INTERDATA

MODEL 80

FUNCTIONAL SPECIFICATION

Revision 0 August 2, 1971
Revision 1 September 14, 1971

TABLE OF CONTENTS

6.0 MEMORY SYSTEM

6.1 Introduction
6.2 System Main Memory
6.3 Read Only Control Memory
6.4 Dynamic Control Store Memory
6.5 Special Configuration

7.1 Configuration and Power
7.2 Basic Processor Chassis
7.3 Special Maintenance Considerations
7.4 Power Supplies

8.1 Maintenance Facilities

9.1 Peripheral Device Controllers

1.0 INTRODUCTION

The Model 80 system is a high performance member of the INTERDATA family of computers with performance 3 to 6 times that of a Model 70. It is strictly compatible with the Model 5 and 70 systems from both hardware and software points of view.

All software generated for the Model 5 and 70 systems will run on this system.

All peripheral devices, 10" or 15"/7" designs, will interface to this system.

The packaging is the same as that of the Model 70. 15" card files, 8 cards per file.

The expansion chassis used for I/O will be identical to that used in the Model 70.

The specifications and features of the system are breifly described below and will be covered in detail in the following chapters.

1.1 Packaging

The Model 70 packaging and display panel will be used. The Model 80 with up to 64K of core will mount in one card file.

1.2 Processor

The Processor consists of a double card assembly CPU/ALU and an I/O card which generates the I/O bus and contains the display controller and TTY controller. The CPU employees a 32 bit instruction word and 16 bit bussing. The basic clock rate is 10 MHz. User level instruction execution times will be typically 450 ns for RR and RS and 900 ns for RX. The basic instruction set will be contained in 768 words of ROM.

1.3 Main Memory

The main memory uses MOS/LSI devices for storage. It has an effective cycle time of approximately 220 ns. Memory banks can be accessed via the Processor or through an external DMA port. The external port has a through-put of approximately 9 megabytes/sec. or may be used as a multiplexed port by up to four devices at a maximum through-put of 5.2 megabytes/sec. The packaging consists of a single 15" memory controller and one-four storage modules of 16K bytes each, cabled directly to the controller. The DMA connections is via the back panel. The DMA port will accomodate the Model 70 selector channel in the multiplexed configuration.

1.4 I/O System

The I/O system is identical to Model 70 and will use the same packaging. The basic processor with minimum memory will accomodate three 15" I/O cards, in addition to the TTY and display.

1.5 Expansion Control Store

The basic Model 80 uses 768 words of ROM control store which is mounted on the CPU/ALU board. Room is provided to mount an additional 256 words of ROM on this board. The control store may be expanded by another 3K words of either dynamic control store (DCS) or ROM by adding additional circuit boards. Two new user instructions are added for manipulating the DCS.

1.6 Power Supply

The volatile semiconductor Main Memory of the Model 80 will be powered by a separate supply that includes built-in battery backup for short term power interrupts and facilities for low power stand-by operation of the memory when the system is shut down. The internal battery will be capable of holding the memory's contents, in the low power non-operating mode, for several minutes. For long term memory protection, the customer will be required to supply 115/230 VAC (±20% @ 47-63Hz). Readily available DC to AC inverters can be used. (The 115/230 volts is required to power fans in the power fail situation.)

The memory will go into the low power stand-by mode whenever the processor is shut down. It will not use the battery or the reliable AC source, however, except on a primary power failure. The Memory Power will not be controlled by the front panel power switch.

The Model 80 Processor, Control Memory, and I/O will use any of the Power Supplies available with the Model 70 system.

2.0 SUMMARY OF CHARACTERISTICS

The speed and capability of the Model 80 can be best outlined by a summary of its characteristics.

2.1 User Memory

- low power MOS semiconductor LSI design
- 250 ns access by CPU
- individual ports for CPU and DMA
- expandable in 16KB increments to 64KB
- multi-bank capability
- built-in instruction look-ahead for CPU
- built in emergency 'keep-alive' power supply and auxiliary power switching

2.2 Control Memory

- bipolar 50 ns semiconductor ROM
- directly addressable to 4K instruction
- dynamic control store memory available on some models
- data storage in dynamic control store memory permits vary fast matrix manipulations, function generation, etc.
- asynchronous timing allows dynamic control store memory to be any speed

2.3 User Instructions

The Model 80 is upward compatible from the Models 3, 4, 5 and 70 at the user instruction level. Specifically, it will implement the same instructions as those of the Model 70 plus those necessary to manipluate the dynamic control store in the micro-level computer on those versions so equipped.

Although the timing of the individual instructions vary somewhat, the following lists the typical speeds for different classes of user instructions. A complete list is given in the appendix.

	GENERAL	MULTIPLY/DIVIDE	SHIFT/ROTATE	BRANCH	FLOAT ADD	FLOAT MULTIPLY
SF	450		450-1900	450-1450		
RR	450	2100		450-850		
RS	450		450-1900	450-850		
RX	900	2500			13.6[1]	19.1

FIGURE 2.1 USER INSTRUCTION TIMING (NANOSECONDS)

2.4 Micro Instructions

The micro instruction word is 32 bits long. In addition to the branch and write instructions, there are three types of instructions to the internal modules. These minimally encoded instructions provide 112 combinations of module/function commands. The micro-instruction can simultaneously direct two operands and a result independently on three of the computer's busses, generate 12 bit immediate field operands, select the address of the next micro instruction, and perform non-encoded micro control of the computers functions such as reading/writing main memory, incrementing user location and memory address registers, controlling the user status register, and decoding the next user instruction.

---

1. micro-seconds

2.5 Internal Modules

The Model 80 hardware architecture will accommodate eight internal modules which communicate over three of the Processor busses. In theory, the function of all of the modules (with the exception of the control module) is arbitrary, and the significance of the various instructions take meaning only when applied to a specific module. The basic Model 80 is composed of a control module(which controls the user memory, control memory, register gating and instruction sequencing) an arithmetic/logic module(capable of 20 operations including hardware multiply/divide and multiple position shifting),and an I/O module(which connects the internal busses of the Processor to the standard INTERDATA Multiplexor (I/O) bus.)

The architecture will accommodate five additional arbitrary modules such as floating point, boolean manipulors, or special nature designs.

2.6 Peripherals

The Model 80 will interface to, and is compatible with, all standard INTERDATA peripheral controllers and controllers designed to the standard INTERDATA Multiplexor bus. Any number of devices up to 255 can be accommodated, but a maximum of 16 can be interfaced directly to the Multiplexor Bus or the Selector Channel's bus (based on the new interface standards).

2.7 Interrupts

The Model 80 has eight hardware priority interrupts, most of which can be masked by various bits of the PSW. The occurrence of a recognized interrupt causes the micro-program to 'trap' to one of eight specific control store locations associated with the interrupts.

2.8 Registers

The Model 80 has 16 general registers, 15 of which may be used as index registers. In addition, there are eight additional general purpose registers available to the microprogrammer, plus 5 registers associated with the user level machine control that are available to the micro-programmer.

2.9 Processor Timing

Communications between modules is request/response. Timing is completely asynchronous (rather than quantized) to achieve maximum speeds. Additionally, interlocks are provided between the control memories and the CPU to facilitate programming the micro machine. The control module operates on a 10MHz clock, allowing a minimum instruction execution in 200 ns. Internal timing within the other modules can be selected to best suit the needs of the module.

4.0 FUNCTIONAL DESCRIPTION OF THE BASIC CPU

The functional characteristics of the CPU can best be described in terms of its registers, busses and related gating. (Refer to Figure 4.1.) There are four busses which are key to the modular design philosophy of the architecture. An understanding of the bus tructure is necessary to determine how each module of the processor interrelates and further how the registers and gating of each module contributes to the function which the module is designed to serve.

4.1 A, B, S and Control Busses

The control bus of the computer is commanded by the control module and in essence is a reflection of that segment of the micro instruction selecting the function and module to be addressed, plus timing to effect transfers of data. Lumped into this also is a means for a module to transfer data to the condition code of the PSW. Following is a description of the control bus signals (shown individually on igure 4.1).

a. Module Select Lines (MDSEL [0:2]) - These three lines contain the address of a module for which the current micro-instruction is intended. One of the eight arbitrary modules can be selected by the instruction to perform some function. These three lines reflect bits (0:2) of the micro-instruction.

b. Function Select Lines (FSEL [0:3]) - These four lines reflect bits 16:19 of the micro-instruction and normally select one of sixteen arbitrary functions to be performed by the selected module.

c. Start (STRT) - STRT signals the internal modules that data is valid on the busses. It is in effect a request from the control module for a response to a micro-instruction. Data is held static on the A & B busses while STRT is active. The control module holds STRT active until it recognizes a response from the module (MFIN) and has stored the results presented on the S Bus.

d. Module Finished (MFIN) - MFIN is a response to the control module from a selected module indicating that it has recognized STRT and completed the selected function. The selected module gates data and other responses onto the S bus prior to returning MFIN. Data and responses must be held on the busses until the control removes STRT. This time is indefinite and depends on events within the control module.

e. Module Signal (MSIG) - This is a control signal manipulated by the selected module to indicate some arbitrary condition to the control module. It is tested by the control module during a normal micro-instruction to the selected device to control a conditional branch in the micro-program.

f. Condition Code Bus (SCC, VCC, CCC, GCC, LCC)
SCC signals the control module that the selected module wishes to manipulate the condition code of the program status word. If the micro-programmer has enabled this manipulation, the condition code is forced to a status specified by the selected module. The status is unconditionally forced into the CPU flags. This is done concurrent with a normal instruction to the selected module.

g. VCC, CCC, GCC, LCC - These lines specify the status forced into the condition code of the program status word and represent overflow, carry, greater than, and less than, respectively.

4.2 A (0:15) B (0:15) S (0:15)

The A, B and S busses are the primary data links between the control and the selected module. Gating of data to/from each of these busses is controlled by the micro-instruction. Almost all the registers of the control module can be gated to/from these busses.

Data is selected by the micro-instruction from two independent sources and transmitted to a selected module over the A & B busses. The module is thus presented simultaneously with two operands. The resulting data is returned to the control module via the S bus. The destination of the S bus is selected by the micro-instruction.

4.3 Typical Bus Exchange

The use of these busses can be summarized by an example.

1. The micro-instruction selects a module (MDSEL 0:2) and directs it to perform some function (FSEL0:3).

2. The operands are selected from somewhere in the control module and gated onto the A and B busses.

3. The control module informs the selected module that all data if valid on the busses and that it may begin (STRT).

4. The selected module performs the function (S) = (A) F (B) and gates the results to the S bus.

5. The selected module may manipulate the condition code via SCC-VCC-GCC-LCC-CCC.

6. The selected module may signal that the result of the operation was zero (for example) by activating MSIG.

7. The selected module activates MFIN to signal the CPU module that the operation is complete and the results are presented on the S bus.

8. The control module recognizes MFIN, gates the S bus to the destination specified by the micro-instruction, and then removes STRT.

9. The selected module deactivates itself when STRT is removed.

4.4 Control Module (CU)

4.4.1 Registers. The following registers are part of the control module although some may reside physically within the ALU.

a. A Stack, B Stack: The A stack and B stack are a redundant set of register banks containing the twenty-four general purpose registers of the CPU. The registers are duplicated to allow simultaneous gating of any register in the stack onto either the A bus or the B bus. Sixteen of these registers are the user general registers; while the remaining eight are available to the micro-programmer as desired. These registers are gated onto the A and B bus and loaded from the S bus under control of the micro-instruction.

b. Memory Data Register (MDR): This register provides the data buffer between the CPU and the user level memory. The MDR can be gated onto the A and B busses and loaded from the S bus under control of the micro-instruction. It is, of course, also loaded under control of the memory when a memory read cycle is requested. Hardware interlocks are employed to synchronize the memory to the CPU.

c. Memory Location Register (MLC): The MLC is a general purpose register which can be gated to either the A or B bus and loaded from the S bus, but posesses additional characteristics intended to facilitate the emulation of the user level repertoire. The register can, under micro-control, be incremented by two, incremented by four, and jammed into the memory address register. This register keeps track of the current instruction location of the emulated machine.

d. Memory Address Register (MAR): This register contains the address of user memory the micro-programmer is reading or writing. It can be loaded from the S bus under control of the micro-instruction, or loaded from the MLC, or incremented by two, or incremented by four under micro-control. The least significant bit of the MAR is used to control byte steering for the byte-oriented instructions of the user repertoire (refer to Section 4.6). As in the MDR, timing conflicts are resolved by hardware interlocks.

e. Program Status Word (PSW): The program status word is a 16-bit register which may be gated onto either the A or B bus and loaded from the S bus under control of the micro-instruction. Various bits of the PSW are used to enable associated hardware interrupts. PSW (12:15) contain the condition code of the user level computer. These bits may be compared and tested against corresponding bits of the user instruction under Module 0 micro-instructions to effect emulation of user branch instructions. Additionally, they can be manipulated by any module designed to do so, if they are enabled by the micro-programmer.

f. User Destination Register, User Source Register (UDR,USR): These two control registers store bits (9:11) and (12:15), respectively, of the current user level instruction being emulated, and allow the micro-programmer to indirectly reference the general registers selected by the user instruction. The UDR is compared to the PSW condition code on

certain micro-instructions to emulate user level branches. Both registers can be examined by gating them onto the A bus under micro-instruction control. The UDR can also be loaded from the S bus.

g. User Instruction Register (UIR), Memory User Destination Register (MUDR) and Memory User Source Register (MUSR). These three registers are loaded with bits (0:7), (8:11), and (12:15), respectively, of the next user level instruction to be emulated. The eight bit opcode stored in the UIR is used to vector to the emulation sequence for the next user instruction. It is also used to interrogate a ROM which has been configured to decode privileged and illegal user level instructions. The contents of the MUDR and MUSR are transferred to the UDR and USR at the beginning of the next emulation.

h. ROM Location Register (RLR): This register stores the current address of the control store instruction. It is loaded from the ROM address gates (RAG) at the beginning of every instruction except interrupt trap instructions and 'execute' type instructions (explained in chapter on micro-programming). The RLR is a (12) bit register allowing direct addressing of control store up to 4K instructions.

i. ROM Instruction Register (RIR): This (32) bit register stores the current micro-instruction. The RIR is the focus of control of the CPU.

j. A Latch, B Latch: The A latch and B latch registers, as their name implies, latch the data presented by the control module to be gated onto the A and B busses. They are the data sources for these busses.

4.4.2 Interrupts

The hardware of the computer provides eight priority interrupts. Each interrupt has a unique 'trap' location associated with it. Recognition of an interrupt causes the instruction stored at its respective trap location to be performed. The RLR contents are preserved to allow the address of the interrupted sequence to be saved if desired so that control can be returned at the completion of the interrupt routine. Certain interrupts are enabled/disabled by bits of the PSW as shown in Figure 4.2. Additionally, all interrupts can be enabled/disabled as a group by micro-instruction. All interrupts not masked by PSW bits are interrogated when a new user level instruction is decoded, regardless of the status of the group enable. The group enable is automatically disabled at the beginning of a user emulation, and must be enabled by instruction if the programmer wishes to recognize the group enabled interrupts within an emulation sequence. The following table lists the pertinent information for each interrupt. The external interrupts 1, 2 and 3 will cause an EXTERNAL INTERRUPT.

INTERRUPT	TRAP	MASK
Primary Power Fail	06_{16}	NONE
Machine Malfunction[1]	05_{16}	PSW02
Display	04_{16}	NONE
Ext. Interrupt	03_{16}	PSW01 (DATA CHANNEL)
Ext. Interrupt	02_{16}	PSW01 (I/O)
Ext. Interrupt	01_{16}	PSW10
Ext. Interrupt	00_{16}	PSW11
Illegal Instruction	07_{16}	NONE
Privileged Instruction	07_{16}	PSW07

1. This interrupt can have several causes and is discussed further in I/O section.

4.4.3 Control Store Memory

The Model 80 can accomodate a maximum of 4K x 32 bits of control store memory. The computer allows data as well as instruction to be retrieved from its control memory. This capability expands its versatility by allowing data such as sine tables, translation tables and matrices to be stored and operated upon efficiently by the micro-programmer.

On models so equipped, the processor can alter its control store (write into its memory). This capability to store and retrieve data provides the power of a hardware computer at micro-instruction speeds.

ROM and dynamic control store can be mixed in 1K by 32 bit blocks.

4.4.4 Module Communications

Communication between the control and the internal modules is asynchronous. The internal logic of the CPU operates on a 10 MHz clock. This determines the fastest communication rate between modules.

A minimum of 2 clock periods (200ns) is necessary to complete an instruction, but the interchange time is dependent upon the external module and the length of time necessary for it to respond. The Control Module's clock is gated rather than free running to allow complete asynchronism between modules.

Module intercommunications is over a standard back panel allowing complete interchangability of card slots. Figure 4.3 illustrates the back panel links between modules.

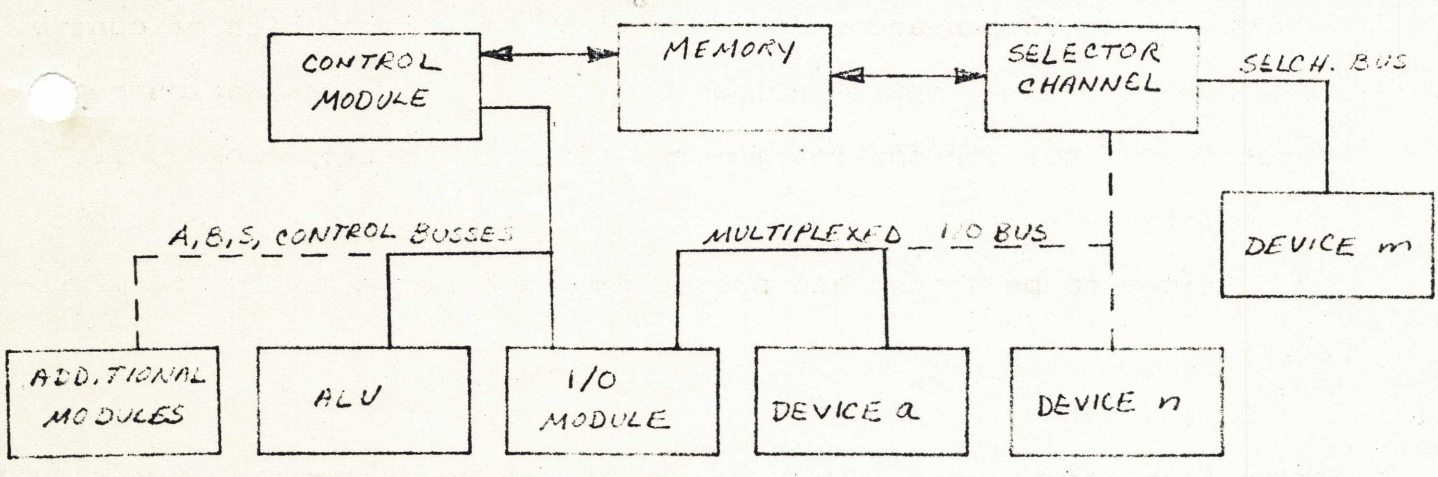


FIGURE 4.3

4.5 Arithmetic Module (ALU)

The ALU (arithmetic-logic unit) is a standard module in the Model 80. It is addressed as Module 1 and it can perform 20 functions.

