0-DR21..X0) received by card R6 and the parity bit (PARITY..) generated for the Read data.

The RESN containing the 8 most significant bits from the Read Data Masker, and the RESN containing ECC and parity have their outputs connected (DATA16.. to DATA23..). Only one of the two is enabled at a time. For a 4-micro fetch or a 22-bit diagnostic Read, the RESN with ECC and parity is enabled. For all other Read operations, the RESN with data from the Read Data Masker is enabled.

Data Accumulation

The RDA is called an accumulator because, in order to structure the Read data into the format that the user is expecting, it is necessary to accumulate data from either one, two or three stacks in memory.

Before beginning to accumulate data for a Read cycle, the RDA must be cleared. This is accomplished by PROM generation of the signals ENRACC/. and DFREAD/. that clear both the RDA and the Read Data Masker that sources the RDA and thereby loading the RDA with all zeroes.

Starting with a cleared RDA is necessary in order to ensure that zeros are loaded into the most significant bits of the RDA when the Read data field width is less than 24 bits.

For example, if a request for a Read of 18 bits is received by the MBU, the using device expects the most significant six bits of the 24-bit field returned to be filled with zeroes. The most significant six bits of the mask would be zero during the accumulation of data from one, two, or three stacks. A zero on the mask line selects the corresponding RDA bit to be reloaded into the same location and, if the RDA does not contain zeros in those locations, the locations contain unknown data at the end of the cycle.

Read Data Parity

Card R6 has parity generation logic that consists of two parity generators (PGSNs) to generate odd parity (PARITY..) on the 16 data bits emerging from the Read Data Corrector. The parity bit (DATA16..) is appended to the Read data during a 4-micro fetch operation or during a 22-bit diagnostic read. The parity bit is always odd over the 16 data bits sent to the processor. (The ECC bits sent during a 22-bit diagnostic Read are not involved in the parity generation.)

Write Data

Write data is received from the user during the third clock (T2) of the memory cycle and is loaded into the Address and Write Data Register (AWDR) on card S6.

The Write data (A.WD00S1 to A.WD23S1) is rotated, merged with corrected Read data, presented to the ECC generator, and loaded into the Write Data Register on card R6 along with the ECC.

Write Data Rotation

Write data must be rotated in order to support bit addressability. For example, if a request is received by the MBU to perform a single-bit write, the single data bit to be written is right-justified when it is loaded into the AWDR from the DATA interface. However, the single bit may be required to be stored in any one of the 16 data bits in the memory word. Therefore, it must be rotated in order to line up with the correct memory bit position and it must be merged into that location so that none of the other 15 bits in the memory word are changed.

Write Data Rotator

The Write Data Rotator is a 24-input, 16-output left rotator that consists of two levels of 4-bit shifters (SH4Rs). The first level of rotation rotates Write data by amounts of zero to three. The rotation amounts are determined by the select line inputs (ROT.01R. and ROT.02R.) to the six 4-Bit Shifters whose outputs (I.WD00R.-I.WD23R.) serve as inputs to the dual set second-level rotator.

The second level of rotation rotates data by two sets of amounts. The amounts of rotation are determined by the enable terms ROT16.R. and ROT16/R. as they are applied to their half of the dual-set second-level rotator.

ROT16.R.

This term permits rotation amounts between 16 and 24. It is an enable for the lower half of the dual set and allows rotation amounts as determined by the select line inputs (ROT..4R. and ROT..8R.). These rotation values, coupled with the first-level rotation values, determine rotation amounts between zero and seven. ROT.16R. supplies a fixed rotation amount of 16.

The hard-wired placement of the 24 outputs of the first-level rotator automatically aligns the outputs of the of the second-level rotator with reference to memory. (Refer to Figures 3-9 and 3-10.) Therefore, if a rotation amount of 16 is requested, bit 0 of user Write data at the input to the Write Rotator would become bit 15 at the output of the Write Rotator. This is an effective realignment of data referencing.

ROT16/R.

This term permits rotation amounts between 0 and 15. It is an enable for the upper-half of the dual set and allows rotation amounts as determined by the select line inputs (ROT..4R. and ROT..8R.). These rotation amounts are the same as those for the lower-half of the dual set and produce a rotation amount between 0 and 15, considering the first-level of rotation.

The outputs of the Write Data Rotator are tied to the outputs of the Read Data Rotator. SH4Rs are tri-state elements and only one set is enabled at a time. The outputs are tied together to support the Diagnostic Echo Write Data micro that transfers data through the Write Data Rotator and loads it into the Read Data Accumulator to be sent back to the processor.

As with the Read Data Rotator, the Write Data Rotator is located on card R6. The control logic to generate the rotate amount is the same as that used for the Read Data Rotator and is located on card S6.

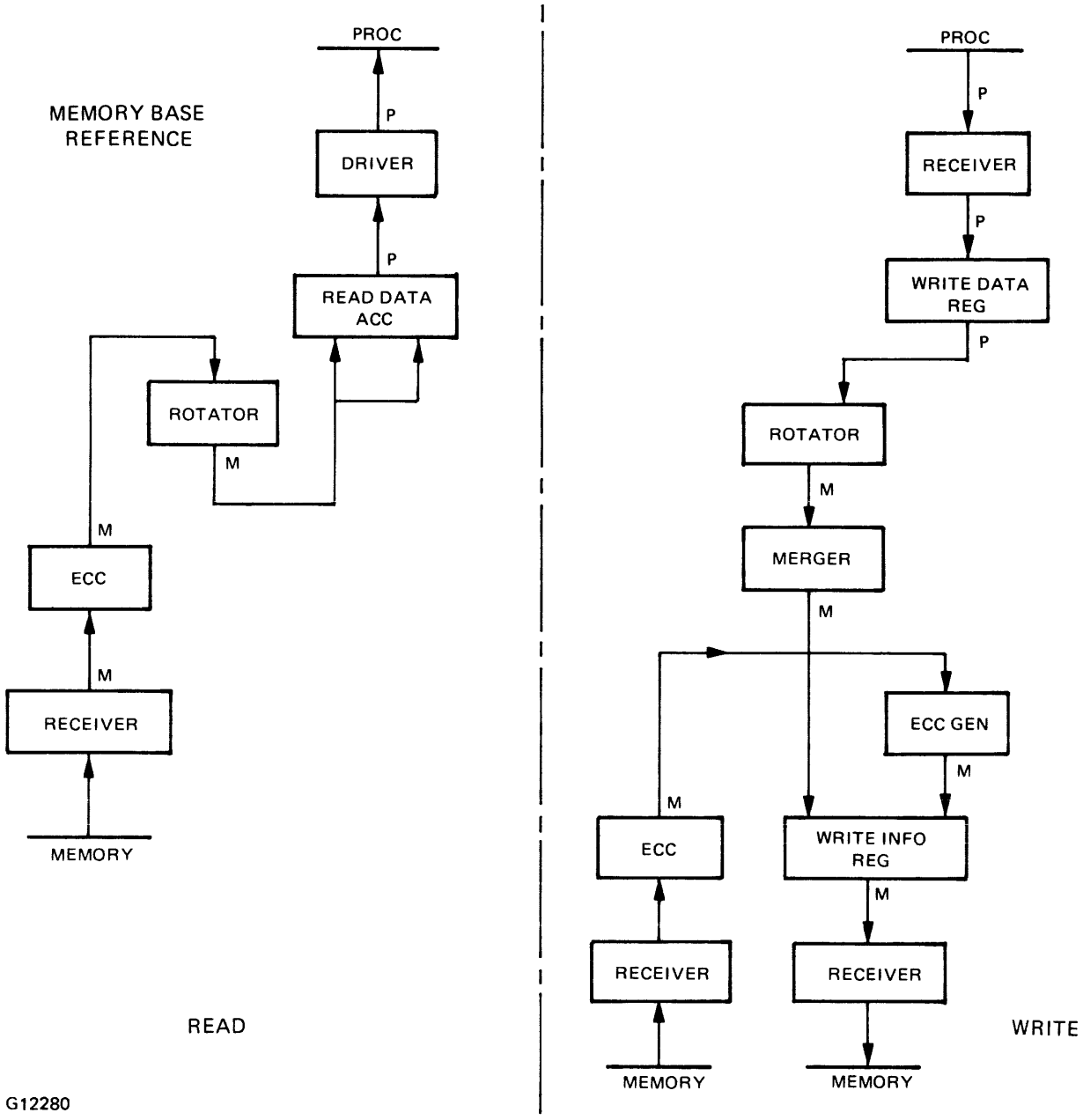


Figure 3-9. Memory/Processor Referencing

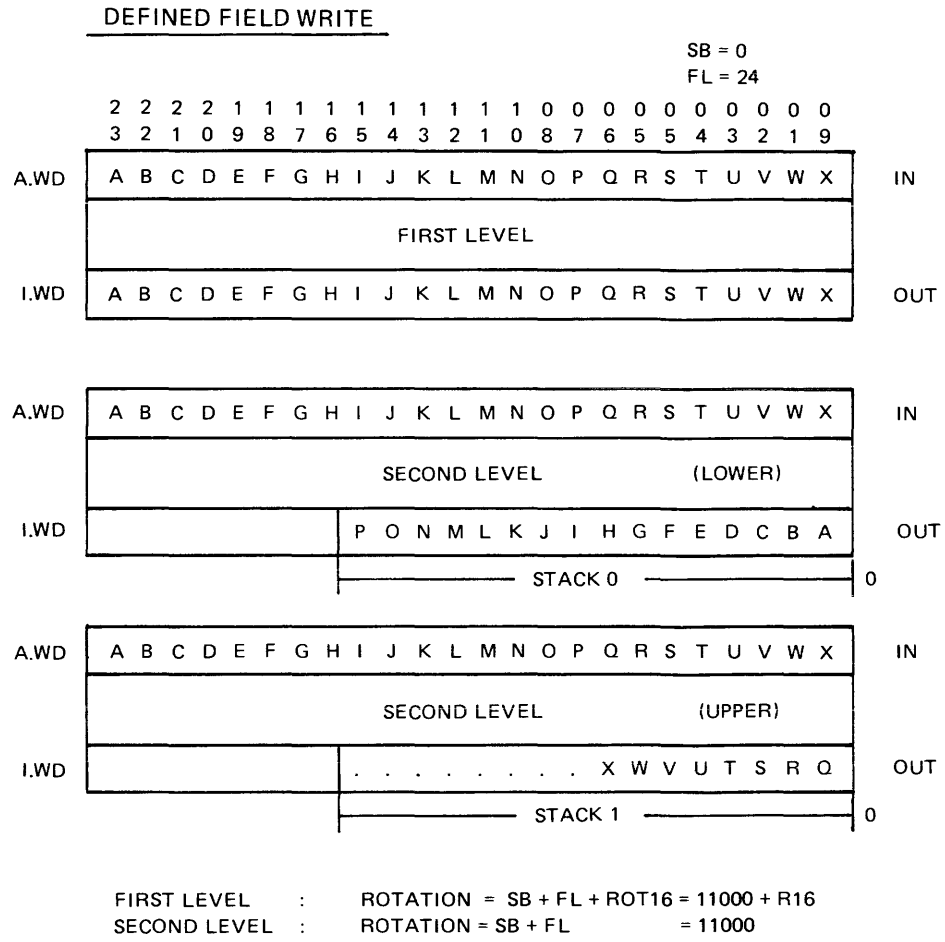


Figure 3-10. Write Rotator Function

Write Data Merger

The Write Data Merger correctly positions the Write data with respect to the memory word and ensures that unaffected data in the memory word is not affected by the new data that is being written.

The Write Data Merger consists of eight dual 4-to-1 multiplexers (S4SNs) and operates very similarly to the Read Data Masker. During a Write operation, the memory word that is to be written into is first read and routed through the Read data correction logic to the Write Data Merger. Concurrently, the Write data is rotated and sent to the Write Data Merger.

Merging is done on a bit-by-bit basis under the control of the mask lines (MASK00R.-MASK15R.). A zero on a mask line selects the corresponding corrected Read data bit (C.RD00R.-C.RD15R.) while a one selects the rotated Write data bit (ROTD00R. to ROTD15R.) as outputs (M.WD00R. to M.WD15R.).

ECC Generation

The Error Correction Code (ECC) used to support the single-error correction, multiple-error detection scheme for the MBU is generated from the data present on the output of the Write Data Merger. The ECC is generated using six parity generators (PGSNs). The ECC is a function of the data bits. (See Table 3-1.)

Two special cases in the generation of the ECC bits are considered next.

1. Detection of an error in the Read data during a Write operation.

   If the error is a single bit error, the bit in error is corrected before it is merged with Write data, new ECC is generated for the word and the word is written into memory error-free. However, if the error is a multiple bit error, the MBU does not know what bits are in error and cannot correct them.

   It is possible that one or more of the bits in error are not replaced by new Write data in the Write Data Merger and thus the new word built by the merger and sent to the ECC generator could contain one or more errors. The ECC generator itself does not recognize errors, but merely calculates ECC on the data provided to it.

   If the normal sequence of events is followed, the ECC generator takes the data that contains an error, generates the correct ECC for the data, and writes the new word into memory. The next time that word is read, there is no error indication because the ECC is correct for the data. Indication of possible errors is lost.

   The MBU contains a mechanism to prevent this situation. One of the outputs of the Full Syndrome Decode PROM (INVECC/.) inverts the ECC. The effect of this signal, which is active when the syndrome indicates a multiple bit error, is to invert the ECC that has been generated for the new memory word. This inverted ECC is then written into memory with the data. The next time that word is read from memory, the inverted ECC, even though correct, ensures detection of another multiple bit error.

   Inversion of the ECC when a multiple error is detected in the Read data is suppressed under two conditions:

   The first condition is a full-stack Write request. In this case, all 16 data bits of the memory word are rewritten with new information. Errors in the Read data does not matter since all the data previously stored in that word is replaced.

   The second condition is a request for a defined field Write of 25. In this case, the MBU performs a 24-bit Write operation but ignores any errors in Read data and only writes good ECC.

   In both cases, the error checking in the Full Syndrome Decode PROM is turned off and errors are neither detected nor reported.

2. A defined field Write of 26.

   In this case, the MBU performs a 24-bit Write operation but always inverts the ECC before it is written into memory, regardless of the state of the syndrome.

Write Data Register

The Write Data Register (WDR), located on card R6, is loaded with the data to be written into memory. The outputs of the WDR (DI00...0 to DI15...0) directly feed the Write Data Bus to the storage boards. The WDR consists of four octal-D flip-flops (RESNs). Two of the RESNs contain the 16 data bits (DI00...0 to DI15...0) to be written into memory, the third RESN holds the ECC generated by the ECC generator (ECC.1... to ECC.6...), and the fourth RESN holds the ECC that is loaded directly from the AWDR on card S6 (A.WD18S1 to A.WD23S1).

The third and fourth RESNs have their outputs connected but only one is enabled at a time. The third RESN, containing the ECC from the ECC generator, is enabled for all Write operations except the 22-bit Diagnostic Write.

Rotation Control

The Rotation Control logic, located on card S6, consists of a Quad 2-to-1 Data Selector (S2-N) and a 4-Bit Binary Adder (AF4N) which interpret starting bit and field length information and generate a 5-bit rotation amount (ROT.01S0 to ROT.16S0) that is used by both the Read and Write rotators on card R6.

The number of positions by which Read or Write data must be rotated depends on four factors: 1) the type of operation, 2) the forward or reverse direction of the access, 3) the bit at which the access is to start, and 4) the length of the access.

The equations used to generate the rotate amounts are given in Table 3-10.

Table 3-10. Rotation and Masking Equations

Operation	Number of Stacks	Stack	Mask Start	Mask End	Rotate Count
Defined Field Read Forward Direction	One	1st	0	FL	SB+FL
	Two	1st	0	FL	SB+FL
		2nd	0	16+SB+FL	SB+FL
	Three	1st	0	FL	SB+FL
		2nd	0	16+SB+FL	SB+FL
		3rd	0	SB+FL	SB+FL
Defined Field Read Reverse Direction	One	1st	0	FL	SB
	Two	1st	0	FL	SB
		2nd	SB	FL	SB
	Three	1st	0	FL	SB
		2nd	SB	FL	SB
		3rd	M16+SB	FL	SB
Defined Field Write Forward Direction	One	1st	SB	SB+FL	R16+SB+FL
	Two	1st	SB	16	R16+SB+FL
		2nd	0	16+SB+FL	SB+FL
	Three	1st	SB	16	R16+SB+FL
		2nd	0	16	SB+FL
		3rd	0	SB+FL	R16+SB+FL

SB = Stack Boundary, FL = Field Length
R16 means that the signal ROT16.R. is active.

Table 3-10. Rotation and Masking Equations (Cont)

Operation	Number of Stacks	Stack	Mask Start	Mask End	Rotate Count
Defined Field Write Reverse Direction	One	1st	SB-FL	SB	R16+SB
	Two	1st	0	SB	R16+SB
		2nd	SB-FL	16	SB
	Three	1st	0	SB	R16+SB
		2nd	0	16	SB
		3rd	SB-FL	16	R16+SB
22-Bit Rd.	One	1st	0	16	0
22-Bit Wr.	One	1st	0	16	0
Echo Write Data	Two	1st	0	24	0
		2nd	M16	24	R16
Fetch	Four	All	0	16	0
Dispatch Read	Two	1st	0	24	8
		2nd	0	8	8
Dispatch Write	Two	1st	0	16	8
		2nd	0	8	24

SB = Stack Boundary, FL = Field Length
R16 means that the signal ROT16.R. is active.

In a Write operation, two additional control signals (ROT16.R. and ROT16/R.) are supplied by card R6. These signals are used to select one of the two groups of 4-Bit Shifters that comprise the second level of the Write Rotator. The appearance of the term R16 in an equation for rotate amount in Table 3-12 indicates that the signal ROT16.R. is active. If R16 does not appear in the equation, the signal ROT16/R. is active.

Masking

The 24-bit mask in the MBU is used during both Read and Write operations. During a read, the mask is used to determine the source of data to be loaded into the Read Data Accumulator. During a write, the mask is used to determine the source of data to be loaded into the Write Data Register.

Mask Generator

The Mask Generator is located on card R6. It consists of six PROMs (PR08s), three of which generate 24 bits of starting mask while the other three generate 24 bits of ending mask. The starting mask and ending mask are wire-ANDed together to form the 24-bit mask that is presented to the Read Data Masker and to the Write Data Merger. The addresses to the Starting and Ending Mask PROMs are supplied by the Mask Control on card S6. Tables 3-11 and 3-12 are truth tables for the masks.

Table 3-11. Starting Mask Truth Table

MCS					Mask															
					23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
16	8	4	2	1																								
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3-12. Ending Mask Truth Table

MCE	MASK			
	2 2 2 2 1 1 1 1	1 1 1 1 1 1 0 0	0 0 0 0 0 0 0 0
16 8 4 2 1	3 2 1 0 9 8 7 6	5 4 3 2 1 0 9 8	7 6 5 4 3 2 1 0
0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1
0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 1
0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1
0 0 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 1 1 1 1 1
0 0 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 1 1 1 1 1 1
0 0 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 1 1 1 1 1 1
0 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1
0 1 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1
0 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1
0 1 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 1 1 1	1 1 1 1 1 1 1 1
0 1 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1
0 1 1 0 1	0 0 0 0 0 0 0 0	0 0 0 1 1 1 1 1	1 1 1 1 1 1 1 1
0 1 1 1 0	0 0 0 0 0 0 0 0	0 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1
0 1 1 1 1	0 0 0 0 0 0 0 0	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 0 0 1	0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 0 1 0	0 0 0 0 0 0 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 0 1 1	0 0 0 0 0 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 1 0 0	0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 1 0 1	0 0 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 1 1 0	0 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 0 1 1 1	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 1 0 0 0	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
1 1 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

During a read, data is loaded into the Read Data Accumulator from the Read Data Rotator if the corresponding mask bit is a one and from the Read Data Accumulator (the same bit is reloaded) if the corresponding mask bit is a zero.

During a write, data is loaded into the Write Data Register from the Write Data Rotator if the corresponding mask bit is one and from the Read Data Corrector if the corresponding mask bit is a zero.

Mask Control

The Mask Control, located on cars S6, consists of a Quad 2-to-1 Data Selector (S2-N) and two 4-Bit Binary Adders (AF4Ns) that interpret starting bit and field length information and generate nine bits of masking information (MC.S1.S0 to MC.S8.S0 and MC.E1.S0 to MC.E8.S0 and MCE16.S0) that supply the values of the starting and ending locations of the mask. These locations depend on four factors: 1) the type of operation, 2) the forward or reverse direction of access, 3) the bit at which the access is to start, and 4) the length of the access.

The equations used to generate the starting and ending locations are given in Table 3-10.

In addition to the mask control supplied by card S6, the most significant bit of the starting mask location (MCS.16R.) is supplied by card R6. The appearance of the term M16 in Table 3-12 indicates that this bit is active.

Error Handling

The Error Log register consists of two octal D flip-flops (RESNs) and one octal tri-state inverter (IWSN) that are supported by three PROMS (for decoding error types) and one hex D flip-flop (for syndrome hold). This register resides on card R6 and supplies error information to the processor for a memory operation in a 24-bit format as illustrated in Figure 3-11.

The MBU detects two types of errors in memory data: correctable single-bit errors and uncorrectable multiple-bit errors.

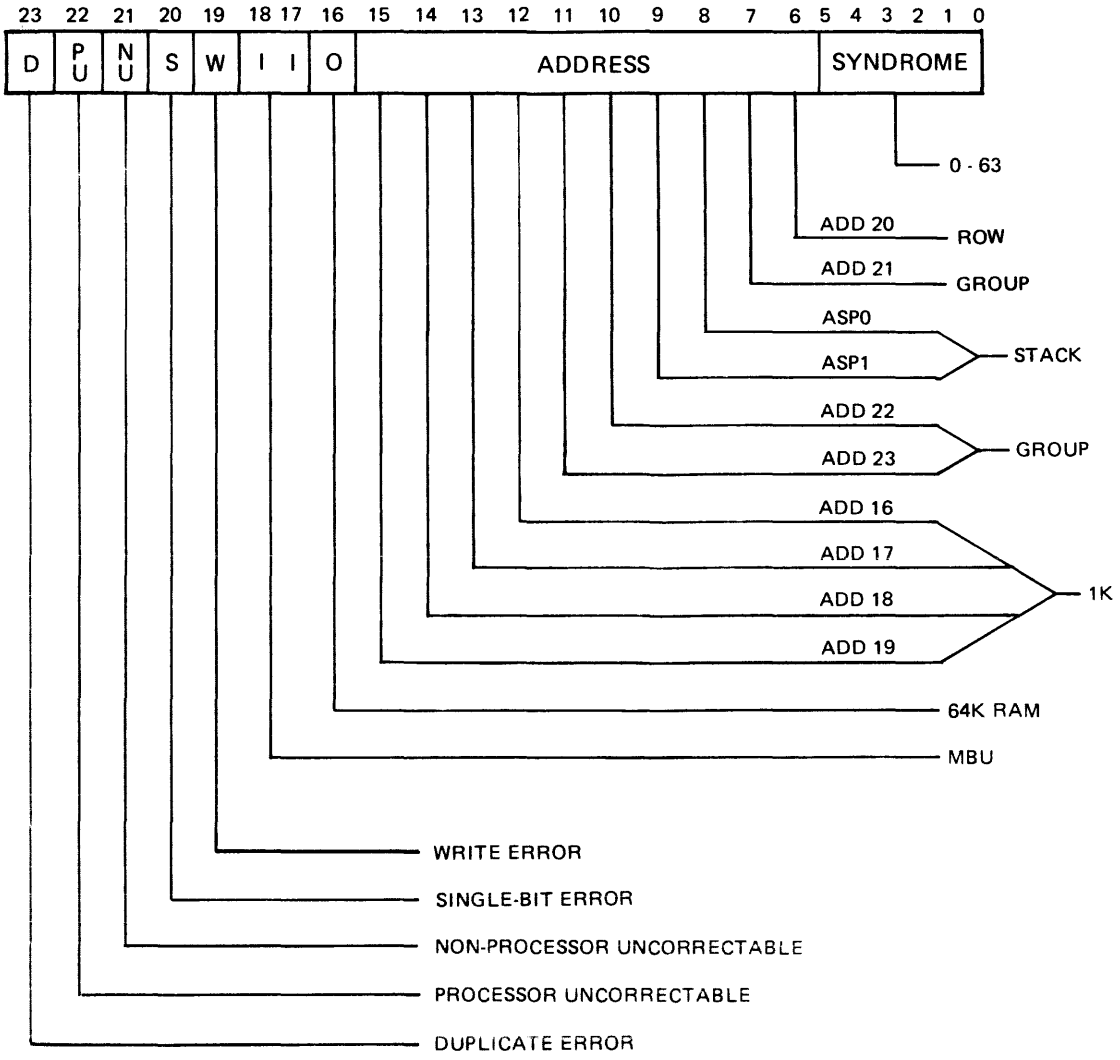
The Syndrome Generator detects errors by encoding the data bits and their associated ECC into a syndrome pattern which is decoded by the Syndrome Decoder PROM as either a single-bit error (SE.....) or an uncorrectable error (UERR....). These errors are clocked into an RESN and serve as inputs to two PROMs that determine error priority and signal that a change has occurred in the ELOG (RELOG...). The processor sets bit one (PERM1) of its memory error register.

