Publication Number B29-253

MODEL 70 MICRO-INSTRUCTION
REFERENCE MANUAL

INTERDATA®
2 CRESCENT PLACE, OCEANPORT, NEW JERSEY 07757 | (201) 229-4040

MODEL 70
MICRO-INSTRUCTION
REFERENCE MANUAL

1. INTRODUCTION TO MICRO-PROGRAMMING

The INTERDATA Processor is a very fast micro-programmable machine controlled by a Read-Only-Memory (ROM). Micro-programs in the ROM direct the flow of information within the registers and core storage of the machine. These micro-programs are a sequence of micro-instructions. Each micro-instruction causes one machine function to be executed; such as transferring the contents of one register to another, addition, subtraction, and logical operations between registers; input/output operations; or core memory accesses. Each micro-instruction constitutes one 16-bit word of ROM. The ROM is a high speed, non-volatile, solid-state device organized into pages of 256 words each.

2. THE MICRO-PROGRAMMED PROCESSOR

The Processor is designed to execute micro-instructions stored in a Read-Only-Memory (ROM). Each micro-instruction performs one hardware function. A series of these micro-instructions (a micro-program) can solve highly complex problems. The Micro-Program in the Processor is an emulator. That is, the program completely simulates the instruction set, interrupt handling features, and display functions of a quite different machine. This emulated computer is similar to the IBM 360 family of Processors and has a very powerful instruction set.

Special hardware is built into the Processor to assist the micro-program in its emulation task. When executing instructions of the emulated machine, the micro-program directs the hardware to read the instruction to be performed from core memory. The instruction decoding circuitry then steers the Processor to a micro-subroutine that has been designed to perform the emulated instruction. The loop is then closed by incrementing the instruction location counter and returning to the point that will fetch the next instruction from core memory.

The Read-Only-Memory is a very fast, non-volatile, solid-state device. The Processor is designed to match the ROM's performance, executing most micro-instructions in 250 nanoseconds. The combination of special hardware and micro-instructions running at many times core speed allows most emulated instructions to be executed in the time it takes to fetch them from core.

3. BLOCK DIAGRAM ANALYSIS

Refer to Figure 1 during the following description. Processor operation is controlled by the Read-Only-Memory (ROM). Locations in ROM are addressed by a 12-bit register comprising an 8-bit incrementing address register (RAL) and a 4-bit non-incrementing page register (RAS). RAS is loaded from the "outer rank" page register (RAH) when RAL is loaded.

Information read out of ROM is placed in a 16-bit data register (RD). Bits 0 through 3 of RD are the micro-instruction operation code. The meaning of the remaining bits in RD depends on the particular micro-operation to be performed.

There are three 16-bit general purpose micro-registers labeled MR0, MR1, and MR2. Two additional 16-bit micro-registers, SRH and SRL, are available for general use. SRH and SRL may also be used together as a 32-bit shift left/shift right register for multiply, divide, and double-precision shifts.

The Program Status Word (PSW) is a 16-bit register which indicates the system status relative to the user program being emulated. Bits 0 through 11 of PSW define machine status. Bits 12 through 15, the Condition Code (CC), may only be loaded from the Flag Register (FLR). When PSW is loaded, Bits 12 through 15 of the S Bus are captured in the FLR. This permutes the user status to the micro-code level. The Location Counter (LOC) is a 16-bit appendum to PSW which holds the address of the next user instruction to be performed.

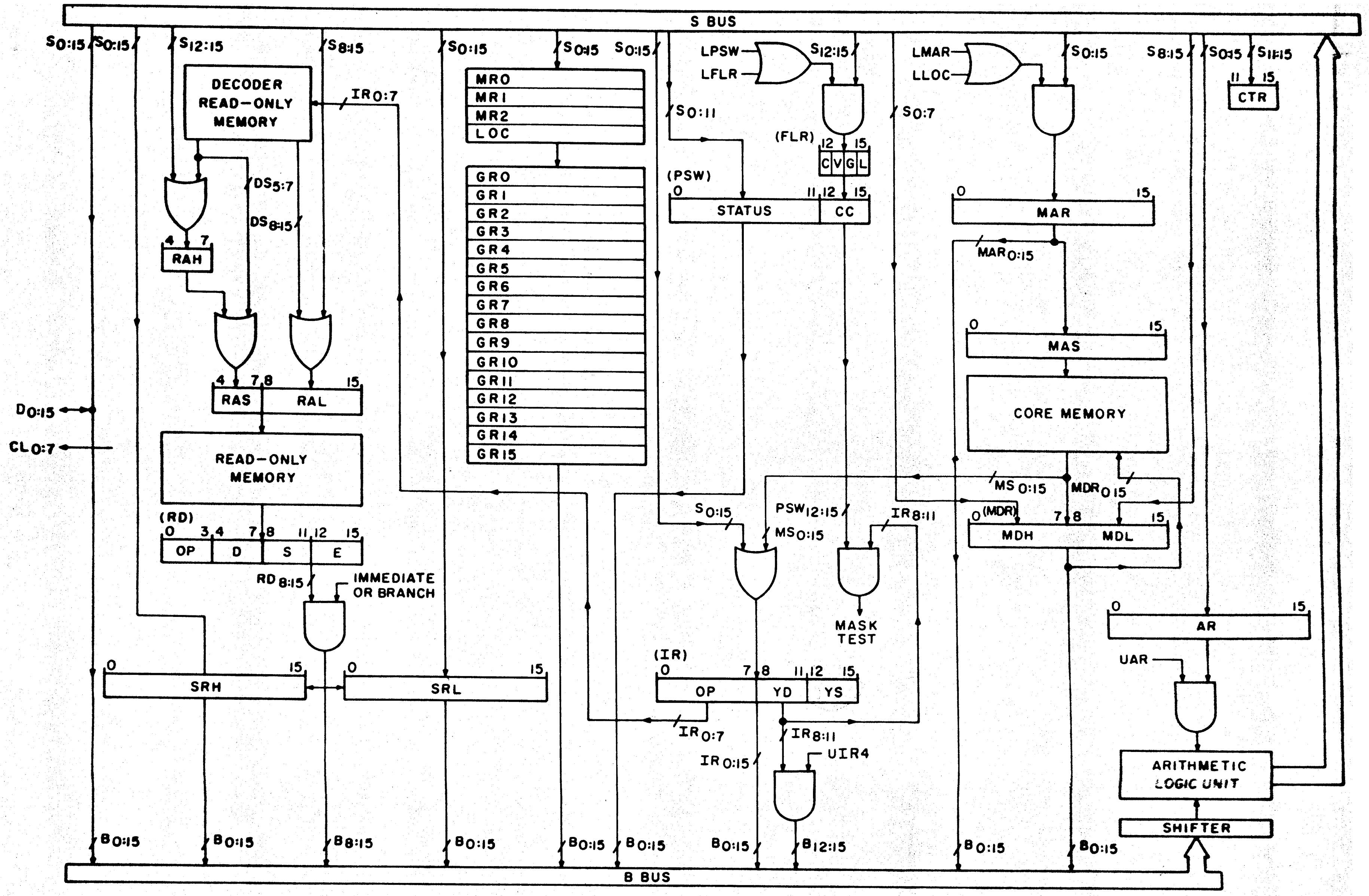

Figure 1. Hardware Block Diagram

The Memory Address Register (MAR) is a 16-bit buffer used to address core memory locations. Memory is actually addressed by the Memory Address Slave Register (MAS). When memory is not busy, MAS is automatically loaded from MAR. The 16-bit Memory Data Register (MDR) holds data read from or written into core memory. MDR is separated into two bytes, MDH and MDL, which may be loaded separately for efficient byte handling.