By design, the ALU is never busy and for the majority of ALU functions, response is within 100 ns. (This allows most ALU instructions to be completed in 200ns.)

For functions which require more than one ALU cycle (i.e., shifts, rotates, multiply-divide), the ALU does not respond with a finish signal (MFIN) until the final results are on the S bus.

Multiply-divide and extended shift operations can be performed only on the 24 general registers and must address the same register pair on both the A & S bus. The same restrictions that apply to these operations on the Mod 5 must be observed in the Model 80.

A user emulated multiply-divide instruction is micro-coded by: selecting the ALU (module 1), addressing the UDR on the S bus, the UDRP1 (UDR plus 1) onto the A Bus, the USR onto the B Bus, and the required function code for the operation. When the ALU signals finish, the results have already been deposited in the UDR.

To implement a shift or rotate instruction, the register to be manipulated is addressed onto the A bus, the shift count is put onto the B bus, and the destination register is loaded from the S bus.

Extended shifts must address the UDRP1 (or equivalent) onto the A bus. The ALU performs its function as defined by the F field of the instruction:

F FIELD	OPERATION
0000	subtract
0001	add
0010	*subtract with carry (employs CPU flags)
0011	*add with carry (employs CPU flags)
0100	not used
0101	logical and
0110	logical exclusive or
0111	logical or
1000	*logical shift right
1001	*logical shift left
1010	*rotate right
1001	*rotate left
1100	*arithmetic shift right
1101	*arithmetic shift left
1110	*signed multiply
1111	signed divide

* When used in conjunction with the "K" bit of the RRC instruction, shifts are double precision, multiply is unsigned and the C, G and L of the PSW are used for add and subtract with carry.

The ALU employs its own gated asynchronous clock to perform the complex functions (shifts, multiply, divide). This is 10 MHz clock enabled by FSELO. The ALU is also allowed to manipulate the A and B buses, a priviledge reserved for it and the control module. The ST bus is generated on the ALU, which is the OR function of the S bus of the computer and the output of the ALU. The ST bus is the "true" source of all data to the registers designated as the S bus destination.

The ALU also manipulates the UDRP1 register for extended shift and multiply/divide instructions.

4.6 I/O Interface Board

The I/O Interface Board contains the I/O Module, the Display Controller and the TTY Controller. The I/O Module generates the Multiplexor Channel D-Bus from the CPU buses (A, B, S, CC, etc.) in addition to performing byte manipulation for the CPU and control of the Machine Control Register (MCR).

Controllers for the Display and TTY, located on the I/O Board, have access to the CPU via the Multiplexor Channel D-Bus.

4.6.1 Display Controller

The Display Controller makes use of the standard Model 70 Display Panel and provides the same operational features as the Model 70 Controller.

The Controller shares D-Bus drivers/receivers with the TTY Controller, and interrupts the CPU over the DSPLYO line provided.

4.6.2 TTY Controller

The TTY Controller connects the Model 33/35 teletype to the Model 80 over the Multiplexor Channel. The Controller provides the serial/parallel conversion and all standard TTY control features in addition to a full character buffer in the receive mode. This allows program service in one character time (100 ms) instead of a bit time (9 ms).

4.6.3 I/O Module (See Figure 4.2)

The I/O Module performs a multiplicity of functions. The main function is to generate a Mod 70 compatible Multiplexor Channel D-Bus from the CPU buses whenever it is addressed by module number 2 and the proper Function Selection is made. The module also performs byte manipulation for the CPU both in conjunction with an I/O operation and without. Finally, the I/O Module contains the Machine Control Register (MCR), which stores machine malfunction conditions, in addition to circuits for primary power fail detection, module STRT time out check and the generation of the Machine Malfunction (MMF) and Primary Power Fail (PPF) interrupts used by the CPU. The system initialize relay, which grounds the SCLRO line, is also located in the I/C Module.

The four Function Select lines together with the KSIG line (see Fig. 5.1 for "K" bit) pick one of 32 possible functions, shown in the following table. The D-Bus functions may be performed with or without an address cycle depending upon the state of FSEL00. KSIG is used to specify register type operations or, in the case of the halfword functions (RDH/WDH), to specify the non D-Bus operations store block (STB) and load block (LB).

FSELOX

	0 1 2 3		KSIG=0		KSIG=1
0	0 0 0 0		RD		RDR
1	0 0 0 1		WD		WDR
2	0 0 1 0		SS		SSR
3	0 0 1 1		OC		OCR
4	0 1 0 0		RDH	*	STBR
5	0 1 0 1		WDH	*	LBR
6	0 1 1 0				ACK
7	0 1 1 1	*	SMCR	*	CMCR
8	1 0 0 0		RDA		RDRA
9	1 0 0 1		WDA		WDRA
A	1 0 1 0		SSA		SSRA
B	1 0 1 1		OCA		OCRA
C	1 1 0 0		RDHA	*	STB
D	1 1 0 1		WDHA	*	LB
E	1 1 1 0	*	EXB		DACK
F	1 1 1 1	*	POW		

* I/O INTERFACE OPERATION WITHOUT USE OF D-BUS

I/O Function Mnemonics are listed on the following table.

MNEMONIC		MODULE FUNCTION	COMMENT
RD	*	Read Data	byte/indexed type
WD	*	Write Data	byte/indexed type
OC	*	Output Command	byte/indexed type
SS	*	Sense Status	byte/indexed type
RDR	*	Read Data	byte/register type
WDR	*	Write Data	byte/register type
OCR	*	Output Command	byte/register type
SSR	*	Sense Status	byte/register type
ACK		Acknowledge Interrupt	byte/register type
RDH	*	Read Data Halfword	2 data cycles for byte controllers
WDH	*	Write Data Halfword	2 data cycles for byte controllers
DACK		Data Channel Acknowledge	Includes an automatic RDH
(*)RDA		Read Data	Same as RD but preceded by a device address cycle
etc.		etc.	
STB		Store Byte	halfword/indexed type
LB		Load Byte	byte/indexed type
STBR		Store Byte	halfword/register type
LBR		Load Byte	byte/register type
SMCR		Sense Machine Control Reg.	
CMCR		Clear Machine Control Reg.	
EXB		Exchange B-Bytes	B to S with byte exchange
POW		Release Initialize Relay	

4.6.3.1 When the I/O Module is addressed and given an I/O function code, it creates a one, two or three cycle I/O operation. The device address on A(08:15) is gated to D(08:15) together with the ADRS control line whenever the address type of functions are specified. The returned SYN signal terminates the address cycle and the module sends the next control function (DA, DR, CMD or SR).

The halfword functions (RDH and WDH) will have a single data cycle when the Halfword (HW) test line is active and two data cycles when a byte oriented controller is addressed.

Output data is gated from the B-Bus to the D-Bus and the input data is gated from the D-Bus to the Data Register and then to the S-Bus.

At the end of the I/O operation, the I/O Module returns a MFIN signal to restart the CPU clock. The active state on the DCR and MSIG lines indicates that the data channel controller requests a memory read with the data readout sent to the channel on a subsequent WDH operation of the I/O Module. Should DCR not be active, the next step is a RDH operation of the I/O Module. At the end of the DACK operation, a control flip-flop is set which forces the HW test condition in the I/O Module to the active state. This permits proper execution of the following halfword function (RDH or WDH). This control flip-flop is cleared at the end of the halfword operation (or any other D-Bus operation that might have been given following the DACK).

An address cycle skip option is available which deletes the specified address cycle when the device specified is still addressed. A copy of the most recent device address is stored in the I/O Module to be checked against the next I/O address. Should they match, the device controller is still addressed and the I/O Module goes directly to the data cycle. For a mismatch, the ADRS cycle is generated and the address register is updated. The address register is cleared by the SCLRO or the DACK operation. It should be noted that this option does not produce the same sequence used in the Mod 5 and Mod 70. Since some device controllers may require special care in programming if this option is used, it is normally disabled, but is available to the user who wishes to utilize it.

4.6.3.2 A block diagram of the I/O Module is shown on Figure 4.2. The control and sequence circuit checks the module address and function code to set up the proper gating between buses. The multiplexors onto the S Bus are used for I/O operation as well as the byte manipulation between the A and B buses and the S Bus.

The PPF Detector checks the 12 VAC input and sets the early PPF (EPF) bit (15) in the MCR. After a suitable delay, the PPF interrupt is sent to the CPU.

The 35 us STRT timer is initiated whenever the module STRT signal leaves the CPU, and is cleared by the MDFIN signal from the module addressed. Should the timer time out before the MDFIN signal arrives, a malfunction exists; i.e., non-existant module, circuit trouble or no SYN signal from the D Bus. For an I/O operation, the false sync. code (0100) is placed on the CC bus and a false MFIN signal is generated to restart the CPU clock. With a module address other than the I/O module, the STF bit (10) in the MCR is set, the MMF interrupt is generated and a false MFIN is sent to restart the CPU clock,

The Machine Control Register (MCR) bit assignments are as follows:

MCR15	EPF	early primary power fail
MCR14	IPF	instruction parity fail (user memory)
MCR13	DPF	data parity fail (user memory)
MCR12	MMF	set when any other MCR bit is set
MCR11	IA	illegal address (user memory)
MCR10	STF	STRT Time fail
MCR09		spare
MCR08		spare

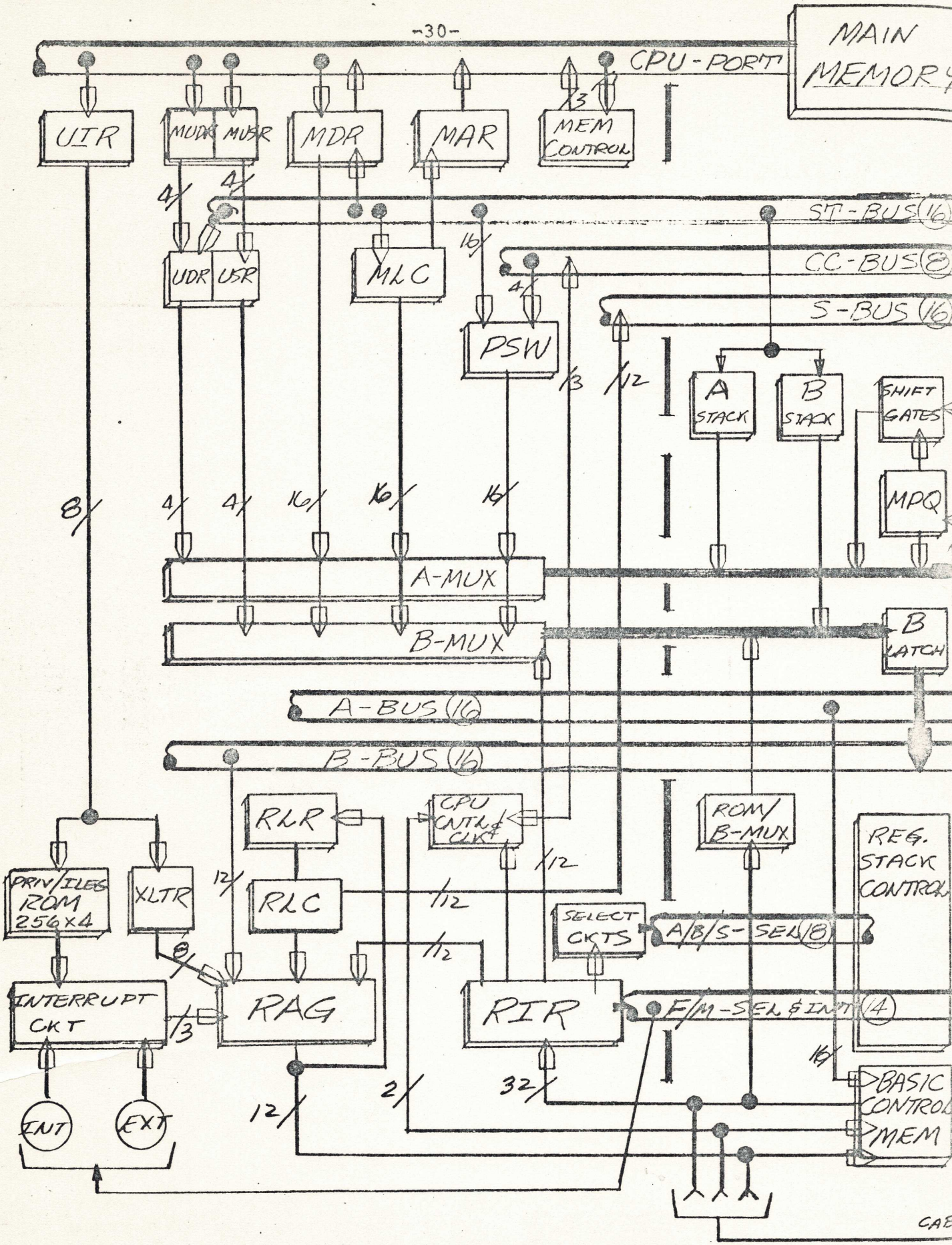
The Sense MCR function code gates MCR (12:15) to the CC-Bus and MCR (08:15) to the S Bus. Another function code will clear the MCR.

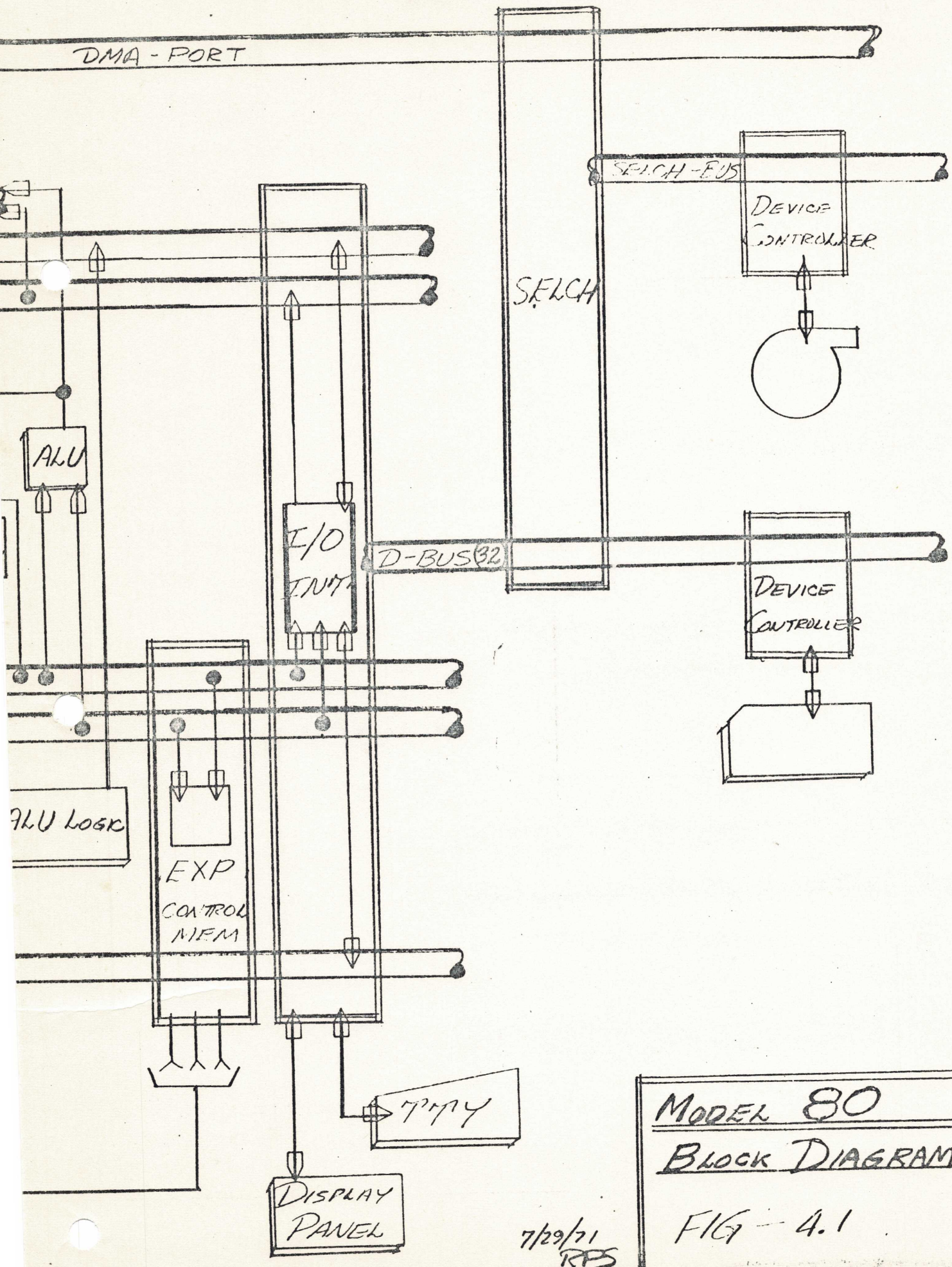
I/O gating is summarized on the following table:

I/O Module Function Gating

Hex Equiv. of FSEL	Function	CSIG	MA15	HW	MA15 = 0 OR HW = 0 DEST. (H)	MA15 = 0 OR HW = 0 DEST. (L)	MA15 = 1 OR HW = 1 DEST. (H)	MA15 = 1 OR HW = 1 DEST. (L)	Other
0	RD	0	0/1	X	D(L) → S(H)	B(L) → S(L)	B(H) → S(H)	D(L) → S(L)	
	RDR	1	X	X	ZERO → S(H)	D(L) → S(L)	NA	NA	
1	WD	0	0/1	X	ZERO → D(H)	B(H) → D(L)	B(H) → D(H)	B(L) → D(L)	
	WDR	1	X	X	B(H) → D(H)	B(L) → D(L)	NA	NA	
2	SS	0	0/1	X	D(L) → S(H)	B(L) → S(L)	B(H) → S(H)	D(L) → S(L)	D(12:15) TO CC
	SSR	1	X	X	ZERO → S(H)	D(L) → S(L)	NA	NA	
3	OC	0	0/1	X	ZERO → D(H)	B(H) → D(L)	B(H) → D(H)	B(L) → D(L)	
	OCR	1	X	X	B(H) → D(H)	B(L) → D(L)	NA	NA	
4	* RDH	0	X	0/1	D(L)1 → S(H)	D(L)2 → S(L)	D(H) → S(H)	D(L) → S(L)	
	STBR	1	X	X	B(H) → S(H)	A(L) → S(L)	NA	NA	
5	* WDH	0	X	0/1	B(H) → D(L)1	B(L) → D(L)2	B(H) → D(H)	B(L) → D(L)	
	LBR	1	X	X	ZERO → S(H)	A(L) → S(L)	NA	NA	
6		0							
	ACK	1	X	X	ZERO → S(H)	D(L) → S(L)	NA	NA	
7	SMCR	0	X	X	MCR(H) → S(H)	MCR(L) → S(L)	NA	NA	MCR(12:15) TO CC
	CMCR	1	X	X	- CLEAR MCR - NO BUS GATING -				
8	RDA	0	0/1	X	SAME AS RD & RDR BY ADDRESS CYCLE		BUT PRECEDED [A(L) → D(L)]		
	RDRA	1	X	X					
9	WDA	0	0/1	X	SAME AS WD & WDR		BUT WITH ADRS CYCE		
	WDRA	1	X	X					
A	SSA	0	0/1	X	SAME AS SS & SSR		BUT WITH ADRS CYCLE		
	SSRA	1	X	X					
B	OCA	0	0/1	X	SAME AS OC & OCR		BUT WITH ADRS CYCLE		
	OCRA	1	X	X					
C	* RDHA	0	X	0/1	SAME AS RDH		BUT WITH ADRS CYCLE		
	STB	1	0/1	X	A(L) → S(H)	B(L) → S(L)	B(H) → S(H)	A(L) → S(L)	
D	* WDHA	0	X	0/1	SAME AS WDH		BUT WITH ADRS CYCLE		
	LB	1	0/1	X	ZERO → S(H)	A(H) → S(L)	ZERO → S(H)	A(L) → S(L)	
E	EXB	0	X	X	B(L) → S(H)	B(H) → S(L)	NA	NA	
	DACK	1	X	X	ADRS & DACK,	ZERO → D(L)	D(H) → S(H)	D(L) → S(L)	AUTO RDH
F	POW	0	X	X	- RELEASE SCLRQ RELAY -				
		1							

⊛ FOR RDH, WDH, RDHA & WDHA; TWO DATA CYCLES WILL BE GENERATED FOR BYTE CONTROLLERS (IE. HW=0).


MAIN MEMORY
CPU-PORT
UIR
MUDR
MUSR
MDR
MAR
MEM CONTROL
ST-BUS (16)
CC-BUS (8)
S-BUS (16)
UDR
USR
MLC
PSW
A STACK
B STACK
SHIFT GATES
MPQ
A-MUX
B-MUX
B LATCH
A-BUS (16)
B-BUS (16)
RLR
CPU CNTL & CLK
ROM/ B-MUX
REG. STACK CONTROL
PRIV/ILEG ROM 256x4
XLTR
RLC
SELECT CKTS
A/B/S-SEL (8)
INTERRUPT CKT
RAG
RIR
E/M-SEL & INT (4)
BASIC CONTROL MEM
INT
EXT

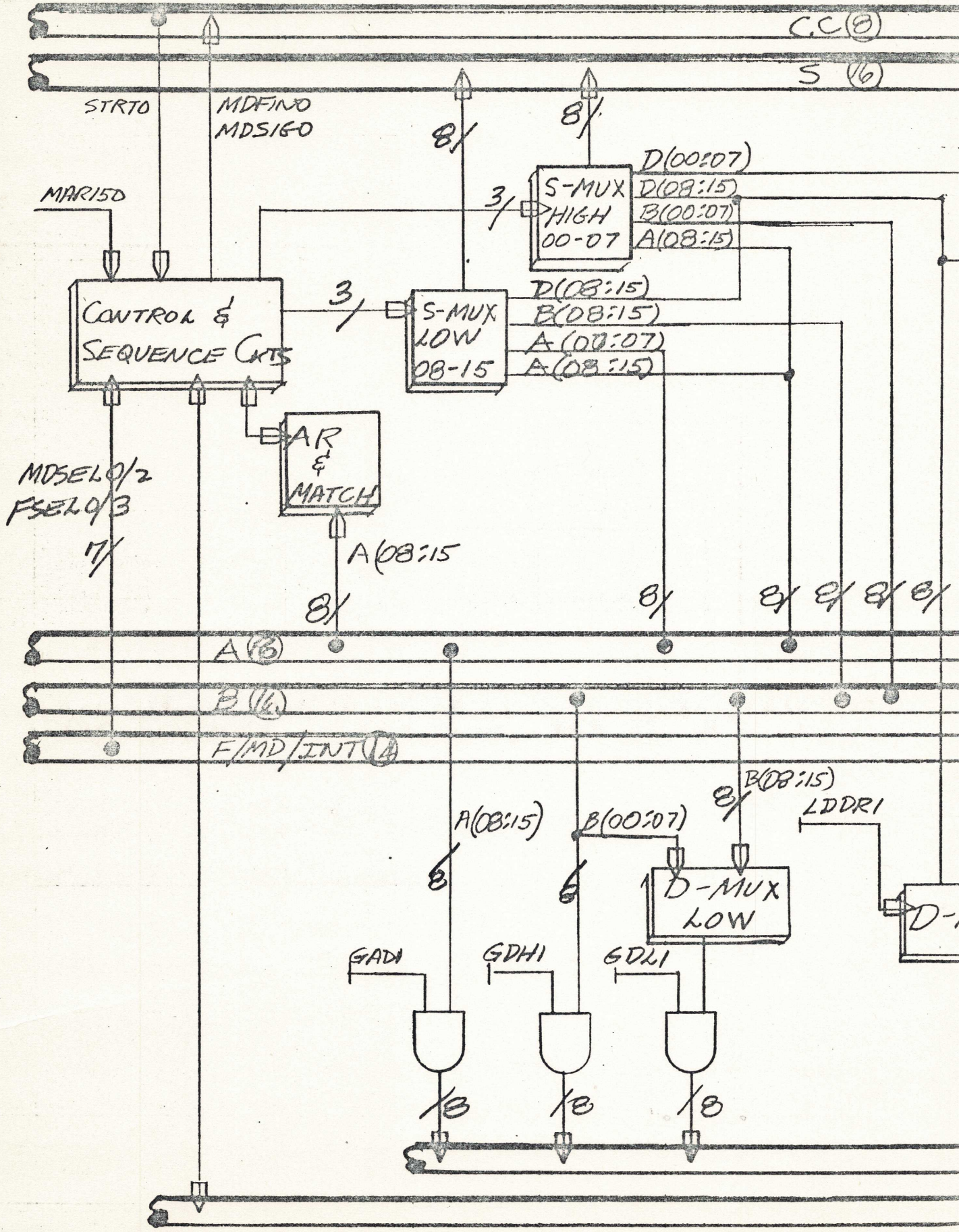


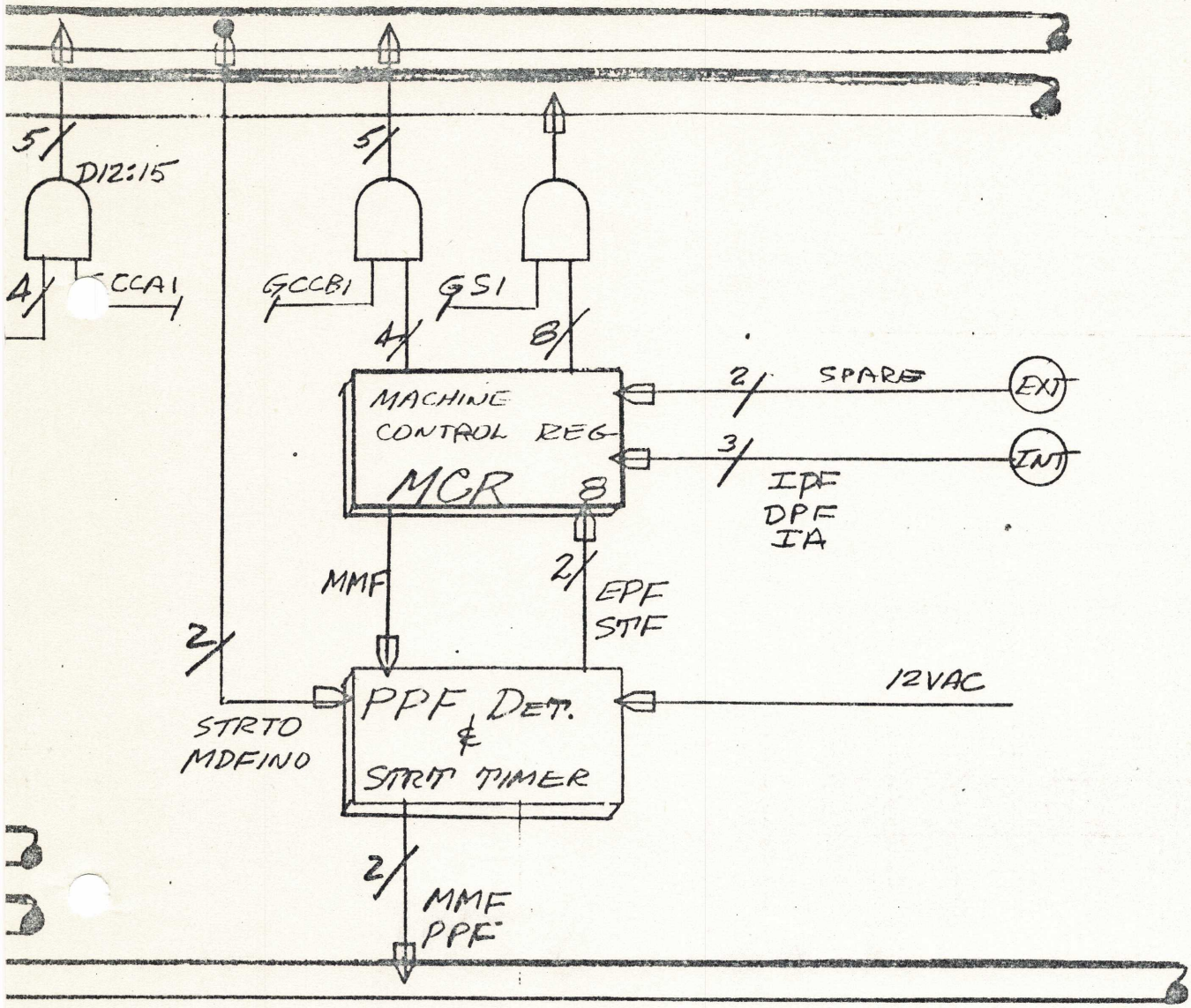
MODEL 80

BLOCK DIAGRAM

FIG – 4.1

7/29/71
RPS


CC(8)
S (6)
STRTO
MDFINO
MDSIGO
8/
8/
D(00:07)
S-MUX
HIGH
00-07
D(08:15)
B(00:07)
A(08:15)
3/
MAR15O
CONTROL &
SEQUENCE CKTS
3/
S-MUX
LOW
08-15
D(08:15)
B(08:15)
A(00:07)
A(08:15)
AR
&
MATCH
MDSELO/2
FSELO/3
7/
A(08:15
8/
8/
8/
8/
8/
8/
A(16)
B(16)
F/MD/INT(4)
B(08:15)
8/
LDDR1
A(08:15)
B(00:07)
8
8
D-MUX
LOW
GADI
GDHI
GDLI
8
8
8



MOD 80 I/O MODULE
BLOCK DIAGRAM

FIG. - 4.2

5. MICROPROGRAMMING

The control store of the Model 80 is a 32-bit word memory which may be read indirectly by instruction to retrieve data and may be written into by instruction if it is a writable memory. On some models, the control memory may be a ROM or RAM type memory and mixed in any practical configuration. The basic Model 80 contains a 768 X 32 ROM array containing the user repertoire and support programs.

The basic instruction format provides the computer with a three address capability, but various options of the repertoire can modify this to range from two to four. Figure 5.1 displays the different types of instructions and their modifiers. It is included in this section to make its understanding more graphic.

The format of the micro-instruction specifies which module is to be addressed, allowing only one module of the computer to be addressed at any one time, and all other modules must ignore the communications in process. Bits (0:2) of the instruction selects the module to which the instruction is addressed.

5.1 Module 0

Module 0 addresses the control module. As shown in Figure 5.1, instructions are interpreted differently for Module 0 than others. In the normal sequence of instructions, (e.g. no branches), the hardware of the control module controls the reading of its memories, and does little otherwise but gate the registers specified by the instruction. When it, itself, is addressed by instruction, it is for the purpose of a conditional transfer.

5.1.1 Branch/Execute Instructions

There are two types of transfers recognized by Module 0. The

most common is the branch. The branch instruction conditionally transfers control of the CPU to a specified address of control memory and proceeds sequencially from there. The second type of transfer, commonly called an "execute", transfers control to a single instruction at a specified address of control memory, then normally returns to the original sequence. Any type of instruction may be "executed" including additional execute instructions to any depth. However, an execute which results in a branch will not return to the continuing sequence. Bit (04) of the instruction determines whether the instruction is a branch or execute type.

5.1.2 Address Link/Register Link

There are two types of Module 0 transfers - address link and register link. They are selected by the state of bit (03) of the instruction.

The linked transfer is similar in function to the user level BAL instruction, and can be used to transfer to subroutines when they may be entered from more than one location. The location of the next sequential instruction following the transfer is deposited in the register specified by the Link field of the instruction (bits 11-15), and a transfer is conditionally executed to the effective address.

When the address link is selected, the transfer address is specified by bits (20-30, 31) of the instruction. Bit 30 is the LSB and bit 20 is the MSB+1. Bit (31) of Module 0 instructions is used as the MSB of the address, and is implemented in this manner solely because of hardware dictation and should be of no practical concern to the micro-programmer working at the micro-assembler level.

The register link is used when the transfer address is contained in a register. In this instruction, a branch is taken to the location contained in the register specified by the B field. Bits 25:30 of the instruction can be used to mask out the least significant bits of the address contained in the B register. (i.e. If instruction bits 25:30 are "ones" the B address is unaffected. If they are "zeros", the corresponding bits of the address (contained in the register specified by the B field) are forced to zero.)

5.1.3 Conditional Branches

All transfers are conditional upon a state selected by the F field of the instruction. By selective coding of the F field, either the condition code of the user level machine or the status of the CPU can be tested.

The F field codes are listed below:

F FIELD	MNEMONIC	OPERATION
0000	BRZ	Branch on CPU zero
0100	BNX	Branch on CPU not zero
0001	BRL	Branch on CPU less
0101	BNL	Branch on CPU not less
0010	BRG	Branch on CPU greater
0110	BNG	Branch on CPU not greater
0011	BUF	Branch if logical product of user M1 field and PSW (12:15) is zero
0111	BNF	Branch if logical product of user M1 field and PSW (12:15) not zero
1000	BRC	Branch on CPU carry
1100	BNC	Branch on CPU no carry
1001	BRV	Branch on CPU overflow
1101	BNV	Branch on CPU no overflow
1010	B	Branch unconditional
1011	AIN	Arm interrupts and Branch
1111	DIN	Disarm interrupts and Branch

The control field of Module 0 instructions is discussed in Section 5.3.

5.2 Non-CPU Instructions

As stated previously, when the module number is not zero, the CPU does not operate on the instruction, and the fields are interpreted differently. The module field (bits 0:2) and the F field (bits 16:19) are interrogated individually by the external modules. There are four types of non-CPU instructions, selected by bits (3:4) of the instruction.

5.2.2 RRX

The RRX is a register-register and transfer instruction. It is effectively a four-address instruction in that it gives the register address of the two operands, the register address of the results, and the location for the next sequential instruction.

The two operands are addressed by the A field (bits 11:15) and the B field (20:24). The contents of these two registers are gated respectively to the A bus and B bus of the computer.

The S field (bits 6:10) selects the destination register to which the results will be gated from the S bus. This field, as well as the A and B fields, may be null selected.

The page address field (bits 25:30) selects the low order address of the next instruction. The high order bits are taken from the current location address. The C field (bit 31) being true makes the transfer conditional upon a signal returned by the addressed module at the completion of the instruction. (The ALU, for example, returns the Carry flag as its signal.) If the module signal, which is designated MSIG, is true, and bit 31 of the micro-instruction is true, the branch does not occur, and the next sequential instruction is executed. Any other condition causes the transfer to be effected.

5.2.2 RRC

The RRC is a register-register control type instruction. The interpretation of the instruction fields is identical to that of the RRX, with the exception of bits (25:31) which contained the page address within a RRX instruction. Bits (25:31) of the RRC instruction provide the micro control of the CPU and are described in Section 5.3.

5.2.3 RIM

The RIM instruction provides an 'immediate' field for ease of generating constants and bit masks. 'Immediate' is the term generally used to infer that the immediate contents is the actual operand rather than the address where the operand will be found. Bits (20:31) of the instruction are gated onto the B bus as the B operand with bit (20) being carried as the sign and most significant four bits. The S field and A field of the instruction are interpreted identical to that of the RRX and RRC instruction.