Correctable errors are corrected by the Partial Syndrome Decoder and reported in the ELOG (ELOGSE/.) as bit 20 (DATA20..). These errors are reported and MBU operations continue.

Uncorrectable errors are reported and, in addition, the MBU determines whether they occurred during a processor-initiated operation (ELOGPU/.) or a non-processor-initiated operation (ELOGNU/.).

An uncorrectable error is reported to the processor via signal UERR1... that is decoded by the Syndrome Decoder. The processor sets bit zero (PERM0) of its memory error register.

Writing 16 bits of data across a stack causes the error detection logic to be disabled (CHECK...) and errors are not reported. When the field length is less than 16, normal error detection is implemented and the ECC is inverted as it is written into memory. This flags the memory word as one that may contain errors.



ASP = ACTIVE STACK POINTER

ASP0	ASP1	
0	0	= STACK 0
1	0	= STACK 1
0	1	= STACK 2
1	1	= STACK 3

G12282

Figure 3-11. Error Log Register

Address Out-of-Bounds

The MBU checks the address information for the operation and determines whether it is within the physical limits of the memory. The logic that determines memory physical limits, on card S6, signals the processor to read the ELOG by providing a signal (AOOB/...) that sets bit 2 (PERM2) of the processor's memory error register.

Error Log Information

Error Log (ELOG) contains the following information:

Bit 23 Duplicate Error (ELOGD/..)
Indicates that at least two errors of the same type were detected since the ELOG was last cleared. The address and syndrome information resident in the ELOG is for the first error.

Bit Processor Uncorrectable Error (ELOGPU/.)
Indicates that an uncorrectable error was detected in a memory operation initiated by the processor. This is the highest priority error. It forces the ELOG to be updated to reflect this error as first even though other error types have occurred previously.

Bit 21 Non-processor Uncorrectable Error (ELOGNU/.)
Indicates that an uncorrectable error was detected in a memory operation not initiated by the processor. This is the second highest priority error. It forces the ELOG to be updated to reflect this error type as first provided that previous errors are not of the processor uncorrectable type.

Bit 20 Single-Bit Error (ELOGSE/.)
Indicates that a single-bit correctable error occurred. This is the lowest priority error. The ELOG is updated if this is the first error detected following an ELOG clear.

Bit 19 Write Error (WR/.EL..)
When TRUE (one), indicates that information contained in the address and syndrome sections of the ELOG pertains to an error that was detected during a Write cycle.

When FALSE (zero), indicates that information contained in the address and syndrome sections of the ELOG pertains to an error that was detected during a Read cycle.

Bit 18 MBU Identification.
Always TRUE.

Bit 17 MBU Identification.
Always TRUE.

Bit 16 RAM Type (64KRAM/.).
When TRUE, indicates a 64K RAM. When FALSE, indicates a 16K RAM.

15-06 Address Bits
These bits contain the address that was being accessed in memory when the error was detected. The address may not be the same address that was received by the MBU because of address modification for multiple-stack accesses. The terms ASP0...0 and ASP1...0 identify the stack in which the error occurred.

05-00 Syndrome Bits
Code defining the type of memory error that occurred. Used to locate the bit position in error.

Diagnostic Logic

The diagnostic logic in the MBU consists of State Machine states that respond to an 11D micro so that data can be written, read or echoed through data paths normally used for MBU operations.

A review of the State Machine flows (see Figures 3-5 and 3-6) shows the five sequences of states required to exercise the five types of diagnostic operations that can be performed by the MBU. These operations, which are defined by the 11D micro and its variants, are 1) Echo Write Data, 2) Echo Address, 3) Write 22 Bits, 4) Read 22 Bits, and 5) Read Error Log.

All the diagnostic operations utilize the normal data paths for transferring data. In some instances (22-bit operations), ECC and syndrome generation are disabled.

Echo Write Data

Write data is echoed through the MBU by means of the Write Rotator and Read Data Accumulator. (Refer to Figure 3-12.) The State Machine supplies rotation and mask amounts to ensure that the 24 bits of Write data are returned to the processor in their correct bit positions.

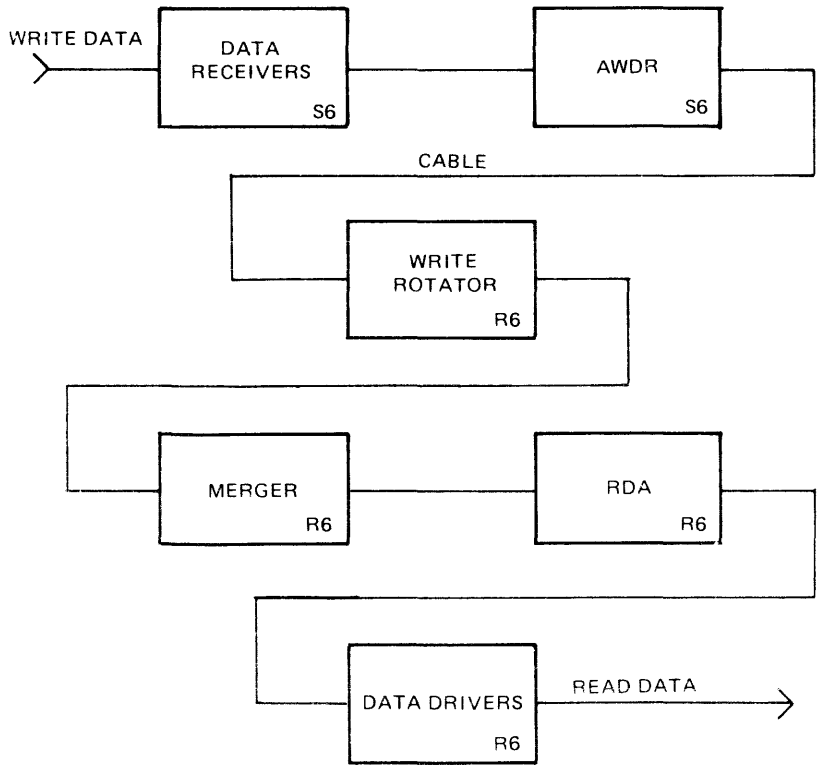


Figure 3-12. 11D Echo Write Data Path

ECHO ADDRESS

Refer to Figure 3-13 for an illustration of the Echo Address Diagnostic path.

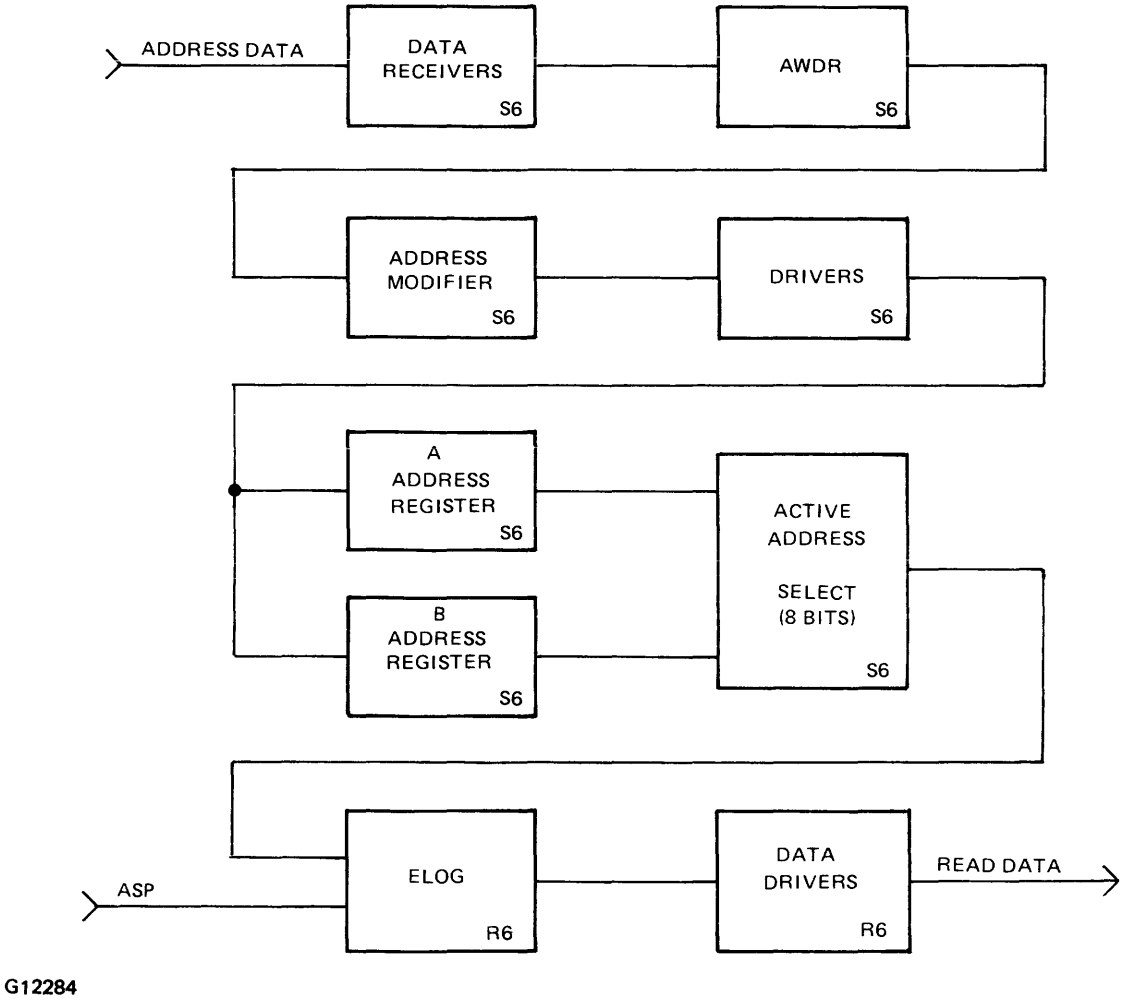
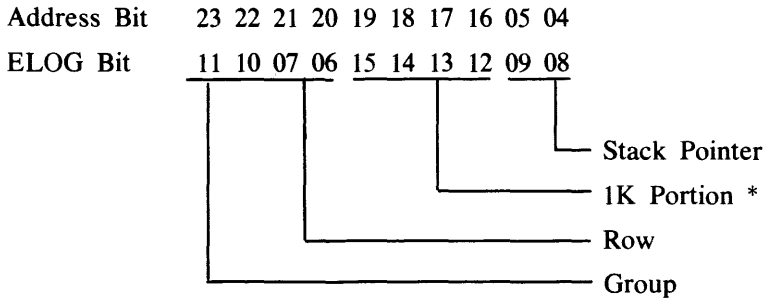


Figure 3-13. 11D Echo Address Data Path

An Echo Address operation echoes 10 address bits from the processor through the ELOG and back to the processor. The address bits that are echoed are shown in Table 3-13.

Table 3-13. ELOG Echo Address Bits



* Indicates a 1K portion of the 16K RAM. The four bits are a binary encoding (0000-1111) of the address of one of the 1K portions of the 16K RAM.

The address echoed is +16 if the direction is forward and –16 if the direction is reverse because the inputs to the ELOG are taken directly from the Active Address Selector, which is selected by the Active Stack Pointer (ASP1...). The Active Address Selector contains selected bits of the modified address.

WRITE 22 BITS

The MBU performs a 22-bit Write to memory through the Write data paths but does not enable ECC generation. The Write Rotator and Merger receive rotation and mask amounts that allow memory referencing of the data field. The Write Data Register receives the 16 bits of rotated and merged data plus six bits of Write data (bits 16-21) from the AWDR. The signal WR22/... enables the RESN to substitute these six bits for the ECC, and the signal WRECC/.. disables the ECC from the memory data-in lines.

Refer to Figure 3-14 for an illustration of the Write 22 bits data path.

READ 22 BITS

The MBU performs a 22-bit Read from memory and attaches odd parity to the 16 data bits. The data read from memory is the data written to memory during a 22-bit write. The MBU provides rotation and mask amounts to ensure that the data is processor referenced.

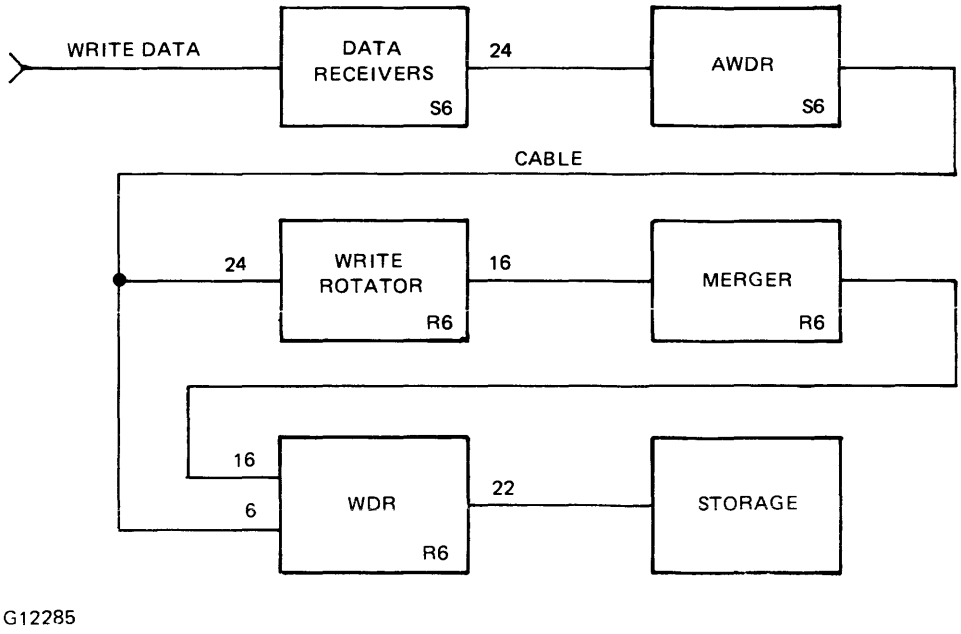


Figure 3-14. 11D Write 22 Bits Data Path

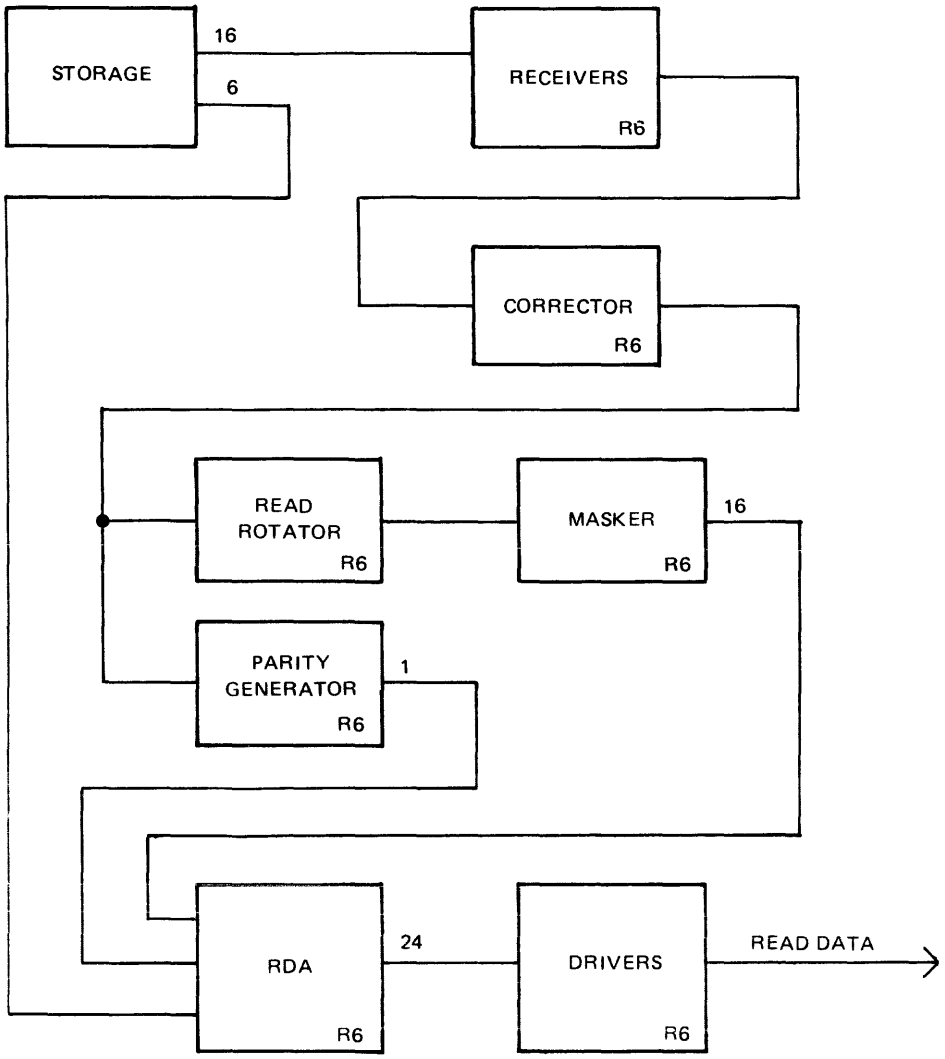
The Read data from memory takes two paths:

1. The 16 data bits are sent through the Corrector, Read Rotator, Masker and Parity Generator. The outputs of the Masker (RDA.00.. to RDA.15..) are inputs to the RESNs that comprise the RDA.
2. The six bits that represent the ECC positions of the 22-bit word (DR16..X0 to DR21..X0) and the output of the Parity Generator (PARITY..) are sent to the RESN that serves as a substitute for the normal RDA RESN (disabled by DFREAD/.). The signal R16+22/. enables these bits into the RESN for transfer to the processor along with the 16 data bits from the Masker.

Refer to Figure 3-15 for an illustration of the Read 22 bits data path.

READ ERROR LOG

The contents of the ELOG are sent directly to the Read Data Drivers in response to a Read ELOG signal (RDELOG/.) from the State Machine. Following the read, the ELOG is cleared by the signal CLRELG/. from the State Machine.



G12286

Figure 3-15. 11D Read 22 Bits Data Path

STORAGE BOARD FUNCTIONAL DETAIL

Each storage board provides a random-access memory array containing 64K (65,536) 22-bit words. The 22-bit word consists of two bytes of data plus six bits for error correction (check bits). The Processor uses the six check bits to support a single-error-correcting, double-error-detecting code. The storage board itself does not participate in error detection or correction.

The primary storage device is a dynamic MOS RAM chip. The chip contains 16,384 addressable bit locations. A fully-loaded storage board holds 88 of these 16K-by-1 RAM chips.

The storage board physically appears to contain four arrays of 16,384 22-bit words each, but these arrays are linked to form two arrays, each 16K by 44 bits in size. For address transfer and location access, 44 RAM chips are activated simultaneously.

The 44-bit array element actually comprises two 22-bit words. At a given time, through a data selector, either of these 22-bit words can be read out to the interface or written into from the interface. Also, one 22-bit word can be read out at the same time that data is being written into the other 22-bit word.

Figure 3-16 is a functional block diagram of the S-Memory storage board.

Functions and Operations

A storage board operation, called a cycle, is characterized by the presentation of an address to the storage board and the use of that address to select a unique word storage location from which information is read, or into which information is stored. The cycles that can be performed by the B 1900 storage boards are 1) Read, 2) Read-Modify-Write, 3) Refresh.

The information word that the storage board delivers on a Read cycle or the read-out phase of a Read-Modify-Write cycle is expected to be an error-free copy of the word that was previously stored in the addressed location during the write-in phase of a Read-Modify-Write cycle. The storage board retains stored information indefinitely so long as the Refresh requirement is satisfied and power is supplied without interruption.

Read Cycle

In a Read cycle, the information in the addressed word location is read out on the data-out (data-read) leads. The read-out is non-destructive.

Read-Modify-Write Cycle

This cycle comprises a Read phase and an ensuing Write phase. During the Read phase, the word in the addressed location is read out on the data-out circuits. Then, during the Write phase, a new word is stored in the location, replacing the previous contents. The Write phase of the cycle is initiated by the control after it processes the data that is read out. The new stored word may include unchanged bits from the read-out word as well as bits whose states depend on the initial states of the read-out bits.

