3-24. The processor consists of the registers and control logic circuits, microsequencer, arithmetic logic unit (ALU), and the I/O control circuits. The registers and control logic provide control to load the external register (external to the ALU), determine and enable register destination and source, determine the memory function required by the processor (read or write cycle), and enable interrupt routines. The microsequencer provides control to perform instruction execution, specific data processing, and control functions. The ALU is used to perform the arithmetic and logical data manipulation operations. The I/O control is used to initiate and terminate I/O operation between the processor and I/O devices.

3-25. REGISTER AND CONTROL LOGIC (Figure 3-3). The register and control logic consists of the memory address and data registers, instruction register, and instruction decode ROM, RAM general register, program status register, alarm register, flag register, repeat counter, interrupt control logic, and the register control logic. The register and control logic provides control to address memory, transmit and receive memory data, hold and decode the current instruction, provide current program status, provide register data (general register), hold status results of ALU operations, and control the interrupt routines.

3-26. Register Control Logic. The register control logic provides primary and secondary function decoding, develops branch control to the microsequence, enable and load controls to various registers, and control to enable the interrupt service routines. The register control logic receives single cycle control (-QSMTH) from the micro-op clock control, op code bit from the instruction register, and control data from the RDR (decode control QRD18 and QRD19, and source and destination control QRD12 thru QRD16 and QRD27 thru QRD31). QRD12 thru QRD16 and QRD27 thru QRD31 are decoded by the register control logic to determine if a register is to receive data (destination) or to send data (source) to another register. For example, when the memory data register is identified as a destination (receive data) from the ALU, XLDMDRU and XLDMDRL are generated and routed to the memory data register to enable loading the input data (XBA00 thru XBA15) on the next clock pulse. The register control logic then accepts process-to-memory data (XEPMDK) and generates enable memory data control (QEPMD, set high) to place the memory data register output on the memory data bus. If a parity error is detected on the memory data, memory parity error (QEVNPTY) is generated and provides control to the alarm register.

3-27. The register control logic provides enable control (-XEIFCH, -XEPFDC, and -XESFDC) to the instruction decode ROM to develop the appropriate RAR start address. The final microinstruction in the service routine outputs -XEIFCH (set low) to the instruction decode ROM to fetch the next instruction. If no interrupt request input to the instruction ROM, -XEIFCH causes the microprogram to branch to the IFETCH routine which increments the location counter and checks for a full word or halfword instruction. A full word instruction (identified by op code bit QIR01) enables -XEPFDC (set low) as output to the instruction decode ROM. -XEPFDC along with the instruction op code causes the instruction decode ROM to generate a branch to the appropriate microprogram routine to fetch the second halfword of the full word instruction. After the second halfword is fetched, -XESFDC is enabled (set low) to generate a branch in the microprogram to the appropriate sequence in the instruction (macro) list. Halfword instructions immediately enable -XESFDC. -XESFDC also causes the instruction ROM to generate a branch to the illegal/privileged instruction routine for undefined or privileged instructions when the processor is in the protect mode. This routine decrements the location counter to identify the undefined or privileged instruction, and stores it in the appropriate location in the interrupt PSW list.

3-28. The register control logic provides load controls to the flag register, program status register, instruction register, repeat counter, and general register file 1 as determined by QRD12 thru QRD16. When an interrupt service routine is required, -XEIFCH is also developed to enable a new start address to the RAR, and enable RDR inhibit control (-XERDSTP) is generated to inhibit new ROM data from being input to the RDR. -XERDSTP also inhibits the RDR when the microinstruction operation requires more than one clock period (-QSMTH, set high).

3-28.1. General Register File 1. The general register file 1 functions as an accumulator or index register in all arithmetic and logical operations. The general register consists of 16 fixed-point registers. Each of these registers contains a 16-bit halfword consisting of two 8-bit bytes. For arithmetic operations, bit zero (leftmost position) is considered the sign bit using twos complement representation. The general registers are numbered hexadecimally (0, 1, 2...D, E, F). Certain restrictions are placed on the general register's operation.

1. General register 0 will not be used as an index register.

2. The R1 and/or R2 field must specify an even-numbered general register for all full word fixed point instructions and halfword fixed point multiply and divide instructions.

3. For branch or index instructions, the R1 field specifies the first of three consecutive registers; the value of R1, therefore, should be equal to or less than 13.

4. With any RR type instruction, the R1 field and the R2 field may specify the same register, but special attention must be given to what the instruction will do. For example, with the EPSR instruction, the R1 field equals the R2 field, the program status is stored in a general register, but the program status is unchanged.

5. In the conditional branch instructions, the R1 field does not specify a register. Instead, it contains a mask value which is tested with the condition code.

General register file 1 is under control of the register control logic for all operations. The register control logic provides -XRAMS to enable general register functions, XLDRAM to load the general registers, -XRAMA 0 thru 3 to specify which of the 16 registers to be read or written into. The general registers receive input data via the A-bus and output data on to the B-bus.

CSV (CARRY SAVE) FLAG HOLDS RESULT OF ONE OF THE FOLLOWING:

1 XCRYOUT CARRY OUT OF ALU (ADDITIONS, ETC)

2 -XCRYOUT CARRY (BORROW) OUT OF ALU (SUBTRACTIONS)

3 XR00 BIT SHIFTED OUT OF ALU (MSB ON LEFT SHIFT)

4 XSRB0 BIT SHIFTED OUT OF ALU (LSB ON RIGHT SHIFT)

5 XBA00 (A-BUS BIT 0 ON INTERRUPT)

6 XBA12 (A-BUS BIT 12 ON INTERRUPT)

7 QCSV (CSV NO CHANGE)

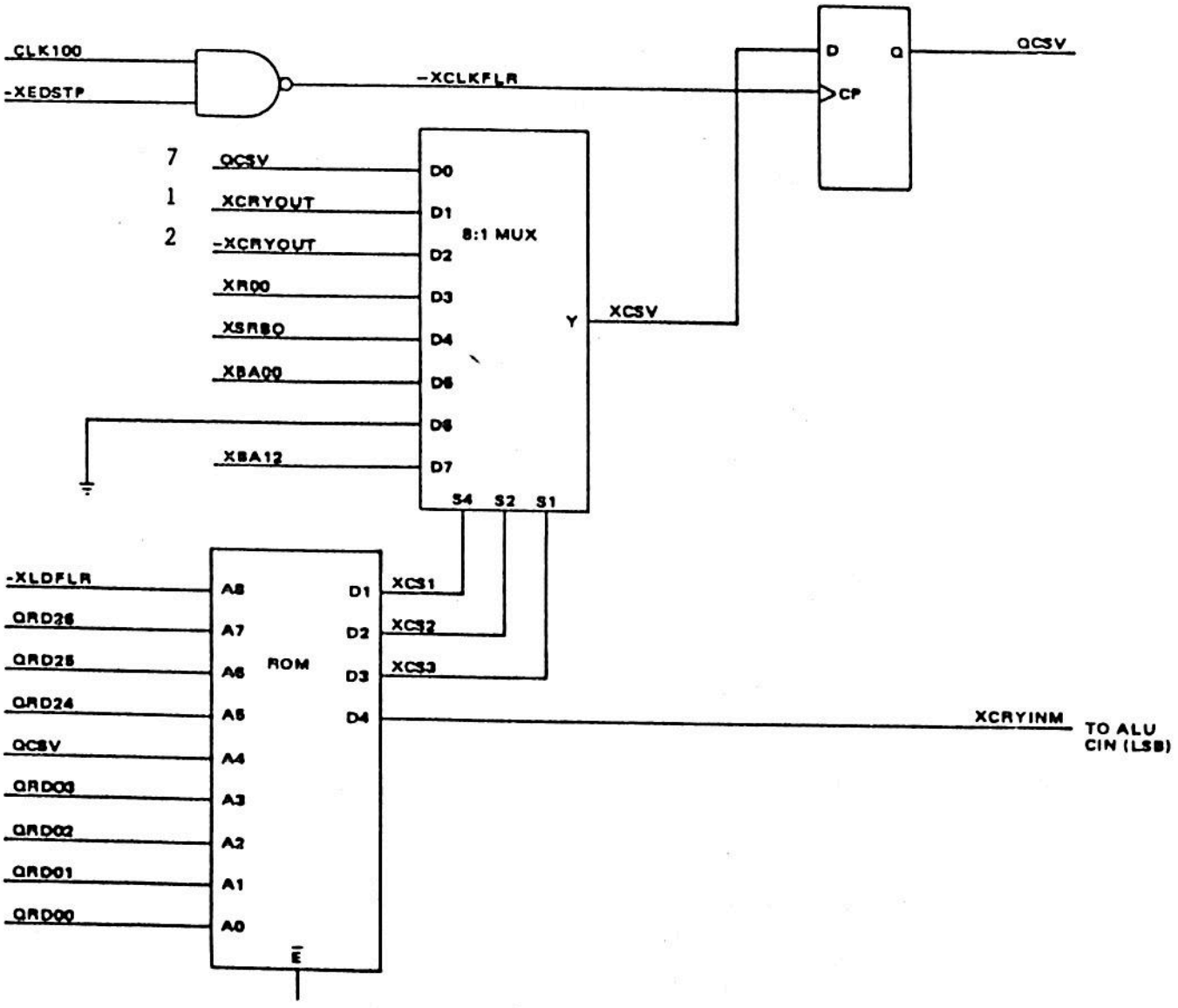


Figure 3-5. Carry Save Flag Logic

QVF (OVERFLOW) FLAG HOLDS ONE OF THE FOLLOWING:

1 OVERFLOW OUTPUT OF MOST SIGNIFICANT ALU
2 ALGEBRAIC OVERFLOW ON LEFT SHIFTS
3 IO TIME OUT BIT
4 A BUS (BIT 13 ON LOAD FLR)
5 A BUS (BIT 1 ON INTERRUPT)
6 QOVF (NO CHANGE)

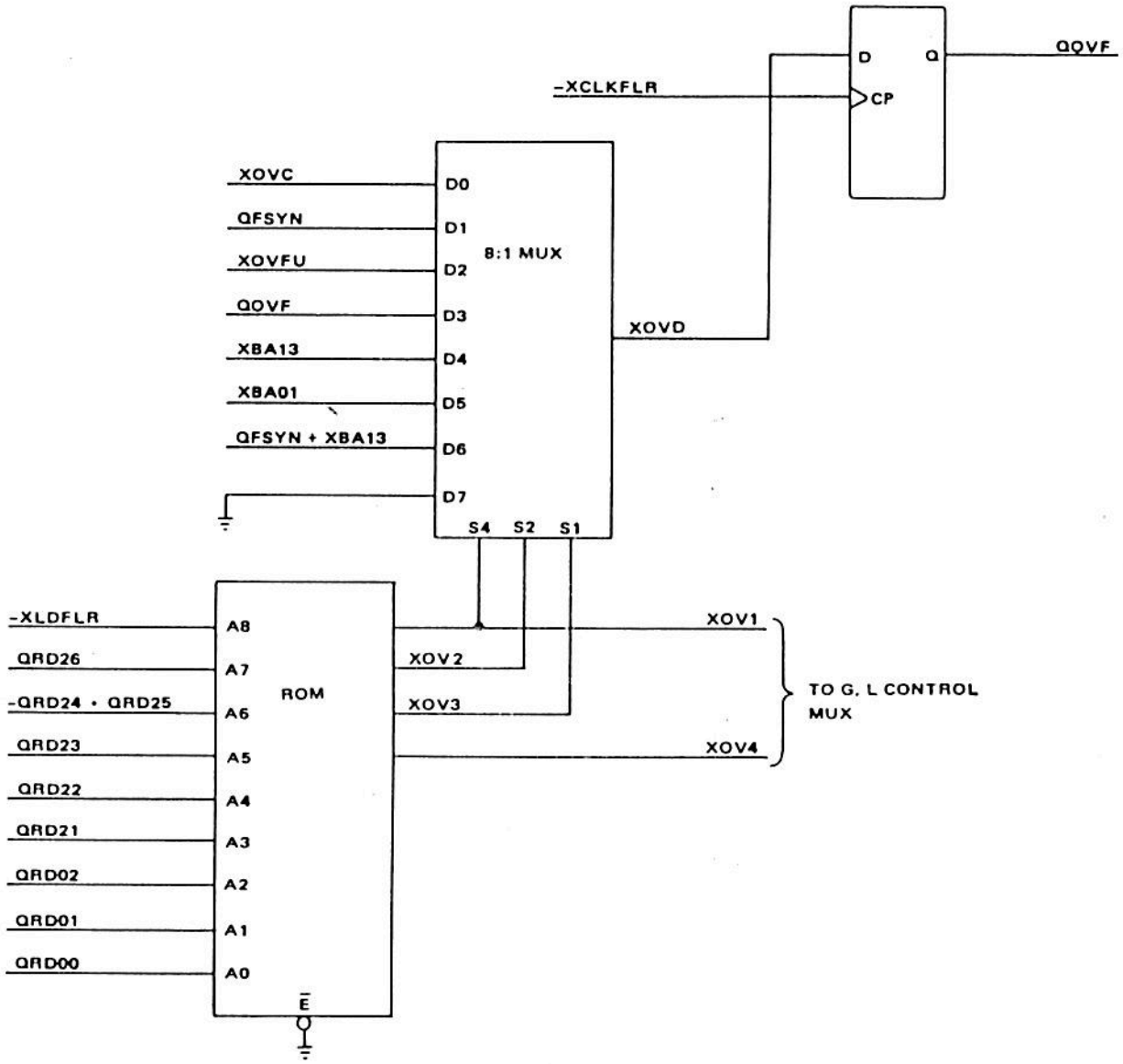


Figure 3-6. Overflow Flag Logic

OG, QL (GREATER THAN, LESS THAN) FLAGS HOLD RESULT OF ONE OF THE FOLLOWING:

- SINGLE PRECISION ALU OPERATIONS (QG & QL SET AS FOLLOWS):

ALU OUTPUT	OG	OL
>0	1	0
=0	0	0
<0	0	1

- DOUBLE PRECISION ALU OPERATIONS (NEW QG, QL SETTINGS REFLECT PREVIOUS SETTINGS)
- A-BUS (BITS 2,3 ON INTERRUPT)
- A-BUS (BITS 14, 15 ON LOAD FLR)