5.2.4 RWT

The RWT is the store or write instruction of the repretoire if the CPU is equipped with a writable control store. There are several notable differences pertaining to this instruction.

a. Although the Module number cannot be zero, it may be any other configuration, as the internal modules are never communicated with.

b. The S field is not interpreted and should be Null selected.

c. The F field is not interpreted.

d. The B field addresses the register containing the address to be written into.

e. The A field addresses the register containing the data to be stored in memory.

f. The LSB of the control store address steers the data to the least/most significant half of memory (see Section 5.4).

Bits (25:31) of the RWT instruction are interpreted as a control field as in a RRC instruction.

5.3 Micro-Control

In order to facilitate the emulation task of the CPU, certain instructions allow an order of micro-control within the CPU. The instructions possessing this capability are the Module 0, the RRC and RWT instructions.

5.3.1 MC Field

The MC field is the user memory micro-control and allows various controls over the user memory instruction location counter (MLC), the user memory address register (MAR), and the reading and writing of the user memory.

The significance of the bits of the MC field are as follows:

MC FIELD	CONTROL
00XX	No memory action
01XX	Instruction Read
10XX	Data Read
11XX	Data Write
XX00	No Register Action
XX01	Increment MAR by two[1]
XX10	Increment MLC and MAR by four
XX11	Jam MLC to MAR

There are certain hardware connotations to the MC operations which are not made apparent by the above table.

a. MC (01XX) informs the hardware that an instruction read will take place and causes the CPU to read the instruction location plus the next location in anticipation of a full word user

---

[1]If this combination is done in conjunction with an instruction read, (MC-0101), the MLC is also incremented by two.

instruction. This results in the second half of the user instruction becoming available in the user's memory data register (MDR) and the first half being stored in the CPU hardware for the instruction emulation.

b. The memory read operations are conditional when used within Module 0 instructions. The read memory is only effected if the operation does not result in a transfer. (This conditioning is used to expedite the emulation of user branch instructions)

c. All micro-control is effected before the instruction occurs with the exception of data read and data write. This control is effected after completion of the instruction, which allows the micro-programmer to use the MAR or MDR as a destination and begin a read/write data immediately. It also allows the execution of the increment and jam control and the addressing of the MAR as the destination register simultaneously, which has functional utility.

5.3.2 D Field (Decode Instruction)

The D field bit informs the CPU to halt the sequential flow of micro-instructions and begin to emulate the next user instruction. The op-code field of the new USER instruction is in the UIR and vectors to a control store address where the emulation sequence begins. This implies that the micro-programmer must have done an instruction read in the current or a prior instruction using oneof the proper MC field designations. The execution of a decode is conditional when used within Module 0 instructions as described in 5.3.1.b.

5.3.3 E Field

This field is used to enable (E) changing of the condition code of the PSW. When changing is enabled, the condition code is changed under control of the module addressed by the current or subsequent instructions. (The ALU, for example, jams its C, V, G and L flags into the condition code upon completion of its function.) The meaning of the condition code is a function of the module addressed.

5.3.4 K Field

The K field of the micro instruction is an extension of the F field of the instruction. It is available only on the RRC and RWT instructions and constitutes the control signal (KSIG) to the modules. Its meaning, just as the F field, is defined by the module addressed by the current RRC instruction. The ALU, for example, reinterprets shifts to be extended precision when 'KSIG' is active. It is also used to extend the functions of the I/O module.

5.4 Control Store Data Storage

Data may be stored in the ROM for the purpose of translation tables, sine tables, etc., or in the case of the dynamic control store, for general programming. Data is retrieved from the control store during execution of RRC, RRX or RIM instructions when the I field bit (instruction bit 05) is true. When the I field bit is set the data addressed onto the B bus is used as the store address of the ROM or DCS, and causes the CPU to replace this data with the addressed data before signaling the addressed module to begin its task. The least significant bit of the B bus selects either the right or left half of the data for replacement on the bus. This facilitates address calculations and packing of the DCS. The second

LSB of the B bus, therefore, addresses the LSB of the control store.

5.5 Interrupts

The hardware of the computer provides eight priority interrupts. Each interrupt has a unique "trap" location associated with it. Recognition of an interrupt causes the instruction stored at its respective trap location to be performed. The RLR contents are preserved to allow the address of the interrupted sequence to be saved if desired so that control can be returned at the completion of the interrupt routine. Certain interrupts can be disabled by bits of the PSW as designated below. Additionally, all interrupts can be enabled/disabled as a group by micro instruction. All interrupts not masked by PSW bits are interrogated when a new user level instruction is decoded, regardless of the status of the group enable. The group enable is automatically disabled at the beginning of a user emulation, and must be enabled by instruction if the programmer wishes to recognize interrupts. The following table lists by priority the pertinent information for each interrupt. The external interrupts, 2 & 3, will cause an EXTERNAL INTERRUPT.

INTERRUPT	TRAP	MASK	GROUP ENABLE
Primary Power Fail	105_{16}	NONE	YES
Machine Malfunction[1]	105_{16}	PSW 02	YES
Display	104_{16}	NONE	YES
Ext. Interrupt 0	103_{16}	PSW 01	YES
Ext. Interrupt 1	102_{16}	PSW 01	YES
Ext. Interrupt 2	101_{16}	PSW 10	YES
Ext. Interrupt 3	100_{16}	PSW 11	YES
Illegal Instruction	107_{16}	NONE	NO
Privileged Instruction	107_{16}	PSW 07	NO

1. This interrupt can have several causes and is discussed further in I/O section.

5.6 Registers

The basic CPU has a register stack containing 16 user registers and 8 general purpose registers for use by the microprogrammer. In addition, the bulk of the remaining CPU registers are also available to the microprogrammer.

A register is available to the microprogrammer if he can address it to one or more of the internal busses. The following list tabulates the addressable registers and their respective address on the designated bus.

BUS ADDRESS (HEX)	S BUS	B BUS	A BUS
00:OF (16 Gen. Registers)	UGR	UGR	UGR
10:17[1]	RGR	RGR	RGR
18	PSW	PSW	PSW
19	MDR	MDR	MDR
1A	MLC	MLC	MLC
1B	MAR	UDRP1	UDRI
1C	USR	NULL	NULL
1D	UDRD	USRI	NULL
1E	UDR	USR	UDR
1F	UDRP1	UXR	UDRP1

[1] RGR07 (bus address '17) is designated NULL register for the S bus by convention of the micro-assembler.

Mnemonic	Register	Comment
UGR	User general registers	16 registers manipulated by emulated language
RGR	Micro-level general registers	8 additional GP registers available to the micro program
PSW	Program Status Word	16 bit register containing interrupt enables and flags
MDR	Memory data register	
MLC	Memory location counter	location counter of emulated program.

Mnemonic	Register	Comment
MAR	Memory address register	
NULL	No register selected	gates 0 to A & B bus
USR	User source register	register selected by bits 12-15 of emulated instruction
USR1	User source register immediate.	bits 12:15 of the emulated instruction gated onto A Bus.
UXR	User index register	same as USR except NULL gated to A Bus if field is Ø
UDR	User destination register	register selected by bits 8:11 of emulated instruction.
UDRP1	User destination register plus 1	register selected by bits 8:11 of emulated instruction +1 (must be odd)
UDRD	User destination register direct	S Bus 12:15 replaces UDR contents.

The last four bits of the PSW contain the condition code of the emulated computer. In general, these bits can be manipulated by any addressed module unless the PSW is the S bus destination or their change has been inhibited by the micro-instruction (see Section 4). The individual bits of the PSW which have hardware implications are:

PSW (01)	ATN (0) and ATN (1) enable
PSW (02)	Machine Malfunction
PSW (07)	Privilege instruction/Memory Protect
PSW (10:11)	ATN (2:3) interrupt enables
PSW (12)	C flag of condition code
PSW (13)	V flag of condition code
PSW (14)	G flag of condition code
PSW (15)	L flag of condition code

PSW bits 3 through 6, and bit 8 may have significance to the microprogram.

The following additional registers have the indicated capabilities and connotations:

1. The MDR receives data asynchronously from Memory
2. The MDR and MAR being addressed cause the processor to interlock with memory when they are the source or destinations of the current instruction and the processor is requesting memory service.
3. The MAR and MLC can be incremented by 2 or 4 from the ROM micro-control.
4. The contents of the MLC can be jammed into the MAR by ROM micro-control.

5.7 Arithmetic Module (ALU) Programming

The ALU (arithmetic-logic unit) is a standard module in the Model 80 hardware. It is addressed as Module 1 in the module field of the instruction and is capable of 20 functions. Communications with the ALU is asynchronous. By design, the ALU is never busy and for the majority of ALU functions response is within 100ns. (This allows an ALU referenced instruction to be completed in 200ns.).

For functions which require more than one ALU cycle (i.e., shifts, rotates, multiply-divide), the ALU does not respond with a finish signal until the completed results are on the S Bus.

Multiply-divide and extended shift operations can be performed only on the 24 general registers and must address the same register pair on both the A & S bus. The same restrictions that apply to these operations on the Model 70 must be adhered to in the Model 80.

A user emulated multiply-divide instruction is micro-coded by: selecting the ALU (module 1), addressing the UDR on the S bus, the UDRP1 onto the A bus, the USR onto the B bus, and the required function code for the operation. When the ALU signals its completion, the results have already been deposited in the UDR.

The ALU generates valid CPU flags for all instructions. The C flag is gated as 'MSIG'.

To implement a shift or rotate instruction, the register to be manipulated is addressed onto the A bus, the shift count is put onto the B bus, and the destination register is addressed from the S bus. For extended shifts, the same register pair must be addressed on both the A and S bus, the even register being addressed to the S bus, and the odd register to the A bus.

Certain of the instructions are redefined by the K field of the RRC instruction (KSIG). When this field is active, all shifts become double precision (0-31 shifts), the multiply instruction is interpreted as unsigned, and the add with carry and subtract with carry,which normally utilizes the CPU flags, utilizes the C,G, and L flags of the PSW condition code.

The ALU performs its function as per the following configurations of the F field.

F FIELD		OPERATION
0000		subtract
0001		add
0010	*	subtract with carry(CPU flags)
0011	*	add with carry (CPU flags)
0100		(not used)
0101		logical and
0110		logical exclusive or
0111		logical or
1000	*	logical shift right
1001	*	logical shift left
1010	*	rotate right
1011	*	rotate left
1100	*	arithmetic shift right
1101	*	arithmetic shift left
1110	*	signed multiply
1111		signed divide

* Note these instructions are redefined by the K field of the RRC instruction.

5.8 I/O Module Programming

The I/O Module performs a multiplicity of functions. In general, it is addressed to communicate with the multiplexor channel. It has the additional capability of performing byte manipulations for the CPU both in conjunction with an I/O exchange and without one. Furthermore, the I/O Module contains the machine control register (MCR) which stores machine trouble conditions and interrupts the CPU. The contents of the MCR can be sensed, tested and cleared. Module number 2 has been assigned to the I/O Module.

5.8.1 Multiplexor Channel

The Multiplexor Channel, generated on the I/O Module, is operationally identical to the Model 70 Multiplexor Bus in all respects. The Multiplexor Bus is a byte or halfword oriented I/O system which communicates with up to 255 peripheral devices.

A single instruction from the CPU contains the device address, the encoded function and up to 16 bits of output when needed. The Multiplexor bus generator provides single or multi-cycle operation to address the device, transmit the decoded function, send or receive over 16 bidirectional data lines and synchronize the exchange.

The normal byte or halfword operation consists of an address cycle and a data cycle. However, during a Read/Write Block sequence, the address cycle is not used. For halfword functions (RDH/WDH) with a byte oriented device controller, two data cycles will be used to transfer the halfword.

5.8.2 Byte Manipulation

The I/O Module has the capability to perform byte manipulation both in conjunction with an I/O operation and without one. The byte steering is under control of the least significant memory address bit in the MAR and also the KSIG line. For halfword operations, this manipulation is inoperative but the double data cycle with packing/unpacking will result when the test line (HW) is inactive.

The encoded I/O Module functions and the byte manipulations are shown on the table at the end of section 4.3.3.2.

5.8.3 Machine Control Register

An interrupt on Machine Malfunction (MMF) is generated when bit 13, 14, or 15 in the Machine Control Register (MCR) is set. The sense MCR instruction (SM) gates MCR (08:15)to S(08:15) and MCR (12:15) to CC (C/V/G/L). The CM instruction clears the MCR as does SCLRO.

The MCR bits are assigned as follows:

MCR15 (L)	EPF	Early PPF
MCR14 (G)	IPF	Instruction Parity Fail-Mem
MCR13 (V)	DPF	Data Parity Fail-Mem
MCR12 (C)	MMF	Set when any other bit is set
MCR11	IA	Illegal Memory Address
MCR10	STF	STRT Time Out Fail
MCF09	SPARE	
MCR08	SPARE	

The STRT Timer (35 us) is activated whenever the CPU sends out the STRT signal to the varous system modules, (ALU, I/O Module, etc.), and is cleared by the MDFIN signal from the module.

Should the time out occur before the MDFIN signal arrives and the module number is not 2 (i.e. I/O module), bits 10 and 12 of the MCR are set, the MMF interrupt occurs and a false MDFIN signal is generated to restart the CPU clock.

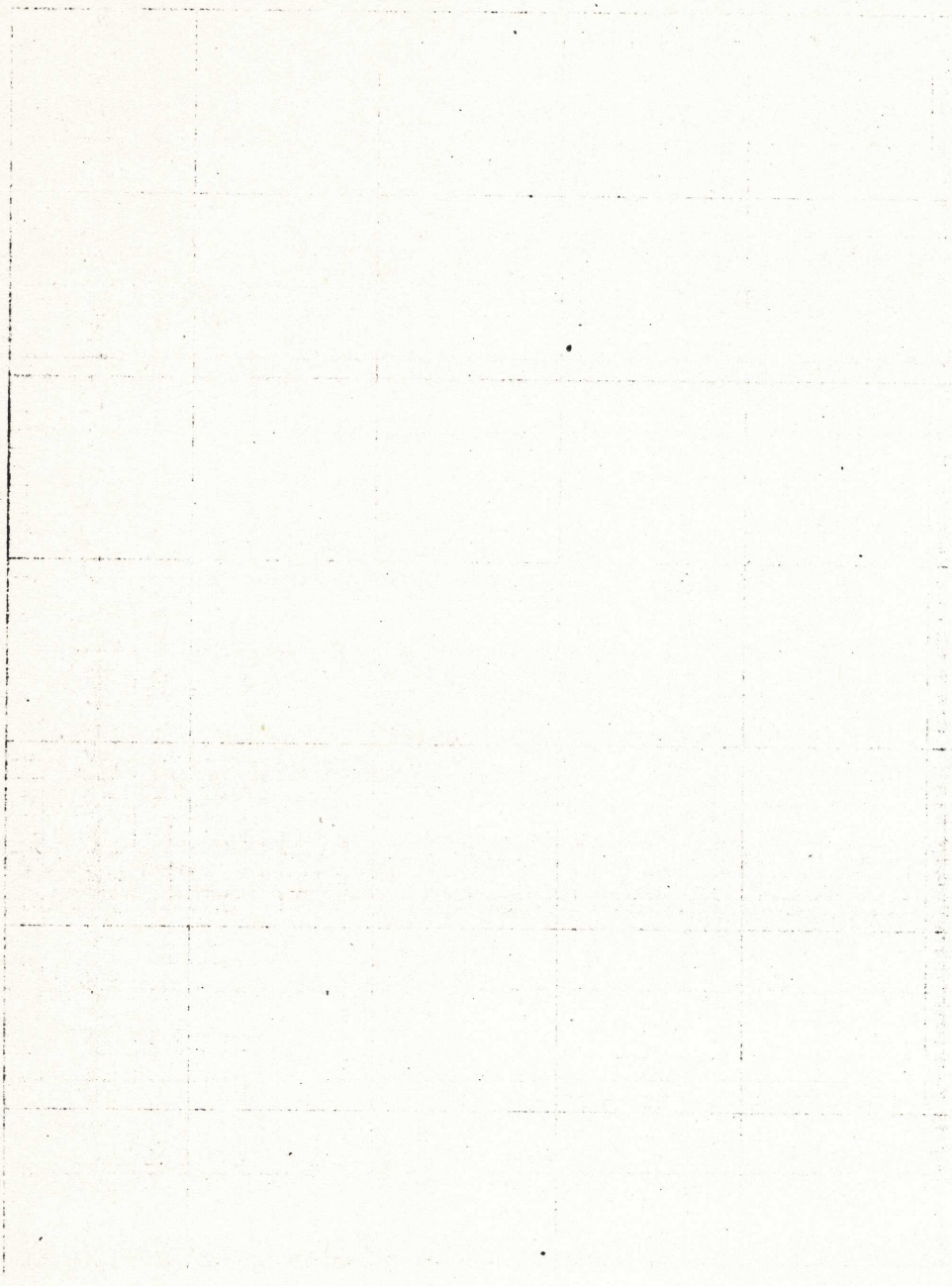
In the case of STRT time out during a D-Bus Operation, the MCR is not changed. However, the condition code bus (CVGL) is set to (0100) and the false MDFIN sent to restart the CPU clock.

5.8.4 I/O Module MSIG

MSIG is a control line sent back to the CPU and used by the RRX instruction for conditional branching. The I/O module activates MSIG any time it attempts to manipulate the condition code (SCC) and any bit is active in the condition code. Additionally, MSIG is active on a DACK instruction when the data channel controller requests a read.