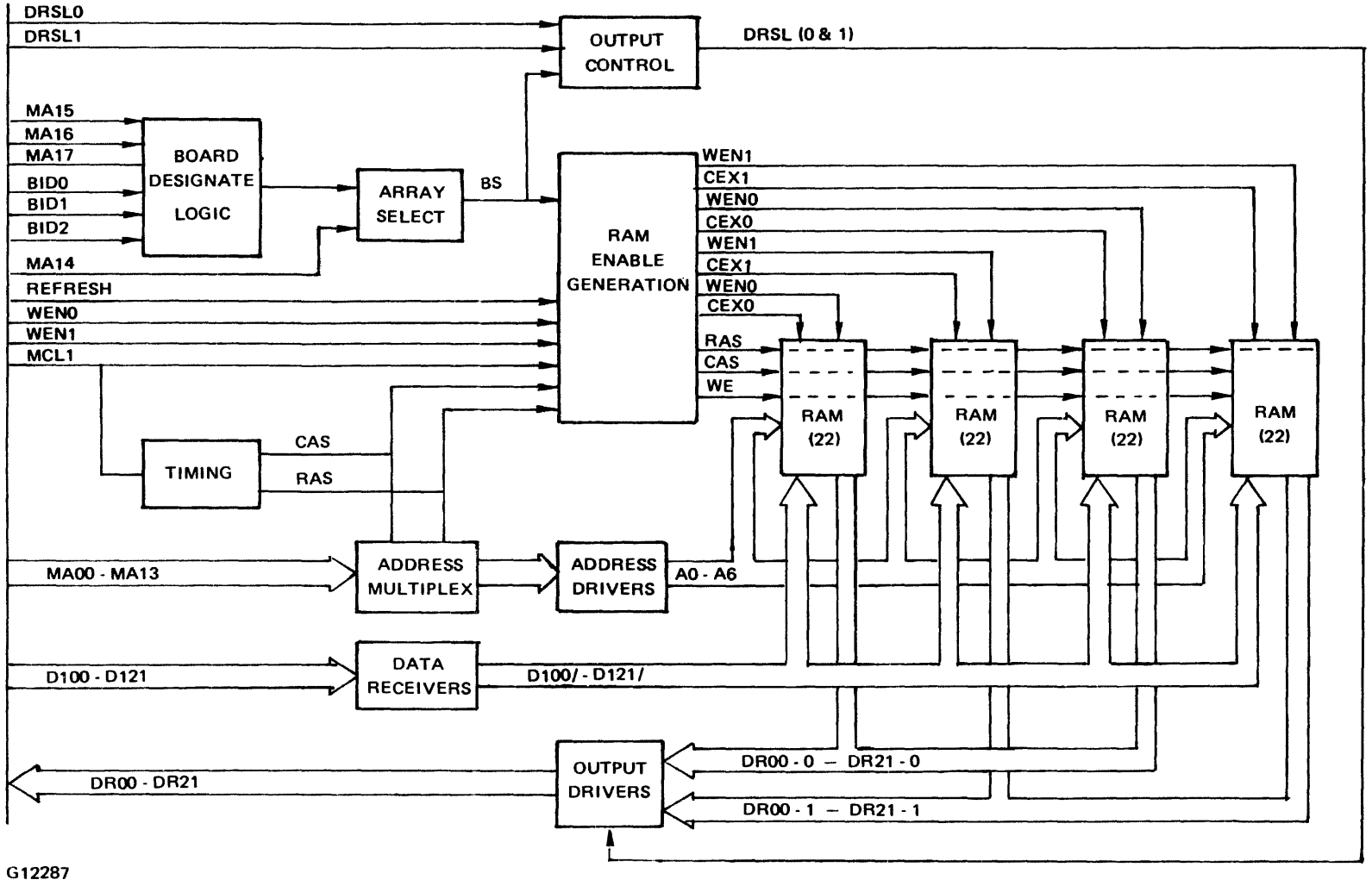


Figure 3-16. Functional Block Diagram, B 1900 Storage Board

Refresh Cycle

The Refresh cycle is an excitation cycle that preserves (recharges) the information stored in the dynamic 16K RAM chips. Refresh cycles are required regularly. No data transfer occurs, and the contents of the storage locations are retained.

The storage board requires a minimum of 128 Refresh cycles within a 2-millisecond period. The 128 cycles correspond to, and must address, the rows of the 128 by 128 matrix inside the RAM chips (A0-A6).

During a Refresh cycle, all arrays are selected so that all RAM chips on the board are excited at the same time.

System Commands

System commands are S-Memory operations required by the B 1900 Processor, which is the primary memory-requesting unit in the system.

STREAM (MICRO-OPERATOR FETCH)

The processor orders a Read of four words in succession from sequentially-addressed memory locations.

READ

The processor orders defined field Reads of zero to 24 bits, beginning at any bit position. This requires one, two or three sequentially-addressed words to be read from memory.

WRITE

The processor orders defined field Writes of zero to 24 bits, beginning at any bit position. This request results in Read-Modify-Write cycles for one, two or three sequentially addressed word locations. The read-out phase of one cycle can occur simultaneously with the write-in phase of another cycle on the same storage board.

SWAP

The processor orders a defined field Swap (Read and replace) of 0-24 bits, beginning at any bit position. For an accessed stack, a Swap is executed as a Read followed by a Read-Modify-Write (concatenated operations) within an uninterrupted storage-board cycle. One, two, or three sequentially addressed words in memory are affected. Simultaneous reading of one word and writing of another word on the same storage board can occur.

REFRESH

Memory control periodically issues a Refresh command, causing Refresh cycles.

Storage Board Input Signals

BID0 BID1 BID2

Board Identification inputs. These inputs provide an encoding of the storage board. They are used, together with MA15 (see below) to designate the specific storage board.

No connection: "1"; ground connection: "0".

MA15 MA16 MA17

Memory Address bits. These bits are used for storage board designation. If the binary patterns on MA15, MA16 and MA17 and BID0, BID1 and BID2 are respectively matched, the storage board is designated. If the two binary patterns do not match, the storage board is not designated. Except for the Refresh cycle, the cycles described previously can only occur when the storage board is designated.

MA14

When a storage board is designated, MA14 selects one of the two internal 16K-word by 44-bit arrays.

MCL1

Memory Clock-1. A HIGH signal state on this lead represents an MCL1 signal and signifies a storage board cycle. The LOW-to-HIGH signal transition initiates a cycle and, after the transition, the MCL1 signal, which is LOW between cycles, remains HIGH throughout the cycle.

MA00-MA13

Memory Address bits 00 through 13, which represent the word address to be used within the RAMs visualized as 128 rows by 128 columns. The MA00 through MA06 bits represent a 7-bit row address and the MA07 through MA13 bits represent a 7-bit column address. For a Refresh cycle, only the MA00 through MA06 bits are significant.

RFSH

Refresh. A HIGH signal state on the RFSH lead specifies a Refresh cycle. The cycle is triggered by the MCL1 signal. A Refresh cycle pre-empts any other cycle.

WEN-0, WEN-1

Write Enable. A HIGH signal state on either the WEN-0 or WEN-1 lead signifies either a Write cycle or the Write portion of a Read-Write or Read-Modify-Write cycle. For the latter two cycles, WEN HIGH initiates the data storage phase. WEN-0 and WEN-1 are used to select which 22-bit word of the internal 44-bit element is written into.

DI00-DI21

Data-In bits 00 through 21. These data bits represent the information to be stored in a memory word location during a Read-Modify-Write cycle.

DRSL-0, DRSL-1

Data-Read Select. A HIGH on either the DRSL-0 or DRSL-1 lead selects the corresponding 22-bit word and causes its transmission on the Read-Data leads. The HIGH states are mutually exclusive.

The word location designated by WEN-0 is selected by DRSL-0; the word location designated by WEN-1 is selected by DRSL-1.

TERM0, TERM1

These are connected to resistors on the storage board that are used to terminate DRSL-0 and DRSL-1.

Storage Board Output Signals

DR00-DR21

Data-Read bits 00 through 21. These signals represent the bits of a word read from a memory location. The signals represent read-out bits only when the DRSL signal is HIGH and only after data read-out has occurred in a Read or Read-Modify-Write cycle. The data-read bits are valid (so long as the DRSL signal is HIGH) until MCL1 goes LOW.

NOTES

1. The HIGH signal state represents a stored data bit having the value '1'', and the LOW signal state represents or is caused by any of the following conditions: 1) a stored data bit with value ''0'', 2) a high-impedance state on the RAM chip outputs, 3) the absence of a RAM chip in the selected internal 16K by 44-bit array, and 4) an MCL1 signal in the LOW state.

2. There is no inversion of bit value representation between the DInn signals and the DRnn signals. For both, a HIGH signal state represents the value ''1'' and a LOW signal state represents the value ''0''.

Addressing

The storage board interface includes 18 chip address leads (MA00-MA17), but only 14 (MA00-MA13) are used for the 16K RAM array selection. MA14 through MA17 are used for board and word selection.

WORD ADDRESS

Address leads MA00-MA13 represent the address that selects one of the 16,384 storage cells contained in each 16K RAM chip. The transfer of these address bits from the interface to the RAM chips is carried out by multiplexing seven address lines through tri-state inverters (IWSNs) by the signals ROW/ and COL/. when a cycle is initiated. No timing or control signal other than the cycle-initiating MCL1 signal is required from the control to effect the word address transfer.

ARRAY AND WORD SELECTION

A fully populated storage board is internally organized as two arrays of 16K words by 44 bits each. When data is written into or read from the storage board, address signal MA14 selects one of the two arrays RASEN0 and RASEN1 while address signals MA00-MA13 (A0-A6) designate one 44-bit element located in that array.

When data is stored, the WEN-0 and WEN-1 signals determine which 22-bit word in the selected 44-bit element is to be written into. When data is read, the DRSL-0 and DRSL-1 signals determine which 22-bit word of the selected 44-bit element is to be routed to the Read data bus. Data can be written into one of the two 22-bit words in the element at the same time the other word is being read.

BOARD DESIGNATION

A storage board can be written into or read from only when it is designated by address signals MA15, MA16 and MA17. A storage board is designated when the bit pattern on the MA15, MA16, and MA17 leads is identical to the bit pattern on the wired-in board ID code leads, BID0, BID1, and BID2 respectively. When a storage board is designated, the read-data bus drivers are enabled by either DRSL-0 or DRSL-1.

REFRESH/NON-REFRESH

When a storage board is designated, an MCL1 signal results in a Read or Read-Modify-Write data-access cycle on the array selected. When the signal on the RFSH (Refresh) lead is HIGH, all arrays (RASEN0 and RASEN1) on the storage board are selected together regardless of MA14, MA15, MA16 and MA17. RFSH HIGH with MCL1 HIGH results in a Refresh cycle.

ADDRESS TRANSFER TIMING

Transfer of the 14-bit word address MA00 through MA13 from the interface to the RAMs requires two 7-bit transfers. The multiplexed transfer is a function carried out on the storage board. The leading edge of the MCL1 signal triggers the first transfer, but the address selection signal and the timing signal effecting the second transfer are obtained from taps on a delay line that is driven by the MCL1 signal. Thus, the address transfer timing is controlled solely on the storage board in response to a MCL1 signal from the control.

Data-Out Drivers

The data-out circuits from the outputs of the RAM chips feed tri-state inverters (IWSNs) that drive the read-data bus (DR00-DR21).

–5 Volt Monitor Circuit

The storage board contains a means to develop the required –5V DC from the –12V input supply. To indicate an improper and damaging positive voltage level on the –5V circuit, a diode-isolated fault indication lead (designated VBBX) is included among the power and ground leads in the storage board interface.

SECTION 4
CIRCUIT OPERATIONAL DETAIL – HOST ADAPTER-3

Host Adapter-3 (HA-3) consists of one logic card that connects directly into the MBU backplane. HA-3 controls user access to the MBU and provides an interface to S-Memory both for dual processors and for multiline subsystems. HA-3 is required when more than one user must communicate with the S-Memory.

Figure 4-1 illustrates HA-3 logic.

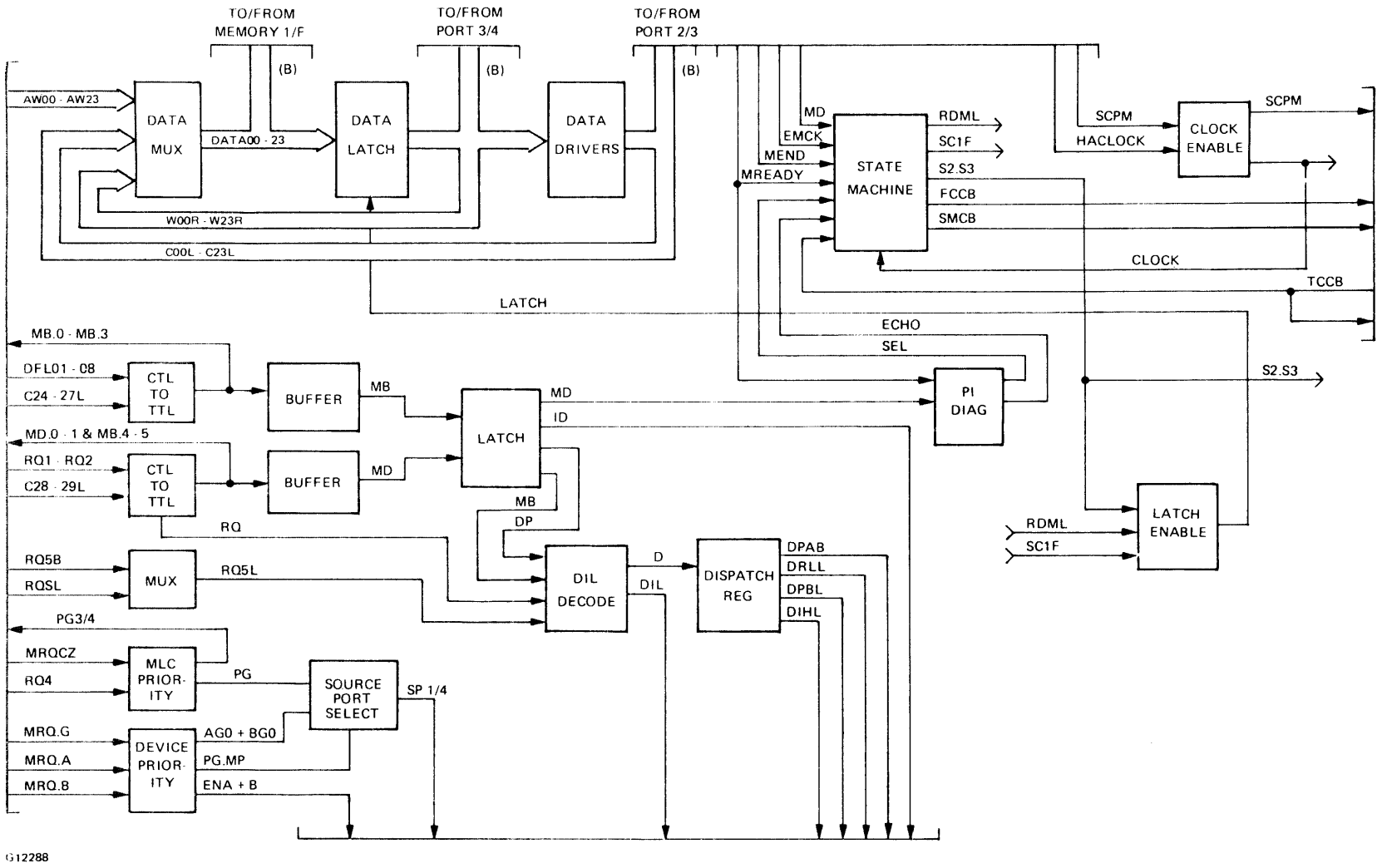


Figure 4-1. HA-3 Functional Block Diagram

PHYSICAL CHARACTERISTICS

Refer to Figure 4-2 for an illustration of HA-3 frontplane cabling.

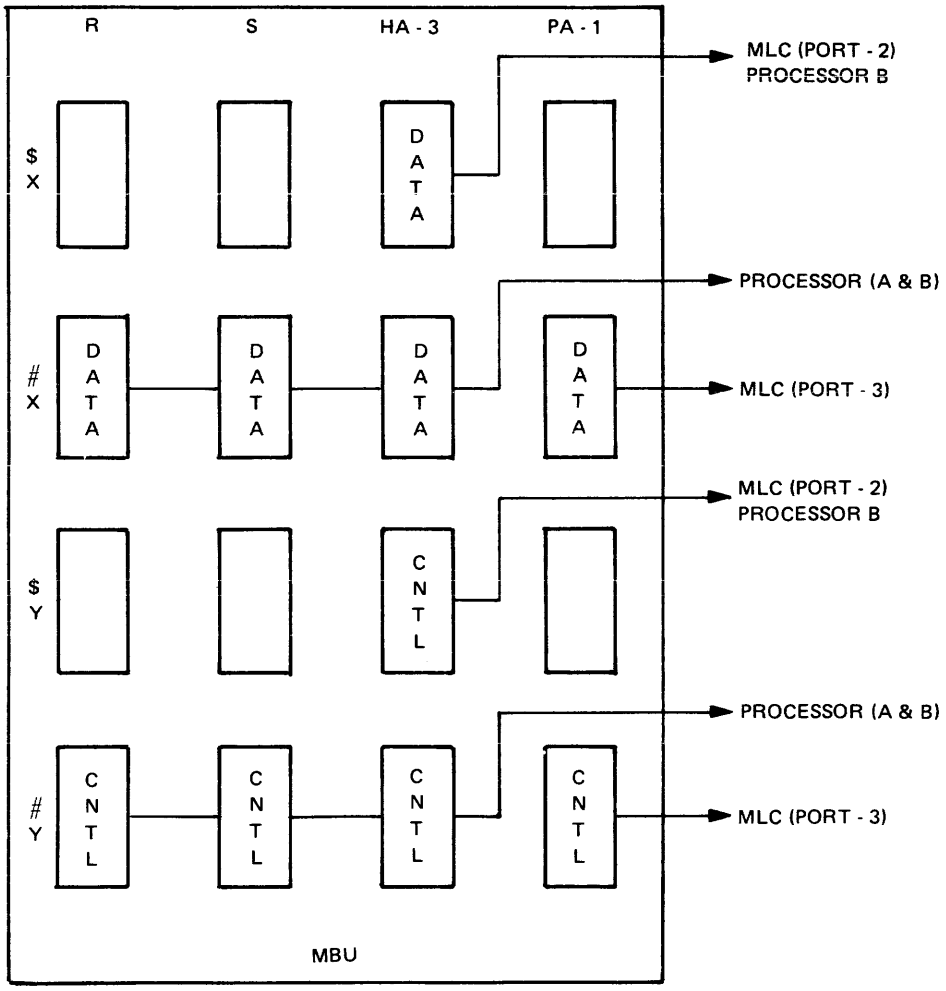


Figure 4-2. HA-3 Frontplane Cabling

HA-3 connects to the MBU data and control cables through the #X and #Y frontplane connectors respectively. One multiline control (MLC) is connected through the $X data and $Y control frontplane connectors. Table 4-1 shows this cabling.

In a dual-processor system, the MLC control cable ($Y) and data cables are shared with Processor B. The control signal SL.OLZ/1 is on pin Z of the $Y data cable. Refer to Table 4-2.

A second MLC is connected through a port adapter (PA-1) that connects to the backplane. System clocks (HACLOCK0/SCPM..A0) are buffered and sent to PA-1 on pin 0QY. Refer to Table 4-3 for frontplane pin assignments and to Table 4-4 for backplane pin assignments.

Table 4-1. MBU to HA-3 Control Cable (Daisy-Chained to Users)

Pin	Name	T0	T1	T2	T3
A	MB.0.A/1	MB.0../.		C#.1../.	C#.1../.
B	MB.1.B/1	MB.1../.		C#.2../.	C#.2../.
C	MB.2.C/1	MB.2../.		C#.4../.	C#.4../.
D	MB.3.D/1	MB.3../.		C#.8../.	C#.8../.
E	MB.4.E/1	MB.4../.		DP.1../.	DP.1../.
F	MB.5.F/1	MB.5../.		DP.2../.	DP.2../.
G	MD.0.G/1	MD.0../.		DP.4../.	DP.4../.
H	MD.1.H/1	MD.1../.		NOWRIT/.	MPCYCLE.
I	MD.2.I/1	MD.2../1			AOUT../.
J	MD.3.J/1	MD.3../.			UERR../.
K	MRQ.GK/1	MRQ.G./.	MRQ.G./.	MRQ.G./.	
L	MEND.L.1	←	←	←	MEND....
M	EMCK.M.1	←	←	←	EMCK....
N	RLOG.N.1	←	←	←	RELOG...
P	CLRB.P/1	CLEAR./.			
Q	EN.A.Q.1	EN.A....	←	←	←
R					
S	MRDY.S.1	MRDY....	←	←	←
T	MRQ.AT/1	MRQ.A./.			
U	RTC..U/1	RTC.../.			
V	PORT.V/1	PORT....			
W	DIL.AW/1	DIL.A./.			
X	DRLL.X/1	DRLL../.			
Y	DIHL.Y/1	DIHL../.			
Z	DPAL.Z/1	DPAL../.			

← Tri-state Bus enabled to processors

Table 4-2. MLC and Processor B Control Cable and Data Cable

CONTROL CABLE						
Pin		**Name**	**T0**	**T1**	**T2**	**T3**
	A	C24L.A21	MB 0	→	C# 1	C# 1
	B	C25L.B21	MB 1	→	C# 2	C# 2
	C	C26L.C21	MB 2		C# 3	C# 4
	D	C27L.D21	MB 3		C# 4	C# 8
	E	C28L.E21	MB 4		DP 1	SP 1
	F	C29L.F21	MB 5		DP 2	SP 2
	G	RQ1L.G21	RQ1L	RQ1L	DP 4	SP 4
	H	RQ2L.H21	RQ2L	RQ2L		UERR/AOUT
*	I	MP.B.I/1	→	→	→	→
	J	RQ4L.J21	RQ4L	RQ4L		
	K	RQ5L.K21	RQ5L	RQ5L		
	L	TCCL.L21	→	→	→	→
	M	FCCL.M21	←	←	←	←
	N	DRLL.N21	←	←	←	←
	P	DIL..P21	←	←	←	←
*	Q	MA.B.Q.1	←	←	←	←
*	R	MRQB.R/1	→	→	→	→
	S	PDPL.S21	→	→	→	→
*	T	DILB.T/1	←	←	←	←
**	U	PDPLCU.1	PORT5	→	→	→
*	V	EN.B.V.1	→	→	→	→
	W	CLRL.W21	←	←	←	←
**	X	EN.C.X.1	PORT 5			
**	Y	DILC.Y/1	PORT 5			
**	Z	MRQ.CZ/1	PORT 5			

DATA CABLE						
Pin		**Name**	**T0**	**T1**	**T2**	**T3**
	A-Y	DATAnnW1	ADDRXX..	→	WRITEXX.	READXX..
***	Z	SL.OLZ/1	SL.OLZ.1	→	→	→

← Bus enabled to devices
→ Bus enabled from devices
* Processor B signals
** Reserved for C-type device
*** Slave On-line: Both master and slave are on-line

Table 4-3. HA-3 Frontplane Pin Assignments

$X	#X	$Y	#Y
$AX – C00L.A21	#AX – DATA00W1	$AY – C24L.A21	#AY – MB.0.A/1
$BX – C01L.A21	#BX – DATA01W1	$BY – C25L.B21	#BY – MB.1.B/1
$CX – C02L.A21	#CX – DATA02W1	$CY – C26L.C21	#CY – MB.2.C/1
$DX – C03L.A21	#DX – DATA03W1	$DY – C27L.D21	#DY – MB.3.D/1
$EX – C04L.A21	#EX – DATA04W1	$EY – C28L.E21	#EY – MB.4.E/1
$FX – C05L.A21	#FX – DATA05W1	$FY – C29L.F21	#FY – MB.5.F/1
$GX – C06L.A21	#GX – DATA06W1	$GY – RQ1L.G21	#GY – MD.0.G/1
$HX – C07L.A21	#HX – DATA07W1	$HY – RQ2L.H21	#HY – MD.1.H/1
$IX – C08L.A21	#IX – DATA08W1	$IY – MP.B.I/1	#IY – MD.2.I/1
$JX – C09L.A21	#JX – DATA09W1	$JY – RQ4L.J21	#JY – MD.3.J/1
$KX – C10L.A21	#KX – DATA10W1	$KY – RQ5L.K21	#KY – MRQ.GK/1
$LX – C11L.A21	#LX – DATA11W1	$LY – TCCL.L21	#LY – MEND.L.1
$MX – C12L.A21	#MX – DATA12W1	$MY – FCCL.M21	#MY – EMCK.M.1
$NX – C13L.A21	#NX – DATA13W1	$NY – DRLL.N21	#NY – RLOG.N/1
$PX – C14L.A21	#PX – DATA14W1	$PY – DIL..P21	#PY – CLRB.P/1
$QX – C15L.A21	#QX – DATA15W1	$QY – MA.B.Q.1	#QY – EN.A.Q.1
$RX – C16L.A21	#RX – DATA16W1	$RY – MRQB.R/1	#RY –
$SX – C17L.A21	#SX – DATA17W1	$SY – PDPL.S21	#SY – MRDY.S.1
$TX – C18L.A21	#TX – DATA18W1	$TY – DILB.T/1	#TY – MRQ.AT/1
$UX – C19L.A21	#UX – DATA19W1	$UY – PDPLCU.1	#UY –
$VX – C20L.A21	#VX – DATA20W1	$VY – EN.B.V.1	#VY – PORT.V/1
$WX – C21L.A21	#WX – DATA21W1	$WY – CLRL.W21	#WY – DILA.W/1
$XX – C22L.A21	#XX – DATA22W1	$XY – EN.C.X.1	#XY – DRLL.X/1
$YX – C23L.A21	#YX – DATA23W1	$YY – DILC.Y/1	#YY – DIHL.Y/1
$ZX –	#ZX – SL.OLZ/1	$ZY – MRQ.CZ/1	#ZY – DPAL.Z/1

