CENTRAL PROCESSOR UNIT (HP 1000 M-SERIES COMPUTER)

THEORY OF OPERATION

NOTE

This document is part of the HP 1000 M, E, and F-Series Computers Engineering and Reference Documentation and is not available separately.

TABLE OF CONTENTS

SECTION I

GENERAL

1. INTRODUCTION

This document is the Theory of Operation for the 21MX Computer series processing and control elements. Memory, power supply and option features are not included in the discussions, except as they directly affect operation of the system components described in detail. This discussion is conducted on a functional level using block diagrams and references to the schematic (F-5060-8352-S). Understanding of this theory is essential for doing detailed repair or troubleshooting of the computer.

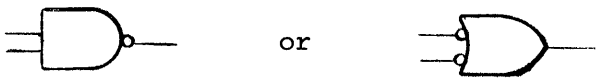
2. REFERENCE INFORMATION

2.1 Binary Signal Levels

Most logic used in the computer is implemented with standard or schottky TTL components, and positive logic is employed. High levels are +2.5 to +3.5V normally. Low levels are 0 to .8V. Some circuitry may depart from these values in special circumstances. The actual values to be expected may be determined from the type, load, and condition of the component. Logically, "high" is "true" and "low" is "false".

2.2 Logic Circuits

Logic circuits in the theory and on the schematic are drawn to aid in understanding the logical function. "Bubbles" on inputs or outputs indicate active low logic levels. For example, a NAND gate may be drawn as



depending on whether it is important that the output be low when both of the inputs are high, or that it be high when one of the inputs is low. A bubble on an "enable" input means the function is enabled when that input is low.

2.3 Signal Names

In general, signal names are alphanumeric identifiers beginning with a letter. The name may contain periods, numerals, and asterisks, and are chosen to be as close to signal usage as possible. Any signal name may have a bar over it to indicate that the signal is active low. For example, $\overline{\text{IBL}}$ is low when the IBL button is pressed, or alternately, is high when the IBL button is not pressed. A signal in the computer is considered "true" when it is high, whether or not its label includes a bar ($\overline{\text{IBL}}$ is true high, although it means "IBL button not pressed").

Buses are names by a sequence of letters, followed by a number indicating bit significance.

2.4 Cross-Referencing

This document references the CPU schematic (F-5060-8352-S). During the discussions, coordinates on the schematic are indicated by brackets. The coordinate will be that of an area of the schematic which contains the logic circuit described in the text.

2.5 Figures and Tables

Figures and Tables used in this theory are numbered according to the section to which they are most relevant. The first two numbers of a figure, separated by a period, indicate the section of text in which they will be found. A third number, following a second period, distinguishes between several figures in a section.

Example: Figure III.202.3 would be found in section III.202 as the third figure of that section.

2.6 Abbreviations

The following abbreviations occur frequently in the text. Their meanings

are given below:

ASG - Alter-Skip Group (machine instruction category)

DCPC - Dual Channel Port Controller (21MX option)

EAU - Extended Arithmetic Unit (machine instruction category)

EIG - Extended Instruction Group (machine instruction category including bit, word, byte, index register instructions).

FF - Flip-flop (single-bit storage element)

I/O - Input/Output

iff - if and only if (logical expression)

LED - Light-emitting diode (used for indicators on 21MX)

μ - Micro- (used as a prefix)

MP - Memory Protect (21MX option)

MRG - Memory Reference Group (machine instruction category)

ROM - Read-Only Memory (used in Control Store, Map Logic, etc).

SRG - Shift-Rotate Group (machine instruction category)

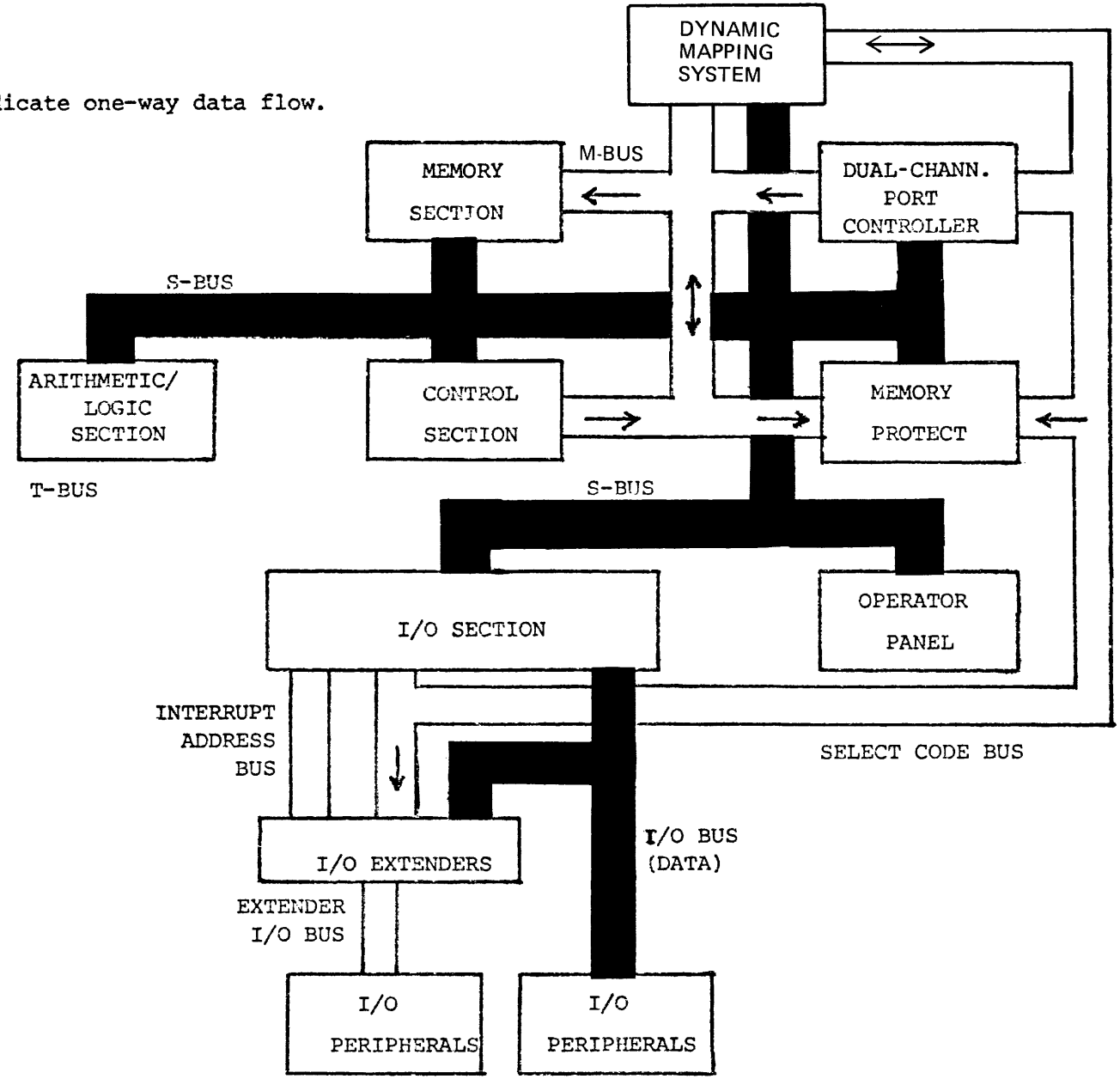
3. OVERALL BLOCK DIAGRAM

Figure 1.1 shows the bus structure of the 21MX with many of its normal options, showing data communications between sections, Figure 1.2 shows a generalized block diagram of the 21MX. Arrows indicate control and status signal paths between sections. This section describes the general tasks and interrelationships of each main section of the mainframe computer. The only sections to be covered by this theory are Control, Arithmetic/Logic, I/O, Power Fail, and Operator Panel.

3.1 Control Section

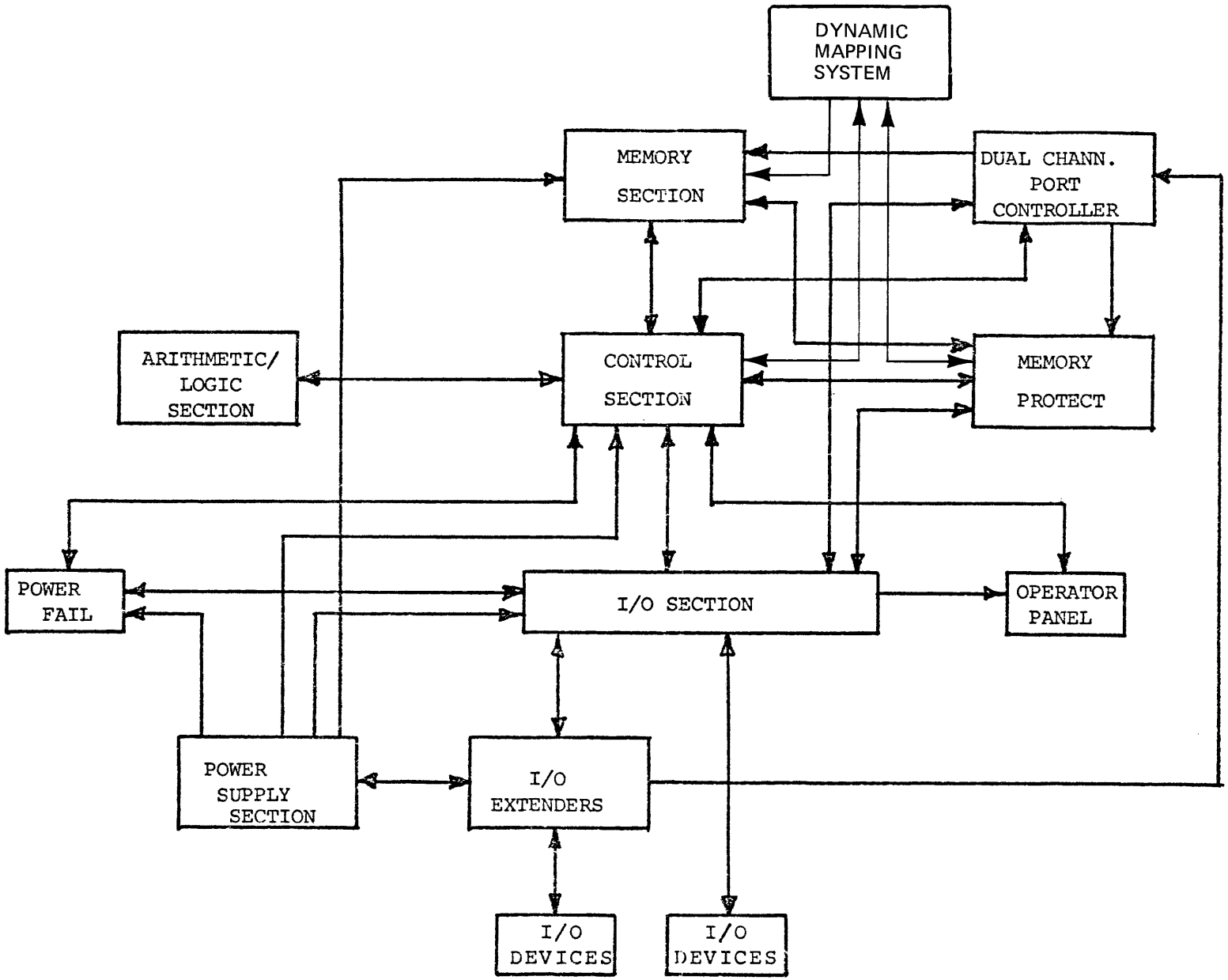
This section contains the registers and logic necessary to control or initialize all other sections, directly or indirectly. Some of the prime functions of this section are:

NOTE: Arrows indicate one-way data flow.



21MX BLOCK DIAGRAM SHOWING BUS STRUCTURE

FIGURE I.1



21MX BLOCK DIAGRAM SHOWING STATUS AND CONTROL PATHS

FIGURE I.2

1) control the execution sequence of microprograms
2) decode microinstruction fields
3) control data manipulations for the entire machine
4) initiate I/O signal sequences
5) control the Operator Panel
6) communicate with Memory Protect option
7) provide system clocks for all other system sections
8) perform main memory references
9) provide CPU synchronization with memory and I/O timing when required
10) provide hardware to allow effective execution of machine instructions

3.2 Arithmetic/Logic Section

This section contains all the working (scratch pad) registers of the machine and provides the logic to perform arithmetic and logical operations on data to be tested and/or stored onto the scratch pads. The prime elements of this section are:

1) a set of registers used by the machine for temporary storage of memory data
2) an arithmetic/logical unit which performs numeric and logical operations on data from many registers in the computer
3) a rotate-shifter, to further modify and manipulate data
4) some status registers to provide indications of results and aid in operations (Overflow, Extend, and special Flag)

3.3 I/O Section

This section provides the hardware link for communication with all peripheral devices. It includes the I/O card slots and the I/O interface cards that plug into them, and defines the nature of I/O device - computer interfacing. Its primary tasks are:

1) generate control signals to the I/O interfaces
2) resolve interrupt request priority conflicts
3) determine the select code of an interrupting device

4) signal the control section of pending interrupts
5) provide special communication lines to I/O extenders to allow more devices than could be loaded into the mainframe
6) provide control interfacing to special options, such as Dual Channel Port Controller, and Memory Protect
7) provide data and status paths for all I/O interfaces

3.4 Power Fail/Auto Restart Section

This is somewhat a part of the I/O section, but performs a very specialized task. This is a small state machine which receives power-up/down signals from the power supply. Its tasks are to:

1) generate a special interrupt from device code 4 (I/O interrupt system need not be enabled) when the power supply signals a power-up or power-down sequence -- this is a switch-selectable feature
2) generate a signal to enter the RUN mode if Auto Restart mode is desired
3) provide a testable flag to indicate whether power is going up or down

3.5 Power Supply Section

This section provides power to all mainframe components and some special purpose signals. Refer to the separate theory of operation for detailed operation of this section. The special signals generated by the power supply do the following tasks:

1) indicate whether memory was "lost" (memory voltages dropped too low) during a line power failure. Memory will not be lost if a sustaining battery preserves the memory voltages during a prolonged line power loss. Otherwise, if the line switch is turned OFF, this signal will flag the CPU that memory was lost when the line power is restored.
2) provide signal sequences to enable the computer to perform orderly power-up and power-down procedures; especially, to provide advance warning of impending power-loss

3) test the condition of the battery periodically and indicate it. This is done with a signal ($\overline{\text{FLASH}}$) which flashes the PF light on the Operator Panel if the sustaining battery voltage is low.

4) provide the ability to daisy-chain power control signals with I/O and memory extender units, so that when one device goes to STANDBY, or power is lost in one unit, then every power supply will follow suit. This is done by means of a cable which plugs into a PWR CONTROL plug on the back panel of each unit, and goes to special sequencing hardware on each supply.

3.6 Memory Section

This section consists of one of several possible versions of controller boards and a set of module boards with which it is designed to operate. The module boards contain semiconductor memory arrays, switch selectable to various address spaces. There is one controller for each of several different types of module boards, corresponding to different memory manufacturers. Only one controller will be in the memory section at a time. Several module boards can be controlled by the one controller. Detailed theory and operation of the memory section are covered in a separate document. The main tasks of this section are to:

1) sustain memory data. Since it is MOS memory, it suffers degradation of data, and must be refreshed. The controller initiates refreshes, and works them around DCPC cycles.

2) when issued READ or WRITE from the CPU or DCPC, perform the desired data reference. The memory address used is assumed to be on the M-Bus. Data is transferred between this address on the selected module board and the T-Register on the controller. The T-Register is the data link between the memory section and the CPU or DCPC.

3) on violation flags from Memory Protect, inhibit memory cycles from the CPU, but not from the DCPC.

4) When main power drops, or the machine goes to standby, sustain memory in a low-power mode.

3.7 Dual Channel Port Controller (DCPC) (Model 12897A)

3.7.1 This option consists of a PC board which plugs into the top slot (#110) of the memory backplane. Its function is to provide two channels, which are identical in operation, which can control block data transfers between almost any I/O device and memory. The detailed programming and operation of DCPC are covered in a separate theory of operation. The only discussion of it in this theory of operation is to detail its effects on the other main sections of the computer.

3.7.2 Each DCPC channel is loaded with a device select code, memory address, word count, and control information, using standard I/O instructions. It is then initiated using an STC instruction. Then whenever the programmed device with a select code matching that selected by a DCPC channel sets its Service Request (SRQ) line, DCPC does the following:

1) prevents the CPU from taking the next I/O cycle
2) controls the I/O system for one I/O cycle to issue control and data
3) send signals to memory and CPU to transfer data between memory and the I/O system
4) interrupt upon completion of the block transfer

3.8 Memory Protect Option (Model 12892A)

This option provides special system protection features. For detailed description, refer to its separate theory of operation.

It allows programmatic setting of a 15-bit "fence". When the MPCK micro-order occurs, a check on the address on the S-Bus is done. If S-Bus < Fence then a violation is flagged if the option was enabled through an I/O STC05. Its functions, once enabled, are:

1) set the violation flag if an MPCK test yields S-Bus < Fence
2) set a violation flag if an I/O instruction is attempted
3) request an interrupt if a violation occurs

4) if a violation occurs, prevent WRITE cycles in memory, and prevent I/O cycles and altering of P or S registers on the CPU

5) generate an interrupt request if a parity error occurs in memory. The interrupt will be serviced whether or not the interrupt system is enabled

3.9 Deleted

3.10 Bus System

There are several data and control paths which carry encoded information between major system elements. These are termed BUSES. Figure I.1 shows the communication paths the main 21MX buses provide. Following is a description of each one.

3.10.1 S-Bus. This is a 16-bit, tri-state, TTL-compatible bus. It is the main data transfer bus of the computer. When not driven high or low, it is pulled to a high state by pull-up resistors [A22-26]. It is found on the CPU board, the front panel, and the memory backplane.

3.10.2 T-Bus. This is a 16-bit, bi-state, TTL-compatible bus. It is internal to the Arithmetic/Logic section. Its only source is the output of the Shift/Rotate logic [F71-77], and its only destinations are the scratch pads, A and B registers and the TAB and TBZ logic [G77-78], [B37].

3.10.3 M-Bus. This is a 16-bit, tri-state, TTL-compatible bus. It holds the address to be referenced by memory. The M-Register on the CPU usually drives it [B-C46]. DCPC can transfer control of the M-Bus to the DCPC address registers with signal $\overline{\text{DMAEN}}$. Memory and Memory Protect are destinations only. DCPC and the control section are sources only.

3.10.4 I/O Bus. This is a 16-bit, bi-state, CTL-compatible data bus for the I/O system. All plug-in I/O interfaces transmit and receive data over the I/O Bus. There is a direction selectable buffer between the S-Bus and I/O Bus [A-G48]. The I/O Bus is pulled low when not driven [C-E46].

3.10.5 Select Code Bus. This is a 6-bit, tri-state, TTL-compatible control bus, which can be sourced by the CPU or DCPC. It holds the device select code for I/O system functions. It is driven by bits 5-0 of the Instruction Register, unless DCPC takes control of it. See Section III.306 for detailed discussion.

3.10.6 Interrupt Address Bus. This is a 6-bit, bi-state, CTL-compatible control bus, which contains the ones complement of the device select code of any interrupt-requesting I/O interface. It is encoded by the I/O system or an I/O extender from the IRQ line of the requesting device. It is the input to the Control Interrupt Register [B54]. It is pulled high if not driven [A54-55].

SECTION II

FUNCTIONAL OPERATION

1. COMPUTER TIMING

All timing for 21MX computers is controlled by a 18.5 MHz clock on the central processor board. Timing cycles for the central processor and I/O section are different, and are described separately in the following paragraphs. These sections operate asynchronously, until they must communicate with each other. Then, the central processor will "freeze" its operations until it is synchronized with the section in question. Memory and DCPC timing are discussed in their separate theories of operation.

1.1 I/O Section Timing

1.1.1 An I/O cycle has a duration of 1.62 μsec, and is the time required to generate all I/O signals required to execute an I/O instruction. The I/O cycle is divided into five 325-nanosecond periods designated T2, T3, T4, T5, T6.

1.1.2 A three-stage counter in the I/O section [A-C76] is clocked by P5 (non-freezable) from the central processor clock, and creates three timing signals TA, TB, TC, which are decoded into the 325-nanosecond periods T2, T3, T4, T5, T6, as shown in Figure II.1.1.

1.1.3 The computer I/O cycle always starts with T2 and ends with T6. Figure II.1.1 shows the times that the various I/O control signals are generated when an I/O cycle occurs.

1.2 Central Processor Timing

1.2.1 Central processor timing is derived from an 18.5 MHz crystal-controlled oscillator [E33] which clocks a three-stage ring counter [F35] every 54 nanoseconds. The counter is decoded from PAFF, PBFF, PCFF to provide six periods of 54 nanoseconds each, designated PØ, P1, P2, P3, P4, P5.

1.2.2 A central processor cycle has a duration of six P-periods, starting at PØ and ending with P5. This is the time required to execute one microinstruction.

1.2.3 Some decoded timing signals are continuous-running. Others may be frozen (inhibited) by central processor "freezes". Figure II.1.2 shows which timing signals are freezable. Freezes inhibit clocks at the end of one P2 to the end of another.

2. CENTRAL PROCESSOR FREEZE

2.1 If execution of a microinstruction is dependent on synchronization with some other element of the system, then at the end of P2, the CPU will disable certain clock signals to prevent completion of the microinstruction until the synchronization is completed. A freeze will perform the following functions:

a) prevent alteration of Arithmetic/Logical section registers
b) prevent alteration of ROM Instruction Register or ROM Address Register
c) prevent loading the Central Interrupt Register
d) Prevent alteration of Overflow, Extend, or Flag flip-flops
e) prevent the CPU from performing any operation to or from the S-Bus
f) prevent the CPU from sending READ or WRITE signals to memory
g) prevent obtaining of data from memory
h) prevent initiation of an I/O cycle

2.2 Below is a list of the timing considerations which are made to determine if a freeze is required [G33]. A freeze will occur at the end of P2 if any of the following conditions are met. The freeze will last until the end of P2 after synchronization has occurred or the conflict is resolved. See figure II.1.2 for the effect of a freeze on clock generation.

a) READ or WRTE micro-order in the OP field, and either memory is busy ($\overline{\text{MSRDY}}$ is low), a memory refresh is in progress ($\overline{\text{REFRESH}}$ low), or DCPC going to take a cycle ($\overline{\text{DMALO}}$ low).

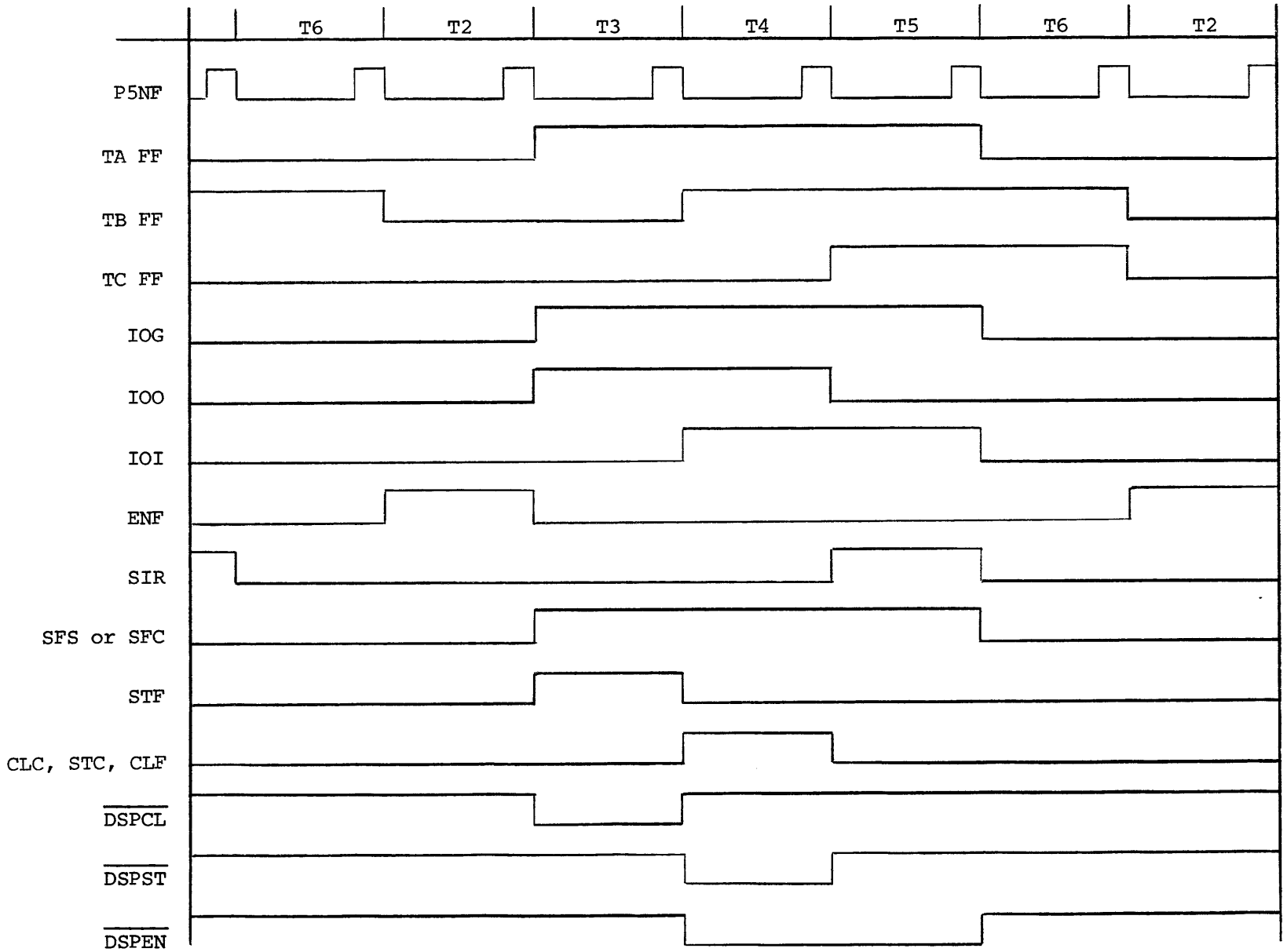
b) T or TAB micro-order in the S-Bus field, and memory is busy ($\overline{\text{MSRDY}}$ low).

c) IOG micro-order in the Special field, and either the I/O system is not ready (T2-period is required), or DCPC is going to take a cycle ($\overline{\text{DMALO}}$ low).

d) CIR is specified in the S-Bus field, and either it is not yet period T6, or DCPC is going to take a cycle ($\overline{\text{DMALO}}$ low).

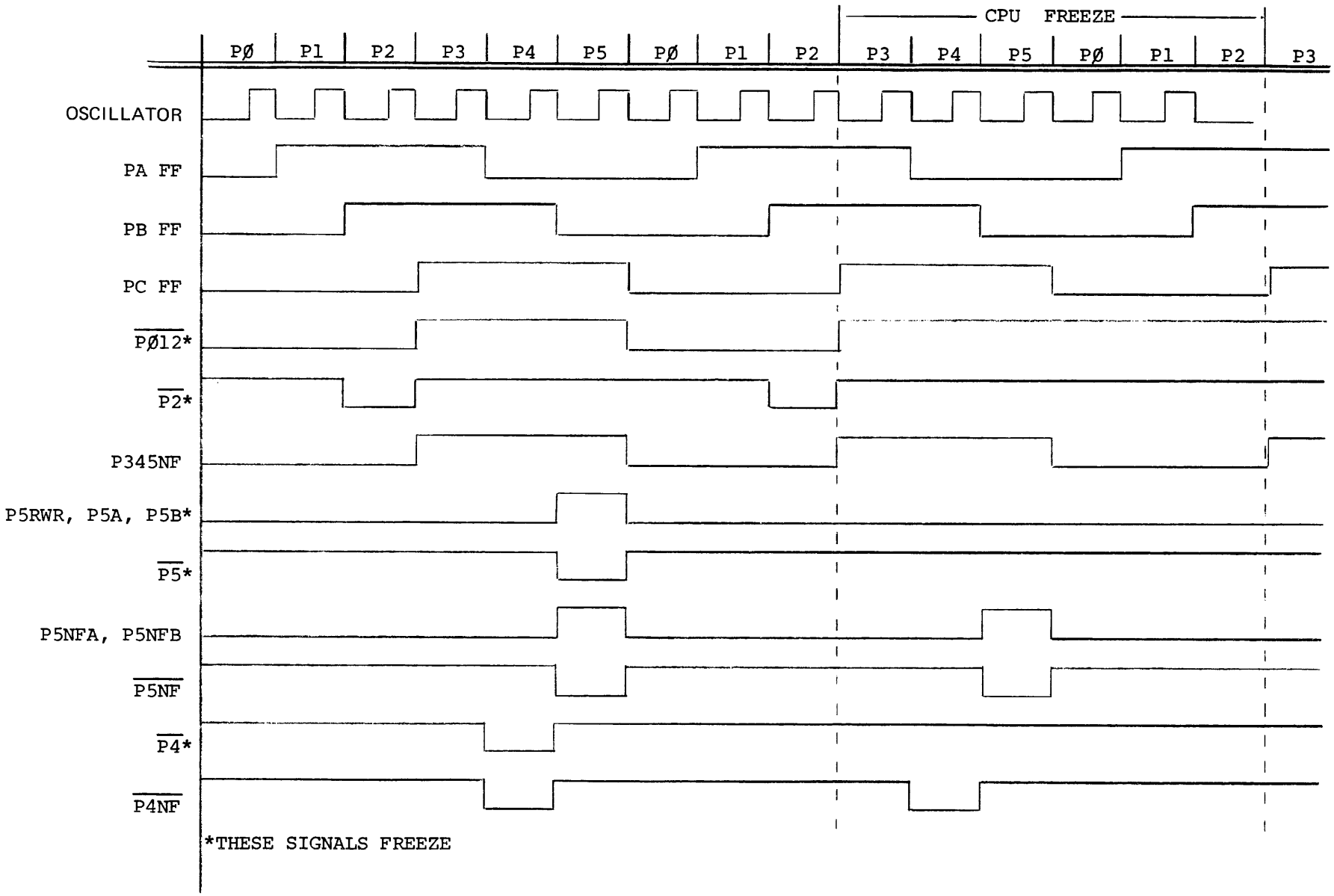
e) DCPC requires use of the S-Bus ($\overline{\text{DMAFRZ}}$ low at time T3).

f) P345NF will be inhibited if memory is doing refreshing. It is non-freezable otherwise. This prevents generation of IAK if a refresh occurs when the S-Bus microinstruction field specifies CIR [H52].



I/O TIMING

FIGURE II.1.1



CENTRAL PROCESSOR TIMING

FIGURE II.1.2

3. MICROINSTRUCTION EXECUTION

Execution of one microinstruction is concurrent with fetching of the next one. The microinstruction being executed is in the RIR. The address of the one being fetched is in the RAR.

3.1 Normal Timing

A microinstruction cycle is one T-period long, and is divided into six P-periods, PØ, P1, ... P5, each of 54 nanoseconds duration. At the end of P5, the RIR is loaded with the 24-bit output of Control Store. Simultaneously, the RAR is incremented by one, and starts accessing the next sequential location of Control Store. Before the micro-instruction is half over, the new Control Store data appears at the inputs to the RIR. It isn't loaded until P5, when this process repeats.

3.2 Non-Sequential Microinstruction Execution

3.2.1 Timing

By the end of P2 during execution, it has been determined whether a JMP, JSB, RTN, or JTAB is to be done and RIR2-Ø has selected the proper input to the RAR (see Section II.5.1-5.2, Word Type 4). Then the RAR is loaded during P2 with the address selected from the multiplexers [D11-20]. This destination address accesses Control Store until P5, when the data from this destination address is ready to load into the RIR. The RAR is incremented every P5, and loaded at P2 if a jump of some kind is indicated (refer to Section III.102).

3.2.2 JSB, RTN, JTAB Micro-Orders, or PON Low

If JSB occurs in the OP field, then the SAVE Register is loaded with the RAR at P2 [A12] concurrently with loading the destination address into the RAR. When RTN or JTAB micro-orders are used, or the PON signal goes low, then the SAVE Register is cleared at P5 [B12], after loading the RAR at P2 with a new address from the multiplexers. The implications of this are:

1) After JTAB or RTN, an RTN micro-order will result in a jump to location Ø. (No multi-level subroutine returns possible.)
2) As long as PON is low, $\overline{\text{PFIFF}}$ is low, making the ROM address multiplexer outputs all Ø. So RAR is loaded with Ø (or 4) each P2.

3.2.3 Interrupt Pending

If an interrupt is pending, or the RUN FF is cleared, then when the output of the RAR input multiplexers are all Ø, bit 2 will be set high. This provides a hardware trap to location 4 if a processor interrupt is pending and a jump or RTN to the fetch routine occurs.

3.3 Jumps and Status Conditions

3.3.1 During sequential operation, the conditions listed in Table II.3.1 are clocked into a register [E71] at the end of every P5 to preserve them for use in testing in subsequent conditional jump microinstructions. The ALU conditions must be recorded as they could change during the next instruction. $\overline{\text{MLOST}}$ is stored to minimize noise, as it comes from the power supply. $\overline{\text{INSTEP}}$ is stored to synchronize it with P5, so releasing the "INSTR STEP" button asynchronously won't interfere with performance of conditional jumps using this condition (e.g., see indirect routine in base set microcode).

3.3.2 If a JMP or JSB micro-order occurs, this condition register is not clocked, so a whole series of conditional jumps may be done based on a single ALU operation.

4. ACCESSING SCHEMES

This section describes the methods by which machine instruction codes are translated into addresses in Control Store where routines to execute them begin.

TABLE II.3.1

CONDITIONS STORED AT END OF MICROINSTRUCTION

SIGNAL	MEANING	SOURCE
$\overline{\text{INSTEP}}$	"INSTR STEP" BUTTON PRESSED IF SIGNAL LOW	FRONT PANEL
$\overline{\text{MLOST}}$	MEMORY POWER WAS LOST (LASTS ONLY 1/2 SECOND AT POWER UP) IF SIGNAL LOW	POWER SUPPLY
TBZ	T-BUS IS ZERO	T-BUS
ONES	ALU OUTPUT IS ALL ONES	ALU
COUT	CARRY GENERATED FROM THE ALU	ALU
ALU15	ALU BIT 15 IS ONE	ALU
ALUØ	ALU BIT Ø IS ONE	ALU

4.1 Control Store Organization

Logically, the Control Store addressing space is divided into 16 256-word modules (Ø-15), each module corresponding to a set of 256x4 bit IC ROMs (or six 256x1 WCS RAMs).

Module Ø contains all the 2100-type instruction routines and instruction fetching routines.

Module 1 contains the HALT routines, including the Boot Loader Initialization routine.

Module 14 contains Floating Point routines.

Module 15 contains the Extended Instruction Group routines.

4.2 Basic Instruction Decoding Scheme

The high order eight bits of the IR are inputs to a set of two 256x4 ROMs [D-E8], whose output corresponds to a Control Store address in module Ø. This address is loaded into the RAR at P2 when the JTAB micro-order is used (high four bits of RAR are loaded with Ø). This address is the start of the machine instruction routine if the IR contains an MRG, ASG, or SRG-type instruction, or a DIV, DLD, or DST instruction. Other instruction codes require further interrogation of bits to determine exactly their type, and hence, the starting address of their routine. For these instruction codes, the JTAB destination address contains a JMP microinstruction with a jump special micro-order specified, which will effect a modified jump, whose destination is partially determined by selected bits of the IR. Jump specials used in this scheme are JIO (IOG), JEAU, J74, J30 (refer to Section II.5). This whole process is illustrated in Figure II.4.1 and Table II.4.1 (refer to base set microcode listing also, for actual routines and jump formats).

4.3 Accessing Modules 2-15

Modules 2-15 are accessed as a result of the fetch jump sequence with instruction codes which JTAB to MACØ and MAC1. Modules 14 and 15 have slightly different schemes because they are adapted to efficient

accessing of Floating Point and EIG instruction routines, respectively. Refer to Figure II.4.1 and Table II.4.2.

The MAC jump tables MACTABLØ and MACTABL1 are tables of 16 jump instructions, each table starting at addresses whose low four bits are Ø. MACTABLØ contains jumps to addresses in modules 3-8, 14. MACTABL1 contains jumps to addresses in modules 2, 8-13, 15.

4.3.1 Primary Map Index

IR bits 7-4 define a primary map index. JMP J74 to MACTABLØ or MACTABL1 will determine a destination in the table which corresponds to a module (bits 7-5 of IR) and a secondary map index to use to jump to the modules (bit 4 of IR).

4.3.2 Secondary Map Index

The destination of the primary mapping contains one of two types of jumps:

1) IR4=0= even table address: an unconditional jump is done to address Ø of the module. At address Ø of the module the microprogrammer may place a jump to any other location in control store. He may do a JMP using J30 to his own jump table.
2) IR4=1= odd table address: a special jump to address Ø of the module is done using the J30 modifier. This results in IR3-0 determining a final destination of Ø-16 within the target module.

4.3.3 ROM Board Addressing

Each 1/2K ROM board is jumper-programmable to respond as any two consecutive (even then odd) modules (see Section III.109). 1K boards (using 512x8-bit ROMs) are programmatically selectable to any two pair of modules, each pair of which are consecutive (even then odd). Up to three ROM boards may be mounted side-by-side on the underside of the CPU board using stand-offs and screws.

4.3.4 Module 2-15 Accessing Summary

Except for modules 14, 15, which are reserved for the Floating Point and Extended Instruction Group routines, each other module has 16 distinct entry points with the current access scheme.

Address Ø has 17 instruction codes which map to it (IR4=Ø, or IR4=1 and IR3-Ø=Ø). This gives a total of 32 different instruction codes which map into each module. Note that for modules 2 and 8-13, that bit 11 doesn't matter. So for these modules, there are 64 instruction codes which map into each one (and 34 codes map through address Ø).

This yields a total of 608 instruction codes and 192 different entry points for microcode expansion beyond the base set.

TABLE II.4.1

JTAB DESTINATION ADDRESSES

IR BITS								DESTINATION ADDRESS	CONTENTS OF DESTINATION ADDRESS
15	14	13	12	11	10	9	8		
0/1	AT LEAST ONE "1"			X	X	X	X	ONE OF 2Ø	START OF MEMORY REFERENCE GROUP ROUTINES
0	0	0	0	X	0	0	X	SRG +1	START OF SRG ROUTINE IF 1ST SHIFT NOT ENABLED
0	0	0	0	X	0	1	X	SRG	START OF SRG ROUTINE IF 1ST SHIFT ENABLED
0	0	0	0	X	1	0	0	ASGNOP	START OF
0	0	0	0	X	1	0	1	ASGCL*	DIFFERENT VERSIONS
0	0	0	0	X	1	1	0	ASGCM*	OF ASG
0	0	0	0	X	1	1	1	ASGCC*	ROUTINES
1	0	0	0	X	1	X	X	IOG	JMP USING JIO MAP TO START OF I/O ROUTINES
1	0	0	0	0	0	0	1	DIV	START OF DIVIDE ROUTINE
1	0	0	0	0	0	X	0	EAU	JMP USING JEAU MAP TO EAU JUMP TABLE
1	0	0	0	1	0	0	0	DLD	START OF DLD ROUTINE
1	0	0	0	1	0	0	1	DST	START OF DST ROUTINE
1	0	0	0	1	0	1	0	MACØ	JUMP USING J74 TO MACTABLØ
1	0	0	0	X	0	1	1	MAC1	JUMP USING J74 TO MACTABL1

NOTE: "X" MEANS THIS BIT DOESN'T AFFECT THE MAPPING

FIGURE II.4.1

BASE SET INSTRUCTION FETCH SEQUENCE

(ARROWS INDICATE JUMP DESTINATIONS)

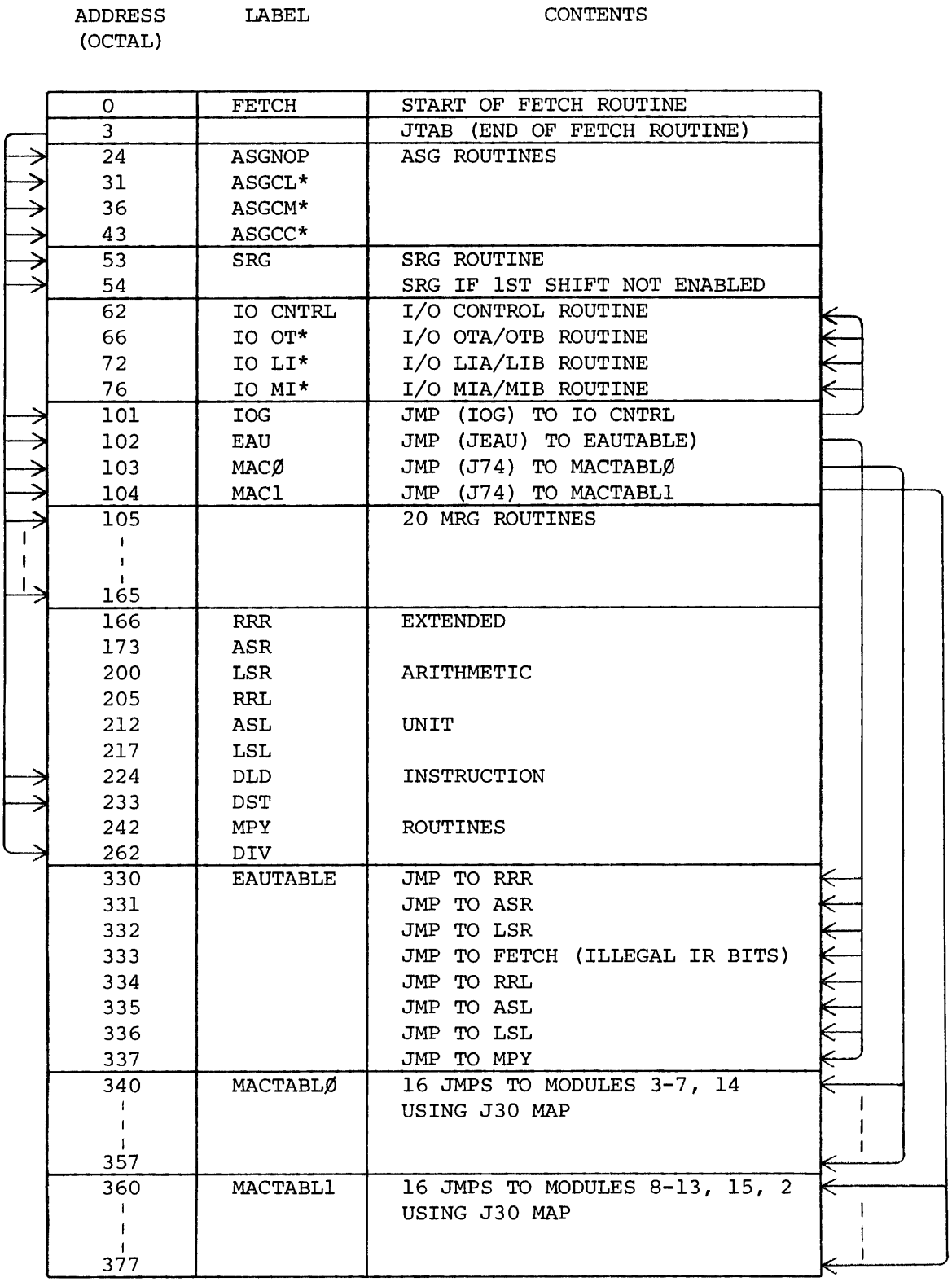


TABLE II.4.2

DETERMINATION OF MODULE ADDRESS USING IR FOR MAC-TYPE INSTRUCTIONS

TABLE SELECT								DISPLACEMENT IN TABLE				JUMP DESTINATION	USAGE BY HP
IR BITS													(CUSTOMER MAY USE ANY MODULE FOR WHICH HE DOES NOT HAVE AN HP OPTION INSTALLED)
JTAB								J74				J30	
JTAB DESTINATION								MODULE SELECT				ADDRESS SELECT	
15	14	13	12	11	10	9	8	7	6	5	4	3 2 1 0	
1	0	0	0	1	0	1	0	0	0	0	0	MOD 14, FADD	FLOATING POINT (STANDARD)
							0	0	0	0	1	MOD 14, FSUB	
							0	0	0	1	0	MOD 14, FMPY	
							0	0	0	1	1	MOD 14, FDIV	
							0	0	1	0	0	MOD 14, IFIX	
							0	0	1	0	1	MOD 14, FLOAT	
							0	0	1	1	0	MOD 3, 1400_8	
				MACTABLØ			0	0	1	1	1	MOD 3, 1400_8 + (IR3-Ø)	
							0	1	0	0	0	MOD 4, 2000_8	
							0	1	0	0	1	MOD 4, 2000_8 + (IR3-Ø)	
							0	1	0	1	0	MOD 5, 2400_8	
							0	1	0	1	1	MOD 5, 2400_8 + (IR3-Ø)	
							0	1	1	0	0	MOD 6, 3000_8	
							0	1	1	0	1	MOD 6, 3000_8 + (IR3-Ø)	
							0	1	1	1	0	MOD 7, 3400_8	
1	0	0	0	1	0	1	0	1	1	1	1	MOD 7, 3400_8 + (IR3-Ø)	
1	0	0	0	X	0	1	1	0	0	0	0	MOD 8, 4000_8	
							1	0	0	0	1	MOD 8, 4000_8 + (IR3-Ø)	
							1	0	0	1	0	MOD 9, 4400_8	
							1	0	0	1	1	MOD 9, 4400_8 + (IR3-Ø)	
							1	0	1	0	0	MOD 10, 5000_8	
							1	0	1	0	1	MOD 10, 5000_8 + (IR3-Ø)	
							1	0	1	1	0	MOD 11, 5400_8	
				MACTABL1			1	0	1	1	1	MOD 11, 5400_8 + (IR3-Ø)	
							1	1	0	0	0	MOD 12, 6000_8	RESERVED FOR CUSTOMERS EXCLUSIVELY
							1	1	0	0	1	MOD 12, 6000_8 (IR3-Ø)	
							1	1	0	1	0	MOD 13, 6400_8	
							1	1	0	1	1	MOD 13, 6400_8 + (IR3-Ø)	
							1	1	1	0	0	MOD 2, 1000_8	
							1	1	1	0	1	MOD 2, 1000_8 + (IR3-Ø)	
							1	1	1	1	0	MOD 15, 7400_8 + (IR3-Ø)	EXTENDED INSTR. GROUP (STD.)
1	0	0	0	X	0	1	1	1	1	1	1	MOD 15, 7420_8 + (IR3-Ø)	

5. MICROINSTRUCTION FORMATS AND EXECUTION

This section contains a description of the form of microinstruction word types, and a description of the form and effect of each microinstruction field within a word type.

A diagram of the form of each word type is in Table II.5.1. Table II.5.2 shows the binary and mnemonic form of each field of the microinstruction words.

5.1 Word Type 4

5.1.1 Word type 4 performs unconditional jumps or subroutine jumps to any location in the ROM address space. The only difference between JMP and JSB is that JSB causes the SAVE Register to be loaded at P2. At P2, the RAR is loaded with the 12-bit address specified in bits 16-5. However, the address will be modified before loading by one of eight mapping schemes determined by the Special Field.

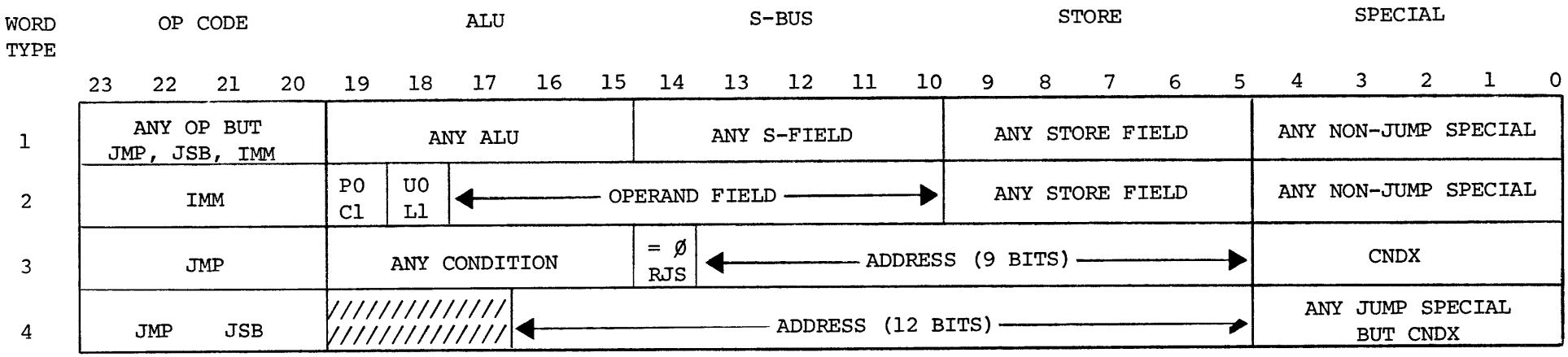
5.1.2 Work type 4 is determined by the presence of JMP or JSB in the OP Field, and the presence of any Special Field code except CNDX. However, specifying JTAB, RTN, or any other Special Field code which ends with -110_2 or -011_2 will not result in a direct JMP or JSB, but will load the JTAB or RTN address. This is due to the nature of the address mapping scheme, discussed below.

5.2 Address Mapping

5.2.1 The three low-order bits of the Special Field (RIR2-RIRØ) constantly control the ROM address mappers [D11-D2Ø], a bank of multiplexers which select one of eight possible sources to apply to the input of the RAR. This is designed to aid in implementation of various types of indexed jumps and returns from subroutines. These mapping schemes are selected as shown by the figure below. A discussion of each scheme and its use follows.

TABLE II.5.1

21MX MICROINSTRUCTION FORMATS



NOTES:

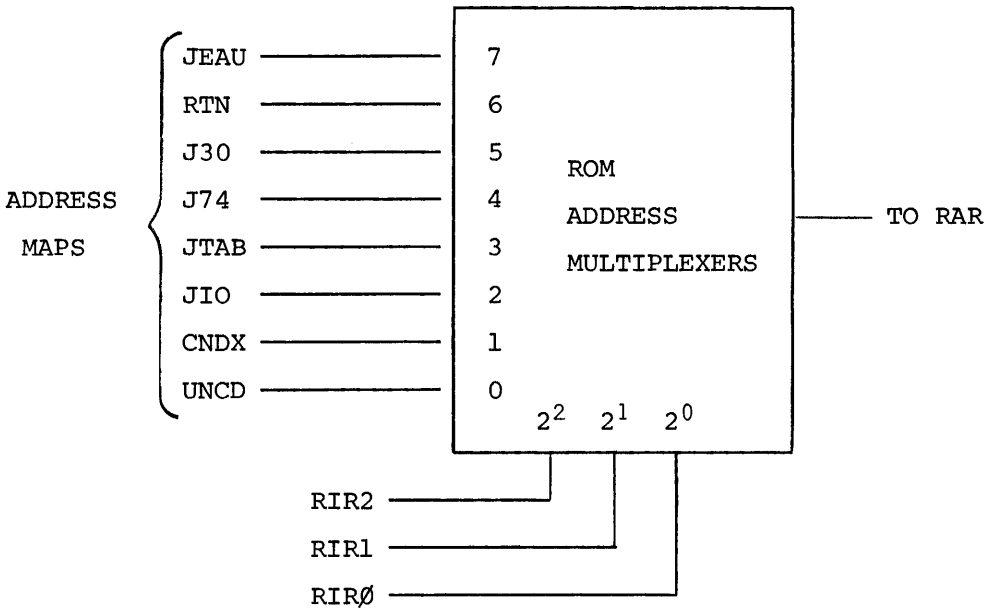
Word type 2. bit-19: if zero then ALU passes else ones complement.
bit-18: if zero then upper byte else lower byte.

Word type 3. bit-14: if zero then reverse jump sense.