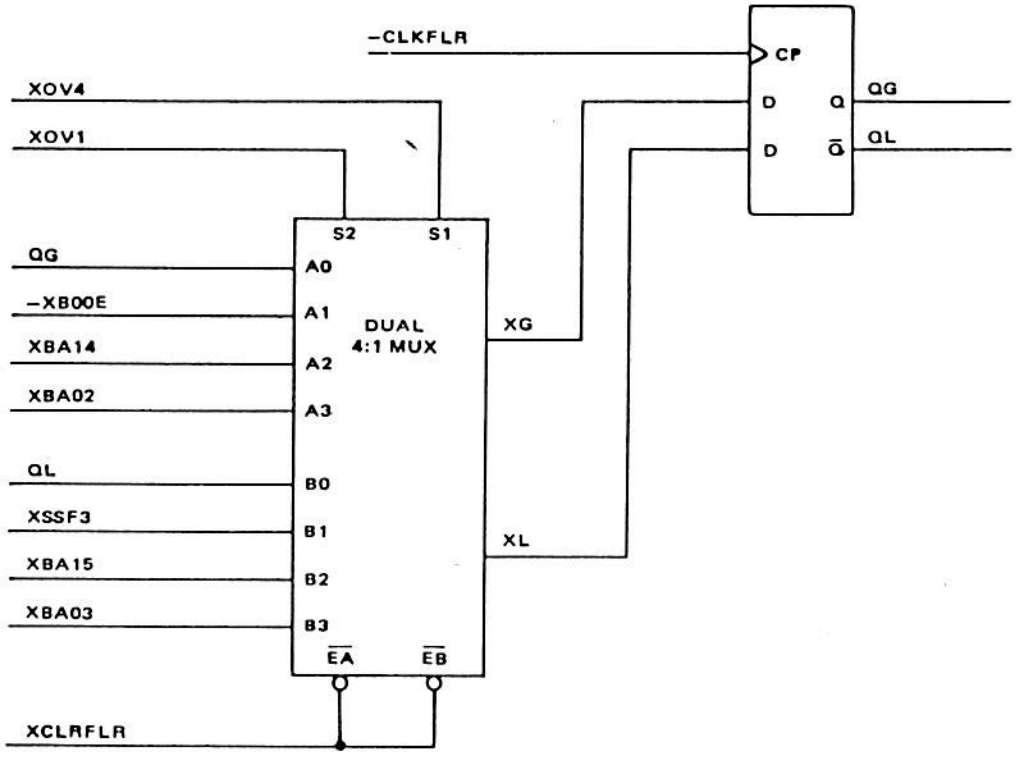
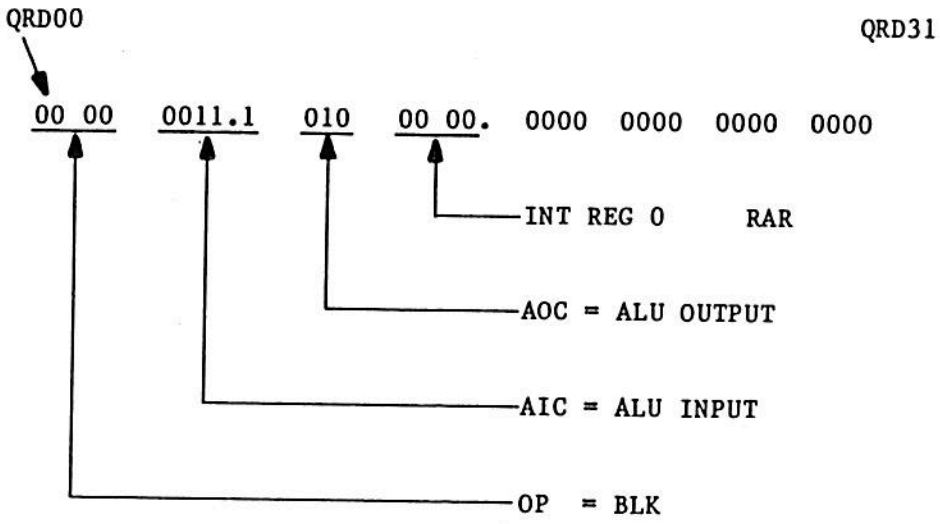


Figure 3-7. Greater Than and Less Than Flag Logic

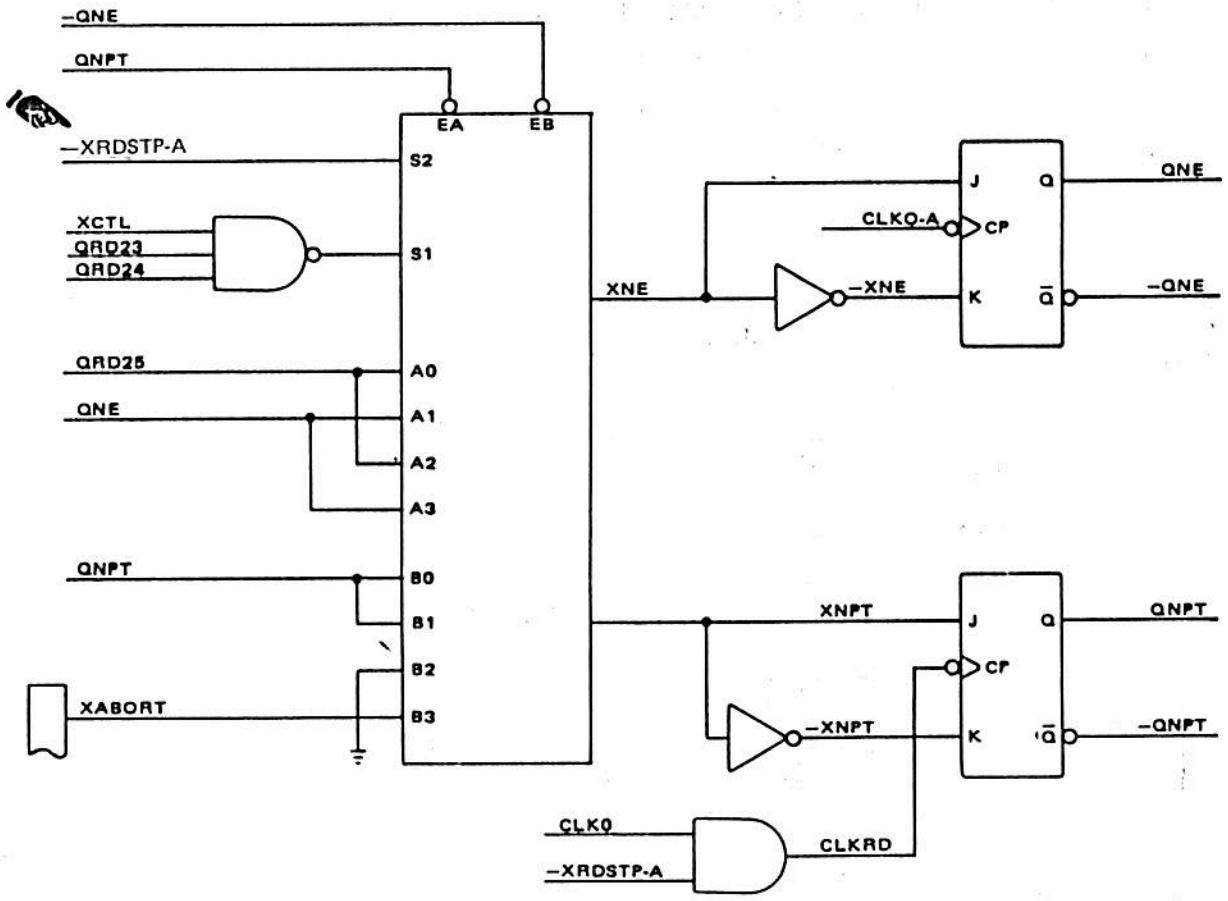
3-47. After an enable/disable PSW bit (in the program status register) is set, a pending interrupt of the corresponding type initiates the interrupt procedure. When an interrupt request is sensed and enabled (fig 3-8 and 3-9), the RDR is cleared by the clear RDR control (-XCLRRD). The interrupt control 1 (-QNPT-A) sets (sets high) certain bits of the RDR and the instruction executed during an interrupt cycle is as follows:



The BLK enables the contents of the RAR to be placed onto the B-bus and stored in internal register 0 (in register file 2). -QNPT-A enables the RAR to be loaded from the A-bus (loaded with start address for interrupt microsequence). External interrupts from an I/O device are initiated by -TATN (attention bit to computer from I/O device) going active. -TATN going active causes XINTRPT to set high, causing the instruction decode ROM to output the starting location of the interrupt service routine. After the addressed device has received its address via the I/O mux bus, -TSYN (sync control bit from I/O device) going active causes ENABLE DEST CONTROL (-XEDSTP) and ENABLE RDR INHIBIT CONTROL (-XERDSTP) to go active. -XEDSTP, by setting LOAD RAR CONTROL (-XLDRAR) high, inhibits the next sequential microinstruction from being loaded into the RAR. -XERDSTP causes -XRDSTP to go low, inhibiting the RDR clock (-CLKRD) from clocking through the microinstruction that was addressed prior to the interrupt.

QNE - ABORT INTERRUPT ENABLE FLIP-FLOP. SET AND RESET UNDER MICROPROGRAM CONTROL

QNPT - ABORT INTERRUPT CYCLE FLIP-FLOP. SET WHEN INTERRUPTS ARE ENABLED (QNE SET HIGH AND ABORT INTERRUPT REQUEST ACTIVE).



SET

QNE = XNE • CLKO-A

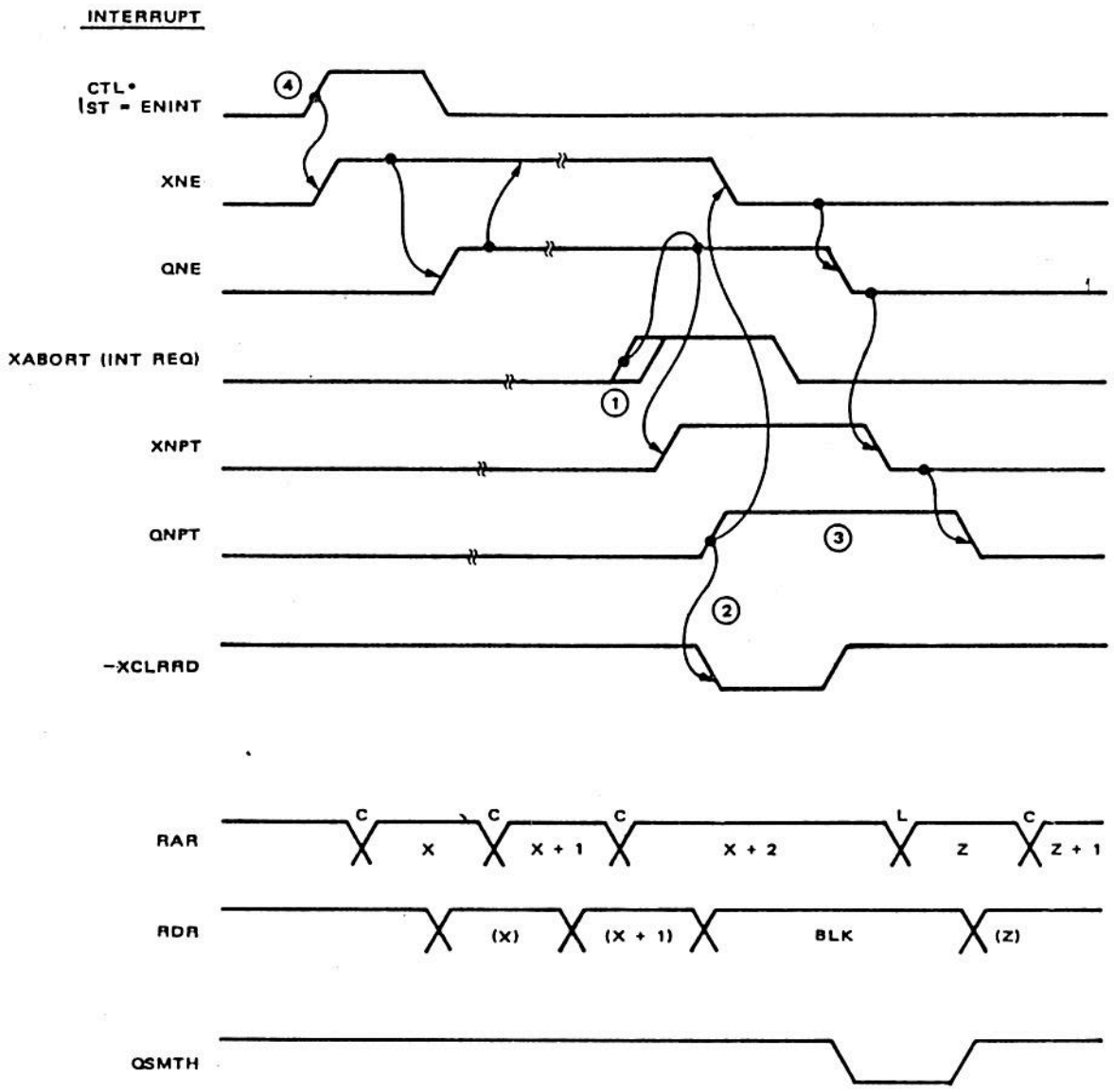
XNE = -QNPT • [(XCTL • QRD23 • QRD24 • QRD25) QNE]

RESET

-QNE = -XNE • CLKO-A

-XNE = XCTL • QRD23 • QRD24 • -QRD25

Figure 3-8. Abort Interrupt Logic



① SEQUENCE STARTED BY INTERRUPT REQUEST WHEN QNE IS ACTIVE

② PRESENCE OF QNPT • QNE CLEARS RDR AND FORCES BLK INSTRUCTION (BY DEFAULT)

③ EXECUTION & TIMING IS IDENTICAL WITH THAT OF BLK, EXCEPT THAT QNPT IS ACTIVE FOR CYCLE

④ INT ENB IS SET & RESET BY CONTROL N-INSTR

Figure 3-9. Abort Interrupt Timing

3-48. Memory Data and Memory Address Registers. The memory data and memory address registers provide temporary storage for memory data and addresses allowing memory operation and microsequence execution to overlap.

3-49. The memory address register (MAR) provides address data to the memory (memory interface function) during a processor-initiated memory cycle. The MAR receives address information from the A-bus consisting of the memory address of the next instruction to be accessed (address supplied by location counter in register file 2), or the memory address of an operand (address supplied by instruction decode ROM). Address input (from the A-bus) is loaded into the MAR under control of the load memory address register control (XLDMAR, set high). When memory is ready to accept the address as input, the enable memory address control (-QEPMA) provides control to place this information onto the memory address bus (-TMA00 thru -TMA14) which is then routed to the memory interface function. Memory address (XBB00 thru XBB15) is placed on the B-bus when memory address requires modification.