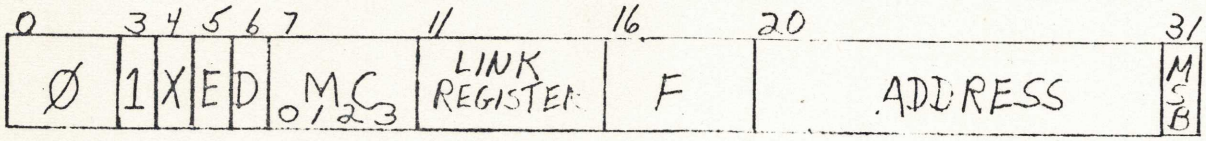
5.9 CPU Flags

The CPU contains a flag register which is independent of the PSW flags and are manipulated by any module which attempts to affect the PSW condition code by activating the 'SCC' control line of the CPU control bus when it is addressed. When the SCC control line is active, the state of the VCC, CCC, GCC and LCC are unconditionally jammed into the CPU flag register and conditionally into the PSW condition code. The changing of the PSW is controlled by the micro-programmer by the E field of the micro-control. The state of the CPU flags can be individually tested by the module Ø instructions.

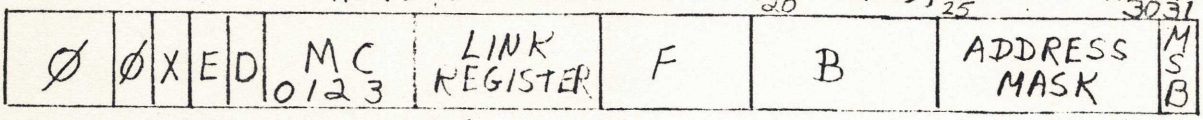


change D to K

K FUNCTION CODE MODIFIER

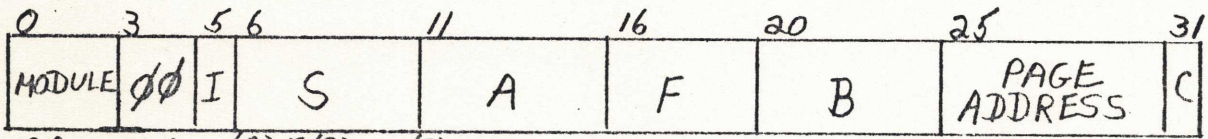


ADDRESS LINK FLAGS ARE TESTED AS PER F
IF TEST PASSES: RLC → (LINK REG), XFER TO ADDRESS

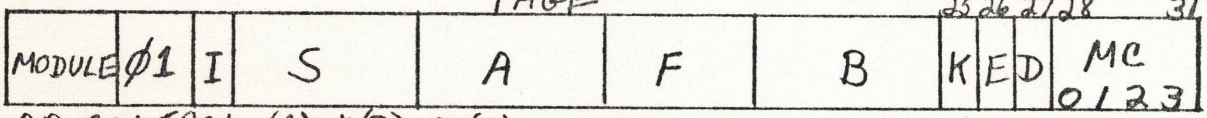


REG-LINK FLAGS ARE TESTED AS PER F
IF TEST PASSES: RLC → LINK ADD, XFER TO ADDRESS+(B)

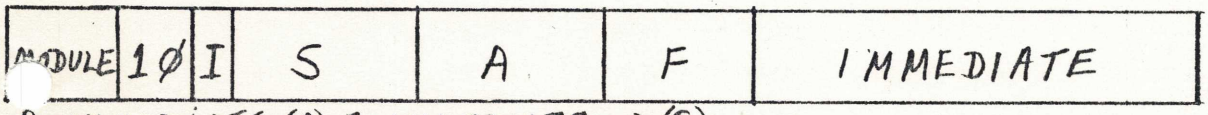
CONTROL INSTRUCTIONS



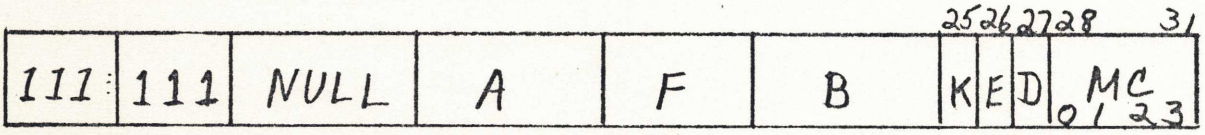
RR XFER (A)F(B)→(S) IF C=1 AND MSIG=1 XFER TO NEXT INSTRUCTION
OTHERWISE XFER TO PAGE ADDRESS ON CURRENT PAGE



RR CONTROL (A) F(B)→(S)



R IMMEDIATE (A) F IMMEDIATE→(S)



R WRITE (A)→ RAM ADDRESS SPECIFIED BY (B)

MODULE INSTRUCTIONS

Selects register gate to A bus
Selects register gated to B Bus
Selects register to receive S Bus
Selects function of addressed module
Enable setting of condition code
If set transfer is conditional
Execute
B field is indirect address of data
Decode next instruction
Unused

MC Field Designations (Memory Control)

00XX No memory action
01XX Instruction Read
10XX Data Read
11XX Data Write
XX00 No Register Action
XX01 Increment MAR by two*
XX10 Increment MAR & MLC by four
XX11 Jam MLC to MAR

*If done in conjunction with an instruction read, the MLC is also incremented by two

Figure 5.1

6. MEMORY SYSTEM

6.1 Introduction

The Model 80 Memory System incorporates the highest performance semiconductor devices available in both the MOS and Bipolar technologies with hardware provisions for a hierarchy of Dynamic MOS Main Memory (user memory) and Bipolar Read Only or Dynamic Control Store Memory (Control memory).

In its basic configuration with read only control memory, the Model 80 emulates completely the INTERDATA Model 70 and is upward compatible with the INTERDATA Models 3, 4 and 5. Dynamic control store provides the hardware necessary for microprogram debugging, dynamic microprogramming, language development and a hierarchy of routines. These later functions are not fully supported by Model 80 software or extensive documentation but the hardware is provided as a development tool for static or dynamic microprogramming.

The Model 80 memory system and its relationship to the rest of the system is shown in Figure 6-1. Main Memory (MM) uses 1024 bit dynamic MOS devices as the storage element. The MOS devices with their drivers and sense-digit circuits are packaged on Memory Storage Unit boards (MSU) to provide 8,192 words by 16 (or 17 with parity) bits. Up to four MSU's can be driven by one Memory Bank Controller for a maximum storage capacity of 65,536 eight-bit bytes. Two independent ports to main memory are provided. The CPU port to memory is for the exclusive use of the Model 80 processor. The DMA port provides access to memory over a multiplexed Direct Memory Access Channel (DMAC). The DMA port provides serial priority control for up to four DMAC's and is given priority over the CPU port. The DMAC bus is compatible with the INTERDATA Model 70 system and accepts Model 70 selector channels.

6.2 System Main Memory

6.2.1 Main Memory Organization

6.2.1.1 Modularity: The Memory system is composed of two building blocks, the Memory Bank Controller (MBC) and the Memory Store Unit (MSU). The assembly of one MBC, Figure 6.2, and one to four MSU's constitute one memory bank (MB) of up to 64K bytes plus halfword parity. Expansion of the MB from 16K bytes to 64K bytes is accomplished by plugging 16K byte MSU's into the mainframe chassis and adding the connecting ribbon cable to the MBC.

6.2.1.2 Memory Bank Controller (MBC) Function:- The MBC controls access to the MSU's from two independent asynchronous ports and from the memory Refresh Generator, Figure 6.2 and connects the requesting port to one of four MSU's. The MBC also performs the following functions:

1. Contains a memory port interlock which senses requests for memory, assigns access to the request of highest priority (one of three including request for refresh) and connects the related address, data and control lines from the selected port to the requested MSU.

2. Provides controls,address and address plus one for the even and odd memory address busses, thereby permitting overlapped memory read cycles for CPU instruction fetches.

3. Generates and checks memory data parity.

4. Provides memory write protect as an option by comparing the incoming 1024 byte block address with its protect code stored in a separate 64 bit high speed stack located on the MBC.

6.2.1.3 Memory Store Unit (MSU) Functions: The MSU contains two separately addressed and timed modules, each 4096 words of 16 bits plus parity. The two modules share 16 sense-digit circuits also on the same board. Each module employs 1024 bit dynamic MOS devices as the storage element. The device used, AMS5002, is presently the fastest 1024 bit MOS RAM available.

The MSU accepts address, data and control from the MBC and generates the requested sequence. One to four MSU's are interconnected with the MBC through a single cable strap at the front of the chassis. All signals to and from the MBC are "wired OR" through this cable.

6.2.2 System Timing and Performance

6.2.2.1 Timing References: The speed performance of a memory system is specified in terms of its read access time and cycle time. Herein this systems performance will be discussed as seen at the two memory ports. All time references are therefore made with respect to a memory request signal CREQO on the CPU port or REQO on the DMA port, measured at the memory port connections on the mainframe backplane.

6.2.2.2 Timing Modes

On the basis that main memory is generally addressed in word sequences of random starting addresses, rather than random by single words, the design of this system is such that the 4K word modules most frequently accessed remain in a higher powered, faster access mode, while those least accessed revert to a lower powered, slower access mode. Memory access time is approximately 70 ns greater in the low power mode, but the available power is thus used more effectively. Moreover, the memory power supply load is not affected by expansion of SU's within the same MBC except for a proportional increase of logic supply current.

6.2.2.2 Basic MSU Performance - The basic subsystem employed in this memory is the 8K byte module of which there are two per MSU. The 8K byte module determines the performance parameters from which overall memory system performance is derived. Cycle time for this module is 275 ns (worst case) and read access time is 190 ns.(worst case).

6.2.2.3 Memory Bank Performance - Timing performance of the system differs from the basic parameters of the 8K byte module for two reasons:

1. The read access time is degraded to 225 ns max. because of the delays introduced in switching two asynchronous ports to one of eight 8K byte modules.

2. The system cycle time is reduced effectively to 225 ns because the MBC is designed to take advantage of multiple 8K byte modules by overlapping memory cycles whenever possible.

6.2.2.4 Overlapped Cycle Operations - Memory cycle overlap requires various functional redundancies in the design which are intended primarily to provide a fast fullword (32 bits) access when the CPU requests an instruction fetch. The functional redundancies thus implemented are used additionally to produce cycle overlaps and thereby enhance the memory system cycle rate. In general, cycle overlaps occur when the memory address is switched, in any sequence, through the eight combinations of A00, A01 and A14.

As illustrated in Figure 6.3, cycles are overlapped for approximately 50ns. thereby producing an effective cycle time of 225 ns. Maximum cycle overlap is achieved when the CPU commands an instruction fetch (CFTCHO drops). In this case, the first cycle is started at address K, Figure 6.4, while the next halfword address K+1, generated within memory, permits a cycle at that address to begin 100ns. later. Both cycles are completed in 375 ns. Moreover, additional requests from the DMA port when interlaced with the K+1 cycle will produce a maximum memory system cycle rate of 5.5 MHz.

6.2.3 CPU Port Characteristics

The CPU to Memory interface contains a total of 39 lines, Fig. 6.5. Fifteen lines are provided for address CA(00:14)0 and 16 lines for data CD(00:15)0 . The remaining eight lines are divided equally for memory control and status functions. Three different sequences may be commanded from the CPU. These are:

1. CPU - Instruction Fetch - An instruction fetch cycle, Fig. 6.6, is started by dropping CREQO concurrent with the presence of valid address CA(00:14)0. Memory then executes the instruction fetch sequence when CFTCHO drops no later than 15ns following CREQO. The first halfword appears on the CPU bus at 215 ns and is signaled by the first memory ready CRDY0 10 later at 225 ns (access time). The second cycle which begins at 100 ns, provides data at 315 ns with a second CRDYO at 325 ns. The two modules exercized for the instruction fetch complete

the entire sequence in 375 ns. while releasing the MBC for the next cycle to the DMA port or Refresh Generator at 325 nsec. Since an instruction fetch requires two memory read cycles, the Write command CWRTO must remain high. A memory parity error occurring at the instruction or data read cycles will cause IPFO or DPFO, respectively, to drop approximately 70 ns. following each CRDYO.

2. DPU - Memory Read: A memory read cycle is obtained and responds identically to the first half of the instruction fetch sequence. In this case, the MFTCHO command is kept high. Data access is obtained at 215 ns., signaled by CRDYO at 225 ns. The MBC is released for the next request at 225 ns. while the cycle in the selected 4K word module terminates at 275 ns. A data parity error will cause DPFO to drop at 285 ns.

3. CPU - Memory Write: A write cycle (Fig. 6.7) is obtained by dropping CREQO concurrent with the presence of valid address CA(00:14)0, and write command CWRTO, while data CD(00:15) 0 may follow 25 ns. later. CRDYO appears at 125 ns while the MBC is released at 225 ns. As in the read cycle, the selected 4K word module completes the write cycle in 275 ns.

6.2.4 Memory Write Protect

Since memory is write protected in blocks of 1K bytes, a write request made with Priviledge Write Mode, PVMDO, low will cause the memory to compare the incoming 1024 byte block address with its protect code stored in a separate 64 bit stack located in the MBC. Upon finding a match the write cycle is aborted and the Write Protect Error (MPRTO) line to the Protect Controller is dropped. When a write request is made with PVMDO high, the memory will execute the write sequence without regard to the internally stored protect code.

6.2.5 DMA Port Characteristics

6.2.5.1 General - The DMA port provides access to memory over a multiplexed Direct Memory Access Channel (DMAC). The port has access priority over the CPU port and also provides serial priority control for up to four DMAC's. The four DMAC's are assigned respective priority when configured to the system.

The DMAC bus is compatible with that of the INTERDATA Model 70 Selector Channels (SELCH) which operate with any standard byte-oriented INTERDATA device controller.

custom or user designed channels may interface to the DMA port directly. The design of the DMA port with priority look ahead can handle multiples and mixtures of custom channels and SELCH's with a data transfer rate of up to 5.2M bytes. When a single custom channel is used,and when it is designed to simulate the CPU port signal timing requirements, the data transfer

rate can be as high as 9M bytes simply by adding a wire jumper across REQO and DREQ1, as provided for on the MBC and indicated in Figure 6.5. DMA port priority over the CPU port is effective when both the CPU and DMA are on cue for memory. As such, the CPU can capture alternate memory cycles even when the DMA bus is requested continuously (REQO on). This priority mode is generally desired since it results in a high memory use efficiency because it allows the CPU to steal memory when it is non-productively awaiting DMAC priority resolution and alteration of data and address on the bus. In this mode, the DMA data transfer rate can still be as high as 5.2M bytes if the CPU is not contending for access, but will generally be between 3M bytes and 4.5M bytes when the CPU captures alternate memory cycles. However, an option is provided on the MBC which will cause all CPU requests to be ignored irregardless of memory status as long as the DMA bus request REQO is on. This provision insures a DMA bus transfer rate of 5.2M bytes at the expense of locking out the CPU.

6.2.5.2 DMA Port Signals - The DMA port interface to memory contains 35 lines, Figure 6.5. Sixteen lines MD(00:15)/MS(00:15) are provided for data to/from memory and fifteen lines MA(00:14) for memory address. The remaining lines are used for memory control and status as follows:

Bus Request	REQO	Up to four DMAC's OR tie into the REQO line. Any DMAC may activate REQO at any time, but the system will not tolerate transient false signals on REQO. The REQO line must be released no later than 100 nsec. after the rising edge of ENO unless the DMAC requires additional memory cycles. This line may be jumpered to DREQ1 on the MBC when a single custom channel is required to run at memory speeds.
Bus Enable	ENO	The MBC responds to a REQO by activating ENO within 10nsec. ENO will remain active for 100 nsec. during which the requesting DMAC of highest priority will capture the DMA bus. The rising edge of ENO shall signal the capturing DMAC to alter the MA(00:14), MD(00:15) and WRTO lines as required within the next 100 nsec. ENO will serve no purpose and must be ignored by the custom channel when REQO and DREQ1 are jumpered (see REQO def.).
Write	WRTO	This signal when low with REQO will cause the memory to execute a write cycle at the indicated address MA(00:14). Timing of this signal shall be equal to that of the MA(00:14) lines.

Memory Ready	INHO	The memory will generate INHO whenever a cycle has been initiated from the DMA bus. When a memory read sequence is generated, the falling edge of INHO indicates that valid data is present on the MS(00:15) lines. INHO will be active for 50 ns. and its rising edge will mark the end of either a read or write sequence at which time the DMAC shall disconnect itself from the DMA bus within the next 100 ns. INHO will function and be timed identically to CRDYO when REQO and DREQ1 are connected for the high speed DMA bus mode.

6.2.5.3 DMA Port Timing - The activity of the DMA bus for which there are multples or combinations of custom channels and SELCH's is illustrated by Figure 6.8. The sequence shown begins any time REQO goes on, and is marked as time T_0 for reference. As indicated in Figure 6.5, the Model 80 MBC receives a memory cycle request DREQ 210 ns. after REQO is presented to the MBC front end. The MBC front end therefore prevents interference to the memory by DMA bus while DMAC priority and data alteration is underway.

During the intervening 210 ns. between REQO and DREQ1, ENO is dropped for 100 ns. during which time the requesting DMAC of highest priority captures the DMA bus. Upon raising ENO at T_0+110 ns. the DMAC, which is indicating a memory read request, Figure 6.8, alters address MA(00:14) within the next 100 ns.

At T_0+210 ns = T_A, the MBC front end generates DREQ1 which is seen at the MBC concurrent with valid address and write control. Prior to DREQ1, the CPU and the Refresh Generator are free to capture memory. When DREQ1 is on and memory is not busy, a cycle is initiated and ENO is again triggered on if REQO is still active, thereby selecting the next DMAC while memory services the current one.

As the memory cycle nears completion, a memory port scan is initiated at T_A+200 and continues until a valid memory bank request is received. The CPU and Refresh Generator may thereby steal the next memory cycle until the next DREQ1 is received at T_A+380 (if another REQO was given). As explained above, the cycle steal privilege given to the CPU while REQO is on may be locked out by making a field modification which causes REQO to block the CPU port scan.

As shown in Figure 6.8, the memory read cycle initiated by the DMAC at time T_A produces an active INHO at T_A+225 concurrent with valid data on MS(00:15). INHO is approximately 50 ns wide and its trailing edge which is concurrent with the trailing edge of ENO shall cause current data MD(00:15), address MA(00:15) and control WRTO to be altered thereby allowing the next DMA cycle to start at T_A+380 ns.