The 16-bit Instruction Register (IR) holds the instruction word of the current user instruction. Bits 8 through 11 may be unloaded to the B Bus Bits 12 through 15 (IR4).

The user's sixteen 16-bit General Registers (GR0 through GR15) are not directly addressable by the micro-program. The General Registers are addressed by IR Bits 8 through 11 (YD) or by IR Bits 12 through 15 (YS). The micro-program can only access the General Registers by specifying the field of IR, YD, or YS, that contains the appropriate General Register address.

The Counter Register (CTR) is a 5-bit decrementing register. It is always initialized to a value of 16; however, it may be preset to any value from 0 to 31 to control the number of repetitions of a single micro-instruction or a sequence of micro-instructions. The Counter is used in the multiply and divide hardware to control the number of shifts.

The 16-bit A Register (AR) holds the second operand for arithmetic and logical micro-operations. It is one of two direct inputs to the Arithmetic Logic Unit (ALU). The other input is the output of the B Bus Shifter. The Shifter can shift B Bus data left or right one position, rotate eight positions, or gate directly into the ALU. The ALU comprises a 16-bit parallel adder/subtracter logic network. The arithmetic or logical result is gated to the 16-bit S Bus.

Bits 0 through 7 of IR, the user's operation code, are used to address locations in the Decoder Read-Only-Memory (DROM). DROM is a separate Read-Only-Memory that has 256 12-bit words. DROM is interrogated prior to entering the micro-subroutine that executes a user instruction. The DROM readout is jammed into the ROM address registers. The DROM readout corresponds to the starting micro-program address for a given user instruction.

Input/Output operations are accomplished by activating one of eight control lines and gating the S Bus onto the Data Lines or gating the Data Lines onto the B Bus.

4. WORD FORMATS

4.1 General

The Processor executes 16 basic micro-instructions (see Table 1). Bits 0 through 3 of the ROM Data Register (RD) specify the operation to be performed.

TABLE 1. BASIC MICRO-INSTRUCTIONS

OP-CODE	MICRO-INSTRUCTION	
0123	Symbolic	Meaning
0000	D	DO
0001	C	COMMAND
0010	T	TEST
0011	B	BRANCH
0100	L	LOAD
0101	L	LOAD IMMEDIATE
0110	O	OR
0111	O	OR IMMEDIATE
1000	N	AND
1001	N	AND IMMEDIATE
1010	X	EXCLUSIVE OR
1011	X	EXCLUSIVE OR IMMEDIATE
1100	A	ADD
1101	A	ADD IMMEDIATE
1110	S	SUBTRACT
1111	S	SUBTRACT IMMEDIATE

Micro-instructions can have any one of four machine language formats, depending on the operation specified by the OP-CODE. The formats are listed in the following paragraphs by the micro-instructions which use them.

4.2 Add, Subtract, And, Or, Exclusive Or, and Load

0 3	4 7	8 11	12 15
OP-CODE	D	S	E

Bit 3 of the Op-Code in this format is always reset.

D = Destination field. The result of the operation is placed into the register whose address is in this field.
S = Source field. The address of the register containing the first operand is in this field. The second operand is contained in the A Register (AR).
E = Extended operation field. Instruction options are specified in this field.

4.3 Add Immediate, Subtract Immediate, And Immediate, Or Immediate, Exclusive Or Immediate, and Load Immediate

0 3	4 7	8 15
OP-CODE	D	DATA

Bit 3 of the op-code in this format is always set.

D = Destination field. The result of the operation is placed into the register whose address is in this field.
DATA = The first operand is contained in this field. The second operand is contained in the A Register (AR).

4.4 Do, Command, and Test

0 3	4 15
OP-CODE	FUNCTION

FUNCTION = Specifies the function to be performed or tested.

4.5 Branch

0 3	4 7	8 15
OP-CODE	CONDITION	ADDRESS

CONDITION = specifies the condition to be tested.
ADDRESS = if the specified condition is true, the program is transferred to the address specified by this field.

5. SOURCE AND DESTINATION REGISTERS

Source registers are only available to non-immediate micro-instructions, RD Bit 3 = 0. The sources that may be addressed are shown in Table 2.

TABLE 2. ADDRESSABLE SOURCES

RD Bits	SYMBOLIC REGISTER	RD Bits	SYMBOLIC REGISTER
8 9 10 11		8 9 10 11	
0 0 0 0	MR0	1 0 0 0	NULL
0 0 0 1	MR1	1 0 0 1	IR
0 0 1 0	MR2	1 0 1 0	MDR
0 0 1 1	LOC	1 0 1 1	IO
0 1 0 0	PSW	1 1 0 0	IR4
0 1 0 1	MAR	1 1 0 1	YS
0 1 1 0	SRH	1 1 1 0	YD
0 1 1 1	SRL	1 1 1 1	YDP1

The source registers may be used freely in the micro-program. The following paragraphs point out special cases.

MAR can be used as a source register at any time. However, MAR may or may not contain the same data as MAS depending on the state of memory busy. In the latter case, when memory becomes un-busy, whatever is in MAR is copied into MAS.

SRH and SRL are special purpose shift registers. The Command micro-instruction specifying Multiply, Divide, Shift Right or Shift Left, uses SRH and SRL as a 32-bit shift register; SRH containing the most significant 16 bits and SRL the least significant 16 bits. When the NULL source, Address '8', is used, zeros are gated to all 16 bits of the B Bus. When MDR is specified as the source, the execution of the micro-instruction is suspended until memory data is available.

When I/O (Address 'B') appears as the source, an input operation is to be performed. Only in a Load micro-instruction can I/O be a source. The nature of the input request is encoded into the Extended Operation Field of the Load instruction. When the device responds, the data is gated onto the B Bus. Completion of the micro-instruction is suspended until the device responds.

Specifying IR4 causes Bits 8 through 11 of IR to be gated onto the B Bus Bits 12 through 15. Zeros are gated onto B Bus Bits 0 through 11.

The user's 16 general purpose registers do not have individual source addresses. Instead, common symbolic addresses - YS (Address 'D'), YD (Address 'E') and YDP1 (Address 'F') cause the General Registers to be selected from IR Bits 8 through 11 or 12 through 15. When YS appears as the source, the General Register whose address is in IR Bits 12 through 15 is gated onto the B Bus. When YD appears as the source, the General Register whose address is in IR Bits 8 through 11 is gated onto the B Bus. When YDP1 (User's Destination plus 1) appears as the source, the odd member of the even/odd pair of General Registers addressed by IR Bits 8 through 11 is gated onto the B Bus.