Table 4-4. HA-3 Backplane Pin Assignments

0X	1X	0Y	1Y
0AX – 4.75	1AX – 4.75	0AY – W08R23B0	1AY – AW20B.P0
0BX – W31R00B0	1BX – AW00B.P0	0BY – DPABA.P0	1BY – AW21B.P0
0CX – W30R01B0	1CX – AW01B.P0	0CY – CLRB..C0	1CY – AW22B.P0
0DX – W29R02B0	1DX – GROUND	0DY – DFL01BP0	1DY – GROUND
0EX – W28R03B0	1EX – AW02B.P0	0EY – DFL02BP0	1EY – AW23B.P0
0FX – W27R04B0	1FX – AW03B.P0	0FY – DFL04BP0	1FY – ID1B..B0
0GX – W26R05B0	1GX – AW04B.P0	0GY – DFL08BP0	1GY – ID2B..B0
0HX – W25R06B0	1HX – AW05B.P0	0HY – DFL16BP0	1HY – ID4B..B0
0IX – W24R07B0	1IX – AW06B.P0	0IY – FDSB..P0	1IY – ID8B..B0
0JX – W23R08B0	1JX – GROUND	0JY – PSC=1BC0	1JY – GROUND
0KX – W22R09B0	1KX – AW07B.P0	0KY – PC2FB.C0	1KY – HACLOCKO
0LX – W21R10B0	1LX – AW08B.P0	0LY – PG4...C0	1LY – SMCB..C0
0MX – W20R11B0	1MX – AW09B.P0	0MY – PARITYA0	1MY – SP1B..C0
0NX – W19R12B0	1NX – AW10B.P0	0NY – RQ1B..P0	1NY – SP2B..C0
0PX – W18R13B0	1PX – AW11B.P0	0PY – RQ2B..P0	1PY – PG3...C0
0QX – W17R14B0	1QX – GROUND	0QY – SCPM..A0	1QY – GROUND
0RX – W16R15B0	1RX – AW12B.P0	0RY – RQ4L.4P0	1RY – TCCB..C0
0SX – W15R16B0	1SX – AW13B.P0	0SY – RQ5B..C0	1SY – FCCB..C0
0TX – W14R17B0	1TX – AW14B.P0	0TY – DRLB..C0	1TY – RQ4L.3P0
0UX – W13R18B0	1UX – AW15B.P0	0UY – DIHB..C0	1UY – SCPM.LD0
0VX – W12R19B0	1VX – AW16B.P0	0VY – DPAB..C0	1VY – DIL.4.C0
0WX –	1WX – GROUND	0WY – DIL.3.C0	1WY – GROUND
0XX – W11R20B0	1XX – AW17B.P0	0XY –	1XY – SP4B..C0
0YX – W10R21B0	1YX – AW18B.P0	0YY –	1YY – PWR.UP.0
0ZX – W09R22B0	1ZX – AW19B.P0	0ZY – 2.0	1ZY – 2.00

FUNCTIONAL CHARACTERISTICS

HA-3 provides the following logical functions for processor/S-Memory interfacing:

1. Priority logic for memory access.
2. Dispatch logic for port functions.
3. Master/slave and on-line/off-line detection logic.
4. Processor B and port connect logic.
5. Diagnostic logic for 11D micro.

HA-3 provides th following logical functions for S-Memory/MLC interfacing:

1. Priority logic for MLC accesses.
2. Dispatch logic for MLC operations.
3. Data multiplexing of CTL bidirectional signals of MLC to TTL bidirectional signals of MBU.
4. MBU error conditions path to the processor.
5. Clear and power-up levels to the MLC.

HA-3 is used when more than one user requires access to S-Memory. Possible users are 1) Processor A, 2) Processor B, and 3) a Multiline subsystem (either HA-3 or PA-1 connect).

HA-3 determines which user can have access and ensures that all data and control for the user is routed correctly. The control mechanism for synchronizing user/S-Memory events is a sequence counter called the State Counter.

State Counter

The State Counter consists of four flip-flops (SC1F-SC4F) that have values from zero to four (S0-S4). These states denote activity within HA-3 as follows:

S0 Idle (Accept Address)
S1 Memory Started
S2 Write
S3 Read
S4 End Read

State 0 (S0)

S0 (SC1F/...) is the idle state for HA-3. Upon receipt of an MRQ.GK/1 request, if memory is ready (MRDY) and access is granted (GO.GLOB.), address information is accepted and State 1 is entered.

State 1 (S1)

S1 (SC1F....) is the memory-started state, in which paths between the user and S-Memory are enabled. SC1F remains set until receipt of a Memory End (MEND.L.1) signal from the MBU and causes SC2F to be set with the next clock (CLOCK1/.). S1 remains set.

State 2 (S2)

S2 (SC2F....) is the Write state of HA-3. It allows data to be latched for transfer to S-Memory (S2.S3/..) and is reset when Memory End occurs.

If Memory Clock (EMCK../.) occurs during S2, SC3F is set and HA-3 enters the Read state. S1 and S2 remain set.

State 3 (S3)

S3 (SC3F....) is the Read state for HA-3, in which data is transferred from S-Memory to the user. An exit from S3 does not occur until S4 is active and is in the process of being reset (coincident reset).

State 4 (S4)

S4 (SC4F....) is the end-Read state of HA-3 where both S3 and S4 are brought to a read-idle condition by the occurrence of either a processor (MP.CYC..) or a multiline (TCCB...0) control signal. SC4F and SC3F are simultaneously reset once S4 is entered. This is the Memory End condition. Since Memory End resets S1 and S2 and sets S4, HA-3 is not completely idle until S4 resets. If S3 is not set, S4 resets with the next clock. If S3 is set, one of the control signals must occur to reset both states.

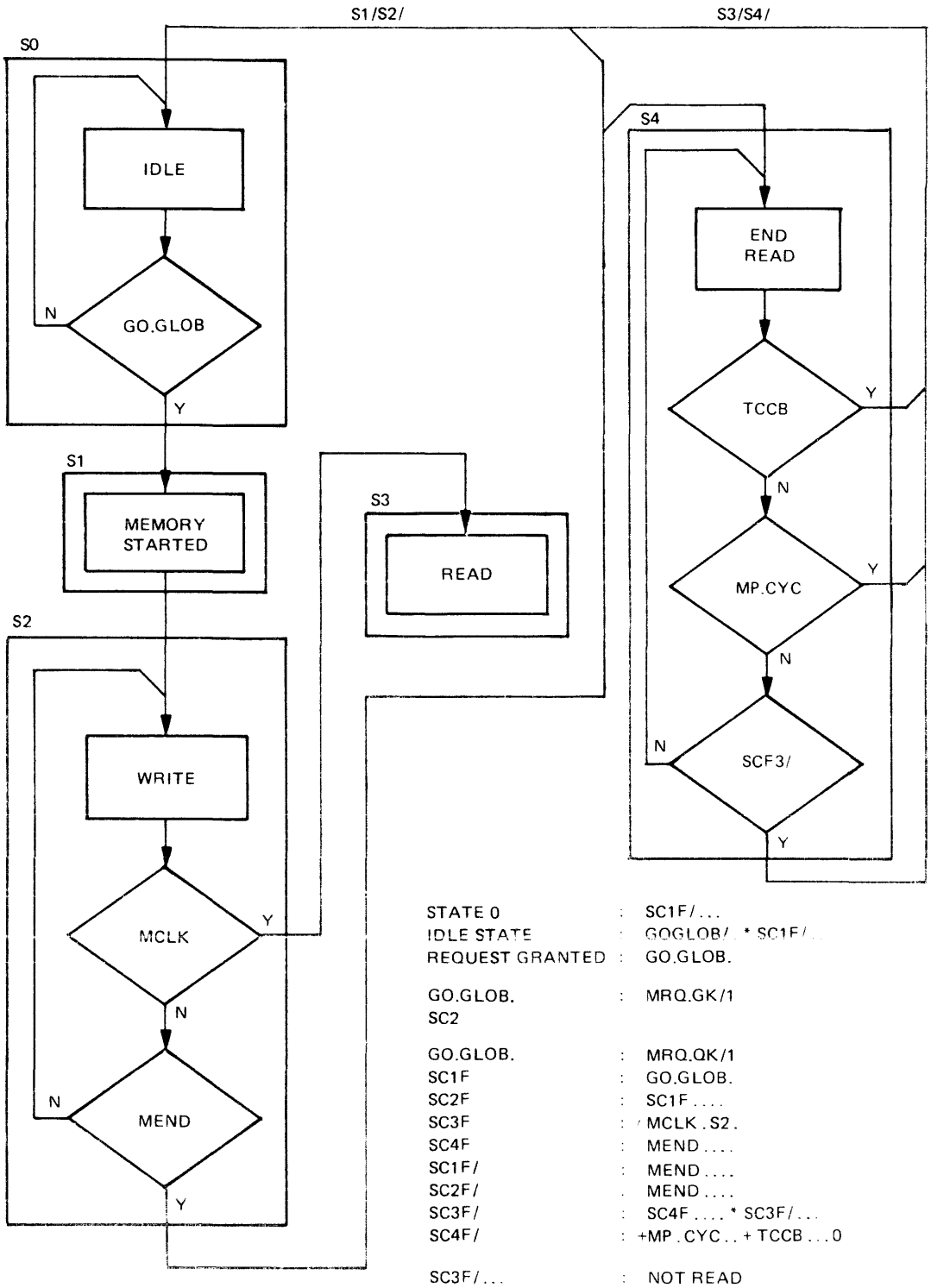
Refer to Figure 4-3 for an illustration of the state flows.

Data Paths

Data paths are provided to allow connection between the S-Memory, processors, and multiline controls. Data can flow independently between S-Memory and the processor or between S-Memory and the multiline control. Once a connection has been established, the data paths are dedicated to the user for the duration of the memory cycle. Once the cycle ends, contention is re-established and another user can gain access to the data paths.

Refer to Figure 4-4 for an illustration of the data paths.

The data paths consist of multiplexors (S4-Ns), latches (L8SNs), tri-state buffers (BXSNs) and line drivers (LDANs) that are enabled and selected by the State Counter during the state(s) associated with the Write, Read or Address portion of a memory access cycle. Since at least three states can be active at one time, their combined effect allows the data paths to open and close so that data may be steered between S-Memory and the user.



STATE 0	:	SC1F/...
IDLE STATE	:	GOGLOB/ * SC1F/...
REQUEST GRANTED	:	GO.GLOB.
GO.GLOB.	:	MRQ.GK/1
SC2		
GO.GLOB.	:	MRQ.QK/1
SC1F	:	GO.GLOB.
SC2F	:	SC1F....
SC3F	:	MCLK.S2.
SC4F	:	MEND....
SC1F/	:	MEND....
SC2F/	:	MEND....
SC3F/	:	SC4F.... * SC3F/...
SC4F/	:	+MP.CYC.. + TCCB...0
SC3F/...	:	NOT READ
MP.CYC..	:	MLC ON PORTS 4 & 5
TCC3...0	:	MLC READ DATA ACCEPTED (PORTS 2 & 3)
ACCEPT ADDRESS	:	GO.GLOB. * SC1F/...
READ IDLE	:	SC3F/... * SC4F/...

Figure 4-3. HA-3 State Flows

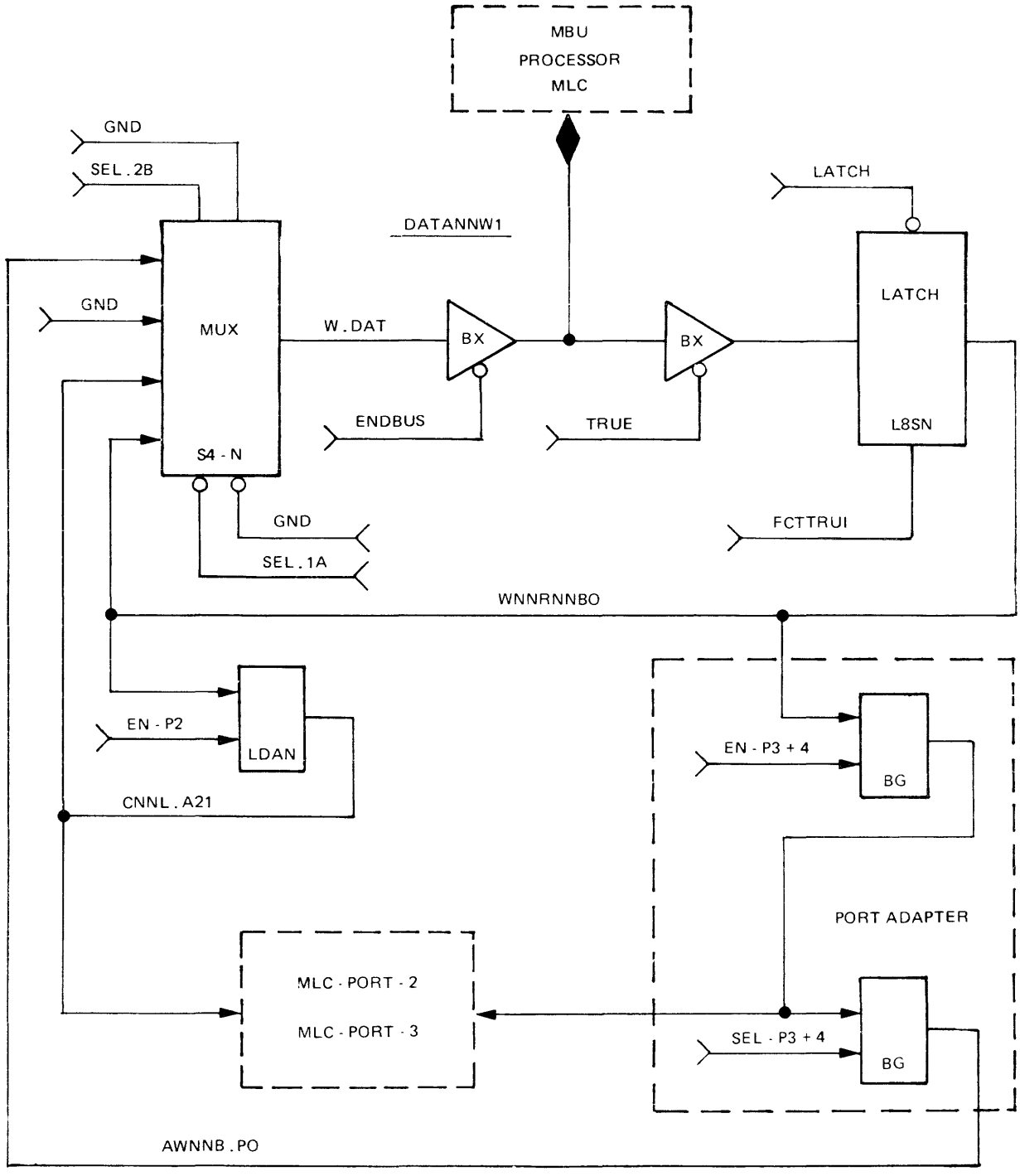


Figure 4-4. HA-3 Data Paths

S-Memory Requests

S-Memory requests are generated by a user whenever an access to S-Memory is required. In a multiple user environment, the burden of priority selection for memory access falls on HA-3. Figures 4-5 and 4-6 illustrate the timing sequence for processor selection by HA-3 for S-Memory access. Figure 4-7 illustrates HA-3 logic that selects which user gets a memory cycle (priority). Figure 4-8 illustrates the processor logic on Card D6 that selects a processor.

Figure 4-5 shows the sequence required to grant access to Processor A when it requests a memory cycle. The number of clocks that are illustrated are only indicative of the "T" times at which an event occurs and not of the actual number of clocks that occur.

Both Processor A and Processor B are enabled while HA-3 is idle. When Processor A requests, both MRQ A and MRQ GLOBAL come TRUE. At T0 time, if MEMORY READY is TRUE, Processor B is disabled and priority is granted to Processor A. When MEMORY END occurs during T3 time, HA-3 starts to become idle and proceeds to enable both Processor A and Processor B for the next request. The processor enables, defined as (P), occur one clock later in each processor.

Figure 4-6 shows the sequence of events that occur when both processors request memory access simultaneously. During the first T0 time, Processor A is granted priority as if a single processor made the request. Once MEMORY END occurs, HA-3 becomes idle and when MEMORY READY comes TRUE at T0 time, Processor B is granted a memory cycle.

Multiline control requests are similar to those for the processors. HA-3 does not make any distinction to the type of user except to establish priority with respect to the origin (port adapter or cable connect) of the user.

Refer to Table 4-5 for request level coding for multiline controls.

Table 4-5. Request Level Coding

Memory Cycle	Dispatch	Request	Mode			
	RQ5L	RQ4L	RQ3L	RQ2L	RQ1L	Function
Normal	0	0	0	0	0	No Request
	0	1	0	0	0	Read Request
	0	1	0	0	1	Write Request
	0	1	0	1	0	Swap Request
Dispatch	1	0	0	0	0	Read and Set Lock
	1	0	0	0	1	Write and Set Low Interrupt
	1	0	0	1	0	Read and Clear Interrupt
	1	0	0	1	1	Write and Set High Interrupt

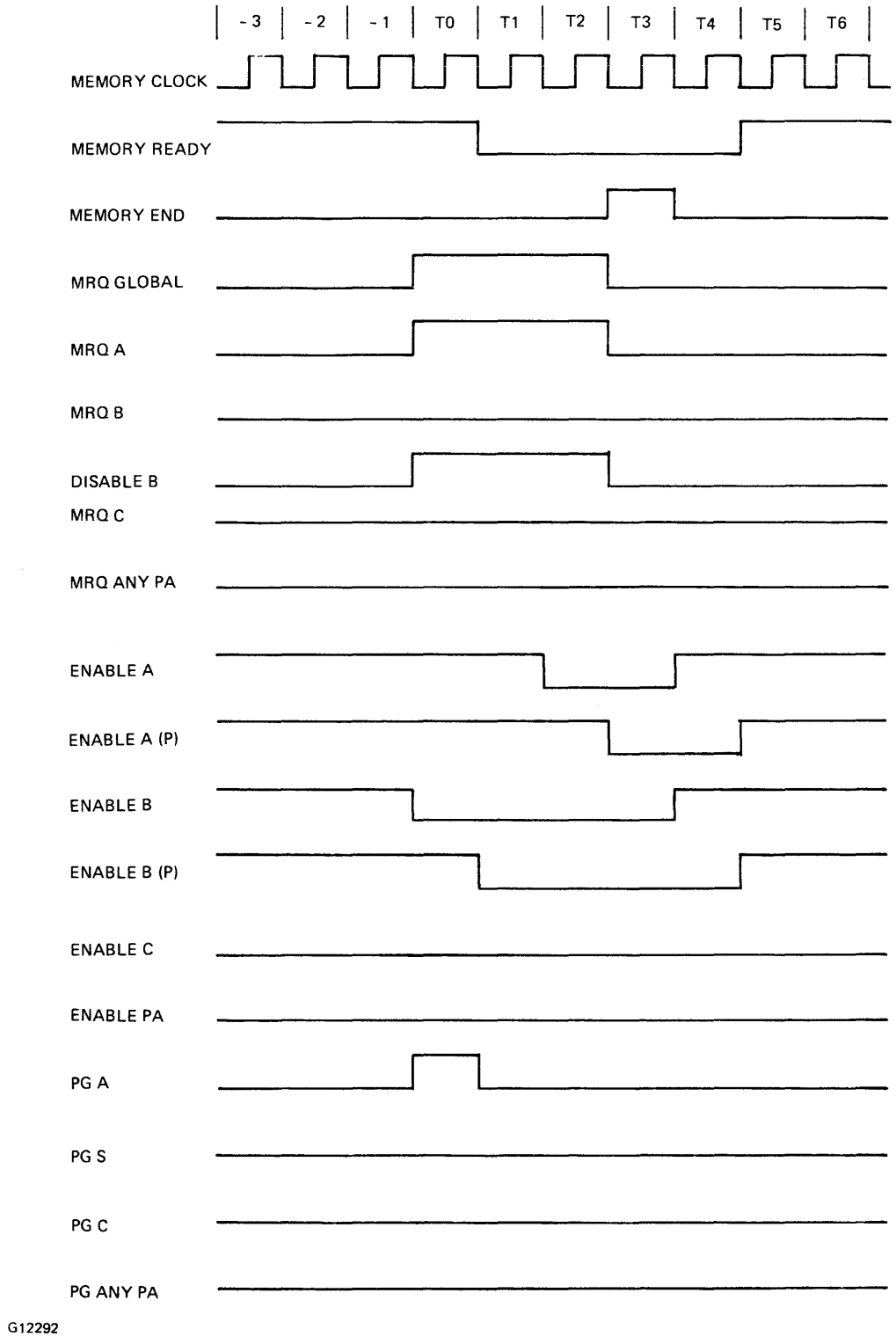


Figure 4-5. S-Memory Request Sequence for Processor A

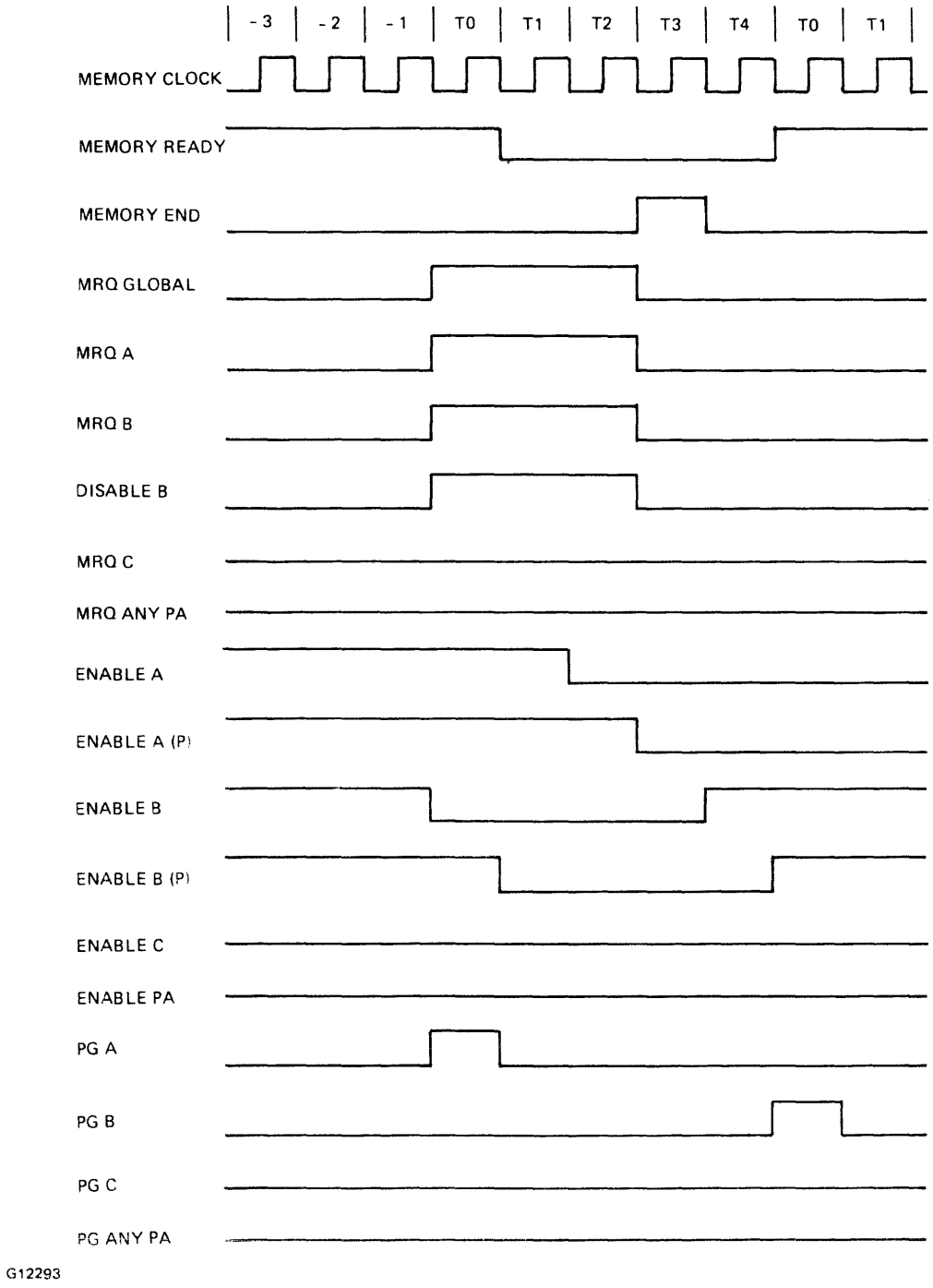


Figure 4-6. S-Memory Request Sequence for Processors A and B

Request Priority

Refer to Figure 4-7.