TABLE II.5.2

21MX MICROINSTRUCTION MNEMONICS

ASSEMBLER CARD COLUMN ⟶	10	15	20	20	30	25	20
BITS POS. ⟶	20-23	0-4	15-19	15-19	10-14	5-9	18-19
BINARY	OP(4)	SPECIAL(5)	ALU(5)	COND(5)	S-BUS(5)	STORE(5)	IMM(2)
00000	NOP	IOFF	INC	TBZ	TAB	TAB	HIGH
00001	ARS	SRG2	OP1	ONES	CAB	CAB	LOW
00010	CRS	L1	OP2	COUT	T	T	CMHI
00011	LGS	L4	ZERO	ALØ	CIR	L	CMLO
00100	MPY	R1	OP3	AL15	IOI	IOO	
00101	DIV	ION	OP4	NMLS	CNTR	CNTR	
00110	LWF	SRG1	SUB	CNT8	DSPL	DSPL	
00111	WRTE	RES2	OP5	FPSP	DSPI	DSPI	
01000	ASG	STFL	OP6	FLAG	ADR	IR	
01001	READ	CLFL	ADD	E	M	M	
01010	ENV	FTCH	OP7	OVFL	B	B	
01011	ENVE	SOV	OP8	RUN	A	A	
01100	JSB	COV	OP9	NHOI	LDR	MEU	
01101	JMP	RPT	OP10	SKPF	RES2	CM	
01110	IMM	SRGE	OP11	ASGN	MEU	PNM	
01111	[BLANK]	NOP	DEC	IR2	NOP	NOP	
10000		MESP	CMPS	NLDR	S1	S1	
10001		MPCK	NOR	NSNG	S2	S2	
10010		IOG	NSAL	NINC	S3	S3	
10011		ICNT	OP13	NDEC	S4	S4	
10100		SHLT	NAND	NRT	S5	S5	
10101		INCI	CMPL	NLT	S6	S6	
10110		RES1	XOR	NSTR	S7	S7	
10111		SRUN	SANL	NRST	S8	S8	
11000		UNCD	NSOL	NSTB	S9	S9	
11001		CNDX	XNOR	NSFP	S10	S10	
11010		JIO	PASL	INT	S11	S11	
11011		JTAB	AND	SRGL	S12	S12	
11100		J74	ONE	RUNE	X	X	
11101		J3Ø	SONL	NOP	Y	Y	
11110		RTN	IOR	CNT4	P	P	
11111		JEAU	PASS	NMEU	S	S	



5.2.2 Note that since the Special Field is five bits wide, yet the mappers look at only the low three bits, that there are four different Special mnemonics which will select a particular address map. This is important for use in microprogramming.

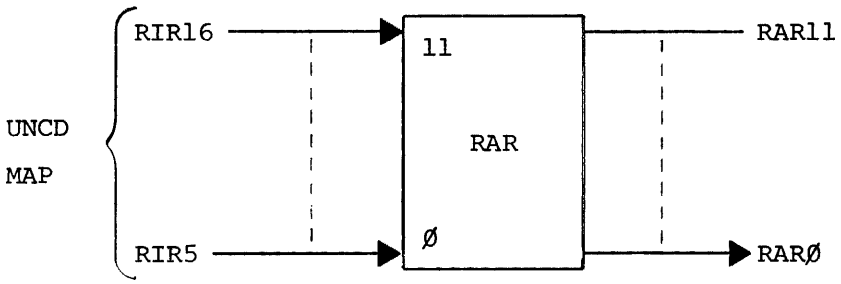
5.2.3 The mnemonics shown in the figure above are the Jump Specials. RTN (11110_2), JTAB (11011_2), and CNDX (11001_2) activate special logic. The other Jump Specials do nothing in the micro-instruction but select an address map. Refer to the discussion of the Special Field for RTN and JTAB. Refer to discussion of word type 3 for CNDX. The other non-jump specials will have their standard effect in addition to selecting an address map (refer also to the Special Field discussion of them).

5.2.4 The following discussion describes the mnemonics which will select each addressing map scheme, the uses of that scheme, and a detailed diagram of the derivation of the map.

UNCD (IOFF, STFL, MESP)

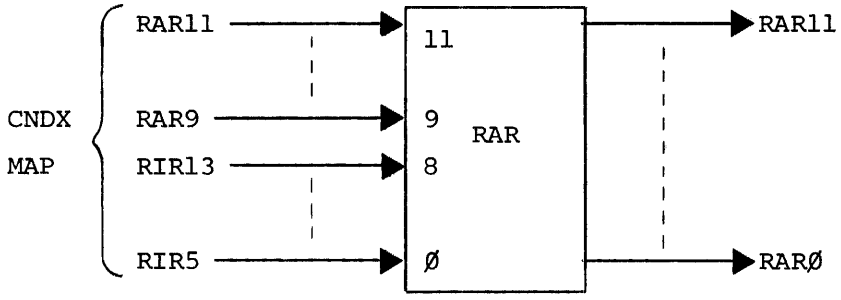
The address loaded into the RAR is the address specified in bits 16-5 of the RIR. JSB is used with IOFF specified in the Base Set microcode to turn off interrupt servicing

at the same time a subroutine jump is made to the indirect address routine for some machine instructions.



CNDX (SRG2, CLFL, MPCK)

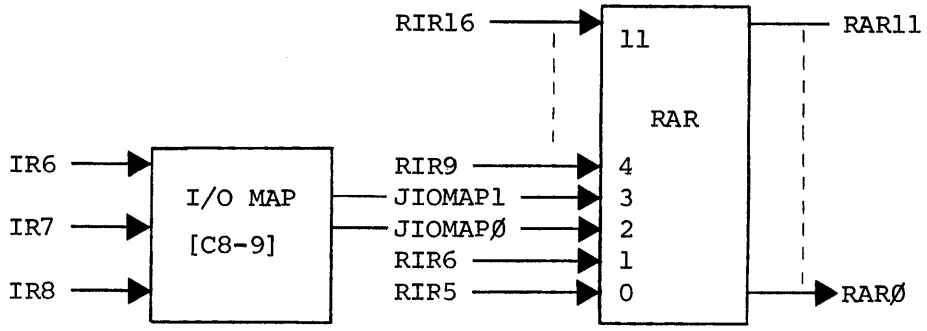
If CNDX is specified, the microinstruction is word type 3. If any of the other specials are specified, then that special will be executed and the CNDX map will be used to load the RAR with a 9-bit address within the current 512-word section specified in the RAR (RAR11-RAR9 are unchanged).



JIO (LI, FTCH, IOG)

Cause RAR to be loaded with the address in RIR16-RIR5, except that RAR3 and RAR2 are determined by the combination of IR bits 6, 7, 8. The high 8 bits of IR do not fully specify the type of I/O instruction being executed. For I/O instructions, the fetch routine causes all machine I/O instructions to jump to one location. From there,

a JMP with IOG is done to jump to the appropriate I/O routine and initiate the I/O control signals.



Below is a table which shows how the I/O map works for all the I/O instructions.

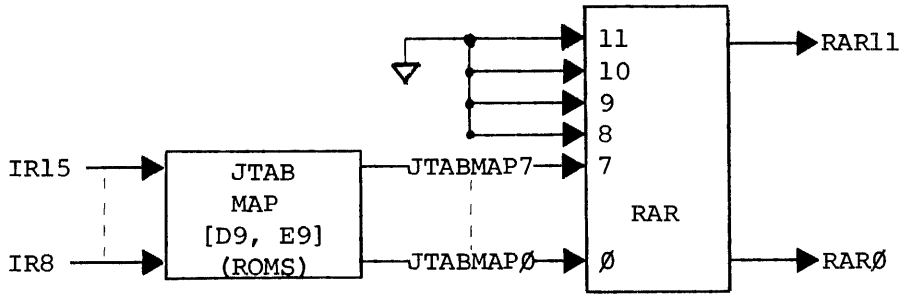
IR BITS 876	I/O INSTRUCTION	JIOMAP1	JIOMAPØ
100	MI*	1	1
101	LI*	1	0
110	OT*	0	1
XXX	ALL OTHERS	0	0

When using this map, the destination address in the address field must end with one of the following combinations of low order bits: 20_8-23_8, 40_8-43_8, 60_8-63_8, 00_8-03_8. The I/O routines are 4 instructions each.

JTAB (ICNT, SOV, L4)

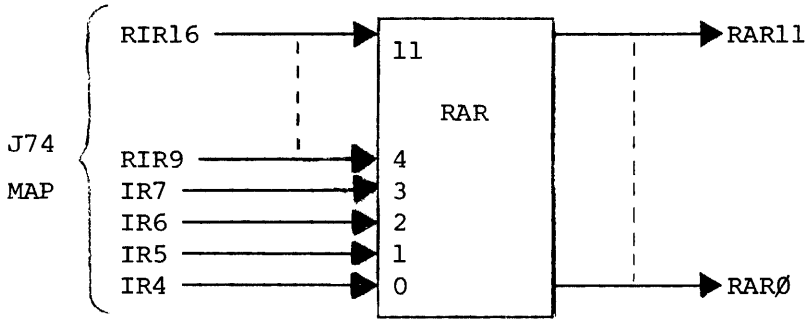
The input to the RAR is determined by the contents of the high 8 bits of the IR. The IR provides an 8-bit address to the ROM look-up table [D9, E9]. This map is used to load the RAR with the starting address of the microcode

routine for the machine instruction in the IR. JTAB will cause the SAVE Register to be cleared, but any of the other listed specials won't.



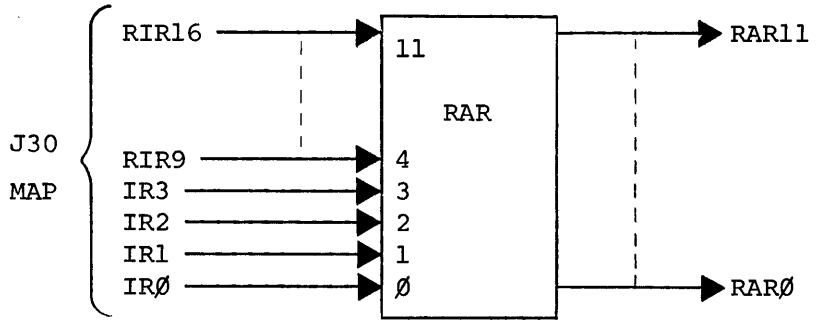
J74 (R1, COV, SHLT)

The address specified in RIR bits 16 through 9 is loaded into RAR11 through RAR4. The low four bits of the RAR are loaded from IR7 through IR4. This is used as the primary MACtable jump. The fetch routine determines from IR15-IR8 whether the IR contains a non-base set macro ($105XXX_8$ or 1 000 X01 1XX XXX XXX_2) and whether it is in the first or second 8 ROM modules. JTAB will cause a jump to MAC1 or MACØ in the Base Set. There, JMP with J74 is done to MACTABLØ or MACTABL1. Refer to section II.4.



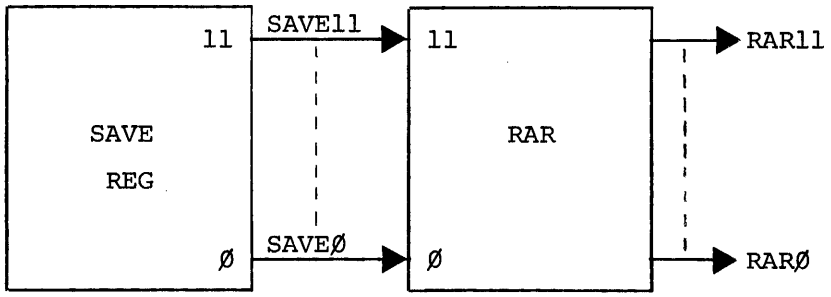
J3Ø (ION, RPT, INCI)

The address in RIR16-RIR9 is loaded into RAR11-RAR4. RAR3-RARØ are loaded from IR3-IRØ. This is used as the secondary MACtable jump. The last 32 locations of module Ø are used to jump to all the other ROM modules. J30 is specified with JMP if the macro is to jump indexed by IR3-IRØ into the ROM module. The microprogrammer will find J30 handy in programming his own routines. Refer to the section on microcode accessing schemes, II.4.



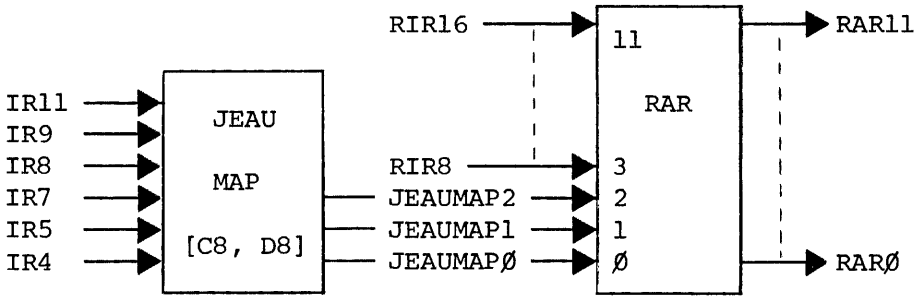
RTN (SRG1, SRGE, RES1)

The contents of the SAVE Register are loaded into the RAR.



JEAU (Res2, NOP, SRUN)

The contents of RIR16-RIR8 are loaded into RAR11-RAR3. RAR2-RARØ are loaded according to the combinations of IR4, 5, 7, 8, 9, 11. This map is used to jump to specific Extend Arithmetic Group instruction routines. The fetch routine JTABs to one location (EAU) for all EAG instructions. There, JMP with JEAU is used to jump to another table (EAUTABLE) from which a JMP to the specific EAG routine is done.



Below is a table of the basic functioning of the JEAU map. Also refer to Table III.108.2

INSTRUCTION	JEAUMAP2	JEAUMAP1	JEAUMAPØ
RRR	0	0	0
ASR	0	0	1
LSR	0	1	0
ILLEGAL	0	1	1
RRL	1	0	0
ASL	1	0	1
LSL	1	1	0
MPY	1	1	1

The DIV, DLD, DST instruction routines are jumped to directly using JTAB. If the illegal EAUMAP occurs due to an illegal instruction in the IR, the destination of the jump in control store contains a jump to the fetch routine.

5.3 Word Type 3

5.3.1 Word type 3 is distinguished by JMP in the OP Field, and CNDX in the Special Field. RAR8-RARØ are loaded during P2 with the address specified in RIR13-RIR5 (RAR11-RAR9 unchanged) if the condition specified in the Condition and RJS (Reverse Jump Sense) Fields is met (see section on Conditional Logic). If RJS (RIR14) is Ø, then the JMP is taken if the condition is not met [A74].

5.3.2 Condition field mnemonics are described below. Read carefully, as some conditions are met when the signal being tested is false. The condition causing JMP to occur (for RJS=1) is defined after the condition field mnemonic. The actual signal name which is input to the conditional logic on the CPU [area around C73] is included in parenthesis if it is different than the mnemonic of the microinstruction field.

CONDITION MNEMONICS

TBZ (TBZF)	The T-Bus was $ØØØØØØ_8$ after the last word-type 1 or 2 μ-instruction.
ONES (ONESF)	The output of the ALU was 177777_8 after the last word-type 1 or 2 μ-instruction.
COUT (COUTF)	There was a carry out of the ALU after the last word-type 1 or 2 μ-instruction.
ALØ (ALUØF)	ALU bit Ø was 1 after the last word-type 1 or 2 μ-instruction.
AL15 (ALU15F)	ALU bit 15 was 1 after the last word-type 1 or 2 μ-instruction.

NMLS ($\overline{\text{MLSTF}}$) Memory power was not lost (condition not met after a power failure where memory power is lost). Memory should be good if met.

CNT8 The Counter output is 11111111_2.

FPSP Front Panel Special Test. No jump for standard front panel. Signal grounded on the standard front panel.

FLAG (FLAGFF) The FLAG FF is set.

E (EXFF) The EXTEND FF is set.

OVFL (OVERFF) The OVERFLOW FF is set.

RUN (RUNFF) The RUN FF is set.

NHOI ($\overline{\text{HOI}}$) There is no interrupt pending and the RUN FF is set.

SKPF An I/O instruction skip condition (SFS, SFC was met. Met only during I/O control-type instructions (see I/O Timing, I/O section).

ASGN ($\overline{\text{ASGN}}$) Microinstruction skip conditions is met if ASG instruction skip conditions are not met. Meaningful only when executing ASG instructions. $\overline{\text{ASGN}}$ is low if the conditions for an ASG skip <u>are</u> met. These skip conditions are decoded and tested by the multiplexer and gates around [E75] on the CPU schematic.

For ASG instructions the IR bits which specify skip tests are:

IRØ: RSS -- Reverse skip sense
IR1: SZ(A/B) -- Skip if A/B Register is zero
IR2: IN(A/B) -- Increment the A/B Register
IR3: SL(A/B) -- Skip if least significant bit (bit Ø) of A/B is zero
IR4: SS(A/B) -- Skip if sign bit (bit 15) of A/B is zero
IR5: SEZ -- Skip if EXTEND bit is zero

ASGN tests for all skip conditions at once. So there are 2 conditions which cause the SZ(A/B) test to succeed: either IN(A/B) is to be performed and the A/B Register is all ones, or IN(A/B) is not to be performed and the A/B Register is all zeros.

Then the equation of the skip ($\overline{\text{ASGN}}$ is low) is:

SKIP($\overline{\text{ASGN}}$ low) = ((TBZF•IR2+ONESF•IR2)⊕IRØ)•IR1
+ ($\overline{\text{EXFF}}$⊕IRØ)•IR5 + MULTIPLEXER SKIP

The ASG multiplexer [E74] output indicates a skip according to the following combinations of IR4, IR3, and IRØ:

IR4 (SS*)	IR3 (SL*)	IRØ (RSS)	CONDITION CAUSING SKIP
0	0	0	NONE
0	0	1	$\overline{\text{IR5}}$•$\overline{\text{IR1}}$
0	1	0	$\overline{\text{ALUØF}}$
0	1	1	ALUØF
1	0	0	$\overline{\text{ALU15F}}$
1	0	1	ALU15F
1	1	0	$\overline{\text{ALUØF}}$+$\overline{\text{ALU15F}}$
1	1	1	ALUØF•ALU15F

IR2 — IR Bit 2 is high.

NOTE: The following 8 mnemonics test the state of front panel buttons. The conditions are met if the buttons are <u>not</u> depressed.

NLDR($\overline{\text{IBL}}$): IBL button

NSNG($\overline{\text{INSTEP}}$): INSTR STEP button

NINC(INCM): INC M button

NDEC($\overline{\text{DECM}}$): DEC M button

NRT($\overline{\text{RIGHT}}$): RIGHT button

NLT($\overline{\text{LEFT}}$): LEFT button

NSTR($\overline{\text{STORE}}$): STORE button

NRST($\overline{\text{DISPLAY}}$): DISPLAY button ("Restore")

NSTB ($\overline{\text{STROBE}}$)	None of the front panel buttons are depressed.
NSFP ($\overline{\text{SFP}}$)	Non-standard front panel installed. The $\overline{\text{SFP}}$ signal is grounded by the front panel PCA (part no. 5060-8343). In the event the user is using his own front panel, but is using the Hewlett-Packard base set microinstruction (ROM), the base set can determine this if his front panel does not ground pin 30 of J1 ($\overline{\text{SFP}}$). The user can then use this to jump to a special set of microcode to control the action desired by manipulating his front panel.
INT	An interrupt is awaiting service.
SRGL	IR3 is "1" and the ALU bit Ø was 1 after the last word type 1 or 2 μ-instruction. Used as part of the SRG routine to test for skip condition.
RUNE (RUNEN)	The operator key switch is not in the LOCK position.
NOP (GROUND)	Jump if RJS = Ø.
CNT4	The low-order 4 bits of the counter are 1111_2.
NMEU	Special condition reserved for use by memory management.

5.4 Word Type 2

5.4.1 This type of microinstruction is distinguished by IMM in the OP Field. The 8-bit literal in RIR17-RIR1Ø is gated onto the S-Bus [C42- D42] and stored into the register specified in the Store Field at the end of the instruction cycle.

5.4.2 RIR18 specifies whether the literal is to be put onto the high (Ø) or low (1) eight bits of the S-Bus [D42]. The other half of the S-Bus is all ones, as it is not being driven by any gates.

5.4.3 RIR19 specifies whether the S-Bus is to be ones complemented (RIR19=1) through the ALU or passed (RIR19=Ø), [G70, G27]. If the store field specifies a register which is loaded off

the S-Bus, then the data can not be complemented before storing into the register. (The ALU will still complement however.)

5.4.4 The Special Field is executed as in word type 1 microinstructions.

5.5 Word Type 1

5.5.1 This type of microinstruction is used to perform all arithmetic, logical, I/O, and memory operations, and to manage communication with special options like memory protect and memory management.

5.5.2 The signals specified in the OP, ALU and Special Fields are asserted at the output of the field decoders during the microinstruction cycle. They are clocked or gated as described in Section 5.6 to function properly with the system.

5.5.3 During PØ-P2, RIR1Ø-RIR13 of the S-Bus Field are selected [A21] to address the Scratch Pad RAMS [B-G22]. The output of the RAMs is latched into the Holding Register [C-F24] at the end of P2. This data is driven onto the S-Bus [C-F25] if RIR14 is high. RIR14=1 in the S-Bus Field determines that the S-Bus is to be driven by the Scratch Pad Registers (S1-S12, X, Y, P, and S). RIR14=0 in the S-Bus Field selects one of the discrete registers to gate onto the S-Bus.

5.5.4 If a timing or resource conflict arises, the FRZ FF [G37] (freeze) is set at the end of P2, disabling most CPU clocks, effectively "freezing" the microinstruction and preventing its completion. FRZ FF is reset at the end of P2 following the end of the conflict, and the instruction may go to completion.

5.5.5 During Direct Memory Access cycles, $\overline{\text{DMAFRZ}}$ is low during T3. This signal inhibits the S-Bus Field decoder [B41] and prevents the CPU from driving the S-Bus [E25], so that DCPC may use it. Unless a word type 3 or 4 is being performed at this time, the processor freezes for one cycle.

5.5.6 During P3-P5, RIR8-RIR5 are selected [A21] to address the Scratch Pad RAMs, to select which register to store the T-Bus into. The Store Field is disabled by JORJ [C33, D34] if JMP or JSB occurs in the OP Field, preventing unselected register alteration.

5.5.7 At P5, the selected register is loaded with the data at its inputs. $\overline{\text{RAMWEN}}$ [D34] is low during P5 to load the T-Bus into the Scratch Pads if RIR9 is 1 or if PNM is specified in the Store Field. It is disabled from going low during a CPU freeze, JMP or JSB in the OP Field, or when a Memory Protect Violation (MPV) occurs and the P or S Register is selected by RIR8-RIR5.

5.5.8 The Special Field is decoded [B-D62] unconditionally as long as the microinstruction is in the RIR. Decoder outputs are used for a variety of control functions, and are used at different time periods. Refer to Section 5.6.2 for detailed description of Special Fields.

5.6 Micro-orders

This section contains a detailed description of the function or effect of each microinstruction mnemonic except for the Condition Field (see Section 5.3.2), and the Jump Specials (see Section 5.2.4). First the mnemonic is given, then the CPU signal name of it in parenthesis, followed by a description of its effect.

5.6.1 OP FIELD

Refer to the detailed theory of the Arithmetic/Logical section for circuit locations and further discussions of some of the arithmetic operations below. The OP Field decoder is at [G62].

NOP ( ) No connection on CPU. No effect.

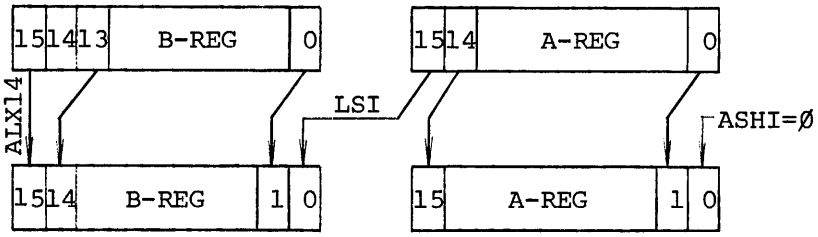
ARS ($\overline{\text{ARSOP}}$) 32-bit arithmetic shift. The microinstruction must be in the form

ARS	PASS	B	B	L1 OR R1
OP	ALU	S-BUS	STORE	SPEC

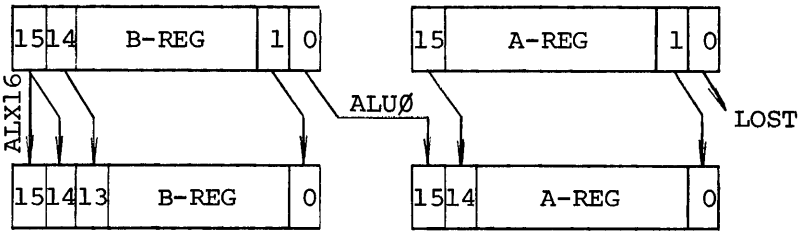
a) If L1: AS1=1, ASØ=Ø, which shifts the A-Register left 1. The B-Register is passed through the ALU, and is shifted left 1 through the shifter (TBS1-Ø, TBSØ=1), with ALX14=ALU15 and LSI=AR15. The T-Bus is then stored into the B-Register. Overflow is set if ALU14≠ALU15 (sign ≠ original B-Register bit 14). The effect is shown below.

b) If R1: AS1=Ø, ASØ=1, which shifts the A-Register right 1, with B-Register Ø → A-Register 15. A-Register Ø is lost. The B-Register is passed through the ALU, and is shifted 1 right through the shifter (TBSØ=Ø, TBS1=Ø), with ALX16=ALU15 (sign extended shift). The T-Bus is then stored in the B-Register. The effect is shown below.

ARS with L1:



ARS with R1:



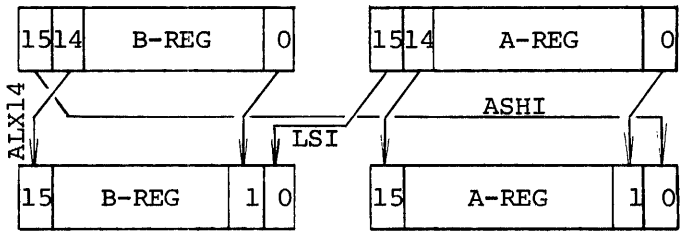
CRS ($\overline{\text{CRSOP}}$) 32-bit circular shift. Requirements:

CRS	PASS	B	B	L1 OR R1
OP	ALU	S-BUS	STORE	SPEC

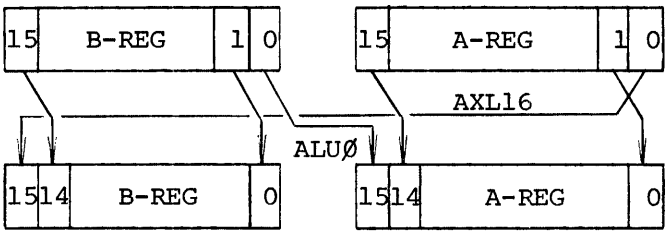
a) If L1: AS1=1, ASØ=Ø, which shifts the A-Register left one, with ARØ←ASHI=ALU15. The B-Register is passed through the ALU and shifted left in the shifter (TBS1=Ø, TBSØ=1) with ALX14=ALU14 and LSI=AR15. The T-Bus is stored back into the B-Register.

b) If R1: AS1=Ø, ASØ=1, which shifts the A-Register right one, with AR15←B-Register Ø. B is passed through the ALU then shifted right (TBSØ=Ø, TBS1=Ø), with ALX16=ARØ. The T-Bus is stored back into the B-Register.

CRS with L1:



CRS with R1:



LGS ($\overline{\text{LGSOP}}$) 32-bit logical shift. Requirements:

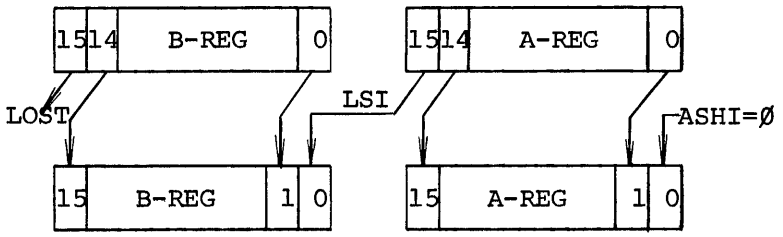
LGS	PASS	B	B	L1 OR R1
OP	ALU	S-BUS	STORE	SPEC

Operation is similar to CRS, ARS above.

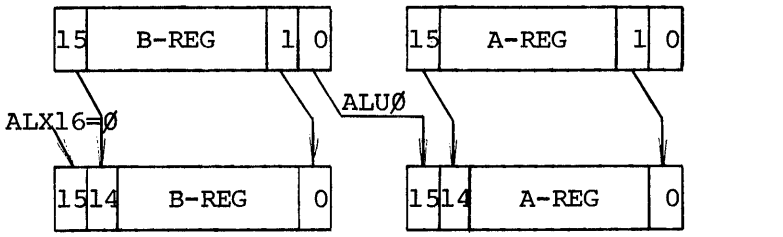
a) L1: As B is shifted in the shifter, ALX14=ALU14, LSI=AR15. ASHI shifts Ø into the A-Register.

b) R1: As B is shifted in the shifter, ALX16=Ø, A-Register 15←B-Register Ø.

LGS with L1:



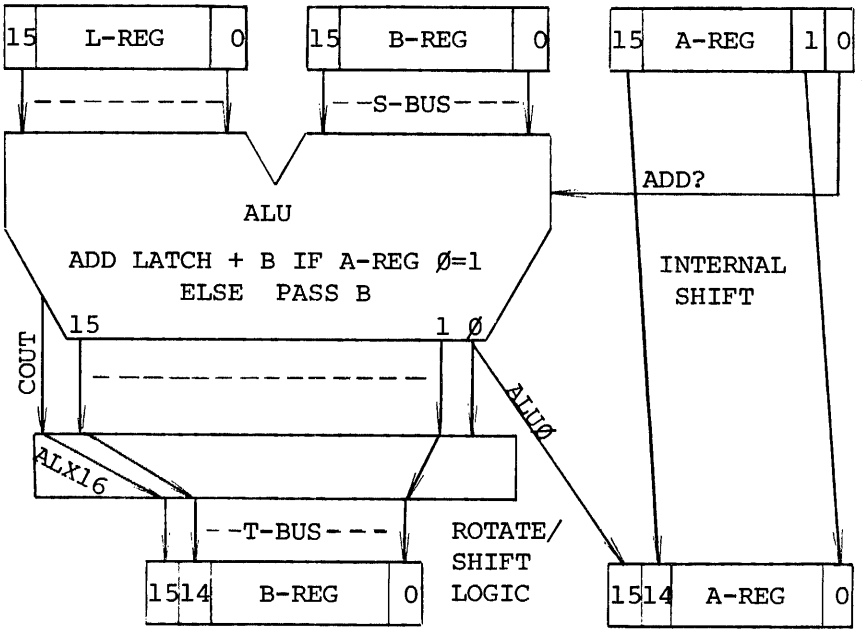
LGS with R1:



MPY (MPYOP) Multiply step. Normally used in a repeat loop as part of a multiply algorithm. Requirements for proper operation:

MPY	ADD	B	B	R1
OP	ALU	S-BUS	STORE	SPEC

The A-Register is shifted right internally, AR15←ALUØ. The B-Register is gated onto the S-Bus. The ALU adds the S-Bus to the L-Register if A-Register bit Ø is a 1, and passes the S-Bus if it is Ø. The output of the ALU is shifted right one, with ALX16=COUT (carry out of ALU). This is stored back into the B-Register via the T-Bus.

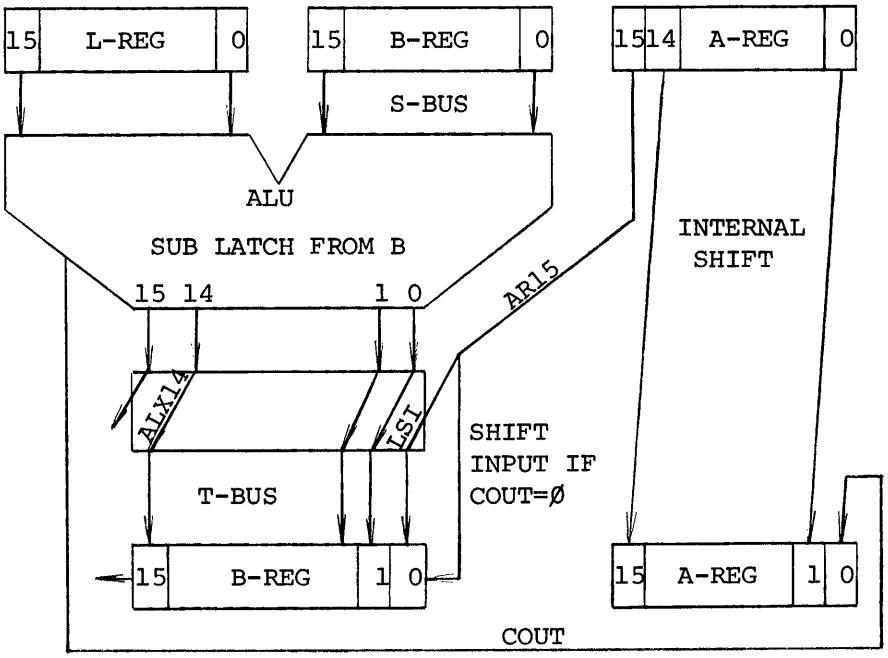


Sixteen repeats will perform the function B+A·L and leave the result in the B and A Registers, least significant bit in A-Register Ø and most significant bit in B-Register 15.

DIV (DIVOP) Divide step. Normally used in a repeat loop as part of a divide algorithm. Requirements for proper operation:

DIV	SUB	B	B	L1
OP	ALU	S-BUS	STORE	SPEC

The A-Register is shifted left one internally with ASHI=COUT (carry from ALU). The ALU subtracts the L-Register from the B-Register. The result is shifted left one in the shifter, with ALX14=ALU14 and LSI=AR15. If COUT=1 (no borrow), then this result is stored into the B-Register via the T-Bus. If COUT=0 (borrow), then the B-Register is shifted left internally (subtraction is effectively not performed), with B-Register bit Ø←AR15. Sixteen repetitions of this instruction will perform (B,A)÷L=quotient in B, remainder in A assuming (B,A) represents a positive number.



INTERNAL SHIFT IF COUT=Ø

LWF ($\overline{\text{LWFOP}}$) Link with flag. If L1 or R1 is specified in the Special Field, the FLAG FF is linked with the ALU to form a 17-bit rotate through the R/S logic. For L1, LSI=FLAGFF [F69] and FLAGFF←ALU15 [C66]. For R1, ALX16=FLAG FF [E69] and FLAGFF←ALUØ [C66].

WRTE ($\overline{\text{WRTEOP}}$) If memory is busy when $\overline{\text{WRTEOP}}$ occurs ($\overline{\text{REFRESH}}$ or $\overline{\text{MSRDY}}$ or $\overline{\text{DMALO}}$ low) the CPU will freeze [G32] until memory is free. Then the WRITE FF [E63] is set at the end of next P5. The WRITE FF is reset at the following P2. Initiates a write cycle in memory. The T-Register should be loaded in the same instruction in which WRTE is specified, as DCPC could destroy the T-Register contents if it was loaded any earlier.

ASG ($\overline{\text{ASGOP}}$) Used during ASG instruction routines. Sets, clears, complements the EXTEND FF according to the combination of IR6, IR7. Also clears the L-Register during P5 so the Overflow/Extend logic will operate correctly during increments.

READ ($\overline{\text{READOP}}$) If memory is busy when $\overline{\text{READOP}}$ occurs ($\overline{\text{REFRESH}}$ or $\overline{\text{MSRDY}}$ or $\overline{\text{DMALO}}$ low) the CPU will freeze until memory is free. Then the READ FF will be set at the end of the next P5. It will be reset at the following P2. This initiates a read from memory. The M-Register must be loaded prior to or during the instruction with READ in the OP Field. Data must be removed from the T-Register exactly two instruction cycles after READ, or else DCPC could destroy the contents of the T-Register, and memory disables T after that time.

ENV	($\overline{\text{ENVOP}}$)	Enables the Overflow logic for the current ALU operation. OVER FF is set at the end of P5 if the L-Register and S-Bus have the same sign bit (bit 15) and ALU15 is different. Caution is advised so that the L-Register sign bit is set properly.
ENVE	($\overline{\text{ENVEOP}}$)	Enables both the Overflow (see above) and Extend logic for the current ALU operation. The EXTEND FF is set at the end of P5 if COUT=1 (carry from the ALU).
JSB	($\overline{\text{JSBOP}}$)	Specify jump or subroutine jump to new location in microcode. See sections on word types 3, 4. The AND of $\overline{\text{JSBOP}}$ and $\overline{\text{JMPOP}}$ ($=\overline{\text{JORJ}}$) prevents storing into the scratch pads [D34], disables the Store Field decoder [C33], disables clocking of the Status Flag Register [E72], and enables the RAR loading logic [G17].
JMP	($\overline{\text{JMPOP}}$)	
IMM	($\overline{\text{IMMOP}}$)	Specifies word type 2. See Section 5.4. $\overline{\text{IMMOP}}$ enables the RIR onto the S-Bus [D41], specifies PASS through the ALU [G69] or CMPS if RIR19=1, and prevents the Holding Register from driving the S-Bus [E25].

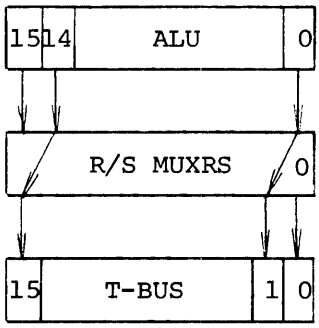
5.6.2 SPECIAL FIELD

The Special Field is used to control special computer options like memory protect and memory management, to control interrupt recognition, to perform special arithmetic/logical operations, to initiate special CPU control features, and to specify jump schemes (Section 5.2.4). The Special Field decoders are at [B62, D62].

IOFF ($\overline{\text{IOFFSP}}$) Clear the INTEN FF at the end of current instruction. This prevents recognizing of interrupts from devices with select codes greater than 5 (i.e., only memory protect and power fail or HALT mode may force RAR to 4 when a jump to Ø is attempted). Used during the JMP,I and JSB,I machine instruction routines to hold off interrupts until after one more instruction is executed.

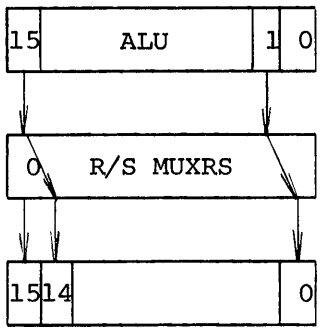
SRG2 ($\overline{\text{SRG2SP}}$) Enables IR bits 0, 1, 2, 4 to the SRG shift/rotate decoder (see section III.115). This sets up a shift or rotate of the ALU onto the T-Bus as required by the SRG instruction.

L1 ($\overline{\text{L1SP}}$) Sets up a left shift (TBSØ=1, TBS1=Ø) in the shift multiplexors [F71-77]. Without a qualifying OP Field command (LWF, ARS, CRS, LGS, MPY, DIV), this command shifts as shown below.



L4 ($\overline{\text{L4SP}}$) Sets up a circular left shift of 4 bit positions (TBSØ=Ø, TBS1=1) in the shift multiplexors. Rotates the ALU four positions before sending it onto the T-Bus.

R1 ($\overline{\text{R1SP}}$) Sets up a right shift (TBSØ=Ø, TBS1=Ø) in the shift multiplexors. Without a qualifying OP Field command (LWF, ARS, CRS, LGS, MPY, DIV), this command shifts as shown below.



ION ($\overline{\text{IONSP}}$) Turns on the INTEN FF [B63] at the end of current microinstruction. Allows normal interrupts to force the RAR to 4_8 when a jump or RTN to address $Ø_8$ is attempted [F12].

SRG1 ($\overline{\text{SRG1SP}}$) Enables IR bits 9,8,7,6 to the shift/rotate decoder (see section III.115). This sets up a shift or rotate of the ALU onto the T-Bus as required by SRG machine instructions.

RES2	($\overline{\text{XCHSP}}$)	Not offered to the microprogrammer. Exchange Flag FF with Extend FF. [D66].
STFL	($\overline{\text{STFLSP}}$)	Set the CPU Flag FF [C66].
CLFL	($\overline{\text{CLFLSP}}$)	Clear the CPU Flag FF [C66].
FTCH	($\overline{\text{FTCH}}$)	Interface signal to Memory Protect. Results: latch the Violation Register from the M-Bus during P5; clear MPV at the end of P5; reset the indirect counter. To be used while the address of the current machine instruction is on the M-Bus prior to its execution. Used to initialize the memory protect error detection logic.
SOV	($\overline{\text{SOVSP}}$)	Set the Overflow FF [A66].
COV	($\overline{\text{COVSP}}$)	Clear the Overflow FF [B67].
RPT	($\overline{\text{RPTSP}}$)	Set the Repeat FF [G14]. The RIR clock is disabled after P5 and the RAR increment is disabled after the following P2. The next microinstruction is repeated and the counter is incremented each succeeding P5 until the low 4 bits of the counter are 1111_2. Then the Repeat FF is cleared at P4 and normal control is restored. The microinstruction after RPT will be repeated the two's complement of the value in the low 4 bits of the counter (with $\emptyset\emptyset\emptyset\emptyset_2$ = 16 times).
SRGE	($\overline{\text{SRGESP}}$)	Clear the Extend FF [G66] if IR5=1.
NOP		No special signals generated.
MESP	($\overline{\text{MESP}}$)	Special purpose signal for use by memory management.

MPCK ($\overline{\text{MPCK}}$) Memory Protect Check. Interface signal to Memory Protect. Must be specified while the address of an impending memory reference is on the S-Bus. This value is compared against the Fence Register to determine if a violation occurred. If the S-Bus < Fence Register then $\overline{\text{MPV}}$ will go low if memory protect is enabled. If $\overline{\text{MPV}}$ is low, then memory references will not affect memory and memory data will appear as Ø, no I/O signals may be generated (IO Group Enable FF [C75] is disabled), and the P and S Registers may not be altered [D35, B22]. FTCH or IAK [H52] will clear the MPV condition.

IOG ($\overline{\text{IOGSP}}$) The processor will freeze until T2. At the end of T2, the IO Group Enable FF [C75] will be set, enabling I/O signal generation (see Section III.303) for one I/O cycle.

ICNT ($\overline{\text{ICNTSP}}$) Increment the Counter [E4].

SHLT ($\overline{\text{SHLTSP}}$) The Run FF will be cleared at the end of the next microinstruction.

INCI ($\overline{\text{INCISP}}$) The indirect counter is incremented on the Memory Protect board. Used after an indirect address level is detected to keep infinite loops from preventing detection of interrupt requests. The INTEN FF is set after 3 increments.

RES1 No signal.

SRUN ($\overline{\text{SRUNSP}}$) Set the Run FF at the end of the current microinstruction.

JTAB ($\overline{\text{JTABSP}}$) Load the RAR at P2 from the JTAB MAP (see Section 5.2.4).

RTN ($\overline{\text{RTNSP}}$) Load the RAR at P2 from the Save Register and clear the Save Register at P5.

5.6.3 STORE AND S-BUS FIELDS

Below is a description fo the mnemonics for the S-Bus and Store fields. The scratch registers are S1-S12, X, Y, P, S. Some of the other mnemonics have special meanings and applications, and are described below. The Store field decoder is shown at [B34] and the S-Bus decoder is shown at [B42]. The scratch pads are addressed directly by the RIR [A21]. The signal name out of the field decoders are indicated in parentheses.

The Store field decoder is disabled by JMP, JSB, or RIR19=1. The S-Bus field decoder is disabled by JMP, JSB, IMM, or $\overline{\text{DMAFRZ}}$ (T3 of a DCPC cycle) [B41].

TAB Store ($\overline{\text{TABST}}$) and S-Bus ($\overline{\text{TABEN}}$). Selects the T, A, or B Register, depending on the setting of $\overline{\text{AAFF}}$ and $\overline{\text{BAFF}}$ [B37] which are set according to the value of the T-Bus whenever the M-Register is changed. Allows A, B to be accessed instead of locations 0, 1 of main memory. See description of T below for timing associated with T.

T-BUS AT STORE INTO M	RESULTANT VALUES $\overline{\text{AAFF}}$	$\overline{\text{BAFF}}$	REGISTER SELECTED BY TAB
Ø	1	1	T
1	0	1	A
2	1	0	B
>2	1	1	T

CAB Store ($\overline{\text{CABST}}$) and S-Bus ($\overline{\text{CABEN}}$). Selects the A-Register if IR-11 is Ø, or the B-Register if IR-11 is 1. [D39, C36].

T — Store ($\overline{TST}$) and S-Bus ($\overline{TREN}$). If from Store field, $\overline{TST}$ is sent to memory if the processor is not frozen [B40]. Used to clock the S-Bus into the T-Register. If from S-Bus field, it freezes the processor until memory is ready [C40].

CIR ($\overline{CIREN}$) — Freeze the processor until T6, then during P3-P5, load the CIR from the Interrupt Address Bus [B54], issue IAK [H52] and gate the CIR onto the S-Bus. [D55], high order 1Ø bits = Ø.

L ($\overline{LST}$) — Load the L-Register from the S-Bus.

IOI ($\overline{IOIEN}$) — Drive the S-Bus from the I/O Bus if the Select Code Bus = Ø or $\geq 10_8$. $\overline{IOIEN}$ is independent of the IOI I/O signal, but is to be used during T5 when the IOI signal is present. When so used, the S-Bus is driven by the source determined by the Select Code Bus as shown below. The I/O Bus will be selected only if $\overline{IOIEN}$ is used. All other sources are gated onto the S-Bus directly during the IOI signal.

SELECT CODE BUS	SOURCE SELECTED BY IOI
00	I/O Bus (=Ø)
01	Front Panel Display
02	DCPC Channel 1 Word Count Register
03	DCPC Channel 2 Word Count Register
04	Central Interrupt Register
05	Memory Protect Violation Register
06	Nothing = 177777_8
07	Nothing = 177777_8
10_8-77_8	I/O Bus (loaded from output buffer of I/O device)

IOO ($\overline{\text{IOOST}}$) This is independent of the IOO signal generated by the I/O signal generators. Gates the S-Bus onto the I/O Bus if the IO Group Enable FF is set.

CNTR S-Bus ($\overline{\text{CNTREN}}$) and Store ($\overline{\text{CNTRST}}$). $\overline{\text{CNTRST}}$ stores the low order 8 bits of the S-Bus into the counter. $\overline{\text{CNTREN}}$ enables the 8-bit counter onto the low 8 bits of the S-Bus. The high 8 bits of the S-Bus are all ones.

DSPL S-Bus ($\overline{\text{DSPLEN}}$), Store ($\overline{\text{DSPLST}}$). Selects the front panel Display Register [B79-80].

DSPI S-Bus ($\overline{\text{DIEN}}$): Selects the Display Indicator Register of the front panel onto bits 5-Ø of the S-Bus. Higher order bits are ones. Store ($\overline{\text{DIST}}$) : Stores the low 6 bits of the S-Bus into the Display Indicator Register on the front panel. High order bits ignored. Note: Bits which are low correspond to LED indicators which are lit on the Display Indicator, as shown below:

S-BUS BITS LOW WITH DSPI GIVEN	5	4	3	2	1	Ø
REGISTER INDICATOR LIT	S	P	T	M	B	A

ADR ($\overline{\text{ADREN}}$) [A-D45] enable bits 9-Ø of the M-Register onto bits 9-Ø of the S-Bus. If IR10=Ø, gate Ø's onto bits 15-1Ø of the S-Bus. If IR10=1, gate M-Register bits 15-1Ø onto the S-Bus. Performs zero/current page addressing for MRG-type machine instructions.

IR ($\overline{\text{IRST}}$) Load IR from the S-Bus [B1].

M		The M-Register is only 15 bits. When enabled onto the S-Bus, bit 15 is low.
LDR	($\overline{\text{LDREN}}$)	Enable the complement of the contents of the loader ROM selected by IR15, IR14, and addressed by counter bits 7-Ø onto the S-Bus bits 3-Ø [C3-4]. Bits 15-4 will be pulled high.
MEU	($\overline{\text{MEST}}$)	Used by memory management.
RES2		No register drives S-Bus. S-Bus is 177777_8.
CM	($\overline{\text{CMST}}$)	Store S-Bus into M-Register if and only if the IR contains an MRG-type instruction [E37] but not jump direct. In hardware, M is loaded if $\overline{\text{CMST}}$ is low and [IR12+IR14+IR13(IR15+$\overline{\text{IR11}}$)]=1.
MEU	($\overline{\text{MEEN}}$)	Used by memory management.
PNM	($\overline{\text{PNMST}}$)	[D34]. Load the S-Bus into M-Register and the T-Bus into the P-Register (part of the scratch pads).
S1-S12 X, Y, P, S		These registers are all in the four 16X4 bit scratch pad registers and are addressed by the RIR through a multiplexer [A21].

5.6.4 ALU FIELD

The ALU field feeds directly into the ALU [A-H28] without decoding. The ALU continuously performs the operation specified by RIR19-15, except that this may be overridden by an IMM or MPY microinstruction to do a PASS or CMPS instead. The operations possible are shown below (S = S-Bus, L = L-Register). Arithmetic is 2's - complement, "+" ≡ logical OR.

INC	S PLUS 1
OP1	(S+L) PLUS 1
OP2	$(S+\overline{L})$ PLUS 1
ZERO	ALL ZERO OUTPUT
OP3	S PLUS $(S \cdot \overline{L})$ PLUS 1
OP4	$(S+\overline{L})$ PLUS $(S \cdot \overline{L})$ PLUS 1
SUB	S MINUS L
OP5	$S \cdot \overline{L}$
OP6	S PLUS (S·L)
ADD	S PLUS L
OP7	$(S+\overline{L})$ PLUS (S·L)
OP8	S·L MINUS 1
OP9	S PLUS S (LOGICAL LEFT SHIFT)
OP10	(S+L) PLUS S
OP11	$(S+\overline{L})$ PLUS S
DEC	S MINUS 1

Note: The following are logical operations.

CMPS	NOT S
NOR	NOT (S OR L)
NSAL	(NOT S) AND L
OP13	ALL ZERO
NAND	NOT (S AND L)
CMPL	NOT L
XOR	S (EXCLUSIVE - OR) L
SANL	S AND (NOT L)
NSOL	(NOT S) OR L
XNOR	NOT (S (EXCLUSIVE - OR) L)
PASL	L
AND	S AND L
ONE	ALL ONES
SONL	S OR (NOT L)
IOR	S OR L
PASS	S

6. MEMORY REFERENCES

6.1 M-Register Operation

6.1.1 Using M

The M-Register must be loaded with the address of main memory to be read before the READ micro-order occurs, or concurrent with it. The M-Register will be loaded at the start of P4 [C43]. If the CM micro-order is used, M will be loaded only if an MRG-type instruction is in the IR [E37]. M may be altered after a reference is initiated, as the M-Bus is clocked into a holding register in memory after 200 nsec into the memory cycle. The M-Register must be loaded prior to using a WRTE command, because write data is being loaded into T when WRTE is given.

6.1.2 TAB Logic

Whenever M is loaded, the A-Addressable and B-Addressable flip-flops (AAFF, BAFF) are set according to the value of the T-Bus [B37] as shown below.

T-BUS	AAFF	BAFF
1	1	0
2	0	1
OTHER	0	0

These flip-flops determine whether the A, B, or T-register will be used when the TAB micro-order is specified (because the A, B registers are addressed as locations Ø, 1 of memory although they are actually hardware registers). If M could be receiving Ø or 1, the INC micro-order should be used in the ALU field. This T-Bus scheme with INC is used to simplify operand fetches during instruction execution. The standard sequence is shown below:

```
READ INC  PNM  P      M←P, P←P+1, set TAB logic, initiate READ
    (wait)
     PASS S1   TAB    Get data from T, A, or B-Register,
                      according to TAB logic.
```

6.2 READ Operations

6.2.1 Freezes

$\overline{\text{MSRDY}}$ ($\overline{\text{Memory Soon Ready}}$) [G31] is high by P2 if memory is ready for another reference by the end of the next P5. If it is not high by P2, or $\overline{\text{DMALO}}$ is low, or $\overline{\text{REFRESH}}$ is low (memory refreshing), then a freeze will occur if READ is specified, until it is safe to proceed.