The destinations that may be addressed are shown in Table 3.

TABLE 3. ADDRESSABLE DESTINATIONS

RD BITS	SYMBOLIC REGISTER
4 5 6 7	
0 0 0 0	RAH * MR0 **
0 0 0 1	RAL * MR1 **
0 0 1 0	PSW * MR2 **
0 0 1 1	LOC
0 1 0 0	FLR
0 1 0 1	MAR
0 1 1 0	SRH
0 1 1 1	SRL
1 0 0 0	AR
1 0 0 1	IR
1 0 1 0	MDR
1 0 1 1	IO
1 1 0 0	CTR
1 1 0 1	YS
1 1 1 0	YD
1 1 1 1	YDP1

* The Bank flip-flop must be reset to load RAH, RAL, or PSW
** The Bank flip-flop must be set to load MR0, MR1, or MR2

The destination registers may be used by the micro-program noting the following restrictions or special cases.

Loading RAH loads only the outer rank page register of the ROM address register. The ROM decodes the page from the inner rank (RAS). When RAL is loaded, RAH is copied into RAS so that all 12 bits of ROM address change simultaneously.

When PSW is loaded, Bits 12 through 15 of the S Bus are captured in the FLR. PSW Bits 12 through 15 (the condition code) can only be loaded from the FLR on a Do micro-instruction.

When LOC is loaded, MAR is loaded simultaneously.

Anytime memory is not Busy, MAR is copied into MAS.

Loading the FLR in an instruction that normally sets flags causes an OR-ing of the resulting flags and S Bus data.

If an attempt is made to load MDR when memory is Busy, execution of that micro-operation is suspended until memory becomes not Busy. If MDR is the destination in a cross-shift operation, only the high byte (MDH) is loaded if MAR is even and the cross-shift function is performed. If MAR is odd, only the low byte (MDL) is loaded and the cross-shift function is not performed. The portion of the MDR (i.e. MDH or MDL) not loaded remains unchanged. If MDR is the destination and cross-shift is not exercised, both bytes (MDH and MDL) are loaded.

When I/O appears as the destination, an output operation is to be performed. Only in a Load micro-instruction can I/O be a destination. The nature of the output operation is encoded into the Extended operation field of the Load instruction. The S Bus is gated onto the Data Lines. Completion of the micro-instruction is suspended until the device responds or a false SYN is generated.

When YS is the destination, S Bus data is copied into the General Register whose address is in IR Bits 12 through 15.

When YD is the destination, the General Register whose address is in IR Bits 8 through 11 is loaded.

When YDP1 is the destination, the odd member of the even/odd pair of General Registers addressed by IR Bits 8 through 11 is loaded.

6. MICRO-INSTRUCTIONS

This section describes each micro-instruction. For each instruction, the assembler format is shown. The machine instruction format is diagrammed, a description of the instruction is provided, and options within the instruction are described.

6.1 Do

D FUNCTION

0 3	4 15
0000	FUNCTION CODE

4	5	6	7	8	9	10	11	12	13	14	15		SYMBOLIC MEANING
1	0									1		MR	MEMORY READ
0	1									1		MW	MEMORY WRITE
1	1									1		PW	PRIVILEDGED WRITE
		1								1		INC	INCREMENT LOC
			1							1		JAM	COPY FLR TO CC
				1						1		PC	PHASE CHANGE
					1					1		CLR	CLEAR
						1				1		SWA	SET WAIT
							1			1		CWA	CLEAR WAIT
								1		1		ALRM	LOAD ALARM FLAGS
										1	1	POW	POWER DOWN

The Do micro-instruction is always written with Bit 14 set.

All zeros in RD are interpreted as illegal unless RD is zero because of initialization. A blanking flip-flop (RUN) covers the initialize period. When the Processor is initialized, the ROM Address Registers and RD are cleared and the RUN flip-flop is reset. Location X'000' in ROM contains all zeros. When this instruction is read, the RUN flip-flop is still reset and the Processor does nothing. The next instruction in sequence (Address X'001') is executed and the RUN flip-flop sets. The next time all zeros occur in RD, the illegal condition will exist, forcing the Processor to Phase Three and ROM address to X'100'.

The DO micro-instruction is executed in one clock period unless a memory cycle is requested (MR, MW, PW) and memory is Busy or **phase change is specified and the new phase** is Phase zero.

In the former case, the DO micro-instruction can take realistically 1, 2, or 3 clocks depending on when the current memory cycle began. In the latter case, the time depends upon the state of Memory Busy and the type of user instruction to be performed.

The type of user instruction is identified by the most significant four bits of the op-code in IR. Table 4 shows the user repertoire. From Table 4, the following list of op-codes by category is derived.

OP CODES

RR Instructions:	0n 1n 2n 3n 8n 9n
RX Instructions:	4n 5n 6n 7n An Bn Dn Fn
RS Instructions:	Cn En

If an RR instruction is to be performed, the DO micro-instruction requires 2 or 3 clocks. If an instruction other than RR is to be performed, the DO micro-instruction requires 4 or 5 clocks.

TABLE 4. INSTRUCTION REPERTOIRE

Op Code MSD ⟶

Op Code LSD ↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			BTBS		STH		STE			SRLS			BXH	STM		
1	BALR		BTFS		BAL		AHM			SLLS			BXLE	LM	SVC	
2	BTCR		BFBS		BTC					STBR			LPSW	STB	SINT	
3	BFCR		BFFS		BFC					LBR			THI	LB		
4	NHR		LIS		NH		ATL			EXBR			NHI	CLB		
5	CLHR		LCS		CLH		ABL			EPSR			CLHI	AL		
6	OHR		AIS		OH		RTL			WBR			OHI	WB		
7	XHR		SIS		XH		RBL			RBR			XHI	RB		
8	LHR		LER		LH		LE			WHR			LHI	WH		
9	CHR		CER		CH		CE			RHR			CHI	RH		
A	AHR		AER		AH		AE			WDR			AHI	WD	RRL	
B	SHR		SER		SH		SE			RDR			SHI	RD	RLL	
C	MHR		MER		MH		ME			MHUR			SRHL	MHU	SRL	
D	DHR		DER		DH		DE			SSR			SLHL	SS	SLL	
E	ACHR				ACH					OCR			SRHA	OC	SRA	
F	SCHR				SCH					AIR			SLHA	AI	SLA	
	RR	RR	RR	RR	RX	RX	RX	RX	RR	RR	RX	RX	RS	RX	RS	RX

When phase change is specified, the new state of the Phase Pointer and the new ROM address depends upon the current phase, the type of user instruction to be executed, and whether or not an interrupt is pending. These factors are summarized in Table 5.

The Phase Pointer has four states. One of these, Phase Zero, exists in hardware only and does not have an associated set of micro-code.