The activity illustrated by Figure 6.8 and described above does not apply in the event that a single custom designed channel is connected to the DMA port and is operated in the high speed mode. In the high speed DMA port mode, all timing conditions, with exception of the Instruction Fetch sequence, generated for the CPU port, are also applicable to the DMA port as illustrated by Figures 6.6 and 6.7 at equivalent speeds.

6.2.5.4 DMA Bus Performance - As described above, the data transfer rate obtained on the DMA bus is generally determined by the options exercized at the time of system configuration as summarized below:

1. The DMA bus may operate with a single custom designed channel in the high speed mode equivalent to CPU port cycle rates; OR, the DMA bus may operate with more than one channel either custom or MODEL 70 SELCH's in the multiplexed mode at slower cycle rates.

2. In either the high speed single channel or multiplexed multiple channel mode configuration, a second option is designed into the MBC which permits locking out the CPU bus whenever a DMA bus REQO is on. This achieves higher DMA bus data rates at a sacrifice of overall system efficiency. Otherwise, the CPU is permitted to capture alternate memory cycles.

Bus Mode	Priority Mode	Byte transfer rate in M bytes during REQO		Memory System Efficiency* [1]
		DMA Port	CPU Port	
1. High Speed DMA (Single Custom Channel)	1. CPU locked out during REQO	7.2 to 9	0	82% to 100%
	2. CPU and DMA alternate during REQO	3.7 to 4.5	3.7 to 4.5	82% to 100%
2. Multiplexed DMA (SELCH's and multiple custom channels)	1. CPU locked out during REQO	5.2		58%
	2. CPU and DMA alternate during REQO	3.1 to 4.5	3.1 to 4.5	69% to 100%

Note *1 Efficiency = $\frac{\text{Memory busy time} \times 100}{\text{Memory busy time + memory stand by time}}$

= Memory busy time = 225 ns.

6.2.6 Memory System Packaging

6.2.6.1 General - As noted above, the Model 80 main memory system plugs as separate units into the system backplane, and is an assembly of one to four SU's electrically connected to one MBC board, Fig. 5.4.

Present plans are to mount the SU's in the four slots adjacent to the MBC and interconnect them with two 50 strand ribbon cables at the front.

6.2.6.2 Memory Bank Package Size - The MBC is a full 14.75 by 15.5 inch board with approximately 80 IC's and two 50-pin 3M cable connectors for connections to the SU boards. The SU boards are equivalent in dimension to the MBC board and will mount 136 memory chips, 54 logic IC's and 700 other descrete components. The 136 memory IC's will be mounted on 0.520 inch by 1.2 inch centers and occupy less than one half of the SU board. Fully assembled, the 64K byte memory bank will occupy five card slots on 0.75 inch centers.

6.2.7 Memory Bank Power Dissipation

One memory bank with four SU's, of which any four modules (two equivalent SU's) at a given time, may be in a precharged mode, will dissipate approximately 140 watts. The exhaust air from this package shall be 60°C maximum. When supplied with 50°C air at 100 cub. feet per min. over the entire volume.

6.2.8 Memory Maintainability

General field testing of the Model 80 memory bank will be performed with either the MBC or one MSU on one backplane extender board. When in this position, the MBC or MSU will lie on top of the extender supports with all of its apparatus side exposed. Since the position of the SU's is interchangeable and the connectors have dedicated addresses, the SU under test may be moved to the outside position and addressed accordingly. Front test points will be numerous on the MBC but scarce on the SU's because of its component density.

6.3 Read Only Control Memory

Up to 1024 words by 32 bits of Read Only Control Memory can be housed on the ALU board. The standard Model 80 instruction set requires 768 words. An additional 256 words can be provided in special systems. TTL compatible 256 x 4 bipolar ROMs with access time of 50nsecs. are used with fixed masked patterns. Pin compatible "programmable once", PROMS are also available for quick turnaround on special low run patterns. Expansion Read Only Control Memory can be provided with up to 3072 words by 32 bits on a singe board in increments of 256 words (available 2nd half of '72). Systems with large (greater than 1K x 32) expansion ROM will require special considerations on power and cooling because of high power requirements of the fast bipolar memory devices (approx. 500 MW. per 256 words). Large ROM systems will also degrade machine performance slightly because of transmission delay to the expansion store.

6.4 Dynamic Control Store Memory

The Dynamic Control Store (DCS) uses high speed 256 bit bipolar semiconductor memory devices with access and cycle time of 50 nsec. It is available on a single board with up to 1024 words by 32 bits in increments of 256 words. The address and data for a read operation are connected to the CPU over a front panel cable. See Figure 6-1, and the DCS must be located adjacent to the CPU/ALU boards. Data is written 16 bits at a time from data on the back-panel A bus and address on the B bus. The DCS can operate asynchronous to the basic CPU Read Only Memory by stopping the clock when the DCS is addressed. This retains the execution times on basic instruction but allows operation with a slower DCS.

6.5 Special Configuration

Figure 6-2 shows Model 80 hardware specialized in two respects - multiple main memory banks and DMAC ports to the Dynamic Control Store. Both of these are potentially very useful configurations, however, they require special system handling as outlined below.

Multiple Main Memory requires special backpanel and cabling in the expansion chassis that contain the additional memory banks. Special handling is also required for parity, memory protect, and the DMAC priority circuits. In any case, the total main memory capacity cannot exceed 65,536 bytes.

The Dynamic Control Store with access to a DMAC bus can be handled with two options (not both). In option A the DCS has a port to the Main Memory DMAC Bus, and the DCS must be assigned address space within the maximum 65,536 Main Memory field.

In option B, the DCS connects to a private DMAC bus. In both cases, the DCS is located in a special expansion chassis directly above the processor and requires special cabling.

BY..... DATE..... SUBJECT..... SHEET NO..... OF.....

CHKD. BY..... DATE..... JOB NO.....

STANDARD MODEL 80

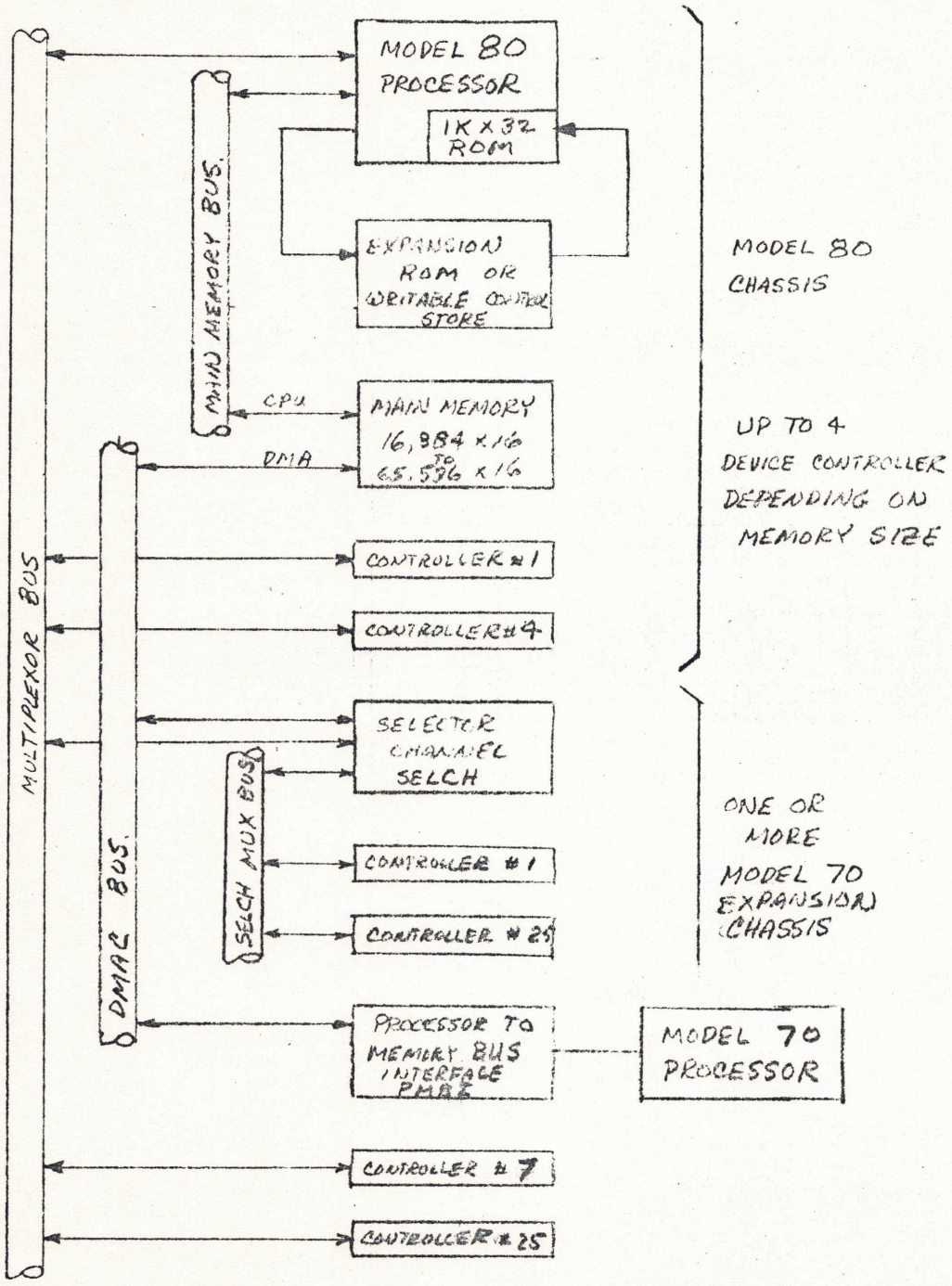
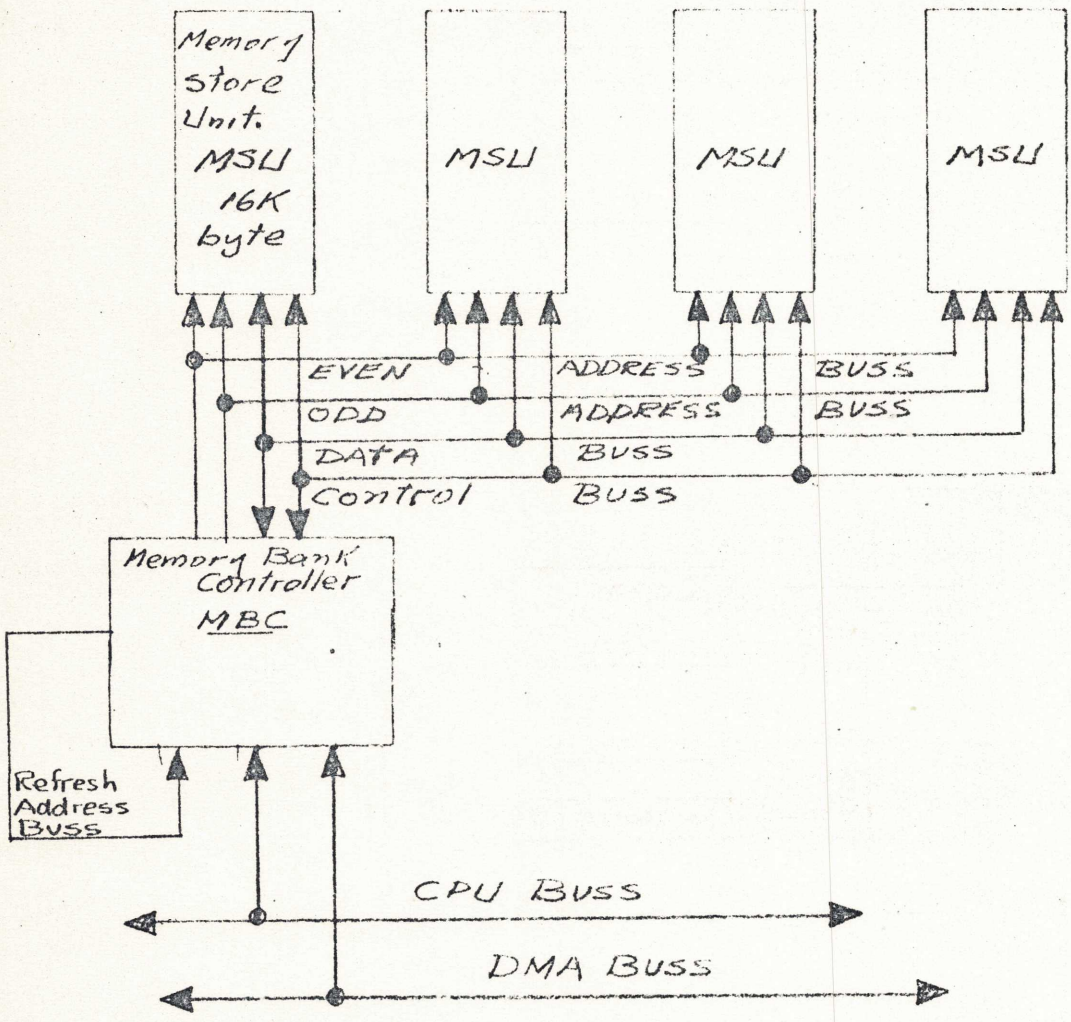
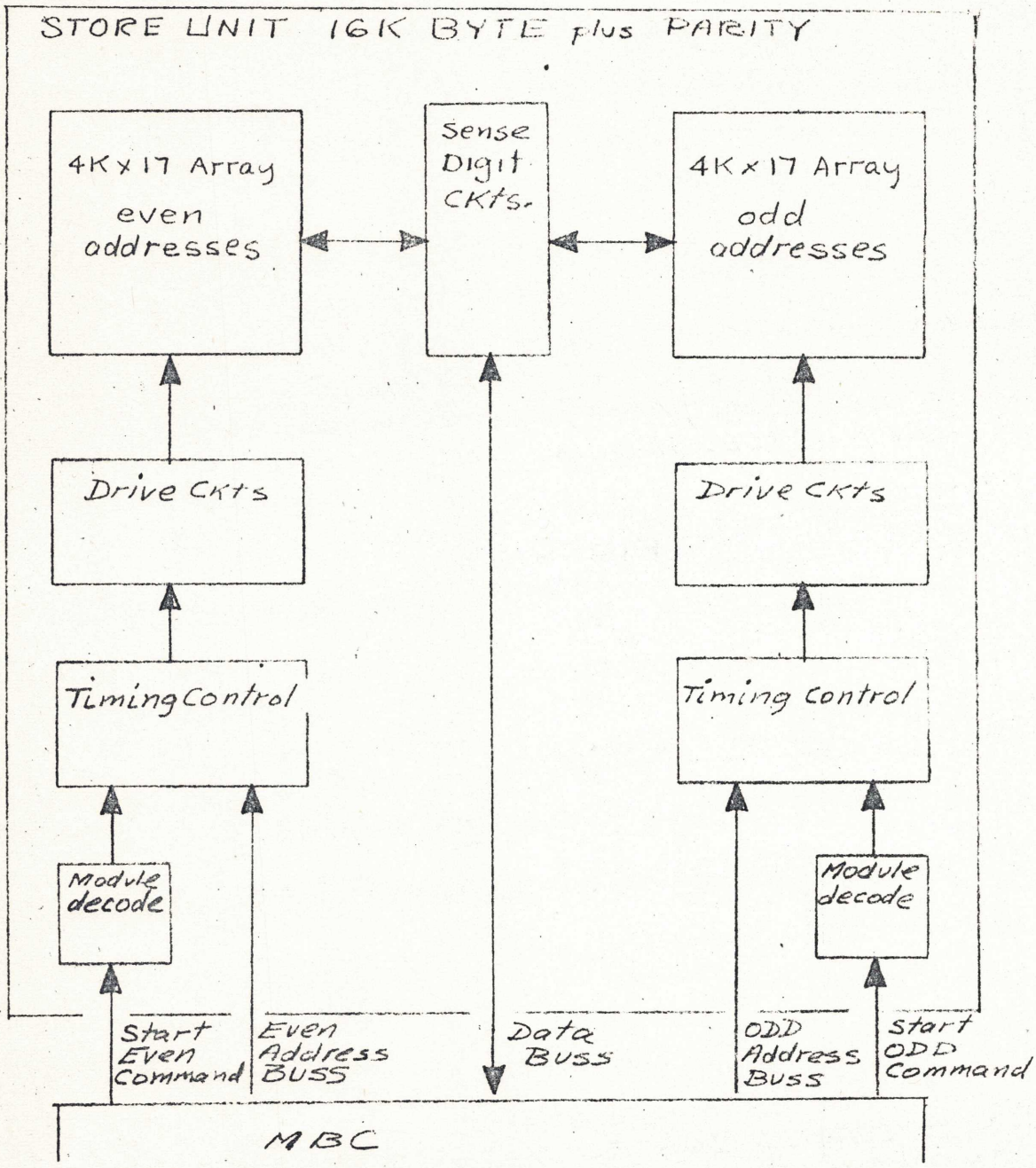


FIG 6.1



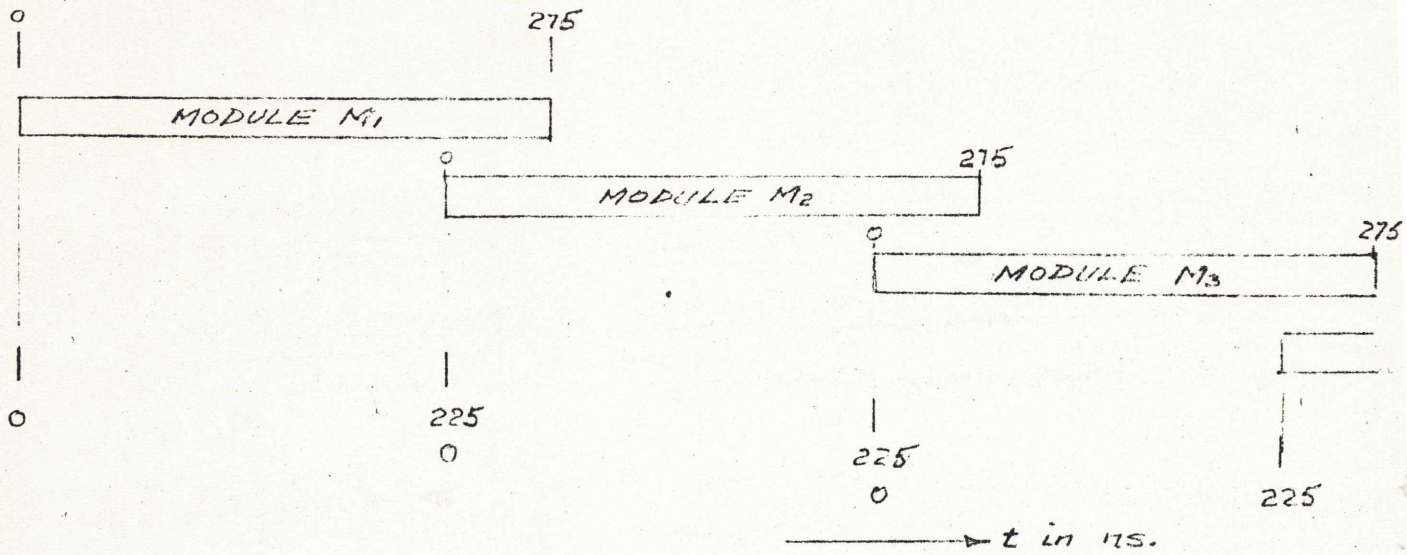
Block Diagram, Model 80 Memory Bank, Fully Expanded