6.2.2 Initiation

A memory read cycle is initiated by the READ FF [F63] at the end of P5 when the READ micro-order is specified. $\overline{\text{READ}}$ is low from P5 through P1, but the falling edge initiates the cycle.

6.2.3 Data Retrieval

Data must be retrieved from memory exactly two microinstructions past the READ. After this time, memory disables the T-Register. If the T or TAB (if AAFF=BAFF=0) micro-orders are in the S-Bus field, and $\overline{\text{MSRDY}}$ is low, a freeze will occur until $\overline{\text{MSRDY}}$ is high. If the TAB logic specifies A or B, then the T-Register is not referenced at all, and no freeze will occur. Location 0, 1 of main memory may be referenced if the TAB logic is not used as prescribed.

6.3 WRITE Operations

6.3.1 Freezes

The WRTE micro-order requires the same freeze operation as READ in Section 6.2.1.

6.3.2 Initiation

A write cycle is initiated by the WRITE FF [E63] at the end of P5 when the WRTE micro-order is given. $\overline{\text{WRITE}}$ is low from P5 through P1, but the falling edge of $\overline{\text{P5}}$ initiates the cycle.

6.3.3 Data must be loaded into the T-Register in the same micro-instruction as WRTE occurs, in order to insure that the DCPC does not destroy the T-Register before WRTE can be executed. If the TAB logic specifies A or B, then T is not altered, although a write is performed to the address in M. Locations Ø and 1 of main memory are accessible if the TAB logic is not used as prescribed.

7. FETCH ROUTINE - DETAILED DESCRIPTION

Refer to the listing for the Base-Set microcode.

LOCATION	MICROINSTRUCTION	
Ø	READ FTCH INC PNM P	Load M with program counter. Increment P through ALU and store back into P. Set TAB logic from the incremented value of P. FTCH only goes to memory protect to clear MPV and initialize logic on the board for the current instruction
1	ION	Enable recognition of I/O interrupts by the processor.
2	CLFL PASS IR TAB	Get instruction from T, A, or B-Register. Put into IR. Clear CPU Flag.
3	READ JTAB INC CM ADR	Form zero or current page address on S-Bus. Store into M if MRG-type instruction. Initiate read cycle. Load RAR from JTAB map, clear Save Register. Set TAB logic from T-Bus.

8. I/O INSTRUCTION EXECUTION

8.1 Fetch Sequence and Initiation

Refer to Section II.4 for general instruction accessing scheme. When an I/O instruction is in the IR, JTAB causes the RAR to be loaded with 0101_8. At this address, there is a JMP IOG ____ ____ IOCNTRL. The IOG micro-order is used by memory protect to check for I/O violations. If a violation occurs ($\overline{\text{MPV}}$ low), the I/O Group Enable FF (IOGEN FF)[C76] is direct-cleared, preventing I/O signal generation. IOG causes a freeze until time T2,* then the IOGEN FF is set, enabling I/O signals, and the jump is done using the JIO Map [C9]. There are four different types of I/O routines required, which are the destinations of the jump at 101_8:

1) IOCNTRL — Non-data I/O routine. For issuing of control signals and testing the skip flag.
2) IO.OT* — For effecting output transfers from A/B to an I/O interface.
3) IO.LI* — For effecting data input transfer from I/O interface to A or B.
4) IO.MI* — For effecting a merge (IOR) of data from I/O interface and A or B.

8.2 Signal Generation

Once IOGEN FF is set, I/O system signals are generated at the proper T periods without processor control (except for IOI; which must be done by the processor) according to bits 9-6,11 of the IR. The Select Code Bus is driven by IR5-0, except during DCPC cycles [F52], selecting which device should respond to the signals. Refer to Section III.303 for detailed I/O system operation and timing. At the end of T5, IOGEN FF is cleared, disabling I/O signals again. ENF (T2) and SIR (T5) and T3 are generated unconditionally once every 1.62 μsec.

*Or if DCPC is going to take a cycle, the CPU will freeze until DCPC is done.

8.3 I/O Routines

The four I/O routines must synchronize with I/O signal generation. The first microinstruction in each routine is performed at T3. NOPs are done until the proper T period for CPU-I/O system interaction. For the IO.OT* routine, CAB must be passed onto the S-Bus before IOO occurs. The S-Bus must be set up with the data to be sent onto the I/O Bus one instruction before IOO, because otherwise transition glitches on S can be transmitted to the I/O Bus. Some I/O interfaces have "ones catcher" input registers, and may get set by these spikes unless the S-Bus is settled down before IOO.

9. MACHINE INSTRUCTION EXECUTION

All types of machine language instructions resident in main memory as programs are executed by microcode routines, and none are executed with separate hardware. That is, there is no special hardware unit required to execute index register, floating point, extended arithmetic, or byte or bit instructions, etc. All are executed under full control of the processor microinstruction set. Some special processor functions can be enabled directly by the processor to aid in execution, but such hardware assistance is limited to the microinstruction cycle in which it is enabled. Only memory references and I/O instruction cycles are executed with the assistance of semi-independent hardware. Memory cycles are merely initiated by the processor and data retrieved, while a memory controller generates the cycle control signals; I/O signals are initiated by the processor, then are generated semi-independent of processor control. Refer to Sections II.5, III.113, III.115 for descriptions of special hardware enabled by microcode.

10. INDIRECT ROUTINE

This section describes the operation of the indirect addressing routine in the module Ø microcode (location 13_8). This routine is used to perform indirect addressing for MRG instructions, and to check for and perform indirect addressing, if needed, for extended arithmetic and floating point instructions. The indirect code is a subroutine, and is terminated by a RTN micro-order.

10.1 Entry Point

The routine entry point is labelled "INDIRECT". The routine expects a memory read and JSB to INDIRECT to have just been executed. An IOFF micro-order may have been executed.

10.2 Execution

The logical operation of the indirect routine is shown in Figure II.10 as a flowchart, with each box corresponding to a microinstruction, whose octal location in the control store is indicated in parenthesis beside

the block. INCI is a special field micro-order which is used to increment an indirect level counter on the memory protect board each time a level of indirect addressing is detected. After it reaches 3, it no longer increments, and pulls $\overline{\text{MPINTON}}$ low, holding the INTEN FF high [A63], re-enabling recognition of I/O interrupts if disabled by an IOFF micro-order. When an interrupt or halt condition is detected, the program address register is decremented and the routine jumps to the interrupt handler at location 4. If the "INSTR STEP" button on the front panel is pressed, the routine will not be interrupted until after the button is released.

11. FRONT PANEL ROUTINES

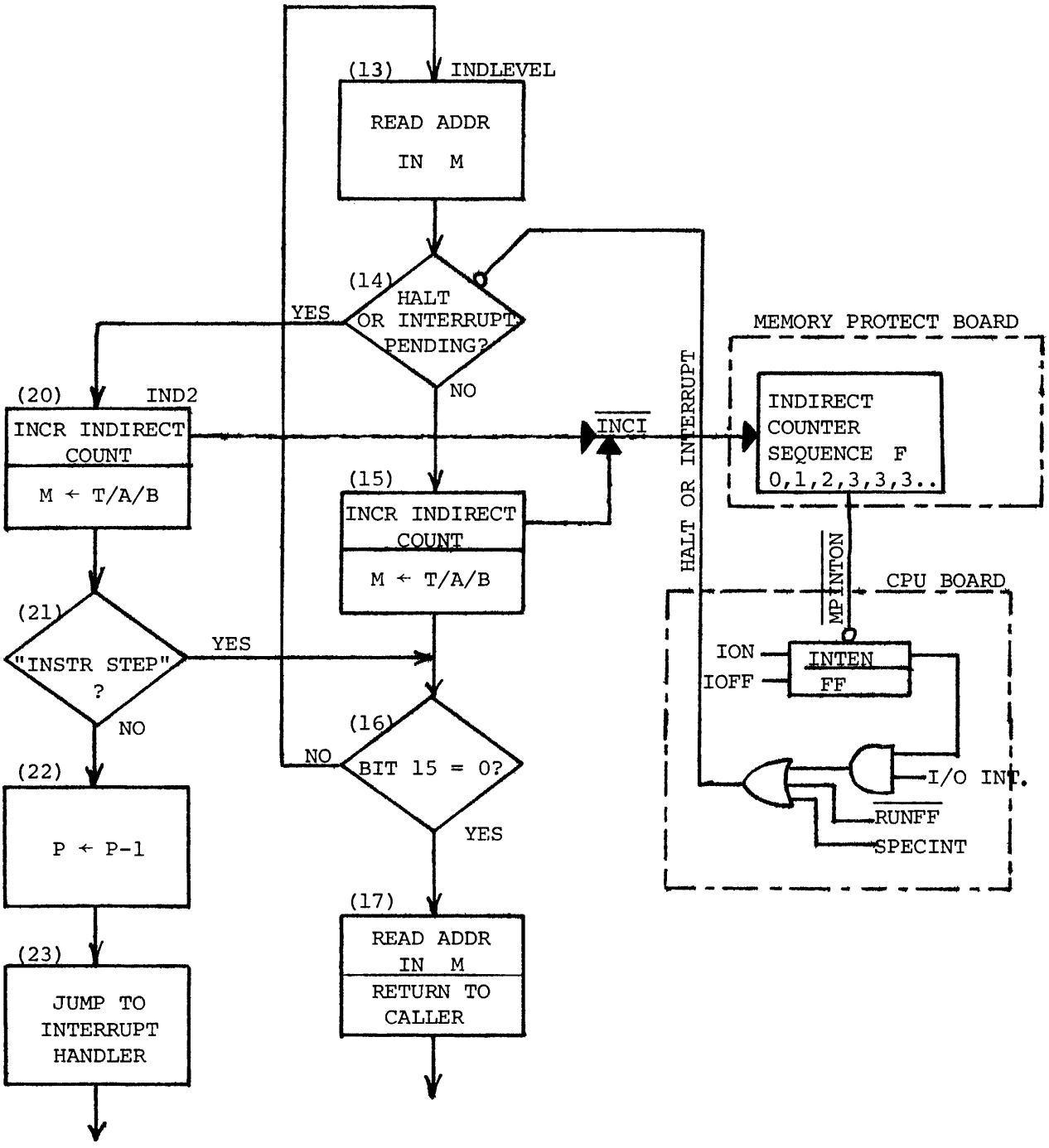
Most of C.S. module 1 is comprised of routines for controlling the front panel operations using the control buttons. Figure II.11 is a general functional flowchart of the organization and operation of front panel routines. Labels of routine locations appear at the upper right of blocks representing routines on the flow chart. These correspond to the labels given to the microprogramming manual listings of the microcode. Each block on the flowchart may represent a large amount of code.

12. LOADER INITIALIZATION ROUTINE

This routine is part of the front panel routines, as shown in Figure II.11. It is entered when the "IBL" button is pressed in the HALT mode and sensed by the SCAN routine. Its operation is described below.

12.1 Determine Memory Size

Find the high memory address ØXXXØØ_8 where the loader should start. This is done by loading P with Ø777ØØ, checking the ability to write and read to memory, then decremenenting P by Ø1ØØØØ_8 each time until valid memory is found. If none found, jump to WAIT routine in front panel routines.



OPERATIONAL FLOWCHART OF INDIRECT ADDRESS ROUTINE

FIGURE II.10

12.2 Select A Loader ROM

The contents of the Switch (S) Register are put into the IR. IR bits 15,14 control the enable inputs of the loader ROMs [C3] and enable the outputs of one of the 14 possible ROMs. Refer to Section III.111 for details of these ROMs.

12.3 Check Select Code

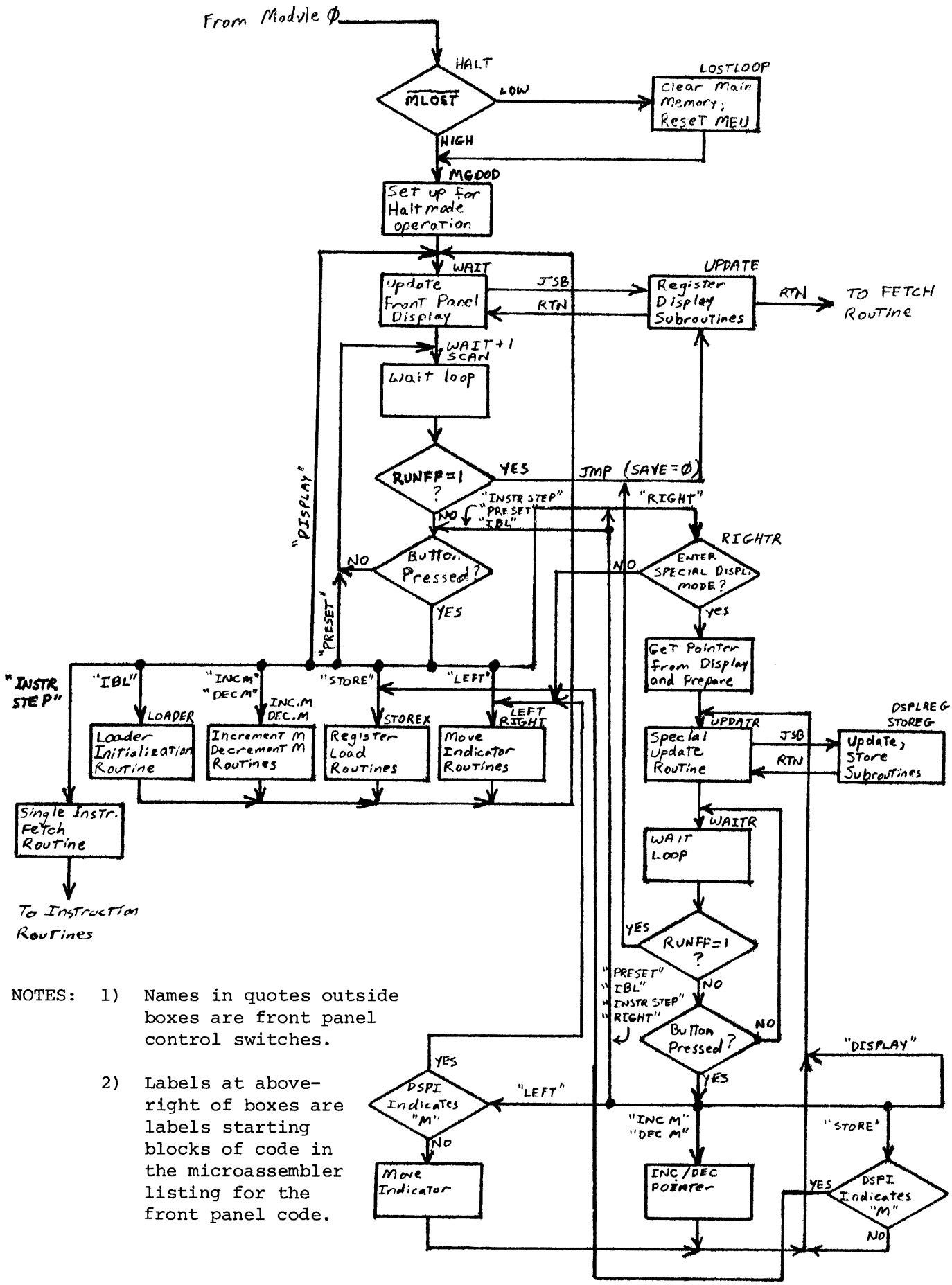
Bits 9-6 of S are masked to determine the select code of the device which I/O instructions in the machine language loader program are to reference. If select code is $< 10_8$, set Overflow and jump to WAIT routine (select code must be $> 10_8$).

12.4 Transfer Loader ROM Contents to Memory

The Counter Register is cleared to address location Ø of the enabled loader ROM (the counter addresses the loader ROMs). Then the contents of four consecutive locations in the ROM are merged into a scratch pad register to form a 16-bit machine instruction, which is stored into the memory. The M-Register is incremented and the process is repeated 64 times, storing the 64-word program in the 256 x 4 bit ROM into the high 64 words of memory.

12.5 Update Select Code in I/O Instructions

Each instruction in the loaded program in memory is checked for I/O type. If it is an I/O group instruction and not an HLT, the select code is changed to that determined in Section 12.3 above. The last word in the program is assumed to be a DCPC control word. Its bits 5-0 are changed to the proper select code, too.



NOTES: 1) Names in quotes outside boxes are front panel control switches.

2) Labels at above-right of boxes are labels starting blocks of code in the microassembler listing for the front panel code.

FUNCTIONAL OPERATION OF FRONT PANEL MICROROUTINES

FIGURE II.11

13. RUN-HALT MODE TRANSITIONS

Since all control functions of the 21MX are microcoded, including front panel routines, transition from one mode of operation to the other is simply a change in microroutines, and not an enabling and disabling of hardware. This section describes how the computer senses the need to go between the READ and HALT routines, and what is done to effect the transition. The RUN FF [G44] notifies the processor of which mode it should enter. It may take a while for the processor to respond to changes in the RUN FF (when RUN FF is high, the RUN mode is requested), as response is micro-code-dependent. Refer also to Section III.118 for details of RUN logic.

13.1 RUN → HALT Mode Transition in Base Set Microcode

The RUN FF will be cleared (HALT mode requested) [G44] if:

a) The HALT button on the front panel is pressed and the key switch is not in the LOCK position (RUNEN = 1).

b) The Power-ON signal has not yet occurred (PONB low).

c) The Parity Response Switch, S1, is in the HALT PE position and a parity error is detected in memory [A57].

d) An I/O HALT instruction cycle is executed.

e) An SHLT micro-order is executed.

When RUN FF is low, then the next time the microcode attempts to load the RAR with Ø, it is forced to 4. At location 4 is a microinstruction which jumps, if the RUN FF is clear, to the front panel routines in module 1. This occurs at the end of machine instruction execution, when the next instruction would normally be fetched by the routine at address Ø. The front panel routines could also be entered as the destination of a JMP CNDX RUN or JMP CNDX NHOI microinstruction. Upon entry into the front panel (or "Halt") routines, the Display Indicator Register is loaded to light the "T" indicator on the panel, and memory protect is initialized with a FTCH micro-order to clear the $\overline{\text{MPV}}$ line, which could otherwise prevent altering some of the internal registers, memory referencing, and I/O instructions if a previous memory protect violation had occurred. FTCH is necessary to allow full control in the HALT mode if memory protect is installed.

13.2 HALT → RUN Transition

The RUN FF will be set [G44] whenever:

a) The power-fail/auto restart logic sets the Run Buffer FF.

b) An SRUN micro-order is executed.

c) The RUN button on the front panel is pressed and the key switch is not in the LOCK position (RUNEN = 1).

During execution of the front panel routines in the HALT mode, conditional JMP microinstructions are done to test the RUN FF. When it is detected as being on, the S-Register is put out to the Display Register, the Display Indicator is made to indicate S, and a jump is done to the fetch routine at location Ø. This completes the transition.

14. POWER-UP SEQUENCE

For details of timing of the PWU and PON signals, and the power fail logic, refer to Section III.400. This section describes the overall response of the computer to the initial power-up sequence. The CPU board can tell no difference between a power-up from STANDBY position of the key switch, or from a line power off condition. Refer to the power supply and memory controller theories of operation for detailed description of their power sequence responses.

14.1 STANDBY

In the STANDBY (key switch position) mode, all power is removed from the CPU board, DCPC board, M.P. board, I/O backplanes, control store, and front panel.

14.2 Voltages up, PON low

When the key switch goes to OPERATE, +5V and -2V are applied to the CPU board, front panel, and control store, and all I/O voltages are turned on. PON and PWU signals [A51] from the power supply will remain low for about 1/2 second after power is restored.

14.2.1 Front Panel Response

When +5 volts comes up, and R-C filter on the clear input to the Display Indicator Register keeps that input low long enough to turn on all the register indicators. These will stay on until turned off by the microcode, providing a visual indieation of proper initiation of processor operation.

14.2.2 Memory Section Response

As long as PON is low, memory is kept in a refresh-only mode of operation. For details, refer to the theory of operation for the appropriate memory controller.

14.2.3 CPU Board and Control Store Response

As long as PON is low, the following conditions are forced:

a) Power Fail Register [B59] is cleared: IRQ4FF=DIRFF=PWUFF= PF1FF=PF2FF=PF3FF=Ø.

b) The PRESET FF is set [F42] (see Section II.17 about PRESETs).

c) Run Buffer and RUN FFs are direct-cleared, forcing a request for the HALT mode of operation [H43].

d) POPIO and CRS I/O signals are sent up the I/O backplane every T5 [C-D59], to initialize all I/O and optional devices, including DCPC, Memory Protect, parity interrupt logic.

e) The Interrupt System Register [E57] is kept cleared, disabling the interrupt system: INTSENFF-INTOVRFF-NRMINTFF=$\overline{\text{CONT4FF}}$=Ø

f) $\overline{\text{RTN}}$ is forced low [E63], clearing the SAVE Register, and loading the RAR every P2.

g) $\overline{\text{PF1FF}}$ is high, forcing the outputs of the RAR mappers to zero [C20]. Since RUNFF is low, the RAR is loaded at P2 with the value $ØØØ4_8$ [F12] (refer to Section III.124). The contents of this address are loaded into the RIR at the end of every P5. That location contains a conditional jump to the halt routines if RUNFF=Ø, which is true.

About 1/2 second after power is applied to the CPU board, PWU will go high, followed by PON going high.

14.3 Voltages Up, PON High

14.3.1 Jump to HALT Routines, Initialize Memory.

When PON goes high, the Power-Fail/Auto Restart sequence begins (for details, see section III.400). But microcode execution does not detect its operation until much later. When PON goes high, the RAR has been addressing location 4. It is now released to operate normally. A jump is done to location 400_8 from location 4. 400_8 is the start of the HALT routines. The $\overline{\text{MLOST}}$ signal is tested [E71]. $\overline{\text{MLOST}}$ will be low if memory sustaining voltages were lost due to lack or failure of a sustaining battery for memory when the LINE Power Switch was turned OFF. As long as memory is supplied with sustaining voltages during power-offs, $\overline{\text{MLOST}}$ will be high when operating power is restored. So if $\overline{\text{MLOST}}$ is low when microcode gets to location 400_8, memory is initialized to all 0 by a M-routine. $\overline{\text{MLOST}}$ will go high six milliseconds after PWU goes up. So it could be low only immediately after PON goes high.

14.3.2 Power Fail Response

After μ-code execution has entered the HALT routines, the Power Fail logic will perform its functions, if any. If the ARS switch S2 is in the ARS position, and the PRESET button is not pressed, then the RUN ff, IRQ4ff, and DIR ff will be set and CONT4FF will go high within a few μ-instructions of entry into the HALT routines. If the PRESET button is pressed or S2 is in the ARS position, none of those actions will occur, and RUNFF will remain low. If memory was initialized to 0, that same μ-routine will do a SHLT in order to insure that the machine stays in the HALT mode whether or not the ARS option has previously set RUNFF. Then μ-code reaches a wait loop, where it waits for a Front Panel button to be pressed or RUNFF to be high. If the Power Fail logic had set RUNFF and IRQ4FF, then at this point the μ-code sets up the machine for RUN mode operation. It will attempt a jump to location 0 to do an instruction fetch. But

IRQ4FF will make $\overline{\text{SPECINT}}$ low to force the RAR to 4 instead of 0. Now at location 4, the RUN ff is set, so execution will continue to location 5, where interrupts are serviced (see section II.16), and from there, execution proceeds normally. The routine at location 5 will fetch and execute the machine instruction in memory location 4.

15. POWER DOWN SEQUENCE

When the Power Supply detects that it is necessary to shut off power due to lack of line power, or some other reasons (see Theory of Operation for the power supply), PWU will go low, followed by PON going low. 500μsec. later, before power is actually removed from the CPU board.

15.1 PWU Low, PON high

When PWU goes low, the Power-Fail logic will set IRQ4FF high and DIRFF low if S2 is in the ARS position (see section III.400 for detailed operation). In this case, the CPU will service the interrupt the next time the RAR is to be loaded with 0 or when the μ-code tests for interrupts. The routine in location 5 (see section III.16) will execute the machine instruction in memory location 4. If S2 is in the ARS position, the machine will proceed with normal operation until PON goes low.

15.2 PWU Low PON Low

When PON goes low, the same conditions are created as described in section II.14.2.2 and II.14.2.3, above.

16. INTERRUPT GENERATION, HANDLING

Generation and recognition of interrupt flags to the CPU is covered in sections III.305 and III.124. This section will describe the way the base set μ-code services interrupt flags from the I/O sections with particular attention paid to the μ-code interrupt servicing routine.

16.1 Getting to the Interrupt Servicing Routine

When the I/O section issues an I/O interrupt flag to the Control section ($\overline{\text{HOI}}$ low or INTFLG high), the select code (inverted) of the interrupt requesting device is present on the IA-BUs during every T5, T6.

The select code could change from one T5-T6 to another if a higher priority device requests interrupt service. A μ-code conditional jump testing INT, or NHOI with RJS, will be successful if an interrupt is pending. If not tested like this, RAR will be forced to 4 the next time a JMP or RTN is done to location 0. Location 4 contains a JMP to HALT if RUNFF if low. Otherwise, the interrupt servicing routine at 5 is done.

16.2 Operation of the Interrupt Servicing Routine

The interrupt servicing routine is shown below. A description of its execution follows.

ADDRESS	LABEL	OP	SPEC	ALU	STOR	SBUS
0005	INTERUPT	READ	CLFL	PASS	M	CIR
0006		JMP	CNDX	TBZ	RJS	INTOK
0007		READ		PASS	M	CIR
0010		JMP	CNDX	TBZ		FETCH
0011	INTOK		IOFF	PASS	IR	T
0012		READ	JTAB	INC	CM	ADR

16.2.1 Location 0005

Freeze the CPU if memory is busy ($\overline{\text{MSRDY}}$ low or $\overline{\text{REFRESH}}$ low) or if the time period is not T6 or if $\overline{\text{DMALO}}$ is low [H32]. If $\overline{\text{DMALO}}$ is low, a DCPC cycle is being initiated, which means that some device has set its FLAG FF to request DCPC service. That device could have its CONTROL set, and could have higher priority than another device requesting interrupt. If the CIR is loaded when $\overline{\text{DMALO}}$ is low, the select code of the device under control of DCPC could be on the IA-BUS. $\overline{\text{DMALO}}$ holds off completing this instruction until DCPC has serviced its device, so it can no longer interrupt. After the freeze, at T6, the CIR (6 bits) is loaded from the IA-BUS at the leading edge of P3, and the IAK I/O signal is issued during P3, P4, P5 of T6 to stop the requesting device from issuing further IRQ and FLAG signals. The CIR is inverted and passed onto the S-Bus with S-Bus bits 15-7 = 0. At the rising edge of P4, M is loaded from the S-Bus. The CPU FLAG FF is cleared at the end of P5. A Memory Read is initiated at the end of P5.

16.2.2 Location 0006.

A jump is performed to location 11_8 if the interrupt select code was non-zero. There is a possibility that a Memory Protect violation following a MPCK special could generate a Special Interrupt request and have location 5 executed before Memory Protect could generate an IRQ, resulting in a zero being read onto the S-Bus in section 16.2.1, above. If this occurs, (it is impossible, in order to maintain the protective features desired, to eliminate this possibility) a second READ of the CIR is attempted in location 0007.

16.2.3 Location 0007

Same operations as in section 16.2.1 above, except no CLFL is done.

16.2.4 Location 0010

Test the results of reading the CIR again. If still 0, no interrupt request occurred. In that case some error occured, and the interrupt service routine is terminated by going onto fetch the next machine instruction. If the special case mentioned in section 16.2.2 occurred, M will have been loaded with 5_8.

16.2.5 Location 11_8

The data from the memory read is in the T-register. This data, the contents of the main memory address equal to the select code recorded by loading the CIR, is passed onto the S-Bus and stored from there into the IR. IOFF disables recognition of subsequent I/O interrupts until after execution of the instruction in the IR and one more machine instruction, unless ION or 3 INCI m-orders occur before that. This allows the instruction in the interrupt Memory location to add one more instruction to be executed to allow a software interrupt routine to be started before the CPU can be interrupted again.

16.2.6 Location 12_8

This μ-instruction is identical to the last μ-instruction of the Fetch routine, section II.7.

17. PRESET

When the PRESET button is pressed on the Front Panel when RUNFF is low, PRSTFF goes high until the button is released [G43], and the following operations are performed:

a) The Parity ff [F44] is cleared, if set.
b) The A and B Addressable ff's (AAFF, BAFF) are cleared [C38]
c) The REPEAT ff is cleared [G13].
d) $\overline{\text{CONT4FF}}$ is set low (enable the Power-Fail logic) [G56].
e) INTSENFF is set low (disable interrupt flag generation for normal interrupt requests) [E56].
f) POPIO and CRS I/O signals are issued every T5, to reset the I/O system interface cards [C56].
g) The IR is cleared.
h) Power-Fail logic response is affected as described in section III.400.
i) μ-instruction execution proceeds normally in the HALT routines.

18. DCPC CYCLES

When the DCPC (slot 110 in Memory Backplane) steals an I/O cycle for a data transfer, several signals are generated by the DCPC board to prevent interference by the CPU and to help control the transfer.

Figure II.18 shows the timing of DCPC signals which affect the CPU during input and output transfers. The following paragraphs describe the operation of each DCPC control signal which affects the CPU. Each of those signal lines is pulled high on the CPU if DCPC is not installed. They all affect the CPU only when low.

18.1 $\overline{\text{DMACYC}}$ [H51]

a) Disable bus driver which gates IR5-0 onto SC-Bus[F52], and enable SC-Bus driver on DCPC to give DCPC control of SC-Bus.

b) Prevent Setting IOGENFF [C75]. This prevents initiating I/O instruction cycles in the CPU during DCPC cycles.

18.2 $\overline{\text{DMAEN}}$ [A50]

Disables the bus drivers which gate the M-Register onto the M-Bus [B-C96], enables DCPC M-registers onto the M-Bus.

18.3 $\overline{\text{DMAFRZ}}$ [H31]

a) Freeze the CPU during T3, unless a JMP or JSB is being performed in μ-code.

b) Disable the bus driver which gates the Holding Register onto the S-Bus [E25].

c) Disable the S-Bus Field Decoder [B41] so all outputs are high

d) Disable the IMM logic bus drivers [D41].

All the above operations are necessary to allow DCPC to use the S-Bus at T3 to transfer data between the I/O - Bus and Memory without interference from the CPU.

18.4 $\overline{\text{DMAIOI}}$ [A41]

Drive the S-Bus with data from the I/O - Bus [E49]

18.5 $\overline{\text{DMAIOO}}$ [E41]

Drive the I/O - Bus with data from the S-Bus [G47].

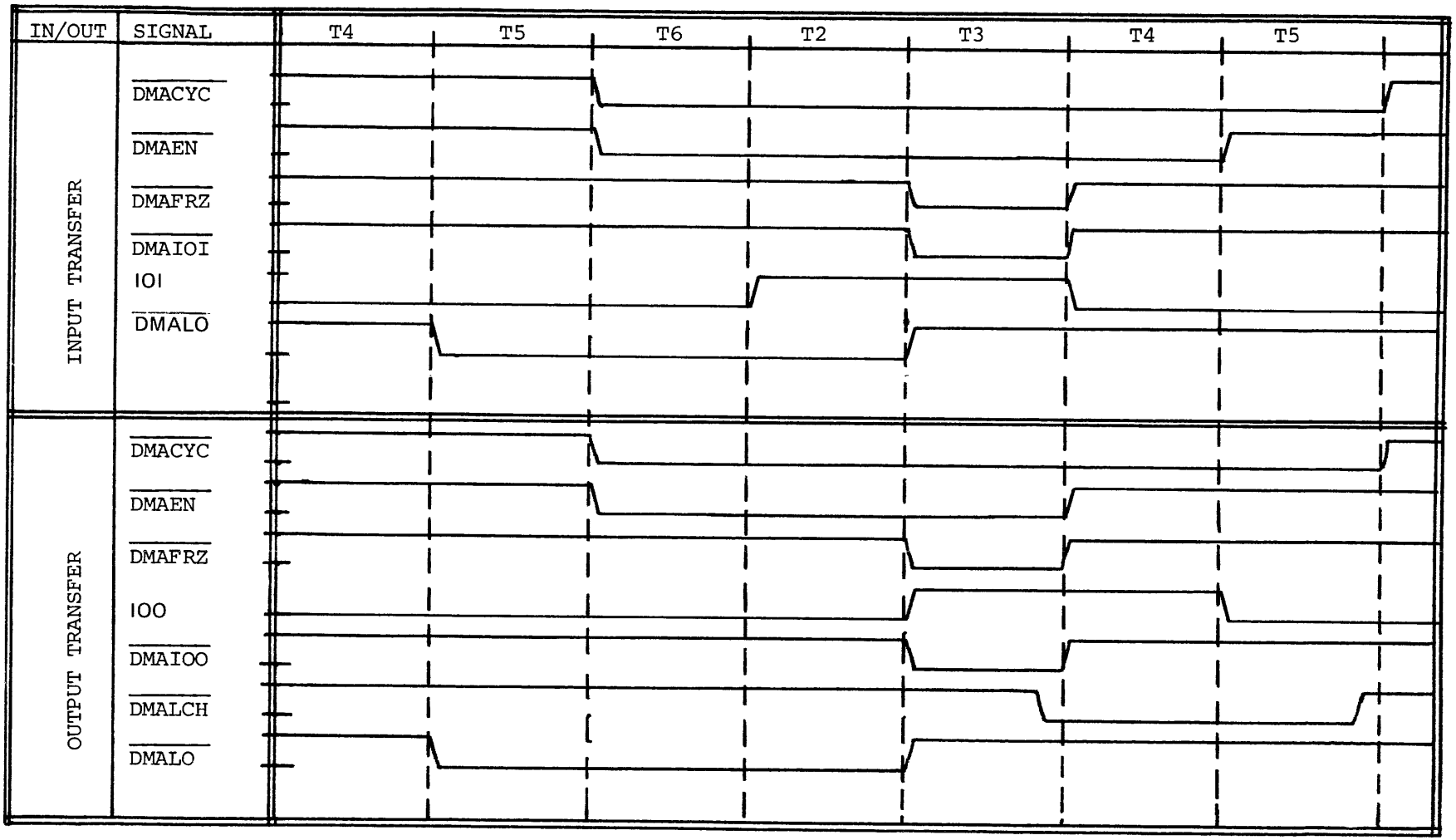
18.6 $\overline{\text{DMALCH}}$ [H51]

Latch the I/O - Bus on itself with the I/O - Bus driver gates [H47].

18.7 $\overline{\text{DMALO}}$ [H31]

Freeze the CPU if it attempts to interfere with DCPC cycle. Refer to section II.2 for details of Freeze operation.

Keep INTOVRFF low to override generation of NRMINTFF high[F56] (see section III.305 for details). This prevents interrupt flags during DCPC cycles (DCPC-requesting devices could issue IRQ at the start of the DCPC cycle).



DCPC CYCLE SIGNALS AFFECTING THE CPU

FIGURE II.18

19 MEMORY PROTECT VIOLATION

For details of Memory Protect option (slot 111 in Memory backplane), refer to the Theory of Operation for the 12892A Memory Protect. This section of the 21MX theory explains how Memory Protect violations affect the CPU board.

19.1 Generation of Memory Protect Violations

$\overline{\text{MPV}}$ [A51] is the Memory Protect Violation flag. It is high normally, but will go low if a violation occurs, and stay low until the next FTCH or CIR μ-order. A violation occurs if the Control for select code 5 has been set previously (STC Ø5 I/O instruction), and one of the following actions occur:

a) IOG μ-order is used, and either IR bits 5 through Ø $\neq$ $Ø1_8$ or a HALT instruction is in the IR.

b) MPCK μ-order is used, and the S-Bus value is less than the value in the Memory Protect Fence Register on the Memory Protect board (values of Ø and 1 on the S-Bus are legal if the IR does not contain a JMP machine instruction).

c) Read or write from a protected page.

d) Base page is mapped.

e) Alteration of memory expansion module (MEM) status register or contents of maps.

19.2 Effect of $\overline{\text{MPV}}$

When $\overline{\text{MPV}}$ goes low it performs the following functions on the CPU boards:

a) Request a Special Interrupt (see section III.305) if the Interrupt system is enabled [D57] ($\overline{\text{SPECINT}}$ low).

b) Direct Clear IOGENFF [C74] and prevent its being set. This disables all I/O system signals from the I/O signal generator logic.

c) Disable the $\overline{\text{RAMWEM}}$ signal [B22] if a store is attempted into the P or S Register in the scratch pads to prevent alteration of P or S.

$\overline{\text{MPV}}$ also affects the Memory Controller. For details, see the individual Theory of Operation.

SECTION III

DETAILED THEORY

100 CONTROL SECTION

The Control Section contains the logic necessary to execute microprograms and fetch and execute programs in main memory. For the purpose of this discussion all registers in the scratch pads are considered part of the Arithmetic/Logic Section, and the front panel is considered to be part of the Control Section. See Figure III.100 for a block diagram of the Control Section.

101 IR [A-C2]

The 16-bit Instruction Register usually holds the current machine instruction code. It is implemented with dual quad latches. During P5 when $\overline{\text{IRST}}$ is low the S-Bus is enabled into the IR. At the end of P5, it is latched. If the PRESET button is pressed, $\overline{\text{PRST}}$ will clear the IR to ø.

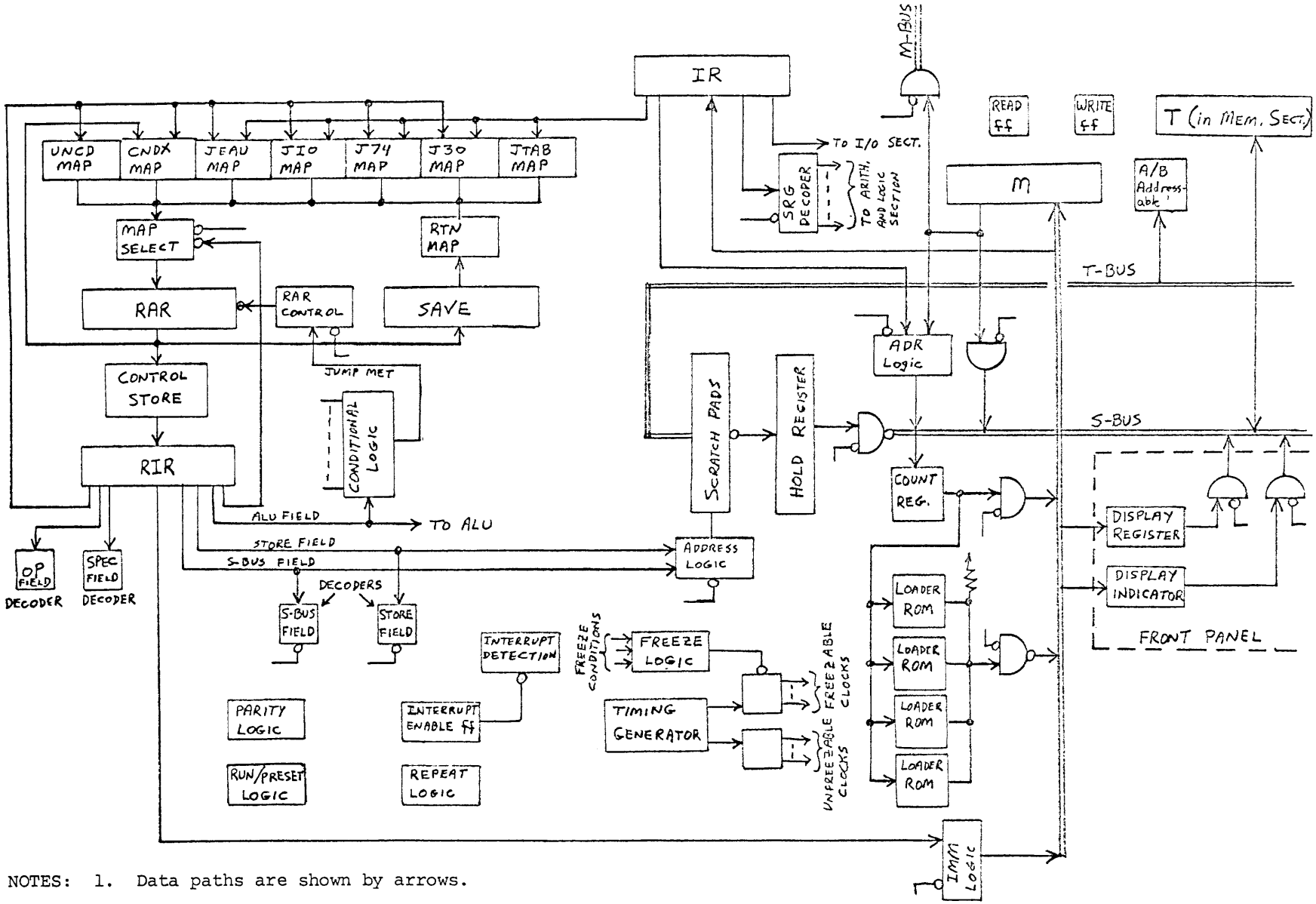
Uses of the IR: RAR map logic, Section 108; CM decoding, Section 105.1; loader ROM selection, Section 111; ASG/SRG decoding, Section 115; condition testing, Section 113; ADR logic, Section 123; I/O signal generation, Section 302; Select Code Bus, Section 306; CAB logic, Section II.5.6.3.

102 RAR [G15-18]

12-bit ROM Address Register, implemented with three 4-bit counters. Contains the address of the next Control Store instruction to be fetched.

102.1 It is incremented every unfrozen P5, unless RPTB FF is set, which inhibits the increment feature during repeat loops (Section 114).

102.2 If any type of branch is to be done, the outputs of the ROM Map Logic are loaded into the RAR during P2. There are three conditions which result in a load [F17].



NOTES:
1. Data paths are shown by arrows.
2. Line with bubble (—o) indicates enable/disable function.
3. Bubble on gate outputs indicate inversion of data.
4. Some parts of other computer sections are shown here for understanding interrelationships.

FIGURE III.100 BLOCK DIAGRAM OF CONTROL SECTION

a) $\overline{\text{RTN}}$ is low, which is an NOR of PONB (power-on), and the speical field micro-orders JTAB and RTN.

b) $\overline{\text{JORJ}}$ is low, which is a NOR of the OP field micro-orders JMP and JSB, and the CNDX micro-order is not asserted (unconditional jump is being done).

c) $\overline{\text{JORJ}}$ is low, CNDX is asserted and $\overline{\text{JMPMET}}$ is low, indicating that the conditional jump microinstruction has tested positive for a jump.

103 RIR [B-E32]

24-bit ROM Instruction Register. Implemented with four Hex-D Registers. Loaded every non-frozen P5 unless RPT FF is high during a repeat loop. Always contains the currently executing microinstruction.

104 SAVE [A-B14]

12-bit Save Register. Holds the return address for subroutine jumps.

104.1 Save is loaded from the RAR on rising edge of P2 if a JSB micro-instruction is being done, to save the address to resume execution from after the subroutine (see Section II.5.6.2 for description of subroutine returns).

104.2 It is cleared at P5 when $\overline{\text{RTN}}$ is low (NOR of JTAB and RTN micro-orders and PONB), which allows the RAR to be loaded from SAVE at P2, before SAVE is cleared. Thus after one RTN, SAVE is all Ø's, and any more RTN before the next JSB will cause a branch to location Ø in Control Store (fetch routine).

105 M [C-D43]

15-bit Memory Address Register implemented with Hex-D and Dual D Registers. Contains main memory address to be referenced.

105.1 M is loaded on rising edge of P4 from the S-Bus when the micro-instruction specifies it is to be loaded. Three store field micro-orders can specify it should be loaded:

a) M -- M is loaded from the S-Bus.

b) PNM [C34] -- M is loaded from the S-Bus and P is loaded from the T-Bus.

c) CM [E36-37] -- M is loaded if the IR indicates an MRG instruction other than a direct jump is to be done. This allows a READ to be initiated in the fetch routine of the operand address of MRG instructions before the first micro-instruction of the MRG routine is done, increasing execution speed. M is loaded if IR12+IR14+IR13($\overline{\text{IR11}}$+IR15) = "1".

105.2 The M-Register is buffered through tri-state line drivers [B46] onto the M-Bus, to address memory, unless $\overline{\text{DMAEN}}$ or $\overline{\text{MBEN}}$ is low. Thus, the CPU is the only section that can alter this M-Register. DCPC controls $\overline{\text{DMAEN}}$, which selects whether DCPC or the CPU has control of the M-Bus. The CPU M-Register pulls MB15 low.

105.3 The M-Register is gated onto the S-Bus by the M micro-order in the S-Bus field [B45]. Bit 15 is held low since CPU-generated memory addresses may be only 15 bits long.

105.4 The M-Register is also used with the ADR micro-order. See Section III.123.

106 READ, WRITE FLIP-FLOPS [E-F63]

Two D flip-flops used to generate $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ signals to memory to initiate memory cycles. When a READ or WRTE micro-order is used in a microinstruction, the appropriate FF is cleared at the end of P5 (start of the next microinstruction), causing $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ to go low. Both FF's are direct-set again at the rising edge of P2, providing a low pulse of 108 ns on the $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ lines to memory.

107 COUNTER [E-F4]

8-bit general-purpose binary counter implemented with two 74191-type synchronous 4-bit counters, set to count up only.

107.1 Count Mode: The count enable input is low if the ICNT special micro-order is specified or the Repeat FF is set [C64]. Then the counter is incremented at the end of P5. When the low four bits are all one, CNTR4 is high. When CNTR4 is high and the count enable input is low, the carry output follows the clock, providing a count clock for the upper four bits. When the upper four bits are high, HIMAX is high.

107.2 Loading: The counter is loaded with the low order eight bits of the S-Bus during P5 if the CNTR store micro-order is used [C36].

107.3 Reading: The counter is gated onto the low order eight bits of the S-Bus by the CNTR S-Bus micro-order [G5]. The high order bits of the S-Bus are pulled high by resistor pull-ups.

107.4 Uses of the Counter: CNT4 or (CNT4·HIMAX) are tested in the condition logic, Section III.113. The counter is used as a counter for repeat loops, Section II.18. The counter provides input addresses to the loader ROMs, Section III.111. It may also be used as a general purpose count by microroutines.

108 RAR MAPPERS

These are a set of 12 8-to-1 multiplexers and their associated logic which select the address to be clocked into the RAR if it is loaded at P2. RIR bits 2-Ø select one of eight possible inputs to the RAR, which are described functionally in Section II.5.2.4. JIO, JEAU, and JTAB are described in more detail below. Section II.5.2.1 shows how each is selected. If $\overline{\text{PFIFF}}$ is low, multiplexer outputs are all Ø. This is used to force RAR to Ø or 4 during initial power up sequence, Section II.14.

108.1 JIO Map [C9]: This map gates the 12-bit jump address field of Word Type 4 microinstructions to the RAR, except bits 2, 3 of the address are replaced with JIOMAPØ and JIPMAP1 instead, per Table III.108.1. Used by base set microcode to jump to one of four four-word I/O routines based on the contents of IR after the JTAB jump has determined the IR contains an I/O type instruction. All IR bits except 8, 7, 6 are ignored for JIO map.

108.2 JEAU Map [B9]: This map gates the 12-bit jump address field of Word Type 4 microinstructions to the RAR, except bits 2, 1, Ø of the address are replaced by JEAUMAP2-Ø, per Table III.108.2. Only IR bits 11, 9, 8, 7, 5, 4 affect JEAUMAP. It is used in the base set microcode after the JTAB jump when the IR contains 1 ØØØ ØØX ØXX XXX XXX (EAU-instructions). JMP JEAU is done to the location of a table of jumps which in turn jump to the start of EAU routines. See Table II.4.1 and Figure II.4.1 for diagrams of the sequence done. The displacement in this table corresponds to IR configurations of EAU instructions. Note that not all bit combinations of the IR in Table III.108.2 correspond to the exact form defined for EAU instructions. If the IR contains an EAU instruction 1 ØØØ ØØX ØXX XXX XXX_2, then JTAB causes a jump to the location labelled EAU. Then when JEAU is applied, it is already assured that IR11 and IR8 are Ø. DIV, DLD, and DST are decoded using JTAB, and are not done using JEAU. The JEAU map may be used more generally than the way it is actually used by the base set microcode.

108.3 Main Look-Up Tables [D-E9]: These are two 256x4 bit ROMs which are addressed by bits 15-8 of the IR. Their concentrated outputs represent an 8-bit address in control store corresponding to the type of machine instruction in the IR. This address is loaded into the RAR when the JTAB micro-order is used and RAR 11-8 are set to Ø.

108.4 Interrupt Jumps [E12,13,15,16]: If any selected RAR multiplexer output is a one, then the OR-tied open collector inverters [E15-16] pull their common output line low, and the OR gate [E13] lets RAR bit 2 be whatever the output of the bit multiplexer is. If all multiplexer outputs are zero (a RTN or JMP or JSB to Ø is attempted) then if RUNFF is low or a special or normal interrupt is pending [E12], the common line is allowed to go high, forcing bit 2 of the address input to RAR high. This forces a jump to 4_8 instead of Ø when a recognized interrupt request exists (see Section III.124). Base set microcode has an interrupt handling routine at location 4 (Section II.16).

IR BITS			JIOMAP		CORRESPONDING I/O INSTRUCTION
8	7	6	1	0	
0	0	0	0	0	HALT
0	0	1	0	0	STF,CLF,STO,CLO
0	1	0	0	0	SFC,SOC
0	1	1	0	0	SFS,SOS
1	0	0	1	1	MIA/MIB
1	0	1	1	0	LIA/LIB
1	1	0	0	1	OTA/OTB
1	1	1	0	0	STC,CLC

DETAILED OPERATION OF JIO MAP LOGIC

TABLE III.108.1

IR BITS						JEAUMAP			CORRESPONDING EAU INSTRUCTION INDICATED
11	9	8	7	5	4	2	1	0	
0	0	0	0	0	0	1	0	0	RRL
			0	0	1	1	0	1	ASL
			0	1	0	1	1	0	LSL
			0	1	1	1	1	1	MPY*
			1	0	0	1	1	1	MPY
			1	0	1	1	1	1	MPY*
			1	1	0	1	1	1	MPY*
0	0	0	1	1	1	1	1	1	MPY*
NON-Ø			0	0	0	0	0	0	RRR*
			0	0	1	0	0	1	ASR*
			0	1	0	0	1	0	LSR*
			0	1	1	0	1	1	illegal
			1	0	0	0	1	1	illegal
			1	0	1	0	1	1	illegal
			1	1	0	0	1	1	illegal
NON-Ø			1	1	1	0	1	1	illegal

*Not all combinations of the IR which yield these JEAUMAP values are defined as EAU instruction codes of the 21MX computer.

DETAILED OPERATION OF JEAU MAP LOGIC

TABLE III.108.2

109 CONTROL STORE

Control store is a black box which contains addressable patterns of data which may be decoded and executed by the control section as micro-instructions. It may consist of Writeable Control Store (WCS), ROMs or pROMs of different types, or ROM simulators. Several different combinations of control store components may be present at any one time, and each may have somewhat different electrical characteristics. Common features will be discussed.

109.1 Programming: Whatever the nature of control store, each component of it must be capable of responding as selectable modules of the address space and not provide outputs when those modules are not addressed (see Section II.4.1 on accessing schemes for description of modules). WCS (Model #12978A) consists of PC boards, each of which is set to respond to one module of addresses. ROMs and pROMs can be mounted on boards which are 2-module or 4-module selectable and which are mounted under the CPU board in the mainframe.