In Phase Zero, the user instruction is loaded into MDR and IR simultaneously. The AR is loaded from YS and the OP flip-flop is set to reflect the length of the user instruction; 0 if RR, 1 if RX or RS. If the user instruction is RX or RS, a second memory read is started to fetch the second half of the instruction word. If the user instruction is not RR, the YS field of IR distinguishes Indexed instructions from unindexed. If the YS field is zero, the instruction is not indexed. If the user instruction is RR or unindexed RS, Phase Zero is exited and the Phase Two entry point is derived from the DROM. If the user instruction is RX or indexed RS, Phase Zero is exited and the appropriate Phase One entry point is generated. There are three micro-sequences associated with Phase One. These sequences are shown below:

```
ROM     ROM
ADRS    DATA
                *INDEXED     RS  ENTRY
0002    CAA3    AMODRS    A     MDR,MDR        INDEX 'A' FIELD
0003    00C2              D     PC+CLR         GO TO PHASE 2
                *NO INDEX    RX  ENTRY
0004    45A3    NMODRX    L     MAR,MDR        ADRS 'A'
0005    08C2              D     MR+PC+CLR      GO TO PHASE 2
                *INDEXED  RX    ENTRY
0006    C5A3    AMODRS    A     MAR,MDR        ADRS 'A' +(X2)
0007    08C2              D     MR+PC+CLR      GO TO PHASE 2
```

Second operand in MDR
AR has index

The Do micro-instruction exiting Phase One of an RX instruction would normally execute in three clocks because a memory read is requested and memory is still busy from the read cycle started as Phase Zero was exited. If the user opcode is 40,41,42, or 43, the Do micro-instruction takes only 1 clock. These user instructions (STH, BAL, BTC, and BFC) do not require a second operand so the memory read is not attempted. An unnecessary memory read does occur on STM, STE, AL, and RH instructions.

TABLE 5. DECODE PHASE CHANGES

CURRENT PHASE	INTERRUPTS ?	NEXT INSTR	INDEXED ?	NEW PHASE	NEW ROM ADDRESS
1				2	From DROM
2	Yes			3	'010'
2	No	RR		0 then 2	From DROM
2	No	RS	No	0 then 2	From DROM
2	No	RS	Yes	0 then 1	'002'
2	No	RX	No	0 then 1	'004'
2	No	RX	Yes	0 then 1	'006'
3		RR		0 then 2	From DROM
3		RS	No	0 then 2	From DROM
3		RS	Yes	0 then 1	'002'
3		RX	No	0 then 1	'004'
3		RX	Yes	0 then 1	'006'

The Phase Two sequence starting address is derived from the Decoder Read-Only-Memory. There are as many Phase Two sequences as there are unique entries in the DROM. Illegal instruction handling is explained in Section 8.

There are three Phase Three entry points: Location X'100' for the Illegal Instruction (RD=0), Location X'010' for Interrupts, and X'00A' for aborted sequences. When Phase 2 is exited, the hardware automatically tests for interrupts. The interrupts tested for are Primary Power Fail (PPF), Machine Malfunction (MALF), Console Attention (CATN), Console Single Mode (SNGL), Data Channel Request (DC), and I/O Attention (ATN). If an interrupt is found pending when the Do micro-instruction that exits Phase Two is executed, the Phase Pointer is set to Phase Three, the INC and MR specifications in the Do micro-instruction are suppressed, and ROM address is forced to X'010'.

The micro-sequence at X'010' determines the nature of the interrupt and executes the appropriate service sequence.

When Phase Two is entered, an Enable flip-flop (EBL) is set. As long as EBL is set, the occurrence of a Data Channel Request (DC) or regular I/O Interrupt (ATN) or a Machine Malfunction (MALF) will cause the user instruction sequence to be aborted. The Processor enters Phase Three and ROM address is forced to X'00A'. The micro-routine at X'00A' decrements the Location Counter by 2 if the user instruction was RR or by 4 if RX or RS, and then enters the common interrupt polling subroutine at X'010'. (Instruction type can be known by testing the OP flip-flop.) After the interrupt is serviced, the user instruction is re-started from the beginning.

The Enable flip-flop is reset whenever the micro-program loads a General Register, loads PSW or LOC, does a Memory Write or Priviledged Write or an I/O operation. When Enable resets, the instruction is not interruptable until the terminating Do micro-instruction is executed.

OPTIONS

The function code bits specify the function(s) to be performed. The Phase Change function has been described; the other functions are described below:

MR	Memory Read
MW	Memory Write
PW	Priviledged Write

A memory cycle consists of two half cycles: Read and Write; in that order. The Read cycle places the contents of the addressed location into MDR. The addressed location then contains zeros (destructive read-out). During the Write cycle, the contents of MDR are written into the addressed location.

Memory Read (MR) is a combination of the Read/Write cycles. The memory location is read-out then restored from MDR. Memory Write (MW) or Priviledged Write (PW) work like this; the memory location is read-out but the data is not saved. Instead, the contents of MDR are written into the addressed location.

Memory Write (MW) and Priveledged Write (PW) commands operate identically if the memory system does not include the Memory Protect option. If the Memory Protect option exists, the PW command unconditionally writes to memory. However, the MW command is subject to memory protect and will be converted to an MR command if the address is in a protected area of memory and if PSW 7 is set. If PSW 7 is reset, the Memory Protect feature is disabled.

INC - Increment LOC - The Location Counter is incremented by two. The result is also copied into MAR.

JAM - Copy FLR to CC - The Condition Code (Bits 12:15) of PSW is loaded from the Flag Register (FLR).

CLR - Clear - The Bank flip-flop and Utility flip-flop are cleared. The Flag Register is cleared. The counter is set to 16.

SWA - Set Wait Alarm - The Wait indicator is set.

CWA - Clear Wait Alarm - The Wait indicator is reset.

ALRM - Copy Alarm Bits - The bits set in the Alarm Register are copied to the Condition Code Register. If JAM is also specified, the Alarm Bits are "ORed" with the Flag Register Bits. The Alarm Register bits are then reset. See Section 8.

POW - Power Down - The system is initialized.

6.2 Command

C FUNCTION

0 3	4 15
0001	FUNCTION CODE

4	5	6	7	8	9	10	11	12	13	14	15	SYMBOLIC MEANING	
1	0											MR	MEMORY READ
0	1											MW	MEMORY WRITE
1	1											PW	PRIVILEDGED WRITE
		1	1							1	1	MPY	MULTIPLY
		1		1						0	1	DIV	DIVIDE
		1			1							RPT	REPEAT
		0	1									SRI	SHIFT RIGHT
		0		1								SLI	SHIFT LEFT
						1	0					SUT	SET UTILITY
						0	1					CUT	RESET UTILITY
						1	1					TUT	TOGGLE UTILITY
								1				SB	SET BANK
									1			CB	CLEAR BANK
										1		CI	CARRY IN
											1	CO	CARRY OUT

The Command micro-instruction results in the performance of the machine functions specified by the Function Code Bits. The functions are described below.