An explanation of the controlling terms for granting memory request priority are as follows:

A.GO../.
T0 time of memory cycle granted to Processor A.

A/ORDSA.
Processor A is either disabled or not requesting. Therefore it was not granted the current memory cycle.

C.ASK...
When neither processor has raised a global request or HA-3 is idle (MEND....), a port device or C-type device (REQ.P+C.) is requesting.

DISA../.
Blocks Processor A enable until one Processor B memory cycle request has been serviced.

EN.AF./.
Controls Processor A enable when Processor A enabled flip-flop is set.

EN.GDA/.
Controls Processor A requests for one clock when EN.AF... is reset by a multiline control request. It is possible for Processor A or Processor B to steal the memory cycle during the clock following the reset.

EN.A.Q.1
Enables signal to processor A.

PG.CTL/.
Active during times T0, T1 and T2 of a multiline control memory cycle.

PG.MP/..
Priority granted to a non-processor device.

REQ/P+C.
Indicates a port request via the MLC priority scanner logic.

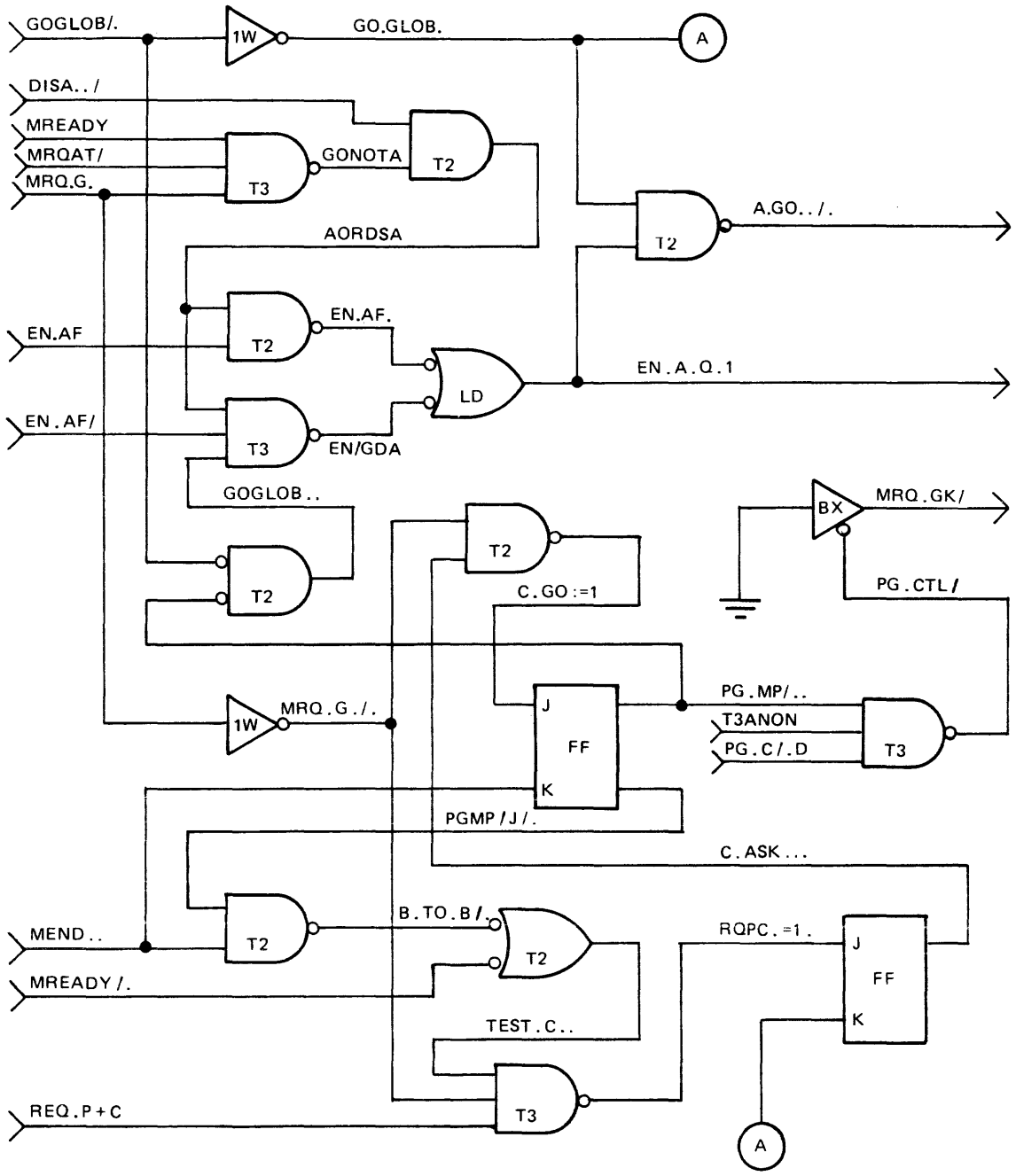
Processor Selection

The processor selection of memory accesses is controlled on card D6 of the processor(s). Refer to Figure 4-8 for an illustration of the logic involved with processor selection.

When HA-3 is idle, the processor (both processors in a dual system) are enabled for memory access (EN.A.Q./ and EN.B.V.1). These two terms are controlled by a tri-state enable (B-PROC..) and are selected as the memory request enable (ENMREQ..) that serves as a gating term for enabling control and data transfers.

When a memory request (MREQ..*.) occurs, all gates are enabled and control terms MRQ.GK/1, MRQ.AB/.and SMBOE/D0) are generated. The HIGH or LOW state of the term B-PROC.. determines which processor is issuing the memory request. If the term is HIGH, Processor A is requesting (MRQA.T/1); if it is LOW, Processor B is requesting (MRQB.R/1).

The term SMRQ/AD1 serves to disable Processor B when Processor A is enabled and requesting a memory cycle. The output of Processor A (SMRQ/AD1) is selected to drive a coax cable through a doghouse connector to Processor B. By means of a jumper, SMRQ/AD1 is available to prevent Processor B from driving the S-Memory interface data and control busses.



G12294

Figure 4-7. Memory Request Priority Logic

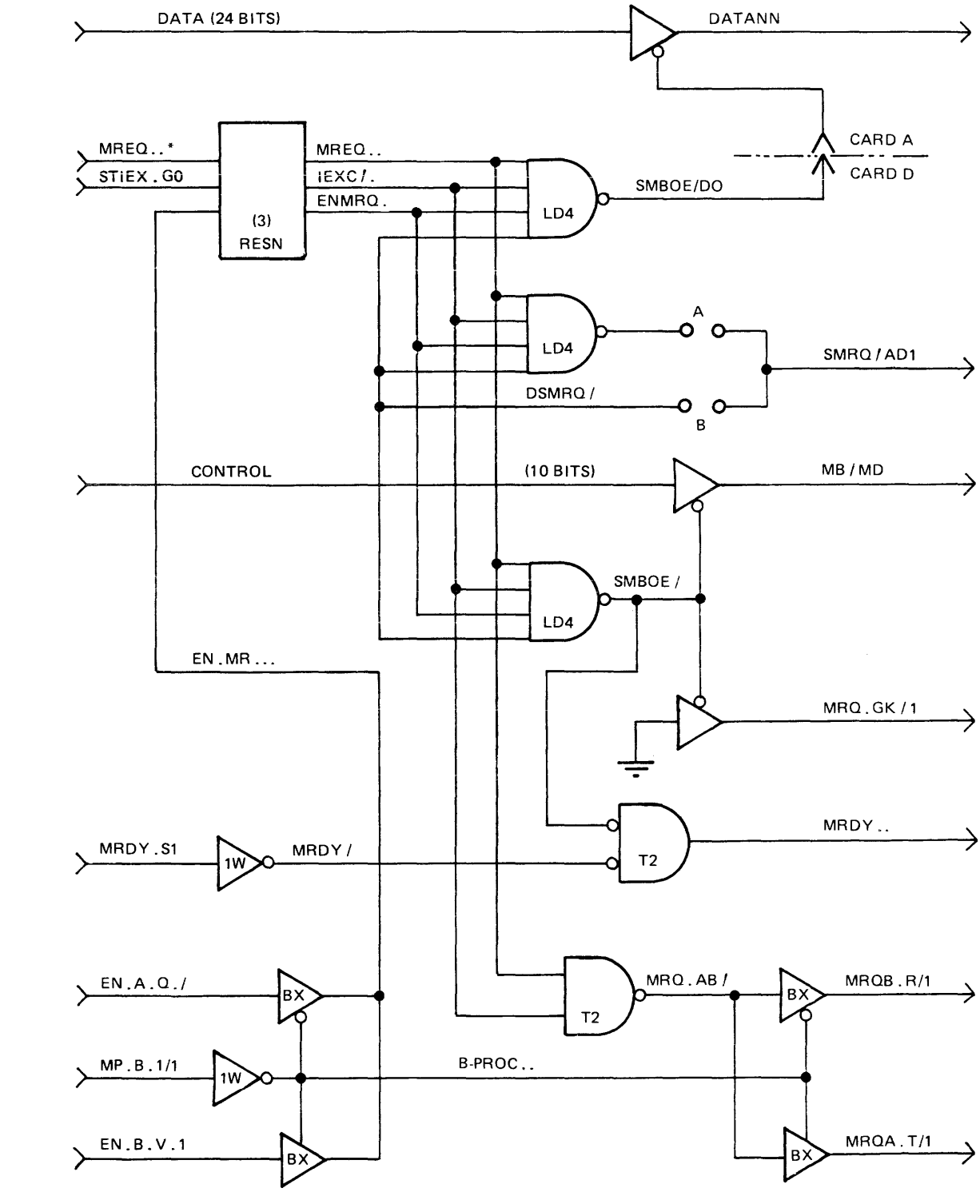


Figure 4-8. Processor Selection Logic for Memory Access Priority

Processor Priority

These terms are sent to HA-3 for priority resolution of the memory request. HA-3 returns one of the enable levels EN.A.Q./ or EN.B.V.1 to disable the processor not issuing the memory request.

The following terms affect the operation of the processors and HA-3. Their effects are explained as follows:

EN.A.Q.1
Enables Processor A's memory request logic.

EN.B.V.1
Enables Processor B's memory request logic.

DATAnnW1
The 24 data lines on the S-Memory Bus that originate on processor Card A, HA-3 or the C-type device.

MRDY....
Processor signal that indicates that a memory cycle has been granted to that processor. (SMBOE/.. * MRDY.S.1).

MRDY.S.1
The MBU is not busy and is ready to accept a request.

MB/MD
Mode and micro bit lines of control cable to the MBU.

MP.B.I/1
Defines the condition of Processor B. A TRUE level indicates that the Processor B control cable is connected and allows selection of appropriate requests and enables.

MRQ.AT/1
Processor A memory request. Active until executed.

MRQ.BR/1
Processor B memory request. Active until executed.

MRQ.GK/1
Global memory request. Raised by the processors, HA-3 (for PA-1 devices) and C-type device.

SMRQ/AD1
Used by Processor A to disable Processor B if simultaneous memory requests occur while HA-3 is idle (both Processor A and Processor B are enabled while HA-3 is idle).

SMBOE/..
Generated on processor Card D to enable the S-Memory Bus during T0, T1 and T2. (MREQ..*. * IEXC/... * ENMRQ...).

Dispatch Cycles

Whenever memory requests (MRQs and RQ5Ls) occur, the ensuing operation can require port selection and dispatch information to be generated. The decoding of source (request issuing) port and dispatch information ensures that both are received by the destination port.

Dispatch cycles allow the processors and multiline controls to communicate between themselves. Information is written into absolute address zero in S-Memory by the device that issues a dispatch interrupt and is read by the device that receives the dispatch interrupt. Also, the Dispatch register is updated for use by the device that receives the dispatch interrupt.

Dispatch Register

The Dispatch register is a 13-bit register that contains information concerning the dispatch. All 13 bits remain unchanged until the device that received the dispatch interrupt performs a Dispatch Read and Clear cycle. The meanings of the 13 bits are as follows:

LOCKOUT
: Informs all devices that the Dispatch register is in use. This bit is set during either a Dispatch Lock or a Write cycle and prevents further dispatches until the Dispatch register is cleared.

SOURCE PORT
: Consists of three bits that contain a binary value of the port (device) that issued the dispatch. These bits are set during a Dispatch Write cycle and are read by the device that receives the dispatch interrupt during a Dispatch Read and Clear cycle.

DESTINATION PORT
: Consists of three bits that contain a binary value of the port (device) that is to receive the dispatch. Once the binary value is decoded, a dispatch interrupt (DIL) is issued by HA-3 to the destination port.

CHANNEL NUMBER
: Four bits that are set by the source port during a Dispatch Write cycle and received by the destination port during a Dispatch Read and Clear cycle. Used to indicate one of 16 channels.

HIGH INTERRUPT
: This bit is set by the source port to expedite the dispatch message sequence. The device on the destination port uses the High Interrupt as an indication that an operation is complete. If the interrupt is not used, the device informs the processor by descriptor linking.

PORT ABSENT
: This bit is set by HA-3 when it detects that a device is not present on the destination port. All ports receive this indicator.

Dispatch Interrupt Level (DIL)

Refer to Figure 4-9 for an illustration of the DIL logic.

HA-3 is capable of issuing six DIL signals in response to a dispatch indication on the control cable. Three bits are detected by HA-3 as destination port information (DPORT.1.-DPORT.4.) and used by a decoder (DC30) as DILs to any one of six ports.

The term CK.DPC.. clocks the port and channel information into an octal flip-flop (RESN) and the lower three outputs are decoded by the decoder during a Dispatch Write cycle (DS.WRF..). The decoded output (DIL.0./.-DIL.5./.) is sent to the destination port.

The signal Processor B is Master (MA.B.Q.1) when HIGH enables DIL.0./. as the master port (Processor B) and DIL.1./. as the slave port (Processor A).

When MA.B.Q.1 is FALSE, Processor A is the master processor and receives the DILA.W/1 (DIL.0./.) and Processor B is the slave processor and receives the DILB.T/1 (DIL.1./.).

Port 2 (DIL..P21) is CTL converted through a line driver (LDAN) for connection to a multiline control. Port 3 (DIL.3.C0) and Port 4 (DIL.4.C0) are sent to devices that are associated with Port Adapter-1 (PA-1). Port 5 (DILC.Y/1) is for a new multiline control that connects directly to the S-Memory bus and uses a processor style interface.

Port Absent Detection

When a DIL is issued by HA-3 and the destination port does not have a device present, the port absent bit in the Dispatch register is set to indicate the absent condition (DPAL....) for that port. Refer to Figure 4-10 for an illustration of the port absent detection logic.

The port absent bit (DPAB:=1.) can be set from the following sources:

1. Processor issuance of a dispatch micro (D.MICRO.).
2. Any port except Port 0 (Port 0 is the master processor and is always present).

The slave processor port (Port 1) is enabled by the Slave On-Line term SL.OLZ/1. If the slave is not on-line, any dispatch to Port 1 reports absent.

Port 2 is enabled by the term PDPL.S21 (Port Device Present on HA-3 cables). If a multiline control is not present, any dispatch to Port 2 reports absent.

Port 3 and Port 4 connect to port adapters, Any dispatch to these ports with no device present requires a return of the backplane signal DPABA.P0 (Dispatch Port Absent). Port 4 is not used although the logic is present. The presence of a third port device other than a second processor is unlikely.

The signal PDPLCU.1 (Port Device C Present) is used in conjunction with a dispatch to Port 5. If a device is not present on Port 5, the port absent bit is set.

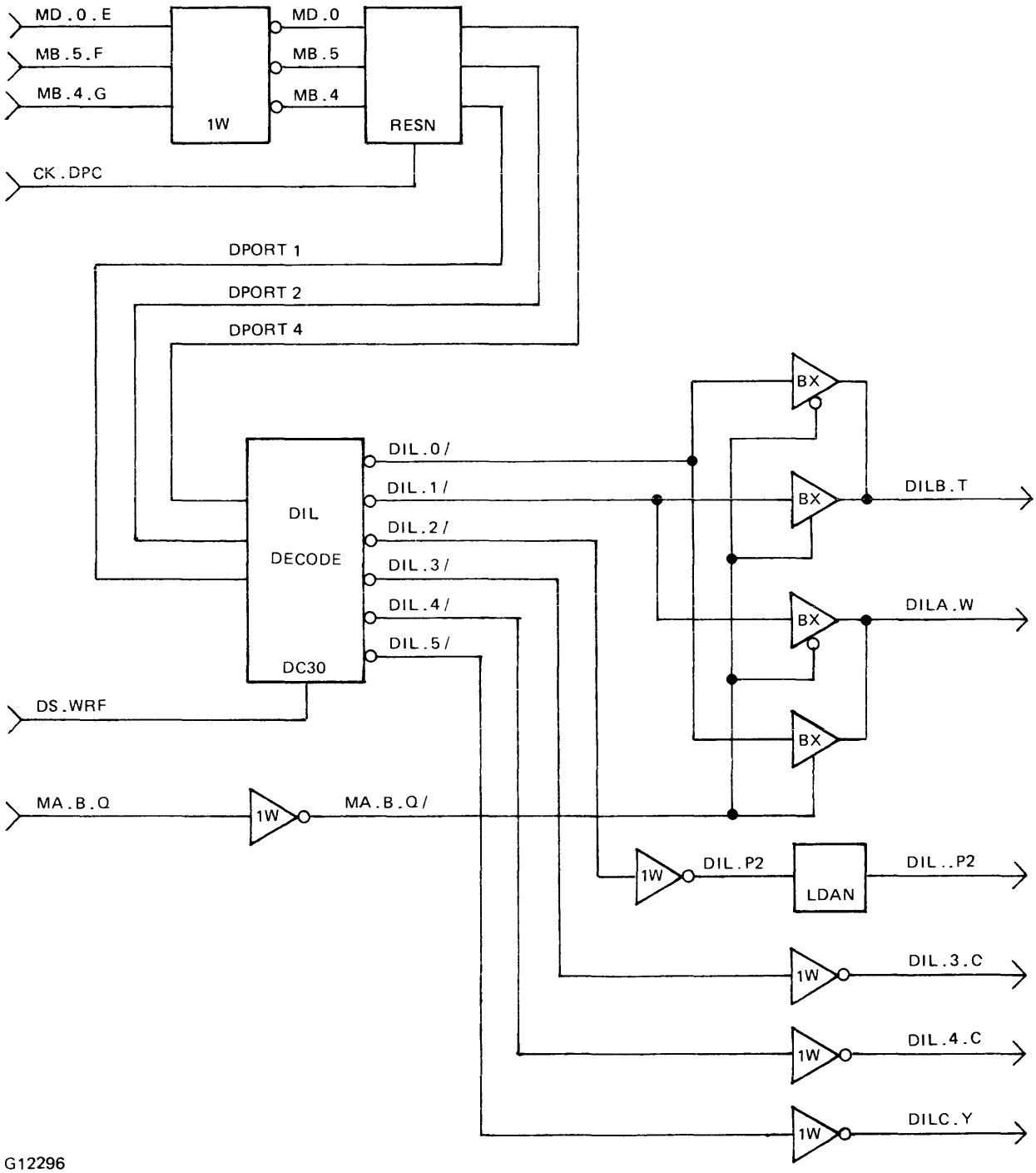


Figure 4-9. Dispatch Interrupt Logic

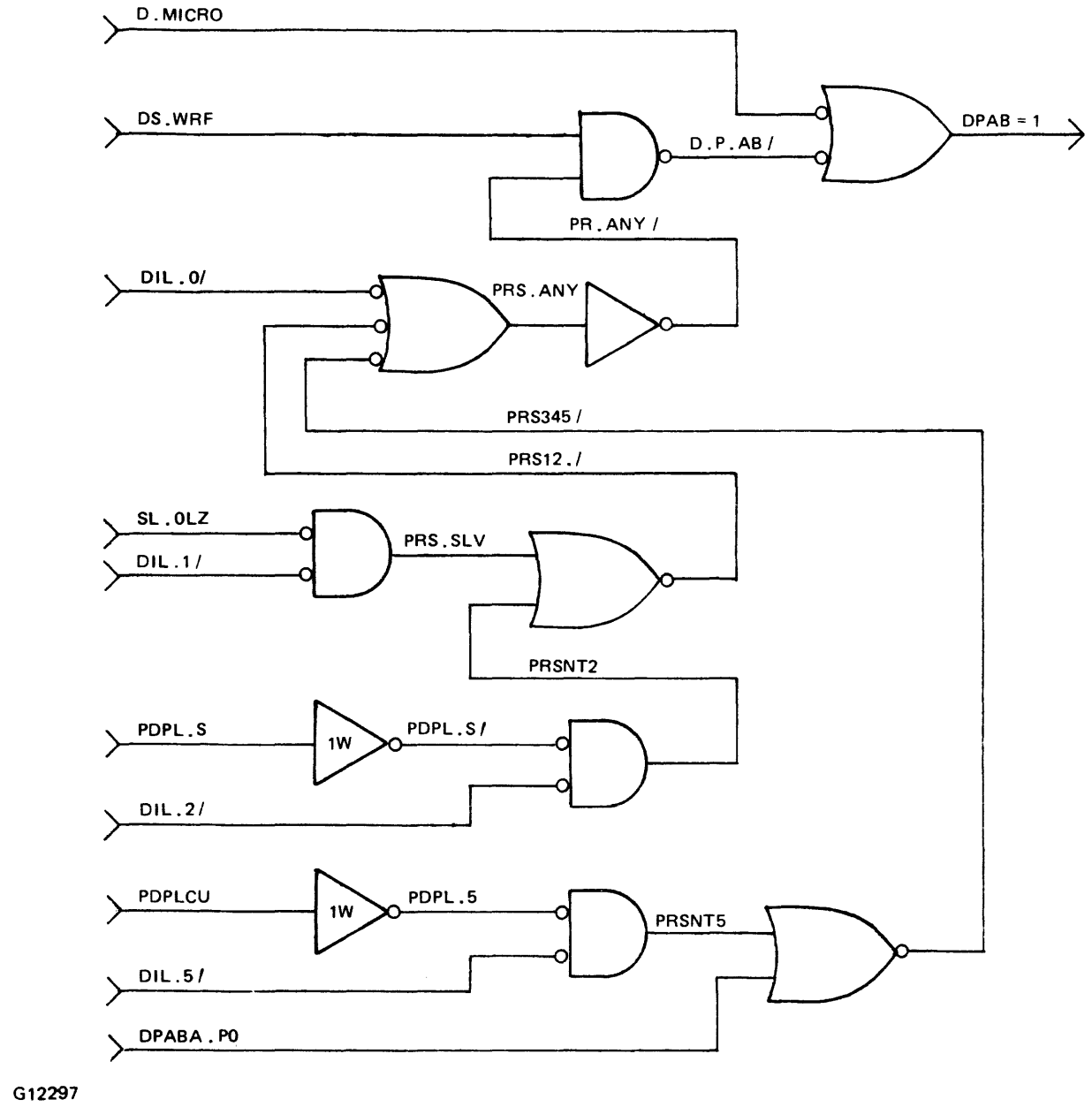


Figure 4-10. Port Absent Detection

Source Port Detection

Source port detection is accomplished by a priority encoder (ENP0), a shift register (RSS0), and tri-state buffering that detect either Processor A or Processor B as the master processor (MA.B.Q/.).