3-50. The memory data register (MDR) consists of an input multiplexer, data register, and a tri-state transceiver (transceiver). During a processor-initiated memory write cycle, the input multiplexer selects memory input data from the A-bus (XBA00 thru XBA15), under control of the select control (XLDMDRM, set low). During a processor-initiated memory read cycle, the input multiplexer selects memory data (-TMD00 thru -TMD15) from the transceiver under control of XLDMDRM set high (the transceiver contains output data from memory). The data register receives the memory data and provides output to the transceiver and the B-bus. Memory data is placed on the B-bus during a processor-initiated memory read cycle when an operand has been fetched from memory. When memory is ready to accept the data as input, the enable memory data control (QEPMD, set high) provides control to place the data onto the memory data bus. The transceiver is connected to the memory data bus and functions as both a receiver and a transmitter. The receive function is always enabled to accept data from the memory data bus; however, due to logic implementation, the data is enabled onto the memory data bus only at the appropriate times. The transmit function is enabled when QEPMD is set high.

3-51. Instruction Register. The instruction register (IR) provides temporary storage for the instruction fetched from memory during a processor-initiated memory read cycle. The IR receives input from the memory data bus (-TMD00 thru -TMD15) and from the A-bus (XBA00 thru XBA15). Data is loaded into the IR from the memory data bus under control of the load instruction register control (XLDIRM, set high). Data is loaded from the A-bus under control of the load instruction register control (XLDIR, set high). The instruction register provides output to the B-bus (XBB00 thru XBB15), the instruction decode ROM (QIR00 thru QIR07), the RAM general register file 1 (QIR08 thru QIR15), and to the register control logic (QIR01).

3-52. Instruction Decode ROM. The instruction decode ROM receives the op code field (QIR00 thru QIR07) from the IR, decodes this data and develops the start address (XBA04 thru XBA15), for the microsequences required to execute the instruction. The enable controls (-XEIFCH, -XEPFDC, and -XESFDC) provide control to the instruction decode ROM to initiate an instruction fetch cycle, indicate if a second halfword is required from memory, or cause a branch to the appropriate sequence in the microprogram. -XEPFDC is set low, and the op code field (QIR00 thru QIR07) generates a branch to the appropriate routine for fetching a second halfword. After the second halfword fetch, -XESFDC set low causes the ROM to branch to the appropriate sequence in the macro list. Halfword instructions immediately set -XESFDC low. -XESFDC also causes a branch to the illegal/privileged instruction routine in the case of undefined instructions or privileged instructions when in the protect mode.

3-53. RAM general register file 1 is a random-access memory (RAM) containing sixteen 16-bit general purpose registers. The data contained in these registers may be altered as required for program execution. The contents of the register specified by the register select controls (QIR08 thru QIR15) are accessed (read out) under control of RAM control -XRAMS set low. New data is stored (written into) in the register specified by QIR08 thru QIR15 under control of RAM control XLDRAM set high. General register data output (XBB00 thru XBB15) is placed on the B-bus when a RAM read cycle is indicated (-XRAMS, set low).

3-54. Repeat Counter. The repeat counter provides control to processor operations requiring multiple iterations. The repeat counter is loaded from the A-bus (-XBA11, XBA12 thru XBA15) under control of the load control (XLDCTR, set high). Decode control (XDECTR, set high) provides control to decrement the repeat counter by one each clock period. The count status (-XCTZRO and -XCTONE) are routed to the register control to provide control to the branch logic and interrupt control logic.

3-55. MICROSEQUENCER (Figure 3-3). The microsequencer provides control to the processor functional circuits to execute specific data processing (add, subtract, multiply, etc) and control functions (fetch, interrupt, program load, etc). The microsequencer consists of a ROM address register (RAR), a read-only memory (ROM), and a ROM data register (RDR).

3-56. ROM Address Register. The RAR is a 12-bit register that maintains the address of the next ROM location to be accessed (fig 3-10 thru 3-12). The RAR is normally incremented by one each cycle time, unless one of the following conditions occurs: branch to a new address, interrupt, RAR specified as a destination of the operation results, or an instruction decode vector process occurs. When RAR is specified as the destination, or a branch-on-index instruction is specified, the RAR receives the start address (XBA04 thru XBA15) from the A-bus. During other branch operations, the RAR receives the branch address input (QRD20 thru QRD31) from the RDR. During an instruction decode operation, the RAR receives the new start address from the instruction decode ROM (XBA04 thru XBA15). During an interrupt, the RAR receives a new address from 12 external lines enabled onto the A-bus.

RAR UPDATE METHODS:

1. AS A DESTINATION FROM A-BUS
2. ADDRESS FIELD OF CURRENT MICROINSTRUCTION (BRANCH)
3. FROM A-BUS WITH A VECTORED ADDRESS DURING AN INTERRUPT OR INSTRUCTION DECODE
4. INCREMENTED BY ONE (NORMAL MODE OF OPERATION)

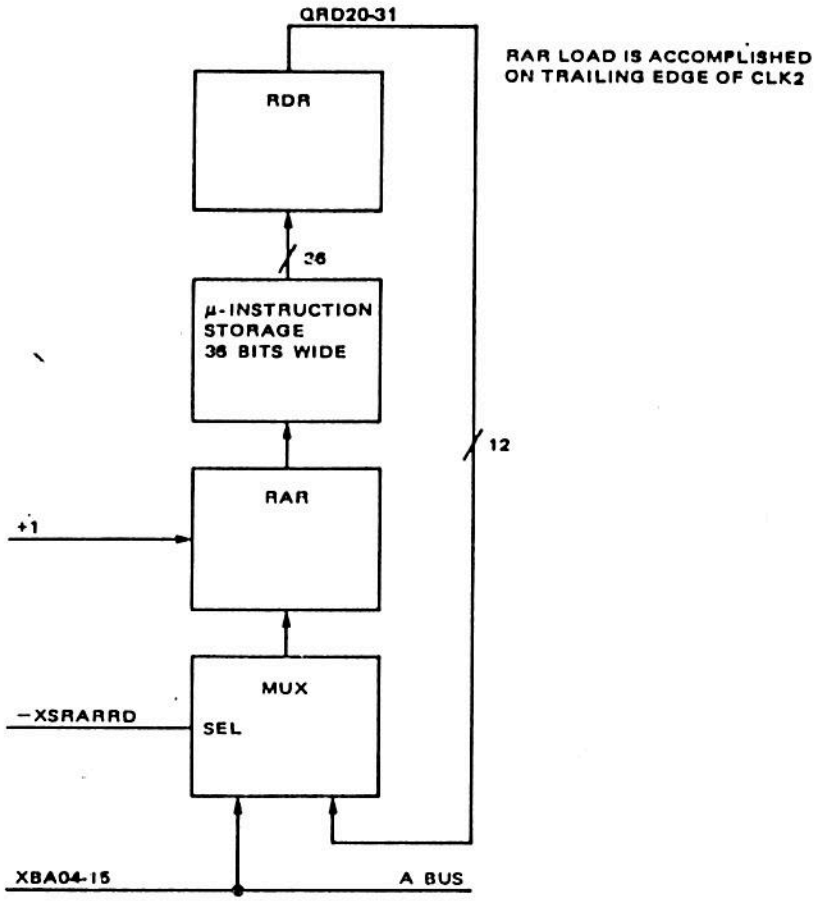
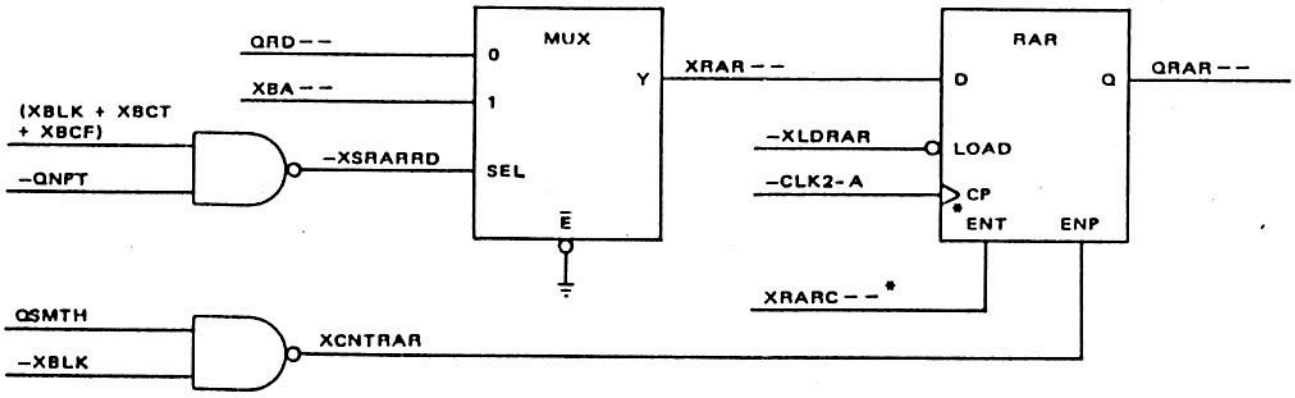


Figure 3-10. RAR Address Input Source

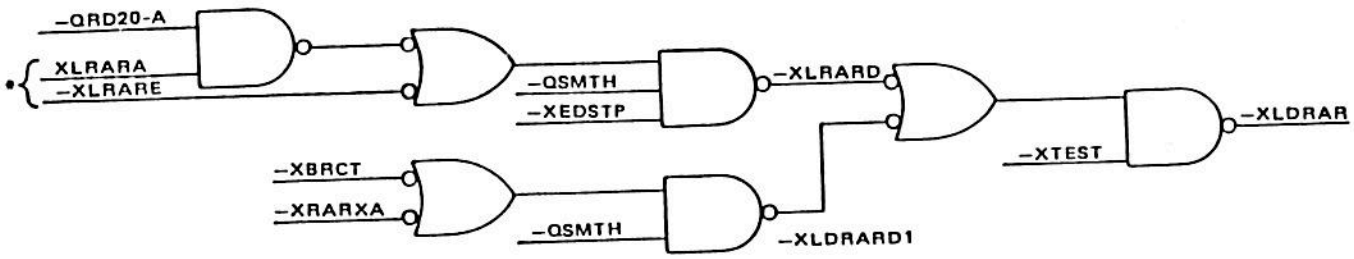


UPDATE MODE	CONDITION	CONTROLS
FROM A-BUS	RAR = DEST	-XSRARRD = 1
		-XLDRAR = 0
		XCNTRAR = X**
FROM ADDRESS	BRANCH	-XSRARRD = 0
FIELD OF RDR		-XLDRAR = 0
OUTPUT		XCNTRAR = X
FROM A-BUS	INTERRUPT	-XSRARRD = 1
	OR INSTRUCTION	-XLDRAR = 0
	DECODE	XCNTRAR = X
INCREMENT	NORMAL PROGRAM	-XSRARRD = X
	SEQUENCING	-XLDRAR = 1
		XCNTRAR = 1

*-ENT CONNECTED TO LOGICAL 1 FOR
LEAST SIGNIFICANT STAGE OF COUNTER

** -X = NOT IMPORTANT

Figure 3-11. RAR Control Logic



-XLRARE = OUTPUT FROM MSB CONTROL ROM. CONTROLS THE UNCONDITIONAL LOADING OF RAR.

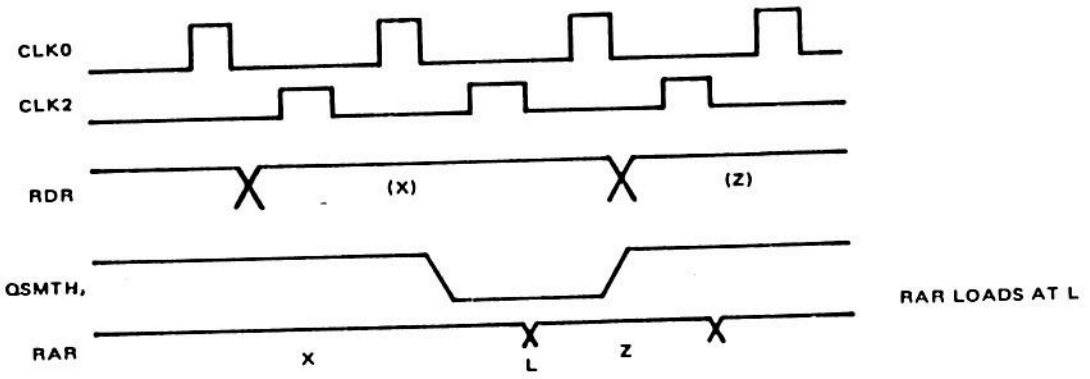
XLRARA = OUTPUT FROM MSB CONTROL ROM. ACTIVE WHEN THERE IS A LOAD RAR CONDITION (RAR = DEST).

XRARXA = INDICATES CTL MICROINSTRUCTION WITH DECODE SPECIFIED.

XLDRAR = [(XLRARA • -QRD20 + XLRARE) • -QSMTH • -XEDSTP]
+ QSMTH (XBRCT + CLT = DECODE)

LOAD MODE 1: UNCONDITIONAL LOAD

- -XLRARE IS ACTIVE (LOW)
- XRDSTP ACTIVATED BY XRDSA (2 CYCLE OPERATION)
- -XEDSTP NOT ACTIVE (DESTINATION ENABLED)



* – FROM MSB CONTROL ROM, U31

Figure 3-12. RAR Load Modes (Sheet 1 of 2)

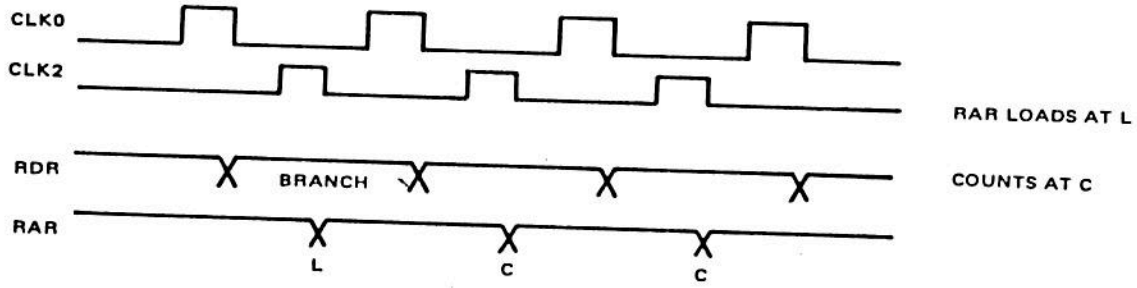
LOAD MODE 2: RAR = DEST LOAD IF DESTINATIONS ARE ENABLED

- XLRARA IS ACTIVE
- QRD20 = 0 (DESTINATION CLOCKS NOT INHIBITED)
- -XEDSTP NOT ACTIVE (DESTINATIONS ENABLED)

SAME TIMING AS MODE 1

LOAD MODE 3: LOAD BRANCH ADDRESS (NOT BRANCH LINK)

- -XBRCT ACTIVE LOADS FROM RDR (ADDRESS PORTION)



LOAD MODE 4: LOAD DECODE ADDRESS - BRANCH TO BEGINNING OF MICRO-CODE ROUTINE

- -XRARXA ACTIVE - LOADS FROM A-BUS

SAME TIMING AS MODE 3

Figure 3-12. RAR Load Modes (Sheet 2 of 2)

Any of the 16 internal registers (located in general register file 2) may be used for entry into a linkage routine. Internal register 0 is reserved for an interrupt subroutine. When either event occurs, the RAR contents are transferred (through the ROM address B-bus multiplexer) into the 12 LSBs of the designated internal register (for storage until subroutine is complete). The FLR contents are simultaneously transferred into the 4 MSBs of the same register. To return from an interrupt subroutine, the RAR is loaded with an address from the internal register 0 (or other designated internal register). The RAR is incremented (by one) when returning from a subroutine, since the address stored (in the designated register) was the address of the branch and link instruction.