MR Memory Read

MW Memory Write

PW Priviledged Write

MPY - Multiply - The Processor multiplies the 16 bit multiplicand in AR by the 16 bit multiplier in SRL. The 32-bit product resides in SRH and SRL. To achieve this, the following setup conditions must exist. The counter contains 16; SRH contains zero; SRL and AR contain the operands; and the Carry Flag is reset. The C MPY instruction executes in sixteen Processor clock periods (16t). After the C MPY instruction, the 32-bit product in SRH and SRL must be shifted right one more position.

DIV - Divide - The Processor divides the 32 bit dividend in SRH and SRL by the 16 bit divisor in AR. To achieve this, the following setup conditions must exist. The counter contains 16; SRH/SRL contains a positive dividend that is less than 65,536 times the divisor. The AR contains the divisor in two's complement negative form; and the Carry Flag is reset. The C DIV instruction executes in 16t. After the C DIV instruction, the quotient resides in SRL and the remainder is in SRH.

RPT - Repeat - If the counter is not zero, the next sequential micro-instruction is repeated the number of times specified in the Counter Register. If the counter is zero, the next sequential micro-instruction is skipped. Any reasonable instruction may be repeated that does not result in a Branch (Branch, Load RAL or Decode Phase Change).

SR1 - Shift Right - The 32-bit shift register (SRH/SRL) is shifted right one position. Carry in and out are specifiable.

SL1 - Shift Left - The 32-bit shift register (SRH/SRL) is shifted left one position. Carry in and out are specifiable.

SUT - Set Utility

CUT - Reset Utility

TUT - Toggle (complement) Utility

The Utility flip-flop has no hardware function assigned to it. It is for program control and may be tested with the test micro-instruction.

SB - Set Bank

CB - Clear Bank

The Bank flip-flop controls the addressing of certain registers.

If no functions are specified, the Assembler leaves the function code field reset.

6.3 Test

T FUNCTION

0 3	4 15
0010	FUNCTION CODE

4	5	6	7	8	9	10	11	12	13	14	15		SYMBOLIC MEANING
1												FAST	FAST I/O INTERRUPT
	1											ATN	I/O ATTENTION
		1										ARST	AUTO RESTART
			1									CATN	CONSOLE ATTENTION
				1								SNGL	CONSOLE SINGLE MODE
					1							UT	UTILITY FLIP-FLOP
						1						MALF	MACHINE MALFUNCTION
							1					PPF	PRIMARY POWER FAIL
								1				DC	DATA CHANNEL REQUEST
									1			DRD	DATA CHANNEL READ
										1		MSK	MASK BITS
											1	OP	USER OPERATION LENGTH

If any of the machine functions specified by the function code bits are true, the Greater than (G) flag is set and the Less than (L) flag is reset. If none of the tested functions are true, the Less than (L) flag is set and the Greater than (G) flag is reset. The Testable functions are:

FAST - High Speed Interrupt - This flip-flop is AC set when the FAST0 interrupt line becomes active. This interrupt line is separate from the normal ATN line and may have any priority the micro-programmer decides. The FAST0 interrupt may be enabled by a PSW bit (PSW04) by back-panel strapping. After the micro-program tests the FAST flip-flop, the flip-flop automatically resets.

ATN - I/O Attention - An external device is requesting Processor service and PSW Bit 1 is set.

ARST - Automatic Restart - This is a hardwire strap option with significance only to the emulator.

CATN - Console Attention - The console EXEcute Switch has been depressed.

SNGL - Console Single Mode - The SNGL flip-flop in the display controller is set.

UT - Utility flip-flop set

MALF - Machine Malfunction - One of the bits in the Machine Malfunction Alarm Register is set. The alarm bits are Parity Error Phase Zero or One, Parity Error Phase Two, and Early Power Fail Detect.

PPF - Primary Power Fail - This signal occurs one millisecond after Early Power Fail Detect. The Micro-Code has one millisecond before system initialize occurs.

DC - Data Channel Request

DRD - Data Channel Read/Write

A special halfword device on the data channel is requesting memory access. The state of DRD determines Read or Write, 0 or 1, respectively.

MSK - Mask Bits - The AND of the Condition Code Bits (PSW 12:15) with IR4 (IR8:11) is tested.

OP - User Operation Length - This flip-flop is adjusted in Phase Zero. It is set to one if the user instruction is RX or RS. It is set to zero if the user instruction is RR.

6.4 Branch

B COND, ADDRESS

0 3	4 7	8 15
0011	CONDITION	ADDRESS

The Branch micro-instruction results in a transfer in the micro-program sequence if any of the specified conditions are true. If none of the conditions are true, the next sequential instruction is performed.

The condition is specified by RD Bits 4 through 7.

RD BIT	CONDITION	
4 5 6 7	SYMBOLIC	MEANING
0 0 0 0	CTR	Branch if Counter not 1
1 X X X	C	Branch if Carry true
X 1 X X	V	Branch if Overflow true
X X 1 X	G	Branch if Greater than zero
X X X 1	L	Branch if Less than zero

The Branch on Counter instruction results in a transfer if the Counter Register does not equal 00001_2. The counter is decremented by one regardless of its previous state.

The conditions C, V, G, or L may be specified singularly or in combination. Each condition bit is matched against the corresponding bit in the Flag Register. If a match is made, the ADDRESS field is copied to RAL, and RAS is loaded from RAH. If no match is made, the next instruction in sequence is executed. The instruction requires two Processor clocks (2t) if a branch is made. If no branch, the instruction takes 1t.

6.5 Load

L D, S, E

0 3	4 7	8 11	12 15
0100	D	S	E

The contents of the register specified by the Source (S) Field are copied into the register specified by the Destination (D) Field.

If neither the Source nor Destination field specify I/O, the E field has the following meaning:

E 12 13 14 15	SYMBOLIC	MEANING
0 0 X X		No shifts.
0 1 X X	SR	Shift Right. The source data is shifted right one bit position and copied into the Destination Register.
1 0 X X	SL	Shift Left. The source data is shifted left one bit position and copied into the Destination Register.
1 1 X X	CS	Cross Shift. The source data is rotated eight bit positions and copied into the Destination Register. If MDR is the Source or Destination, the cross-shift will occur only if MAR is even. If MDR is the Destination and MAR is even, only the high byte (MDH) of MDR is loaded. If MAR is odd, only the low byte (MDL) of MDR is loaded. The byte not loaded remains unchanged.
X X 0 0	NC	No Carry. The Carry Flag is unaltered.
X X 0 1	CO	Carry Out but not in. If a one is shifted out, the Carry Flag is set. If a zero is shifted out, the Carry Flag is reset.
X X 1 0	CI	Carry In but not out. The state of the Carry Flag is shifted into the most significant bit if Shift Right or the least significant bit if Shift Left.
X X 1 1	C	Carry In and Out.

If no options are specified, the Assembler sets the E Field to 0011.