Refer to Figure 4-11 for an illustration of the source port detection logic.

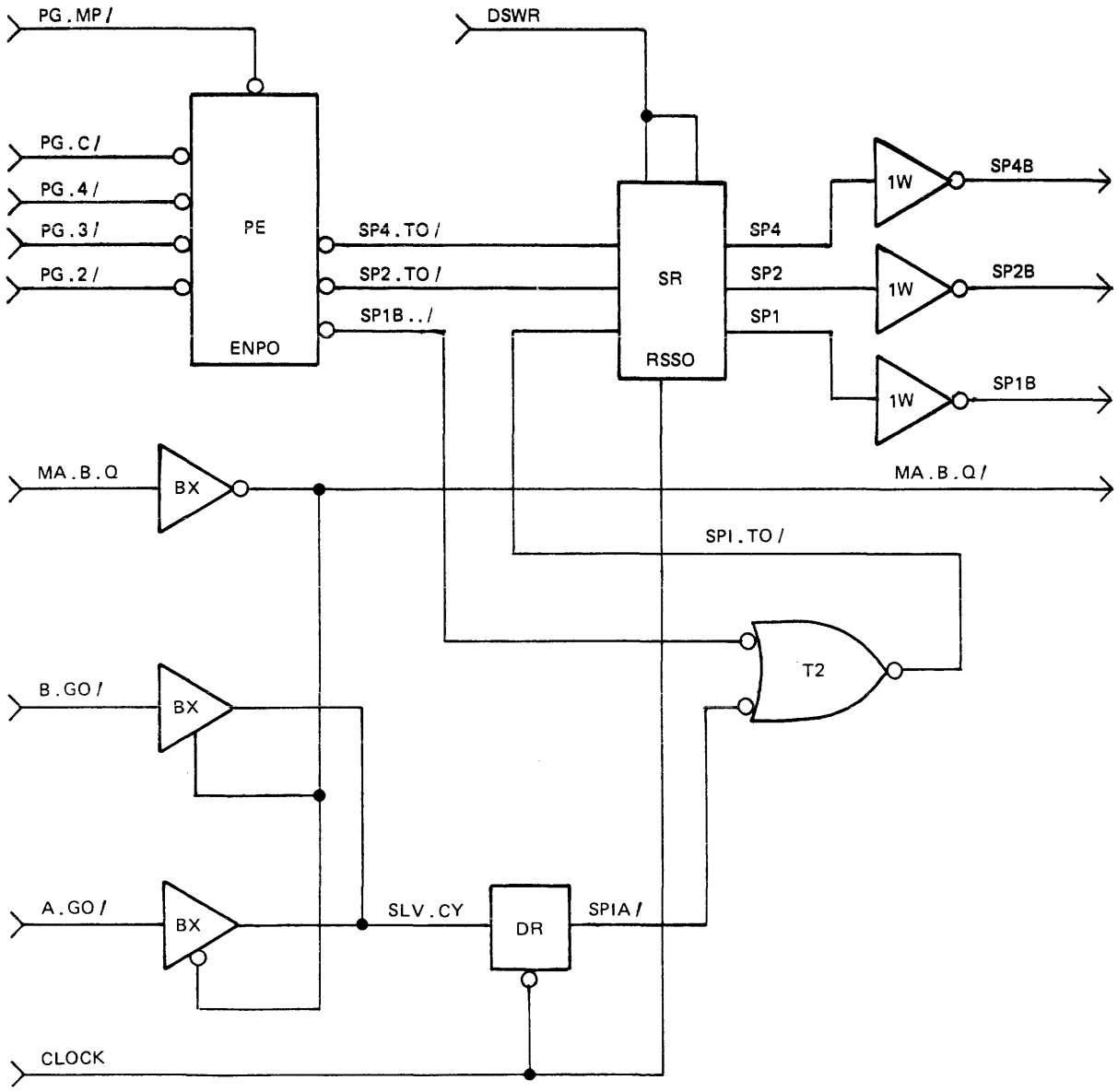


Figure 4-11. Source Port Detection

During a Dispatch Write operation, the priority encoder issues a binary equivalent of the highest input of the four inputs (port 2 through port 5) and the shift register holds the value of the source port decode. The term MA.B.Q/. (B is Master) enables either Processor A or Processor B as source port information (SLV.CY/.) and ANDs the clocked output (SP1A../.) with SP1B../. from the priority encoder (SP1.T0/.) as an input to the shift register.

If Processor B is the master processor and Processor A is dispatching, the output from the priority encoder is seven (111). The term A.GO../., which is LOW, is enabled and clocked through and ANDed as a zero. The terms SP4.T0/., SP2.T0/., and SP1.T0/. are gated through the shift register as a six (110) and inverted to reflect the source port as port 1 (001).

This logic selects the master processor as port 0 and the slave processor as port 1. Also, the multiline control ports are assigned as port 2 through port 5.

Diagnostic Logic

When one of the diagnostic operations is issued by the processor, mode bits MD0, MD1, MD2, and MD3 are decoded as a diagnostic enable (PI.DIAG.). This allows decoder DC30 to decode micro bits MC0, MC1, and MC5 into four Echo commands and three Read commands. By means of the data path logic, these seven commands steer data to or from the ports.

The diagnostic operations that can take place, depending on the selected variants in the 11D microinstruction, are defined in Table 4-6.

Table 4-6. 11D Micro Operations affecting HA-3

Micro Bits							Mode Bits				
5/	4/	3/	2/	1/	0/	REQ/	3/	2/	1/	0/	Operation
0	0	0	1	1	1	0	0	0	1	0	Echo latch
0	0	0	1	1	0	0	0	0	1	0	Echo Port 2
0	0	0	1	0	1	0	0	0	1	0	Echo Port 3
0	0	0	1	0	0	0	0	0	1	0	Echo Port 4
1	0	0	1	1	1	0	0	0	1	0	Read Latch
1	0	0	1	1	0	0	0	0	1	0	Read PA in
1	0	0	1	0	1	0	0	0	1	0	Read FALSE level
1	0	0	1	0	0	0	0	0	1	0	Read tri-state S-bus

Figures 4-12 through 4-20 illustrate the various paths selected by the different diagnostic commands.

MICRO: 0B1A, 0B5A, 0B9A and 0BDA

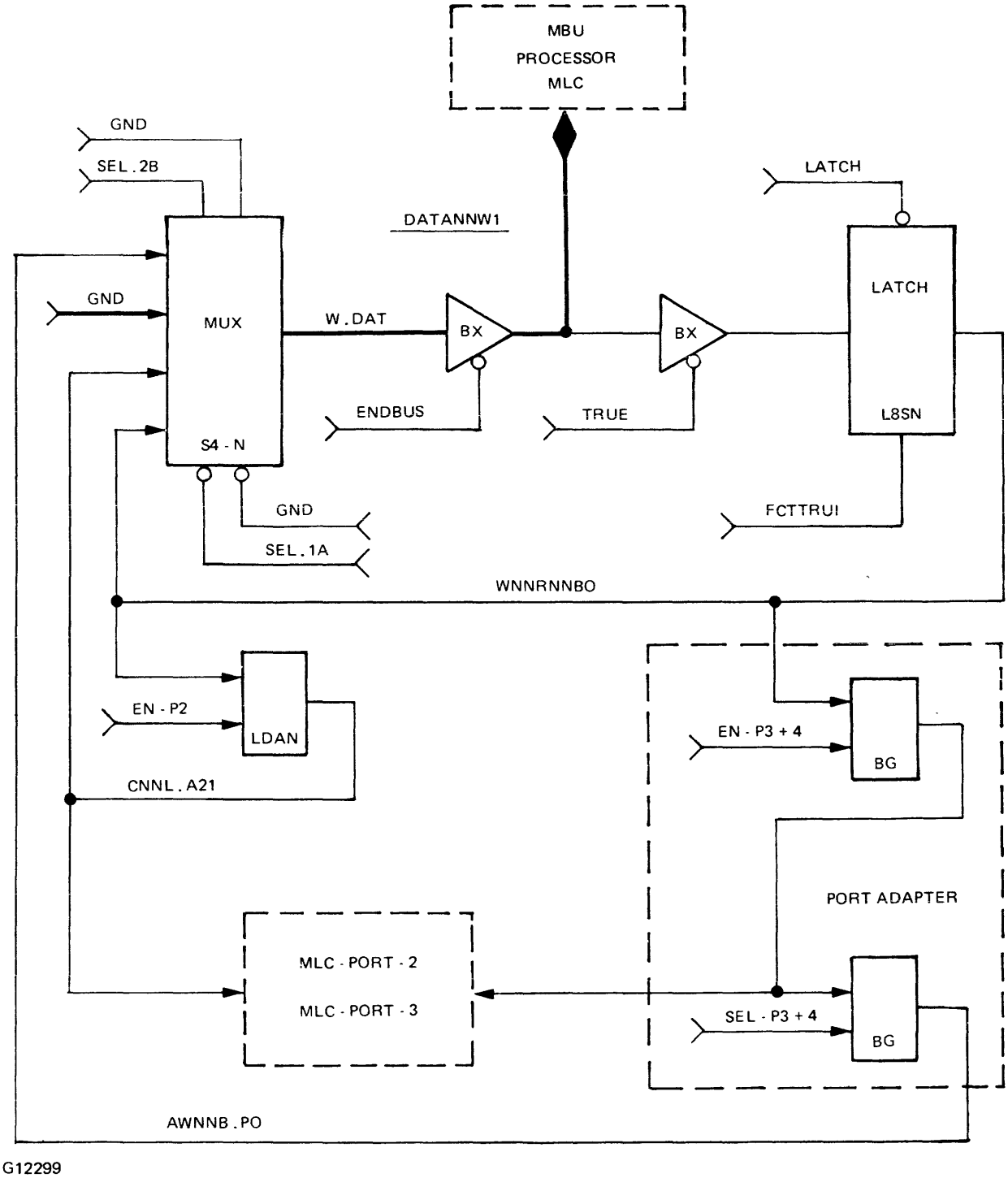
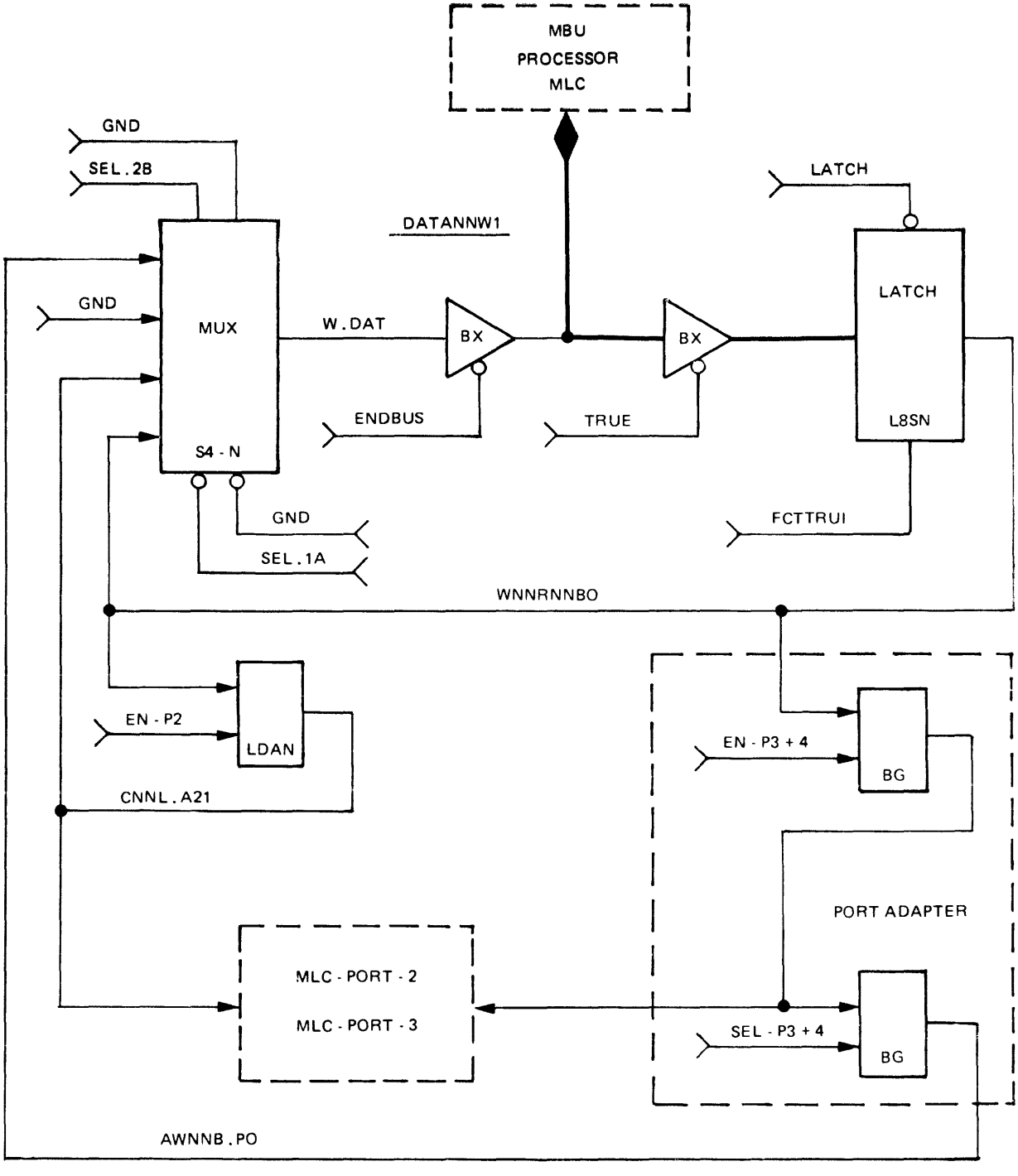


Figure 4-12. Diagnostic Read FALSE Level (data path)

MICRO: 0B(3+7+B+F)(8+9+A+B)



G12300

Figure 4-13. Diagnostic Echo Write Data (all variants)

MICRO: 0B(3+7+B+F)8 or 0B(1+5+7+D)8

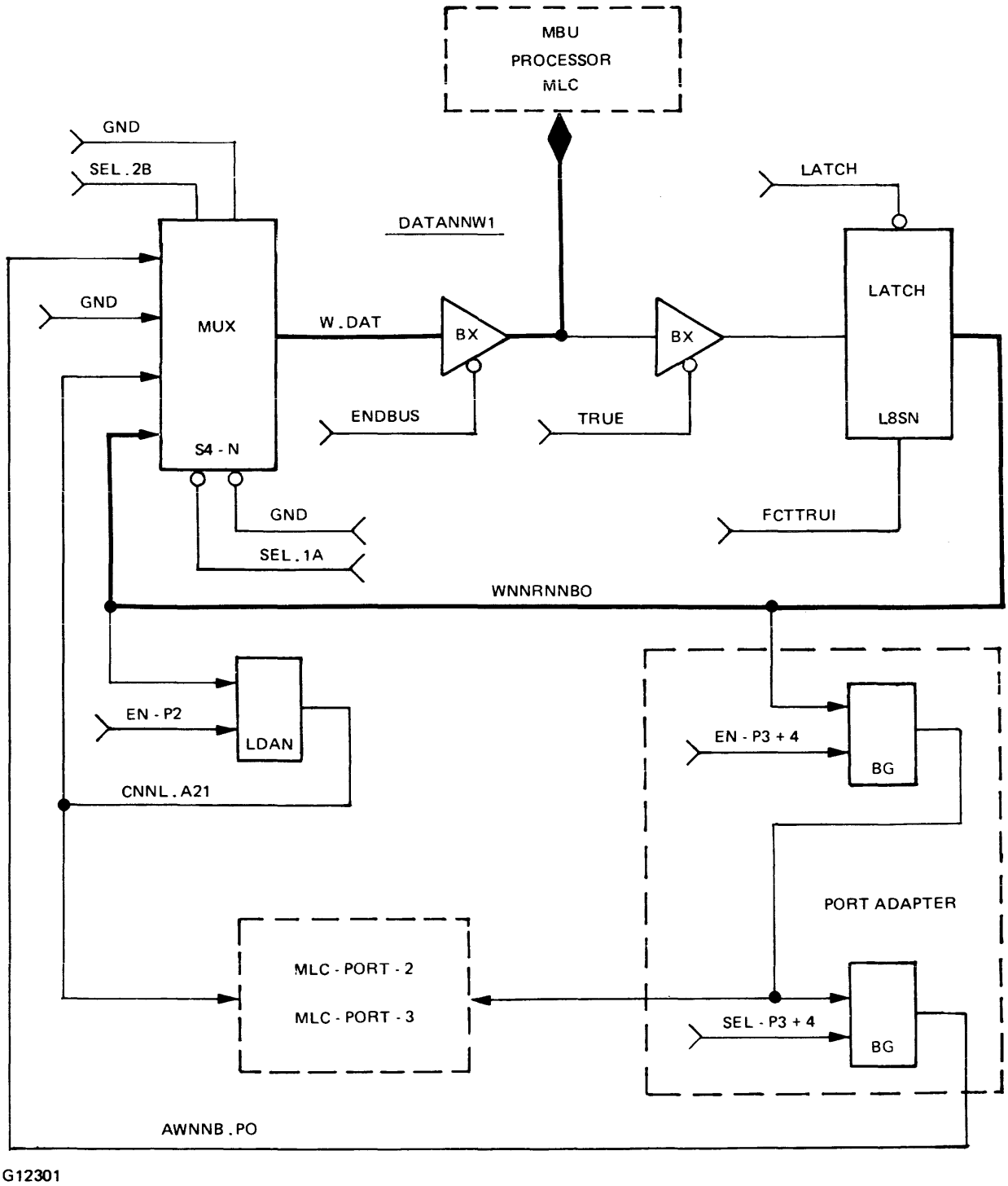


Figure 4-14. Diagnostic Read or Echo Latch

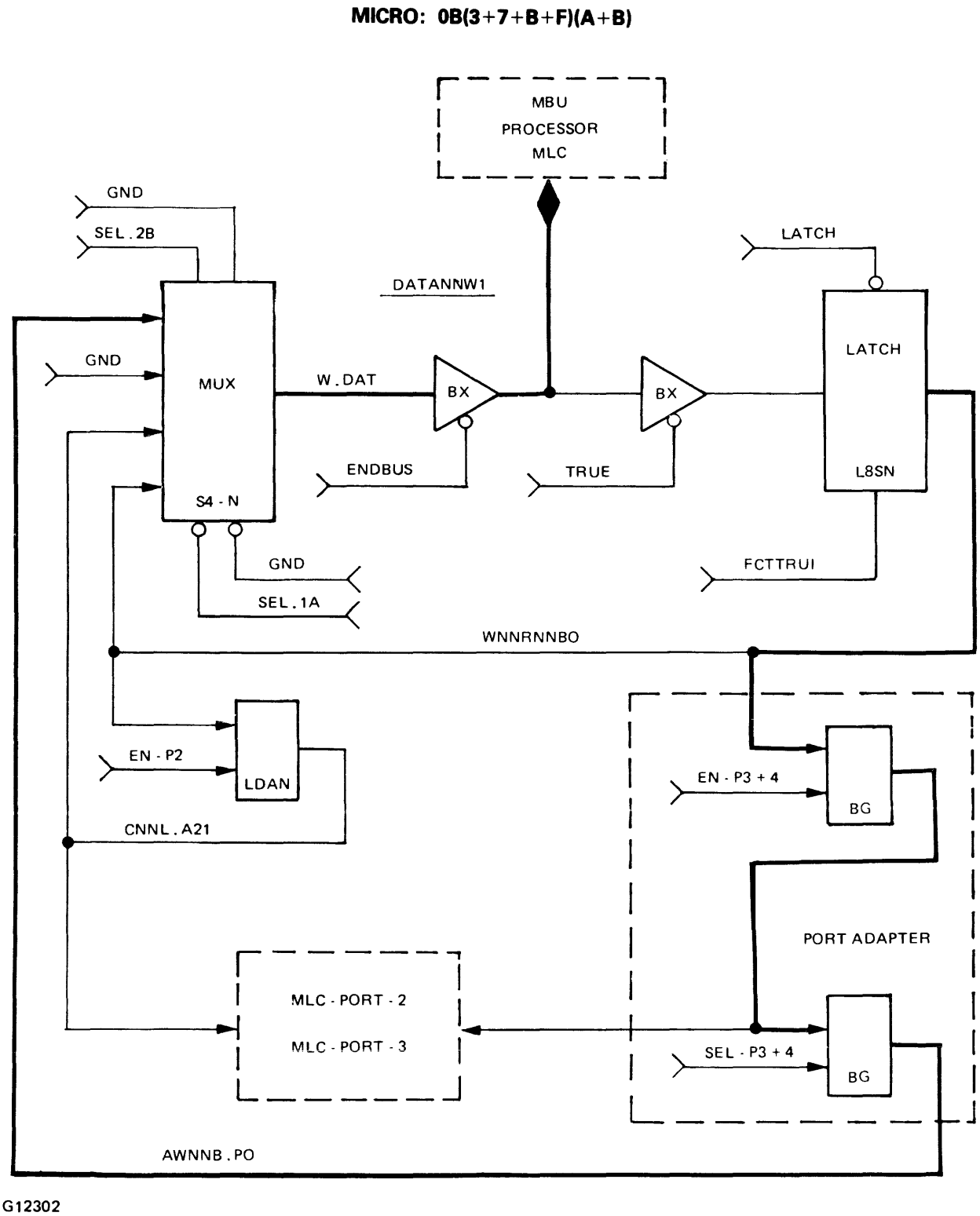


Figure 4-15. Diagnostic Echo Port 3 or 4 (PA-1)