FIG 6.2



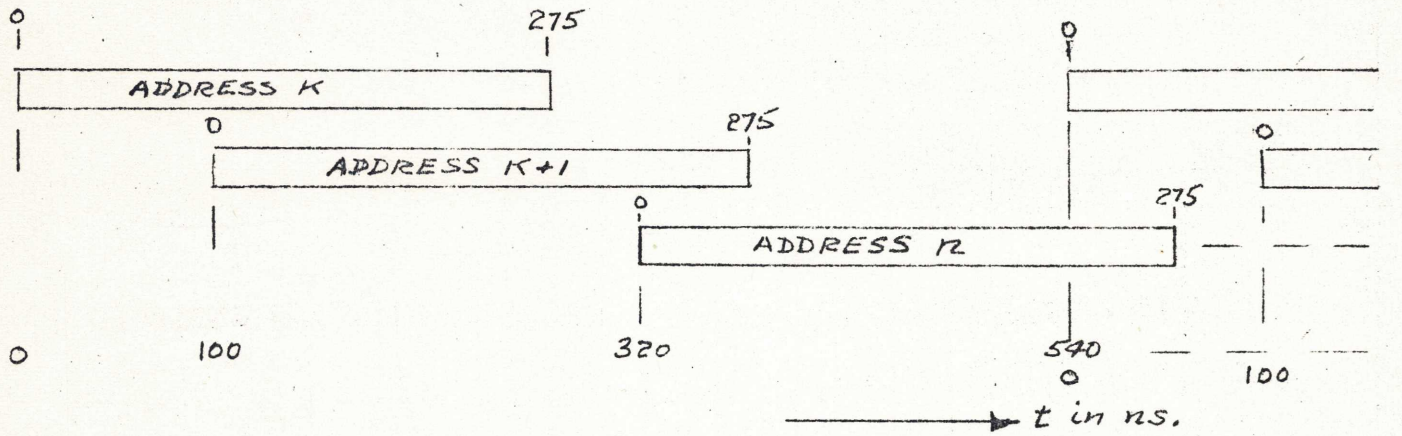
Block Diagram, Memory Store Unit.

FIG 6.3



RANDOM MEMORY SYSTEM CYCLE OVERLAP PROVIDING EFFECTIVE 220 ns. CYCLE TIME WHEN EACH MODULE $M_1, M_2 - - - M_8$, ACCESSED IN ANY SEQUENCE, HAS NOT BEEN ACCESSED IN THE PREVIOUS 275 NS.

FIG 6.3



MAXIMUM MEMORY SYSTEM HALF WORD CYCLE RATE OF 5.5 MHz, OBTAINED WITH CPU REQUESTING INSTRUCTION FETCHES INTERLACED WITH RANDOM ACCESSES FROM THE DMA BUSS.

FIG 6.4

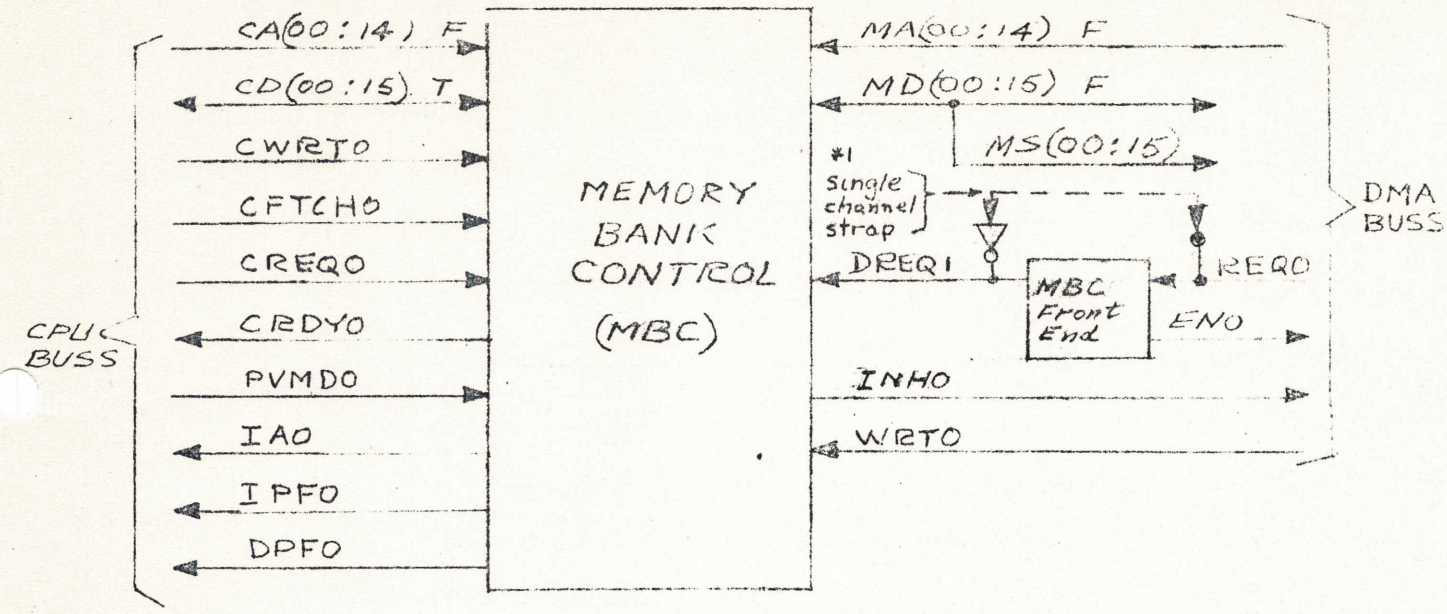


FIG 6.5 MEMORY PORT INTERFACES

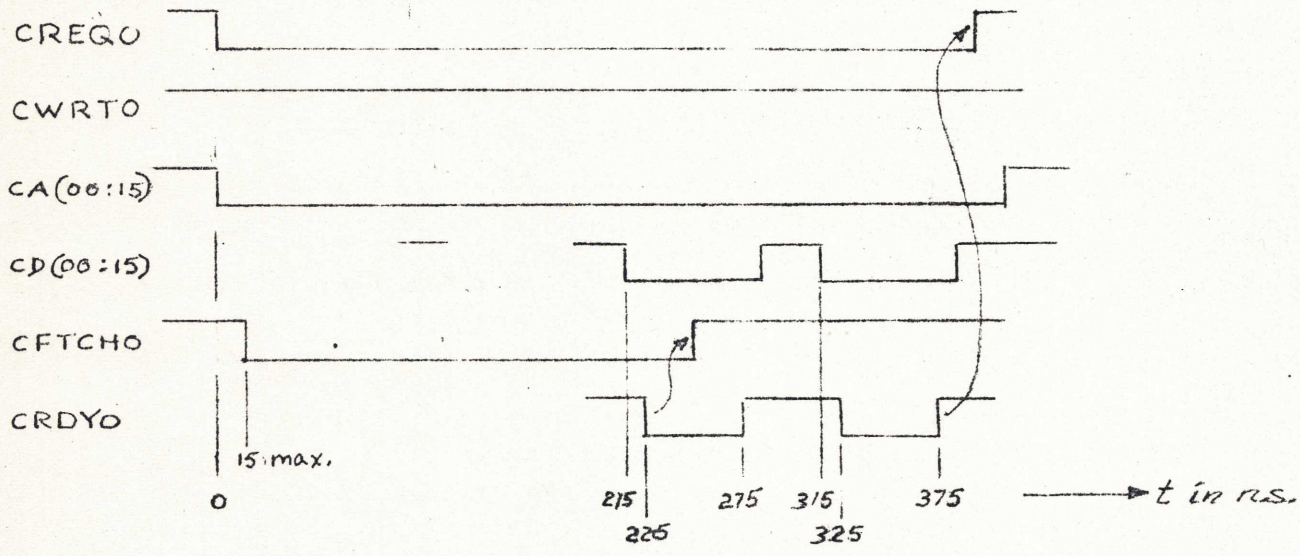


FIG 6.6 CPU INSTRUCTION FETCH CYCLE

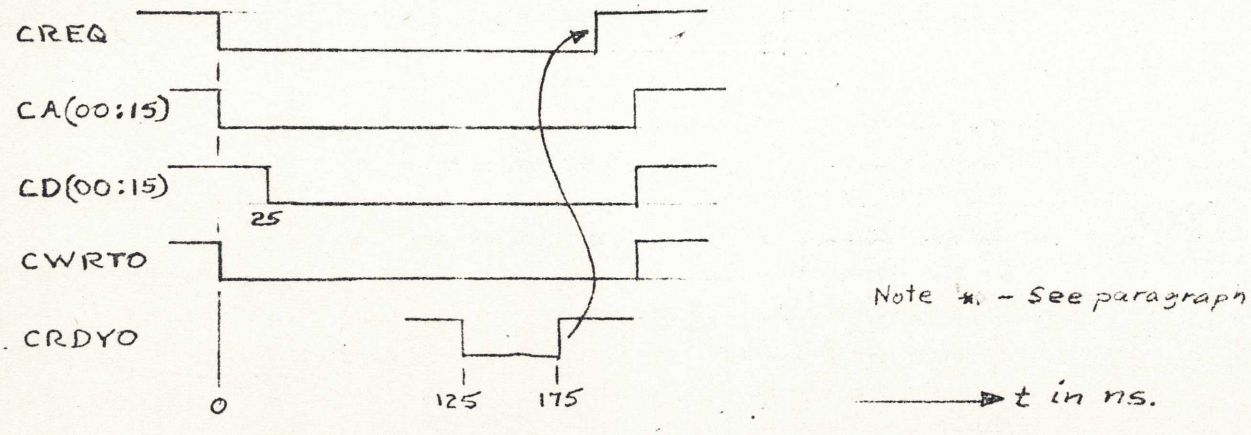
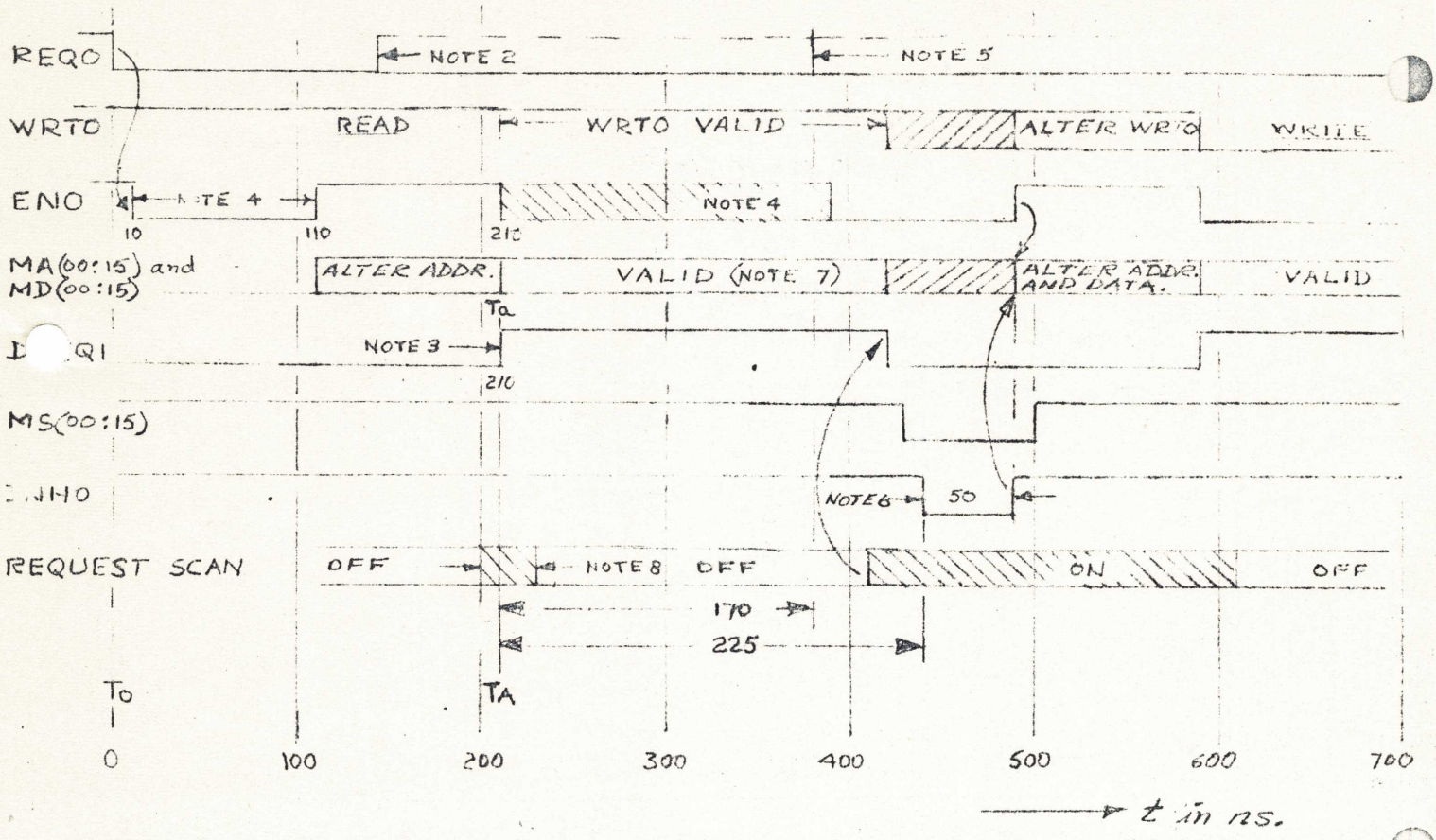


FIG 6.7 CPU MEMORY WRITE CYCLE



NOTES:

1) REQO may fall at any time. The delay from REQO to ENO will be approx. 10 ns.

2) The DMAC which captures ENO must remove REQO before the start of memory timing TA

3) Memory timing begins at TA, 210 ns. following REQO at To. This allows 10 ns for ENO to fall, 100 ns. for DMAC capture and 100 ns. for address and write data to be altered.

4) ENO will be approx. 100 ns. minimum to 280 ns. depending on the concurrent status of memory.

5) The latest time following TA which the DMAC can request to insure capturing the next cycle.

6) The fall of INHO at Tn + 225 ns. indicates that the MS(00:15) lines have valid read data or that in the case of write the memory cycle is completed. INHO is 50 ns wide and its trailing edge shall cause current data on the MD(00:15) and MA(00:15) to be removed

7) The MA(00:15) and MD(00:15) lines shall be valid for 200 ns. following TA and shall alter in less than 100 ns following ENO

8) The memory begins scanning for requests following Tn+200 ns. until a valid memory bank request is received. Once captured by the DMA bus one of two priority modes may be selected.

FIG. 6.8 MODEL 80 DMA PORT TIMING

SPECIAL CONFIGURATIONS

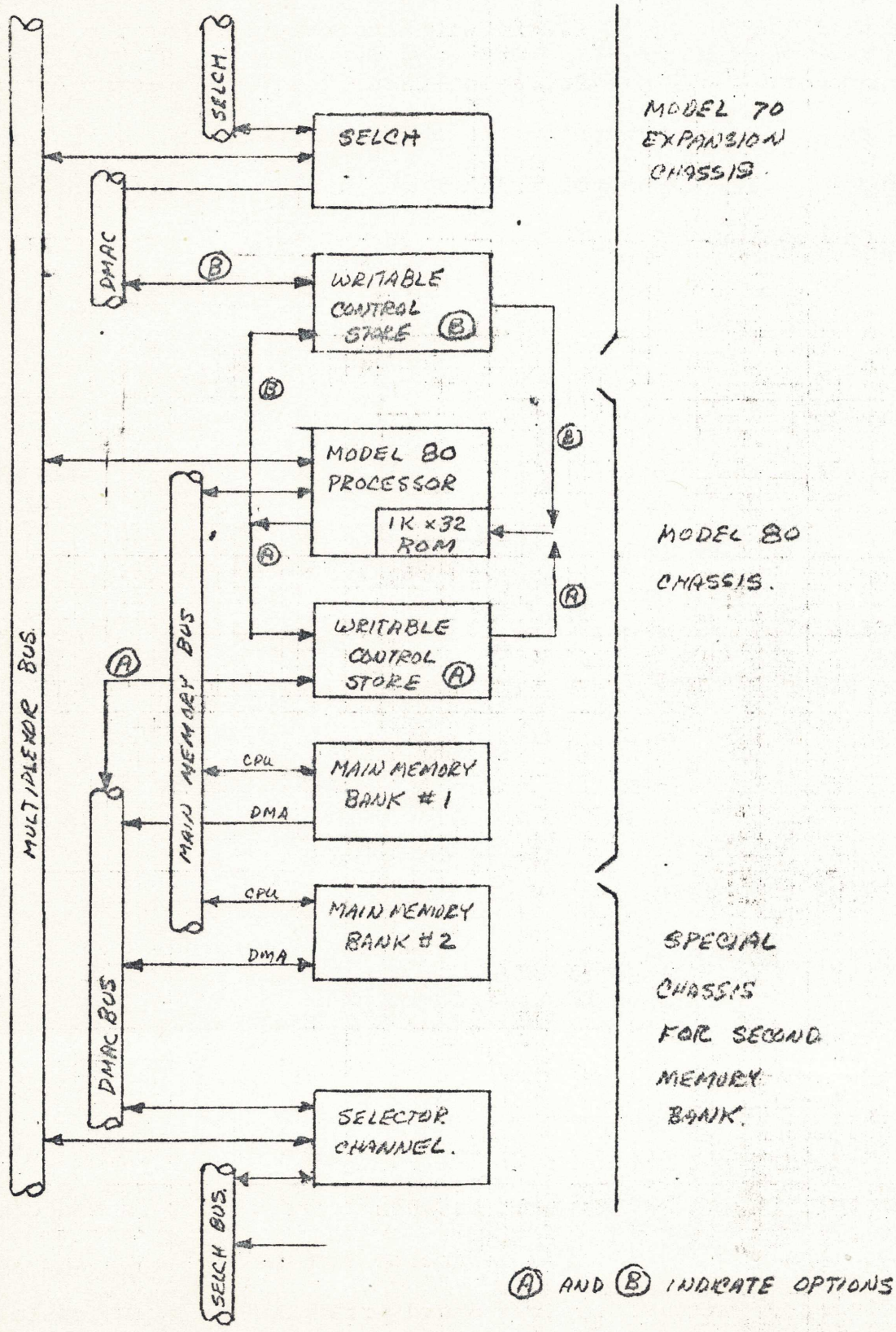


FIGURE 6.9

7.1 Configuration and Power

Standard Model 80 packaging and configuration is completely compatible with the Model 70 system with the exception of the backpanel interconnect on the Basic Processor Chassis itself. Both 70 and 80 systems use the same expansion chassis and expansion cables and the SELCH and all device controllers are the same except for the Memory Protect Controller.