If the Destination (D) Field specifies I/O, an output operation is to be performed. The E Field has the following meaning:

E 12 13 14 15	SYMBOLIC	MEANING
0 0 X X		No shift.
0 1 X X	SR	Shift Right.
1 0 X X	SL	Shift Left.
1 1 X X	CS	Cross Shift.
X X 0 0	DCAK	Data Channel Acknowledge. The source data is present on the Data Lines and the Address and Data Channel Acknowledge Control lines are active. The highest priority interrupting Data Channel device becomes the On Line device.
X X 0 1	ADRS	Address. The source data is present on the Data Lines and the Address Control Line is active. The device that detects its address becomes On Line and responds with a Sync.
X X 1 0	DA	Data Available. The source data is present on the Data Lines and the Data Available Control Line is active. The On Line device accepts the data and responds with a Sync.
X X 1 1	OC	Output Command. The source data is present on the Data Lines and the Output Command Control Line is active. The On Line device accepts the command data and responds with a Sync.

If no options are specified, the Assembler sets the E Field to 0000.

If the Source (S) Field specifies I/O, an input operation is to be performed. The E Field has the following meaning:

E 12	 13	 14	 15	SYMBOLIC	MEANING
0	0	X	X		No Shifts.
0	1	X	X	SR	Shift Right.
1	0	X	X	SL	Shift Left.
1	1	X	X	CS	Cross Shift.
X	X	0	0		Illegal Function.
X	X	0	1	ACK	Acknowledge Interrupt. The Acknowledge Interrupt Control Line is active. The highest in priority interrupting device responds by placing its address on the Data Lines. The input data is copied to the Destination Register.
X	X	1	0	DR	Data Request. The Data Request Control Line is active. The On Line device responds by placing data on the Data Lines. The input data is copied to the Destination Register.
X	X	1	1	STAT	Status Request. The Status Request Control Line is active. The On Line device responds by placing status data on the Data Lines. The input data is copied to the Destination Register.

The Carry Flag is not affected by I/O operations that specify shifts unless the FLR is the explicit destination. If, on any I/O operation, the device fails to respond within 45 microseconds, a false Sync is generated which sets the Overflow Flag and, if the E Field specifies Status Request, forces the Input status to '0004'. The instruction is aborted.

The halfword I/O capability requires special micro-code attention. When a halfword device controller is addressed, the HW control line is active. This control line affects the Processor only when a DR or DA operation is executed and allows the Processor to input/output one 16-bit halfword of data. All other I/O operations are strictly byte oriented. If a DR or DA operation is executed to a halfword device, cross shift is suppressed if it was specified in the LOAD micro-instruction and RAL Bit 14 is forced to a one. Forcing RAL Bit 14 makes the Processor skip two micro-instructions if the least significant hex digit of the address of the DR or DA instruction is 0, 3, 4, 7, 8, B, C, or F. If the least significant address digit is 1, 2, 5, 6, 9, A, D or E, no skip occurs.

6.6 Load Immediate

L D, DATA

0 3	4 7	8 15
0101	D	DATA

The eight bits from the DATA Field are copied into the register specified by the Destination (D) Field. The most significant eight bits are zeroed.

6.7 Or

O D, S, E

0 3	4 7	8 11	12 15
0110	D	S	E

The contents of the A Register (AR) are logically added to the contents of the register specified by the Source (S) Field. The logical sum is loaded into the register specified by the Destination (D) Field.

The E Field has the following meaning:

E				SYMBOLIC	MEANING
12	13	14	15		
1	X	X	X	NA	No AR. The AR is not unloaded. Zeros are ORed with the source data.
X	0	X	X	NF	No Flags. The G and L flags are not modified.
X	1	X	X		G and L are modified to reflect the algebraic magnitude of the result.
X	X	0	0	NC	No Carry. Carry is not modified.
X	X	1	0	CI	Carry In but not out. No affect.
X	X	0	1	CO	Carry Out but not in. The Carry flag is reset.
X	X	1	1	C	Carry In and Out. The Carry Flag is reset.

If no options are specified, the E Field is set to 0111.

6.8 OR Immediate

0 D, DATA

0 3	4 7	8 15
0111	D	DATA

The contents of the A Register (AR) are logically added to the eight bits of the Data Field. The most significant eight bits of the B Bus are zero. The logical sum is loaded into the register specified by the Destination (D) Field.

6.9 And

N D, S, E

0 3	4 7	8 11	12 15
1000	D	S	E

The contents of the A Register (AR) are logically multiplied by the contents of the register specified by the Source (S) Field. The logical product is loaded into the register specified by the Destination (D) Field.

The E Field has the following meaning.

E				SYMBOLIC	MEANING
12	13	14	15		
1	X	X	X	NA	No AR. The AR is not unloaded. The result will be zero.
X	0	X	X	NF	No Flags. The G and L flags are not modified.
X	1	X	X		G and L are modified to reflect the algebraic magnitude of the result.
X	X	0	0	NC	No Carry. Carry is not modified.
X	X	1	0	CI	Carry In but not Out. No affect.
X	X	0	1	CO	Carry Out but not In. The Carry flag is reset.
X	X	1	1	C	Carry In and Out. The Carry flag is reset.

If no options are specified, the E Field is set to 0111.

6.10 And Immediate

N D, DATA

0 3	4 7	8 15
1001	D	DATA

The contents of the A Register (AR) are logically multiplied by the eight bits of the Data Field. The most significant eight bits of the B Bus are zero. The logical product is loaded into the register specified by the Destination (D) Field.

6.11 Exclusive Or

X D, S, E

0 3	4 7	8 11	12 15
1010	D	S	E

The contents of the A Register (AR) are logically subtracted from the contents of the register specified by the Source (S) Field. The logical difference is loaded into the register specified by the Destination (D) Field. The E Field has the following meaning:

E 12 13 14 15	SYMBOLIC	MEANING
1 X X X	NA	No AR. The AR is not unloaded. The source data is unaltered.
X 0 X X X 1 X X	NF	No Flags. The G and L flags are not modified. The G and L flags are modified to reflect the algebraic magnitude of the result.
X X 0 0	NC	No Carry. Carry is not modified.
X X 1 0	CI	Carry In but not Out. No affect.
X X 0 1	CO	Carry Out but not In. The Carry flag is reset.
X X 1 1	C	Carry In and Out. The Carry flag is reset.

If no options are specified, the E Field is set to 0111.

6.12 Exclusive Or Immediate

X D, DATA

0 3	4 7	8 15
1011	D	DATA

The contents of the A Register (AR) are logically subtracted from the eight bits of the Data Field. The most significant eight bits of the B Bus are zero. The logical difference is loaded into the register specified by the Destination (D) Field.

6.13 Add

A D, S, E

0 3	4 7	8 11	12 15
1100	D	S	E

The contents of the A Register (AR) are algebraically added to the contents of the register specified by the Source (S) Field. The sum is loaded into the register specified by the Destination (D) Field.