MICRO: 0B(3+7+B+F)9

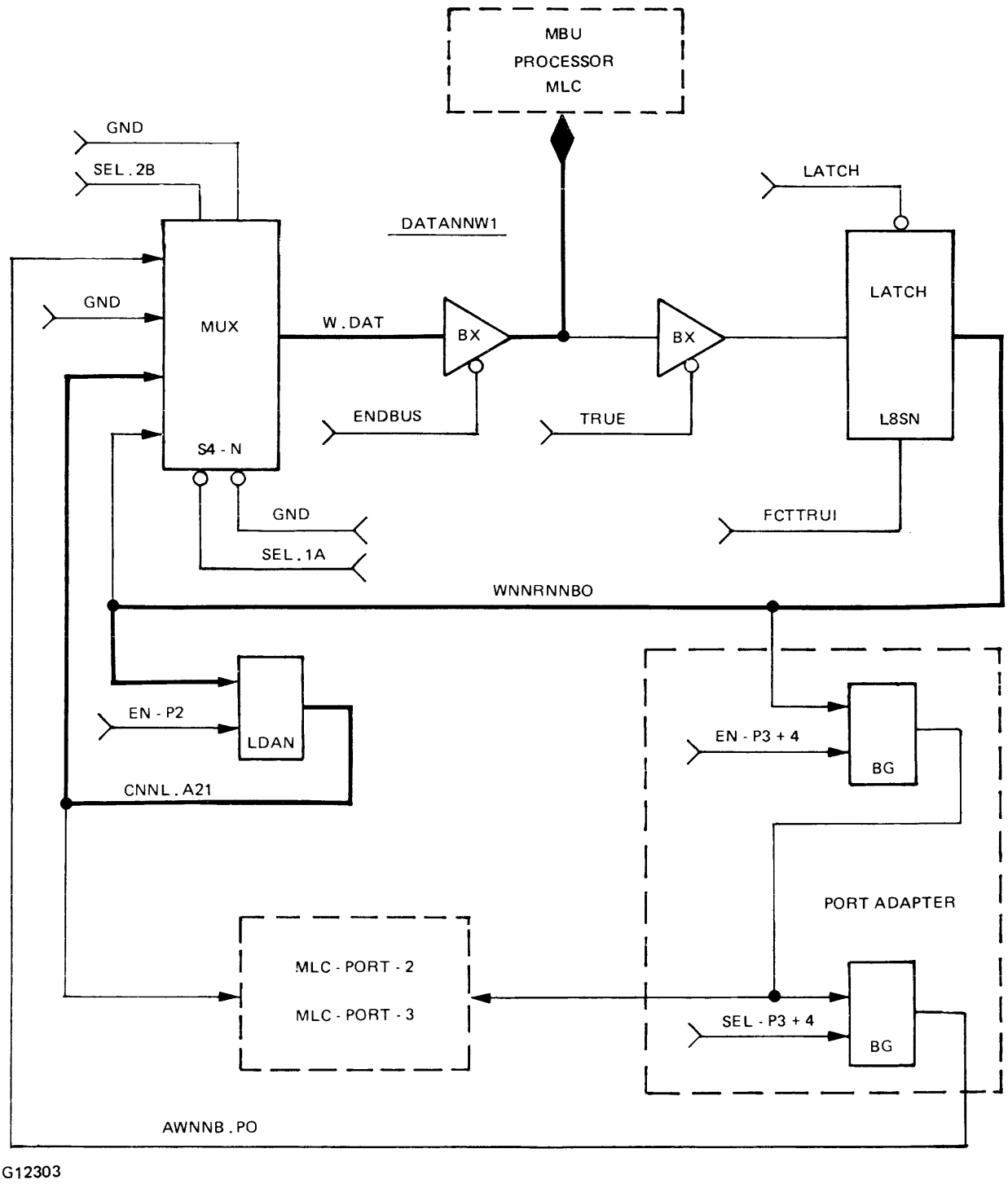


Figure 4-16. Diagnostic Echo Port 2

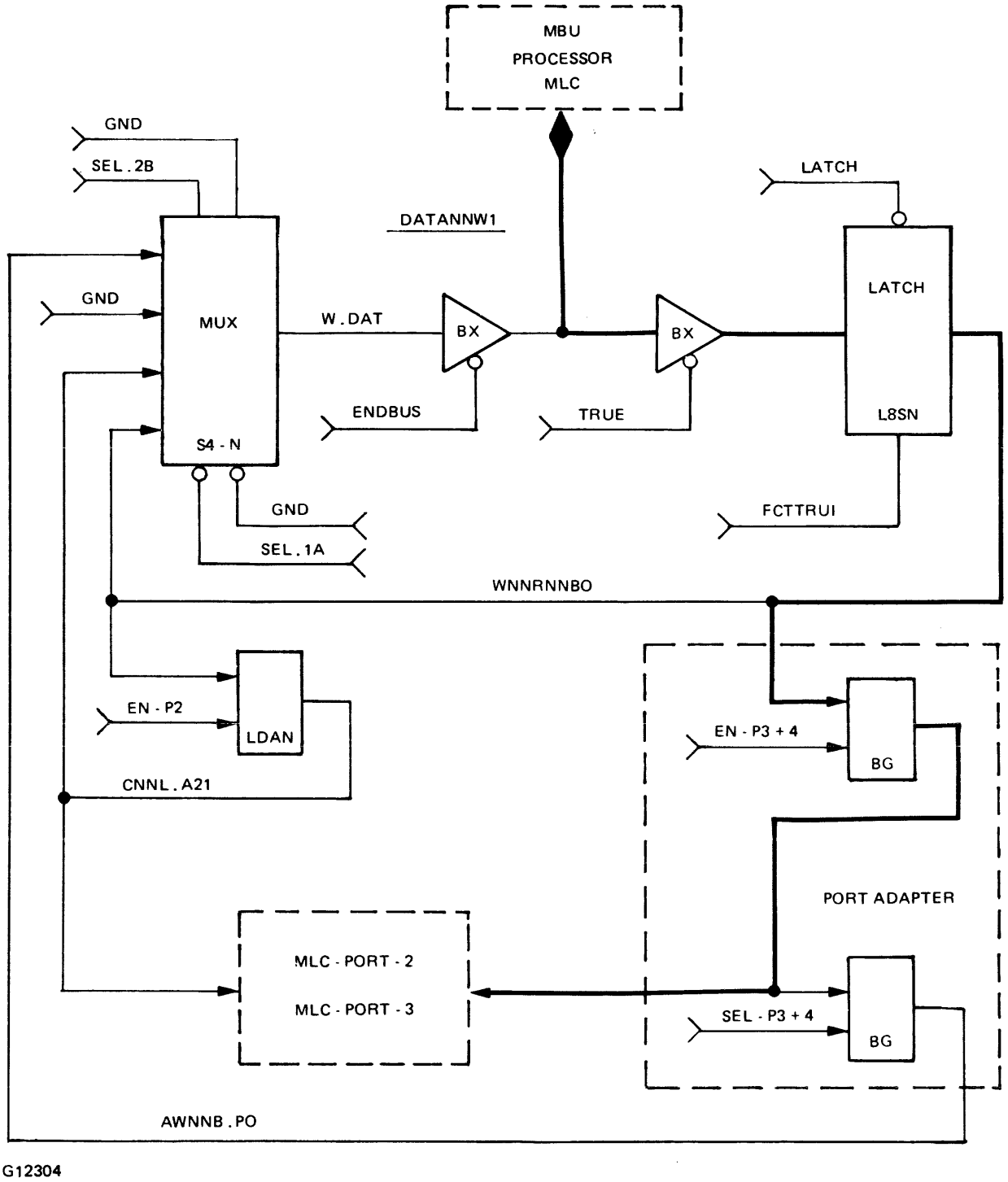


Figure 4-17. Memory Read Data to Port 3 or 4 (PA-1)

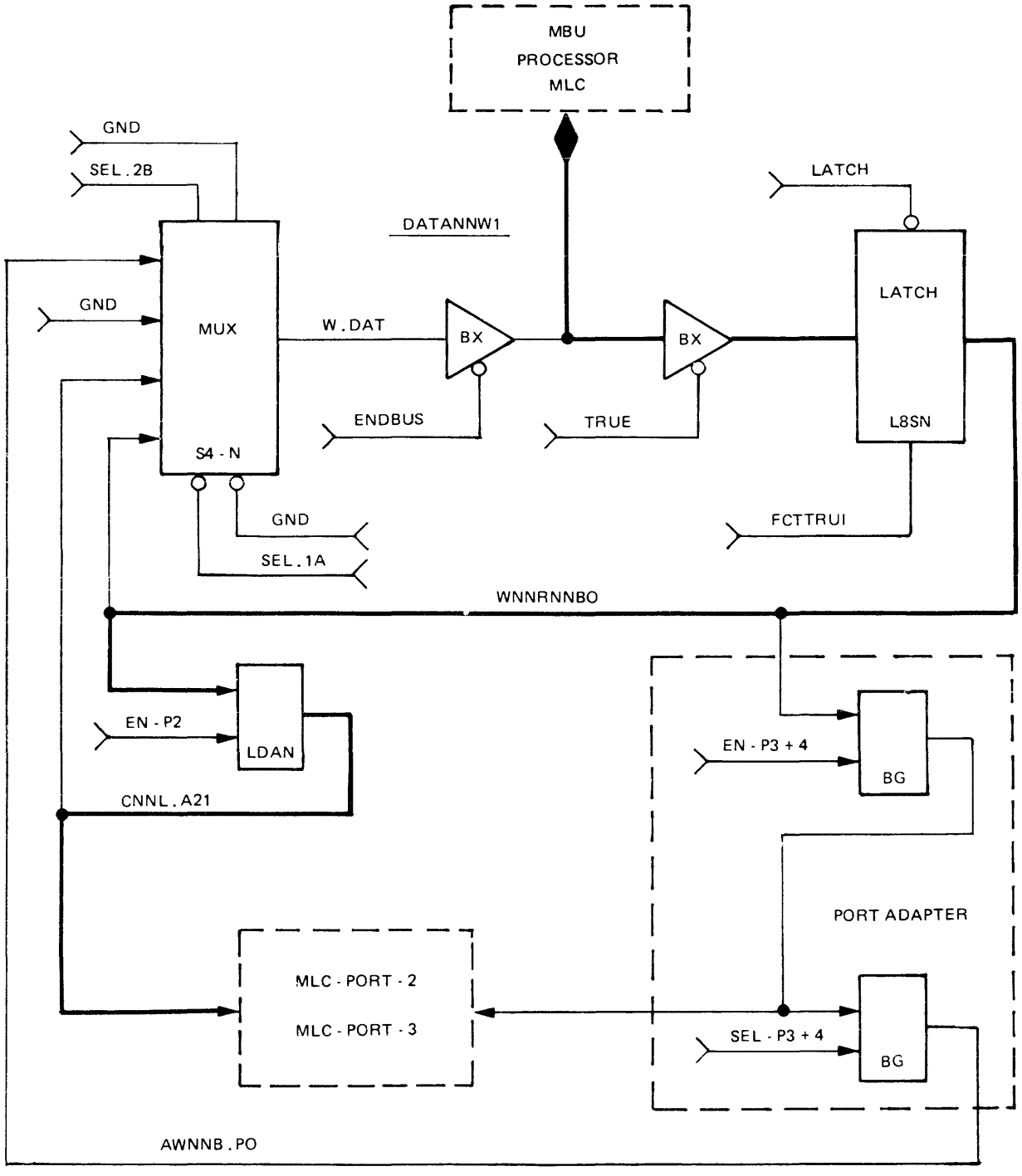


Figure 4-18. Memory Read Data to Port 2

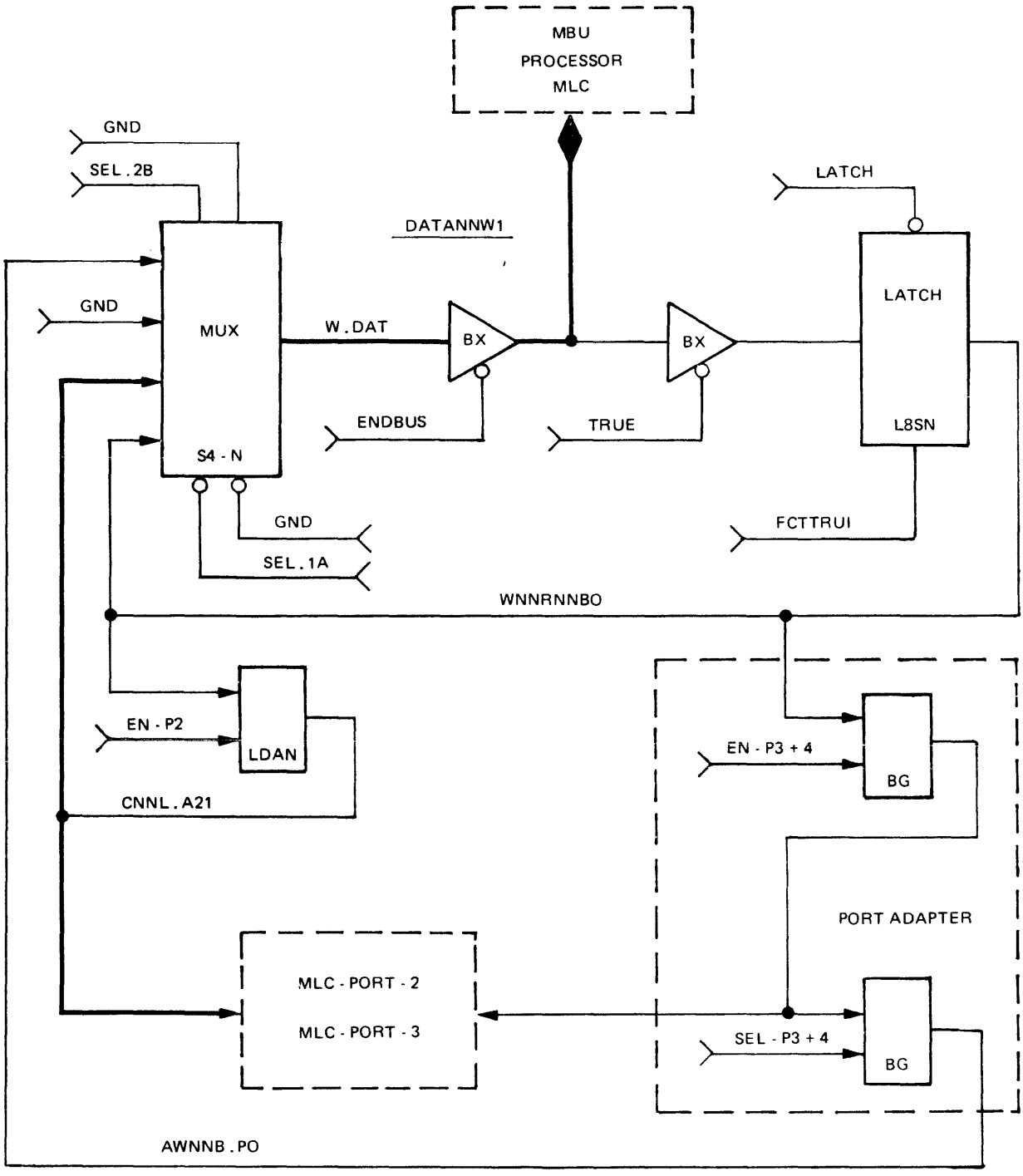


Figure 4-19. Address and Write Data to MBU from Port 2

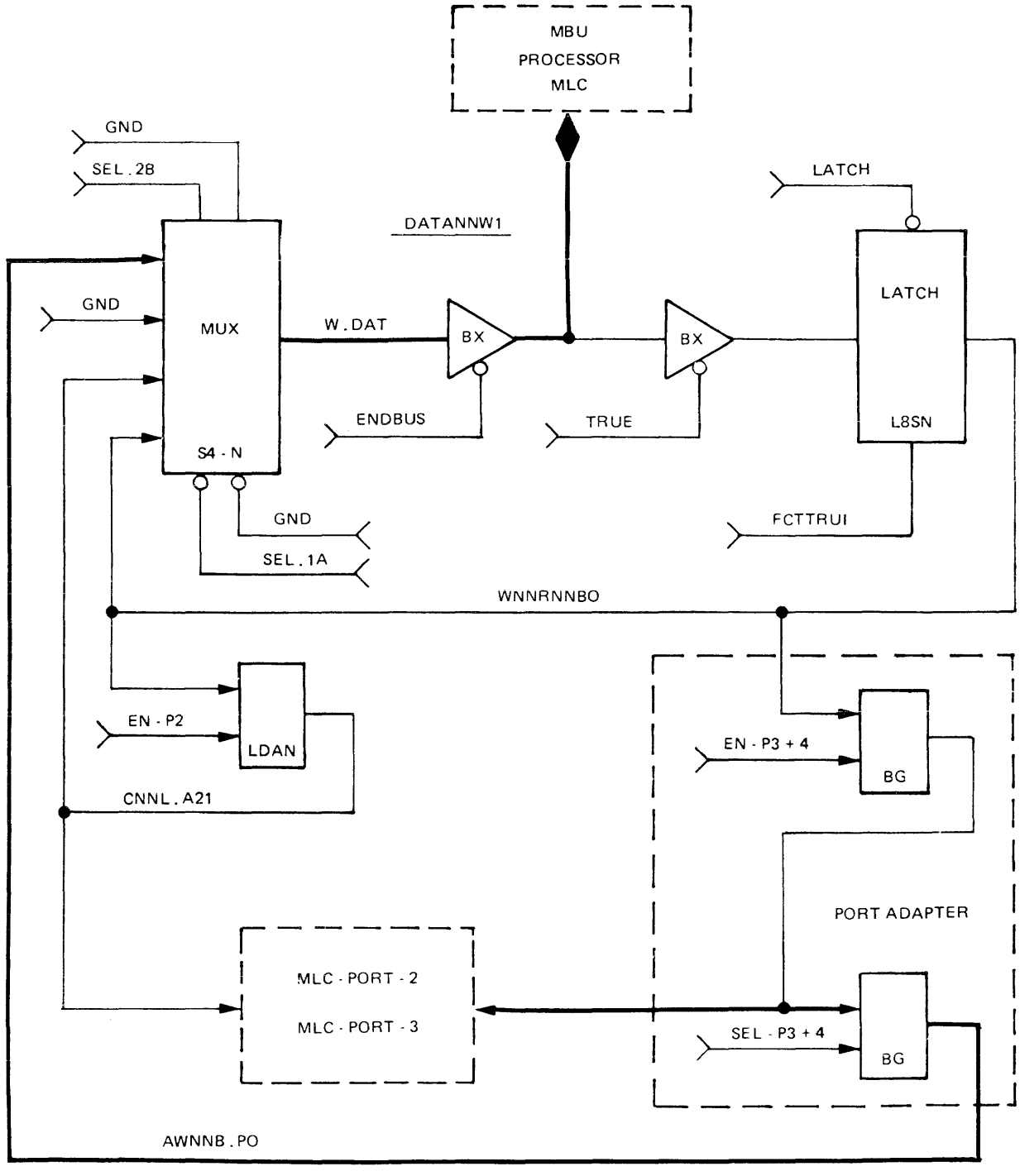


Figure 4-20. Address and Write Data to MBU from Port 3 or 4

APPENDIX A
GLOSSARY OF TERMS

A.....a.
A-data. (Control signal; a = A or B.) Specifies when A-data should be passed through by the ALU.

A+B...a.
A-data plus B-data. (Control signal; a = A or B.) Specifies when A-data plus B-data should be performed by the ALU.

A–B...a.
A-data minus B-data. (Control signal; a = A or B.) Specifies when A-data minus B-data should be performed by the ALU.

A–B–1.a.
A-data minus B-data minus 1. (Control signal; a = A or B.) Specifies when A-data minus B-data minus 1 should be performed by the ALU.

A–B/..a.
A-data minus B-data NOT. (Control signal; a = A or B; / specifies active LOW.) Specifies when A-data minus B-data complement should be performed by the ALU.

A-1...a.
A-data minus 1. (Control signal; a = A or B.) Specifies when A-data minus 1 should be performed by the ALU.

A/....a.
A-data NOT. (Control signal; a = A or B; / specifies active LOW.) Specifies when A-data should be complemented by the ALU.

ABMXSnC0
A-data or B-data to MEX. (Control signals; n = 0 or 1.) Select the card functions to be output to the MEX.

ADD...E.
Add. (Control signal.) Used by the ALU on card E.

ADDBCDE.
Add BCD. (Control signal.) When TRUE, data is modified for binary coded decimal addition.

ALUnn.a.
ALU output. (Data path; nn = 00 thru 23, a = A or B or E.)

ALUMO.E.
ALU mode. (Control signal.) Specifies the type of function to be performed in the ALU.

AMXSnna.
Card A to MEX, (Control signal; nn = 01, 02, 11, or 12; a = A or B.) Selects card functions and enables them to the MEX.

APADnn*.
A-pad data. (nn = 00 thru 23.) Scratchpad A-data output lines.

APADDnA.
A Pad address. (Control signals; n = 0 thru 3.) Scratchpad A address lines.

AU>FAnA.
AU replaces FA. (Control signal; n = 0 = MEX, n = 1 = ALU.) Selects AU (meaning: ALU or MEX) as input to the FA register.

AUASn.C0
Arithmetic (logic) unit. (Control signals; n = 0 thru 4.) Control of ALU functions.

A0SLn...
A0 replaces A1. (Control signals; n = 0 or 1.) Select one of four data inputs to the A0 register.

A0>A1...
A0 replaces A1. (Control signal.) Moves A0 register data to A1 register.

A1nn....
A1 register. (Data lines; nn = 00 thru 23.)

A1=SW...
A1 equals switches. (Control signal.) Goes TRUE when A1 register contents are equal to the value entered in the console switches.

B.....a.
B-data. (Control signal; a = A or B.) Specifies when ALU should complement B-data.

B/....a.
B-data NOT. (Control signal; a = A or B; / specifies active LOW.) Specifies when ALU should pass B-data complement.

BIASnnB.
Bias. (Control signal; nn = 00 thru 23.) Output of bias logic; input to CPL.

BBIT..W1
B bits. (nn = 00 thru 23.) Interface lines to S-Memory.

BCDan.E.
BCD. (Control signal; a = A thru F; n = 1 or 2.) Provides carry correction for binary coded decimal addition.

BCDADDE.
BCD add. (Control signal.) When TRUE, data is modified for BCD addition.

BCDOUTE.
BCD out. Binarily coded output from the ALU.

BDATnna.
B-data. (Data path, nn = 00 thru 23; a = card A or card B.) Data path to ALU.

BMUXOEC2
Card B multiplexor. (Control signal.) Enables data from card B to the MEX.

BPADnn*.
B-Pad. (Data lines; nn = 00 thru 23.) Data lines for Scratchpad B.

BPADDnB.
B-Pad address. (Address line; n = 0 thru 4.) Address lines for Scratchpad B.

BR....A.
BR register. (Control signal.) Specifies when BR register is selected as a sink.