109.2 Inputs: The 12-bit RAR addresses the control store with TTL level signals. ROMs and pROMs used by HP must have high input impedance ("Ø" input current = -250μa) because in general, on HP ROM boards, each bipolar ROM IC chip address bit input is driven directly from the RAR without buffering. RAR lines not used for direct input to ROMs are used to decode the module numbers which will enable a set of ROMs. See Figure III.109.1 for a diagram of a typical ROM board.

109.3 Enabling/Disabling: The signals RMX, ENRMX, ROMEN [D31] control enabling and disabling of entire control store, as shown in Figure III.109.2.

109.3.1 RMX: TTL Signal. Sourced from either WCS or a ROM simulator, used for diagnostic testing by field service. RMX is a floating signal on WCS if the "module Ø" switch is OFF, and must be pulled high to enable WCS. When the WCS "module Ø" switch is ON and RAR11-RAR8

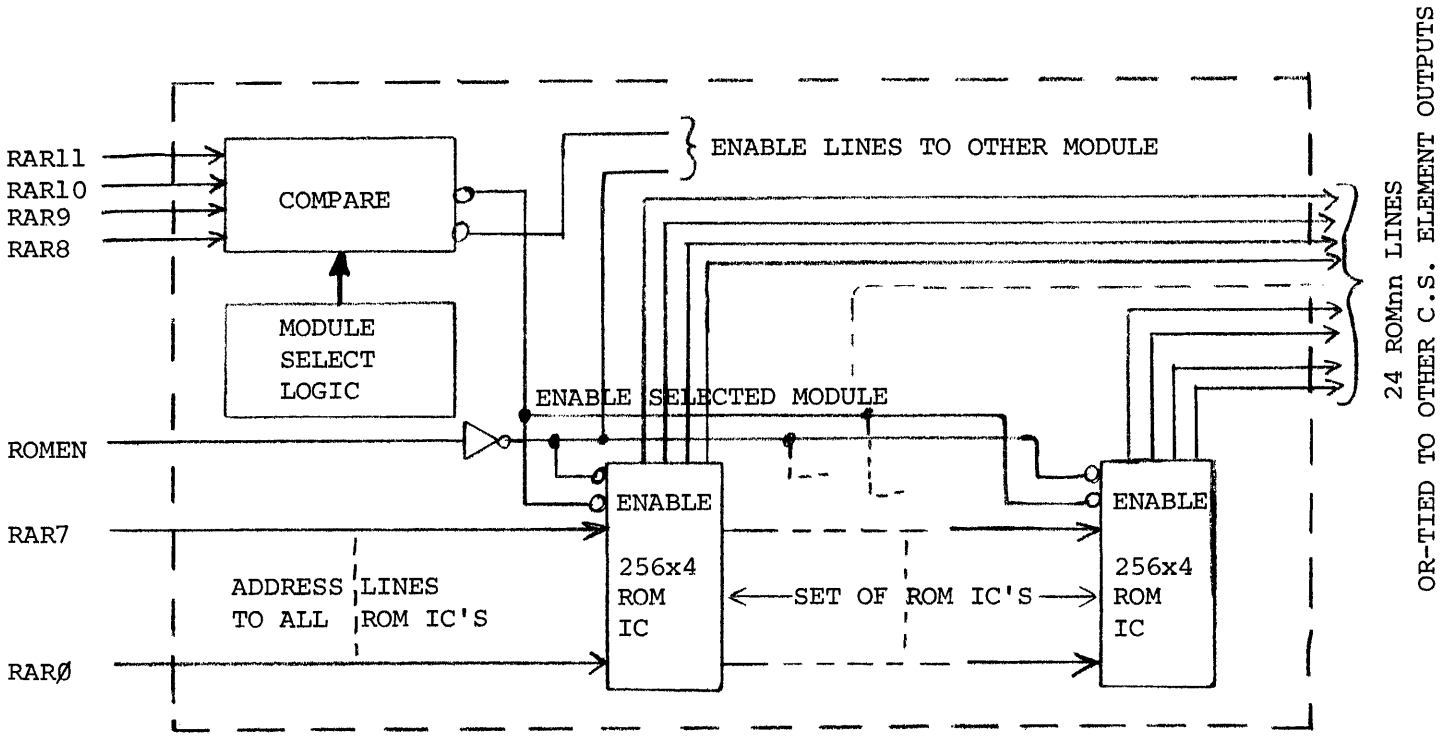
are Ø, WCS pulls RMX low which disables all of control store, causing ROM23-ROM0 to go to all ones. A JEAU PASS S S microinstruction will be done until RAR no longer addresses module 0, when RMX will go high again. The ROM simulator provides a bi-state TTL signal on the RMX line. It is pulled low when a select switch disables computer control store and enables a small alterable RAM control store in the simulator to act as the computer's control store.

109.3.2 ENRMX: TTL Signal. It is an input on WCS boards to enable WCS to respond as control store. When ENRMX goes low, WCS will not output data to control store (its tri-state driver outputs will be disabled) Tied directly to RMX so that RMX controls enabling of all control store.

109.3.3 ROMEN: TTL Signal. Follow RMX exactly. Received by every HP ROM board to enable the ROM outputs.

109.4 Outputs The outputs of all elements of Control Store must be TTL compatible and OR-tieable. When an element is not being addressed by the RAR, its outputs must be disabled (appear as an open or high impedance circuit). See Sections 109.3.2 and 109.3.3. The outputs must be 24 bits wide and appear on the ROM23-ROMØ lines [A-C31] as inputs to the RIR. Pull-up resistors keep the ROMnn lines high unless forced to another value, and provide faster rise time.

109.5 Timing: From the time the RAR is clocked to the time the data is valid at the input of the RIR must be less than half a micro-instruction cycle, .162 μsec (if the RAR is loaded at P2, control store must have valid data on ROMnn by the end of P5 to operate properly).



BLOCK DIAGRAM OF TYPICAL 2-MODULE ROM BOARD

FIGURE III.109.1

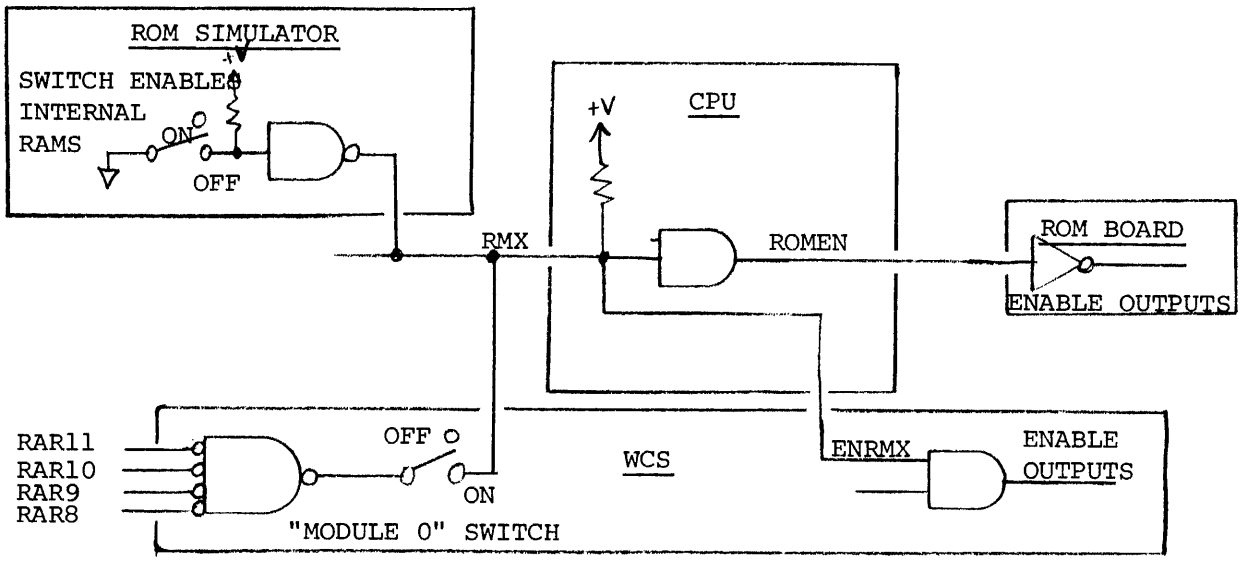


DIAGRAM OF CONTROL STORE ENABLE LOGIC

FIGURE III.109.2

110 FIELD DECODERS

The output of the RIR is divided into five fields of four or five bits each (refer to Section II.5 for formats): OP, SPECIAL, ALU (and condition field), S-BUS, and STORE. Operand fields for IMM, JMP, and JSB operations are taken directly from the RIR. IMM modifiers are discussed in Section III.112. Each of the five standard fields represents a binary-encoded function, which must be decoded to generate a control or selection function during execution of the microinstruction.

110.1 OP Field Decoder [G62]: RIR bits 23-20 are continuously decoded by a 4-to-16 line decoder. One output of the decoder will be low, corresponding to the one of 16 OP signals to be generated.

110.2 Special Field Decoder [B-D62]: RIR bits 4-Ø are continuously decoded by these two 4-to-16 line decoders. If RIR4=Ø, the decoder at [B62] is enabled. If RIR4=1, the decoder at [D62] is enabled. The one which is enabled will have one of its outputs low, causing generation of a special signal (described in Section II.5.6).

110.3 Store Field Decoder [C34]: Cross-reference Section II.5.5.6-7. RIR bits 9-5 specify one of 32 registers to be stored into during microinstruction execution.

110.3.1 If RIR9=Ø, the selected register is decoded by the 4-to-16 line decoder [C34] from RAR8-5. However, this decoder is disabled if JMP or JSB is specified in the OP field (JORJ signal = 1), since the store field is not valid during JMP or JSB.

110.3.2 If RIR9=1, the selected register is in the scratch pad (Section III.202). Then the 4-to-16 line decoder is disabled, and RIR8-5 is gated by the multiplexer [A21] during P3, 4, 5 to address the scratch pad RAMs. If JMP or JSB is specified, or MPV is high and the P or S-Registers are selected, then the scratch pads will be disabled from being altered [D34]. Otherwise $\overline{\text{RAMWEN}}$ will go low during P5, writing the T-Bus into the scratch pads.

110.4 S-Bus Field Decoder [B42]: RIR bits 14-10 specify one of 32 registers to be gated onto the S-Bus during microinstruction execution.

110.4.1 If RIR14=Ø, the selected register is decoded by the 4-to-16 line decoder [B42] from RIR13-10. However, the decoder outputs will be disabled if RIR14=1, if JMP or JSB or IMM is in the OP field, or if a DCPC S-Bus transfer is being done ($\overline{\text{DMAFRZ}}$ low during T3).

110.4.2 If RIR14=1, the decoder is disabled, and the Holding Register [C-F24] (Section III.203) is gated onto the S-Bus (cross-reference Section II.5.5.3), which holds the Scratch Pad Register addressed directly by RIR8-5 through the address multiplexer [A21]. The Holding Register buffer to the S-Bus [C25, F25] is disabled if RIR14=Ø or IMM is in the OP field, or a DCPC S-Bus data transfer is in progress ($\overline{\text{DMAFRZ}}$ low during T3).

110.5 ALU/Condition Fields: RIR bits 19-15 feed directly into the ALU and conditional logic (Sections III.204, III.113).

111 LOADER ROMS [B3-4, C3-4]

These are four IC component locations which may contain 256x4 bit ROMs. U36 contains the standard paper tape bootstrap loader ROM. All other locations contain sockets designed to accept optional ROMs/pROMs.

111.1 Selection: Exactly one loader ROM is enabled at a time, according to IR bits 15,14.

111.2 Addressing: All ROMs are continuously addressed by the Counter Register [E-F4].

111.3 Output: The 4-bit outputs of all ROMs are OR-tied. If a selected location does not contain a ROM, the pull-up resistors will hold the output lines high. When LDR is in the S-Bus field, these outputs are gated onto the low 4 bits of the S-Bus through an inverter [C5]. The high 12 bits of the S-Bus are not driven by any register, and are pulled high.

112 IMMEDIATE LOGIC [C-D42, G70, G27]

This logic helps implement word type 2 microinstructions (refer to Section II.5.4). It is enabled by IMM in the OP field.

112.1 Data Gating [C-D42]: RIR17-RIR10 are inputs to two tri-state drivers which drive the S-Bus when enabled. One will be enabled if IMM is in the OP field and DCPC is not using the S-Bus ($\overline{\text{DMAFRZ}}$ is high). RIR18=Ø gates RIR17-10 onto S-Bus 15-8 and RIR18=1 gates the RIR bits onto S-Bus 7-0. The S-Bus field and Holding Register buffer are disabled to prevent registers from being dumped onto the S-Bus during IMM.

112.2 ALU Operation [G70, G27]: IMM in the OP field forces PASS high [G70], which forces the ALU to do a "PASS S-Bus" function. RIR19 controls the mode of the ALU function, which determines whether the final output of the ALU will be the complement of the S-Bus or not (refer to Section III.204).

113 CONDITIONAL LOGIC [AREA OF A-E73]

This logic is used to test μ-instruction jump conditions. Its single output, $\overline{\text{JMPMET}}$, is low if the test would indicate a jump if JMP CNDX was in the RIR. Condition testing is done continuously, and not just when a JMP CNDX is being executed. See Section II.5.3.2 for descriptions of the input condition signals.

113.1 Condition Selection [A-C73]: RIR19-15 continuously controls one of the two 16-to-1 conditional multiplexers to select one of 32 conditions. The multiplexer output will be low only when RIR19 enables it and the input signal selected by RIR18-15 is high. If either multiplexer output is low, the output of gate U272C is high. This is exclusive-ORed with RIR14 (Reverse Jump Sense bit) to create $\overline{\text{JMPMET}}$ [A74]. Then $\overline{\text{JMPMET}}$ is low if the selected input line is not the same logical value as RIR14. $\overline{\text{JMPMET}}$ is used to load RAR if a JMP CNDX is done (see Section III.102.2).

113.2 Condition Register [E72]: This is a dual 4-bit latch which is used to latch conditions which must be preserved during sequences of JMPs. Its inputs are latched every P5 unless JMP or JSB is in the RIR. The conditions from the ALU are latched here (TBZ, ONES, COUT, ALØ, AL15). $\overline{\text{INSTEP}}$ is held here, because during the indirect address microroutine, if an indirect address occurred while the INSTR STEP button was depressed, it is possible that the button could be released during execution of JMP CNDX NSNG ____ ____ and cause an error. It is latched here to assure changes can be detected only at P5, after the JMP CNDX RAR load has been done. No other front panel switch signal is latched. $\overline{\text{MLOST}}$ is latched for the same reason as $\overline{\text{INSTEP}}$.

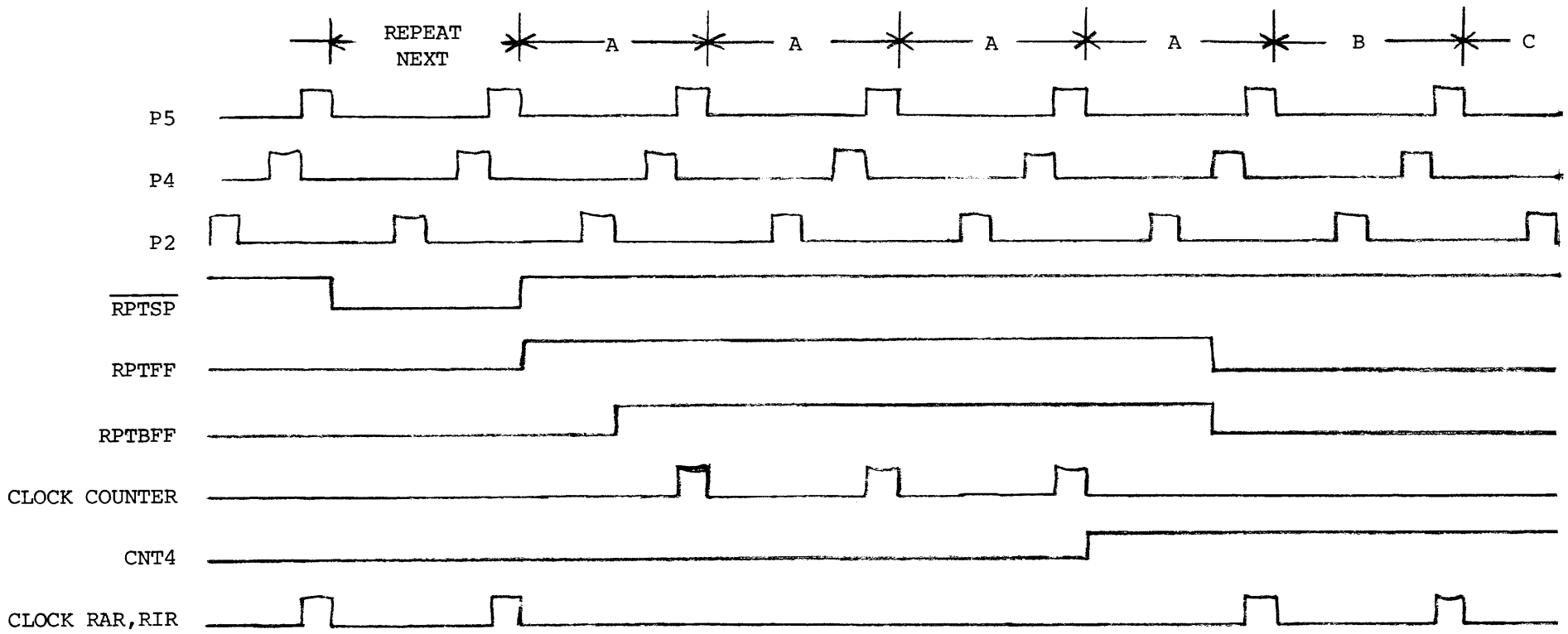
113.3 ASG Skip Conditions: Skip tests for the ASG-type machine instructions are done by the logic around [E74]. $\overline{\text{ASGN}}$ is low if the ASG skip tests would indicate a machine instruction skip (P←P+1) if an ASG instruction were being executed. Refer to Section II.5.3.2 for details.

114 REPEAT LOGIC [G14]

Consists of the RPT FF and RPTB FF and their associated logic. Used to prevent RAR and RIR from changing to repeat a microinstruction a set number of times. See Figure III.114 for a timing diagram.

114.1 Initializing: The RPT special micro-order ($\overline{\text{RPTSP}}$ low) will cause the RPT FF to be set at the end of P5, and $\overline{\text{RPT FF}}$ goes low. $\overline{\text{RPT FF}}$ disables further clocking of the RIR [E31], and enables the Counter Register to be clocked at the end of every P5 [C64]. At the end of the next P2, RPTBFF is set high, disabling further clocking of RAR. RPTBFF is used to insure that the RAR clock enable does not change too near a clock time.

114.2 Terminating: When CNT4 is high (4 low bits of counter = 1), both RPTFF and RPTBFF are cleared at the start of P4, re-enabling RAR and RIR clocking, and microinstruction execution proceeds normally.



TIMING DIAGRAM - REPEAT LOGIC

(INITIAL COUNTER CONTENTS = $XXXX1100_2$)

FIGURE III.114

115 SRG DECODERS [G8-9]

This group of decoders and multiplexers are used to set up SRG instruction-type shifts in the R/S logic to shift the ALU output onto the T-Bus. $\overline{\text{SRG1SP}}$ or $\overline{\text{SRG2SP}}$ low multiplex different IR bits to the decoders, which set up the R/S logic (TBSØ, TBS1) and create the proper linkages to affect SRG-type shifts. These signals are shown in Table III.115. See Section III.205 for the effect of these signals.

116 A/B ADDRESSABLE LOGIC [B37]

The A and B Addressable FF's are used to access A or B as memory locations Ø and 1, respectively. The TAB micro-order is used in either the Store or Enable field to let this logic decide whether T, A, or B is to be used in the data transfer. The AAFF and BAFF (flip-flops) are clocked at the end of P5 whenever M, PNM, or CM occur in the Store field (whenever M may be loaded with an address). They are set according to the value on the low order 15 bits of the T-Bus at that time, per Table III.116. Thereafter, whenever TAB occurs in the Store or S-Bus field, the register selected is T, A, or B depending on AAFF or BAFF, per Table III.116.

117 FREEZE LOGIC [G33, G37]

See Section II.2 for description of a freeze, the conditions causing it, and the effect of it. When a freeze condition has been decoded, the output of the OR gate U352-8 [G33] goes high. At the end of P2, the FREEZE FF [G37] is set ($\overline{\text{FRZFF}}$ goes low). $\overline{\text{FRZFF}}$ disables distribution of "freezable" clock signals to the system, to prevent operations which would execute micro-instructions. The FREEZE FF is clocked every P2. When U352-8 is not high at P2, the FREEZE FF is cleared at the end of P2, re-enabling all clock generation.

SRG1 / SRG2	IR BITS 9 8 7 6 / 4 2 1 0	SIGNALS GENERATED R/S CONTROL TBSØ	TBS1	OTHER SIGNALS AT TRUE VALUE	SRG SHIFT	EFFECTS SHIFT DIRECTION	SHIFT CONDITIONS CREATED
	0 X X X	1	1	NONE	NONE	--	--
	1 0 0 0	1	0	SH1, ARISS	*LS	L1	ALX14=ALU15 LSI=Ø
	1 0 0 1	0	0	SH1, ARISS	*RS	R1	ALX16 =ALU15
	1 0 1 0	1	0	ROT1	R*L	L1	ALX14=ALU14 LSI=ALU15
	1 0 1 1	0	0	ROT1	R*R	R1	ALX16=ALUØ
	1 1 0 0	1	0	ARISS	*LR	L1	LSI=Ø ALX14=Ø
	1 1 0 1	0	0	ER*NE, LWE	ER*	R1	EXFF←ALUØ ALX16=EXFF
	1 1 1 0	1	0	EL*NE, LWE	EL*	L1	EXFF←ALU15 LSI=EXFF ALX14=ALU14
	1 1 1 1	0	1	NONE	*LF	L4	--

SRG DECODER OPERATION

TABLE III.115

VALUE OF T-BUS BITS 14-Ø	AFTER CM,M,PNM AAFF	BAFF	REGISTER SELECTED BY "TAB"
0	0	0	T
1	1	0	A
2	0	1	B
OVER 2	0	0	T

TAB LOGIC

TABLE III.116

118 RUN/PRESET LOGIC [G43]

This logic is used to control the RUN and PRESET FF's from the front panel and from internal computer functions. There is no true distinction between RUN and HALT modes of operation, as microcode executes all HALT functions. The RUN FF is used as a test flag to indicate when the HALT routines should be entered. Table III.118 shows how each FF is set and cleared.

118.1 RUN FF [G44]: This J-K FF provides the testable flag, RUNFF, with which the machine determines if it is in the RUN or HALT mode.

118.2 HALT BUFFER FF [H43]: The clock to this D-FF comes from the front panel HALT button. The R/C connection to the clock provides additional de-bouncing of the HLTB line. It can be set only if RUNEN is high (key switch not in lock position). It is cleared when RUNFF is low so that the K input of the RUNFF is kept low in the HALT mode.

118.3 RUN BUFFER FF [G43]: The clock to this J-K FF is the debounced and strobed run button, RUNB signal, from the front panel. RUNEN must be high (key switch not in LOCK position) to let the RUN switch set it. It is cleared when RUNFF is high, so that the J input of the RUNFF is kept low in the RUN mode.

118.4 PRESET FF [F43]: This D-FF, when set, provides signals which initialize the computer to a predefined "PRESET" state (refer to Section II.17):

a) Direct-clear the Parity Error FF, PARFF [F44]
b) Direct-clear AAFF and BAFF [B37]
c) Generate CRS and POPIO to the I/O system every T5 [C-D59]
d) Generate $\overline{\text{EXHLT2}}$, per Table III.118 [A59]
e) Clear IRQ4FF if set [B56], at end of T6
f) Direct-clear the INTSEN FF (disable the interrupt system) [E56]
g) Direct-set the Power Fail Control FF ($\overline{\text{CONT4FF}}$ goes low)
h) Direct-clear the IR [B2]
i) Clear the REPEAT FF, RPTFF [G13] at end of P5
j) Direct-clear the Overflow FF [B67]

FF	OPERATION	SIGNAL	COND.	SOURCE
RUN	SET AT P5↑	$\overline{\text{SRUNSP}}$	LOW	SRUN IN SPECIAL FIELD
		------	HIGH	RUN BUFFER FF SET
	CLEARED AT P5↑	------	HIGH	HALT BUFFER FF SET
	CLEARED DIRECT	PONB	LOW	POWER SUPPLY - POWER NOT YET ON AND STABLE, OR ABOUT TO SHUT OFF
		$\overline{\text{EXHLT1}}$	LOW	$\overline{\text{PE}}$ LOW AND S1 SET TO HALT ON PARITY ERROR
		$\overline{\text{EXHLT2}}$	LOW	PRESET FF SET, S2 SET TO ARS, AND PWU LOW (POWER NOT UP)
		$\overline{\text{EXHLT3}}$	LOW	HALT I/O INSTRUCTION, T3
HALT BUFFER FF	SET AT P5↑	$\overline{\text{SHLTSP}}$	LOW	SHLT IN SPECIAL FIELD
	SET	HLTB↑		HALT BUTTON PRESSED ON FRONT PANEL AND RUNEN HIGH
	CLEARED	RUNFF	LOW	HELD CLEAR IN HALT MODE
RUN BUFFER FF	SET	RUNB↑	↑	RUN BUTTON PRESSED ON FRONT PANEL AND RUNEN HIGH
		PWUST	↑	S2 SET TO ARS POSITION, POWER-UP SEQUENCE BEGUN (SECTION III.400) AND PRESET FF NOT SET
	CLEARED	$\overline{\text{RUNFF}}$	LOW	RUN FF SET
		PONB	LOW	POWER NOT YET UP AND STABLE, OR ABOUT TO SHUT OFF.
PRESET	SET AT P5↓	PONB	LOW	POWER NOT YET UP AND STABLE, OR ABOUT TO SHUT OFF
		$\overline{\text{PRSTB}}$	LOW	PRESET BUTTON DEPRESSED ON FRONT PANEL AND RUN FF NOT SET
	CLEARED AT P5↓	PONB	HIGH	POWER IS GUARANTEED STABLE
		$\overline{\text{PASTB}}$	HIGH	PRESET BUTTON RELEASED

OPERATION OF RUN, PRESET LOGIC

TABLE III.118

119 FRONT PANEL

The front panel is a separate assembly (#5060-8343) from the CPU. It contains two registers, switches and debouncers, and LED indicators to allow CPU-operator communication. Refer to the schematic of the front panel and the block diagram (Figure III.119.1) in the following discussions. All LED's are driven by TTL gates in their low state. Intensity is controlled with a 200 Ω current/limiting resistor in series with the LED. Voltage accross the LED is about 2V, with a current of ∿ 20 ma. See Section I.11 for descriptions of the interface signals.

119.1 Display Register (DSPL): 16-bit R-S Register implemented with NAND gates. May be directly set/reset by pressing row of rocker switches. The LED to display a particular bit of DSPL is directly above the rocker switch for that bit on the panel. DSPL cannot be cleared by storing from the S-Bus. Whenever data is to be transferred from the CPU to the DSPL, $\overline{\text{DSPCL}}$ is issued before $\overline{\text{DSPST}}$ to clear DSPL before storing into it (see Section II.11). A rocker switch on the front panel may be used to do a DSPCL (clear DSPL to Ø).

119.2 Display Indicator (DSPI): This is an 8-bit register, 6 bits of which are used to indicate which internal register is being displayed. It is loaded from S-Bus 7-Ø. Bits 5-Ø control an array of six LED's which will be active if the corresponding bit is low. As seen from the operator side of the front panel, the labeled LED which will be lit is shown below.

BIT LOW	7	6	5	4	3	2	1	Ø
LED ON	NONE	NONE	S	P	T	M	B	A

The high order bits of the DSPI are not stored from or enabled to the S-Bus. An R-C network on the clear input makes this control pin voltage rise much slower than the power supply +5 volt during initial turn-on. This causes all LED's to be turned on during initial turn-on, to provide the operator with a guaranteed visual indication of restoration of power.

119.3 Direct LED Displays: Six flip-flops from the CPU drive LED's through inverters on the front panel: RUN, EXTEND, OVERFLOW, INTERRUPT SYSTEM ENABLE (INTL), POWER FAIL (PF), PARITY ERROR (PARFF). These provide visual indication of CPU-register states.

119.4 Control Switches: There are several rocker switches which create signals to the CPU see Section I.11 for definitions and functions. Two methods of eliminating switch bounce transients are used: passive filter, using just an R-C constant to smooth the switch signal; active filter using a transistor and flip-flop to provide an extended and synchronous strobe (see Section III.119.5) for switch signals which must be valid at specific times in the CPU.

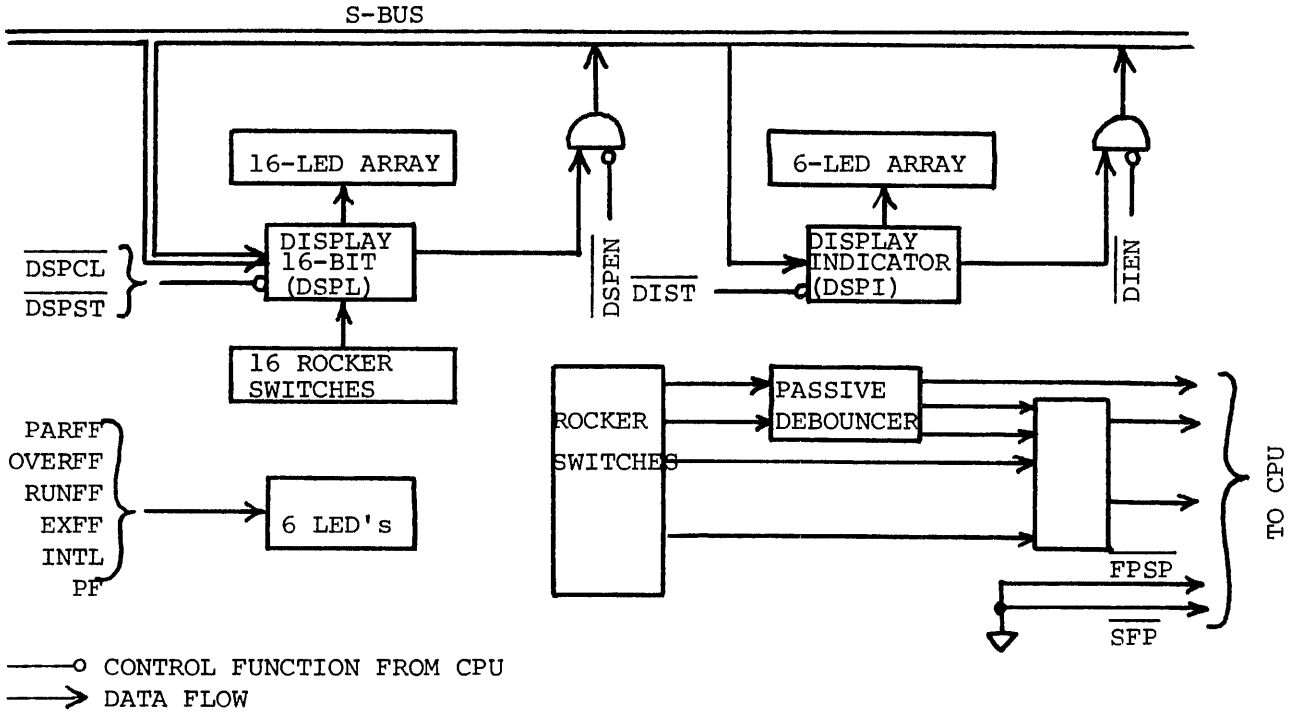
a) HLTB is debounced using only a passive filter, as it only clocks the HALT Buffer FF on the CPU.

b) $\overline{\text{PRSTB}}$, RUNB — Debounced passively and actively to start only after P5.

c) $\overline{\text{LEFT}}$, $\overline{\text{RIGHT}}$, $\overline{\text{INCM}}$, $\overline{\text{DECM}}$, $\overline{\text{INSTEP}}$, $\overline{\text{IBL}}$, $\overline{\text{STORE}}$, $\overline{\text{DISPLAY}}$ — Debounced actively to ensure they start only at end of P5, so when front panel microcode is doing JMP CNDX, the transition won't occur around P2 when it could foul up the $\overline{\text{JMPMET}}$ timing.

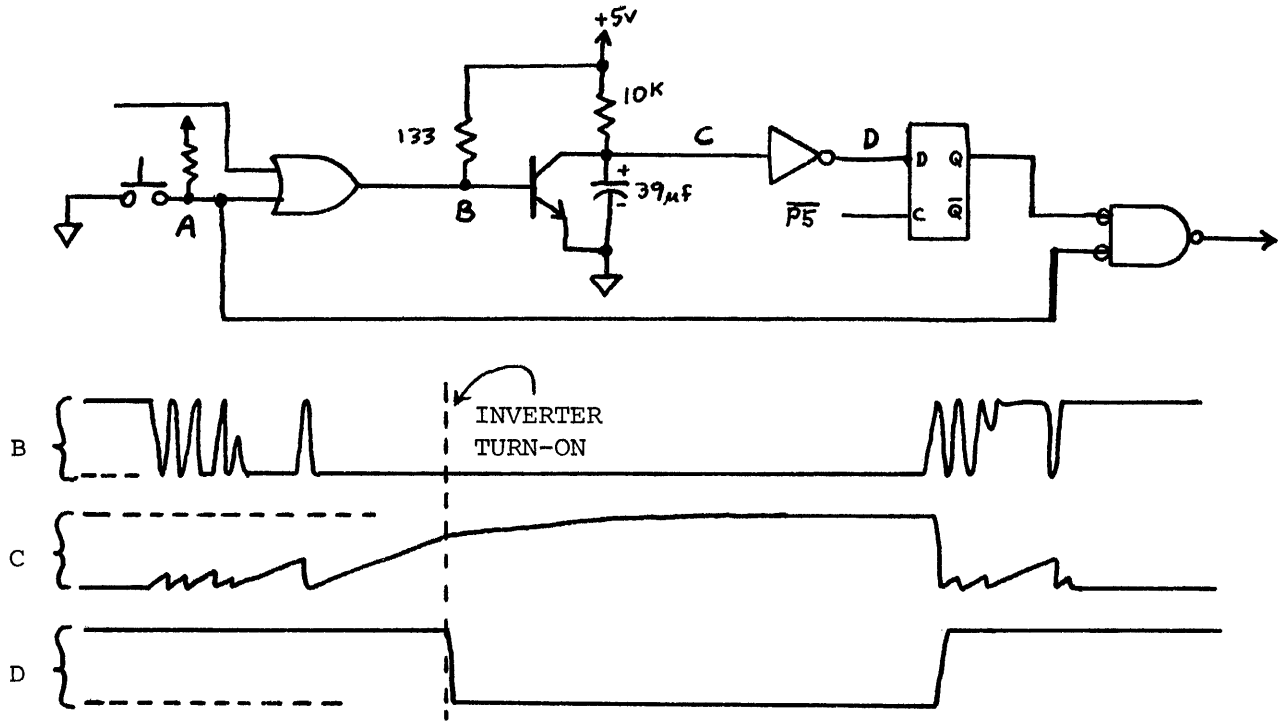
119.5 Active Debouncer ($\overline{\text{STROBE}}$): Debouncing is illustrated in Figure III.119.2. All rocker switches which use active debouncing use a common circuit. Pressing any switch will cause the debouncer to operate, as shown in Figure III.119.2. As long as no switch is pressed, the transistor is ON, rapidly dumping the change on the capacitor. When the switch makes contact, point A is pulled low, and B goes low, turning off the transistor. The time constant of the inverter low-input resistance and the capacitor start raising voltage B. At two volts, the 10K resistor and capacitor bring point B to +5V after the inverter turns on.

This rise time means B must be low for at around 40 ms before the inverter output goes low. The strobe is gated with the switch so the switch signal will be started at the end of P5. When the switch is released, the transistor is turned on and dumps the capacitor charge very fast. The strobe signal is detectable with microcode. $\overline{\text{STROBE}}$ will be low when any one of the control switches is pressed, except HALT.



FRONT PANEL BLOCK DIAGRAM

FIGURE III.119.1



ACTIVE DEBOUNCER AND TIMING EXAMPLE

FIGURE III.119.2

120 PARITY LOGIC [F44, A57]

Parity errors are detected within memory. $\overline{PE}$ will be low only when a parity error occurs. This sets the Parity FF, PARFF [F44], to light the indicator on the front panel.

PARFF is cleared by the PRESET FF or by $\overline{RESPE}$ (low) when IAK is issued if memory protect is requesting interrupt service.

$\overline{PE}$ will reset RUNFF with $\overline{EXHLT1}$ if switch S1 is in the "HALT or PARITY" position [A57]. Otherwise $\overline{PE}$ will not affect CPU operation. Memory protect controls PE interrupts.

121 INTERRUPT ENABLE FF [A64]

J-K FF, clocked at end of every P5. ION and IOFF special micro-orders turn it on, off. $\overline{\text{MPINTON}}$ from memory protect will direct set it when indirect counter reaches 3.

INTENFF enables I/O interrupt requests to be recognized by the control section after they are resolved and allowed by the I/O section. INTENFF low, no interrupts from DCPC or I/O devices can be sensed by the CPU. Memory protect, parity, and power fail interrupts are still recognized. Also see Section III.124.

122 T-REGISTER

16-bit S-Bus Register. Holds data for all memory references. T is contained within the memory controller in the memory section. Following a READ micro-order, it can be enabled onto the S-Bus with valid data exactly two microinstructions later (2nd 325 nsec of the READ cycle). After that, it is disabled from the S-Bus by the memory system.

123 ADR LOGIC [D45, B3]

Consists of a set of S-Bus drivers which select data to put onto the S-Bus when the ADR micro-order is used in the S-Bus field. It creates an address on the S-Bus which is the operand address of MRG-type instructions in the IR. IR7-Ø is driven onto the S-Bus 7-Ø at [B3]. If IR10=Ø, then IR9-8 are driven onto S-Bus 9-8, and S-Bus 15-10 are driven to Ø [E45] by selecting one 8-bit driver. If IR10=1, the other driver is selected [D45] which drives M-Register bits 15 (zero) to 10 and IR9-8 onto S-Bus 15-10.

124 INTERRUPT JUMP LOGIC [F12]

This combinatorial logic converts I/O section interrupt signals into flags to be used by the control section for testing for interrupt conditions.

124.1 INTFLG: High if the INTEN FF is on and a normal interrupt is pending, or if $\overline{\text{SPECINT}}$ is low (memory protect, parity, or power fail interrupt requested). Used for JMP CNDX test for interrupt requests. Note that if INTENFF is low (see Section III.121), then normal interrupts will not be detected.

124.2 $\overline{\text{HOI}}$: Low if INTFLG is high or the RUNFF is not set. May be tested by JMP CNDX microinstructions with NHOI condition mnemonic. $\overline{\text{HOI}}$ also controls the line which forces loading of RAR to 4 instead of Ø when low (see Section III.108.4).

125. CPU CLOCK GENERATION [E34, F35, G38]

125.1 Oscillator [E34]

The oscillator circuit uses a 18.5 MHz crystal and an ECL gate feeding back on itself to maintain accurate clock frequency [E34]. Transistor Q1 is connected across the outputs of the second ECL gate so that it is turned on during one half of each oscillation. When turned on, Q1 drives current through CR1, providing a high logic level across R21 to NAND gate U392D, making the clock input high on the Ring Counter comprised of PAFF, PBFF, PCFF. During the other half of the oscillation, Q1 is shut off, and R21 pulls down the input of U392D, clocking the Ring Counter with a high-to-low transition.

125.2 The Clock Generator [F35]

The 3-stage Ring Counter is clocked at 18.5 MHz (54.05 nsec clock). The Ring Counter cycles through six states, decoded from PAFF, PBFF, PCFF, of 54.05 nsec each. Each is called a P-period, labelled PØ, P1, P2, P3, P4, P5. CPU timing decoding is shown in Figure II.1.2. Various combination of the P-periods are done by a bank of gates [F-H38] to provide inverted, multiple, and buffered P-period clocks to the system. Some gates include the FREEZE ff to disable clocks if a freeze condition occurs (see section II.102)

125.3 External Control - [E35]

Two NAND gates provide external triggering of the clock generator circuits through pins on the J3 connector. CLKXEN is normally pulled high by R19B, enabling the 18.5 MHz oscillator. If it is pulled to Common (low level), the 18.5 MHz oscillator will not be able to propagate a trigger to the Ring Counter. Then the Ring Counter can be advanced only by a high-to-low transition on the CLKX line. Each time this is done, the Clock Generator advances one P-period. The CLKX line is pulled high by R19C if not driven externally.

200 ARITHMETIC/LOGIC SECTION

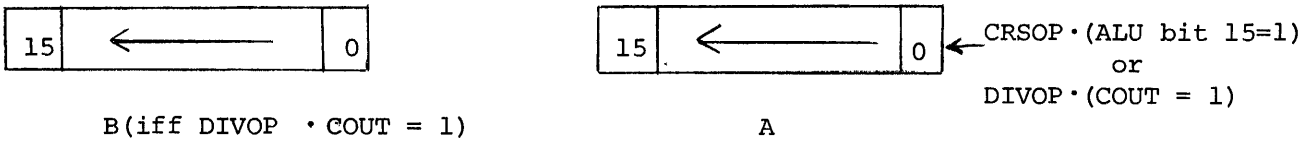
A block diagram of this section of the CPU board is shown in Figure III.200. This section of the computer contains the logic which performing arithmetic and logical operations on data, and contains the data registers of the Computer which are used for data operations. Refer to Section II.5.5 for additional operational detail.

201 A,B, REGISTERS AND LOGIC

A [E-G2] and B [E-G3] are each 16-bit Parallel-load, Right/left shift registers. They are the registers referenced by most machine instructions. They may be loaded from the T-Bus, and may be gated onto the S-Bus through tri-state bus drivers [B-G6]. A and B are clocked at the trailing edge of P5. The mode inputs S_1 and S_2 on each register select the action to be taken at that time, as shown in Table III.201.

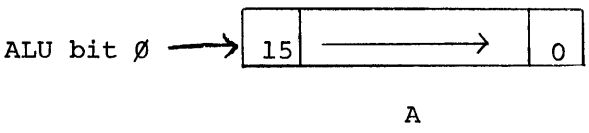
201.1 Left Shift

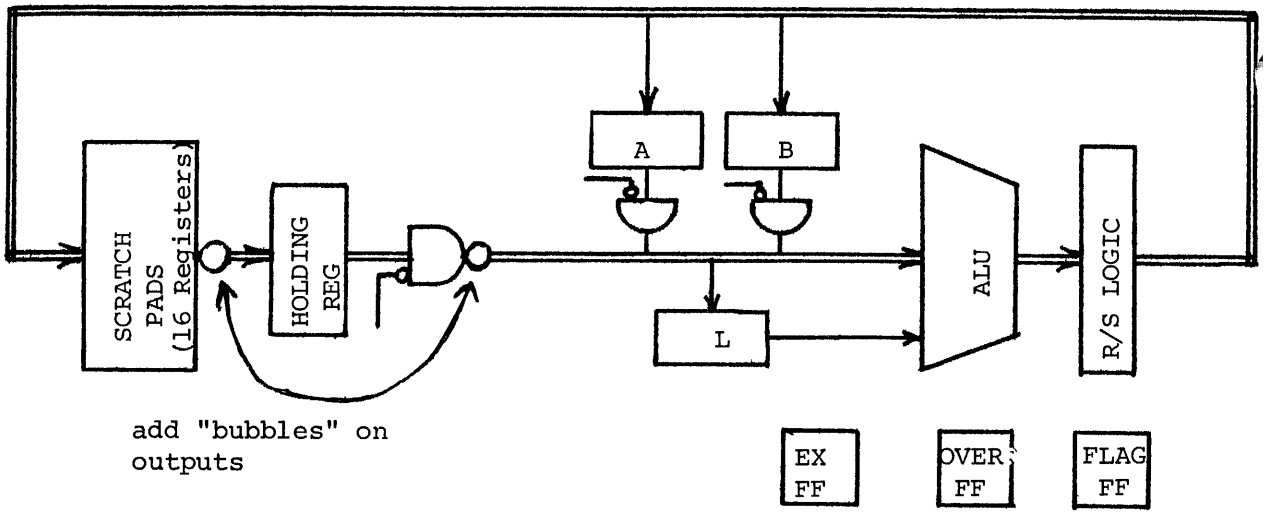
When a left shift is done, the outputs of each register shift as shown:



201.1 Right Shift

When a Right shift is done, the outputs of the A register are shifted as shown (B cannot be shifted right):





BLOCK DIAGRAM OF ARITHMETIC/LOGIC SECTION

FIGURE III.200

	MODE SELECT S1	MODE SELECT SØ	ACTION AT P5↓	CONDITIONS CAUSING THIS MODE SELECTION
A-Reg (S1=AS1, SØ=ASØ)	Ø	Ø	NO CHANGE	NO OPERATION SPECIFIED
	0	1	Shift Right One Bit	MPY in Op field, or CRS,ARS,or LGS in Op field and Rl in Special field.
	1	0	Shift Left One Bit	DIV in Op field, or a) B in store field and b) Ll in Special field and c)LGS,CRS or ARS in Op field.
	1	1	Load From T-Bus	A selected from TAB, CAB, or "A" μ-order in store field
B-Reg (S1=BSTORE, SØ=BSØ)	Ø	Ø	NO CHANGE	No operation specified
	Ø	1		Not possible
	1	Ø	Shift Left One Bit	B selected by TAB,CAB, or B in store field, and $\overline{COUT}$ is high and DIV is in Op field
	1	1	Load from T-Bus	B selected by TAB,CAB, or B in store field, and $\overline{COUT}$ is low if DIV is in Op field

OPERATION OF A,B REGISTERS

FIGURE III.201

202 SCRATCH PADS [B-G22]

These are 16 16-bit general purpose registers contained in an array of 4 16X4 bit bipolar RAM IC's. They contain the registers called S1 through S12, P(Program Address), S(Switch), X and Y(Index). They are addressed by the RIR through a 4-bit multiplexer [A21] which selects RIR8-5 during times P3-P5, and RIR13-1Ø during PØ-2. If RIR9=1, or PNM is in the Store field of the RIR, and the OP field does not contain JMP or JSB, then during P5, $\overline{\text{RAMWEN}}$ is low [D34], which writes the data on the T-Bus into the register in the Scratch Pad selected by the store field.

The outputs of the RAMS are open-collector and must have pullup resistors to provide fast rise times. The output of the RAM is the inverse of the value stored into the addressed register from the T-Bus.

203 HOLDING REGISTER [B-F24]

This is a 16-bit latch comprised of two Dual 4-bit latches. The latch is open during PØ-2. At the leading edge of P3, the enable input goes high, latching the data from the Scratch Pad register selected by RIR13-10. This preserves the data from the Scratch Pads to drive the S-Bus while RIR8-5 select a scratch register into which to store the T-Bus data. The output of this latch is gated onto the S-Bus through an inverting tri-state bus driver if RIR14=1 (S-Bus RIR field selects one of the Scratch Pad registers) and there is no IMM in the OP field and $\overline{\text{DMAFRZ}}$ is not low (see section II.18) [D25].

204 ALU [A-G28, A-G29, A-G30]

This logic performs logical and arithmetic operations on the data on the S-Bus and in the L-Register. If consists of 4 74181-type function generators with a 74182-type carry generator logic chip. The A=B outputs are tied together so iff the output of the ALU is 177777_8, the ONES logic signal will be high.

204.1 Function Selection

The operation to be performed is determined by RIR bits 19-15 normally (refer to Table II.5.2 for a list of operations, and to section II.5.5 for descriptions of operations). RIR bit 19 controls the mode input.

This disables the internal carry logic when high, and creates a logical function instead of an arithmetic one. When either IMM is in the Op field of the RIR or when the MPY μ-order is in the Op field and A-Register bit Ø is Ø, the PASS line is high. This sets the function select inputs to low and the carry into the low-order function generator to 1. This forces the ALU to pass the S-Bus. If RIR19 is low, it will be complemented at the output. No modification will be done to it otherwise.

204.2 Carry Out

Operations containing the word "MINUS" in their description in section II.5.5 are done as follows.

a) the operand following the word "MINUS" is ones-complemented internal to the ALU.

b) this "negative" value is then added to the rest of the expression enternally.

c) the carry-in to the lower-order ALU IC is held low iff RIR18 is low and PASS is low. It changes the operation from ones-complement to two's complement. It is a carry-in of one if it is low

$\overline{\text{COUT}}$ will be low iff the twos complement addition result produces a carry out of bit 15 of the ALU output.

205 ROTATE/SHIFT LOGIC

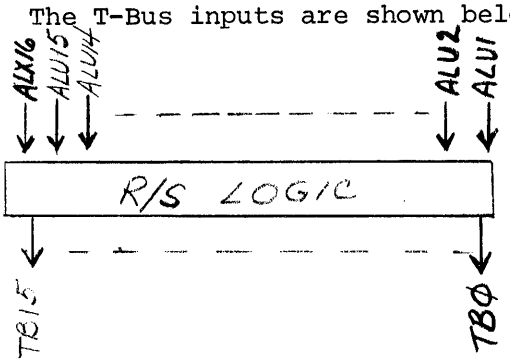
The output of the ALU is input to a bank of 8 dual 4-to-1 multiplexers [F71-77]. The Control Lines TBSØ, TBS1 select one of the four inputs to create the effect of one of four shifts. Several different μ-orders can affect the shift selected, and what the state of the bit shifted into the most or least significant bit will be. Refer to section III.115 for effects of the SRG-decoders. Extend and OVERFLOW registers also participate in shifts.

205.1 PASS (TBSØ = TBS1 = 1)

No Special μ-order affecting the R/S logic is in the RIR. The ALU is not shifted, but passed onto the T-Bus.

205.2 R1 (TBSØ = TBS1 = Ø)

Enabled by R1 μ-order or some IR patterns when SRG1 or SRG2 μ-orders are used. The T-Bus inputs are shown below.

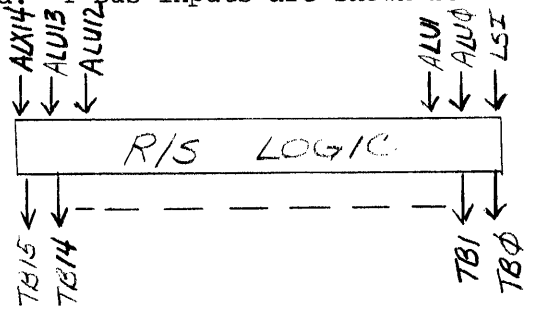


ALX16 will assume the state of one of the following signals [D69]:

A Reg. bit Ø; FLAG FF; EXFF; ALUØ. The signal selected depends on the presence of the following μ-orders: SRG1; SRG2; ARS; LWF; MPY. Refer to section describing these μ-orders for details.

205.3 L1 (TBSØ = 1, TBSØ = Ø)

Enabled by L1 μ-order or some IR patterns when SRG1 or SRG2 μ-orders are used. T-Bus inputs are shown below:



ALX14 will assume the state of ALU14, unless the ARS, SRG1, or SRG2 μ-orders are used, in which case it would have the value of ALU15[C79]. See sections describing these μ-orders for details.

LSI will assume the stae of ALU15, A-Register bit Ø, EXFF, or FLAGFF, depending on the presence of the following μ-orders [F69]: LGS; ARS; CRS; DIV; LWF; SRG1; SRG2.

205.4 L4 (TBSØ - Ø, TBS1 = 1)

The T-Bus will contain the ALU outputs as they would appear after a 4-bit rotate left.

206 FLAG FLIP-FLOP [C67]

This D flip-flop is clocked at the trailing edge of P5 only when the STFL, CLFL, RES2, or LWF μ-orders are used. It may be set/cleared by STFL, CLFL, or exchanged with the EXTEND FF by RES2, or participate in Ll or Rl shifts.