3-57. The RAR provides control (ROM address, QRAR04 thru QRAR15) to select the one of four ROM banks (eight ROM banks in a 4,096 configuration) and one of 512 ROM locations within the selected bank as illustrated in figure 3-13.

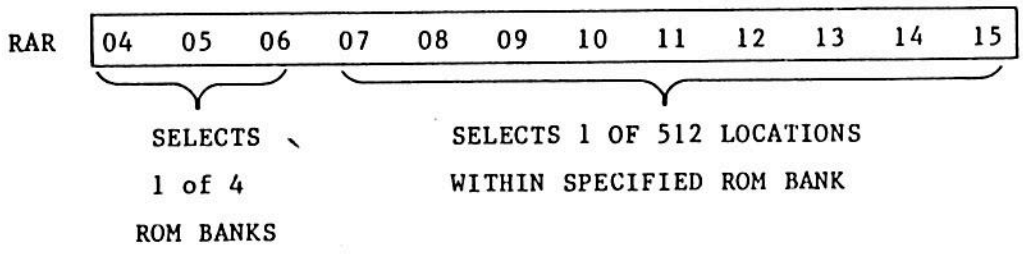
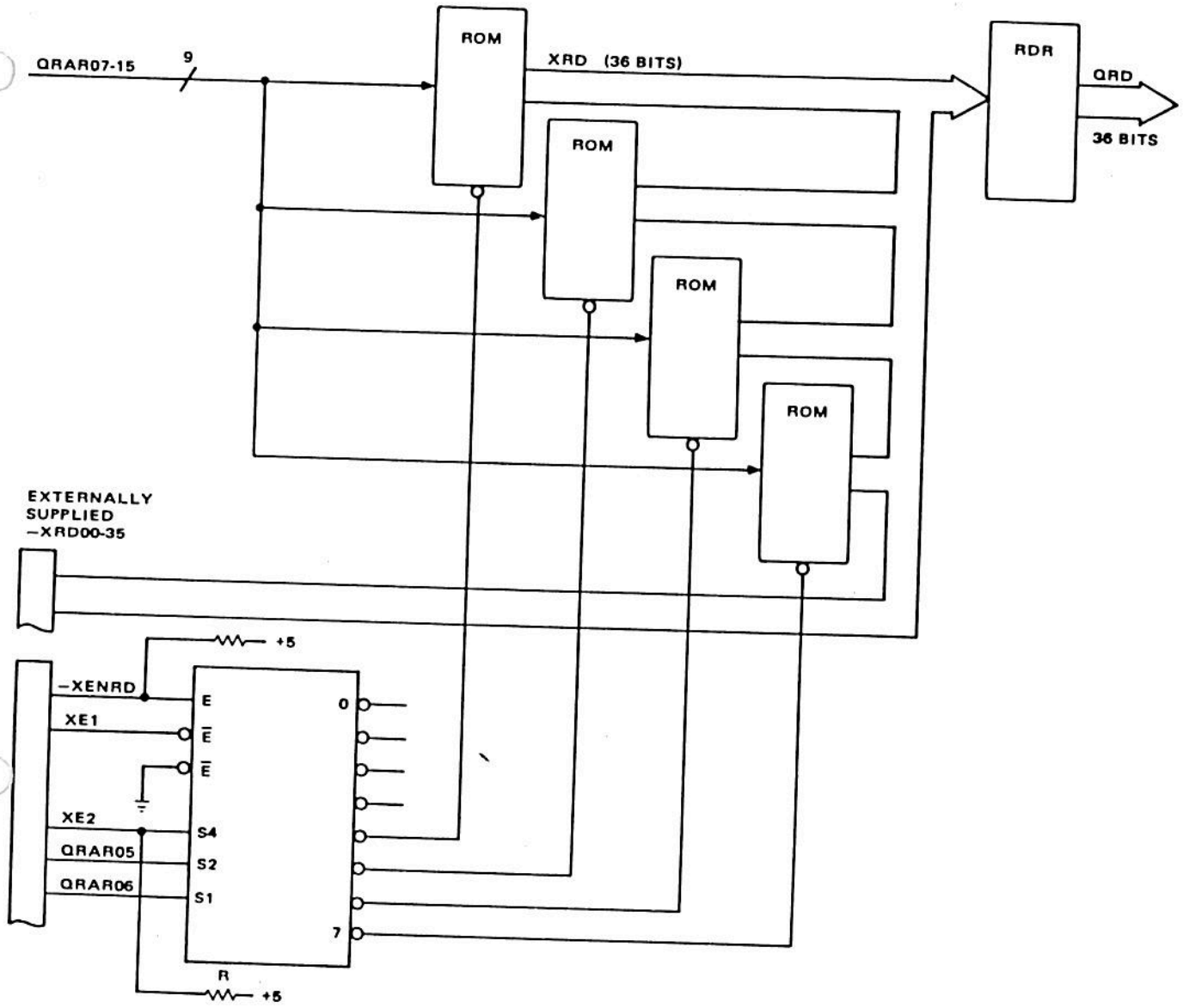


Figure 3-13. RAR Address Word

3-58. Read-Only Memory. The microsequencer is configured around a programmable ROM providing storage for 2,048 microinstructions (36-bit words). The microprogram may be expanded to 4,096 microinstructions with the addition of a second ROM logic card. The ROM decodes the address word (from the RAR), and selects the specified ROM bank and ROM location. The selected microinstruction (XRD00 thru XRD35) is output to the RDR (fig 3-14).

3-59. ROM Data Register. The RDR provides temporary storage for the 36-bit microinstruction word as it is decoded and executed. Figure 3-15 illustrates the clock control logic and timing for the RDR loading. The fetch of the next sequential

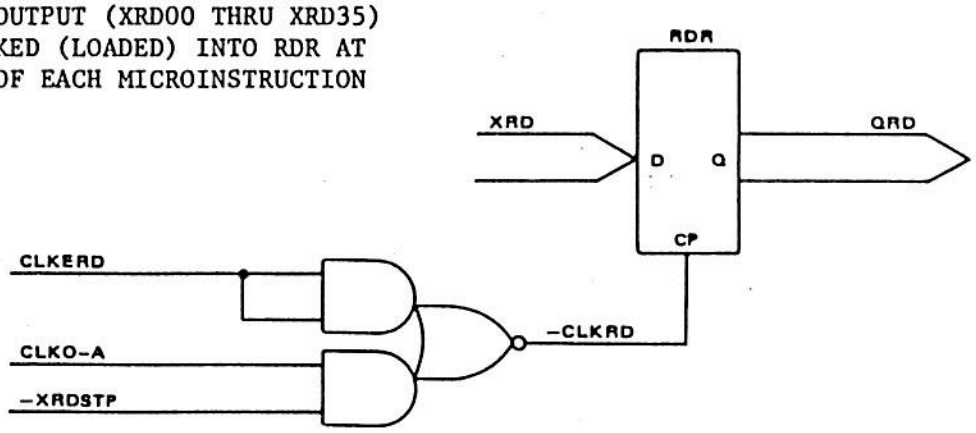


-XENRD - LOGICAL 1 WHEN ROMS ON CARD ARE TO BE SELECTED
- LOGICAL 0 WHEN XRD00-35 TO BE SUPPLIED FROM OFF CARD.

	CONDITION	-XENRD	XE1	XE2
①	DISABLE ON CARD ROM - XRD SUPPLIED FROM EXTERNAL ROM	PULLED LOW FROM EXTERNAL SOURCE	d.c.	d.c.
②	CARD CONTAINS LOWER 2K ROM	LOGICAL 1	WIRE TO QRARO4	PULLED UP VIA R
③	CARD CONTAINS UPPER 2K ROM	LOGICAL 1	WIRE TO GROUND	WIRE TO QRARO4

Figure 3-14. ROM Control Logic

ROM OUTPUT (XRD00 THRU XRD35)
CLOCKED (LOADED) INTO RDR AT
END OF EACH MICROINSTRUCTION



RDR CLOCKS: CLKERD + CLKO • -XRDSTP

CLKERD - EXTERNAL SIGNAL USED BY MAINTENANCE EQUIPMENT TO LOAD RDR WITH EXTERNALLY SELECTABLE DATA

-XRDSTP - INHIBITS RDR FROM CLOCKING DURING A MULTICYCLE INSTRUCTION (XRDSTP MUST BE INACTIVE FOR RDR TO CLOCK)

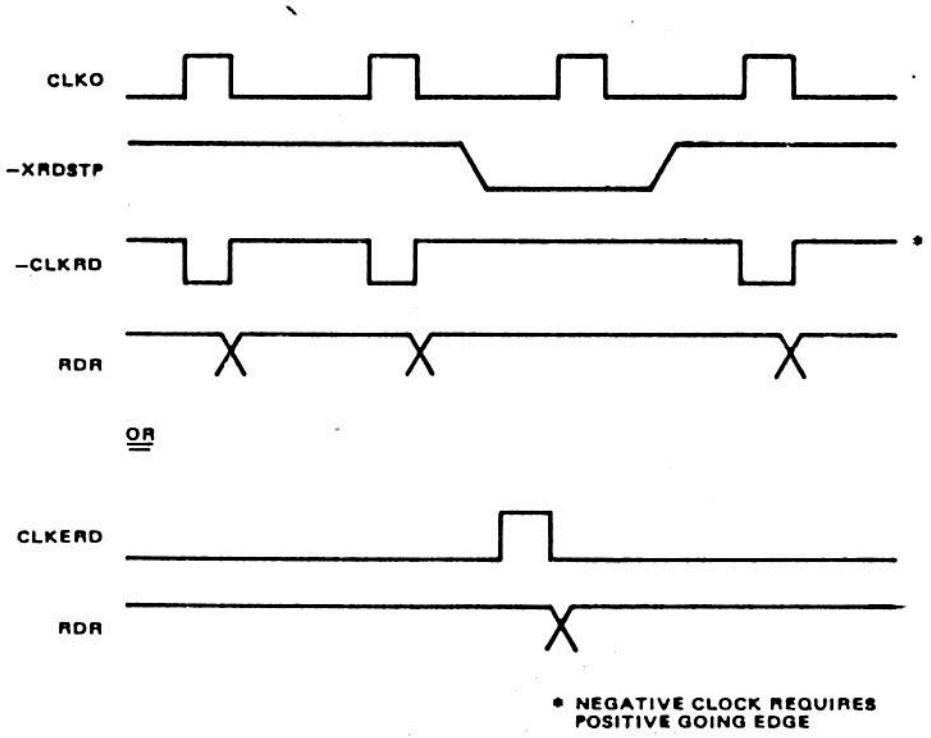
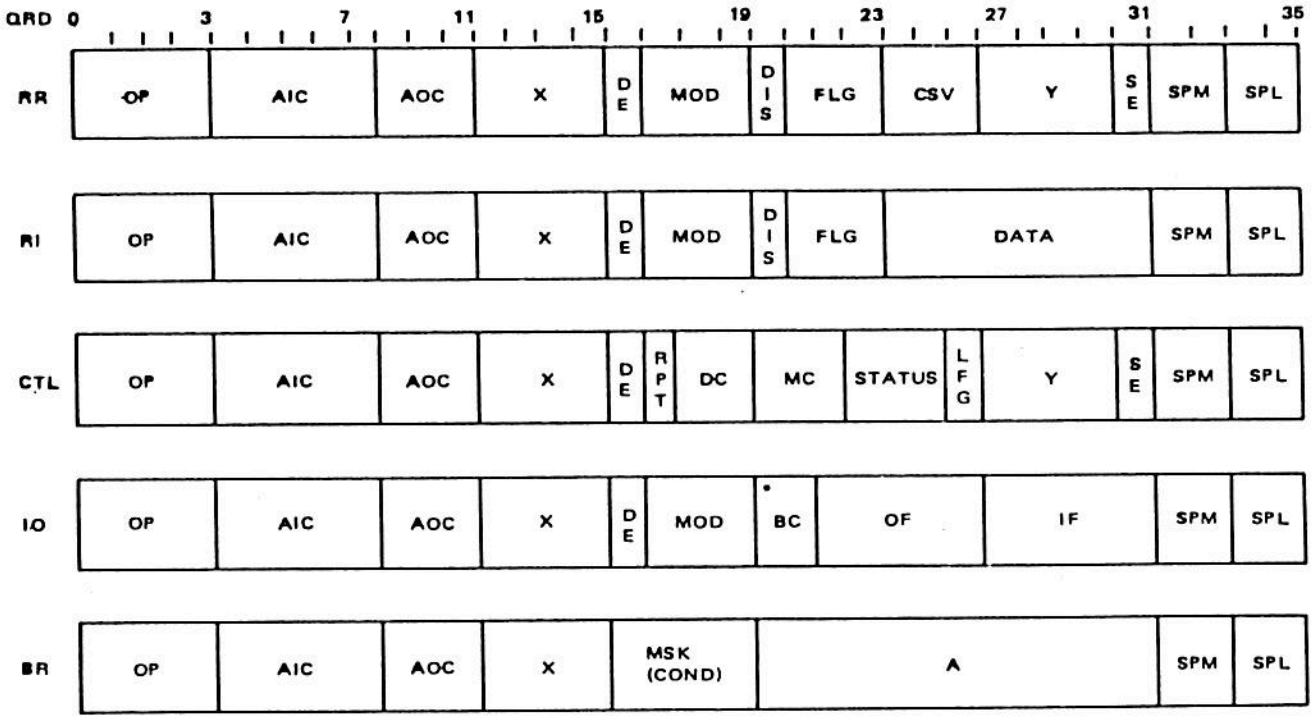


Figure 3-15. RDR Control Logic and Timing

microinstruction. Microinstructions are organized into five basic categories: register-to-register (RR), immediate-to-register (RI), control (CTL), input/output (I/O), and branch (BR). Each basic type of microinstruction has a fixed format, as shown in figure 3-16. Table 3-3 contains field definitions.