The E Field has the following meaning:

E 12 13 14 15	SYMBOLIC	MEANING
1 X X X	NA	No AR. The AR is not gated to the ALU. Zeros are added to the Source data.
X 0 X X X 1 X X	NF	No Flags. The G, L, and V flags are not modified. The G and L flags are adjusted to reflect the algebraic magnitude of the result. The V flag is adjusted to reflect the overflow condition.
X X 0 0	NC	No Carry. The Carry flag does not participate, nor is it modified.
X X 1 0	CI	Carry In but not Out. The Carry flag is added with the least significant bit of the sum.
X X 0 1	CO	Carry Out but not In. The state of the arithmetic carry is saved in the Carry flag.
X X 1 1	C	Carry In and Carry Out.

If no options are specified, the E Field is set to 0111.

6.14 Add Immediate

A D, DATA

0 3	4 7	8 15
1101	D	DATA

The contents of the A Register (AR) are algebraically added to the eight bits of the Data Field. The most significant eight bits of the B Bus are zero. The sum is loaded into the register specified by the Destination (D) Field.

6.15 Subtract

S . D, S, E

0 3	4 7	8 11	12 15
1110	D	S	E

The contents of the A Register (AR) are algebraically subtracted from the contents of the register specified by the Source (S) Field. The difference is loaded into the register specified by the Destination (D) Field.

The E Field has the following meaning:

E 12 13 14 15	SYMBOLIC	MEANING
1 X X X	NA	No AR. The AR is not gated to the ALU. Zeros are subtracted from the Source data.
X 0 X X	NF	No Flags. The G, L, and V flags are not modified.
X 0 X X		The G and L flags are adjusted to reflect the algebraic magnitude of the result. The V flag is adjusted to reflect the overflow condition.
X X 0 0	NC	No Carry. The Carry flag does not participate nor is it modified.
X X 1 0	CI	Carry In but not Out. The Carry flag represents the borrow situation from Bit 15 of the Source data. This borrow participates in the subtraction.
X X 0 1	CO	Carry Out but not In. The state of the arithmetic carry (borrow) is saved in the Carry flag.
X X 1 1	C	Carry In and Carry Out.

If no options are specified, the E Field is set to 0111.

6.16 Subtract Immediate

S D, DATA

0 3	4 7	8 15
1111	D	DATA

The contents of the A Register (AR) are algebraically subtracted from the eight bits of the Data Field. The most significant eight bits of the B Bus are zero. The difference is loaded into the register specified by the destination (D) Field.

7. MULTIPLEXOR CHANNEL

The Multiplexor Channel is a byte or halfword oriented I/O system which communicates with up to 255 peripheral devices. The Multiplexor Bus consists of 30 lines; 16 bi-directional Data Lines, 9 Control Lines, 5 Test Lines, and an Initialize Line.

A list of the lines in the Multiplexor Bus follows.

Data Lines	D00:15	(Processor ⟷ Device)	16 Lines
Control Lines	SR	(→)	1 Line
	DR	(→)	1 Line
	CMD	(→)	1 Line
	DA	(→)	1 Line
	ADRS	(→)	1 Line
	ACK	(→)	1 Line
	DACK	(→)	1 Line
	CL07	(→)	1 Line
Test Lines	ATN	(←)	1 Line
	SYN	(←)	1 Line
	HW	(←)	1 Line
	DC	(←)	1 Line
	DRD	(←)	1 Line
Initialize	SCLRO	(→)	1 Line

Figure 2 is a block diagram of the Multiplexor Channel. The following general definitions apply to the lines in the **Multiplexor Bus.**

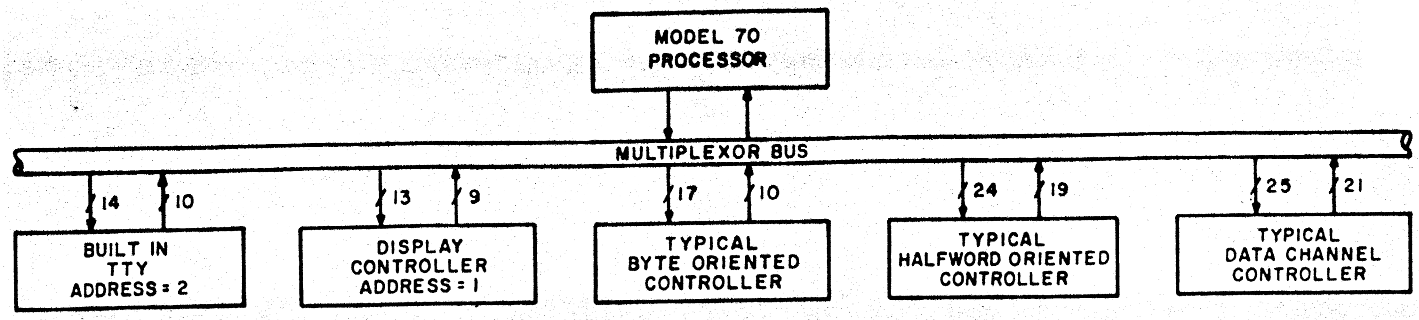

Figure 2. Multiplexor Channel Block Diagram

7.1 Data Lines D00:15

The Data Lines are used to transfer one 8-bit byte or one 16-bit halfword of data between the Processor and the Device. One byte of data is transferred from the Processor to the Device over D08:15 when accompanied by either an Address (ADRS) or a Command (CMD) control line. One byte of data or one halfword of data is transferred from the Processor to the Device when accompanied by the Data Available (DA) control line. The Device, in response to an Acknowledge (ACK) control line or a Sense Status (SSR) control line, sends one byte of address or status information to the Processor over D08:15. In response to a Data Request (DR) control line, the Device sends either an 8-bit byte or a 16-bit halfword of data to the Processor. The device always sends a Synchronize (SYN) signal to the Processor to indicate that it has either received the data from the Processor or that it has sent the data to the Processor. The SYN signal is removed immediately after the Processor removes the control line.

7.2 Control Lines

CONTROL LINE	FUNCTION	DESCRIPTION
0	ADRS (Address)	The Processor presents an address byte on D08:15. The Device Controller that decodes its Device Number becomes addressed or on line and responds with a SYN.
1	DA (Data Available)	The Processor presents data on D00:15 for the device. The Device Controller accepts the low order byte or the entire halfword and responds with a SYN.
2	DR (Data Request)	The Device Controller presents data to D08:15 or D00:15, followed by a SYN. If the Device Controller is halfword oriented, the HW test line is also active.
3	STAT (Status Request)	The Device Controller presents the device status byte to D08:15, followed by a SYN.
4	CMD (Command)	The Processor presents a command byte on D08:15 for the device. The Device Controller accepts the command byte and responds with a SYN.
5	ACK (Acknowledge)	**The Device Controller nearest the Processor that originated an ATN presents its address on D08:15, followed by a SYN.**
6	DCAK (Data Channel Acknowledge)	**The micro-program should present an address of zero on D00:15. The ADRS and DCAK control lines are simultaneously active. The Data Channel Controller nearest the Processor that originated a DC becomes addressed and responds with a SYN. Addressing device zero (a null address) leave all other Device Controllers un-addressed.**
7	CL070	**This control line is activated by the Processor when a power fail condition is detected by the Processor if the power fail option is equipped. The line is held active until the SCLR0 signal occurs.**

7.3 Test Lines

ATN Attention. Any Device desiring to interrupt the Processor will activate the ATN line and hold this line until an ACK is received from the Processor.