207 EXTEND FLIP-FLOP [G67]

This JK flip flop is clocked at the trailing edge of P5. It may be exchanged with FLAGFF using the RES2 μ-order, or participate in SRG-type operations using the μ-orders, SRG1, SRG2, SRGE, or be set/cleared by the ASG μ-order, or be set if an ALU operation results in a $\overline{\text{COUT}}$ low level using the ENVE μ-order.

208 OVERFLOW FLIP-FLOP [B67]

This J-K flip-flop is clocked at the trailing edge of P5. It may be set/cleared/tested by I/O instruction cycles, or set/cleared by SOV/COV μ-orders, or set during execution of an ARS μ-order with a Ll μ-order if ALU15 $\neq$ ALU14, or set by the ENV or ENVE μ-orders if S-Bus 15 = LATCH 15 $\neq$ ALU15.

209 L(LATCH) - REGISTER [B-F27]

This is a 16-bit latch-type register which holds an operand from the S-Bus for input to the ALU. The latch is open only during P5 when the L μ-order is used in the store field. It is cleared at P5 when the ASG μ-order is used. Bit 15 is used by the OVERFLOW Logic (section III.208).

300 I/O SECTION

A block diagram of the I/O section is shown in Figure III.300. All signals which are sent to or received from I/O device interface cards are pulled low unless driven high. Outputs are current source in high state, open circuit otherwise. Lines are pulled down through resistors on CPU board and interface cards to -2v. I/O signals to interface boards are driven by 8T13 (Signetics) - type line drivers.

301 I/O TIMING GENERATOR [A-C76]

This is the logic which provides clock signals which are used to maintain synchrony throughout the I/O section, interface boards, I/O extenders and Memory Protect, DCPC, and Power-Fail/Auto Restart operation.

301.1 The I/O section timing is explained in general in section II.1.1. A cyclic counter is used to generate TA, TB, TC signals. TAFF, TBFF, TCFF are buffered and sent off the CPU board to the DCPC to be used for generating DCPC I/O sequences.

301.2 TAFF, TBFF, TCFF are used by the signal generation logic to create I/O signal timing. Three time periods are buffered directly up the I/O backplane:

a) T3

b) T2 renamed ENF (ENable Flag). It is used to set interface card Flag ff's synchronously to begin interrupt priority resolution and to ensure no flag gets set in the middle of some other I/O operation. Flags are only to be set during T2, before I/O control signals are generated.

c) T5, renamed SIR (Set Interrupt Requests). Occurs after I/O Control Signals to set the Interrupt Request ff on the device interface with the highest priority which is ready to interrupt.

302 INTERFACE CARDS

302.1 General

The I/O interface cards provide control functions for interfacing I/O devices and special control functions (e.g.: Time-Base Generators, Writeable

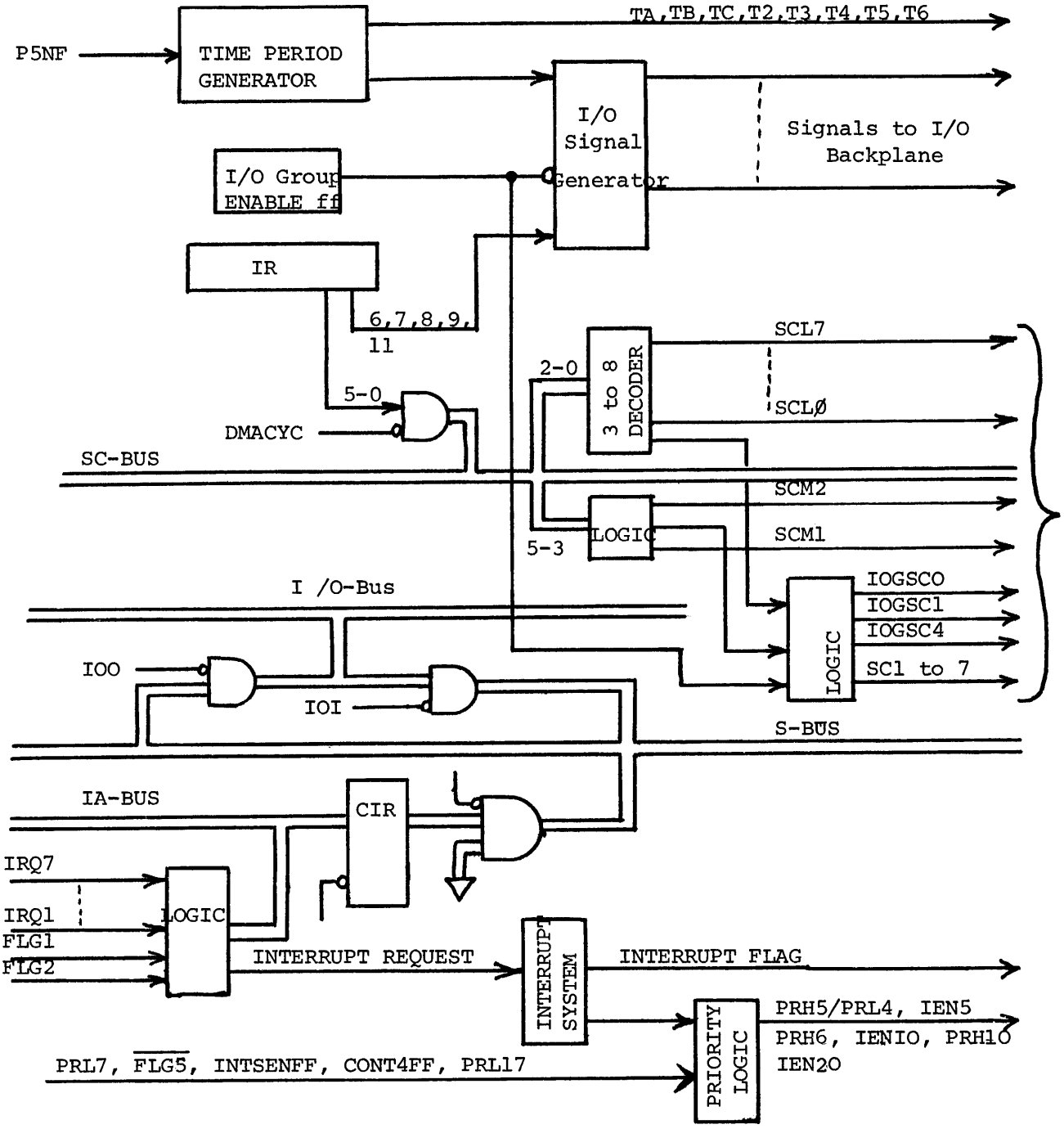
Control Store) to the computer through the I/O section. They provide the logic necessary to buffer data between an external device and the computer, to request interrupt service when an operation is complete, to execute control functions on cue from the computer and to provide electrical signal interfacing between external devices and the computer. The exact functions performed by any interface card are, of course, purely dependent on the design of the card. Sometimes a device requires 2 or 3 interface cards. Some interface cards cannot generate interrupt requests, etc. Figure III.302.1 shows the design of a typical interface card. Elements will be discussed when appropriate in the Theory of Operation of the I/O section.

302.2 Pin Assignments

302.2.1 The end of the interface card which plugs into the I/O backplane has 86 connector pins, numbered 1 through 86 through which the interface board gets power, control and data paths to the main computer I/O section. Figure III.302.2 shows the connector pin number assignments for interface cards. Odd-numbered pins 1 through 85 are on the component side, as shown. Even-numbered pins are on the other side, with pin 2 directly opposite pin 1.

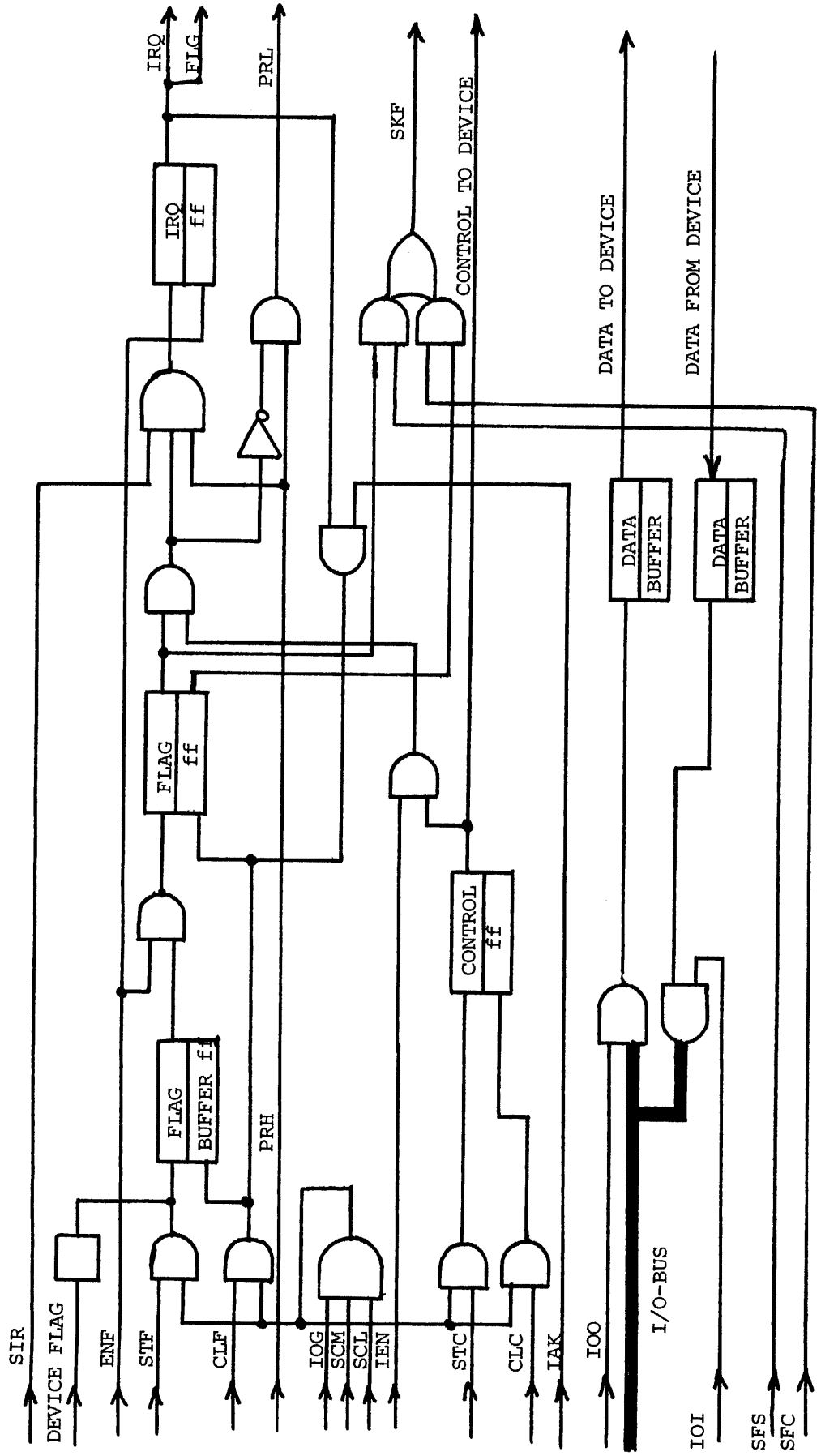
302.2.2 The other end of the interface card is used for various purposes depending on the specific interface card. Most standard I/O devices connect through a cable to this end which has a 48-pin connector. Pins 1 through 24 are on the component side, as shown in Figure III.302.2 and consecutively-lettered pins A through BB (G,I,O,Q, omitted) are on the other side, pin A opposite pin 1.

302.2.3 Refer to Table III.302 for a list of the signals assigned to each pin of every I/O backplane slot. A particular interface card may not use all the signals (refer to individual interface kit schematics). All I/O backplane slots have identical pin assignments so that any interface card may be plugged into any slot. Some pins have slightly different significance in each I/O slot which make that slot unique with respect to I/O select code, and interrupt priority and requesting. (SCL, SCM, PRL, PRH, IRQ, and FLG signals). Note that there are two sets of



BLOCK DIAGRAM OF I/O SECTION

FIGURE III.300



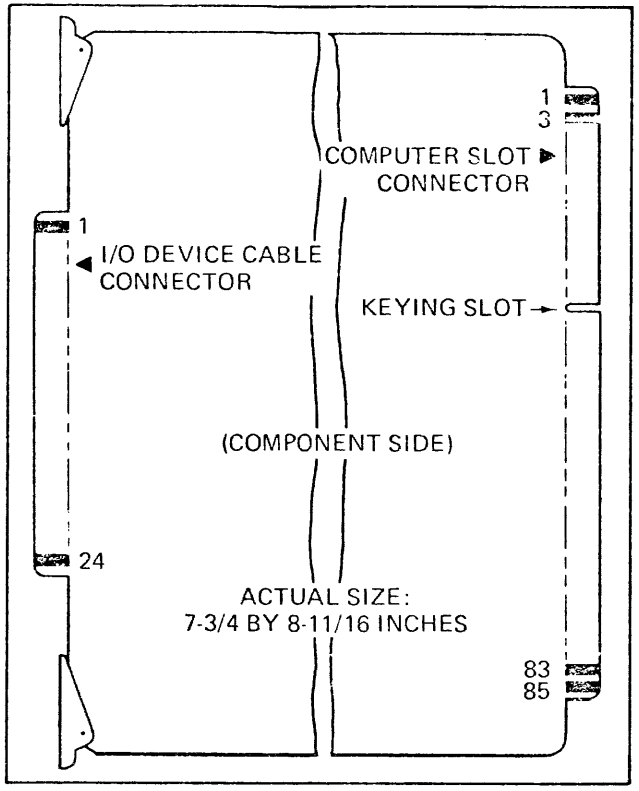
BLOCK DIAGRAM OF TYPICAL I/O INTERFACE BOARD

FIGURE III.302.1

TABLE III.302. Interface Card-to-Computer Pin Connections

PIN	SIGNAL MNEMONIC AND DEFINITION	PIN	SIGNAL MNEMONIC AND DEFINITION
1	GND: Ground	2	GND: Ground
3	PRL: Priority Low	4	FLGL: Flag signal, Lower Select Code
5	SFC: Skip Flag Clear	6	IRQL: Interrupt Request, Lower Select Code
7	CLF: Clear (reset) Flag FF	8	IEN: Interrupt Enable
9	STF: Set Flag FF	10	IAK: Interrupt Acknowledge
11	T3: Machine time T3 to I/O	12	SKF: Skip Flag (True if SFS or SFC test is true)
13	CRS: Control Reset	14	SCM: Select Code Most Significant Digit (Lower Address)
15	IOG: I/O Group Instruction	16	SCL: Select Code Least Significant Digit (Lower Address)
17	POPIO: Power On Preset to I/O	18	IOBI 16: I/O Bus Input, Bit 16
19	SRQ: Service Request to DCPC	20	IOO: I/O Date Output Signal
21	CLC: Clear (reset) Control FF	22	STC: Set Control FF
23	PRH: Priority High	24	IOI: I/O Data Input Signal
25	SFS: Skip Flag Set	26	IOBI 0: I/O Bus Input, Bit 0
27	IOBI 8: I/O Bus Input, Bit 8	28	IOBI 9: I/O Bus Input, Bit 9
29	IOBI 1: I/O Bus Input, Bit 1	30	IOBI 2: I/O Bus Input, Bit 2
31	IOBI 10: I/O Bus Input, Bit 10	32	SIR: Set Interrupt Request (T5)
33	IRQH: Interrupt Request, Higher Select Code	34	SCL: Select Code Least Significant Digit (Higher Address)
35	IOBO 0: I/O Bus Output, Bit 0	36	+30 volts
37	SCM: Select Code Most Significant Digit (Higher Address)	38	IOBO 1: I/O Bus Output, Bit 1
39	+5.00 volts	40	+5.00 volts
41	IOBO 2: I/O Bus Output, Bit 2	42	IOBO 4: I/O Bus Output, Bit 4
43	+12 volts	44	+12 volts
45	IOBO 3: I/O Bus Output, Bit 3	46	ENF: Enable Flag (T2)
47	-2 volts	48	-2 volts
49	FLGH: Flag Signal , Higher Select Code	50	RUN
51	IOBO 5: I/O Bus Output, Bit 5	52	IOBO 7: I/O Bus Output, Bit 7
53	IOBO 6: I/O Bus Output, Bit 6	54	IOBO 8: I/O Bus Output, Bit 8
55	IOBO 11: I/O Bus Output, Bit 11	56	IOBO 9: I/O Bus Output, Bit 9
57	IOBO 12: I/O Bus Output, Bit 12	58	IOBO 10: I/O Bus Output,Bit 10
59	(Not used)	60	IOBO 11: I/O Bus Output,Bit 11
61	IOBO 13: I/O Bus Output, Bit 13	62	EDT: End Data Transfer (from DCPC)
63	(Not used)	64	IOBI 3: I/O Bus Input, Bit 3
65	IOBO 14: I/O Bus Output, Bit 14	66	PON: Power On Normal
67	(Not used)	68	(Not used)
69	-12 volts	70	-12 volts
71	(Not used)	72	(Not used)
73	SFSB: Skip Flag Set Buffered	74	IOBO 15: I/O Bus Output, Bit 15
75	Not Used	76	(Not used)
77	IOBI 4: I/O Bus Input, Bit 4	78	IOBI 12: I/O Bus Input, Bit 12
79	IOBI 13: I/O Bus Input, Bit 13	80	IOBI 5: I/O Bus Input, Bit 5
81	IOBI 6: I/O Bus Input, Bit 6	82	IOBI 14: I/O Bus Input, Bit 15
83	IOBI 15: I/O Bus Input, Bit 15	84	IOBI 7: I/O Bus Input, Bit 7
85	GND: Ground	86	GND: Ground

NOTES:
1. Pins 1&2, are connected together on slot connector and on interface card. Pins 39&40, 43&44, 47&48, 69&70, and 85&86 are also connected together.
2. Corresponding IOBO and IOBI bit lines are connectd together on the slot connector.



STANDARD I/O INTERFACE CARD PIN ASSIGNMENTS

FIGURE III.302.2

IR BITS						I/O Signal	Time Periods
11	10	9	8	7	6		
X	X	X	X	X	X	IOG	T3-T5
X	X	X	0	0	0	CLEAR RUN	T3
X	X	X	0	0	1	STF	T3
X	X	X	0	1	0	SFC	T3-T5
X	X	X	0	1	1	SFS	T3-T5
X	X	X	1	1	0	IOO	T3-T4
X	X	1	X	X	X	CLF	T4
0	X	X	1	1	1	STC	T4
1	X	X	1	1	1	CLC	T4
X	X	X	1	0	X	____	____

X indicates don't care bits

I/O SIGNAL GENERATION

TABLE III.303.1

these signals: One set for a lower select code (the slot number of that I/O backplane connector) and a higher select code (for the next higher slot number). This is for interface cards which are used in pairs, using up two select codes for all the necessary control and data functions of the device.

303 SIGNAL GENERATOR [C76,A-F78]

This logic, which includes the I/O Group Enable ff[C76], generates I/O signals specified by IR bits 6,7,8,9,11 at the proper time periods.

303.1 Normal (non-controlled) I/O Cycle

With the I/O Group Enable ff (IOGENFF) off, the only I/O signals generated are ENF, T3, SIR (section III.301.2). DCPC could steal an I/O cycle. It has its own signal generation logic, and may create IOO, IOI, CLC, STC, STF, CLF, IOG, and EDT from the DCPC board. The I/O signal lines will be low until driven high (current source) by CPU or DCPC.

303.2 I/O Instruction Cycle

303.2.1 When the IOG μ-order occurs (IOGSP high), a CPU freeze occurs until T2. At T2, IOGEN ff will be set, unless $\overline{\text{MPV}}$ is low or $\overline{\text{DMACYC}}$ is low. If Memory Protect is enabled (previous STC 5 command), then $\overline{\text{MPV}}$ will go low during T2 resetting IOGENFF before T3. When IOGENFF is set, I/O signals will be generated according to IR bits 6,7,8,9,11 as shown in Table III.303.1. IOGENFF is buffered and sent as IOG. Note that the machine instruction CLF actually results in STF, CLF, in that order. During T5, the K input to the IOGEN ff is made high, and IOGENFF is set low at the end of the next P5, disabling further signal generation.

303.2.2 In addition to the standard I/O signals, if the select code on the Select Code-Bus (SC-Bus) is Ø, 1, or 4, internal signals are generated to control the interrupt system, Front Panel, Extend ff, and Power-Fail systems as indicated in Table III.303.2.

Select Code / I/O Signal	IOGSC0	IOGSC1	IOGSC4
STF	INTSENFF ← 1 AT T3↑ [D56]	OVERFF ← 1 AT P5↑ of T3 [B66]	NO EFFECT
CLF	INTSENFF ← 0 AT T4↑ [E56]	OVERFF ← 0 AT P5↑ of T4 [B67]	NO EFFECT
SFC	SKPF = SKF = 1 during T3-T5 iff INTSENFF=0 [E59]	SKF = 0, SKPF=1 during T3-T5 iff OVERFF=0 [B68]	SKPF=SKF=1 during T3-T5 iff DIRFF=0 (power failure)
SFS	SKPF=SKF=1 during T3-T5 iff INTSENFF=1 [E59]	SKF=0, SKPF=1 during T3-T5 iff OVERFF=1 [B68]	NO EFFECT SKPF=SKF=0
IOO	NO EFFECT	$\overline{\text{DSPCL}}$=0 during T3, $\overline{\text{DSPST}}$=0 during T4 (clear and load $\overline{\text{DSPL}}$ from S-Bus [B80])	NO EFFECT
IOI	S-Bus=I/O Bus (all Ø) during T4-T5	$\overline{\text{DSPEN}}$=0 during T4-T5 (S-Bus=DSPL)	S-BUS=CIR during T4-T5 [C55]
STC	NO EFFECT	[B80] NO EFFECT	$\overline{\text{CONT4FF}}$ ← 0 at T4↑ [F56]
CLC	CRS=1 during T4 [D59]	NO EFFECT	$\overline{\text{CONT4FF}}$ ← 0 at T4↑ [F56]

EFFECTS OF I/O SIGNALS FOR SELECT CODES 0,1,4

TABLE III.303.2

303.2.3 IOI is a microcode function, and does not occur by contents of the IR. IOI occurs only if the IOI S-Bus μ-order is used during T4-T5 [A78].

304 PRIORITY CHAIN LOGIC [E-G59]

The priority chain establishes hierarchy of interruptibility. DCPC, Memory Protect, Power Fail logic, and each I/O interface card are assigned a fixed interrupt priority. Priority is directly related to device select code: the lower the select code, the higher its priority. When a high priority device is ready to interrupt, it disables the priority chain to higher select codes, preventing them from interrupting. The process is shown in Figure III.304, and the role of the priority in device interrupt requests is shown in Figure III.302.1.

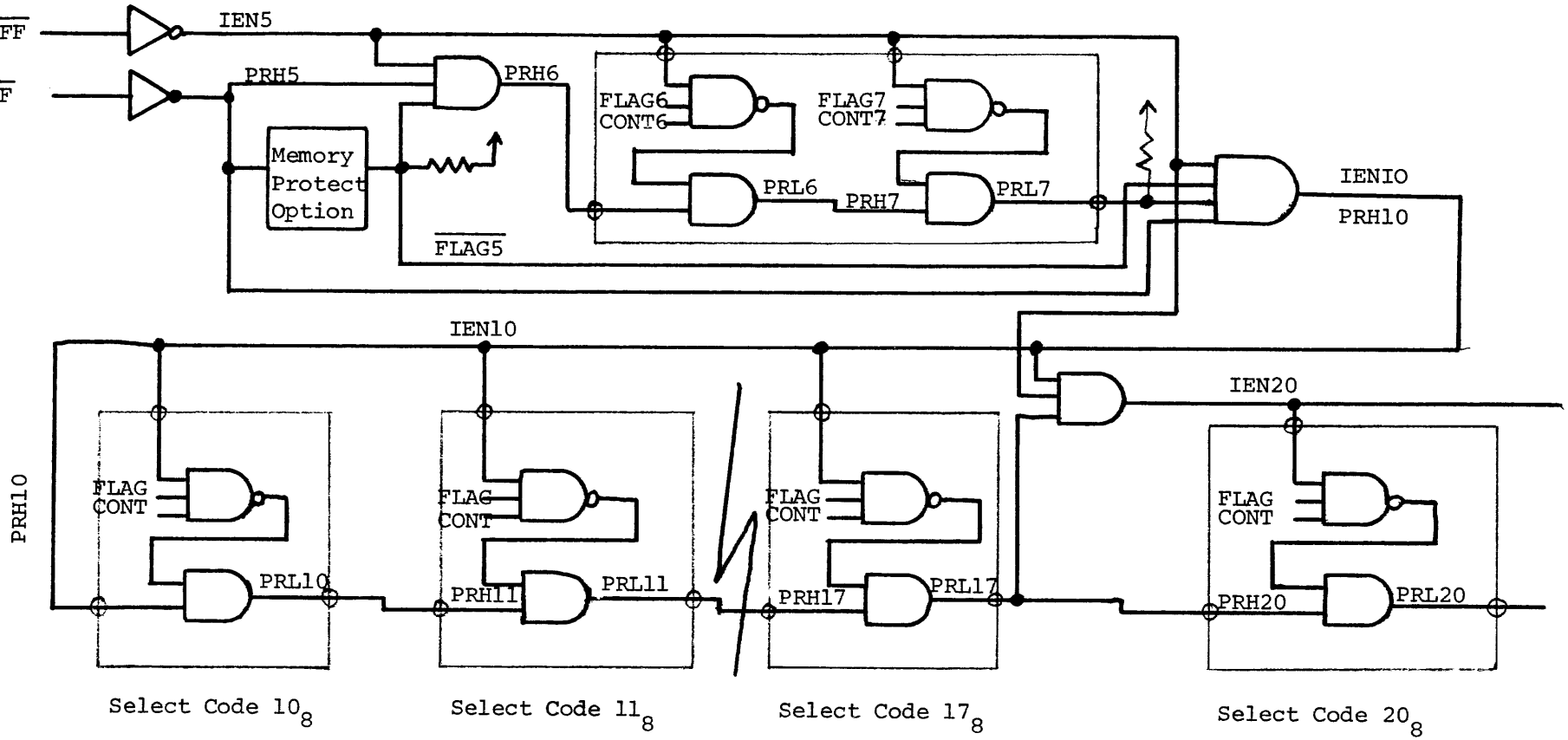
303.1 Hierarchy

Order in the priority chain, from highest priority to lowest, is: Power Fail/Auto Restart logic; Memory Protect and Parity; DCPC channel 1; DCPC channel 2; I/O select code positions 10 through 77, in that order.

The priority chain is received on a card by the PRH line, which is the priority output of the next lower select code card. PRH will be high if no lower select code device is going to request an interrupt. PRL is the priority line output from a card, and will be low if PRH is low or the device is going to request an interrupt (usually because the FLAG ff and CONTROL ff and IEN and PRH are all high).

304.2 Operation

All PRHXX and PRLXX signals will be 0 if $\overline{\text{CONT4FF}}$ is high. Only Power Fail can interrupt then. The priority chain can be enabled only if INTSENFF is high. This is buffered as the IEN5 signal, which goes to Memory Protect and DCPC. When $\overline{\text{CONT4FF}}$ = 0, PRH5/PRL4 is high. If M.P. is not going to interrupt, $\overline{\text{FLG5}}$ is high. Then PRH6 to DCPC will be high. If DCPC is not going to interrupt, PRL7 will be high. PRH10 and IEN10 will be high if no select code $<10_8$ is going to interrupt. The PRH/PRL chain propagates serially through the computer I/O slots. To prevent propagation delays from being too great in the event of an interrupt, thus failing to disable all lower priority devices before they can interrupt, IEN signals connect up to 8 cards in parallel (IENS goes to DCPC, MP; IEN10 goes to select codes $10\text{-}17_8$).



NOTE: Logic in boxes represents typical I/O interface and priority logic. All other logic is on CPU board in I/O section, or in DCPC or MP as indicated.

PRIORITY CHAIN OPERATION

Figure III.304

305 INTERRUPT LOGIC [around E57]

This logic controls generation of interrupt flags to the CPU in an orderly fashion. There are two categories of interrupts:

a) Normal Interrupts: interrupts due to DCPC or standard I/O devices sending IRQ and FLG signals during T5-T6, which interrupt requests can be ignored by the CPU by turning off the INTERRUPT ENABLE ff. Resolving of such requests is dependent on I/O timing and may be overridden during I/O cycles, interrupt servicing, or DCPC cycles.

b) Special Interrupts; interrupts which must be generated without delay to perform special services. Memory Protect, Parity, and Power-Fail Interrupts are Special Interrupts.

See Figure III.305 for a diagram of this logic.

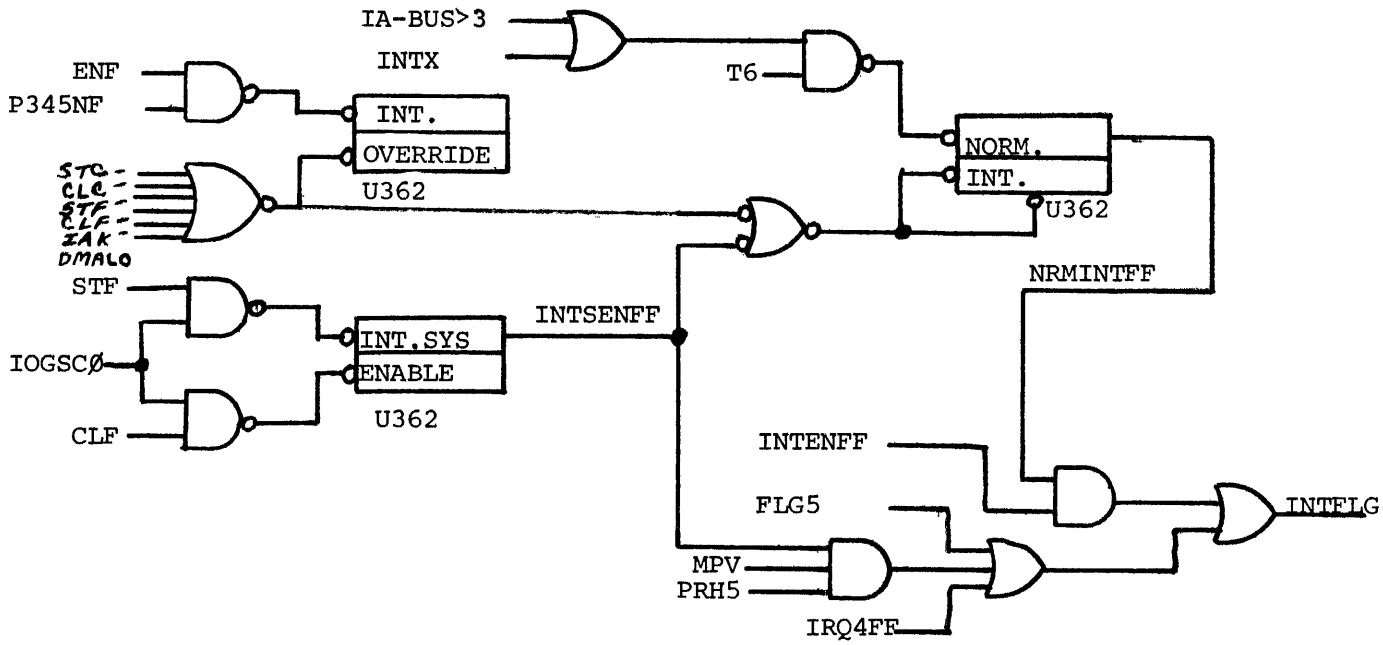
305.1 Normal Interrupts

For Normal Interrupts to be generated, the priority chain must be enabled ($\overline{\text{CONT4FF}} = \emptyset$) and INTSENFF must be 1 (Interrupt System Enable ff) [E57]. INTSENFF is set by a STF $\emptyset$ machine instruction and cleared by CLF $\emptyset$, or PRESET. When INTSENFF is set, the INTL signal [G60] to the Front Panel lights the Interrupt System indicator.

305.1.1 The Interrupt Request:

A typical I/O device (Figure III.302.1) will request an interrupt in the following way:

1) When the device finishes an operation, the FLAG Buffer is set on the interface board.
2) At the next T2 (ENF), the FLAG is set.
3) If the CONTROL is set, too, and IEN and PRH are high, then the interface pulls PRL low, insuring that all lower priority interfaces cannot interrupt. If a higher priority has its FLAG and CONTROL set, then PRH will be low, and this interface cannot request.
4) If FLAG, CONTROL, IEN, and PRH are all 1, then during T5-T6 the IRQ and FLG will be high. If microcode executes the CIR μ-order, then at T6 with IRQ high, IAK (P3,4,5, of T6) will reset the FLAG and FLAG Buffer to prevent further requesting.



INTERRUPT SYSTEM LOGIC

FIGURE III.305

305.1.2 Interrupt Address [B-D52]

Only one interface may request an interrupt at any T5-T6 period. Each interface in the I/O backplane will activate one of seven IRQ lines and one of the two FLG lines (FLG1, FLG2) when requesting. The inverter array at [B-D52] encodes theses signals into the select code of the interrupting interface and places it on the IA-Bus, inverted. The IA-Bus also goes to I/O extenders. If an interface in an I/O extender is requesting, its address will be present on the IA-Bus and $\overline{\text{INTX}}$ will be Ø [E51]. See figure III.300. If there is an address on the IA-Bus (not all ones), or $\overline{\text{INTX}}$ is low, then INTREQ will be 1 [D54], which indicates an IRQ is occurring.

305.1.3 Generating Normal Interrupts [E57]

NRMINTFF is set at T6 if INTREQ is 1 (U362-5) (Normal Interrupt ff), unless U362 pin 6 is low. If pin 6 is low, NRMINTFF will be held low, holding off generation of the interrupt from INTREQ. This will occur if INTSENFF is clear, or if the Interrupt Override ff (INTOVRFF) is clear. INTROVRFF is set every T2, but it is cleared if any I/O control signal is issued, if a DCPC cycle is in progress, or if an Interrupt Acknowledge (IAK) is generated. This prevents Normal Interrupts during operations which may affect interrupt priority. For example, if select code 50_8 is ready to interrupt, but a STC12 instruction occurs with FLAG12 ff already set, then there may not be enough time to disable 50_8 from generating IRQ at T5. INTOVRFF will prevent the interrupts until the priorities can be firmly established by T5. IAK clears INTOVRFF to clear NRMINTFF once the interrupt has been acknowledged by the CPU. DMACYC clears INTOVRFF because DCPC issues STC,CLC,CLF,and STF commands during its cycles.

305.2 Special Interrupts [D57]

Special interrupts are recognized independent of the normal interrupt logic because it is of critical importance that they be recognized before fetch of the next machine instruction.

305.2.1 Parity Interrupt

If Memory Protect is enabled to interrupt on detection of Memory parity errors and S1 on the CPU [A56] is in the INT/IGNORE position, then $\overline{\text{FLAG5}}$ will go low at T2 after $\overline{\text{PE}}$ goes low, flagging $\overline{\text{SPECINT}}$ = Ø. INTSENFF need not be 1. PRH5 must be 1, however.

305.2.2 Memory Protect Interrupt:

When a Memory Protect or I/O violation occurs, $\overline{\text{MPV}}$ goes low, which disables certain CPU functions. MP can only interrupt if PRH5 = 1 and INTSENFF = 1. Then $\overline{\text{MPV}}$ will cause $\overline{\text{SPECINT}}$ to go low. $\overline{\text{MPV}}$ is used instead of FLAG5 because the interrupt request must occur as soon as possible to service it before the next machine instruction can be fetched. $\overline{\text{FLAG5}}$ will not go low until time T2.

305.2.3 Power Fail Interrupt

Whenever IRQ4FF is set (see section III.400), $\overline{\text{SPECINT}}$ is low. INTSENFF need not be set.

305.3 Interrupt Recognition [D56]

NRMINTFF is gated with the INTERRUPT Enable ff, which controls recognition of generated Normal Interrupts in the CPU (see section III.121). The OR of Special Interrupt with gated NRMINTFF is the Interrupt Flag in the Interrupt Detect Logic of the Control Section (section III.124).

306 SELECT CODE BUS [F51]

The 6-bit Select Code Bus (SC-Bus) contains the select code of the I/O interface card addressed by I/O instructions. It is always enabled, always addressing an interface card slot. Refer to figure III.300.

306.1 Sources

Normally, IR bits 5-0 drive the SC-Bus through a tri-state driver [F52]. When DCPC takes a cycle, $\overline{\text{DMACYC}}$ goes low, transferring control of the SC-Bus to DCPC.

306.2 Destinations

The SC-Bus goes directly to the I/O extenders, where it is decoded to address the extender I/O backplane slots. It is also decoded on the CPU to address the computer mainframe I/O slots, and options and special functions. It also is used to address DCPC and MP.

306.3 Decoding

306.3.1 Bits 0-2 of the SC-Bus are decoded [F53] into eight signals, SCLØ through SCL7, which are driven onto the I/O backplane. They correspond to the least significant octal digit of the select code. SCLØ is the SCL input for slots 10_8, 20_8, SCL1 is the SCL input for slots 11_8, 21_8, etc.

306.3.2 Bits 5-3 of SC-Bus are decoded [G53] into three signals, SCM1, SCM2, and SCMØ (not labeled as such, U203A), corresponding to the upper digit of the select code. SCM1 is the SCM input to I/O slots 10_8-17_8, SCM2 is for slots 20_8-27_8, etc. A slot is addressed by AND-ing SCM and SCL on the interface card. (Figure III.302.1)

306.3.3 SCLØ, SCL1, SCL4, SCMØ, and IOG are decoded [F-G55] (U182) to address internal CPU features which are controllable through the I/O section as select codes Ø,1,4 (see section III.303.2.2)

307 I/O BUS GATING [A-H48]

The I/O-Bus is driven by either the CPU (output data transfer) or an I/O interface card (input data transfer). It can drive the S-Bus through a tri-state buffer [D-G49]. Refer to Figure III.300.

307.1 I/O-Bus to S-Bus

The I/O Bus is buffered onto the S-Bus iff the SC-Bus is Ø or greater than 7_8 and either the IOI μ-order is used, or DCPC is doing an input transfer ($\overline{\text{DMAIOI}}$ is low at the T3).

307.2 S-Bus to I/O-Bus

The S-Bus data is buffered onto the I/O-Bus if the IOO μ-order is used and the I/O Group Enable ff is set, or if DCPC executes an

output transfer $\overline{\text{DMAIOO}}$ is low during T3). If DCPC sends $\overline{\text{DMAIOO}}$ at T3, then it will hold $\overline{\text{DMALCH}}$ low from P4 or T3 to P4 of T5. This latches the I/O Bus on itself so that the S-Bus may be freed for CPU use while DCPC completes the I/O data transfer with the latched S-Bus data.

308 CENTRAL INTERRUPT REGISTER [C54]

The CIR is loaded from the IA-Bus with the complement of the select code of an interrupt-requesting interface at P3 of T6 when the CIR μ-order is used (CPU freezes until T6 when CIR occurs in S-Bus field) [H52]. IAK is generated at the same time to prevent future IRQ's from the requesting interface. The CIR is buffered through an inverting tri-state driver onto the S-Bus and the high-order 10 bits of the S-Bus are driven low. The CIR may be buffered onto the S-Bus without clocking it during T4-T5 of an I/O instruction cycle when the IOI S-Bus μ-order is specified and the SC-Bus = 04_8.

400 POWER FAIL/AUTO RESTART SECTION [B56-60, E57]

This section is responsible for controlling the power-up and power-down response of the computer. Functional operation of power-up and power-down are described in Sections II.14, II.15.

The Power Fail Logic (hereafter call PF) is actually a small state-machine. The state is determined by IRQ4FF and DIRFF in the 6-bit Register U382 [B59]. The next state is determined by the present state and the conditions selected by the dual 4-to-1 multiplexer U381 [B58]. Figure III.400.1 illustrates the state machine operation, and Figure III.400.3 shows the logical organization.

The PF logic responds to changes in the state of the PWU and PON signals from the power supply, shown in Figure III.400.2. During initial power-up, PWU comes up before PON. Both go high only after the power supply voltages are well established. When a power failure occurs, PWU goes low 500μsec before PON. Both go low before operating power is removed from the CPU board.

401 INITIAL CONDITIONS

When PON is low, all ff's in the PF Register [B59] and the CONTROL 4 ff [E57] in the interrupt system register are kept low by direct-clear. So initially, when PON first comes high: $\overline{\text{CONT4FF}}$ = Ø, IRQ4FF = Ø, DIRFF = Ø, PWUFF = 0, and PF1FF = PF2FF = PF3FF = Ø. Also, $\overline{\text{PON}}$ holds the RUN ff clear. So initially, RUNFF = 0 [H43]. The PF Register is clocked at the end of every T6. PWUFF is clocked at T6 to synchronize power fail detection with the rest of the logic.

402 POWER-UP

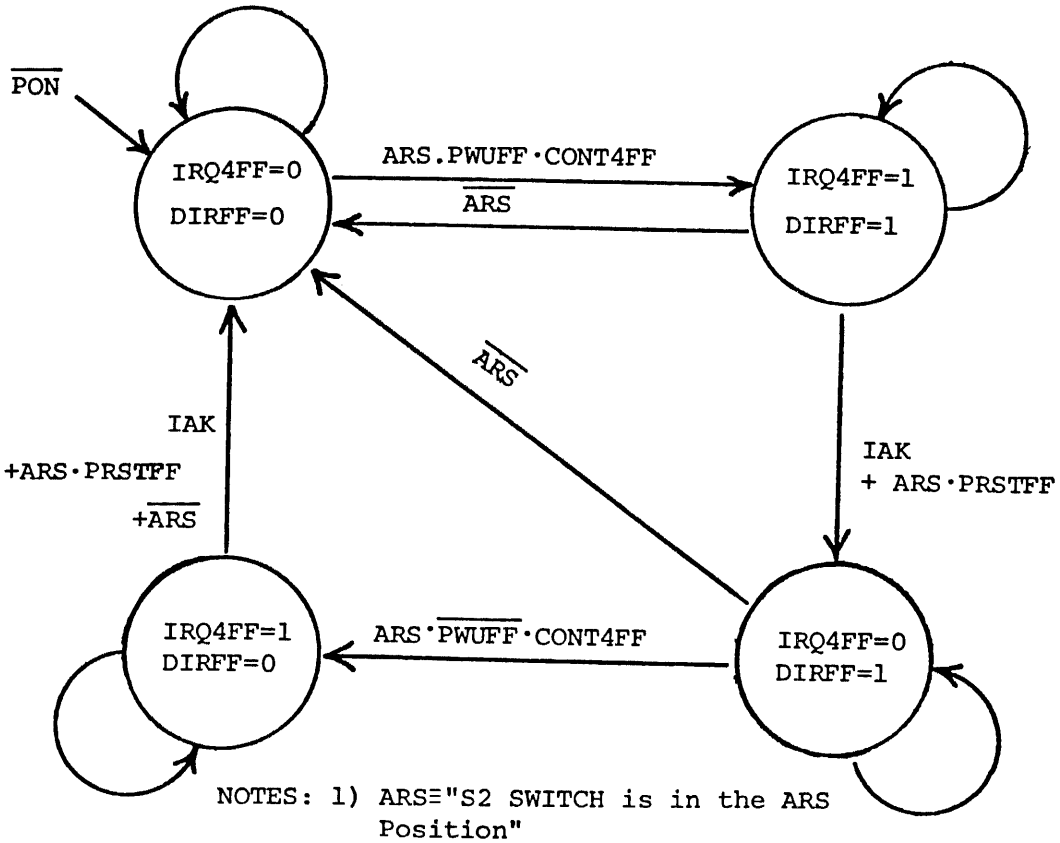
There are three sequences of PF logic states possible for initial power-up, depending on the state of the PRESET ff and the ARS switch, S2.

402.1 Switch S2 in $\overline{\text{ARS}}$ Position

In this position, S2 disables the next-state logic (outputs=Ø), and IRQ4FF = DIRFF= Ø always. It also prevents PWUST from setting the RUN ff, so the machine must come up in the HALT mode. $\overline{\text{CONT4FF}}$ remains Ø, so the PF indicator on the Front Panel is not turned on.

402.2 Switch S2 in ARS Position and PRESET ff Set

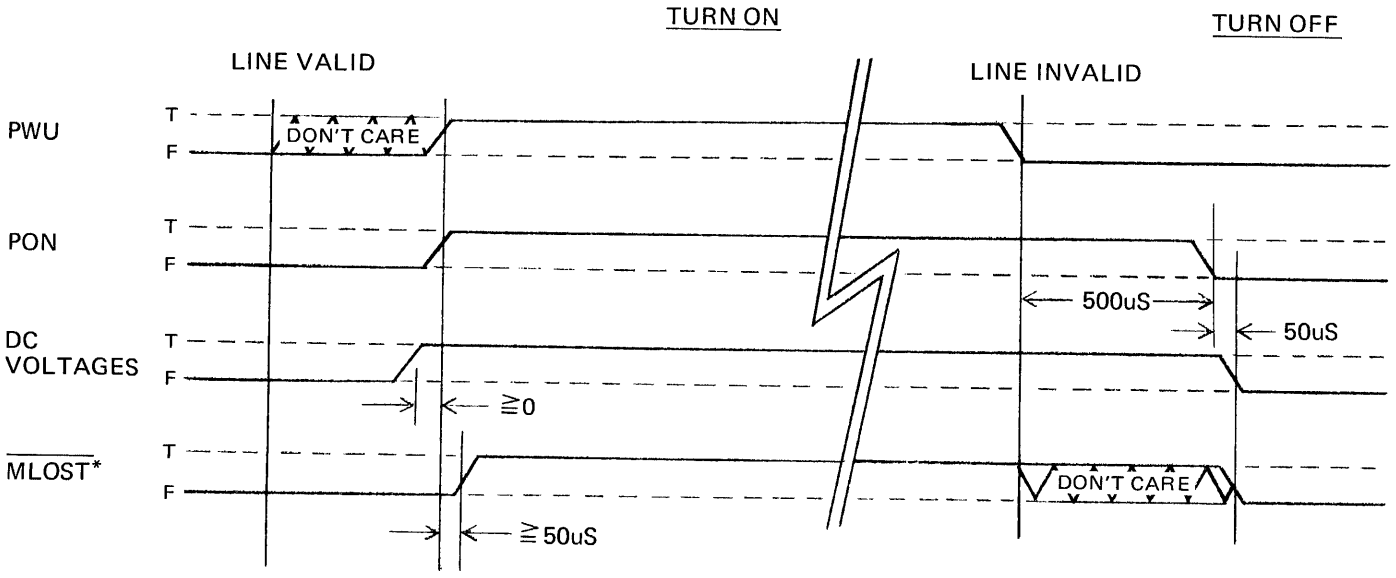
The sequence is shown in Figure III.402.a. PRSTFF holds $\overline{\text{CONT4FF}}$ = Ø



NOTES: 1) ARS≡"S2 SWITCH is in the ARS Position"
2) CONT4FF≡$\overline{\text{CONT4FF}}$ signal = 0
3) All state transitions occur at the end of time T6

POWER FAIL LOGIC STATE DIAGRAM

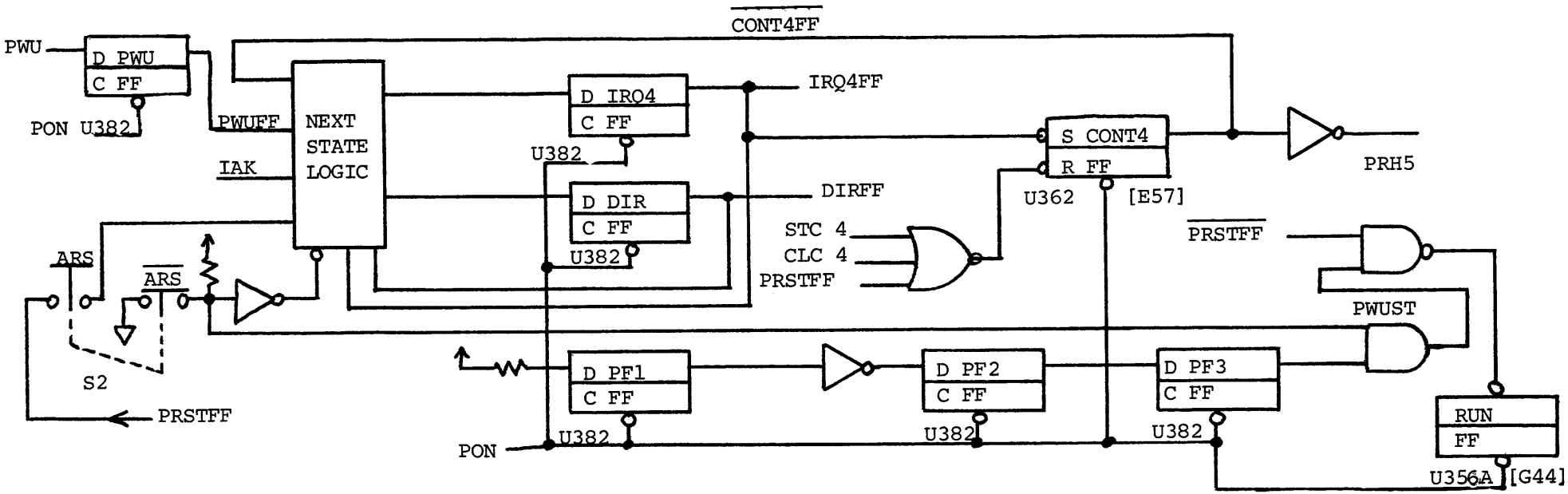
FIGURE III.400.1



* NORMALLY HIGH. GOES INTO LOW WHEN MEMORY IS LOST DUE TO BATTERY DISCHARGE DURING POWER OUTAGE.

POWER SUPPLY TIMING SPECIFICATIONS

FIGURE III.400.2



LOGICAL DIAGRAM - POWER FAIL LOGIC

FIGURE III.400.3

[F56] and prevents setting RUNFF high. Whe IRQ4FF gets set, it is cleared on the next T6 (refer to Figure III.400.1).

402.3 Switch S2 in ARS Position and PRESET ff Cleared

The sequence is shown in Figure III.402b. Note that PWUST is allowed to occur the second T6 after PON goes high, setting RUNFF←1. At the same time IRQ4FF is set. This delay through the PFn ff's prevents setting RUN until an interrupt request is generated. $\overline{\text{CONT4FF}}$ ← 1 when IRQ4FF ← 1, which sets PF high and PRH5 low. This disables the interrupt priority chain and turns on the PF indicator on the front panel. IRQ4FF requests a special interrupt and forms select code 4 on the IA-Bus. IRQ4FF remains high until an IAK is issued by the processor in response to the PF interrupt request. $\overline{\text{CONT4FF}}$ will be high until set low by a STC 4 or CLC 4 I/O instruction, preventing all other interrupts until then. The state of DIR may be tested with a SFC 4 I/O instruction [E59] (see section III.303.2).