TABLE 3-3. MICROINSTRUCTION FIELD DEFINITIONS

Field	Definitions
OP	4-Bit microinstruction operation code
AIC	5-Bit ALU input control
AOC	3-Bit output destination control
X	4-Bit destination control
Y	4-Bit source control
DE	1-Bit destination indicator control
DC	2-Bit decode control for RAR input source
MOD	3-Bit modifier control used for byte manipulations
DIS	1-Bit destination clock control
FLG	3-Bit flag register load control
CSV	3-Bit carry save control
SE	1-Bit source indicator control
RPT	1-Bit repeat microinstruction control
LFG	1-Bit
MC	3-Bit memory control
STATUS	3-Bit
BC	2-Bit I/O bus control
OF	5-Bit output code
IF	5-Bit input code
DATA	8-Bit data word
NDX	4-Bit index word
MSK	4-Bit branch mask
A	12-Bit branch address
SPM	2-Bit Cordic function select control
SPL	2-Bit ARB ARC enable control



FUNCTION	OP CODE	FORMAT	CRY
BLK	0000	BR	0
BCT	0001	BR	0
BCF	0010	BR	0
BIF	0011	BR	0
IN	0100	IO	0
OUT	0101	IO	0
ALU	0110	RI	0
ALU	0111	RI	1
ALU	1000	RR	0
ALU	1001	RR	1
ALU	1010	RR	CSV
ALU	1011	RR	-CVS
ALU	1100	CTL	0
ALU	1101	CTL	1
MUL/DIV	1110	RR	-CVS
MUL/DIV	1111	RR	-CVS

*Not Used

Figure 3-16. Microinstruction Word Formats

3-60. Microinstruction Formats. Tables 3-4 thru 3-14 explain encoding of the various microinstruction word fields. For each format defined in table 3-4, there are 228 different states of the remaining bits in the microinstruction word. The combined status of the OP-field (RQD00 thru QRD03), AIC-field (QRD04 thru QRD08), AOC-field (QRD09 thru QRD11), and the DIS-field (QRD20) determine the operation, internal destination, and carry into the LSB portion of the ALU. Table 3-4 provides the OP-field. The SPM field (QRD32 and QRD33) provides function select control for the CORDIC option as defined in table 3-5.4. The SPL field (QRD34 and QRD35) provides control to enable single or double precision arithmetic operations and transfer control between the ARB and optional ARC as shown in table 3-5.3. Tables 3-5.1 and 3-5.2 provide field definitions for the MC field and the OF field. Tables 3-5 and 3-6 provide field definitions for the AIC- and AOC-fields and table 3-7 provides definitions of terminology used for AIC and AOC. A DIS-field of one (QRD20, set high) inhibits loading the destination register for comparison operations. A magnitude comparison is performed. Pretest of the operand to determine unlike signs is required for algebraic comparisons. The MOD-field (QRD16 thru QRD19) determines the byte/digit manipulation required on the input data to the ALU for register transfer, algebraic, and logical operations as defined in table 3-8. The MOD-field also determines the end bit values for positioning, multiplication, and division operations as defined in table 3-9. The X-field (QRD12 thru QRD15) and Y-field (QRD27 thru QRD31) provide destination and source control by specifying registers for operation as defined in table 3-10. The FLG-field (QRD21 thru QRD23) and CSV-field (QRD24 thru QRD25) determine the status saved in the FLR as defined in tables 3-11 and 3-12.

3-61. The RI microinstruction format performs ALU operations between the 8-bit DATA-field (data, QRD24 thru QRD31) and a register specified in the X-field. The 8 MSBs of the immediate operand placed on the B-bus are set to zero (set low). The combined state of the OP-, AIC-, AOC-, and DIS-fields determines the operation to be performed. The MOD-field determines the byte/digit manipulation operation on the immediate operand from the B-bus. The FLG-field (QRD21 thru QRD23) determines the status saved in the overflow, greater than and less than flags in the flag register. The carry save flag is not altered.

3-62. The CTL microinstruction format is a specialized register-to-register operation. No byte/digit manipulation or positioning operations are performed, the status of the result is not saved, and a comparison operation is not performed.

TABLE 3-4. OP CODE FORMATS

OP		Format	Description	Input Carry
Hex	Binary			
0	0000	BR	Branch and link	0
1	0001		Branch on condition true	
2	0010		Branch on condition false	
3	0011		Branch indexed on false	
4	0100	I/O	Input	0
5	0101		Output	
6	0110	RI	Immediate-to-register	0
7	0111			1
8	1000	RR	Register-to-register	0
9	1001			1
A	1010			CSV
B	1011			-CSV
C	1100	CTL	Register-to-register control	0
D	1101			1
E	1110	(DIV) RR	Divide step	-CSV
F	1111	(MUL) RR	Multiply step	-CSV

CSV = Carry save flag of the status register

TABLE 3-5. AIC-FIELD DEFINITIONS

AIC		ALU Function	
Hex	Binary	Carry In = 0	Carry In = 1
00	00000	A + Q	A + Q + 1
01	00001	A + B	A + B + 1
02	00010	Q	Q + 1
03	00011	B	B + 1
04	00100	A	A + 1
05	00101	D + B	D + B + 1
06	00110	D + Q	D + Q + 1
07	00111	D	D + 1
09	01001	B - A - 1	B - A
0A	01010	Q - 1	Q
0B	01011	B - 1	B
0C	01100	A - 1	A
0D	01101	B - D - 1	B - D
0F	01111	$\overline{D}$	- D
11	10001	A - B - 1	A - B
12	10010	$\overline{Q}$	- Q
13	10011	$\overline{B}$	- B
14	10100	$\overline{A}$	- A
15	10101	D - B - 1	D - B
16	10110	D - Q - 1	D - Q
17	10111	D - 1	D
18	11000	A $\wedge$ B	A $\wedge$ B $\leftarrow$
19	11001	A $\vee$ B	A $\vee$ B $\leftarrow$
1A	11010	A $\oplus$ B	A $\oplus$ B $\leftarrow$
1C	11100	D $\wedge$ B	D $\wedge$ B $\leftarrow$
1D	11101	D $\vee$ B	D $\vee$ B $\leftarrow$
1E	11110	D $\oplus$ B	D $\oplus$ B $\leftarrow$

NOTE: CODES NOT USED

08	01000
0E	01110
10	10000
18	11011
1F	11111

TABLE 3-5.1. MC FIELD DEFINITION

MC	FUNCTION
001	Read First Half of Instruction
010	Read Second Half of Instruction
011	Read Operand
100	Not Used (Read)
101	Write
110	Write Privileged
111	Not Used (Write)
000	No Operation

TABLE 3-5.2. OF FIELD DEFINITIONS

OF	I/O OPERATION
00001	Address
00010	Command
00011	Data Available
00100	Data Request
00101	Acknowledge Interrupt
00111	Status Request

TABLE 3-5.3. ARB AND ARC ENABLE CONTROL

DIS	SPL	FUNCTION
0	00	Single Precision, Disable ARC Dest. Clock
0	01	Transfer from ARC to ARB, Disable ARC Dest. Clock
0	11	Double Precision
1*	01	Double Precision, Disable ARB and ARC Dest. Clocks
1*	10	Transfer from ARB to ARC, Disable ARB Dest. Clock
1*	11	Single Precision, Disable ARB Dest. Clock

*Also Disables Dest. Clocks to External Registers

TABLE 3-5.4. SPM FIELD DEFINITION

SPM	XMARD=1, LOAD QMARA-B WITH	XMARD=0, COR CARD OPERATION	**
00	QMARA-B*	No Operation	
01	Zeros	Increment Iteration Counter	
10	PSW10-11	Load Shift Counter, Shift Algebraic	
11	PSW08-09	Load Shift Register, Shift Logical	

*If Input Microinstruction, Load With TDMA-B.

**XMARD=1 when the MAR is the Destination.

TABLE 3-6. AOC-FIELD DEFINITIONS

AOC		Scratch Register Load Control	Scratch Register Input Shift Control	Output Data Control
Octal	Binary			
0	000	Load Q	No shift	ALU output
1	001	(Not used)	(Not used)	(Not used)
2	010	Load RAM	No shift	RAM location A
3	011	Load RAM	No shift	ALU output
4	100	Load RAM and Q	Right shift	ALU output
5	101	Load RAM	Right shift	ALU output
6	110	Load RAM and Q	Left shift	ALU output
7	111	Load RAM	Left shift	ALU output

TABLE 3-7. AIC AND AOC TERMINOLOGY DEFINITIONS

Format	Source Control		ALU Address		Implied Function and Source Restrictions	ALU Input Control AOC Restrictions
	DE	SE	A	B		
CTL or RR	0	0	Y	X	I ← I ⓕ I	—
	0	1	(Y)		I ← I ⓕ E	
	1	0			E ← E ⓕ I	
	1	1		(X)	E ← ⓕ E	*Unary operations only
RI	0	-	(Im)	X	I ← I ⓕ Im**	—
	1			(X)	E ← ⓕ Im**	*Unary operations only
BLK	-	-	(A)	X	I ← 'RAR'; 'RAR' ← A	—
BIF	-	-	(A)	X	'RAR' ← I ⓕ A	
I/O	0	-	(IF)	X	I ← ⓕ I/O; I/O ← ⓕ I	*Unary operations only
	1			(X)	E ← ⓕ I/O; I/O ← ⓕ E	
BCT and BCF	-	-	(A)	(X)	RAR ← A	—

I = Register internal to ALU; E = Register external to ALU; - = Not specified;
( ) = Input value which should not be meaningful to the operation;
**Im = B-bus input modified by MOD-field, if any;
*Unary operations = Invert, twos complement, transfer, increment, or decrement

TABLE 3-8. BYTE/DIGIT MANIPULATIONS

D_0	D_1	D_2	D_3	B-bus input

MOD		(QMAR15) External Control	Resulting Word				Function
Octal	Binary						
0	000	X	D_0	D_1	D_2	D_3	Parallel load
1	001	X	D_2	D_3	D_0	D_1	Byte exchange
2	010	0	D_2	D_3	P	P	Byte insert
		1	P	P	D_2	D_3	
3	011	0	0	0	D_0	D_1	Byte extract
		1	0	0	D_2	D_3	
4	100	X	0	0	0	D_0	Hex digit 0 extract
5	101	X	0	0	0	D_1	Hex digit 1 extract
6	110	X	0	0	0	D_2	Hex digit 2 extract
7	111	X	0	0	0	D_3	Hex digit 3 extract

X = Don't care; P = Previous contents; 0 = Zero

TABLE 3-9. BINARY POSITIONING CONTROL

MOD		AOC	ALU MSB	Q LSB	
Octal	Binary	LSB	Input	Input	Function
0, 1	00X	0	0	Hi Z	Right logical shift; set ALU MSB to zero (when OP = 14 or 15, set ALU MSB to carry out)
		1	Hi Z	0	Left logical shift; set Q LSB to zero
2, 3	01X	0	B-Bus ∨ OVF	Hi Z	Multiply; set ALU MSB to true sign (F ⊕ -F)
		1	Hi Z	ALU MSB Carry	Divide; set Q LSB to quotient bit (ALU carry)
4, 5	10X	0	C	Hi Z	Right algebraic shift; set ALU MSB to CSV
		1	Hi Z	0	Left algebraic shift, set Q LSB to CSV
6, 7	11X	0	Q LSB	Hi Z	Right rotate; set ALU MSB to Q LSB
		1	Hi Z	ALU MSB	Left rotate; set Q LSB to ALU MSB

Hi Z = Hi impedance to permit shift in opposite direction

TABLE 3-10. INTERPRETATION OF X- AND Y-FIELDS

Format	DE	SE	Operation
RR, CTL	0	0	I(X)←I(X) op I(Y) or I(X)←op I(Y)
	0	1	I(X)←I(X) op E(Y) or I(X)←op E(Y)
	1	0	E(Y)←E(Y) op I(X) or E(Y)←op I(X)
	1	1	E(X)←op E(Y) only

E(X) = external register selected by X-field

E(Y) = external register selected by Y-field

I(X) = internal register selected by X-field

I(Y) = internal register selected by Y-field

TABLE 3-11. FLG-FIELD DEFINITION

FLG			
Octal	Binary	Status Saved in V Flag of FLR	Status Saved in G and L Flags of FLR
0	000	-	-
1	001	-	Single precision algebraic result
2	010	-	Multiprecision algebraic result
3	011	Algebraic overflow	Multiprecision algebraic result
4	100	Algebraic overflow	Single precision algebraic result
5	101	*V Flag or carry save exclusive ORed with left-shifted out bit of ALU	Single precision algebraic result
6	110	Algebraic overflow	-
7	111	*V Flag or carry save exclusive ORed with left-shifted out bit of ALU MSB	-

- = No change;
*Only for MOD code of algebraic shift or rotate, otherwise set to zero

TABLE 3-12. CSV-FIELD DEFINITION

CSV		
Octal	Binary	Status Saved in C Flag of FLR
0	000	-
1	001	Carry from ALU MSB
2	010	-
3	011	Borrow from ALU MSB
4	100	-
5	101	Left-shifted out bit from ALU MSB
6	110	-
7	111	Right-shifted out bit from Q LSB

- = No change

These functions are replaced with fields which may be used to perform control of external (external to ALU operations) logic structures at the same time that register-to-register ALU operations are performed. The RPT-field (QRD17) is used to execute the next microinstruction repeatedly by controlling clocks to the RAR and RDR (such as for bit positioning, multiplication, and division operations). The STATUS-field (QRD23 thru QRD25) in conjunction with the LFG-field (QRD26) provides control to clear the FLR (flag register) or load the 4 MSBs of the ALU results (table 3-13) into the FLR and to control microprogram interrupts. Interrupts occur only on the first clock of a microinstruction execution requiring multiple clock cycles (such as an input/output operation).