7.2 Basic Processor Chassis

The Basic Model 80 chassis, see Figure 7-1, contains the Processor in the three top slots, main memory up to 65,536 bytes, and one, two or three additional locations for a Dynamic Control Store (DCS), expansion ROM, or I/O depending on Memory size. The CPU and ALU are a double board arrangement where only the CPU plugs into the backpanel. The ALU always occupies the slot below the CPU. The two center slots are wired for a Memory Bank Controller (MBC) or a DCS. The Memory Storage Units (MSU) occupy the slots immediately below the MBC and connect to the MBC by a front ribbon cable. The MSU's do not use back panel wiring. A full 64K byte memory will occupy 5 slots thus filling the basic chassis with processor and memory. The bottom three slots in the basic chassis are wired for standard I/O (3-15" I/O or 6-7" I/O), but may not be usable if MSU's occupy the space.

7.3 Special Maintenance Considerations

The ALU board is attached to the underside of the CPU with approximately 100 connector pins between boards. I.C.'s and components on the ALU are located on the bottom side so they are accessable for maintenance and the two board arrangement is pluggable. Both boards can be put on a single extender.

The five board Main Memory (MMBC and four MSU's) is interconnected by a flat, flexible ribbon cable. All components are located on the top side of the respective boards. Individual MSU's can be connected to the MBC while outside of the chassis for maintenance access. Also, MSU's can be relocated in the stack.

7.4 Power Supplies

The Model 80 system uses a separate power supply for the semiconductor main memory, with battery backup facilities to maintain the memory for up to several minutes. Hold up is sufficient to ride out fairly serious line faults and/or bring up long term stand-by power. Long term stand-by power is the responsibility of the user. It can be provided at a 115/230 VAC 50-60Hz input to the memory power supply. Stand-by power requirements are highly system dependent, the main memory itself consumes less than 5 watts but the cabinet heat rises due to hot castings immediately after power fail may require considerable fan power. Power for the processor and I/O device controllers is provided by one or more INTERDATA bulk regulators.

Note that Writable Control Memory does not have stand-by power provisions. To save the contents of Writable Control Memory, it must be transferred to Main Memory (or disk, drum, etc.) during the 5 millisecond holdup time of the bulk supply. Thepower consumption of the Model 80 boards is given in Table 7-1. Physical dimensions of the power supplies is shown in Figure 7-2.

		BULK P5 Amps	MEMORY P5M Amps	MEMORY P25M Amps
CPU		5		
ALU		7		
IOU		2		
MMBC		2	1	
ISU			1.5	2
MSU			1.5	2
MSU			1.5	*
MSU			1.5	*
EXP ROM	1024 X 32	5		
	2048 X 32	9		
	3072 X 32	13		
WCM	256 X 32	5		
	512 X 32	9		
	768 X 32	13		
	1024 X 32	17		

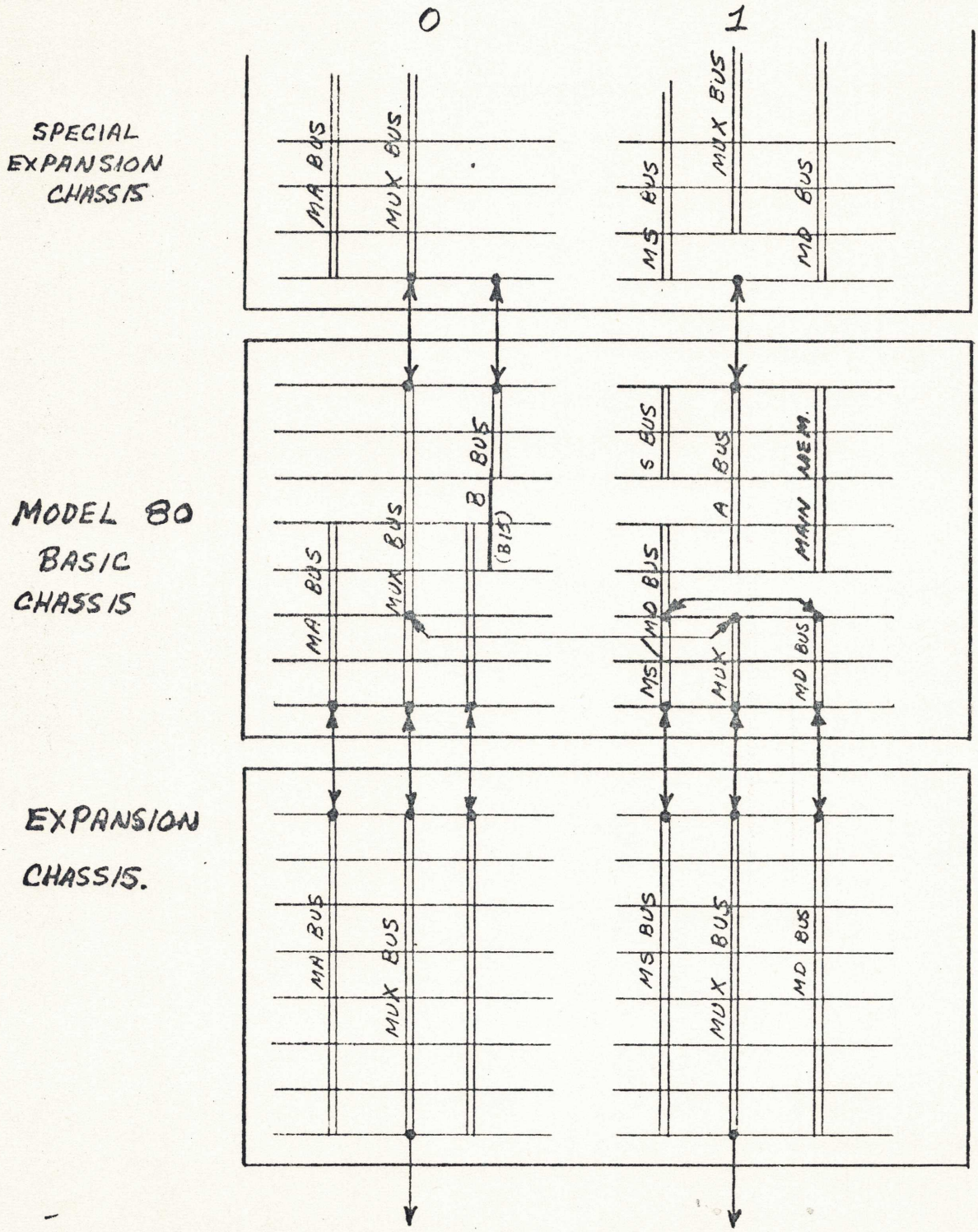
TABLE 7-1

* Two MSU's are active at any given time.

BY..... DATE..... SUBJECT..... SHEET NO..... OF.....

CHKD. BY..... DATE..... JOB NO.....

MODEL 80 CONFIG



BACK PANEL BUS CONFIGURATION
(REAR VIEW)

QTIONS.

H OR OTHER DMA CONTROLLER

SPECIAL DMAC TO WCM ONLY

OR IOU

CPU

ALU

C WCM OR EXP. ROM

MMBC

MSU OR D.C.

MSU D.C

MSU D.C

OR D.C

D.C

D.C

D.C

D.C

D.C

D.C

OR D.C

FIG 7-1

8.1 Maintenance Facilities

(MOTTS BOX et al ... will be written later.)

9.1 Peripheral Device Controllers

(Same as Model 70)

10.1.0 User Programming - The Model 80 is upward compatible from the Model 3, Model 4, Model 5., and Model 70. User programming, therefore, in most respects, is identical. In order to manipulate the dynamic control store (DCS) of the Model 80, two new user instructions will be provided.

10.2.0 Dynamic Control Store Utilization - In order to effectively use the dynamic control store option, the following capability will be provided.

10.2.1 A new instruction, Execute Control Store, will be added to the Mod 80 that will be executed only if an expansion control store is connected.

ECS R1, A(X2) [RS]

E9	R1	X2	A

On execution of this instruction, the micro-processor will use the R1 field as an extended op code and vector through a table set up by the user to up to 16 unique entry points in the control store. Thus, a user can "invent" up to 16 new instructions (of course, any of these "new" instructions could further decode the user level instruction in micro code, if more than 16 are desired).

The X2 and A fields can be interpreted as arguments, pointers, etc., and by the particular "instruction" being implemented by the Dynamic Control Store programmer.

Notice that the ECS instruction does not directly reference control store addresses and thus, when properly coded, cannot crash the system when operating in the non-supervisor mode.

10.2.2 One other new instruction, Command Control Store, will be added. It will be a privileged instruction and allows direct manipulation of, and access to, the control store.

CCS R1, A(X2) [RS]

E8	R1	X2	A

The micro-machine interprets the R1 field as an op code extension and will perform the following operations:

R1 = 0: Write block to control store
1: Read block from control store
2: Write block to machine registers
3: Read block from machine registers
4: Branch to control store

Others may be added if required.

10.2.3 To generate and debug Dynamic Control Store, the following minimum configuration is required:

- Dynamic Control Store
- TTY
- 16K byte Main Memory

A debug package (similar to CLUB) will be available for interactive debugging(on the TTY) of the micro-code.

MODEL 80 EXECUTION TIMES

Type	Instruction	Execution Time Mnemonic	In usec	Comments
Load and Store Instructions	Load Halfword (RR)	LHR	.45	
	Load Halfword	LH	.9	
	Load Halfword Immediate	LHI	.45	
	Load Immediate Short	LIS	.45	
	Load Complement Short	LCS	.45	
	Load Multiple	LM	1.25+.4n	n=no. of regs.
	Store Halfword	STH	1.3	
	Store Multiple	STM	1.45+.4n	n=no. of regs.
Fixed Point Arithmetic Instructions	Add Halfword (RR)	AHR	.45	
	Add Halfword	AH	.9	
	Add Halfword Immediate	AHI	.45	
	Add Immediate Short	AIS	.45	
	Add Halfword to Memory	AHM	1.35	
	Add with Carry Halfword (RR)	ACHR	.45	
	Add with Carry Halfword	ACH	.9	
	Subtract Halfword (RR)	SHR	.45	
	Subtract Halfword	SH	.9	
	Subtract Halfword Immediate	SHI	.45	
	Subtract Immediate Short	SIS	.45	
	Subtract with Carry Halfword (RR)	SCHR	.45	
	Subtract with Carry Halfword	SCH	.9	
	Compare Halfword (RR)	CHR	1.25/1.85	
	Compare Halfword	CH	1.25/1.85	
	Compare Halfword Immediate	CHI	2.1	
	Multiply Halfword (RR)	MH	2.1	
	Multiply Halfword	MH	2.55	
	Multiply Halfword Unsigned (RR)	MHUR	2.25	
	Multiply Halfword Unsigned	MHU	2.55	
	Divide Halfword (RR)	DHR	2.750	
	Divide Halfword	DH		
Floating Point Instructions	Add (RR)	AER	13.5	
	Add	AE	11	
	Subtract (RR)	SER	12	
	Subtract	SE	12	
	Compare (RR)	CER	4	
	Compare	CE	4	
	Multiply (RR)	MER	19.5	
	Multiply	ME	19.1	
	Divide (RR)	DER	30	
	Divide	DE	30	
	Load (RR)	LER	2	
	Load	LE	2	
	Store	STE	2	

Fixed Point Logical Instructions	AND Halfword (RR)	NHR	.45	
	AND Halfword	NH	.9	
	AND Halfword Immediate	NHI	.45	
	OR Halfword (RR)	OHR	.45	
	OR Halfword	OH	.9	
	OR Halfword Immediate	OHI	.45	
	Exclusive OR Halfword (RR)	XRH	.45	
	Exclusive OR Halfword	XH	.9	
	Exclusive OR Halfword Immediate	XHI	.45	
	Compare Logical Halfword (RR)	CLHR	.45	
	Compare Logical Halfword	CLH	.9	
	Compare Logical Halfword Immediate	CLHI	.45	
	Test Halfword Immediate	THI	.45	
Shift Instructions	Shift Right Halfword Logical	SRHL	.5+.1n	n=no. of Shifts
	Shift Right Short Logical	SRSL	.3+.1n	n-no. of Shifts
	Shift Right Logical (Fullword)	SRL	.7+.1n	n=no. of Shifts
	Shift Left Halfword Logical	SLHL	.5+.1n	n=no. of Shifts
	Shift Left Short Logical	SLSL	.3+.1n	n=no. of Shifts
	Shift Left Logical (Fullword)	SLL	.7+.1n	n=no. of Shifts
	Shift Right Halfword Arithmetic	SRHA	.5+.1n	n=no. of Shifts
	Shift Right Arithmetic (Fullword)	SRA	.7+.1n	n=no. of Shifts
	Shift Left Halfword Arithmetic	SLHA	.5+.1n	n=no. of Shifts
	Shift Left Arithmetic (Fullword)	SLA	.7+.1n	n=no. of Shifts
	Rotate Right Logical (Fullword)	RRL	.7+.1n	n=no. of Shifts
	Rotate Left Logical (Fullword)	RLL	.7+.1n	n=no. of Shifts
Byte Handling Instructions	Load Byte (RR)	LBR	.45	
	Load Byte	LB	.9	
	Store Byte (RR)	STBR	.45	
	Store Byte	STB	1.35	
	Exchange Byte (RR)	EXBR	.45	
	Compare Logical Byte	CLB	1.05	
Branch Instructions	Branch on True Condition (RR)	BTCR	.45/.85	*
	Branch on True Condition	BTC	.45/.85	
	Branch on True Condition Forward	BTFS	.45/1.45	
	Branch on True Condition Backward	BTBS	.45/1.45	
	Branch on False Condition (RR)	BFCR	.45/.85	
	Branch on False Condition	BFC	.45/.85	
	Branch on False Condition Forward	BFFS	.45/1.45	
	Branch on False Condition Backward	BFBS	.45/1.45	
	Branch AND Link (RR)	BALR	1.05	
	Branch AND Link	BAL	1.05	
	Branch on Index High	BXH	2.3/2.95	
	Branch on Index Low or Equal	BXLE	2.3/2.95	

*no branch/branch

Extended Mnemonics Instructions	Branch on Zero	BZ	.45/.85	*
	Branch on Not Zero	BNZ	.45/.85	
	Branch on Plus	BP	.45/.85	
	Branch on Not Plus	BNP	.45/.85	
	Branch on Minus	BM	.45/.85	
	Branch on Not Minus	BNM	.45/.85	
	Branch on Carry	BC	.45/.85	
	Branch on Overflow	BO	.45/.85	
	Branch on Low	BL	.45/.85	
	Branch on Not Low	BNL	.45/.85	
	Branch on Equal	BE	.45/.85	
	Branch on Not Equal	BNE	.45/.85	
	No Operation (RR)	BOPR	.45/.85	
	No Operation	NOP	.45/.85	
Status Control Instructions	Load Program Status Word	LPSW	2.15	
	Exchange Program Status (RR)	EPSR	.85	
	Supervisor Call	SVC	3.8	
List Handling Instructions	Add to Top of List	ATL	4.3/4.6	4.6 List Wrap
	Add to Bottom of List	ABL	4.3/4.6	4.6 List Wrap
	Remove from Top of List	RTL	4.5/4.8	4.8 List Wrap
	Remove from Bottom of List	RBL	4.5/4.8	4.8 List Wrap
Input-Output Instructions	Autoload	AL		
	Simulate Interrupt	SINT		
	Read Data (byte) (RR)	RDR	1.1	**
	Read Date (byte)	RD	2.0	
	Read Halfword (RR)	RHR	1.1	
	Read Halfword	RH	2.0	
	Read Block (RR)	RBR	1.0 MB	
	Read Block	RB	1.0 MB	
	Write Data (byte) (RR)	WDR	1.1	
	Write Data (byte)	WD	2.0	
	Write Halfword (RR)	WHR	1.1	
	Write Halfword	WH	1.45	
	Write Block (RR)	WBR	1.0 MB	
	Write Block	WB	1.0 MB	
	Sense Status (RR)	SSR	1.1	
	Sense Status	SS	2.0	
	Output Command (RR)	OCR	1.1	
	Output Command	OC	2.0	
	Acknowledge Interrupt (RR)	AIR	1.1	
	Acknowledge Interrupt	AI	2.0	

* no branch/branch

** no address/with address user level always with address