403 NORMAL STATE

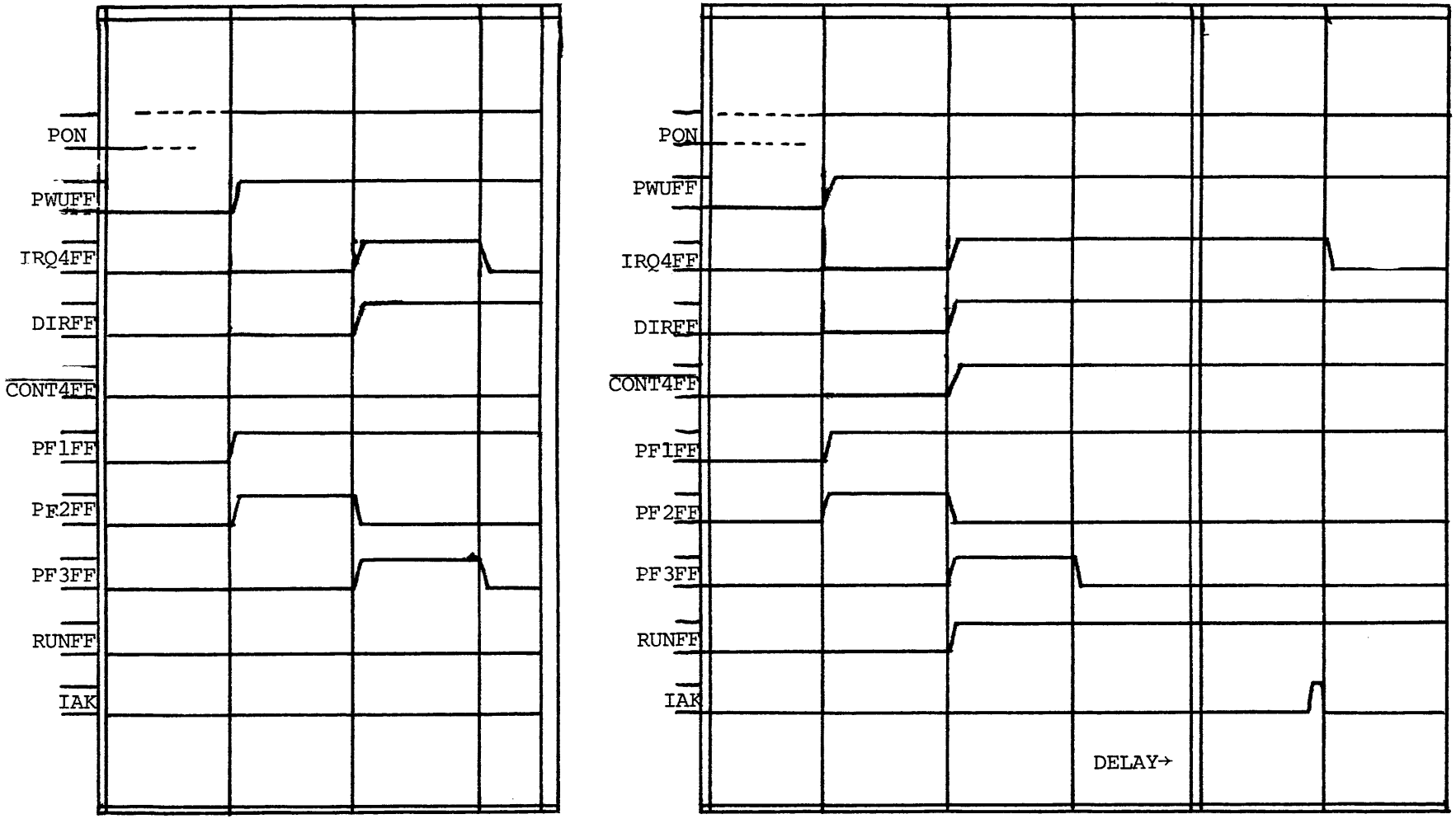
If the Switch S2 is ever put into the $\overline{\text{ARS}}$ position, IRQ4FF and DIRFF will both be set to 0 at the next T6. The PF1FF, PF2FF, PF3FF create a 5T-period pulse on PF3FF after the second T6 following PON going high. Their state after that is PF1FF=1, PF2FF=PF3FF=Ø until PON goes low again. After the PF interrupt request is granted, IRQ4FF=0 and DIRFF=1 as long as power is up. A power failure is signaled by PWUFF going low.

404 POWER-DOWN

There are three possible response sequences to PWU going low. No response will occur until PON goes low unless $\overline{\text{CONT4FF}}$ has been set high.

404.1 Switch S2 in $\overline{\text{ARS}}$ Position

IRQ4FF and DIRFF = Ø (see section III.403) and no warning is given to the processor prior to PON going low.



a) ARS and PRESET

b) ARS without PRESET

NOTES: (1) each single vertical line represents the end of a T6 period.

TIMING DIAGRAMS - POWER UP SEQUENCE

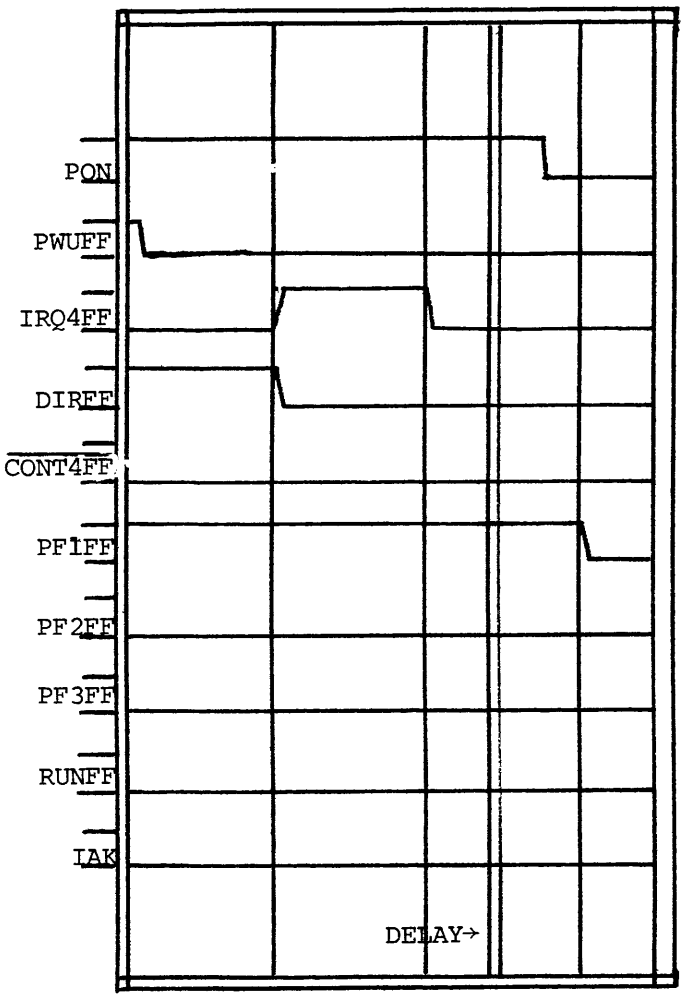
FIGURE III.402.

404.2 Switch S2 in ARS Position, PRESET ff Set

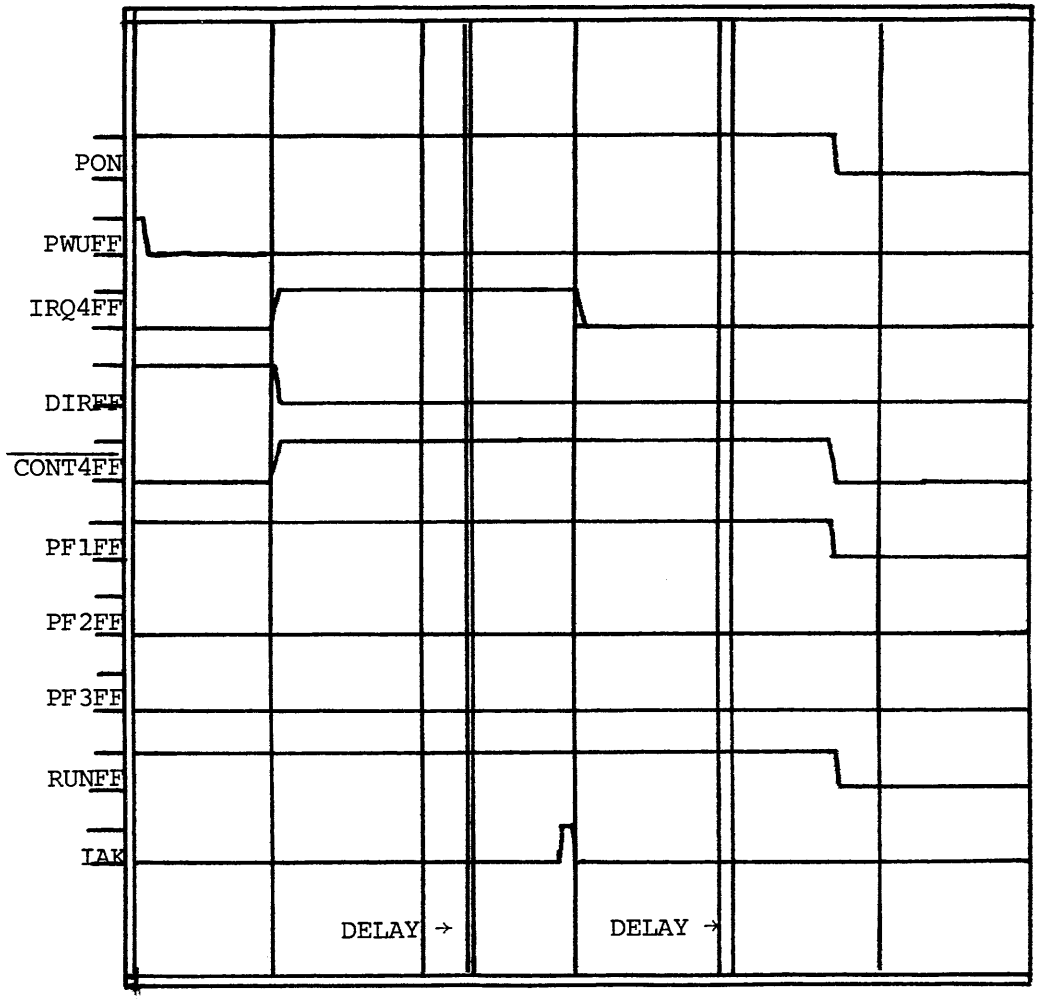
See Figure III.404a for timing diagram. IRQ4FF is set low the T6 after it is set high. Since the machine must be in the HALT mode anyway to set PRESET, this sequence is of no significant importance; no machine instructions will be fetched anyway. PRESET holds $\overline{\text{CONT4FF}}$=0 so that the interrupt priority chain is not disabled.

404.3 Switch S2 in ARS Position, PRESET not ff Cleared

See Figure III.404b for timing diagram. When IRQ4FF goes high, $\overline{\text{CONT4FF}} \leftarrow 1$, disabling all other interrupts. IRQ4FF also requests a special interrupt and forms select code 4 on the IA-Bus. IRQ4FF remains high until IAK is issued by the CPU. When PON goes low, initial conditions are re-established (section III.401).



a) ARS with PRESET (HALT mode)

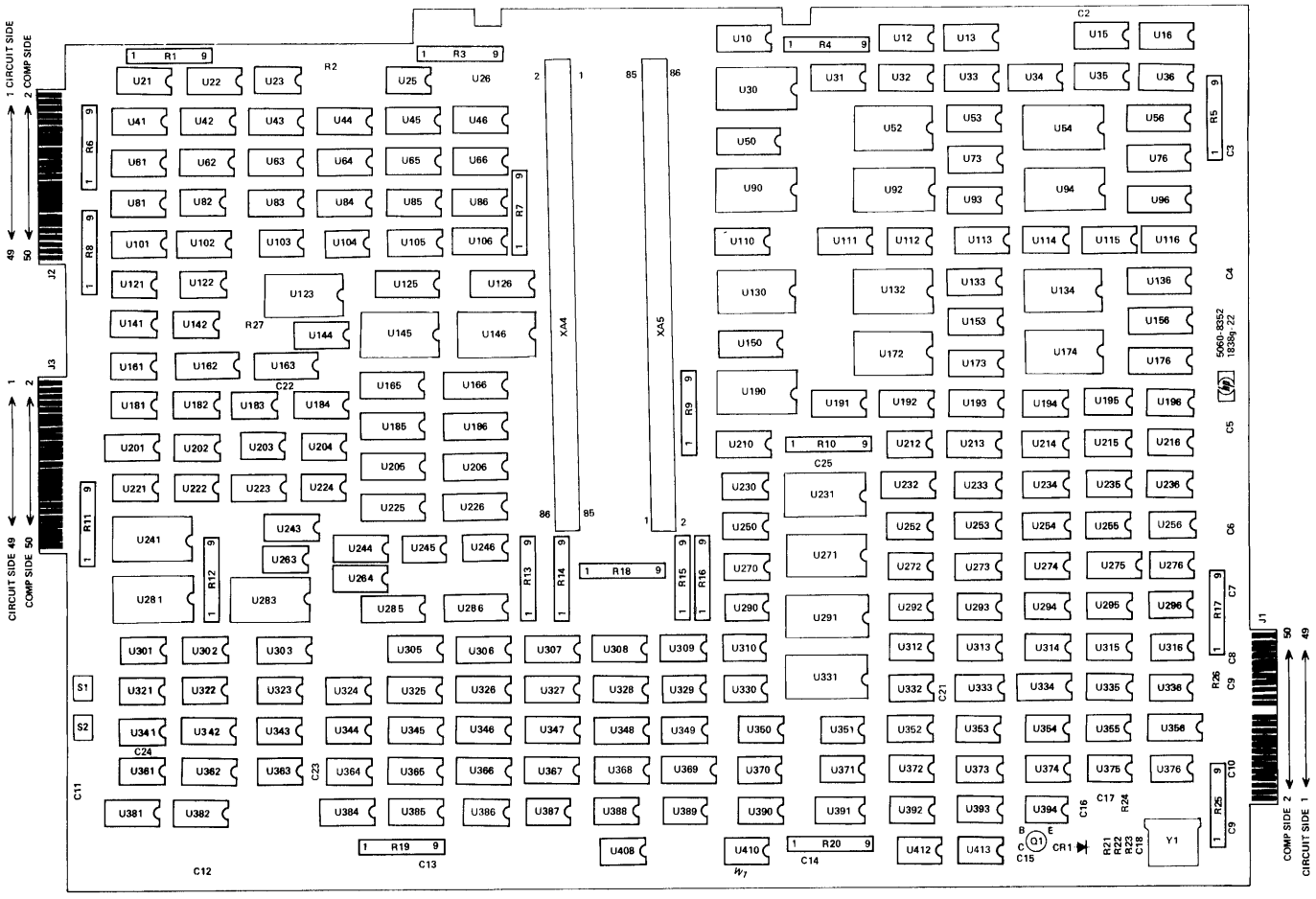


b) ARS without PRESET (RUN Mode)

NOTE: Each single vertical line represents the end of a T6 period.

TIMING DIAGRAMS - POWER-DOWN SEQUENCE

Figure III.404



CPU Assy.
5060-8352

ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER	UM
00	C21	CAP 680PF 5%		0140-0208		U	1	
01	C1-6,11-15	CAP 0.1UF		0150-0121		U	11	
01	C7-10,16,19,23-25	CAP .01UF		0160-2055		U	9	
00	C22	CAP 300PF 5%		0160-2207		U	1	
00	R26	RES 100 5% .25		0683-1015		D	1	
01	R27	RES 1K 1%.125		0757-0280		D	1	
00	R2	RES 511 1%.125		0757-0416		D	1	
01	U13,15,16	SOCKET 16 DIP LO		1200-0482		U	3	
00	J4,5	CONN PC2X43.156D		1251-4139		U	2	
01 03	R1,5-9,12,17,19,20,25	RES NET 8X1K		1810-0121		U	11	
01 03	R3,4,10,11,13-16 18	RES NET 8X500		1810-0132		U	9	
	U413	XTAL OSC 18.5		1813-0135		U	1	
01	U105,106	IC HPROM-1024-5		1816-0015		3	2	
01 01	U106 MADE	I.C. ROM 4X256 FROM 1816-0015		1816-0392		3	1	
01 01	U105 MADE	I.C. ROM 4X256 FROM 1816-0015		1816-0393		3	1	
		I.C. ROM 4X256		1816-0420 PART NO CONT		3	1	

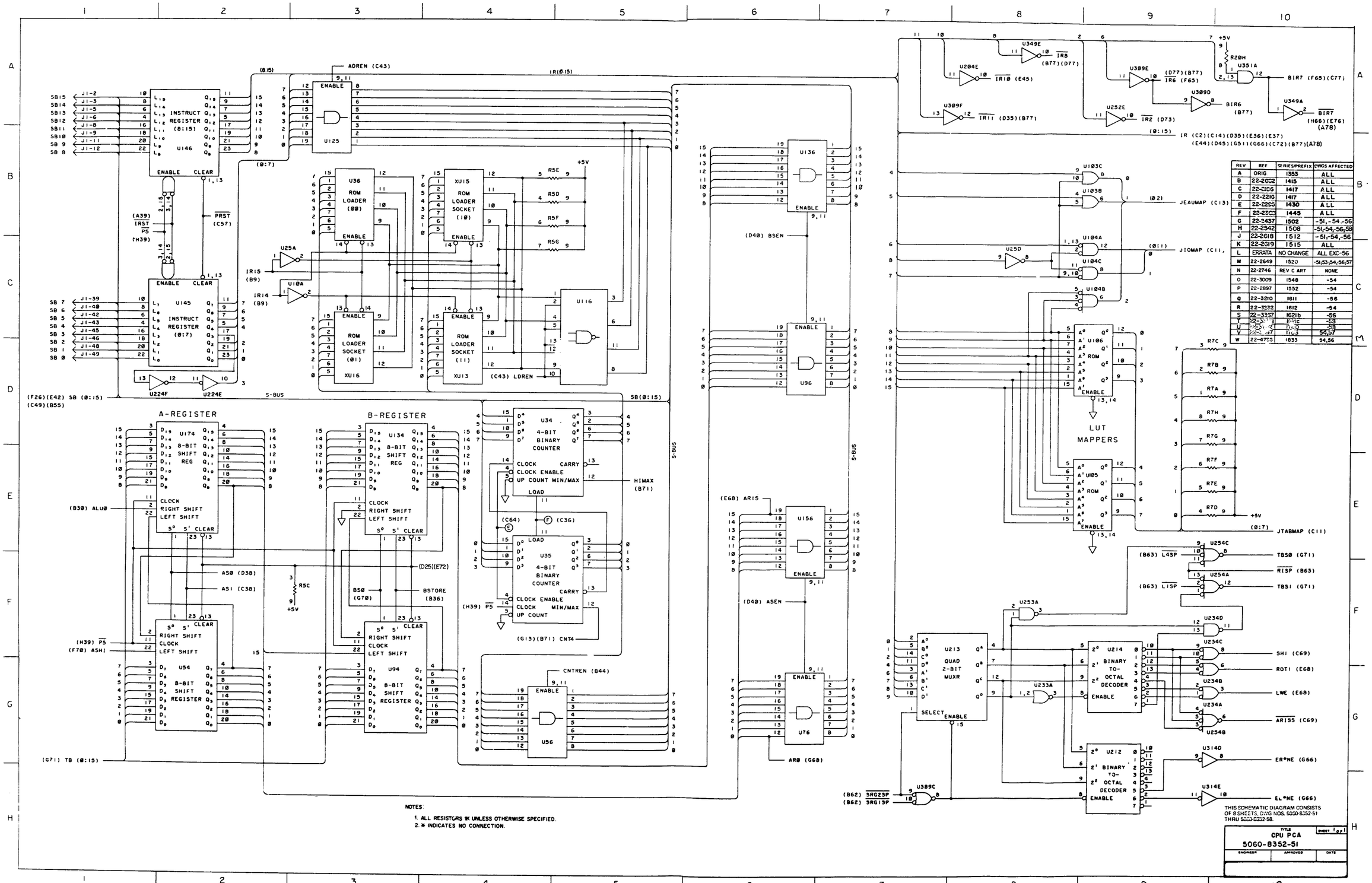
ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER	UM
		PART NO CONT		1816-0420				
01	U36							
01	MADE	FROM 1816-0782						
		IC PROM 4X256		1816-0782		3	1	
01	U36							
		IC SN7400N		1820-0054		U	7	
01	U234,253,256,315,							
03	322,350,335							
		IC SN7410N		1820-0068		U	3	
01	U295,329,355							
		IC SN7420N		1820-0069		U	1	
01	U321							
		IC SN7474N		1820-0077		U	2	
01	U263,343							
		IC MC3001P		1820-0141		U	2	
01	U122,236							
		IC SN7404N		1820-0174		U	7	
01	U10,25,275,309,349,							
03	364,376							
		L/T BUY RES. USE		1820-0205		U	4	
01	U103,292,313,324							
		L/T BUY RES. USE		1820-0239		U	3	
01	U233,341,408							
		IC SN74H00N		1820-0370		U	7	
01	U121,216,250,272,							
03	294,350,392							
		IC SN74H10N		1820-0371		U	2	
01	U333,344							
		IC SN74H11N		1820-0372		U	5	
01	U182,202,254,274,387							
		IC SN74H21N		1820-0374		U	2	
01	U63,388							
		IC SN74H52N		1820-0379		U	9	
01	U194,195,235,301,							
03	352,354,372,374,							
05	375							
		IC SN74H61N		1820-0384		U	3	
01	U215,353,373							
		IC SN74H04N		1820-0424		U	6	
				PART NO CONT				

ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER	UM
		PART NO CONT		1820-0424				
01	U183,204,221,255,							
03	314,323							
		IC SN74H102N		1820-0469		U	1	
01	U390							
		IC SN74154N		1820-0495		U	5	
01	U123,241,281,283,331							
		IC SN7408N		1820-0511		U	3	
01	U302,316,361							
		IC SN74H74N		1820-0512		U	1	
01	U293							
		IC SN7423N		1820-0538		U	3	
01	U115,144,342							
		IC SN7437N		1820-0539		U	1	
01	U386							
		IC SN74191N		1820-0545		U	3	
01	U22,42,62							
		IC SN74H01N		1820-0605		U	1	
01	U276							
		IC SN74181N		1820-0606		U	4	
01	U52,92,132,172							
		IC MC4006P		1820-0608		U	3	
01	U142,212,214							
		IC MC8309P		1820-0610		U	8	
01	U33,53,73,93,133,							
03	153,173,193							
		IC SN74182N		1820-0611		U	1	
01	U111							
		IC SN74H05N		1820-0613		U	6	
01	U23,82,141,161,181,							
03	290							
		IC 93L08DC		1820-0614		U	2	
01	U90,130							
		IC 9322PC		1820-0616		U	1	
01	U110							
		IC SN7438N		1820-0621		U	1	
01	U116							
		IC SN74151N		1820-0622		U	13	
				PART NO CONT				

ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER	UM
		PART NO CONT		1820-0622				
01	U43-46,63-66,83-86,							
03	232							
		IC 9314PC		1820-0626		U	1	
01	U362							
		IC SN74150N		1820-0640		U	2	
01	U271,291							
		IC SN7432N		1820-0661		U	4	
01	U196,230,296,312							
		IC SN74S00N		1820-0681		U	1	
01	U222							
		IC SN74S04N		1820-0683		U	4	
01	U224,252,310,363							
		IC SN74S10N		1820-0685		U	1	
01	U332							
		IC SN74S11N		1820-0686		U	3	
01	U343,351,410							
		IC SN74S20N		1820-0688		U	1	
01	U270							
		IC SN74S40N		1820-0690		U	2	
01	U370,371							
		IC SN74S74N		1820-0693		U	4	
01	U243,245,246,336							
		IC SN74S86N		1820-0694		U	1	
01	U273							
		IC SN74H106N		1820-0715		U	6	
01	U201,334,356,369,							
03	368,391							
		IC 9308PC		1820-0742		U	5	
01	U30,145,146,190,231							
		IC HP147A		1820-0755		U	18	
01	U56,76,96,125,126,							
03	136,156,163,165,166							
05	176,185,186,205,206							
07	226,285,286							
		IC HP147B		1820-0756		U	4	
01	U50,150,162,225							
		IC SN7427N		1820-0782		U	1	
				PART NO CONT				

M-Series CPU Assembly Parts LIst (6050-8352) Sht. 5 of 5

ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER	UM
		PART NO CONT		1820-0782				
01	U114							
		IC SN74174N		1820-0788		U	3	
01	U21,41,382							
		IC SN7489N		1820-1028		U	4	
01	U31,32,191,192							
		IC SN74198N		1820-1032		U	4	
01	U54,94,134,174							
		IC SN74S174N		1820-1076		U	7	
01	U61,81,101,102,223,							
03	244,264							
		IC SN74S157N		1820-1077		U	1	
01	U213							
		IC 8113B		1820-1080		U	20	
01	U184,210,303,305-308							
03	325-328,345-348							
05	365-367,384,385							
		IC SN74S133N		1820-1130		U	1	
01	U113							
		IC SN74LS27N		1820-1206		U	1	
01	U104							
		IC SN74LS153N		1820-1244		U	1	
01	U381							
		IC 74LS191		1820-1278		U	2	
01	U34,35							
		IC SN74S08N		1820-1367		U	3	
01	U112,389,412							
		SW TGL 2POS Z PC		3101-1213		U	2	
00	S1,2							
		LABEL-USA		7120-0830		L	1	
		WIRE JUMPERS		8159-0005		D	2	
00	W1,2							
		DIP INSERTION		5061-3469		A	1	
		BOARD-ETCHED		5080-9741		1	1	
		IC 21MXE DSC LDR		5081-2361		3	1	
01	U15							
		CPU MRS TST STAT		ET13425		1	0	
				PART NO CONT				



REV	REF	SERIES/PREFIX	CYNGS AFFECTED
A	ORIG	1353	ALL
B	22-2002	1415	ALL
C	22-2106	1417	ALL
D	22-2216	1417	ALL
E	22-2256	1430	ALL
F	22-2303	1445	ALL
G	22-2437	1502	-51,-54,-56
H	22-2542	1508	-51,-54,-56,58
J	22-2618	1512	-51,-54,-56
K	22-2619	1515	ALL
L	ERRATA	NO CHANGE	ALL EXC-56
M	22-2649	1520	-51,53,54,56,57
N	22-2746	REV C ART	NONE
O	22-3009	1548	-54
P	22-2897	1552	-54
Q	22-3210	1611	-56
R	22-3332	1612	-54
S	22-3357	1621b	-55
T	22-3[illegible]	[illegible]	-53
U	22-3[illegible]	[illegible]	-58
V	22-3[illegible]	[illegible]	54,57
W	22-4736	1633	54,56

NOTES:
1. ALL RESISTORS 1K UNLESS OTHERWISE SPECIFIED.
2. * INDICATES NO CONNECTION.

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58.

IC -141/ -142

REV	REFERENCE
A	ORIG.
B	22-2082
C	22-2186
D	22-2216
E	22-2296
F	22-2388
K	22-2619
L	ERRATA
N	22-2746

SAVE REGISTER

INTERRUPT JUMP LOGIC

REPEAT LOGIC

ROM ADDRESS REGISTER

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE	SHEET 1 of 7
CPU PCA 5060-8352-52	
ENGINEER / APPROVED / DATE	
HEWLETT PACKARD CO DATA SYSTEMS DEVELOPMENT DIVISION	

REV	REFERENCE
A	ORIG.
B	22-2082
C	22-2186
D	22-2216
E	22-2296
F	22-2388
K	22-2619
L	ERRATA
M	22-2649
N	22-2746

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE: CPU PCA
5060-8352-53
SHEET 1 OF 1
ENGINEER | APPROVED | DATE
HEWLETT PACKARD CO
DATA SYSTEMS DEVELOPMENT DIVISION

REV	REFERENCE
A	ORIG
B	22-2082
C	22-2188
D	22-2216
E	22-2296
F	22-2388
G	22-2437
H	22-2542
J	22-2618
K	22-2819
L	ERRATA
M	22-2649
N	22-2746
O	22-3009
P	22-2897
R	22-3232
V	22-3947
W	22-4736
Y	22-5363

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE: CPU PCA
5030-8352-54

ENGINEER | APPROVED | DATE

SHEET 1 OF 1

IC -147/ -148

S BUS FIELD DECODER

U123 BINARY TO HEXADEC DECODER

M-REGISTER

U244 6-BIT REGISTER

U264 6-BIT REGISTER

REV	REFERENCE
A	ORIG.
B	22-2082
C	22-2188
D	22-2216
E	22-2296
F	22-2388
K	22-6919
L	ERRATA
N	22-2746

ALL RESISTORS ARE 500 OHMS

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE	SHEET 1 OF 8
CPU PCA	
5060-8352-55	
ENGINEER	APPROVED / DATE
HEWLETT PACKARD CO DATA SYSTEMS DEVELOPMENT DIVISION	

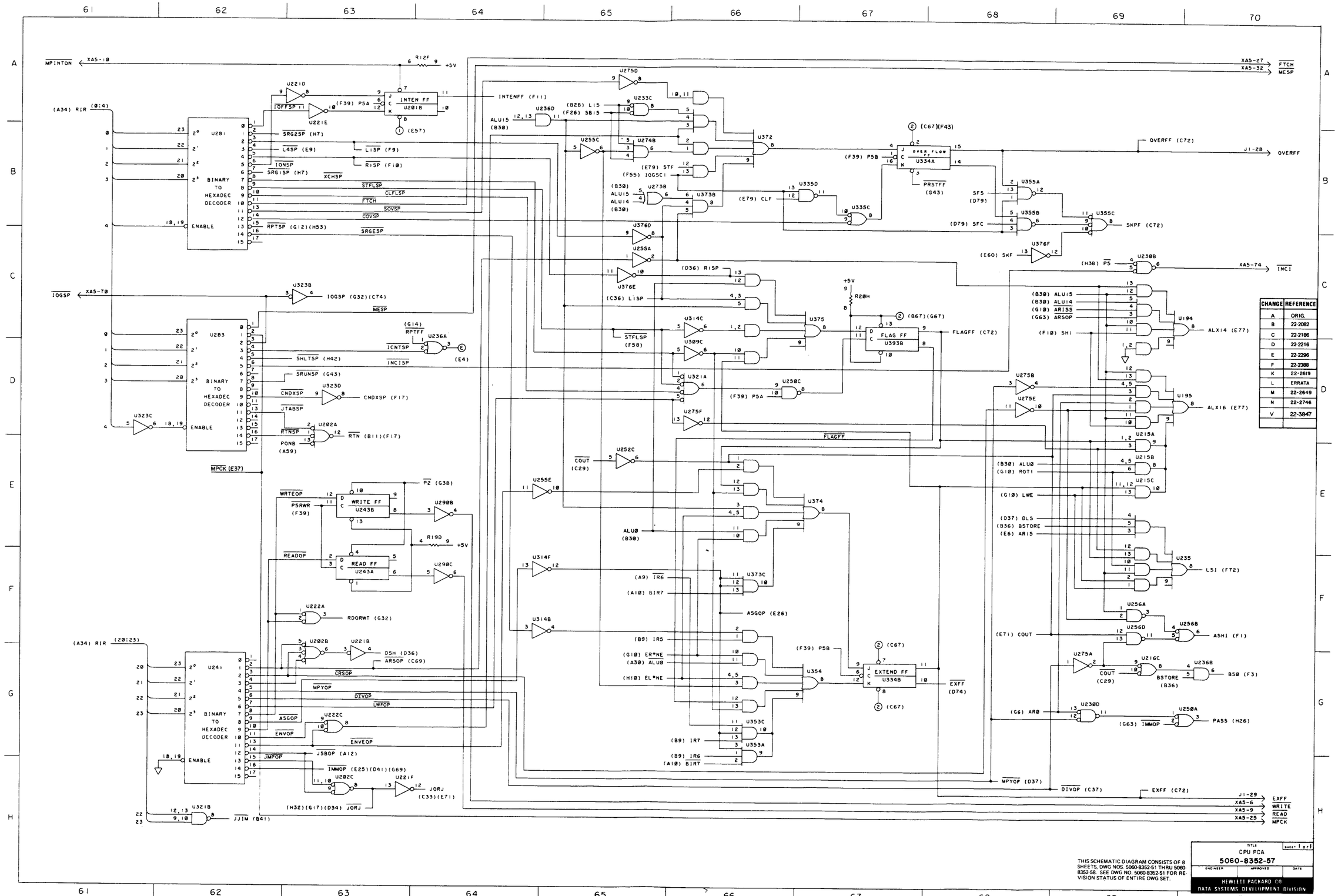
CHANGE	REFERENCE
A	ORIG.
B	22-2087
C	22-2186
D	22-2216
E	22-2296
F	22-2388
G	22-2437
H	22-2542
J	22-2618
K	22-2619
L	ERRATA
M	22-2649
N	22-2746
Q	22-3210
S	22-3367
T	22-4736

CENTRAL INTERRUPT REGISTER

POWER FAIL REGISTER

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

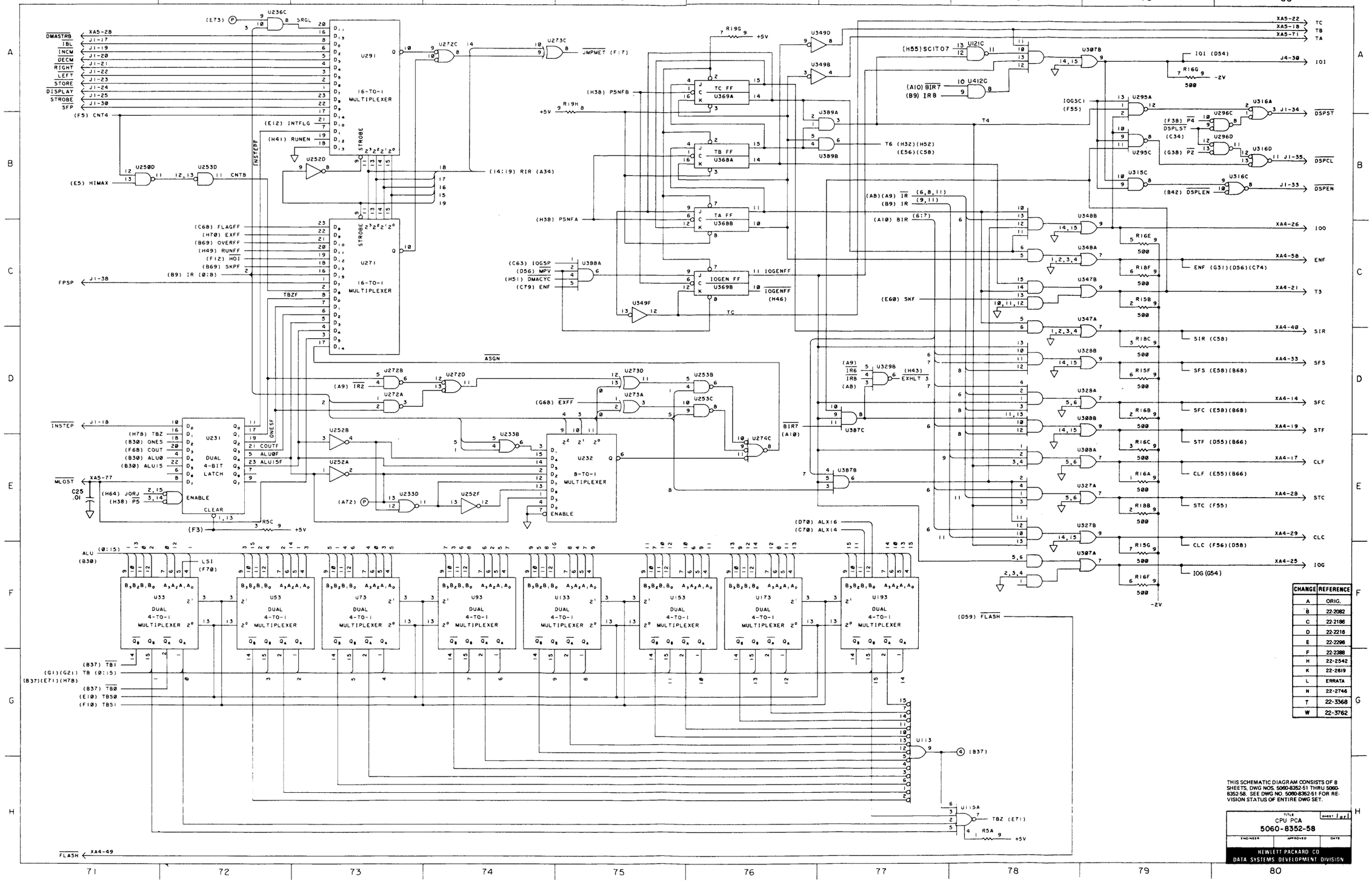
TITLE	SHEET 1 of 1
CPU PCA	
5060-8352-56	
ENGINEER	APPROVED / DATE



CHANGE	REFERENCE
A	ORIG.
B	22-2082
C	22-2186
D	22-2216
E	22-2296
F	22-2388
K	22-2619
L	ERRATA
M	22-2649
N	22-2746
V	22-3847

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE: CPU PCA
SHEET 1 OF 8
5060-8352-57
ENGINEER | APPROVED | DATE
HEWLETT PACKARD CO
DATA SYSTEMS DEVELOPMENT DIVISION



CHANGE	REFERENCE
A	ORIG.
B	22-2082
C	22-2186
D	22-2216
E	22-2298
F	22-2388
H	22-2542
K	22-2619
L	ERRATA
N	22-2746
T	22-3368
W	22-3762

THIS SCHEMATIC DIAGRAM CONSISTS OF 8 SHEETS, DWG NOS. 5060-8352-51 THRU 5060-8352-58. SEE DWG NO. 5060-8352-51 FOR REVISION STATUS OF ENTIRE DWG SET.

TITLE	SHEET 1 of 1
CPU PCA	
5060-8352-58	
ENGINEER / APPROVED / DATE	
HEWLETT PACKARD CO DATA SYSTEMS DEVELOPMENT DIVISION	