TABLE 3-13. STATUS-FIELD CODE DEFINITIONS

STATUS		LFG Bit State	Mnemonic Identifier	Function
Octal	Binary			
0	000	0	-	No change to FLR
		1	LFG	Load FLR from A-bus MSD
1	001	0	JMC	FLR contents replaces the condition code, Zero FLR
		1	-	Load FLR from A-bus MSD
2	010	0	ALM	No change to FLR, ALM replaces the condition code
		1	-	Load FLR from A-bus MSD
3	011	0	JAM	FLR replaces condition code, No change to FLR
		1	-	Load FLR from A-bus MSD
4	100	0	AMC	No change to FLR, ALM replaces condition code, Zero ALM
		1	-	Load FLR from A-bus MSD
5	101	0	CLR	Zero FLR
		1	-	Load FLR from A-bus MSD
6	110	0	RIE	Reset interrupt enable
		1	IEL	Load FLR from A-bus MSD; reset interrupt enable
7	111	0	SIE	Set interrupt enable
		1	-	Load FLR from A-bus MSD; set interrupt enable

3-63. Response to an internal interrupt inhibits execution of the current microinstruction, stores the microinstruction address and the contents of the flag register in internal register 0, loads a 12-bit externally (external to the ALU) specified address into RAR, and resets the interrupt enable control. A new address may be transferred into RAR on interrupt to provide a priority interrupt structure. Interrupts are reenabled under microprogram control. The DC-field (decode control, QRD18 and QRD19) may also be used to enter a new address into RAR under microprogram control as defined in table 3-14. The MC-field (QRD20 thru QRD22) provides control to initiate transfer operations between the processor and memory as defined in table 3-5.1.

3-63.1. Response to an external interrupt allows the execution of the current microinstruction. Upon completion of the microinstruction, the RAR is externally loaded with the 12-bit address pointing to the beginning of the I/O service routine. The PSW containing the flag register and location of the next microinstruction is transferred into memory and exchanged with the PSW already residing in memory that contains the proper flag configuration and location to service the interrupt. Normal microprogram control is returned upon completion of the service and retrieval of the old PSW.

TABLE 3-14. DC-FIELD DEFINITIONS

DC		Identifier	Function
Octal	Binary		
0	00	-	RAR controlled by destination REF
1	01	-XEPFDC	RAR loads from external source
2	10	-XESFDC	
3	11	-XEIFCH	

3-64. The I/O microinstruction format is used to enable processor I/O operations. During an input operation, the input data is generated by an external source (I/O device or an option) and routed through the I/O multiplexer bus to the B-bus.

During an output operation, the output data is generated within the processor and routed through the A-bus to the I/O multiplexer bus and to the external destination (an I/O device or an option). The BC-field (QRD20 and QRD21) provides I/O multiplexer bus control. The OF-field (output flag, QRD22 thru QRD26) provides a 5-bit code to determine the output control line or lines to be enabled. Definitions of the OF-field coding are provided in table 3-5.2. The IF-field (input flag, QRD27 thru QRD31) for all I/O microinstructions will be 00001 to enable sync (-TSYN) to terminate the microinstruction. The DE-field (QRD16) provides control to identify an external destination.

3-88. MEMORY DETAILED FUNCTIONAL BLOCK DIAGRAM DESCRIPTION (Figure 3-26).

3-89. The memory is a high-speed random access, semiconductor storage device providing storage for instructions or data. There are two types of configurations for the memory. The first type contains a maximum of 16 read/write modules, with each module providing storage capacity for up to 8192 halfwords. A fully extended memory of this type can provide a maximum of 131,072 halfwords. The second type contains a maximum of four read/write modules, with each module providing storage capacity for up to 32,768 halfwords. A fully extended memory of this type contains a maximum of 131,072 halfwords. The memory consists of a memory interface function plus either of the two memory types mentioned.

3-90. MEMORY INTERFACE FUNCTION. The memory interface (MIF) provides interface between the processor and memory storage and between the DMA bus and memory storage (fig 3-27). The MIF processes memory access requests (from the processor and the DMA devices), provides control to the refresh memory cycle, determines request priority, identifies the memory bank to be accessed, and provides the processed memory address to the RAM memory. The MIF controls priority of memory access request, granting priority to the refresh memory cycle first, followed by the DMA request and then the processor request. The MIF then provides timing control to memory operations through a phase counter and clock delay line.

3-91. REFRESH MEMORY CYCLE. The refresh memory cycle is initiated by the refresh internal counter by refresh count control (QREFCT1 and QREFCT4 set high). QREFCT1 and QREFCT4 provide timing for the refresh request control logic. The refresh request control logic generates refresh frequency control (QREFREQ and -QREFREQ) every 14.8 microseconds. -QREFREQ with the memory enable control logic generates memory enable control (-QMENB-A) to enable memory. QREFREQ (set high) with DMA cycle control (QMEMDMA) and memory phase count 0 (XPHASE0) generate refresh memory control (-XMEMREF). -XMEMREF is inverted to generate XMEMREF (set high) to enable the refresh memory cycle control logic. The refresh memory cycle control logic's outputs, refresh address control (QMEMRED and -QMEMRED) and refresh control (XREF, XREF-A, and -XREF-A) provide enables and disables to the refresh address counter and the memory address control logic. QMEMREF and memory phase count 1 (XPHASE1) enable the refresh address counter to increment to the next row or memory to be refreshed. XREF (set high) inhibits the memory address word (TMA03 thru TMA14) from the memory address control lines (XARR03 thru XARR14). -XREF-A (set low) enables the refresh address (QREFAD0 thru QREFAD5) to be placed on the memory address control lines. A read cycle then takes place, refreshing that row of memory.

3-92. DMA Memory Cycle. A DMA memory cycle is initiated by a DMA device (I/O) device) by DMA memory access request (-TDMAREQ, set low) (fig 3-28). -TDMAREQ provides control to the DMA request and memory cycle control logic to enable DMA request control (QDMAREQ, set high). When the current memory cycle is complete, DMA access request is granted (QMEMDMA) if the memory modules do not require a refresh cycle (-QREFREQ, set high). QMEMDMA then sets DMA enable control (-TEN, set low) to notify the DMA device that memory access has been granted. If the memory request is for a memory write cycle (-TDMAWRT, set low), the DMA device provides memory address (-TMA, -TMAB, -TMA00 thru -TMA14) to the MIF over the memory address bus. The memory address is held on the memory address bus until the DMA data strobe is enabled (-TDMAST set low). Device provides memory data (-TMD00 thru -TMD15) to the memory modules over the memory data bus and holds this data until -TDMAST is set low. If the memory request is for a memory read cycle, the DMA device provides the memory address over the memory address bus and waits for the data to be placed on the memory data bus, ad indicated by -TDMAST (set low).

request is for a memory read cycle, the DMA device provides the memory address over the memory address bus and waits for the data to be placed on the memory data bus, as indicated by -TMAST (set low).

3-93. Processor Memory Cycle. A processor memory cycle is initiated by the OP-field (op code, QRD00 thru QRD02) and the MC-field (memory controls, QRD20 thru QRD22). The op code and memory controls are decoded by the memory enable control logic to determine the type of memory cycle requested (write or read). The processor request is granted if a refresh or DMA memory cycle request has not been sensed (-QMEMDMA, set high). The MIF enables memory access granted (-QMAG, set low) to notify the processor that memory access has been granted. If the memory request is for a memory write cycle, the processor provides memory address (-TMA00 thru -TMA14) to the MIF over the memory address bus. The processor provides memory data (-TMD00 thru -TMD15) to the memory modules over the memory data bus under control of the enable process-to-memory data control (XEPMDK, set high). If the memory request is for a memory read cycle, memory read enable control (-QRDENB-A set low) provides control to place the memory data (-TMD00 thru -TMD15) on the memory data bus which is then routed to the processor.

3-94. Memory timing control is provided to the memory phase counter and clock delay lines. The 3-bit phase counter decodes the system clock (CLKM0) to provide 200-nanosecond memory phases. The clock delay lines provide incremental timing control for functions within the memory phases. The delay line taps are successive 5-nanosecond delays individually referenced to system clock (CLKM0). Figure 3-28, sheets 1 thru 3, illustrates the control provided by the phase counter and delay line during a refresh memory cycle, a processor memory cycle, and a typical DMA request memory cycle.

3-95. The memory bank controls (-TMAA and -TMAB) and memory address word controls (TMA00 thru TMA02) are decoded by the memory bank select control logic to enable the appropriate memory module for accessing. The bank select controls (XADRA and XADRB) and address controls (XADR00 thru XADR02) are routed to each memory module as input; however, only one memory module responds to that bank and address configuration.

3-96. MEMORY MODULE. For 8K memory module configurations, each of the from 1 to 16 memory modules (memory circuit cards) contains 8,192 18-bit word storage capacity. Each module is divided into two banks containing 4,096 18-bit word storage capacity.

A bank of memory is selected by decoding the bank select controls (XADRA, XADRB, and XADR00 thru XADR02, common to both banks). The location within a selected bank is enabled by address controls (XA0 thru XA11). The memory enable control (-QMENB-A, set low) provides control to enable the memory operations. For 32K memory module configurations, each of the 1 to 4 memory modules (memory circuit cards) contains 32,768 18-bit word storage capacity. Each module is divided into two banks containing 16,384 18-bit word storage capacity. A bank of memory is selected by decoding the bank select controls (XADR00, XADRA, XADRB common to all banks). The location within a selected bank is enabled by address controls (XA0 thru XA06), row address strobes (-RAS1 thru -RAS4), and column address strobes (-CAS0 and -CAS1). The memory enable control (-QMENB-A) set low provides control to enable memory operations.

3-97. For 8K memory module configurations, if the requested memory cycle is for a write cycle, data is written into (stored) the selected memory location under control of the write controls (-QWRITE-A, -QWRTENB-A). -QWRTENB-A provides control to generate write controls -DWRITE1 and -DWRITE2 which enable the I/O control logic to accept memory data -TMD00 thru -TMD15 as input to the memory. -QWRITE-A provides control to generate write controls -W1 and -W2 (set low) which enables the RAM to store -TMD00 thru -TMD15 into the selected RAM location. For 32K memory module configurations, data is written into (stored) the selected memory location under control of write control (-QWRTENB-A). -QWRTENB-A provides control to generate write enables (-WE0 and -WE1) which enables the selected location in the RAM memory to allow data to be written from the memory data bus (TMD00 thru TMD15) via the memory data transceivers. Read control (DRD) is set low, allowing data to be passed through the transceivers.

3-98. For 8K memory module configurations, if the requested memory cycle is for a read cycle, data is read from the selected RAM location under control of the memory read enable control (QRDENB-A), and write control (QWRITE-A). QRDENB-A provides control to generate DREAD, and QWRITE-A provides control to generate XR. Read control XR (set high) provides control to the RAM to read data out of the selected RAM location. The RAM data is then routed to the I/O control logic. DREAD (set high) then provides control to place the memory data (-TMD00 thru -TMD15) onto the memory data bus. For 32K memory module configurations, data is read from the selected RAM location under control of memory read enable control (QRDENB-A).

QRDENB-A provides control to generate driver read enable (DRD). DRD set high enables the driver function of the transceiver devices to output the selected location in the RAM to the memory data bus (TMD00 thru TMD15).

3-99. The memory modules consist of 36 metal oxide semiconductors (MOS) dynamic RAM integrated circuits. The MOS RAM requires a refresh cycle at least once every millisecond. A refresh cycle is accomplished by executing a memory read cycle on all 64 combinations of addresses -A0 thru -A5 or, for a 32K memory module, -A0 thru -A6. Alternatively, the refresh requirement can be satisfied during normal memory use by exercising the 64 combinations of -A0 thru -A5, or A0 thru A6, through either a read cycle, or a write cycle at least once every 1 millisecond.

3-100. In the event of a prime power failure, the memory may be placed in a standby mode to conserve battery power. However, refreshing must be maintained to preserve the memory contents. The +5v AUX, the +12v, and -5v supplies are required for stand-by operation, the main +5v supply is not required. Memory may be retained for periods of 30 seconds to 2 hours (after prime power failure), depending on the battery size.

3-101. MASTER CLOCK DETAILED BLOCK DIAGRAM DESCRIPTION (Figure 3-29).

3-102. The master clock contains a 20-MHz clock oscillator counted down to develop 5-MHz clocks to provide the timing control to the computer operations. The 20-MHz clock (-CLK20) is counted down by the divide-by-four logic to develop two 5-MHz clocks (CLK0 and CLK2), 50 nanoseconds wide, staggered in time by 100 nanoseconds (fig 3-30).

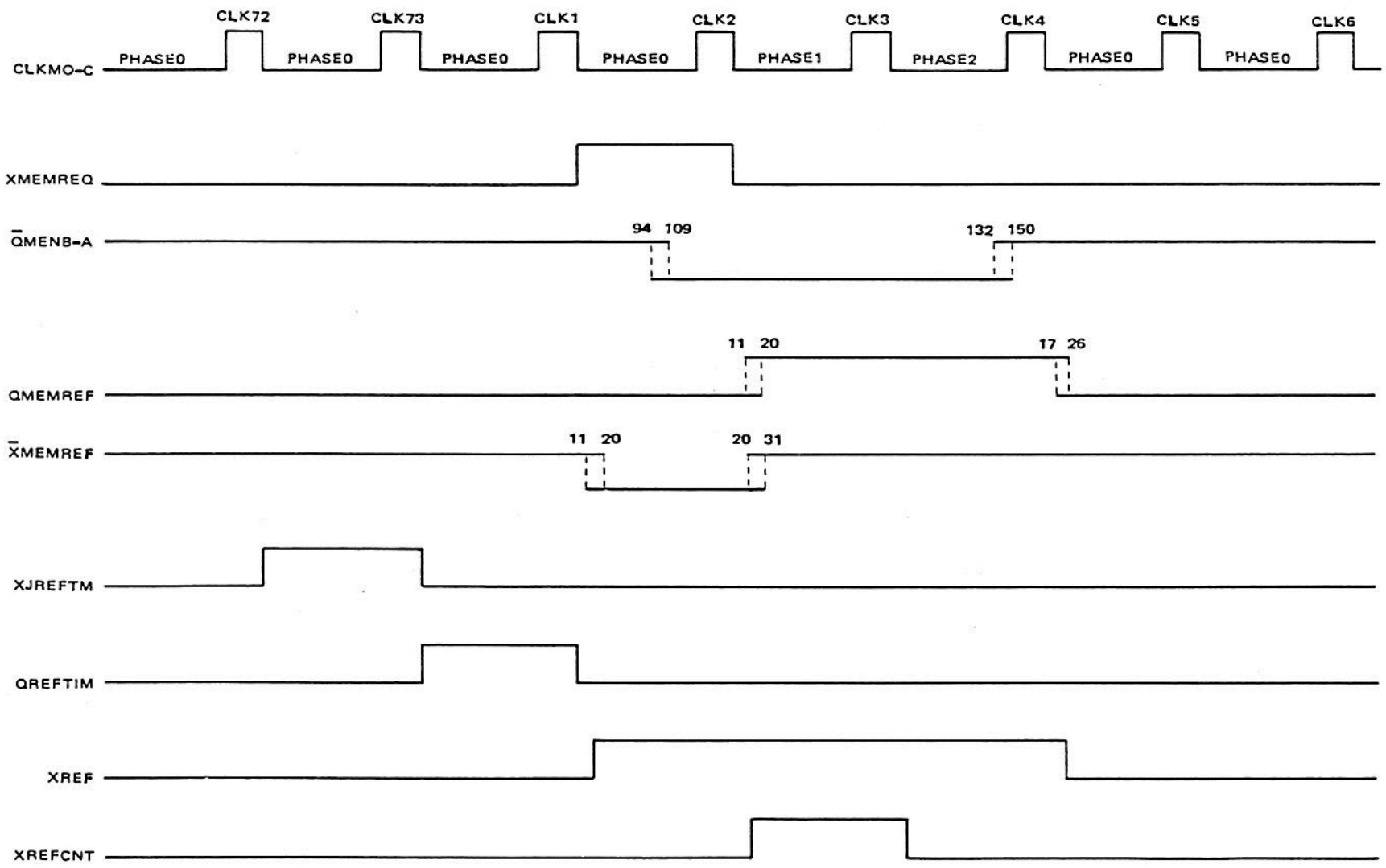
3-103. Figures 3-31 thru 3-33 illustrate processor and ALU timing control provided by the master clock. While most microinstructions are executed in one clock period (200 nanoseconds), some require more than one clock period, such as the following operations: I/O micro-operations, operations involving RAR as a source or destination, branch and link or interrupt operations, repeated microinstructions, and micro-operations conflicting with memory operations. When executing a multicycle operation, the RAR is permitted to update but the clock control to the RDR is inhibited (fig 3-31).