BRnn....
Branch. (Data lines; nn = 04 thru 23.) Data lines from the Fast Branch Adder on card J to the A0 register.

BRSn....
Branch select lines from card H.

BRTPOEC2
BR or TEMP. (Control signal.) Used to select BR, LR, or TEMP register.

BSKIP/B0
Bias skip. (Control signal.) Used to specify when to use bias skip.

BSUMnnE.
B-data sum. (Data lines; nn = 00 thru 23.) Data lines from ALU after gating with the mask generator.

BYFCP.B.
Bias by F and CP. (Control signal.) Used to specify when to bias by the F and CP registers.

BYS/..B.
Bias by S. (Control signal.) Used to specify when to bias by S.

BYUNT/B.
Bias by unit. (Control signal.) Used to specify when to bias by unit.

C-RCLK..
CA or RC clock. Used when sending or receiving Command Active or Response Complete signals.

Cn....E.
Carry level. (Logic; n = 1 thru 3.) Used to extract CYL.

CA.....
Command active. (Control signal.) Indicates that processor to I/O data transfer is in process.

CA.CLKK0
Cache clock. (Control signal.) Used to clock in Cache data.

CACFIL..
Cache fill. (Control signal.) Used when loading Cache memory.

CACHE...
Cache. (Level.) When TRUE, indicates that the micro source is Cache.

CARnn/a.
Carry. (Control signal; nn = 04, 08, 12, 16, or 20; a = A thru C.) Carry output from each four bits of the ALU.

CAREGnF.
CA register. (Data bits; n = 0 thru 3.) CA register data bits.

CBREGnF.
CB register. (Data bits; n = 0 thru 3.)

CLRB...0
Clear bus. (Level.) Goes HIGH when CLEAR is pressed.

CLRCA.G0
Clear Cache. (Control signal.) When TRUE, resets Cache Key storage validity bit.

CLRK....
Clear K. (Control signal.) Produced by GPCLR.GO. Clears all counters. Also used to generate slow clocks.

CLRTM/G0
Clear timer. (Control signal.) Clears utility timer.

CMInn...
Console M register. (Data lines; nn = 00 thru 14.)

CN/...B.
Carry input NOT. (/ specifies active LOW.) Carry input to the ALU.

CNTL.aW1
S-Memory. (Control lines; a = A thru G.) S-Memory control lines.

CPnn..E.
CP register. (Data lines; nn = 00 thru 05.) CP register data lines.

CPLn....
CPL register. (n = 0 thru 4)

CPRGnnB.
CP register. (Data lines; nn = 00 thru 07.) CP register output data lines.

CRKPE/G0
Clear Key parity error. (Control signal.) Clears the Cache Key parity error latch.

CSJMHT..
Console switch jam halt. (Control signal.) Goes TRUE and halts the processor when A1 equals the console switches.

CSWnn..1
Console switch. (Data lines; nn = 00 thru 23.) Console switch data lines.

CYF...E.
Carry flip-flop. (Level.)

CYL...ED
Carry level. (Level.) TRUE when a carry-out from the ALU occurs.

DATA/.D0
Data NOT. (Control signal; / specifies active LOW.) Enables ENCARCD2 which selects CA or RC.

DATA>SA2
Data replaces S-Memory. (Control signal.) Specifies when data is to be sent to S-Memory. (FALSE enables, TRUE disables)

DISM..G0
Disable M. (Level.) TRUE disables M register data to the MOP bus.

DISPLYA1
Display. (Level.) Enables Console Lamp register.

DSA54.C0
Disable address lines 5 and 4. (Level.) Disables address lines during Load Cache Word operation.

DSCPa...
Distribute System Clock pulse. (Control signal; a = A thru J.) Clock.

DSUMnn*.
D sum. (Data lines; nn = 00 thru 23.) Data lines are the corrected BCD sum/diff when doing BCD arithmetics.

END3F.E0
End 3F. (Control signal.) Informs control when the Normalize X (3F) micro is complete.

ENLDMAG0
Enable load M. (Control signal.) Enables loading of the M register.

ENOR.MG0
Enable OR M. (Control signal.) Enables ORing of the M register.

EN4B/.E.
Enable 4-bit bus. (Control signal.) Enables the multiplexors providing the 4-bit register output.

ERRLMCP1
ERROR lamp. (Control signal.) Lights the ERROR lamp.

EXnn.XT0
Exchange. (Data lines; nn = 00 thru 23.) Bidirectional data lines that interface processor with soft I/O controls.

FARGnnA.
FA register. (nn = 00 thru 23.)

FASNKnA.
FA sink. (Control signal; n = 1 or 2.) Enables FA register.

FBnn..B.
FB register. (nn = 16 thru 23.) FB register drive lines.

FBADP/G0
Force bad parity. (Control signal.) Used to test parity logic

FBInSnB.
FB register in. (n = 1 or 2.) Specifies the inputs to FB.

FBLITnF.
FB literal. (Control signal; n = 0 thru 3.) Used for masking.

FBRGnnB.
FB register. (Data lines; nn = 00 thru 23.) Output data lines from FB register.

FCMISS.1
Force miss. (Control signal.) Force the term MISS.... to go TRUE.

FETCH...
Fetch. (Control signal.) Gets the next micro.

FHS...C.
Fetch S. (Control signal.) When TRUE, causes a micro fetch from S-Memory.

FHU.....
Fetch U. (Control signal.) When TRUE, causes a fetch from U register.

FLnn..B.
FL register. (Data lines; nn = 00 thru 23.) FL register data lines.

FLDOWNB.
FL down. (Control signal.) Prevents FL from going negative.

FLESFLB.
FL register equal to SFL. (Control signal.)

FLGE24B.
FL register greater than or equal to 24. (Control signal.)

FLGSFLB.
FL greater than SFL. (Control signal.)

FLLSFLB.
FL register less than SFL. (Control signal.)

FLL24.B.
FL register less than 24. (Control signal.)

FLNEQ0B0
FL register not equal to zero. (Control signal.)

FLNSFLB.
FL register not equal to SFL. (Control signal.)

FMUXSnD0
F multiplex select. (Control signal; n = 0 or 1.) Selects the output for card F.

FREEZE/.
Freeze. (Level; / specifies active LOW.) Micro source is the M register; micro in M is fetched and executed repeatedly.

FWRDn/H0
Force word bit number. (Control signal; n = 0 or 1.) Used to increment word number in A register in FHS or Cache Write (7E) micro.

GAP....1
Gap. (Control signal.) Cassette tape gap.

GPCLR.D1
General processor clear. (Control signal.) When HIGH, certain registers are cleared.

HALTRQC0
Halt request. (Control signal.) Protection halt.

HIT.....
Hit. (Control signal.) When TRUE, a valid Cache Key compare has occurred.

HJMXSnG0
H, J card to MEX. (Control signal; n = 0 or 1.) Selects H and J card outputs.

I/OCLK..
Input/output clock. 4-MHz or 6-MHZ clock to the I/O subsystem.

INCA0/G0
Increment A0 register. (Control signal; / specifies active LOW.) When LOW, this signal increments the A0 register.

INFBn.B.
Inhibit FB. (Control signal; n = 1 or 2.) Zeros out the input to FL when FL attempts to underflow. (n = 1 or 2)

INHDEC..
Inhibit decode. (Control signal.) Disables the output of the decode PROMs.

INHEXC..
Inhibit MEX. (Level.) When TRUE, this level inhibits the Main Exchange.

INPDnna.
Input D. (Control lines; nn = 00 thru 23, a = card A or card B.) Input to Scratchpad from a temporary register (TEMP).

INT-ORC0
Interrupt OR. (Control signal.) All interrupts ORed.

INTSW.G1
Interrupt switch. (Control level.) Level from the INTERRUPT switch.

IOEXnn..
Input/output exchange. (Data lines; nn = 00 thru 23.) Data lines to an from soft I/O controls.

JAMHLT..
Jam halt. (Control signal.) Indicate a hardware failure; halts the system.

JMXOE/G0
Card J multiplexor output enable. (Control signal; / specifies active LOW.) When LOW, enables multiplexors on card J to output data to the Main Exchange.

KADnn/J1
Key address. (Address lines; nn = 06 thru 15.) Cache Key address lines.

KEYPAR..
Key parity. (Control signal.) Parity bit to be written into Cache Key memory.

KEYWE/..
Key Write. (Control signal.) Specifies when to write Cache Key data.

KINnn.J1
Key in. (Data lines; nn = 16 thru 23.) Data lines into Cache Key.

KOUTnn..
Key out. (Data lines; nn = 16 thur 23.) Data lines out of Cache Key.

KPARERH0
Key parity error. (Control signal) Indicates occurence of a Cache Key parity error.

LOAD..G0
Load. (Control signal.) When HIGH, enables loading of the A0 register.

LCPLnna.
Literal or CPL. (Data lines; nn = 00 thru 04, a = A or B.) Data lines for literal or CPL.

LDAn..G0
Load A. (Control signal; n = 0 or 1.) Load registers.

LDBRLRC2
Load BR, LR. (Control signal.) Used with LDFALRC2 to load the BR or LR registers.

LDFALRC2
Load FA, LR. (Control signal.) Used with LDBRLRC2 to load FA or LR registers.

LDM/.GB0
Load M. (Control signal; / specifies active LOW.) When LOW, enables loading of the M register.

LMASKnE.
Length mask. (Control signal; n = 0 thru 4.) Originates from CP; specifies mask length.

LMPCK/..
Lamp clock. (Control signal; / specifies active LOW.) Control signal for the Console Lamp register.

LN.....
Last nano. (Control signal.) Indicates end of the micro being executed.

LNNDC/C0
Last nano, nano decode complete. (Control signal; / specifies active LOW.) Indicates that the current micro is complete and that it is time to advance the pipeline.

LPEN/.A.
Literal or CPL enable. (Control signal; / specifies active LOW.) Specifies when literal or CPL is selected for the B-data lines.

LR....A.
LR register. (Control signal.) Specifies that LR register is sink.

LREGnnF.
L register. (Data lines; nn = 00 thru 23.) L register data lines.

LRTPOEC2
LR or TEMP output enable. (Control signal.) Used to enable LR or TEMP register to the Main Exchange.

LS#n/.*.
Last sequence. (Control signal; / specifies active LOW; n = 0 thru 5.) Output from PROMs; indicates how many sequences are required for execution of a micro.

LSUX..E.
Least significant unit of X register. (Control signal.)

LSUY..E.
Least significant unit of Y register. (Control signal.)

MASKnn*.
Mask. (nn = 00 thru 23.) Output of the mask generator.

MASKNnC0
Mask. (n = 0 thru 4.) Specifies amount to increment or decrement registers.

MASTERG0
Master. (Control signal.) Sets bit 3 in MSSW register.

MAXSnnA.
Maximum S-Memory. (Hardware constant; nn = 18 thru 23.) Specifies maximum size of S-Memory.

MBRnn.H1
M branch. (Data lines; nn = 00 thru 11.) Micro branch literal data lines.

MEX<F/D0
MEX is replaced by F. (Control signal; / specifies active LOW.) When LOW, card F data is enabled to the Main Exchange.

MEX<FBF.
MEX is replaced by FB. (Control signal.) FB data is enabled to the Main Exchange.

MEXnnBT0
Main Exchange. (24-bit bus; nn = 00 thru 23.) The processor's primary data path.

MISS....
Miss. (Level.) Indicates that the Cache Key and A register did not compare.

MOPnnT0
Micro-operator lines. (16-bit bus; nn = 00 thru 15.) The 16-bit bus from the M register.

MPARERH0
M parity error. (Level.) When TRUE, indicates that there is a microinstruction parity error.

MRnn....
M register. (Control signal; nn = 00 thru 15.) Output of the M register.

MSBX..E.
Most significant bit of X. (Control signal.) The most significant bit of the X register, specified by CPL.

MSOURCG0
M source select. (Control signal.) When TRUE the Main Exchange is the source for M; when FALSE, Cache is the source. (Normally FALSE.)

MSSWn.C.
Micro source switch. (Register; n = 0 thru 3.) A 4-bit register that indicates micro source or master/slave state.

MX<Ea.E.
MEX replaced by E. (Control signal; a = L. M, or U.) Causes the L, M, or U register on card E to be output to the Main Exchange.

MX<4B/F.
MEX replaced by 4-bit bus. (Control signal.) Causes the 4-bit data bus to be output to the Main Exchange.

MX+Unn..
MEX or U. (Data lines; nn = 00 thru 15.) Main Exchange or micro storage data lines.

MX>IO...
MEX replaces I/O. (Control signal.) Gates Main Exchange to the I/O bus.

MXnn....
MEX. (Data lines; nn = 00 thru 23.) Main Exchange data lines that are gated to the Console Lamp register.

NDC.....
Nano decode complete. (Control signal.) Specifies completion of the nano decoding process.

NORM..E.
Bias. (Control signal.) Specifies that the Normalize X (3F) micro is in process.

NORMAL/.
Normal. (Level; / specifies active LOW.) Indicates that the micro source is Cache unless there is a miss.

NOWRT/C1
No write. (Control signal; / specifies active LOW.) Terminates an S-Memory Write operation.

ONLINEG0
On line. (Control signal.) Specifies when processor is on-line in dual-processor environment.

PADS8.K0
Pad source. (Clock.) Used for read/write control of the Scratchpad.

PADWE/a.
Pad write enable. (Control signal; a = A or B.) Controls Scratchpad Read and Write functions.

PERMn.C.
Perm. (Register; n = 0 thru 3.) A 4-bit register that indicates memory-related errors.

PERPn.C.
Perp. (Register; n = 0 thru 3.) A 4-bit register that indicates processor-related errors.

POP/..G0
Pop. (Control signal; / specifies active LOW.) When LOW, initiates a pop of the TAS register.

PS#n....
Present status number. (Control signal; n = 0 thru 5.) Output from the status counter that indicates the nano number being executed.

PUSH/.G0
Push. (Control signal.) When LOW, initiates a push to the TAS register.

RC.....
Response Complete. (Control signal.) Indicates that the I/O control has accepted the data.

RCV4B/D0
Receive 4 bits. (Control signal; / specifies active LOW.) Routes the 4 bits of all 4-bit fields of card F.

RDADOBC.
Read address out of bounds. (Control signal.) When TRUE, sets CD register bit 1.

REFCLK..
Reference clock. (Clock signal.) Used to set up the system clock.

RELOG.D1
Read error log. (Control signal.) Used to indicate that the memory error register (ELOG) has changed.

RGINnna.
Register in. (Data lines; nn = 00 thru 23; a = A or B.) Input register data lines.

ROTCn.F.
Rotate control. (Control signal; n = 0 thru 4.) Used on rotator.

RUN...W1
Run. (Level.) Means that the processor is not in the MTR mode and the START push button has been pushed.

SCPMn..0
System clock pulse master. (Clocks; n = 0 thru 5.) I/O buffered clocks.

SELn..B.
Select. (Control signal; n = 1 or 2.) Controls the selection of parameters for the Bias (3E) micro.

SEL24.B.
Select 24. (Control signal.) Specifies when the Bias (3E) micro is to use 24 instead of the FL value.

SELENnA.
Select enable (Control signal; n = 0 or 1.) Enables 2-to-1 multiplexor of Scratchpad A or the ALU to the Main Exchange.

SELF..B.
Select F. (Control signal.) When FALSE (O), MEX is selected; When TRUE (1) FA is selected.

SELLITB.
Select literal. (Control signal.) Specifies when to use literal.

SELLITC0
Select literal. (Control signal.) When FALSE (0), Scratchpad A is selected; when TRUE (1) literal is selected.

SINKa.B.
Sink. (Control signal; a = A thru F.) Specifies which 4-bit field of FB to sink. (a = A thru F)

SLAVE...
Slave. (Control signal.) Sets Bit 3 in MSSW register to indicate this processor is the slave.

SLOWSO..
Slow source. (Control signal.) Indicates a register that needs more clocks to source.

SMEM/...
S-Memory. (Level; / specifies active LOW.) Indicates that the micro source is S-Memory only.

SMEMnnA.
S-Memory. (Data lines; nn = 00 thru 23.) Receive data lines from S-memory.

SNKPAD..
Sink Pad. (Control signal.) Specifies when Scratchpad is sink.

SPADDnC0
Scratchpad address. (Address lines; n = 0 thru 3.) Scratchpad address lines.

SPAWE.C2
Scratchpad A Write enable. (Control signal.) Controls writing of Scratchpad A. FALSE (0) enables Scratchpad A Write; TRUE (1) disables Scratchpad A Write.

SPWE/.C1
Scratchpad Write enable. (Control signal.) Enables writing to Scratchpads.

SR.....0
Service Request. (Control line.) General I/O service request line.

STALMPC1
State lamp. (Level.) Lights the STATE lamp on the D/M panel.

START...
Start. (Level.) Level from the START pushbutton.

STIEX/G0
Set inhibit execute. (Level; / specifies active LOW.) When LOW, disables the outputs of the nano register.

STKnn/..
Stack. (Data lines; nn = 00 thru 23.) Data lines out of stack memory.

STKADn..
Stack address. (Address lines; n = 0, 1, 2, 4, or 8.) Stack address lines.

STKCLKK0
Stack clock. (Clock.) Determines timing of the stack write enable.

SYSCLK..
System clock. (Clock.) System clock to the I/O controls.

TAPE..G1
Tape. (Level.) Specifies that micro source is cassette tape only.

TASnn...
Top of A-Stack. (Data lines; nn = 00 thru 23.) Data lines from the TAS.

TBUSn.Z0
T bus. (Data bus; n = 0 thru 6.) Bidirectional 7-bit data bus used during Dispatch operations for port and channel information.

TIMOUTD1
Time out. (Timer.) Monitors NXC/. An error is forced if there is no change within 600 ms.

TMEM/.A.
Temporary memory. (Control signal.) Enables temporary memory register.

TOP/.nA.
To processor NOT. (Control signal; / specifies active LOW; n = 1 or 2.) Specifies when to pass data to S-Memory. (n = 1 or 2)

TOPDnnB.
To Pad. (Data lines; nn = 00 thru 23.) Input data lines to holding register for Scratchpad B.

TPPERR.1
Tape parity error. (Control signal.) Indicates a tape parity error.

TR*MnnF.
T register and mask. Masked data from the T register.

TREGnnF.
T register. (Data lines; nn = 00 thru 23.) T register data lines.

TSTY/.C0
Test Y. (Control signal; / specifies active LOW.) Sets echo latch.

UDP.....
U data present. (Control signal.) Indicate that valid data is in the U register.

ULITn.A.
Use literal. (Control signal.) Specifies when to use literal.

UREGnn..
U register. (Data lines; nn = 00 thru 07.) U register data lines.

USMSK/D0
Use mask. (Control signal.) Specifies when to use mask data.

USTRnn..
Micro storage. (Output lines; nn = 00 thru 15.) Output from the Cache storage of micros.

VALID...
Valid. (Level.) Indicate when data in Cache Memory is valid.

WAIT....
Wait. (Control signal.) Suspends micro decoding.

WR/aa.C0
Write. (Control signal; aa = alpha characters.) Enables the writing of registers specified by aa.

WRKEY...
Write Key. (Control signal.) Specifies writing of Cache Keys.

WRMICRD0
Write micro. (Control signal.) Specifies the writing of Cache Memory.

X<YGPnE.
X less than Y. (Control signal; n = 0 thru 4.) X register is less than Y register.

X>EXCH..
X replaces by MEX. (Control signal.) Causes X data to replace Main Exchange data.

X>YGPnE.
X greater than Y. (Control signal; n = 0 thru 4.) X register is greater than Y register.

Xa....E.
X subregisters. (Control signal; a = subregisters A thru F.) X terms out of the ALU.

XEYCYFE.
X equals Y and CYF. (Level.) TRUE when X register equals Y register and Carry FF is TRUE.

XNEG0.E.
X not equal to 0. (Control signal.) X register bit 24 is TRUE, indicating that X contains a non-zero value.

XREGnnE.
X register bit. (nn = 00 thru 23.) X register bit, as specified by nn.

XSn...E.
S signals. (Control lines; n = 0 or 1.) Control lines for the X register.

Ya....E.
Y subregister. (Control signal; a = subregisters A thru F.) Terms out of the ALU.

YNEG0.E.
Y is not equal to 0. (Level.) Indicates Y register is not equal to zero.

YREGnn.E
Y register bit. (nn = 00 thru 23.) Y register bit, as specified by nn.

YSn...E.
Y signals. (Control lines.) Control lines for the operation of Y register.

1ERD/.D0
1E Read Dispatch. (Control signal.) Control signal for the execution of the Dispatch (1E) micro.

1ROTnnF.
1 rotate. (Data lines; nn = 00 thru 23.) Output from the first stage of the rotator.

1USEC...
1-microsecond clock pulse to I/O controls.

1024US..
1024-microsecond clock pulse.

128USEC.
128-microsecond clock pulse.

16USEC..
16-microsecond clock pulse.

2ROTnnF.
2 rotate. (Data lines; nn = 00 thru 23.) Output of the second stage of the rotator.

24-NLJD0
24 minus N left-justified. (Control signal.) Enables mask generator into left-justified mode.

256USON.
256-microsecond clock pulse.

3+1+0/B.
Manipulate term. (Control signal; / specifies active LOW.) Gating term for the Set Carry Flip-Flop (6E) micro.

3F....D0
3F micro. (Control signal.) When TRUE instructs X register to perform a Normalize X (3F) micro.

32US....
32-microsecond clock pulse

36CSKPF0
3C, 6C skip perform. (Control signal.) When TRUE, indicates that a skip has been taken.

4&ROTnD0
4 bit and rotate. (Control signal; n = 0 thru 4.) Selects 4-bit ALU functions or selects T register rotate amount.

4BCDRnB0
4-bits from card B. (Data lines; n = 0 thru 3.) 4 bit data bus from card B.

4CCDRnE0
4-bits from card C. (Data lines; n = 0 thru 3.) 4 bit data bus from card C.

4C5CBRF.
4C, 5C branch. (Control signal.) When FALSE, the branch condition has been met.

4ECDRnE0
4 bits from card E. (Data lines; n = 0 thru 3.) 4-bit data bus from card E.

4FBMnnD0
4 bits from F. (MOP lines; nn = 08, 09, 10, 11.) Output from one of the 4-bit subregisters of F.

4FBRSnD0
4 bit function box register select. (Control signal; n = 0 thru 2.) Selects which card's 4-bit register data will go to the 4-bit function box.

4LREGnF.
4 bit L subregister. (Subregister; n = 0 thru 3.) 4-bit subregister of the L register.

4REGn.F.
4-bit register. (Data lines; n = thru 3.) 4-bit register data path.

4SKPSnD0
4 skip. (Control signal; n = 0 thru 2.) Selects skip condition.

4TREGnF.
4 bits from T. (Data lines; n = 0 thru 3.) 4-bit data path from the T register.

4USEC...
4-microsecond clock pulse.

512USON.
512-microsecond clock pulse.

6E....C.
6E micro. (Control signal.) Decode of Carry FF Manipulate (6E) micro; CP register bit 7.

Documentation Evaluation Form

Title: B 1900 System Technical Manual
Vol. 3: Theory of Operation

Form No: 1127396
Date: November 1980

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized in ensuing revisions to improve this manual.

Please check type of Suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

From:

Name _____

Title _____

Company _____

Address _____

_____

Phone Number _____ Date _____

Remove form and mail to:

Burroughs Corporation
Documentation Dept., TIO - West
P.O. Box 4040
El Monte, CA 91734