BASIC INSTRUCTION SET MICROPROGRAM LISTING

APPENDIX A

This appendix provides a complete microassembly listing of the base instruction set microprogram for the HP 21MX M-Series Computer.

```
0001                    MICMX,L
0002                          ORG                   0
0003                    ********************************************************************************
0004                    *
0005                    *       21MX MICRO-CODE
0006                    *       MODULE 0
0007                    *
0008                    *       1976-04-08-1500
0009                    ********************************************************************************
0010                    FADD      EQU                 %7125
0011                    FSUB      EQU                 %7126
0012                    FMPY      EQU                 %7221
0013                    FDIV      EQU                 %7262
0014                    IFIX      EQU                 %7000
0015                    FLOAT     EQU                 %7025
0016                    ********************************************************************************
0017                    *       FETCH ROUTINE
0018                    ********************************************************************************
0019 00000  220 074712  FETCH     READ FTCH INC  PNM  P          M<=P; P<=P+1; READ NEW INSTR.
0020 00001  017 136745            ION                            ENABLE INTERRUPT RECOGNITION
0021 00002  017 100411            CLFL PASS IR   TAB             IR<= T/A/B; CLR FLAG FF
0022 00003  220 020673            READ JTAB INC  CM   ADR        JUMP THRU TABLE; LOAD M IF MRG IN
0023                    ********************************************************************************
0024 00004  325 120031  HORI      JMP  CNDX RUN  RJS  HALT       RUN MODE IMPLIES AN INTERUPT
0025                    ********************************************************************************
0026                    *       INTERRUPT RESPONSE ROUTINE
0027                    ********************************************************************************
0028 00005  237 106451  INTERUPT READ CLFL PASS M    CIR        M<=CIR; READ TRAP CELL; CLR FLAG
0029 00006  320 000471            JMP  CNDX TBZ  RJS  INTOK      CHECK IF CIR IS VALID
0030 00007  237 106457            READ       PASS M    CIR       M<=CIR; READ TRAP CELL
0031 00010  320 040031            JMP  CNDX TBZ       FETCH      IF NO INT BY NOW, IGNORE
0032 00011  017 104400  INTOK     IOFF PASS IR   T              IR<= TRAP CELL, DISABLE INT RECOG
0033 00012  220 020673            READ JTAB INC  CM   ADR        JUMP THRU TABLE; LOAD M IF MRG IN
0034                    ********************************************************************************
0035                    *       INDIRECT ROUTINE
0036                    ********************************************************************************
0037 00013  220 022457  INDLEVEL READ      INC  M    M          READ NEXT LEVEL
0038 00014  326 001031            JMP  CNDX NHOI RJS  IND2       HALT OR INTERRUPT?
0039 00015  017 100465  INDIRECT      INCI PASS M    TAB        M<=T/A/B; INCR INDIRECT COUNT
0040 00016  322 040571            JMP  CNDX AL15      INDLEVEL   CHECK FOR ANOTHER LEVEL OF INDIRE
0041 00017  220 022476            READ RTN  INC  M    M          READ EFFECTIVE ADDRESS, RETURN
0042 00020  017 100465  IND2           INCI PASS M    TAB        M<=T/A/B; INCR INDIRECT COUNT
0043 00021  330 100731            JMP  CNDX NSNG RJS  INDIRECT+1 JUMP BACK FOR SINGLE INSTRUCTION
0044 00022  007 175717                 DEC  P    P    P          RESET P
0045 00023  320 000230            JMP                 HORI       HALT OR INTERUPT


0047                    ********************************************************************************
0048                    *       ALTER-SKIP GROUP
0049                    ********************************************************************************
0050 00024  017 102757  ASGNOP          PASS       CAB        SET UP SKIP TEST
0051 00025  327 042431            JMP  CNDX ASGN      ASGNSKP    JUMP IF ASG SKIP NOT MET
0052 00026  200 075717            ASG       INC  P    P          P<=P+1; ENABLE ASG HARDWARE
0053 00027  327 100031            JMP  CNDX IR2  RJS  FETCH      DONE IF NOT INA/B
0054 00030  260 002076            ENVE RTN  INC  CAB  CAB        A/B <= A/B PLUS 1
0055                    *
0056 00031  001 136057  ASGCL*          ZERO CAB              CLEAR  A/B REGISTER
0057 00032  327 042431            JMP  CNDX ASGN      ASGNSKP    JUMP IF ASG SKIP NOT MET
0058 00033  200 075717            ASG       INC  P    P          P<=P+1; ENABLE ASG HARDWARE
0059 00034  327 100031            JMP  CNDX IR2  RJS  FETCH      DONE IF NOT INA/B
0060 00035  260 002076            ENVE RTN  INC  CAB  CAB        A/B <= A/B PLUS 1
0061                    *
0062 00036  010 002057  ASGCM*          CMPS CAB  CAB        A/B <= NOT A/B
0063 00037  327 042431            JMP  CNDX ASGN      ASGNSKP    JUMP IF ASG SKIP NOT MET
0064 00040  200 075717            ASG       INC  P    P          P<=P+1; ENABLE ASG HARDWARE
0065 00041  327 100031            JMP  CNDX IR2  RJS  FETCH      DONE IF NOT INA/B
0066 00042  260 002076            ENVE RTN  INC  CAB  CAB        A/B <= A/B PLUS 1
0067                    *
0068 00043  016 036057  ASGCC*          ONE  CAB              CLR & COMP A/B REGISTER
0069 00044  327 042431            JMP  CNDX ASGN      ASGNSKP    JUMP IF ASG SKIP NOT MET
0070 00045  200 075717            ASG       INC  P    P          P<=P+1; ENABLE ASG HARDWARE
0071 00046  327 100031            JMP  CNDX IR2  RJS  FETCH      DONE IF NOT INA/B
0072 00047  260 002076            ENVE RTN  INC  CAB  CAB        A/B <= A/B PLUS 1
0073                    *
0074 00050  217 136757  ASGNSKP  ASG                            NO SKIP; ENABLE ASG HARDWARE
0075 00051  327 100031            JMP  CNDX IR2  RJS  FETCH      DONE IF NOT INA/B
0076 00052  260 002076            ENVE RTN  INC  CAB  CAB        A/B <= A/B PLUS 1
0077                    ********************************************************************************
0078                    *       SHIFT/ROTATE GROUP
0079                    ********************************************************************************
0080 00053  017 102046  SRG             SRG1 PASS CAB  CAB        FIRST SHIFT
```

```
0081 00054  017 102056                SRGE PASS CAB  CAB       CHECK FOR CLEAR E; SET SLA TEST
0082 00055  335 103031          JMP   CNDX SRGL RJS  *+3       SRGL IS SLA TEST
0083 00056  017 102041                SRG2 PASS CAB  CAB       SECOND SHIFT
0084 00057  000 075736                RTN  INC  P    P         P<=P+1, WHEN LSB = 0
0085 00060  017 102041                SRG2 PASS CAB  CAB       SECOND SHIFT
0086 00061  017 136776  RETURN        RTN

0088                    ********************************************************************************
0089                    *              I/O GROUP
0090                    ********************************************************************************
0091 00062  017 136757  IOCNTRL       NOP                      ALLOW TIME TO GET SKIP FLAG
0092 00063  326 100031          JMP   CNDX SKPF RJS  FETCH     CHECK SKIP FLAG
0093 00064  000 075736                RTN  INC  P    P         P <= P + 1
0094 00065  017 136757                NOP
0095                    *
0096 00066  017 102757  IO.OT*              PASS      CAB       SET UP S-BUS
0097 00067  017 102217                     PASS IOO  CAB       I/O-BUS <= A/B
0098 00070  017 102236                RTN  PASS IOO  CAB       HOLD I/O-BUS VALID
0099 00071  017 136757                NOP
0100                    *
0101 00072  017 136757  IO.LI*        NOP                      SYNCHRONIZE IOI PULSE
0102 00073  017 136757                NOP
0103 00074  017 110076                RTN  PASS CAB  IOI       A/B <= I/O-BUS
0104 00075  017 136757                NOP
0105                    *
0106 00076  017 136757  IO.MI*        NOP                      SYNCHRONIZE IOI PULSE
0107 00077  017 102157                     PASS L    CAB       L <= A/B FOR ALU OPERATION
0108 00100  017 010076                RTN  IOR  CAB  IOI       A/B<= (A/B) + (I/O BUS)
0109                    *
0110                    ********************************************************************************
0111                    *       IO GROUP/ EAU GROUP/ MAC GROUP JUMPS
0112                    ********************************************************************************
0113 00101  320 003122  IOG       JMP  IOG             IOCNTRL
0114 00102  320 015437  EAU       JMP  JEAU            EAUTABLE
0115 00103  320 016034  MAC0      JMP  J74             MACTABL0
0116 00104  320 017034  MAC1      JMP  J74             MACTABL1

0118                    ********************************************************************************
0119                    *       MEMORY REFERENCE GROUP
0120                    ********************************************************************************
0121 00105  300 000670  AND,I      JSB                      INDIRECT
0122 00106  017 126157  AND                PASS L    A       L <= A
0123 00107  015 100576                RTN  AND  A    TAB     A <= T/A/B AND L
0124                    *
0125 00110  300 000670  CP*,I      JSB                      INDIRECT
0126 00111  017 102157  CP*                PASS L    CAB     L <= A/B
0127 00112  013 000757                     XOR       TAB     T-BUS <= T/A/B XOR L
0128 00113  320 040031          JMP   CNDX TBZ       FETCH   JUMP TO FETCH IF EQUAL
0129 00114  000 075736                RTN  INC  P    P       P<= P+1 IF NOT EQUAL
0130                    *
0131 00115  300 000670  XOR,I      JSB                      INDIRECT
0132 00116  017 126157  XOR                PASS L    A       L <= A
0133 00117  013 000576                RTN  XOR  A    TAB     A <= T/A/B XOR L
0134                    *
0135 00120  300 000670  IOR,I      JSB                      INDIRECT
0136 00121  017 126157  IOR                PASS L    A
0137 00122  017 000576                RTN  IOR  A    TAB     A <= T/A/B IOR L
0138                    *
0139 00123  300 000670  ST*,I      JSB                      INDIRECT
0140 00124  017 122761  ST*           MPCK PASS         M       MEM PROTECT CHECK OF ADDRESS
0141 00125  177 102036          WRTE RTN  PASS TAB  CAB       T/A/B <= A/B; WRITE
0142                    *
0143                    *
0144 00126  300 000670  AD*,I      JSB                      INDIRECT
0145 00127  017 102157  AD*                PASS L    CAB     L <= A/B
0146 00130  264 100076          ENVE RTN  ADD  CAB  TAB     A/B <= I/A/B PLUS L
0147                    *
0148 00131  300 000640  JSB,I      JSB  IOFF            INDIRECT  DISABLE INTERRUPT RECOGNITION
0149 00132  017 122761  JSB           MPCK PASS         M       MEM PROTECT CHECKS THIS ADDR
0150 00133  177 174017          WRTE      PASS TAB  P       T/A/B <= RETURN ADDRESS; WRITE
0151 00134  000 023736                RTN  INC  P    M       P <= M + 1
0152                    *
0153 00135  300 000670  ISZ,I      JSB                      INDIRECT
0154 00136  017 122761  ISZ           MPCK PASS         M       MEM PROTECT CHECKS THIS ADDR
0155 00137  000 001017                     INC  S1   TAB     S1 <= T/A/B + 1
0156 00140  177 140017          WRTE      PASS TAB  S1      T <= S1; WRITE
0157 00141  320 000031          JMP   CNDX TBZ  RJS  FETCH   ZERO?  NO, DONE.
0158 00142  000 075736                RTN  INC  P    P       YES, P <= P+1
```

```
0159                        *
0160 00143   300 000670   LD*,1    JSB                  INDIRECT
0161 00144   017 100076   LD*           RTN  PASS CAB  TAB        A/B <= T/A/B


0163 00145   017 136765   JMP,I         INCI                              COUNT ONE INDIRECT LEVEL
0164 00146   017 101000                 IOFF PASS S1    TAB               DISABLE INT RECOGNITION;S1<=T/A/B
0165 00147   322 046531          JMP    CNDX AL15       JINDL             JMP IF ANOTHER LEVEL OF INDIRECT
0166 00150   017 140761                 MPCK PASS       S1                MEM PROT CHECKS DESTINATION ADDR
0167 00151   017 141736                 RTN  PASS P     S1                P <= DESTINATION ADDR
0168                        *
0169 00152   220 040457   JINDL  READ        INC  M     S1                READ NEXT LEVEL
0170 00153   326 007031          JMP    CNDX NHOI RJS   HORICK           JMP IF HALT OR INT
0171 00154   017 101025                 INCI PASS S1    TAB               S1 <= T/A/B; COUNT INDIRECT LEVEL
0172 00155   322 046531          JMP    CNDX AL15       JINDL             JMP IF ANOTHER LEVEL OF INDIRECT
0173 00156   017 140761                 MPCK PASS       S1                MEM PROT CHECKS DESTINATION ADDR
0174 00157   017 141736                 RTN  PASS P     S1                P <= DESTINATION ADDR
0175                        *
0176 00160   017 101025   HORICK        INCI PASS S1    TAB               S1 <= T/A/B;  COUNT INDIRECT LEVE
0177 00161   330 106671          JMP    CNDX NSNG RJS   JINDL+3           JUMP BACK FOR SINGLE INSTRUCTION
0178 00162   007 175717                 DEC  P     P                       RESET P
0179 00163   320 000230          JMP                    HURI              HALT OR INTERUPT
0180 00164   017 121021   JMP           MPCK PASS S1    ADR               S1<=DESTINATION ADDR; CHECK WITH
0181 00165   017 141736                 RTN  PASS P     S1                P <= DESTINATION ADDRESS


0183                        ********************************************************************************
0184                        *       EAU MICROPROGRAMS
0185                        ********************************************************************************
0186 00166   010 021017   RRR                CMPS S1    ADR
0187 00167   000 041017                      INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0188 00170   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0189 00171   057 124504          CRS   R1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0190 00172   017 136776                RTN
0191                        *
0192 00173   010 021017   ASR                CMPS S1    ADR
0193 00174   000 041014                COV   INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0194 00175   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0195 00176   037 124504          ARS   R1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0196 00177   017 136776                RTN
0197                        *
0198 00200   010 021017   LSR                CMPS S1    ADR
0199 00201   000 041017                      INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0200 00202   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0201 00203   077 124504          LGS   R1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0202 00204   017 136776                RTN
0203                        *
0204 00205   010 021017   RRL                CMPS S1    ADR
0205 00206   000 041017                      INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0206 00207   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0207 00210   057 124502          CRS   L1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0208 00211   017 136776                RTN
0209                        *
0210 00212   010 021017   ASL                CMPS S1    ADR
0211 00213   000 041014                COV   INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0212 00214   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0213 00215   037 124502          ARS   L1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0214 00216   017 136776                RTN
0215                        *
0216 00217   010 021017   LSL                CMPS S1    ADR
0217 00220   000 041017                      INC  S1    S1                S1 <= TWO'S COMP OF SHIFTS
0218 00221   017 140255                RPT   PASS CNTR  S1                SET UP COUNTER FOR REPEAT
0219 00222   077 124502          LGS   L1    PASS B     B                 DOUBLE-WORD SHIFT REPEAT
0220 00223   017 136776                RTN
0221                        *
0222 00224   220 074457   DLD    READ        INC  M     P                 READ MEMORY ADDRESS
0223 00225   300 000640          JSB   IOFF                INDIRECT          JSB TO GET M<=ADDR OF FIRST WORD
0224 00226   000 023017                      INC  S1    M                 S1 <= ADDRESS OF SECOND WORD
0225 00227   017 100557                      PASS A     TAB               A <= FIRST DATA WORD
0226 00230   220 040457          READ        INC  M     S1                M<=ADDR OF SECOND WORD; READ
0227 00231   000 075717                      INC  P     P                 P <= P + 1
0228 00232   017 100536                RTN   PASS B     TAB               B <= SECOND DATA WORD
0229                        *
0230 00233   220 074457   DST    READ        INC  M     P                 READ MEMORY ADDRESS
0231 00234   300 000640          JSB   IOFF                INDIRECT          JSB TO GET M <= ADDR OF FIRST WOR
0232 00235   000 023021                MPCK  INC  S1    M                 MP CHECK FIRST ADDR; S1<=SECOND A
0233 00236   177 126017          WRTE        PASS TAB   A                 STORE A INTO FIRST LOCATION
0234 00237   000 040461                MPCK  INC  M     S1                MP CHECK S1; M<=S1
0235 00240   177 124017          WRTE        PASS TAB   B                 STORE B INTO SECOND LOCATION
0236 00241   000 075736                RTN   INC  P     P                 UPDATE P
```

```
0238 00242  220 074457  MPY      READ       INC  M     P          M <= P; READ
0239 00243  300 000640           JSB  IOFF                INDIRECT  JSB TO GET M <= ADDR OF OPERAND
0240 00244  000 075717                      INC  P     P          UPDATE P
0241 00245  017 101057                      PASS S2    TAB        S2 <= MULTIPLIER
0242 00246  017 127114  MPYX          COV   PASS S3    A          S3<=A(MULTIPLICAND); CLEAR OVFL
0243 00247  001 136517                      ZERO B                CLEAR B FOR MULTIPLY
0244 00250  017 142157                      PASS L     S2         L <= S2 (MULTIPLIER)
0245 00251  017 124255               RPT   PASS CNTR  B          CLEAR COUNTER; SET REPEAT FF
0246 00252  104 124504           MPY  R1    ADD  B     B          MPY STEP (X16); (B,A)<=A TIMES L
0247 00253  017 144757                      PASS       S3         TEST MULTIPLICAND
0248 00254  322 012731           JMP  CNDX  AL15 RJS   *+2        JUMP IF POSITIVE
0249 00255  003 024517                      SUB  B     B          UNDO LAST MPY STEP IF NEGATIVE
0250 00256  017 142757                      PASS       S2         TEST MULTIPLIER
0251 00257  322 003071           JMP  CNDX  AL15 RJS   RETURN     JUMP IF POSITIVE
0252 00260  017 144157                      PASS L     S3         L <= MULTIPLICAND
0253 00261  003 024536               RTN   SUB  B     B          B<=B MINUS L (CORRECTS FOR NEG. M
0254                     *
0255                     *
0256 00262  220 074457  DIV      READ       INC  M     P          M <= P; READ
0257 00263  300 000640           JSB  IOFF                INDIRECT  JSB TO GET M <= ADDR OF OPERAND
0258 00264  000 075717                      INC  P     P          UPDATE P
0259 00265  010 001157                      CMPS S4    TAB        S4 <= DVSR(CM)::SAVE ORIG SIGN
0260 00266  010 047017                      CMPS S1    S4         S1 <= DVSR
0261 00267  322 013471           JMP  CNDX  AL15 RJS   *+2        JMP IF DVSR NEGATIVE
0262 00270  000 047017                      INC  S1    S4         S1 <= DVSR(2CM)
0263 00271  017 140157                      PASS L     S1         L <= ABS VALUE(DVSR)
0264 00272  010 025117                      CMPS S3    B          S3 <= DVNDHI(2CM)
0265 00273  322 054071           JMP  CNDX  AL15       DIVS       JMP IF DVND POSITIVE
0266 00274  017 144517                      PASS B     S3           IF DVND IS NEGATIVE...
0267 00275  010 027057                      CMPS S2    A            FORM DVND(2CM)
0268 00276  000 042557                      INC  A     S2           IN B,A-REGISTER
0269 00277  321 014071           JMP  CNDX  COUT RJS   DIVS         *
0270 00300  000 044517                      INC  B     S3           *
0271 00301  003 024753  DIVS          SOV   SUB        B          CHECK FOR DVSR TOO SMALL
0272 00302  322 000031           JMP  CNDX  AL15 RJS   FETCH      <=DVND TOO LARGE)
0273 00303  077 124502           LGS  L1    PASS B     B          SHIFT OUT SIGN BIT OF FULL WORD
0274 00304  001 137054                COV   ZERO S2               CLEAR OVFL,S2,& CNTR
0275 00305  017 142255                RPT   PASS CNTR S2         AND SET RPTFF
0276 00306  123 024502           DIV  L1    SUB  B     B          DIV(1ox); A<=QUO(POS); B<=REM*2
0277 00307  157 144142           LWF  L1    PASS L     S3         L <= FLG <= DVND SIGN(CM)
0278 00310  010 027017                      CMPS S1    A          S1 <= QUO(CM)
0279 00311  320 155071           JMP  CNDX  ONES       QZERO      IF QUO=0,THEN NO FURTHER TESTING
0280 00312  013 047057                      XOR  S2    S4         S2(15) <= EXPECTED SIGN OF QUO
0281 00313  322 014671           JMP  CNDX  AL15 RJS   *+2        JMP IF POSITIVE WAS EXPECTED
0282 00314  000 040557                      INC  A     S1           ELSE  A <= QUO(2CM)
0283 00315  017 142157                      PASS L     S2         L(15) <= EXPECTED SIGN OF QUO
0284 00316  013 026757                      XOR        A          COMPARE TO FINAL SIGN OF QUO
0285 00317  322 015071           JMP  CNDX  AL15 RJS   *+2        JMP IF OK
0286 00320  017 136753                SOV                          ELSE INDICATE OVERFLOW
0287 00321  017 124504  QZERO       R1    PASS B     B          B <= (REM*2)/2
0288 00322  324 040031           JMP  CNDX  FLAG       FETCH      CHECK SGN OF DVND
0289 00323  010 024517                      CMPS B     B          IF NEG,THEN FORM 2-COMP OF
0290 00324  000 024536               RTN   INC  B     B          REM & STORE IN B


0292                     ORG                      330B
0293                     ********************************************************************************
0294                     *      EAU TABLE
0295                     ********************************************************************************
0296 00330  320 007330  EAUTABLE JMP               RRR
0297 00331  320 007570           JMP               ASR
0298 00332  320 010030           JMP               LSR
0299 00333  320 000030           JMP               FETCH      ILLEGAL IR CODE FOR EAU GROUP
0300 00334  320 010270           JMP               RRL
0301 00335  320 010530           JMP               ASL
0302 00336  320 010770           JMP               LSL
0303 00337  320 012130           JMP               MPY
0304                     ********************************************************************************
0305                     *      MAC TABLE
0306                     ********************************************************************************
0307 00340  321 145270  MACTABL0 JMP               FADD       / FLOATING POINT
0308 00341  321 145330           JMP               FSUB       / FLOATING POINT
0309 00342  321 151070           JMP               FMPY       / FLOATING POINT
0310 00343  321 153130           JMP               FDIV       / FLOATING POINT
0311 00344  321 140030           JMP               IFIX       / FLOATING POINT
0312 00345  321 141270           JMP               FLOAT      / FLOATING POINT
0313 00346  320 060030           JMP               %1400      *** PROBABLE FUTURE HP USE
0314 00347  320 060035           JMP  J30          %1400      *** PROBABLE FUTURE HP USE
0315 00350  320 100030           JMP               %2000      *** PROBABLE FUTURE HP USE
0316 00351  320 100035           JMP  J30          %2000      *** PROBABLE FUTURE HP USE
0317 00352  320 120030           JMP               %2400      *** PROBABLE FUTURE HP USE
0318 00353  320 120035           JMP  J30          %2400      *** PROBABLE FUTURE HP USE
0319 00354  320 140030           JMP               %3000      *** PROBABLE FUTURE HP USE
0320 00355  320 140035           JMP  J30          %3000      *** PROBABLE FUTURE HP USE
0321 00356  320 160030           JMP               %3400      *** PROBABLE FUTURE HP USE
0322 00357  320 160035           JMP  J30          %3400      *** PROBABLE FUTURE HP USE
0323                     ********************************************************************************
```

```
0324 00360  321 000030  MACTABL1 JMP                 %4000     *** PROBABLE FUTURE HP USE
0325 00361  321 000035           JMP  J30            %4000     *** PROBABLE FUTURE HP USE
0326 00362  321 020030           JMP                 %4400     *** PROBABLE FUTURE HP USE
0327 00363  321 020035           JMP  J30            %4400     *** PROBABLE FUTURE HP USE
0328 00364  321 040030           JMP                 %5000     *** PROBABLE FUTURE HP USE
0329 00365  321 040035           JMP  J30            %5000     *** PROBABLE FUTURE HP USE
0330 00366  321 060030           JMP                 %5400     *** PROBABLE FUTURE HP USE
0331 00367  321 060035           JMP  J30            %5400     *** PROBABLE FUTURE HP USE
0332 00370  321 100030           JMP                 %6000     +++RESERVED FOR CUSTOMER ONLY
0333 00371  321 100035           JMP  J30            %6000     +++RESERVED FOR CUSTOMER ONLY
0334 00372  321 120030           JMP                 %6400     +++RESERVED FOR CUSTOMER ONLY
0335 00373  321 120035           JMP  J30            %6400     +++RESERVED FOR CUSTOMER ONLY
0336 00374  320 040030           JMP                 %1000     / RESERVED FOR HP USE
0337 00375  320 040035           JMP  J30            %1000     / RESERVED FOR HP USE
0338 00376  321 160035           JMP  J30            %7400     / BASE SET EXTENSION
0339 00377  321 161035           JMP  J30            %7420     / BASE SET EXTENSION
0340                    ********************************************************************************


0342                            ORG                 400B
0343                    ********************************************************************************
0344                    *
0345                    *       21MX MICRO-CODE
0346                    *       MODULE 1
0347                    *
0348                    ********************************************************************************
0349                    DISPLAYA EQU                %376
0350                    DISPLAYT EQU                %367
0351                    DISPLAYS EQU                %337
0352                    NSFPMOD  EQU                %5420     ENTRY FOR NON STD FRONT PANEL
0353                    ********************************************************************************
0354                    *       MEMORY INITIALIZATION ROUTINE
0355                    ********************************************************************************
0356 00400  322 161171  HALT     JMP  CNDX NMLS      MGOOD     JUMP IF MEMORY NOT LOST
0357 00401  341 004617           IMM       HIGH MEU  %102      ENABLE SYSTEM MAP
0358 00402  347 101017           IMM       LOW  S1   %340      S1 <= 2'S COMP OF 32
0359 00403  001 137057                     ZERO S2             CLR S2 (MAP ADDR)
0360 00404  017 142557                     PASS A    S2        CLR A-REG
0361 00405  017 142517                     PASS B    S2        CLR B-REG
0362 00406  017 142117                     PASS T    S2        CLR T REG
0363 00407  347 077117           IMM       CMH1 S3   %337      S3 <= "LOAD ADDR REG" COMMAND
0364 00410  347 076264  LOSTLOOP IMM  SHLT LOW  CNTR %337      CNTR<=COMP OF 32; CLEAR RUN FF
0365 00411  017 144617                     PASS MEU  S3        LOAD 0 INTO ADDR REG ON MEU
0366 00412  017 142620  MAPLOOP       MESP PASS MEU  S2        LOAD MAP IN MEU
0367 00413  000 043063                ICNT INC  S2   S2        INC MAP ADDR
0368 00414  323 020531           JMP  CNDX CNT8 RJS  MAPLOOP   LOOP(*32)
0369 00415  001 137717                     ZERO P              CLR P REG
0370 00416  160 074717           WRTE      INC  PNM  P         M<=P; P<=P+1; WRITE ZERO DATA
0371 00417  322 020731           JMP  CNDX AL15 RJS  *-1       LOOP UNTIL M=077777
0372 00420  000 041017                     INC  S1   S1        INC MAP CNTR
0373 00421  320 020431           JMP  CNDX TBZ  RJS  LOSTLOOP  LOOP (*32)
0374 00422  341 000617           IMM       HIGH MEU  %100      DISABLE ALL MAPS NOW...


0376                    ********************************************************************************
0377                    *       FRONT PANEL STANDARD SCAN ROUTINES
0378                    ********************************************************************************
0379 00423  334 175471  MGOOD    JMP  CNDX NSFP      NOFRONT   JUMP IF   NON-STANDARD FRONT PANE
0380 00424  017 115752                FTCH PASS S    DSPL      S<=DISPLAY;INITIALIZE MEM. PROTEC
0381 00425  330 121371           JMP  CNDX NSNG RJS  WAIT      JUMP IF "INSTR STEP" PRESSED
0382 00426  347 156357           IMM       LOW  DSPI DISPLAYT  ACTIVATE "T" INDICATOR IN DSPI
0383 00427  300 024270  WAIT     JSB                 UPDATE    UPDATE DISPLAY WITH PROPER DATA
0384 00430  334 021431           JMP  CNDX NSTB RJS  *         WAIT FOR BUTTON RELEASES
0385 00431  325 164231           JMP  CNDX RUN       RUN
0386 00432  334 175471           JMP  CNDX NSFP      NOFRONT   JUMP IF NON-STANDARD FRONT PANEL
0387 00433  334 061471           JMP  CNDX NSTB      *-2
0388 00434  017 136757  SCAN     NOP                           SCAN FOR SWITCH PRESSED
0389                    *                                     NOP ONE CYCLE TO SET SWITCH CONDI
0390 00435  332 122571           JMP  CNDX NLT  RJS  LEFT
0391 00436  331 023431           JMP  CNDX NINC RJS  INC.M
0392 00437  331 123531           JMP  CNDX NDEC RJS  DEC.M
0393 00440  333 025471           JMP  CNDX NSTR RJS  STOREX
0394 00441  333 121371           JMP  CNDX NRST RJS  WAIT
0395 00442  332 032231  SCANRT   JMP  CNDX NRT  RJS  RIGHTR    JUMP IF "RIGHT" TO TEST FOR ENTRY
0396                    *                                     INTO SPECIAL DISPLAY ROUTINE.
0397 00443  330 026171           JMP  CNDX NLDR RJS  LOADER
0398 00444  325 164231           JMP  CNDX RUN       RUN
0399 00445  330 161431           JMP  CNDX NSNG      WAIT+1    JMP IF "INSTR STEP" NOT PRESSED
0400 00446  335 040271           JMP  CNDX INT       INTERUPT  SERVICE ANY PENDING INTERRUPT
0401 00447  220 074712           READ FTCH INC  PNM  P         DO STANDARD FETCH ROUTINE
0402 00450  017 176305                ION  PASS DSPL S         DISPLAY <= S
0403 00451  017 100411                CLFL PASS IR   TAB
0404 00452  220 020673           READ JTAB INC  CM   ADR
```

```
0406                    ************************************************************************
0407                    *      DISPLAY INDICATOR SHIFT ROUTINES
0408                    ************************************************************************
0409 00453  017 117004  LEFT          R1   PASS S1   DSPI       S1<=DSPI SHIFTED RIGHT ONE
0410 00454  321 122771          JMP  CNDX AL0  RJS  LEFTA      JUMP IF DSPI WRAP-AROUND REQUIRED
0411 00455  017 140357  LEFTB              PASS DSPI S1        DSPI <= DSPI SHIFTED RIGHT ONE
0412 00456  320 021370          JMP                 WAIT       JUMP TO STANDARD SCAN ROUTINES
0413 00457  347 076357  LEFTA    IMM            LOW  DSPI DISPLAYS  DSPI WRAP-AROUND A TO S
0414 00460  320 021370          JMP                 WAIT       JUMP TO STANDARD SCAN ROUTINES
0415 00461  347 076157  RIGHT    IMM            LOW  L    DISPLAYS
0416 00462  017 016750               STFL IOR           DSPI       SET FLAG; TEST DSPI
0417 00463  320 123331          JMP  CNDX ONES RJS  RIGHTA     JUMP IF WRAP-AROUND OF DSPI REQD
0418 00464  157 117002          LWF  L1   PASS S1   DSPI       S1<=DSPI SHIFTED LEFT ONE
0419 00465  320 022670          JMP                 LEFTB
0420 00466  347 174357  RIGHTA   IMM            LOW  DSPI DISPLAYA  DSPI WRAP-AROUND S TO A
0421 00467  320 021370          JMP                 WAIT       JUMP TO STANDARD SCAN ROUTINES
0422                    ************************************************************************
0423                    *      INC M, DEC M ROUTINES
0424                    ************************************************************************
0425 00470  000 023017  INC.M              INC  S1   M          S1 <= M + 1
0426 00471  320 023570          JMP                 DEC.M+1
0427 00472  007 123017  DEC.M              DEC  S1   M          S1 <= M - 1
0428 00473  017 140457                     PASS M    S1         M <= S1
0429 00474  320 021370          JMP  UNCD           WAIT       JUMP TO STANDARD SCAN ROUTINE
0430                    ************************************************************************
0431                    *      SPECIAL TEST TO EXIT SPECIAL DISPLAY LOOP
0432                    ************************************************************************
0433 00475  017 116417  LEFTR              PASS IR   DSPI       CHECK FOR "M" DSPI
0434 00476  327 122571          JMP  CNDX IR2  RJS  LEFT       JUMP IF "M" TO LEAVE SPECIAL CODE
0435 00477  347 166357          IMM            LOW  DSPI %373   DSPI <= "M" (SHIFT FROM "T")
0436 00500  017 142317                     PASS DSPL S2         SHOW POINTER ON DISPLAY
0437 00501  320 033130          JMP  UNCD           WAITR      WAIT FOR BUTTON RELEASE IN SPECIA


0439                    ************************************************************************
0440                    *      STORE AND UPDATE ROUTINES
0441                    ************************************************************************
0442                    *                  THE REGISTER INDICATED IN DSPI IS THE BIT POSITION WH
0443                    *                  LOW. ALL OTHER BITS ARE 1.  THE ORDER (MSB TO LSB) IS
0444                    *                        S  P  T  M  B  A
0445                    *                  THE INDICATED REGISTER IS DETERMINED BY LOADING DSPI
0446                    *                  THE IR, AND JUMPING USING J30 TO GET TO THE APPROPRIA
0447                    *                  STORE OR UPDATE ROUTINE.  OTHER CODE IS INTERSPERSED
0448                    *                  FOR MAXIMUM CONTROL STORE EFFICIENCY
0449 00502  017 116417  STORE              PASS IR   DSPI
0450 00503  320 024035          JMP  J30            %0500      JMP TO STORE SELECTED REGISTER
0451 00504  347 076357  RUN      IMM            LOW  DSPI DISPLAYS  DSPI <= "S".   THE SAVE REGISTER
0452                    *                                      ZERO AT THIS POINT SO THE NEXT RT
0453                    *                                      WILL INITIATE THE FETCH ROUTINE
0454                    ************************************************************************
0455 00505  017 116417  UPDATE             PASS IR   DSPI
0456 00506  320 025035          JMP  J30            %520       JMP TO DISPLAY SELECTED REGISTER
0457                    ************************************************************************
0458 00507  177 114017          WRTE       PASS TAB  DSPL       STORE T
0459 00510  000 023017                     INC  S1   M
0460 00511  000 040457                     INC  M    S1         INCREMENT M, SET TAB LOGIC
0461 00512  320 021430          JMP  UNCD           WAIT+1
0462 00513  000 014476               RTN  INC  M    DSPL       STORE M
0463 00514  017 115776  STORES      RTN  PASS S    DSPL       STORE S
0464 00515  017 114536               RTN  PASS B    DSPL       STORE B
0465 00516  017 114576               RTN  PASS A    DSPL       STORE A
0466 00517  347 136157          IMM        LOW  L    %357       P OR S TO BE DISPLAYED
0467 00520  017 016757                     IOR       DSPI       MASK OUT "S"
0468 00521  320 164631          JMP  CNDX ONES       STORES     JUMP IF "S" INDICATED
0469 00522  017 115736               RTN  PASS P    DSPL       STORE P
0470                    ************************************************************************
0471                    ****** OVFL REG. STORE--PART OF SPECIAL DISPLAY ROUTINES ***************
0472 00523  017 115013  STOROO       SOV  PASS S1   DSPL       CHECK DISPLAY
0473 00524  321 165331          JMP  CNDX AL0        *+2
0474 00525  017 136754               COV                      CLEAR OVERFLOW
0475 00526  017 136776               RTN
0476                    ************************************************************************
0477 00527  220 022457          READ       INC  M    M          UPDATE T, READ M, SET TAB LOGIC
0478 00530  017 100336               RTN  PASS DSPL TAB        DSPL <= MEM DATA
0479                    ************************************************************************
0480 00531  300 024130  STOREX   JSB                 STORE      STORE ROUTINES END WITH RTN
0481 00532  320 021370          JMP                 WAIT       JUMP TO STANDARD SCAN ROUTINES
0482                    ************************************************************************
0483 00533  017 122336               RTN  PASS DSPL M          UPDATE M
0484 00534  017 176336  UPDATES      RTN  PASS DSPL S          UPDATE S
0485 00535  017 124336               RTN  PASS DSPL B          UPDATE B
0486 00536  017 126336               RTN  PASS DSPL A          UPDATE A
0487 00537  347 136157          IMM        LOW  L    357B       P OR S INDICATED
0488 00540  017 016757                     IOR       DSPI       MASK OUT "S"
0489 00541  320 165631          JMP  CNDX ONES       UPDATES
0490 00542  017 174336               RTN  PASS DSPL P          UPDATE P
```

```
0492                    ************************************************************************
0493                    *       21MX ROM BOOTSTRAP MEMORY LOADER ROUTINE
0494                    ************************************************************************
0495 00543  341 177053  LOADER    IMM  SOV  HIGH S2   %177       FORM 0111111111111111 (MAX ADDR)
0496 00544  353 137017            IMM       CMHI S1   %357       FORM 0001000000000000 (10K) IN S1
0497                    ****** DETERMINE MEMORY SIZE, STARTING ADDR FOR LOADER *****************
0498 00545  347 000157  SIZE      IMM       LOW  L    %300       FORM 1111111111000000 IN L
0499 00546  015 143717                      AND  P    S2         FORM STARTING ADDR IN P
0500 00547  010 075217                      CMPS S5   P          FORM TWO'S COMP
0501 00550  000 051217                      INC  S5   S5         OF SA IN S5
0502 00551  017 142457                      PASS M    S2         PUT LAST ADDR INTO M
0503 00552  320 161371            JMP  CNDX ONES      WAIT       TEST FOR NO READ/WRTE CAPABILITY
0504 00553  177 150117            WRTE      PASS T    S5         PASS INTO T
0505 00554  017 140157                      PASS L    S1         UPDATE LAST ADDR WHILE WAITING
0506 00555  223 043057            READ      SUB  S2   S2         TO RETRIEVE DATA
0507 00556  017 150157                      PASS L    S5         COMPARE WHAT WAS READ FROM MEM.
0508 00557  013 004757                      XOR       T          TO DATA WRITTEN (S3)
0509 00560  320 026271            JMP  CNDX TBZ  RJS  SIZE       IF IT CHECKS, WE HAVE CORRECT STR
0510                    ****** CHECK SELECT CODE IN  S REG.     *********************************
0511 00561  347 000157            IMM       LOW  L    %300       FORM 1111111111000000 IN L
0512 00562  347 164257            IMM       LOW  CNTR %372       CNTR GETS -6
0513 00563  017 176417                      PASS IR   S          SET UP LOADER SELECT BIT
0514 00564  017 177155                 RPT  PASS S4   S          SET UP S-REG FOR SHIFT
0515 00565  017 147144                 R1   PASS S4   S4         SHIFT SELECT CODE INTO BITS(0-5)
0516 00566  013 147157                      SANL S4   S4         MASK OFF SEL. CODE
0517 00567  347 160157            IMM       LOW  L    %370       FORM 1111111111111000 <=-10B) IN
0518 00570  004 147153                 SOV  ADD  S4   S4         SUB 10B FROM SEL CODE; SAVE IN S4
0519 00571  322 061371            JMP  CNDX AL15      WAIT       IF NEG RESULT, SCB < 10B; RTN W/
0520                    ****** PREPARE FOR LOADER TRANSFER **************************************
0521 00572  344 000257            IMM       LOW  CNTR %0         CLEAR CNTR (ROM ADDR REG)
0522 00573  017 174454                 COV  PASS M    P          PUT SA IN M;CLR OVF = NO OPER ERR
0523                    ****** TRANSFER CONTENTS OF LOADER ROM TO MEMORY ************************
0524 00574  017 131003  LOOP1          L4   PASS S1   LDR        PASS XXXXXXXXAAAAXXXX INTO S1;CNT
0525 00575  017 140163                 ICNT PASS L    S1         CNTR=X01
0526 00576  015 131003                 L4   AND  S1   LDR        FORM XXXXAAAABBBBXXXX IN S1;CNTR=
0527 00577  017 140163                 ICNT PASS L    S1         CNTR=X10
0528 00600  015 131003                 L4   AND  S1   LDR        FORM AAAABBBBCCCCXXXX IN S1;CNTR=
0529 00601  017 140163                 ICNT PASS L    S1         CNTR=X11
0530 00602  012 031017                      NAND S1   LDR        FORM AAAABBBBCCCCDDDD (CMPL FORM)
0531 00603  177 140117            WRTE      PASS T    S1         WRITE INTO MEMORY
0532 00604  000 023063                 ICNT INC  S2   M          UPDATE MEM ADDR;CNTR=X00
0533 00605  017 142457                      PASS M    S2         PASS NEW ADDR INTO M
0534 00606  344 000157            IMM       LOW  L    %0         FORM 1111111100000000 IN L
0535 00607  017 022757                      IOR       M          MASK M TO SEE IF LAST WORD OF LDR
0536 00610  320 127631            JMP  CNDX ONES RJS  LOOP1      IF M(0-8)=11111111, DON'TLOOP


0538                    ************************************************************************
0539 00611  347 000257            IMM       LOW  CNTR %300       SET UP COUNT TO FIND LAST WORD
0540 00612  344 077110            IMM  STFL LOW  S3   %037
0541 00613  157 145102            LWF  L1   PASS S3   S3         FORM 1111111000111111 IN S3
0542 00614  017 175017                      PASS S1   P          PASS SA INTO S1
0543                    ****** CHECK INSTRUCTION IN MEMORY FOR I/O TYPE *************************
0544 00615  237 140457  NUWRD     READ      PASS M    S1         PASS SA INTO M & READ FIRST INSTR
0545 00616  340 026157            IMM       HIGH L    %013       FORM COMP OF 1111010000000000 IN
0546 00617  017 105057                      PASS S2   T          SAVE WORD IN S2
0547 00620  013 143017                      SANL S1   S2         MASK UPPER BITS FOR I/O TYPE
0548 00621  341 166157            IMM       HIGH L    %173       FORM 0111101111111111 IN L
0549 00622  013 040757                      XOR       S1         NOW CHECK FOR I/O TYPE
0550 00623  320 171531            JMP  CNDX ONES      HTST       IF MATCH OCCURS, JUMP OUT OF LOOP
0551                    ************************************************************************
0552 00624  000 023023  UPDT           ICNT INC  S1   M          OTHERWISE UPDATE M IN S1
0553 00625  323 030671            JMP  CNDX CNT8 RJS  NUWRD      LOOP BACK
0554 00626  017 146154                 COV  PASS L    S4         PASS (SCB-10B) INTO L
0555 00627  004 143051                 CLFL ADD  S2   S2         CHNG SC OF DCPC CNTRL WORD
0556 00630  177 142117            WRTE      PASS T    S2         SAVE IN MEM
0557 00631  320 021370            JMP                 WAIT       RETURN TO SCAN ROUTINE
0558                    ****** UPDATE SELECT CODE IN I/O INSTRUCTION ****************************
0559 00632  017 144157  HTST               PASS L    S3         PASS 1111111000111111 INTO L
0560 00633  014 042757                      NSOL      S2         BLEND TO CHECK FOR...000...OF HLT
0561 00634  320 171231            JMP  CNDX ONES      UPD1       IF FOUND GET NEXT INSTR
0562 00635  347 016157            IMM       LOW  L    %307       FORM 1111111111000111 IN L
0563 00636  013 142757                      SANL      S2         MASK BITS TO CHECK FOR SC < 10B
0564 00637  320 071231            JMP  CNDX TBZ       UPDT       IF SO, RTN TO LOOP
0565 00640  017 146157                      PASS L    S4         PASS (SCB-10B) INTO L
0566 00641  004 143057                      ADD  S2   S2         ADD TO SC FROM INSTR
0567 00642  177 142117            WRTE      PASS T    S2         PASS INTO T AND WRITE INTO MEMORY
0568 00643  320 031230            JMP                 UPDT       RTN TO LOOP
```

```
0570                    ********************************************************************************
0571                    *       SPECIAL DISPLAY ROUTINES
0572                    ********************************************************************************
0573 00644  017 116417  RIGHTR              PASS IR    DSPI        "RIGHT" PRESSED: IR <= DSPI
0574 00645  327 163071            JMP  CNDX IR2        RIGHT       JUMP IF M NOT SELECTED BY DSPI
0575 00646  017 115057                      PASS S2    DSPL        S2 <= DSPL (POINTER)
0576 00647  322 023071            JMP  CNDX AL15 RJS   RIGHT       JMP IF DSPL BIT 15 WASNT SET
0577 00650  347 156357            IMM        LOW  DSPI %367        DSPI <= "T"
0578                    ********************************************************************************
0579 00651  006 042157  UPDATR              OP9  L     S2          CHECK DSPL BIT 14, STORE S2 IN L
0580 00652  322 074671            JMP  CNDX AL15       MEUMAPS     JUMP IF S2 BIT 14 = 1 TO UPDATE M
0581 00653  350 007003            IMM  L4   CMHI S1    %003        S1 <= MASK FOR REGISTERS = 140017
0582 00654  015 141057                      AND  S2    S1          S2 <= S2 MASK OUT UNUSED BITS
0583 00655  017 142417                      PASS IR    S2          SET REGISTER SELECTION
0584 00656  333 073071            JMP  CNDX NSTR       READREG     JUMP IF STORE BUTTON NOT PRESSED
0585 00657  300 037035            JSB  J30             STOREG      SELECTED REGISTER <= DISPLAY
0586 00660  320 033130            JMP  UNCD            WAITR       WAIT FOR NEXT BUTTON
0587 00661  300 036035  READREG   JSB  J30             DSPLREG     DISPLAY <= SELECTED REGISTER
0588                    ********************************************************************************
0589 00662  334 033131  WAITR     JMP  CNDX NSTB RJS   *           WAIT FOR BUTTON RELEASE
0590 00663  325 164231            JMP  CNDX RUN        RUN         JUMP IF RUN INDICATOR LIT
0591 00664  334 073171            JMP  CNDX NSTB       *-1         JUMP BACK IF NO BUTTON PRESSED
0592                    *
0593 00665  017 136757            NOP                              WAIT ONE CYCLE FOR SETTING SWITCH
0594 00666  332 123671            JMP  CNDX NLT  RJS   LEFTR       JUMP IF "LEFT" PRESSED
0595 00667  331 073531            JMP  CNDX NINC       NOTINC      JUMP IF "INCM" NOT PRESSED
0596 00670  000 043057                      INC  S2    S2          INCREMENT POINTER
0597 00671  320 034030            JMP  UNCD            DECMR+1
0598 00672  331 174231  NOTINC    JMP  CNDX NDEC       NOTDEC
0599 00673  340 000157            IMM        HIGH L    %000        CHECK FOR
0600 00674  015 142757                      AND       S2          DECREMENT OF
0601 00675  320 033771            JMP  CNDX TBZ  RJS   DECMR       ZERO COUNT
0602 00676  000 143057                      OP1  S2    S2          S2 OR L PLUS 1 (WRAP AROUND COUNT
0603 00677  007 143057  DECMR             DEC  S2    S2          DECREMENT POINTER
0604 00700  017 116417                      PASS IR    DSPI        IR <= DSPI
0605 00701  327 172471            JMP  CNDX IR2        UPDATR      JUMP IF M NOT INDICATED
0606 00702  017 142317                      PASS DSPL  S2          UPDATE DISPLAY
0607 00703  320 033130            JMP  UNCD            WAITR       WITH NEW POINTER VALUE AND JUMP
0608 00704  333 134031  NOTDEC    JMP  CNDX NRST RJS   DECMR+1     JUMP IF "DISPLAY" PRESSED
0609 00705  333 074471            JMP  CNDX NSTR       *+4         JUMP IF STORE NOT PRESSED
0610 00706  017 116417                      PASS IR    DSPI
0611 00707  327 125471            JMP  CNDX IR2  RJS   STOREX      JUMP IF M SELECTED. LEAVE SPECIAL
0612 00710  320 032470            JMP  UNCD            UPDATR      M NOT SELECTED
0613 00711  320 022130            JMP  UNCD            SCANRT      JUMP TO STD ROUTINES


0615                    ********************************************************************************
0616 00712  347 172417  STOREE    IMM        LOW  IR    %375        SET UP SRG TYPE ER* SHIFT
0617 00713  017 114741                SRG2 PASS        DSPL        SET E ACCORDING TO DSPL BIT 0
0618 00714  017 136776                RTN
0619                    ********************************************************************************
0620                    ****** MEU MAP MANIPULATIONS ***************************************************
0621 00715  346 000157  MEUMAPS  IMM        LOW  L     %200        S1 <= MASK OF LOW 7 BITS
0622 00716  013 143017                      SANL S1    S2
0623 00717  340 176157            IMM        HIGH L     %077        L <= 037777B
0624 00720  016 141057                      SONL S2    S1          S2 <= MASK OUT BITS 13 TO 8
0625 00721  343 076157            IMM        HIGH L     %337        OR IN BIT 13
0626 00722  016 141017                      SONL S1    S1
0627 00723  017 140617                      PASS MEU   S1          SEND MAP NO. TO MEU
0628 00724  333 075371            JMP  CNDX NSTR       READMAP     JUMP IF STORE NOT PRESSED
0629 00725  017 114620                MESP PASS MEU     DSPL        MEU MAP <= DISPLAY
0630 00726  320 033130            JMP  UNCD            WAITR
0631 00727  017 134320  READMAP        MESP PASS DSPL   MEU         DISPLAY <= MEU MAP
0632 00730  320 033130            JMP  UNCD            WAITR
0633                    ************* LINK FOR NON-STANDARD FRONT PANEL CODE MODULE ************
0634 00731  321 061030  NOFRONT  JMP                   NSFPMOD     GO TO MOD 11 FOR NON STD FRONT PA


0636                            ORG                   740B
0637                    ********************************************************************************
0638                    *       SHORT SUBROUTINES TO STORE/DISPLAY SELECTED REGISTERS
0639                    ********************************************************************************
0640 00740  017 170336  DSPLREG       RTN  PASS DSPL X           PASS REG TO FRONT PANEL AND RETUR
0641 00741  017 172336                RTN  PASS DSPL Y
0642 00742  017 112336                RTN  PASS DSPL CNTR
0643 00743  017 144336                RTN  PASS DSPL S3
0644 00744  017 146336                RTN  PASS DSPL S4
0645 00745  017 150336                RTN  PASS DSPL S5
0646 00746  017 152336                RTN  PASS DSPL S6
0647 00747  017 154336                RTN  PASS DSPL S7
0648 00750  017 156336                RTN  PASS DSPL S8
0649 00751  017 160336                RTN  PASS DSPL S9
0650 00752  017 162336                RTN  PASS DSPL S10
0651 00753  017 164336                RTN  PASS DSPL S11
```

```
0652 00754  017 166336              RTN  PASS DSPL S12
0653 00755  343 176336         IMM  RTN  HIGH DSPL 377B
0654 00756  343 176336         IMM  RTN  HIGH DSPL 377B
0655 00757  343 176336         IMM  RTN  HIGH DSPL 377B
0656                    ********************************************************************
0657 00760  017 115636  STOREG      RTN  PASS X    DSPL      STORE INTO REG FROM FRONT PANEL
0658 00761  017 115676              RTN  PASS Y    DSPL
0659 00762  017 114276              RTN  PASS CNTR DSPL
0660 00763  017 115136              RTN  PASS S3   DSPL
0661 00764  017 115176              RTN  PASS S4   DSPL
0662 00765  017 115236              RTN  PASS S5   DSPL
0663 00766  017 115276              RTN  PASS S6   DSPL
0664 00767  017 115336              RTN  PASS S7   DSPL
0665 00770  017 115376              RTN  PASS S8   DSPL
0666 00771  017 115436              RTN  PASS S9   DSPL
0667 00772  017 115476              RTN  PASS S10  DSPL
0668 00773  017 115536              RTN  PASS S11  DSPL
0669 00774  017 115576              RTN  PASS S12  DSPL
0670 00775  343 176336         IMM  RTN  HIGH DSPL %377
0671 00776  320 025170         JMP  UNCD           STOROO
0672 00777  320 034530         JMP  UNCD           STOREE
0673                    ********************************************************************
0674                         END
```

```
0676                    ********************************************************************
0677                    *       LOOK UP TABLE USED BY THE JTAB SPECIAL
0678                    ********************************************************************
0679 01000  000 000053         DEF                 SRG         1
0680 01001  000 000053         DEF                 SRG         2
0681 01002  000 000053         DEF                 SRG         3
0682 01003  000 000024         DEF                 ASGNOP      4
0683 01004  000 000031         DEF                 ASGCL*      5
0684 01005  000 000036         DEF                 ASGCM*      6
0685 01006  000 000043         DEF                 ASGCC*      7
0686 01007  000 000054         DEF                 SRG+1      10
0687 01010  000 000053         DEF                 SRG        11
0688 01011  000 000053         DEF                 SRG        12
0689 01012  000 000053         DEF                 SRG        13
0690 01013  000 000024         DEF                 ASGNOP     14
0691 01014  000 000031         DEF                 ASGCL*     15
0692 01015  000 000036         DEF                 ASGCM*     16
0693 01016  000 000043         DEF                 ASGCC*     17
0694 01017  000 000106         DEF                 AND        20
0695 01020  000 000106         DEF                 AND        21
0696 01021  000 000106         DEF                 AND        22
0697 01022  000 000106         DEF                 AND        23
0698 01023  000 000106         DEF                 AND        24
0699 01024  000 000106         DEF                 AND        25
0700 01025  000 000106         DEF                 AND        26
0701 01026  000 000106         DEF                 AND        27
0702 01027  000 000132         DEF                 JSB        30
0703 01030  000 000132         DEF                 JSB        31
0704 01031  000 000132         DEF                 JSB        32
0705 01032  000 000132         DEF                 JSB        33
0706 01033  000 000132         DEF                 JSB        34
0707 01034  000 000132         DEF                 JSB        35
0708 01035  000 000132         DEF                 JSB        36
0709 01036  000 000132         DEF                 JSB        37
0710 01037  000 000116         DEF                 XOR        40
0711 01040  000 000116         DEF                 XOR        41
0712 01041  000 000116         DEF                 XOR        42
0713 01042  000 000116         DEF                 XOR        43
0714 01043  000 000116         DEF                 XOR        44
0715 01044  000 000116         DEF                 XOR        45
0716 01045  000 000116         DEF                 XOR        46
0717 01046  000 000116         DEF                 XOR        47
0718 01047  000 000164         DEF                 JMP        50
0719 01050  000 000164         DEF                 JMP        51
0720 01051  000 000164         DEF                 JMP        52
0721 01052  000 000164         DEF                 JMP        53
0722 01053  000 000164         DEF                 JMP        54
0723 01054  000 000164         DEF                 JMP        55
0724 01055  000 000164         DEF                 JMP        56
0725 01056  000 000164         DEF                 JMP        57
0726 01057  000 000121         DEF                 IOR        60
0727 01060  000 000121         DEF                 IOR        61
0728 01061  000 000121         DEF                 IOR        62
```

```
0729 01062   000 000121          DEF             IOR         63
0731 01063   000 000121          DEF             IOR         64
0732 01064   000 000121          DEF             IOR         65
0733 01065   000 000121          DEF             IOR         66
0734 01066   000 000121          DEF             IOR         67
0735 01067   000 000136          DEF             ISZ         70
0736 01070   000 000136          DEF             ISZ         71
0737 01071   000 000136          DEF             ISZ         72
0738 01072   000 000136          DEF             ISZ         73
0739 01073   000 000136          DEF             ISZ         74
0740 01074   000 000136          DEF             ISZ         75
0741 01075   000 000136          DEF             ISZ         76
0742 01076   000 000136          DEF             ISZ         77
0743 01077   000 000127          DEF             AD*         100
0744 01100   000 000127          DEF             AD*         101
0745 01101   000 000127          DEF             AD*         102
0746 01102   000 000127          DEF             AD*         103
0747 01103   000 000127          DEF             AD*         104
0748 01104   000 000127          DEF             AD*         105
0749 01105   000 000127          DEF             AD*         106
0750 01106   000 000127          DEF             AD*         107
0751 01107   000 000127          DEF             AD*         110
0752 01110   000 000127          DEF             AD*         111

0753 01111   000 000127          DEF             AD*         112
0754 01112   000 000127          DEF             AD*         113
0755 01113   000 000127          DEF             AD*         114
0756 01114   000 000127          DEF             AD*         115
0757 01115   000 000127          DEF             AD*         116
0758 01116   000 000127          DEF             AD*         117
0759 01117   000 000111          DEF             CP*         120
0760 01120   000 000111          DEF             CP*         121
0761 01121   000 000111          DEF             CP*         122
0762 01122   000 000111          DEF             CP*         123
0763 01123   000 000111          DEF             CP*         124
0764 01124   000 000111          DEF             CP*         125
0765 01125   000 000111          DEF             CP*         126
0766 01126   000 000111          DEF             CP*         127
0767 01127   000 000111          DEF             CP*         130
0768 01130   000 000111          DEF             CP*         131
0769 01131   000 000111          DEF             CP*         132
0770 01132   000 000111          DEF             CP*         133
0771 01133   000 000111          DEF             CP*         134
0772 01134   000 000111          DEF             CP*         135
0773 01135   000 000111          DEF             CP*         136
0774 01136   000 000111          DEF             CP*         137
0775 01137   000 000144          DEF             LD*         140
0776 01140   000 000144          DEF             LD*         141
0777 01141   000 000144          DEF             LD*         142
0778 01142   000 000144          DEF             LD*         143
0779 01143   000 000144          DEF             LD*         144
0780 01144   000 000144          DEF             LD*         145
0782 01145   000 000144          DEF             LD*         146
0783 01146   000 000144          DEF             LD*         147
0784 01147   000 000144          DEF             LD*         150
0785 01150   000 000144          DEF             LD*         151
0786 01151   000 000144          DEF             LD*         152
0787 01152   000 000144          DEF             LD*         153
0788 01153   000 000144          DEF             LD*         154
0789 01154   000 000144          DEF             LD*         155
0790 01155   000 000144          DEF             LD*         156
0791 01156   000 000144          DEF             LD*         157
0792 01157   000 000124          DEF             ST*         160
0793 01160   000 000124          DEF             ST*         161
0794 01161   000 000124          DEF             ST*         162
0795 01162   000 000124          DEF             ST*         163
0796 01163   000 000124          DEF             ST*         164