3-104. The RDR and destinations are clocked on the trailing edge of clock pulse CLK0 (fig 3-32). The RAR is clocked on trailing edge of the half-cycle-clock pulse (CLK2). XRDSA provides control to determine when an extra clock period is required to complete an operation such as when the RAR is a source (provides data output) or is a destination (to receive data input), or during an I/O operation.

3-105. Clock pulses (CLK0 and -CLK100) provide control to the ALU to timing control during ALU operations. -CLK100 provides control to load data into the LSI circuits of the ALU (fig 3-33). CLK0 provides the basic timing to the ALU for input data and output data transfer and manipulations.

3-106. MAINTENANCE PANEL DETAILED BLOCK DIAGRAM DESCRIPTION (Figure 3-34).

3-107. The maintenance panel (MP) provides facilities for manual application of power to the computer, monitoring of the computer operational status, manual control of program loading, and manual control of single instruction execution. The MP consists of controls and indicators used to master clear the computer logic circuits, perform maintenance operations, to manually insert data and addresses into the memory, and control diagnostic program load operations. The MP function consists of the maintenance panel (with the controls and indicators) and the interface logic (fig 3-35).



REFRESH MEMORY CYCLE

Figure 3-28. Memory Cycle Timing (Sheet 1 of 3)

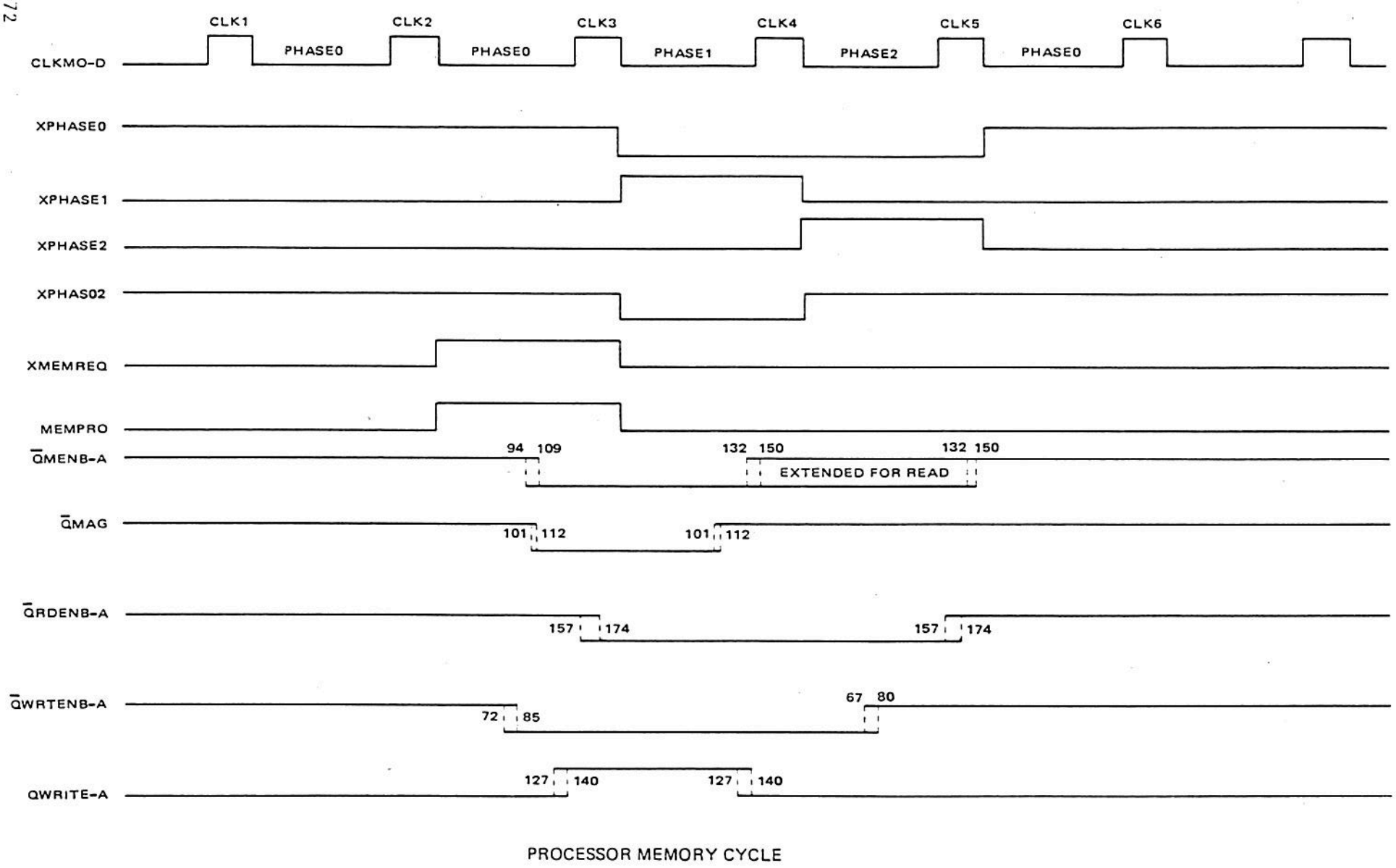


Figure 3-28. Memory Cycle Timing (Sheet 2 of 3)

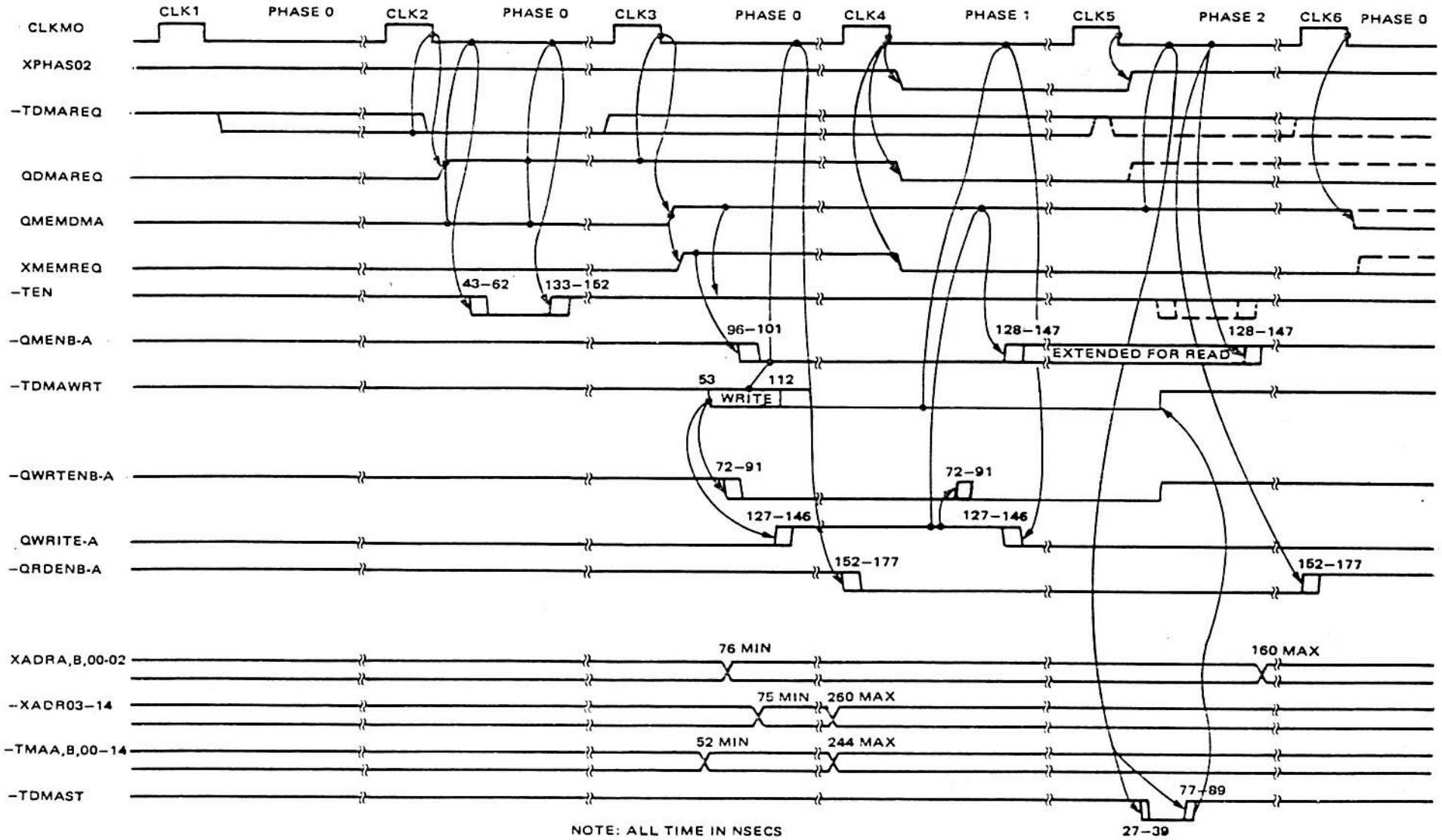
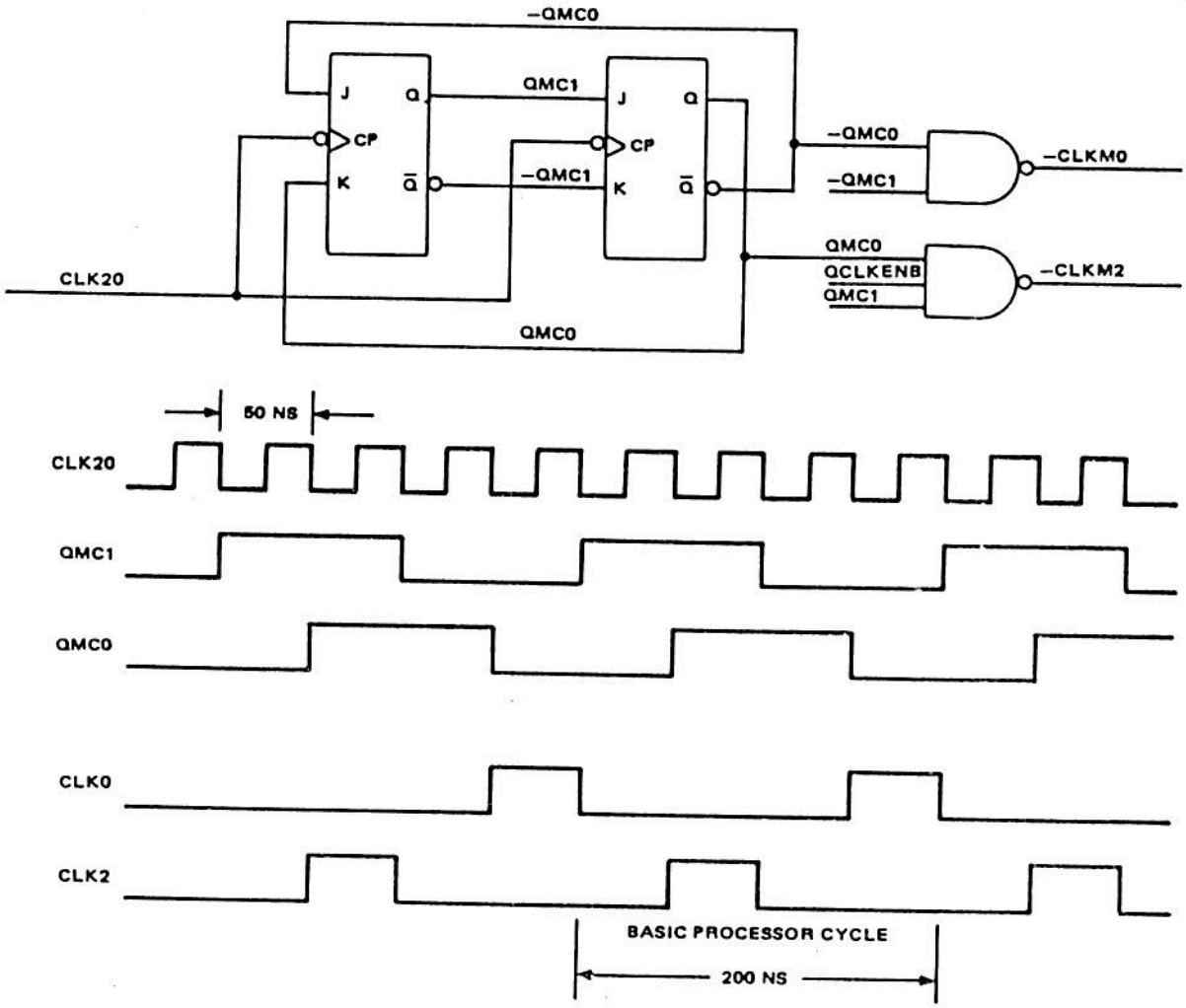


Figure 3-28. Memory Cycle Timing (Sheet 3 of 3)

DIVIDE BY 4 LOGIC (CLOCK OSCILLATOR = 20 MHz)



-CLKM0: FREE RUNNING 5-MHz CLOCK WITH A 50 NS ACTIVE LOW LEVEL DURATION

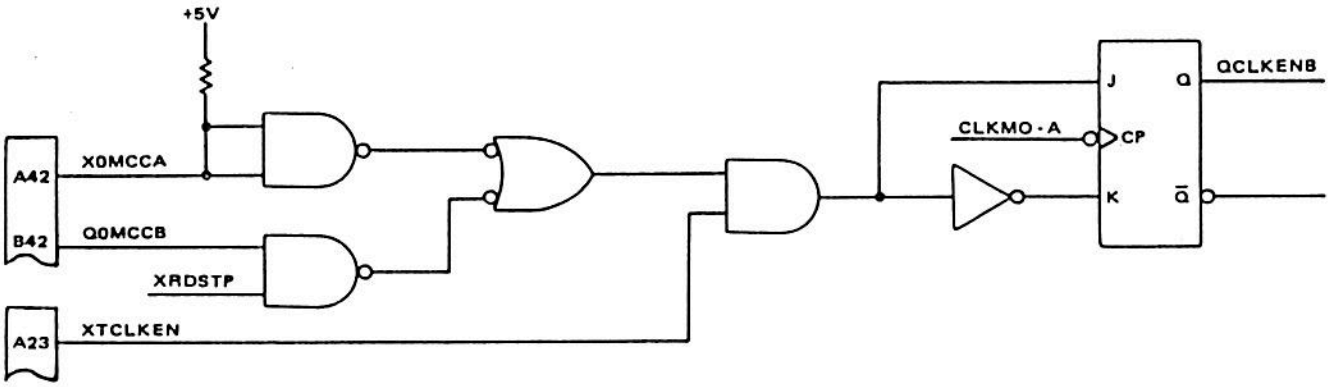
-CLKM2: FREE RUNNING 5-MHz CLOCK WITH A 50 NS ACTIVE LOW LEVEL DURATION (180° OUT OF PHASE [100 NS] WITH -CLKM0)

-CLK0: SAME AS -CLKM0 EXCEPT IT IS GATED WITH QCLKENB AND CAN BE DISABLED

-CLK2: SAME AS -CLKM2 EXCEPT IT IS GATED WITH QCLKENB AND CAN BE DISABLED

Figure 3-30. Clock Generation Diagram (Sheet 1 of 2)

QCLKENB - USED TO DISABLE -CLKO AND -CLKZ DURING POWER UP/DOWN SEQUENCING



- QCLKENB SETS ON TRAILING EDGE OF CLKMO
- $QCLKENB_{(SET)} = XTCLKEN \cdot (XQMCCA + Q0MCCB \cdot XRDSTP)$

TERMS

XTCLKEN - MUST BE SET HIGH TO ENABLE QCLKENB → CLKO, CLK2
- GENERATED ON IOC BOARD TO DISABLE CLOCKS DURING POWER SEQUENCING (UP/DOWN)

XTCLKEN IS NORMALLY HIGH
QCLKENB IS NORMALLY HIGH

Figure 3-30. Clock Generation Diagram (Sheet 2 of 2)

-XRDSTP - ACTIVE DURING MULTICYCLE OPERATIONS. RDR UPDATE INHIBITED
QSMTH - NORMALLY ACTIVE (FOR SINGLE CYCLE OPERATION). ACTIVE DURING FIRST CYCLE OF MULTICYCLE OPERATION.

EXAMPLE: I/O REQUIRES AT LEAST 3 CYCLES

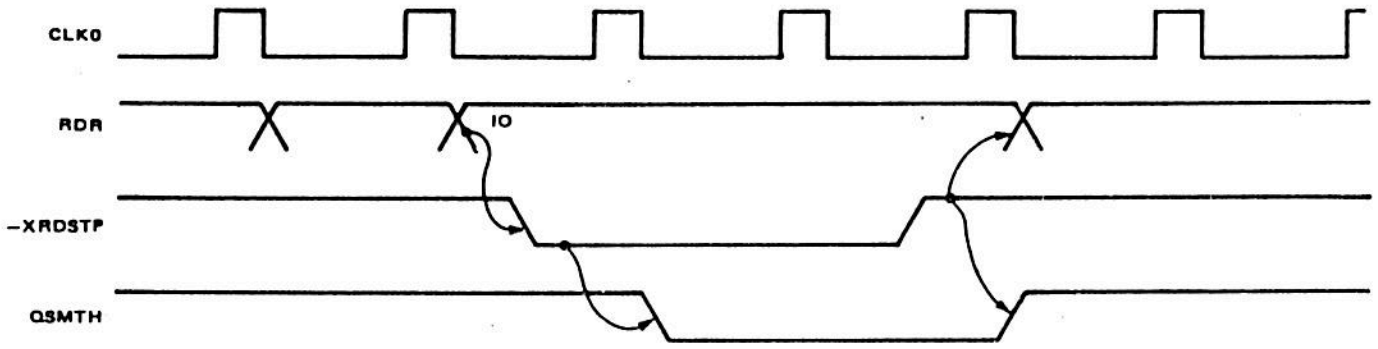
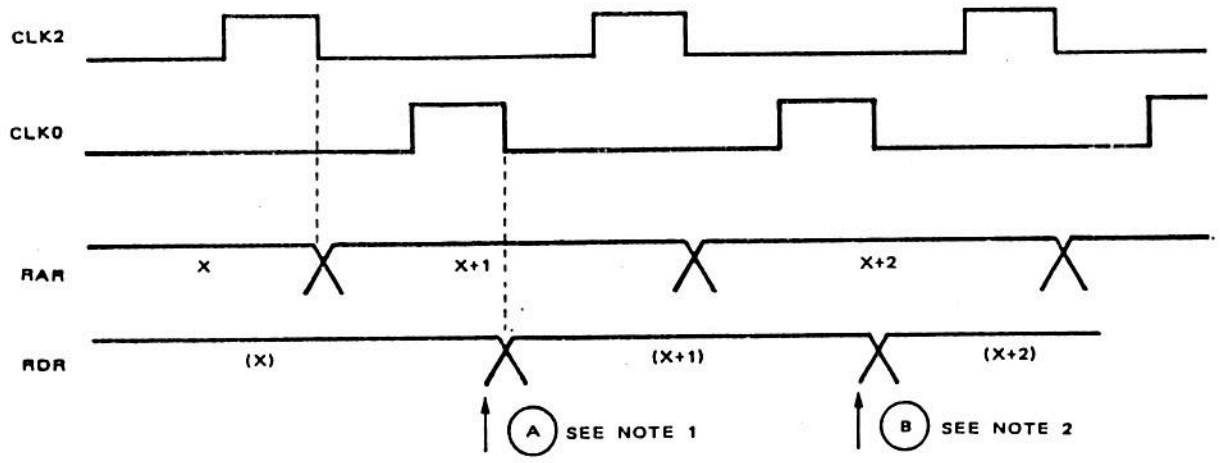


Figure 3-31. Processor Cycle Timing



RAR NORMALLY UPDATES WITH EACH CLK2 (SHOWN INCREMENTING IN TIMING DIAGRAM). THIS ADDRESS IS PRESENTED TO THE MICROINSTRUCTION ROM AS THE NEXT ADDRESS. THE ROM PRODUCES THE NEW MICROINSTRUCTION WHICH IS LOADED INTO THE RDR BY THE NEXT CLKO (100 ns LATER). THIS MICROINSTRUCTION IS EXECUTED DURING THE PROCESSOR CYCLE (SEE FIGURE 3-31) AND ANY DESTINATIONS ARE CLOCKED BY THE NEXT CLKO.

EXAMPLE:

ADDRESS	INSTRUCTION	
015	LI	REGO ← REG 1
016	AR	REG5 ← REG5 + REG 1

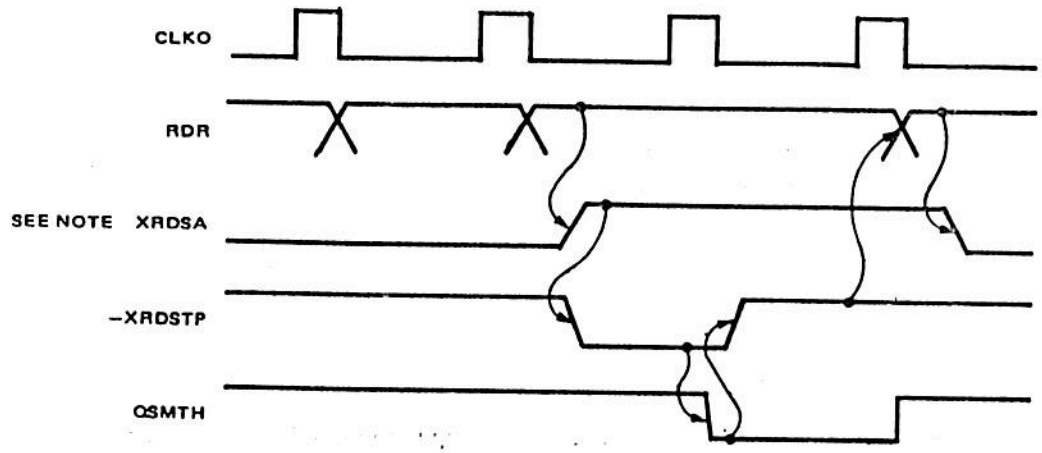
WITH REFERENCE TO THE ABOVE DIAGRAM:

X = 015
(X) = LI RO, R1
X + 1 = 016
(X + 1) = AR RS, R1

NOTES:

1. RO IS LOADED AT TIME (A)
2. R5 IS LOADED AT TIME (B)

Figure 3-32. Microsequencer Timing (Sheet 1 of 2)



XRDSA CONTROL TIMING

-XERDSTP - EXTERNAL CONTROL FOR XRDSTP. ACTIVATED BY EXTERNAL HARDWARE (i.e. I/O CONTROL HARDWARE ON IOC).

-XENABST - EXECUTES 2 CYCLE MICROINSTRUCTION WHEN PERFORMING INPUT REGISTER (WITH MOD) TO INPUT REGISTER.

NOTE: XRDSA IS ACTIVE FOR THE FOLLOWING 2 CYCLE MICROINSTRUCTIONS:

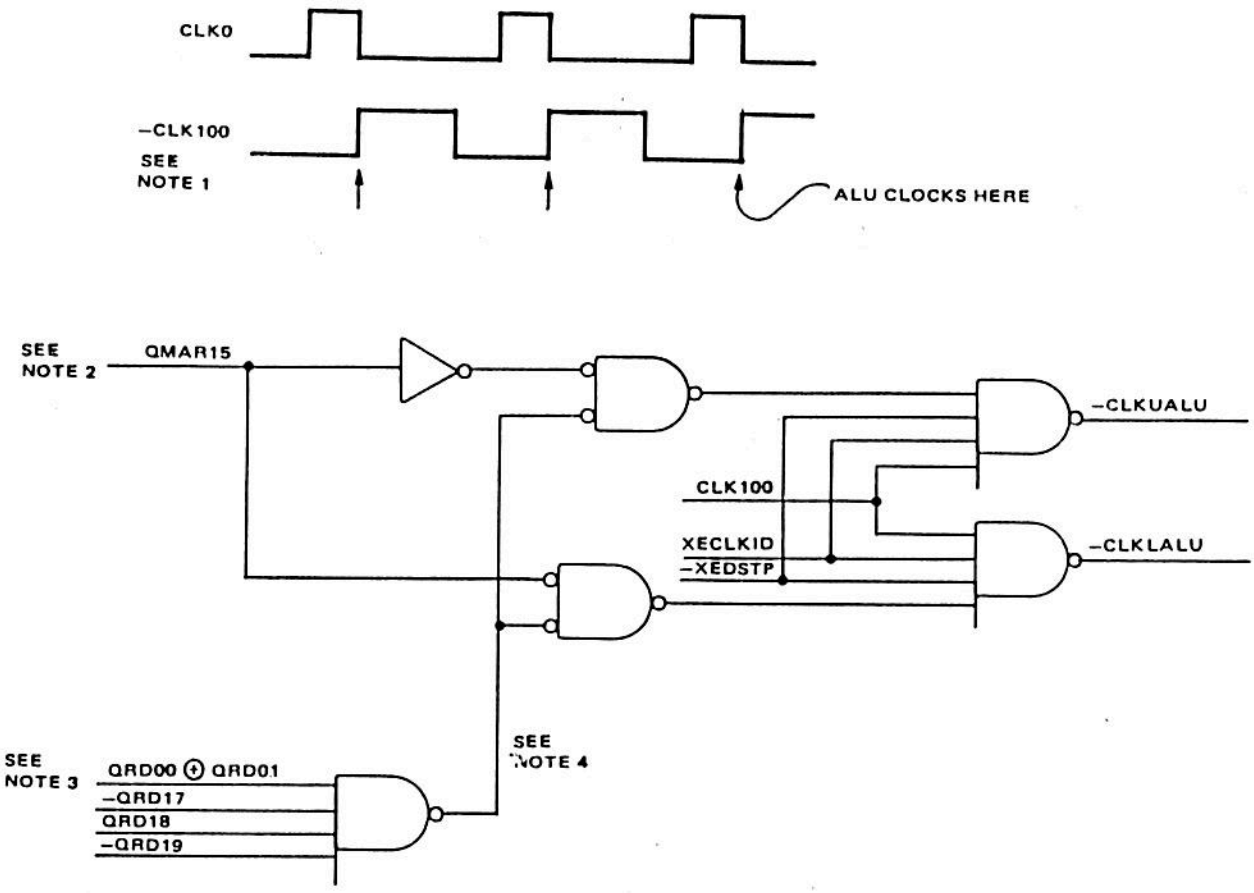
BLK

BIF (IF CONDITIONS MET)

EXTERNAL SOURCE = RAR

EXTERNAL DESTINATION = RAR

Figure 3-32. Microsequencer Timing (Sheet 2 of 2)



$$\text{CLKUALU} = \text{CLK100} \cdot \text{XECLKID} \cdot \text{-XEDSTP} \cdot \overline{(\text{QMAR15} \cdot \text{MOD-FIELD*})}$$

$$\text{CLKLALU} = \text{CLK100} \cdot \text{XECLKID} \cdot \text{-XEDSTP} \cdot \overline{(\text{-QMAR15} \cdot \text{MOD-FIELD*})}$$

*MOD-FIELD = BYTE INSERT

NOTES:

1. -CLK100 PROVIDES CONTROL TO CLOCK UPPER EIGHT BITS (UPPER TWO 2901's) INDEPENDENTLY OF LOWER EIGHT BITS.

   -CLK100 USED IN CONJUNCTION WITH BYTE INSERT MODE (EXECUTED AS DEFINED BY MOD-FIELD).

2. QMAR15 DETERMINES WHICH CLOCK IS DISABLED.

3. ACTIVE FOR RR, RI, AND IO MICROINSTRUCTIONS.

4. DISABLE TERM - DISABLES ONE OF THE ALU CLOCKS WHEN MOD-FIELD = 010 (BYTE INSERT).

Figure 3-33. ALU Clock Control