0797 01164   000 000124          DEF             ST*         165
0798 01165   000 000124          DEF             ST*         166
0799 01166   000 000124          DEF             ST*         167
0800 01167   000 000124          DEF             ST*         170
0801 01170   000 000124          DEF             ST*         171
0802 01171   000 000124          DEF             ST*         172
0803 01172   000 000124          DEF             ST*         173
```

```
0804 01173  000 000124          DEF             ST*         174
0805 01174  000 000124          DEF             ST*         175
0806 01175  000 000124          DEF             ST*         176
0807 01176  000 000124          DEF             ST*         177
0808 01177  000 000102          DEF             EAU         200 MPY,ASL,LSL,RRL
0809 01200  000 000262          DEF             DIV         201
0810 01201  000 000102          DEF             EAU         202 ASR,LSR,RRR
0811 01202  000 000104          DEF             MAC1        203
0812 01203  000 000101          DEF             IOG         204 HLT,STF,SFC,SFS
0813 01204  000 000101          DEF             IOG         205 MIA,LIA,OTA,STC
0814 01205  000 000101          DEF             IOG         206 HLT,CLF
0815 01206  000 000101          DEF             IOG         207 MIA,LIA,OTA,STC
0816 01207  000 000224          DEF             DLD         210
0817 01210  000 000233          DEF             DST         211
0818 01211  000 000103          DEF             MAC0        212
0819 01212  000 000104          DEF             MAC1        213
0820 01213  000 000101          DEF             IOG         214 HLT,STF,SFC,SFS
0821 01214  000 000101          DEF             IOG         215 MIB,LIB,OTB,CLC
0822 01215  000 000101          DEF             IOG         216 HLT,CLF
0823 01216  000 000101          DEF             IOG         217 MIB,LIB,OTB,CLC
0824 01217  000 000105          DEF             AND,I       220
0825 01220  000 000105          DEF             AND,I       221
0826 01221  000 000105          DEF             AND,I       222
0827 01222  000 000105          DEF             AND,I       223
0828 01223  000 000105          DEF             AND,I       224
0829 01224  000 000105          DEF             AND,I       225

0830 01225  000 000105          DEF             AND,I       226
0831 01226  000 000105          DEF             AND,I       227
0833 01227  000 000131          DEF             JSB,I       230
0834 01230  000 000131          DEF             JSB,I       231
0835 01231  000 000131          DEF             JSB,I       232
0836 01232  000 000131          DEF             JSB,I       233
0837 01233  000 000131          DEF             JSB,I       234
0838 01234  000 000131          DEF             JSB,I       235
0839 01235  000 000131          DEF             JSB,I       236
0840 01236  000 000131          DEF             JSB,I       237
0841 01237  000 000115          DEF             XOR,I       240
0842 01240  000 000115          DEF             XOR,I       241
0843 01241  000 000115          DEF             XOR,I       242
0844 01242  000 000115          DEF             XOR,I       243
0845 01243  000 000115          DEF             XOR,I       244
0846 01244  000 000115          DEF             XOR,I       245
0847 01245  000 000115          DEF             XOR,I       246
0848 01246  000 000115          DEF             XOR,I       247
0849 01247  000 000145          DEF             JMP,I       250
0850 01250  000 000145          DEF             JMP,I       251
0851 01251  000 000145          DEF             JMP,I       252
0852 01252  000 000145          DEF             JMP,I       253
0853 01253  000 000145          DEF             JMP,I       254
0854 01254  000 000145          DEF             JMP,I       255
0855 01255  000 000145          DEF             JMP,I       256
0856 01256  000 000145          DEF             JMP,I       257
0857 01257  000 000120          DEF             IOR,I       260
0858 01260  000 000120          DEF             IOR,I       261
0859 01261  000 000120          DEF             IOR,I       262
0860 01262  000 000120          DEF             IOR,I       263
0861 01263  000 000120          DEF             IOR,I       264
0862 01264  000 000120          DEF             IOR,I       265
0863 01265  000 000120          DEF             IOR,I       266
0864 01266  000 000120          DEF             IOR,I       267
0865 01267  000 000135          DEF             ISZ,I       270
0866 01270  000 000135          DEF             ISZ,I       271
0867 01271  000 000135          DEF             ISZ,I       272
0868 01272  000 000135          DEF             ISZ,I       273
0869 01273  000 000135          DEF             ISZ,I       274
0870 01274  000 000135          DEF             ISZ,I       275
0871 01275  000 000135          DEF             ISZ,I       276
0872 01276  000 000135          DEF             ISZ,I       277
0873 01277  000 000126          DEF             AD*,I       300
0874 01300  000 000126          DEF             AD*,I       301
0875 01301  000 000126          DEF             AD*,I       302
0876 01302  000 000126          DEF             AD*,I       303
0877 01303  000 000126          DEF             AD*,I       304
0878 01304  000 000126          DEF             AD*,I       305
```

```
0879 01305  000 000126          DEF          AD*,I      306
0880 01306  000 000126          DEF          AD*,I      307
0881 01307  000 000126          DEF          AD*,I      310
0882 01310  000 000126          DEF          AD*,I      311
0884 01311  000 000126          DEF          AD*,I      312
0885 01312  000 000126          DEF          AD*,I      313
0886 01313  000 000126          DEF          AD*,I      314
0887 01314  000 000126          DEF          AD*,I      315
0888 01315  000 000126          DEF          AD*,I      316
0889 01316  000 000126          DEF          AD*,I      317
0890 01317  000 000110          DEF          CP*,I      320
0891 01320  000 000110          DEF          CP*,I      321
0892 01321  000 000110          DEF          CP*,I      322
0893 01322  000 000110          DEF          CP*,I      323
0894 01323  000 000110          DEF          CP*,I      324
0895 01324  000 000110          DEF          CP*,I      325
0896 01325  000 000110          DEF          CP*,I      326
0897 01326  000 000110          DEF          CP*,I      327
0898 01327  000 000110          DEF          CP*,I      330
0899 01330  000 000110          DEF          CP*,I      331
0900 01331  000 000110          DEF          CP*,I      332
0901 01332  000 000110          DEF          CP*,I      333
0902 01333  000 000110          DEF          CP*,I      334
0903 01334  000 000110          DEF          CP*,I      335
0904 01335  000 000110          DEF          CP*,I      336
0905 01336  000 000110          DEF          CP*,I      337
0906 01337  000 000143          DEF          LD*,I      340
0907 01340  000 000143          DEF          LD*,I      341

0908 01341  000 000143          DEF          LD*,I      342
0909 01342  000 000143          DEF          LD*,I      343
0910 01343  000 000143          DEF          LD*,I      344
0911 01344  000 000143          DEF          LD*,I      345
0912 01345  000 000143          DEF          LD*,I      346
0913 01346  000 000143          DEF          LD*,I      347
0914 01347  000 000143          DEF          LD*,I      350
0915 01350  000 000143          DEF          LD*,I      351
0916 01351  000 000143          DEF          LD*,I      352
0917 01352  000 000143          DEF          LD*,I      353
0918 01353  000 000143          DEF          LD*,I      354
0919 01354  000 000143          DEF          LD*,I      355
0920 01355  000 000143          DEF          LD*,I      356
0921 01356  000 000143          DEF          LD*,I      357
0922 01357  000 000123          DEF          ST*,I      360
0923 01360  000 000123          DEF          ST*,I      361
0924 01361  000 000123          DEF          ST*,I      362
0925 01362  000 000123          DEF          ST*,I      363
0926 01363  000 000123          DEF          ST*,I      364
0927 01364  000 000123          DEF          ST*,I      365
0928 01365  000 000123          DEF          ST*,I      366
0929 01366  000 000123          DEF          ST*,I      367
0930 01367  000 000123          DEF          ST*,I      370
0931 01370  000 000123          DEF          ST*,I      371
0932 01371  000 000123          DEF          ST*,I      372
0933 01372  000 000123          DEF          ST*,I      373
0934 01373  000 000123          DEF          ST*,I      374
0935 01374  000 000123          DEF          ST*,I      375
0936 01375  000 000123          DEF          ST*,I      376
0937 01376  000 000123          DEF          ST*,I      377
0938                            END
```

```
0001                        MICMX,L
0002                              ORG               %7000
0003                        ********************************************************************************
0004                        *
0005                        *       21MX MICRO-CODE
0006                        *       MODULE 14: FLOATING POINT INSTRUCTIONS
0007                        *     1 JULY 1975  CORRECTED BUG THAT CAUSED WRONG ANSWERS
0008                        *          WHEN ADDENDS WERE CLOSE TO 32768 APART AND NEGATIVE
0009                        *
0010                        ********************************************************************************
0011                        INDIRECT EQU                %0015
0012                        MPYX     EQU                %0246
0013                        ********************************************************************************
0014                        *
0015 07000   017 125414   IFIX           COV   PASS S9   B          CLEAR THE OVFL AND PUT EXP IN S9
0016 07001   321 100171             JMP  CNDX ALO  RJS  *+2        TEST FOR NEG EXP
0017 07002   001 136576             RTN  ZERO A                    IF EXP<0 WE CAN'T FIX
0018 07003   017 126517                  PASS B    A              PUT HIBITS IN B-REG
0019 07004   344 000157             IMM       LOW  L    %000      PUT 'UP-8' MASK IN L
0020 07005   015 160557                  AND  A    S9             MASK LEAST SIG. 8 BITS INTO A
0021 07006   015 161457                  AND  S10  S9             SAVE BITS FOR ROUND-OFF
0022 07007   013 161404             R1   SANL S9   S9             MASK EXP INTO S9 WITHOUT SIGN
0023 07010   347 140157             IMM       LOW  L    %360      PUT -20(B8) INTO L
0024 07011   004 161413                  SOV  ADD  S9   S9         CHECK TO SEE IF EXP TOO LARGE
0025 07012   320 140771             JMP  CNDX ONES      NOSHIFT    OR IF NO SHIFT REQUIRED
0026 07013   322 005231             JMP  CNDX AL15 RJS  OVER       IF SO THEN WE CAN'T FIX
0027 07014   000 061417                  INC  S9   S9             START LOOP TO SHIFT DIGITS
0028 07015   017 160255             RPT  PASS CNTR S9             PASS # OF SHIFTS INTO CNTR
0029 07016   037 124504             ARS  R1   PASS B    B         32-BIT SHIFT
0030 07017   017 126157   NOSHIFT        PASS L    A              HOLD LEFTOVER BITS IN L
0031 07020   017 124554             COV  PASS A    B              PUT INTEGER INTO A-REG
0032 07021   322 017631             JMP  CNDX AL15 RJS  RTNFP      TEST FOR NEG INTEGER
0033 07022   017 062757                  IOR       S10            IF NEG THEN CHECK FOR TRUNC. BITS
0034 07023   320 057631             JMP  CNDX TBZ       RTNFP      IF ALL ZEROS WE ARE DONE
0035 07024   000 024576             RTN  INC  A    B              OTHERWISE INC THE INTEGER & RTN
0036                        ********************************************************************************

0038                        ********************************************************************************
0039 07025   017 126517   FLOAT          PASS B    A              PUT INTEGER IN B-REG
0040 07026   001 136557                  ZERO A                   CLEAR A-REG
0041 07027   357 141417             IMM       CMLO S9   %360      STORE +15(B10) IN EXP REG
0042 07030   321 142530             JMP                 PACK
0043                        *
0044                        ********************************************************************************
0045                        *
0046 07031   017 101317   FLD            PASS S7   TAB          STORE HIBITS IN S7
0047 07032   000 023017                  INC  S1   M            INC ADDRS FOR NEXT READ
0048 07033   340 000157             IMM       HIGH L    %000      STORE 'LO-8' MASK IN L
0049 07034   220 040457             READ      INC  M    S1        READ SECOND HALF OF WRD
0050 07035   015 125417                  AND  S9   B            MEANWHILE,MASK EXP OF WRD1 INTO S
0051 07036   017 101217                  PASS S5   TAB          STORE WRD2 LOBITS/EXP IN S5
0052 07037   013 125457                  SANL S10  B            MASK LOBITS OF WRD1 INTO S10
0053 07040   013 151257                  SANL S6   S5           MASK LOBITS OF WRD2 INTO S6
0054 07041   015 151204             R1   AND  S5   S5           MASK EXP OF WRD2 INTO S1I WITHOUT
0055 07042   321 102271             JMP  CNDX ALO  RJS  *+3      IF SIGN WAS POS, JMP
0056 07043   346 000157             IMM       LOW  L    %200      OTHERWISE PUT -200(B8) INTO L
0057 07044   004 151217                  ADD  S5   S5           ADD TO EXP OF WRD2
0058 07045   017 161404             R1   PASS S9   S9           MASK EXP OF WRD1 INTO S9 WITHOUT
0059 07046   321 102471             JMP  CNDX ALO  RJS  *+3      IF SIGN WAS POS, JMP
0060 07047   346 000157             IMM       LOW  L    %200      OTHERWISE PUT -200(B8) INTO L
0061 07050   004 161417                  ADD  S9   S9           ADD TO EXP OF WRD1
0062 07051   017 127536             RTN  PASS S11  A            PUT HIBITS OF WRD1 INTO S3 & RTN
0063                        ********************************************************************************

0065                        ********************************************************************************
0066 07052   017 126154   PACK           COV  PASS L    A          CLR OVFL AND PUT WRD1 LOBITS INTO
0067 07053   001 137457                  ZERO S10                 CLEAR COUNTER REG
0068 07054   017 024757                  IOR       B              PASS THRU ALU WITH HIBITS
0069 07055   320 057631             JMP  CNDX TBZ       RTNFP      IF A/B IS ZERO,RTN
0070 07056   346 003517             IMM       LOW  S11  %201      STORE -177(B8) IN S11
0071 07057   037 124742   NRMLZ     ARS  L1   PASS       B        TEST IF NUMBER IS NORMALIZED
0072 07060   325 043231             JMP  CNDX OVFL      RND        IF SO, JMP TO ROUNDING ROUTINE
0073 07061   077 124502             LGS  L1   PASS B    B         IF NOT, DO 32-BIT LEFT-SHIFT
0074 07062   000 063457                  INC  S10  S10            INC THE EXP CNTR
0075 07063   321 142770             JMP                 NRMLZ      GO BACK TO CHECK FOR NORMAL NUMBE
0076 07064   322 043331   RND       JMP  CNDX AL15      *+2        SINCE B WAS JUST PASSED THRU ALU
0077 07065   007 165517                  DEC  S11  S11            CHECK SGN & ADJUST ROUND OFF
0078 07066   017 164154                  COV  PASS L    S11       PUT 'ROUND' INTO L
0079 07067   003 026557                  SUB  A    A              ACTUALLY; ADD 200(B8) TO LOBITS
0080 07070   321 004171             JMP  CNDX COUT RJS  XPNT     IF NO COUT FROM LOBITS, OK, JMP
0081 07071   340 000157             IMM       HIGH L    %0        CLR L(15) FOR OVERFLOW
0082 07072   240 024517             ENV       INC  B    B         IF COUT, INC HIBITS AND CHECK FOR
0083 07073   325 003771             JMP  CNDX OVFL RJS  *+4       IF NO OVFL, OK, JMP
```

```
0084 07074  017 124504                R1    PASS B    B          OVFL IMPLIES B/A= 1000...
0085 07075  000 061414                COV   INC  S9   S9         SO WE SHIFT B TO FORM 0100...&
0086 07076  321 144170          JMP                     XPNT       BUMP EXP, THEN JMP
0087 07077  037 124742          ARS   L1    PASS        B          IF B NEQ 100..,CHECK IF B=111..
0088 07100  325 044171          JMP   CNDX  OVFL        XPNT       IF NOT, JMP
0089 07101  077 124502          LGS   L1    PASS B      B          RE-NORMALIZE
0090 07102  000 063457                      INC  S10    S10
0091 07103  017 162153  XPNT          SOV   PASS L      S10        CLR OVFL AND PUT EXP INTO L
0092 07104  003 061417                      SUB  S9     S9         SUB CALC EXP FROM ORIG EXP
0093 07105  346 000157          IMM         LOW  L      %200       PUT -200(B8) INTO L
0094 07106  003 060757                      SUB         S9         TEST FOR EXP UNDERFLO
0095 07107  322 045131          JMP   CNDX  AL15        UNFLO      IF SO, JMP
0096 07110  004 160757                      ADD         S9         TEST FOR EXP OVERFLOW
0097 07111  322 017671          JMP   CNDX  AL15 RJS    OVFLO      IF SO, JMP (TO 7375)
0098 07112  157 160742          LWF   L1    PASS        S9         PASS EXP SIGN INTO FLAG-REG
0099 07113  157 161402          LWF   L1    PASS S9     S9         SHIFT EXP WITH SIGN
0100 07114  340 000157          IMM         HIGH L      %000       STORE 'LO-8' MASK IN L
0101 07115  015 161457                      AND  S10    S9         MASK EXP INTO S10
0102 07116  013 127417                      SANL S9     A          MASK LOBITS INTO S9
0103 07117  017 124557                      PASS A      B          PUT HIBITS INTO A-REG
0104 07120  017 160154                COV   PASS L      S9         PUT LOBITS INTO L
0105 07121  017 062536                RTN   IOR  B      S10        COMBINE WITH EXP AND STORE IN B-R
0106 07122  001 136557  UNFLO               ZERO A                 CLEAR A-REG; OVFL=1
0107 07123  001 136536                RTN   ZERO B                 NOW CLR B-REG AND RTN
0108                    *
0109 07124  341 176576  OVER    IMM   RTN   HIGH A     %177       SET UP ERROR CONDITION IN A
0110                    ********************************************************************************


0112                    ********************************************************************************
0113 07125  017 136750  FADD          STFL
0114                    *
0115 07126  220 074457  FSUB    READ        INC  M      P          PASS P INTO M TO READ ADDR OF WRD
0116 07127  300 000670          JSB                     INDIRECT   CHECK FOR INDIRECTS
0117 07130  301 141470          JSB                     FLD        UNPACK WRDS INTO SCRATCH REGS
0118 07131  017 154517                      PASS B      S7         CHECK FOR WRD2=0
0119 07132  320 005631          JMP   CNDX  TBZ  RJS    *+2        IF NOT,CONTINUE
0120 07133  346 001217          IMM         LOW  S5     %200       IF SO,MAKE EXP MOST NEG (-200,B8)
0121 07134  017 164757                      PASS        S11        CHECK FOR WRD1=0
0122 07135  320 005771          JMP   CNDX  TBZ  RJS    *+2        IF NOT,CONTINUE
0123 07136  346 001417          IMM         LOW  S9     %200       IF SO,MAKE EXP MOST NEG (-200,B8)
0124 07137  324 046531          JMP   CNDX  FLAG        DIFR       IF DOING ADD,SKIP AHEAD
0125 07140  010 024517                      CMPS B      B          FORM 2-COMP OF HIBITS IN B
0126 07141  010 053257                      CMPS S6     S6         FORM 2-COMP OF
0127 07142  000 053257                      INC  S6     S6         LOBITS OF WRD2
0128 07143  321 006531          JMP   CNDX  COUT RJS    DIFR       IF COUT OCCURS
0129 07144  000 024517                      INC  B      B          BUMP HIBITS
0130 07145  322 006531          JMP   CNDX  AL15 RJS    DIFR       CHECK SGN; IF POS,JMP
0131 07146  017 124742                L1    PASS        B          IF NEG,CHECK FOR MOST
0132 07147  320 006531          JMP   CNDX  TBZ  RJS    DIFR       NEG #(100...)
0133 07150  017 124504                R1    PASS B      B          IF SO,SHIFT BACK (010...)
0134 07151  000 051217                      INC  S5     S5         &BUMP EXP
0135 07152  017 152554  DIFR          COV   PASS A      S6
0136 07153  017 150157                      PASS L      S5         FIND DIFF IN EXPS
0137 07154  003 061351                CLFL  SUB  S8     S9         &STORE IN S8; FLG=0
0138 07155  320 047731          JMP   CNDX  TBZ         ADD2       IF DIFF=0,JMP TO ADD STEP
0139 07156  322 047131          JMP   CNDX  AL15        RVRS       IF NEG,WRD2>WRD1
0140 07157  010 057357                      CMPS S8     S6         FORM -DIFF
0141 07160  000 057357                      INC  S8     S8         & STORE -DIFF IN S8
0142 07161  321 147430          JMP                     SWAMPCHK
0143 07162  017 124157  RVRS              PASS L      B          HOLD B IN L
0144 07163  017 164517                      PASS B      S11        WRD1<WRD2; FILL B,A
0145 07164  017 162557                      PASS A      S10        WITH S11,S10
0146 07165  015 037517                      PASL S11               ALSO FILL S11,S10,S9
0147 07166  017 153457                      PASS S10    S6         WITH B,S6,S5
0148 07167  017 151417                      PASS S9     S5
0149 07170  347 120157  SWAMPCHK IMM         LOW  L      %350       FORM -30(B8) IN L
0150 07171  003 056757                      SUB         S8         IF -DIFF>-31,RTN WITH LARGER #
0151 07172  322 050731          JMP   CNDX  AL15        OUT        JMP TO RESTORE A,B
0152 07173  037 124504  SHIFT   ARS   R1    PASS B      B          NOW START SHIFT LOOP
0153 07174  000 057357                      INC  S8     S8         INC COUNTER
0154 07175  320 007571          JMP   CNDX  TBZ  RJS    SHIFT      LOOP UNTIL DONE
0155                    *
0156                    ***** CONTINUED ON NEXT PAGE ***************************************************


0158                    *
0159 07176  157 164142  ADD2    LWF   L1    PASS L      S11        FLG <= SIGN; L <= HIBITS
0160 07177  244 124517          ENV         ADD  B      B          ADD HIBITS; B <= RESULTS; OVFL?
0161 07200  017 162157                      PASS L      S10        L <= LOBITS
0162 07201  004 126557                      ADD  A      A          ADD LOBITS; A <= RESULTS
0163 07202  321 010271          JMP   CNDX  COUT RJS    *+3        CHECK FOR COUT FROM LOBITS
0164 07203  341 176157          IMM         HIGH L      %177       L <= 0111111111111111
0165 07204  240 024517          ENV         INC  B      B          B <= B + 1; OVFL?
0166 07205  325 016271          JMP   CNDX  OVFL RJS    PRSUB      IF NO OVERFLOW, RETURN
```

```
0167 07206  324 010531          JMP  CNDX FLAG RJS  OFLOW     CHECK IF ADDEND, AUGEND WERE POS
0168 07207  341 176157          IMM       HIGH L    %177      L <= 0111111111111111
0169 07210  013 024757                    XOR       B         TEST FOR B = 1000000000000000
0170 07211  320 156271          JMP  CNDX ONES      PKSUB     IF TRUE, THEN IGNORE OVERFLOW
0171                    *
0172 07212  157 124504  OFLOW    LWF  R1   PASS B    B         DO FULLWORD SHIFT
0173 07213  157 126544          LWF  R1   PASS A    A            USING FLAG TO INJECT SIGN BIT
0174 07214  000 061417                    INC  S9     S9        BUMP EXPONENT
0175 07215  321 156270          JMP                 PKSUB
0176                    *
0177 07216  017 164517  OUT               PASS B    S11       PASS MUCH LARGER WRD INTO B,A
0178 07217  017 162557                    PASS A    S10
0179 07220  321 156270          JMP                 PKSUB
0180                    ********************************************************************************


0182                    ********************************************************************************
0183 07221  220 074457  FMPY    READ      INC  M    P         PASS P INTO M TO READ ADDR OF WRD
0184 07222  300 000670          JSB                 INDIRECT  CHECK FOR INDIRECTS
0185 07223  301 141470          JSB                 FLD       STORE ARGS IN SCRATCH REGS
0186 07224  000 061417                    INC  S9     S9
0187 07225  017 150157                    PASS L    S5        FORM EXP1+EXP2+1
0188 07226  004 161417                    ADD  S9     S9        AND SAVE IN S9
0189 07227  017 162544               R1   PASS A    S10       FORM (WRD1 LOBITS)/2 IN A
0190 07230  017 155057                    PASS S2   S7        PASS WRD2 HIBITS INTO S2
0191 07231  300 012330          JSB                 MPYX      JMP TO MPY SUB & RTN WITH
0192 07232  017 125217                    PASS S5   B         HIBITS IN B; SAVE IN S5
0193 07233  017 165057                    PASS S2   S11       PASS WRD1 HIBITS INTO S2
0194 07234  017 127517                    PASS S11  A         LOBITS INTO A; SAVE IN S11
0195 07235  017 152544               R1   PASS A    S6        FORM (WRD2 LOBITS)/2 IN A
0196 07236  300 012330          JSB                 MPYX      JMP TO MPY SUB & RTN WITH
0197 07237  017 126157                    PASS L    A         LOBITS IN A; PASS INTO L
0198 07240  004 164557                    ADD  A    S11       ADD BOTH LOBITS & CHK FOR COUT
0199 07241  321 012171          JMP  CNDX COUT RJS  *+2       (ELSE TRUNCATE DIGITS)
0200 07242  000 024517                    INC  B    B         IF COUT,BUMP HIBITS
0201 07243  017 124157                    PASS L    B         ADD HIBITS AND SAVE IN S11
0202 07244  004 151517                    ADD  S11  S5
0203 07245  017 154557                    PASS A    S7        PASS WRD2 HIBITS INTO A
0204 07246  300 012330          JSB                 MPYX      JMP TO MPY SUB & RTN WITH
0205 07247  017 126544               R1   PASS A    A         LOBITS IN A; SAVE LOBITS/2
0206 07250  017 126154               COV  PASS L    A         ADD LOBITS/2 TO HIBITS SUM &
0207 07251  244 164542          ENV  L1   ADD  A    S11       SHFT L1 TO REORIENT
0208 07252  322 012671          JMP  CNDX AL15 RJS  *+3       CHECK FOR CARRY INTO OR
0209 07253  325 052771          JMP  CNDX OVFL      *+4       BORROW FROM HIBITS &
0210 07254  007 124517                    DEC  B    B         ADJUST ACCORDINGLY
0211 07255  301 142530          JSB                 PACK
0212 07256  000 075736               RTN  INC  P    P
0213 07257  000 024517                    INC  B    B         CAN'T OVFL FROM HIBITS
0214 07260  301 142530          JSB                 PACK
0215 07261  000 075736               RTN  INC  P    P
0216                    ********************************************************************************


0218                    ********************************************************************************
0219 07262  220 074457  FDIV    READ      INC  M    P         PASS P INTO M TO READ ADDR OF WRD
0220 07263  300 000670          JSB                 INDIRECT  CHECK FOR INDIRECTS
0221 07264  301 141470          JSB                 FLD
0222 07265  010 054554               COV  CMPS A    S7        PASS WRD2 HIBITS & CHECK
0223 07266  320 156131          JMP  CNDX ONES      DBYZR     FOR DIV BY ZERO
0224 07267  322 053471          JMP  CNDX AL15      *+2       SINCE WE USE SAME DVSR,MAKE POS
0225 07270  000 027313               SOV  INC  S7   A         NOW & SAVE SGN IN OVFL
0226 07271  017 150157                    PASS L    S5        FORM EXP1-EXP2+1
0227 07272  003 061417                    SUB  S9   S9        & SAVE IN S9
0228 07273  000 061417                    INC  S9   S9
0229 07274  017 162557                    PASS A    S10       FILL B,A WITH WRD1 AS DVND
0230 07275  017 164517                    PASS B    S11       & PRESHIFT TO AVOID OVFL
0231 07276  037 124504          ARS  R1   PASS B    B
0232 07277  301 156370          JSB                 DIVX      JMP TO SPECIAL DIV SUB
0233 07300  017 127217                    PASS S5   A         SAVE QUO1 IN S5
0234 07301  017 124757                    PASS      B         PASS QUO & CHECK FOR ODD/EVEN
0235 07302  321 114231          JMP  CNDX AL0  RJS  *+2       TO SIMULATE FIRST
0236 07303  007 124517                    DEC  B    B         LEFT SHIFT IN DIV ROUTINE
0237 07304  001 136557                    ZERO A              CLR DVND LOBITS; DVSR SAME
0238 07305  301 156370          JSB                 DIVX      JMP TO SPEC DIV SUB
0239 07306  017 127517                    PASS S11  A         SAVE QU02 IN S11
0240 07307  017 152504               R1   PASS B    S6        FORM (WRD2 LOBITS)/4 IN
0241 07310  017 124504               R1   PASS B    B         B(=DVND HIBITS)
0242 07311  001 136557                    ZERO A              CLR DVND LOBITS; DVSR SAME
0243 07312  301 156370          JSB                 DIVX      JMP TO SPEC DIV SUB
0244 07313  010 026557                    CMPS A    A         FORM 2-COMP OF QU03
0245 07314  000 026557                    INC  A    A         AS MPLR
0246 07315  017 151057                    PASS S2   S5        PASS QU01 AS MCND
0247 07316  300 012330          JSB                 MPYX      JMP TO MPY SUB
0248 07317  017 125317                    PASS S7   B         SAVE PROD HIBITS IN S5
0249 07320  001 136517                    ZERO B              PRE-CLR B
```

```
0250 07321  017 164757                PASS       S11        CHECK SGN OF QU02
0251 07322  322 015231          JMP   CNDX AL15 RJS  *+2        & EXTEND AS ALL 3'S(POS)
0252 07323  016 036517                ONE   B                   OR ALL 1'S(NEG)
0253 07324  017 154757                PASS       S7         CHECK SGN OF -QU01*QU03
0254 07325  322 015371          JMP   CNDX AL15 RJS  *+2        IF NEG,SUB 1 FROM B
0255 07326  007 124517                DEC   B    B
0256 07327  017 155302           L1   PASS S7    S7         REORIENT PROD (ADJUST EXP,REALLY)
0257 07330  017 154542           L1   PASS A     S7
0258 07331  017 126157                PASS L     A
0259 07332  004 164557                ADD   A    S11        ADD TO QU02
0260 07333  321 015671          JMP   CNDX COUT RJS  *+2        IF COUT OCCURRED
0261 07334  000 024517                INC   B    B          BUMP HIBITS OF RESULT
0262 07335  077 124502          LGS   L1   PASS B    B          SHIFT FULLWRD TO ORIENT RESULT
0263 07336  017 150157                PASS L     S5         ADD QU01 TO HIBITS
0264 07337  004 124517                ADD   B    B
0265 07340  301 142530          JSB                PACK
0266 07341  000 075736                RTN   INC   P    P
0267 07342  001 136557  DBYZR         ZERO  A                CLR LOBITS
0268 07343  352 000517          IMM   CMH1  B    %200       FORM 0111111100000000 IN HIBITS
0269 07344  017 125417                PASS S9    B          ALSO PASS INTO EXP
0270 07345  301 142530  PKSUB JSB                PACK       REPACK A,B REGS
0271 07346  000 075736                RTN   INC   P    P          INC P AND RETURN
0272                    ********************************************************************
```

```
0274                    ********************************************************************
0275                    *
0276                    *     THIS IS A SPECIAL SUB FOR F.P. DIV
0277                    *        IT ASSUMES THAT DVSR IS ALWAYS POS
0278                    *                   THAT ORIG DVSR SGN IS IN OVFL REG
0279                    *                   THAT YOU HAVE PREVIOUSLY DONE FIRST LEFT SHIFT
0280                    *                AND THAT NO ERROR COND NEED BE CHECKED
0281                    *        (BUT IT IS FAST)
0282                    *
0283                    ********************************************************************
0284 07347  344 000257  DIVX      IMM        LOW  CNTR %0       CLR CNTR
0285 07350  157 124742           LWF   L1   PASS       B       CHECK FOR NEG DVND & SAVE SGN IN
0286 07351  322 016771           JMP   CNDX AL15 RJS  READY     IF POS, WE ARE READY
0287 07352  010 024517                CMPS B    B          COMP HIBITS
0288 07353  010 026557                CMPS A    A          COMP LOBITS
0289 07354  000 026557                INC  A    A          FORM 2-COMP OF LOBITS
0290 07355  321 016771           JMP   CNDX COUT RJS  READY     IF NO COUT,OK
0291 07356  000 024517                INC  B    B          ELSE BUMP HIBITS
0292 07357  017 154155  READY         RPT  PASS L    S7         PASS DVSR INTO L; SET RPTFF
0293 07360  123 024502           DIV   L1   SUB  B    B          PERFORM DIV STEP(16X)
0294 07361  017 124504           R1   PASS B    B          FORM REM IN B
0295 07362  324 017271           JMP   CNDX FLAG RJS  *+3        IF REM SGN IS TO BE NEG
0296 07363  010 024517                CMPS B    B          (DETERMINED BY DVND),THEN
0297 07364  000 024517                INC  B    B          FORM 2-COMP IN B
0298 07365  325 057471           JMP   CNDX OVFL       *+4        CHECK ORIG DVSR SGN; IF POS.
0299 07366  324 017631           JMP   CNDX FLAG RJS  RTNFP     LOOK FOR NEG DVND
0300 07367  010 026557                CMPS A    A          WHICH MEANS FORM
0301 07370  000 026576           RTN  INC  A    A          NEG QU0 IN A & RTN
0302 07371  324 057631           JMP   CNDX FLAG       RTNFP      ELSE IF NEG,LOOK FOR POS DVND
0303 07372  010 026557                CMPS A    A          WHICH MEANS FORM
0304 07373  000 026576           RTN  INC  A    A          NEG QU0 IN A & RTN
0305                    ********************************************************************
0306                    *
0307 07374  017 136776  RTNFP         RTN
0308                    *
0309 07375  016 036544  OVFLO         R1   ONE  A                PUT MOST POS # AND MOST POS EXP
0310 07376  347 174536           IMM  RTN  LOW  B    %37o       INTO A,B-REGS; OVFLO=1
0311                    ********************************************************************
```

```
0313                             ORG                %7400
0314                    ********************************************************************
0315                    *
0316                    *      21MX MICRO-CODE
0317                    *      MODULE 15: EXTENDED INSTRUCTION GROUP
0318                    *
0319                    * 1976-2-4-1730 CMW FIX
0320                    ********************************************************************
0321                    FETCH    EQU                %0000
0322                    ********************************************************************
0323                    *   JUMP TABLES - ENTERED FROM BASE SET
0324                    ********************************************************************
0325 07400  321 163170           JMP                EADRX      SAX/SBX
0326 07401  017 103636                RTN  PASS X    CAB        CAX/CBX
0327 07402  321 163170           JMP                EADRX      LAX/LBX
0328 07403  321 163030           JMP                EADR       STX
0329 07404  017 170076                RTN  PASS CAB  X          CXA/CXB
0330 07405  321 163030           JMP                EADR       LDX
0331 07406  321 163030           JMP                EADR       ADX
0332 07407  321 164070           JMP                XABX       XAX/XBX
```

```
0333 07410   321 163630          JMP                     EADRY     SAY/SBY
0334 07411   017 103676               RTN  PASS Y        CAB       CAY/CBY
0335 07412   321 163630          JMP                     EADRY     LAY/LBY
0336 07413   321 163030          JMP                     EADR      STY
0337 07414   017 172076               RTN  PASS CAB  Y   CYA/CYB
0338 07415   321 163030          JMP                     EADR      LDY
0339 07416   321 163030          JMP                     EADR      ADY
0340 07417   321 164230          JMP                     XABY      XAY/XBY
0341 07420   321 164370          JMP                     ISX
0342 07421   321 164670          JMP                     DSX
0343 07422   321 177130          JMP                     JLY
0344 07423   321 172670          JMP                     LBT
0345 07424   321 171770          JMP                     SBT
0346 07425   321 173370          JMP                     MBT
0347 07426   321 175170          JMP                     CBT
0348 07427   321 174170          JMP                     SFB
0349 07430   321 164530          JMP                     ISY
0350 07431   321 165030          JMP                     DSY
0351 07432   321 177430          JMP                     JPY
0352 07433   321 167330          JMP                     SBSCBS    SBS
0353 07434   321 167330          JMP                     SBSCBS    CBS
0354 07435   321 166630          JMP                     TBS
0355 07436   321 170230          JMP                     CMW
0356 07437   321 171170          JMP                     MVW


0358                     ************************************************************************
0359                     *       INDEX REGISTER INSTRUCTIONS
0360                     ************************************************************************
0361                     *                               DISPLACEMENT FROM FINISH CORREPONDS TO
0362                     *                               DISPLACEMENT FROM 7400B FOR INSTRS. LISTED
0363                     *                               IN COMMENT FIELD BELOW.
0364 07440   000 022461 FINISH        MPCK INC  M       M         SAX/SBX
0365 07441   177 102036          WRTE RTN  PASS TAB  CAB
0366 07442   017 100076               RTN  PASS CAB  TAB         LAX/LBX
0367 07443   000 022461               MPCK INC  M       M         STX
0368 07444   177 170036          WRTE RTN  PASS TAB  X
0369 07445   017 101636               RTN  PASS X    TAB         LDX
0370 07446   017 100157                    PASS L    TAB         ADX
0371 07447   264 171636          ENVE RTN  ADD  X    X
0372 07450   000 022461               MPCK INC  M       M         SAY/SBY
0373 07451   177 102036          WRTE RTN  PASS TAB  CAB
0374 07452   017 100076               RTN  PASS CAB  TAB         LAY/LBY
0375 07453   000 022461               MPCK INC  M       M         STY
0376 07454   177 172036          WRTE RTN  PASS TAB  Y
0377 07455   017 101676               RTN  PASS Y    TAB         LDY
0378 07456   017 100157                    PASS L    TAB         ADY
0379 07457   264 173676          ENVE RTN  ADD  Y    Y
0380                     ************************************************************************
0381                     *                               EADR IS COMMON TO LD*,ST*,AD*
0382 07460   220 074717 EADR      READ      INC  PNM  P         READ WORD 2 . P<=ADDR OF NEXT INS
0383 07461   301 165630          JSB                 INDBIT    CHECK FOR INDIRECT,GET OPERAND
0384 07462   321 162035          JMP  J30            FINISH    JUMP TO COMPLETE INSTRUCTION
0385                     ************************************************************************
0386                     *                               EADRX DOES EFFECTIVE ADDR FOR
0387                     *                               SAX,SBX,LAX,LBX INSTRS.
0388 07463   220 074717 EADRX     READ      INC  PNM  P         READ ADDRESS OF WORD 2
0389 07464   017 170157                    PASS L    X
0390 07465   017 100457                    PASS M    TAB       M<=CONTENTS OF WORD 2.
0391 07466   322 023471          JMP  CNDX AL15 RJS  DIRECT    JUMP IF NO INDIRECT.
0392                     ************************************************************************
0393                     *                               INDIRECT ROUTINE FOR INDEXED INST
0394 07467   220 022457 EADRI     READ      INC  M    M         READ INDIRECT ADDRESS
0395 07470   301 165630          JSB                 INDBIT    JSB TO INDIRECT ROUTINE
0396                     ************************************************************************
0397                     *                               COMPUTE INDEXED ADDRESS THEN JUMP
0398 07471   004 123017 DIRECT              ADD  S1   M         S1<=TARGET ADDR. + X OR Y.
0399 07472   220 040457          READ      INC  M    S1        READ INDEXED ADDRESS.
0400 07473   321 162035          JMP  J30            FINISH    JUMP TO COMPLETE THE INSTRUCTION
0401                     ************************************************************************
0402                     *                               EADRY COMPUTES EFFECTIVE ADDRESS
0403                     *                               FOR SAY,SBY,LAY,LBY INSTRS.
0404 07474   220 074717 EADRY     READ      INC  PNM  P
0405 07475   017 172157                    PASS L    Y
0406 07476   017 100457                    PASS M    TAB       M<= CONTENTS OF WORD 2.
0407 07477   322 023471          JMP  CNDX AL15 RJS  DIRECT    JUMP IF NO INDIRECTS.
0408 07500   321 163370          JMP                 EADRI     JUMP TO DO INDIRECT ROUTINE


0410                     ************************************************************************
0411 07501   017 103017 XABX                PASS S1   CAB       EXCHANGE A/B WITH X
0412 07502   017 170057                    PASS CAB  X
0413 07503   017 141636                RTN  PASS X    S1
0414                     ************************************************************************
0415 07504   017 103017 XABY                PASS S1   CAB       EXCHANGE A/B WITH Y
0416 07505   017 172057                    PASS CAB  Y
```

```
0417 07506  017 141676                RTN  PASS Y    S1
0418                    ********************************************************************
0419 07507  000 071617  ISX           INC  X    X          INCREMENT X, SKIP IF ZERO
0420 07510  320 067271          JMP   CNDX TBZ            SKIP
0421 07511  017 136776  RETURN        RTN
0422                    ********************************************************************
0423 07512  000 073657  ISY           INC  Y    Y          INCREMENT Y, SKIP IF ZERO.
0424 07513  320 067271          JMP   CNDX TBZ            SKIP
0425 07514  017 136776              RTN
0426                    ********************************************************************
0427 07515  007 171617  DSX           DEC  X    X          DECREMENT X, SKIP IF ZERO.
0428 07516  320 067271          JMP   CNDX TBZ            SKIP
0429 07517  017 136776              RTN
0430                    ********************************************************************
0431 07520  007 173657  DSY           DEC  Y    Y          DECREMENT Y, SKIP IF ZERO.
0432 07521  320 067271          JMP   CNDX TBZ            SKIP
0433 07522  017 136776              RTN


0435                    ********************************************************************
0436                    *   GENERAL INDIRECT ROUTINE FOR INDEX BIT INSTR
0437                    *       COMMON ROUTINES FOR WORD/BYTE INSTRUCTIONS
0438                    ********************************************************************
0439                    *                                        INITIALIZATION FOR WORD,BYTE
0440 07523  000 075217  INITCM        INC  S5   P        S5<= ADDRESS OF WORD 3.
0441 07524  220 050457          READ  INC  M    S5         READ ADDRESS OF WORD 3.
0442 07525  000 051157              INC  S4   S5         S4<= ADDRESS OF NEXT INSTRUCTION.
0443 07526  017 101117              PASS S3   TAB        S3<= CONTENTS OF WORD 3.
0444 07527  320 065531          JMP   CNDX TBZ    *+3      JUMP IF WORD 3 = 0 (NO INTERRUPT)
0445 07530  000 075717              INC  P    P          P<=ADDRESS OF WORD 3 (FOR EXIT)
0446 07531  001 155336              RTN  ZERO S7   S7      S7<=0 AND RETURN TO CALLER.
0447 07532  220 074710          READ  STFL INC  PNM  P     READ ADDRESS OF WORD 2. P<=P+1.
0448 07533  001 155317              ZERO S7   S7         S7 <= 0.
0449                    ********************************************************************
0450                    *                                        COMMON INDIRECT IMBEDDED IN INITC
0451 07534  017 100457  INDBIT        PASS M    TAB        M <= CONTENTS OF LAST READ ADDRES
0452 07535  322 026271          JMP   CNDX AL15 RJS  CONTBIT  JUMP IF NO INDIRECT.
0453 07536  220 022465  INDLBIT  READ  INCI INC  M    M     READ ADDRESS IN M
0454 07537  326 065631          JMP   CNDX NHOI   INDBIT    JUMP IF NO HALT OR INTERRUPT PEND
0455 07540  017 100457  IND2BIT        PASS M    TAB        M<= CONTENTS OF LAST READ ADDRESS
0456 07541  330 125671          JMP   CNDX NSNG RJS  INDBIT+1 JUMP IF SINGLE-INSTRUCT. MODE
0457 07542  007 175717  DEC2          DEC  P    P
0458 07543  007 175717              DEC  P    P          P <= ADDRESS OF WORD 1.
0459 07544  320 000030          JMP              FETCH      ATTEMPT JUMP TO FETCH ROUTINE.
0460 07545  324 066371  CONTBIT  JMP   CNDX FLAG   *+2      FLAG IDENTIFIES CALLER TO INDBIT
0461 07546  220 022476          READ  RTN  INC  M    M     READ ADDRESS AND RETURN.
0462                    ********************************************************************
0463 07547  220 022451          READ  CLFL INC  M    M     CALLER=INITCM--RESET FLAG,READ CO
0464 07550  017 101257              PASS S6   TAB        S6 <= COUNT FOR THIS INSTRUCTION
0465 07551  320 026571          JMP   CNDX TBZ  RJS  RTNCNT  JUMP IF COUNT NOT ZERO.
0466 07552  301 176770          JSB              EXIT       END THE INSTRUCTION.
0467 07553  017 153136  RTNCNT        RTN  PASS S3   S6     S3 <= COUNT, RETURN TO CALLER.


0469                    ********************************************************************
0470                    *       BIT INSTRUCTIONS
0471                    ********************************************************************
0472 07554  220 074717  TBS       READ       INC  PNM  P
0473 07555  301 165630          JSB                  INDBIT    GET MASK
0474 07556  017 100157              PASS L    TAB        L <= MASK.
0475 07557  220 074457          READ       INC  M    P
0476 07560  301 165630          JSB                  INDBIT    GET WORD TO BE TESTED
0477 07561  015 101017              AND  S1   TAB        LOGICAL AND OF MASK, WORD UNDER T
0478 07562  013 041017              XOR  S1   S1         S1 <= 0 IF ALL MASK BITS SET IN W
0479 07563  320 067271          JMP   CNDX TBZ         SKIP       SKIP IF ALL MASK BITS SET IN WORD
0480 07564  000 075717              INC  P    P          SKIP NEXT MACHINE INSTRUCTION.
0481 07565  000 075736  SKIP          RTN  INC  P    P     ADJUST P, JUMP TO FETCH ROUTINE.
0482                    ********************************************************************
0483 07566  220 074717  SBSCBS  READ       INC  PNM  P
0484 07567  301 165630          JSB                  INDBIT    OBTAIN BIT MASK
0485 07570  017 100157              PASS L    TAB        L <= BIT MASK
0486 07571  220 074457          READ       INC  M    P
0487 07572  301 165630          JSB                  INDBIT    OBTAIN WORD TO BE OPERATED ON.
0488 07573  327 170031          JMP   CNDX IR2         CBS        JUMP IF INSTRUCTION IS CBS.
0489 07574  017 001017              IOR  S1   TAB        SET BITS IN WORD,PUT IN S1
0490 07575  000 022461          MPCK       INC  M    M     MEMORY PROTECT CHECK.
0491 07576  177 140017          WRTE       PASS TAB  S1     REWRITE WORD TO MEMORY. RETURN TO
0492 07577  000 075736              RTN  INC  P    P
0493 07600  013 101017  CBS           SANL S1   TAB        S1 <= MEMORY WORD WITH BITS CLEAR
0494 07601  000 022461          MPCK       INC  M    M
0495 07602  177 140017          WRTE       PASS TAB  S1     REWRITE WORD TO MEMORY.
0496 07603  000 075736              RTN  INC  P    P     RETURN TO FETCH ROUTINE.
0497                    ********************************************************************
0498                    *       WORD INSTRUCTIONS
0499                    ********************************************************************
0500 07604  301 165170  CMW       JSB                  INITCM    INITIALIZE
```

```
0501 07605  220 026457          READ      INC  M     A          READ FROM ARRAY A.
0502 07606  017 100157                    PASS L     TAB        L <= WORD FROM ARRAY A
0503 07607  220 024457          READ      INC  M     B          READ ADDRESS IN ARRAY B.
0504 07610  000 024517                    INC  B     B          INCREMENT ARRAY B POINTER.
0505 07611  017 101017                    PASS S1    TAB        S1 := ARRAY B WORD
0506 07612  013 040757                    XOR        S1         TEST THE SIGN BITS
0507 07613  322 076431          JMP  CNDX AL15       DIFNT      ARE THE SIGN BITS DIFFERENT?
0508 07614  003 040757                    SUB        S1         TEST FOR EQUALITY
0509 07615  320 036471          JMP  CNDX TBZ  RJS    CHAL15     JUMP IF UNEQUAL.
0510 07616  000 026557                    INC  A     A          INCREMENT ARRAY A POINTER.
0511 07617  007 145117                    DEC  S3    S3         DECREMENT COUNT.
0512 07620  320 076771          JMP  CNDX TBZ        EXIT       JUMP TO EXIT IF COUNT IS ZERO.
0513 07621  335 030271          JMP  CNDX INT  RJS    CMW+1      JUMP IF NOT INTERRUPTED.
0514 07622  321 176170          JMP                   INTPEND
0515                    ********************************************************************************
0516 07623  301 165170 MVW     JSB                   INITCM     INITIALIZE.
0517 07624  220 026457          READ      INC  M     A          READ FROM ARRAY A
0518 07625  000 026557                    INC  A     A          INCREMENT ARRAY A POINTER.
0519 07626  017 101017                    PASS S1    TAB        S1 <= CONTENTS OF WORD OF ARRAY A
0520 07627  000 024457                    INC  M     B          M <= ADDRESS FROM ARRAY B
0521 07630  017 122761               MPCK PASS       M          MEMORY PROTECT CHECK-- BIT 15 LOW
0522 07631  177 140017          WRTE      PASS TAB   S1         WRITE WORD INTO ARRAY B.
0523 07632  000 024517                    INC  B     B          ADVANCE ARRAY B POINTER.
0524 07633  007 145117                    DEC  S3    S3         DECREMENT COUNT.


0525 07634  320 076771          JMP  CNDX TBZ        EXIT       EXIT IF COUNT IS ZERO.
0526 07635  335 031231          JMP  CNDX INT  RJS    MVW+1      JUMP IF NOT INTERRUPTED.
0527 07636  321 176170          JMP                   INTPEND


0529                    ********************************************************************************
0530                    *         BYTE INSTRUCTIONS
0531                    ********************************************************************************
0532 07637  340 000157 SBT     IMM       HIGH L     %000       L <= 000377B.
0533 07640  015 127017 STBYTE             AND  S1    A          S1 <= RIGHT BYTE OF A REG.
0534 07641  157 125244          LWF  R1   PASS S6    B          S6 <= WORD ADDRESS. FLAG SET IF B
0535 07642  220 052461          READ MPCK INC  M     S6         READ WORD ADDRESS,CHECK FOR MP VI
0536 07643  324 032331          JMP  CNDX FLAG RJS    STEVEN     JUMP IF STORE TO EVEN BYTE.
0537 07644  013 101417                    SANL S9    TAB        MASK OUT EVEN BYTE OF MEMORY WORD
0538 07645  321 172470          JMP                   MERGE
0539 07646  015 101417 STEVEN            AND  S9    TAB        MASK OUT ODD BYTE OF MEMORY WORD.
0540 07647  017 141003               L4   PASS S1    S1         EXCHANGE BYTES IN REGISTER CONTAI
0541 07650  017 141003               L4   PASS S1    S1         BYTE TO BE STORED.
0542 07651  017 160157 MERGE           PASS L     S9         L <= MEMORY WORD WITH TARGET BYTE
0543 07652  017 041017                    IOR  S1    S1         S1 <= WORD WITH BYTES MERGED.
0544 07653  000 024517                    INC  B     B          INCREMENT BYTE ADDRESS.
0545 07654  177 140036          WRTE RTN  PASS TAB   S1         WRITE NEW WORD BACK INTO WORD ADD
0546                    ********************************************************************************
0547 07655  017 125057 LBT             PASS S2    B          S2 <= BYTE ADDRESS.
0548 07656  000 024517                    INC  B     B          INCREMENT BYTE ADDRESS FOR NEXT I
0549 07657  340 000151 LDBYTE  IMM  CLFL HIGH L     %000       L <= 000377B.  CLEAR CPU FLAG.
0550 07660  157 143244          LWF  R1   PASS S6    S2         S6 <= WORD ADDRESS OF BYTE. SET F
0551 07661  220 052457          READ      INC  M     S6         ODD BYTE.  READ WORD ADDRESS.
0552 07662  324 073331          JMP  CNDX FLAG       LODD       JUMP IF BYTE IS ODD.
0553 07663  013 100543               L4   SANL A     TAB        MASK OUT EVEN BYTE AND MOVE ODD B
0554 07664  017 126543               L4   PASS A     A          TO EVEN BYTE OF A REG.
0555 07665  017 136776               RTN                     RETURN TO CALLER OR FETCH.
0556 07666  015 100576 LODD          RTN  AND  A     TAB        MASK OUT EVEN BYTE,LOAD INTO A. R
0557                    ********************************************************************************
0558 07667  301 165170 MBT     JSB                   INITCM     INITIALIZE.
0559 07670  017 127057                    PASS S2    A          S2 <= ADDRESS START OF ARRAY A.
0560 07671  301 172770          JSB                   LDBYTE     LOAD BYTE FROM ARRAY A.
0561 07672  000 043051               CLFL INC  S2    S2         RESET FLAG, S2 <= NEXT BYTE ADDR
0562 07673  301 172030          JSB                   STBYTE     STORE BYTE INTO BYTE ADDRESS IN B
0563 07674  007 145117                    DEC  S3    S3         DECREMENT COUNT
0564 07675  320 034031          JMP  CNDX TBZ  RJS    NOTDAN     JUMP IF COUNT NOT ZERO.
0565 07676  017 142557                    PASS A     S2         A <= 1 + LAST ARRAY A BYTE ADDRES
0566 07677  321 176770          JMP                   EXIT
0567 07700  335 033471 NOTDAN   JMP  CNDX INT  RJS    MBT+2      JUMP IF NOT INTERRUPTED.
0568 07701  017 142557                    PASS A     S2         A <= 1 + LAST ARRAY A ADDRESS MOV
0569 07702  321 176170          JMP                   INTPEND


0571                    ********************************************************************************
0572 07703  340 000157 SFB     IMM       HIGH L     %000       L <= 000377B.
0573 07704  015 127117                    AND  S3    A          S3 <= TEST BYTE IN LOW-ORDER BYTE
0574 07705  013 127143               L4   SANL S4    A          S4 <= TERMINATION BYTE IN
0575 07706  017 147143               L4   PASS S4    S4         LOW-ORDER BYTE
0576 07707  017 127357                    PASS S8    A          S8 <= SAVE ORIGINAL CONTENTS OF A
0577 07710  301 172670 CONTSFB  JSB                   LBT        LOAD BYTE INTO A REG FROM ADDRESS
0578 07711  017 126157                    PASS L     A          L <= BYTE TO TESTED IN LOW BYTE.
0579 07712  017 156557                    PASS A     S8         RESTORE ORIGINAL CONTENTS OF A RE
0580 07713  013 045257                    XOR  S6    S3         COMPARE BYTE TO TEST BYTE.
0581 07714  320 034731          JMP  CNDX TBZ  RJS    NOMATCH    JUMP IF UNEQUAL.
0582 07715  007 124536               RTN  DEC  B     B          B <= BYTE ADDRESS OF MATCH. GO TO
```

```
0583 07716  013 047017  NOMATCH            XOR  S1   S4        COMPARE BYTE TO TERMINATION BYTE.
0584 07717  320 035071           JMP  CNDX TBZ  RJS  INTTST    JUMP IF UNEQUAL.
0585 07720  000 075736                RTN  INC  P    P         SKIP NEXT MACHINE INSTRUCTION AND
0586 07721  335 034431  INTTST   JMP  CNDX INT  RJS  CONTSFB   JUMP IF NOT INTERRUPTED.
0587 07722  007 175736                RTN  DEC  P    P         P <= INSTRUCTION ADDRESS, GO TO F
0588                    ********************************************************************
0589 07723  301 165170  CBT      JSB                 INITCM    INITIALIZE.
0590 07724  017 127357                     PASS S8   A         S8 <= POINTER FOR ARRAY A.
0591 07725  017 157057                     PASS S2   S8        S2 <= NEXT BYTE ADDRESS IN ARRAY
0592 07726  301 172770           JSB                 LDBYTE    LOAD ARRAY A BYTE INTO A REG.
0593 07727  017 127417                     PASS S9   A         S9 <= ARRAY A BYTE.
0594 07730  000 043357                     INC  S8   S2        INCREMENT ARRAY A POINTER.
0595 07731  301 172670           JSB                 LBT       A <= BYTE FROM ARRAY B (ADDRESS I
0596 07732  017 160157                     PASS L    S9        L <= BYTE FROM ARRAY A
0597 07733  003 027017                     SUB  S1   A         SUBTRACT BYTE FROM ARRAY B - A.
0598 07734  320 036371           JMP  CNDX TBZ  RJS  CHAL15B   JUMP IF BYTES NOT EQUAL.
0599 07735  007 145117                     DEC  S3   S3        DECREMENT THE COUNT.
0600 07736  320 036071           JMP  CNDX TBZ  RJS  CONTCBT   JUMP IF COUNT IS NOT ZERO.
0601 07737  017 156557                     PASS A    S8        EQUAL EXIT... A <= 1+LAST MOVED B
0602 07740  321 176770           JMP                 EXIT
0603 07741  335 035271  CONTCBT  JMP  CNDX INT  RJS  CBT+2     JUMP IF NOT INTERRUPTED.
0604 07742  017 156557                     PASS A    S8        A <= NEXT BYTE ADDRESS OF ARRAY A


0606                    ********************************************************************
0607                    *        COMMON ROUTINES TO MOVE, COMPARE INSTRUCTIONS
0608                    ********************************************************************
0609                    *                                                  INTERRUPT EXIT
0610 07743  000 050457  INTPEND            INC  M    S5        M <= ADDRESS OF WORD 3
0611 07744  177 144017           WRTE      PASS TAB  S3        WRITE REMAINING COUNT INTO WORD 3
0612 07745  007 175717                     DEC  P    P
0613 07746  007 175736                RTN  DEC  P    P         P <= ADDRESS OF WORD 1, GO TO FET
0614                    ********************************************************************
0615                    *                                                  EXIT TESTS FOR CBT,CMW
0616 07747  007 156557  CHAL15B            DEC  A    S8        A <= BYTE ADDRESS OF MISMATCH.
0617 07750  017 141017  DIFNT              PASS S1   S1        CHECK RESULT OF COMPARE
0618 07751  322 036571  CHAL15   JMP  CNDX AL15 RJS  SKIP1     JUMP IF SIGN BIT IS ZERO.
0619 07752  000 047157                     INC  S4   S4        SKIP ONE MACHINE INSTRUCTION.
0620 07753  000 047157  SKIP1              INC  S4   S4        SKIP ONE MACHINE INSTRUCTION.
0621 07754  007 145117                     DEC  S3   S3        DECREMENT THE COUNT.
0622 07755  017 144157                     PASS L    S3        L <= COUNT REMAINING
0623 07756  004 124517                     ADD  B    B         B <= FIRST ADDR. IN ARRAY B + COU
0624                    ********************************************************************
0625                    *                                                  COMPLETION EXIT
0626 07757  000 050457  EXIT               INC  M    S5        M <= ADDRESS OF WORD 3
0627 07760  177 154017           WRTE      PASS TAB  S7        WORD 3 <= ZERO.
0628 07761  017 147736                RTN  PASS P    S4        P <= NEXT MACHINE INSTR. TO EXECU
0629                    ********************************************************************
0630                    *     JUMP INSTRUCTIONS
0631                    ********************************************************************
0632 07762  220 074717  JLY      READ      INC  PNM  P         READ ADDRESS OF WORD 2.
0633 07763  301 165630           JSB                 INDBIT    CHECK FOR INDIRECT,M<= DESTINATIO
0634 07764  340 120417           IMM       HIGH IR   %050      MACHINE JMP INTO IR TO SET LOW MP
0635 07765  017 122461                MPCK PASS M    M         DO MP CHECK ON JUMP TARGET ADDRES
0636 07766  017 175657                     PASS Y    P         Y <= ADDRESS OF FOLLOWING MACHINE
0637 07767  017 123736                RTN  PASS P    M         P <= DESTINATION ADDRESS, JUMP TO
0638                    ********************************************************************
0639 07770  220 074717  JPY      READ      INC  PNM  P         READ ADDRESS OF WORD 2.
0640 07771  017 172157                     PASS L    Y         L <= INDEX REG. Y.
0641 07772  004 101017                     ADD  S1   TAB       S1 <= INDEXED JUMP ADDRESS.
0642 07773  340 120417           IMM       HIGH IR   %050      MACHINE JMP INTO IR TO SET LOW MP
0643 07774  017 140457                     PASS M    S1        M<= INDEXED ADDRESS, WITH BIT 15
0644 07775  017 122761                MPCK PASS      M         MP CHECK ON 15-BIT DESTINATION AD
0645 07776  017 123736                RTN  PASS P    M         P <= DESTINATION ADDRESS. GO TO F
0646                    ********************************************************************
0647                    END
```