HW Halfword. Any halfword-oriented Device, other than Data Channel devices, will activate the HW line when it becomes addressed and hold this line for as long as it is addressed. When the HW test line is active and a Micro-Program does a DA or DR I/O operation, if Cross Shift was specified on that I/O Load, Cross Shift is suppressed so the 16 bits of data are not modified and ROM Address Register Bit 14 (RAL14) is forced to a one. If RAL14 was already set, RAL remains unchanged.

DC Data Channel Request. Any Data Channel Device desiring to interrupt the Processor will activate the DC line and hold this line until a DCAK is received from the Processor.

DRD Data Channel Read. The selected Data Channel Device controls the state of the DCR line - low for Read, high for Write, from the Device.

SYN Synchronize. This signal is generated by the Device to inform the Processor that it has properly responded to a control line.

SCLR System Clear. This is a metallic contact to ground that occurs during Power Fail, Power Up, or Initialize.

NOTE

All Control Lines, except ACK and DCAK, are connected in parallel to an External Interrupt. The ACK line is connected in series with all Devices. The DCAK line is connected in series with all Data Channel devices. If no interrupt is pending in the first Controller ACK or DCAK arrives at, the signal is passed on daisy chain fashion to the next Controller and so on until it is captured by the interrupting Controller. See definitions of ACK and DCAK.

7.4 Micro-Programming Input/Output

Input/Output operations are initiated when a Load micro-instruction specifying IO as the source or destination is executed. A Device Controller must be addressed before any data can be transferred. To address a device, the device number must reside in some register.

```
L   SRL,  'nn'              DEVICE NUMBER

L   IO, SRL, ADRS           ADDRESS THE DEVICE

B   V, ERROR                FALSE SYNC
```

If the device is operational, it responds with a SYN. If SYN is not received within approximately 45 microseconds, a false sync (FSYN) is generated which sets the V flag in FLR and aborts the micro-instruction. If the Micro-programmer is going to be concerned about the lack of response from a device, he may follow up the ADRS micro-instruction with a Branch on Overflow.

If FSYN occurs on ADRS, one can be confident that it will also occur on subsequent I/O operations to the same device. Thus, the micro-programmer may wait until later to catch the FSYN condition.

Sample Input/Output sequences for some standard devices follow. Obviously, routines for every device cannot be shown and the routines shown may not fit every possible application. Device characteristics can be found in the documentation for each device. The micro-programmer must understand that device documentation is slanted towards the Model 70 user repertoire and timing problems may occur when a device is driven with micro-code.

The following general significance is attached to the status byte from a device.

DEVICE STATUS

0	1	2	3	4	5	6	7	
				1				Busy. The device Controller is not ready to transfer data.
					1			Examine status. Either an FSYN has occurred or an unusual status exists. The entire status byte must be examined.
						1		End Of Medium. This bit is set at the termination of the device medium, such as end of card. Not all devices use this bit.
							1	Device Unavailable. The device is powered down or otherwise off line.

When device status is requested, it is convenient to use the Flag Register (FLR) as the destination. Branch instructions can then test the low order four status bits.

TELETYPE PAPER TAPE INPUT

The following instructions start the tape reader:

```
        L  SRL, '02'               TTY DEVICE NUMBER
        L  IO, SRL, ADRS           ADDRESS THE DEVICE
        L  SRL, '98'               COMMAND BYTE
        L  IO, SRL, CMD            SELECTS WRITE MODE
SENSE0  L  FLR, IO, STAT
        B  VGL, ERROR
        B  C, SENSE0
        L  SRL, '91'               X-ON CHARACTER
        L  IO, SRL, DA
```

As soon as the TTY processes the X-ON Character, the tape reader will begin advancing the tape.

```
SENSE1      L  FLR, IO, STAT       WAIT FOR X-ON
            B  VGL, ERROR          CHARACTER TO BE
            B  C, SENSE1           PROCESSED
            L  SRL, '94'
            L  IO, SRL, CMD        SWITCH TO READ MODE

SENSE2      L  FLR, IO, STAT       STATUS REQUEST
            B  VGL, ERROR          BAD STATUS
            B  C, SENSE2           LOOP IF BUSY
            L  SRL, IO, DR         READ DATA
```

After processing the data byte in SRL, return to SENSE2 to receive next data byte.

```
            •
            •
            •
```

The following instructions stop the TTY reader

```
            L  SRH, '98'           COMMAND BYTE
            L  IO, SRH, CMD        SWITCH TO WRITE MODE
SENSE3      L  FLR, IO, STAT       STATUS REQUEST
            B  VGL, ERROR
            B  C, SENSE3
            L  SRH, '93'           X-OFF CHARACTER
            L  IO, SRH, DA         OUTPUT IT
SENSE4      L  FLR, IO, STAT       WAIT FOR CHARACTER
            B  C, SENSE4           TO BE PROCESSED
            •
            •
            •
```

The Busy loops at SENSE1 and SENSE4 are not really necessary. It is generally recommended, however, that the Processor waits on Busy before switching a device from the Write Mode to the Read Mode. It is also good practice to leave a device in an un-Busy condition when the routine has finished.

TELETYPE PAPER TAPE OUTPUT

Start the TTY punch

```
              L  SRH, '02'             TTY DEVICE NUMBER
              L  IO, SRH, ADRS         ADDRESS THE DEVICE
              L  SRH, '98'             COMMAND BYTE
              L  IO, SRH, CMD          SELECT WRITE MODE
SENSE1        L  FLR, IO, STAT         STATUS REQUEST
              B  VGL, ERROR            BAD STATUS
              B  C, SENSE1             LOOP IF BUSY
              L  SRH, '92'             TAPE ON CHARACTER
              L  IO, SRH, DA           OUTPUT IT
SENSE2        L  FLR, IO, STAT
              B  VGL, ERROR            BAD STATUS
              B  C, SENSE2             LOOP IF BUSY
              •
```

Load data to output into SRH, then

```
              •
              L  IO, SRH, DA
              •
```

Loop back to SENSE2 to output subsequent data bytes

```
              •
              •
```

Stop the TTY punch

```
SENSE3        L  FLR, IO, STAT         WAIT FOR LAST
              B  C, SENSE 3            CHARACTER
              L  SRH, '94'             TAPE OFF CHARACTER
              L  IO, SRH, DA
SENSE4        L  FLR, IO, STAT         WAIT FOR CHARACTER
              B  C, SENSE4             TO BE PROCESSED
              •
              •
```

HIGH SPEED PAPER TAPE READER

The following instructions start the HSPTR

```
              L  SRH, '03'             HSPTR DEVICE NUMBER
              L  IO, SRH, ADR          ADDRESS THE DEVICE
              L  SRH, '99'             COMMAND BYTE
              L  IO, SRH, CMD          READ MODE
SENSE1        L  FLR, IO, STAT         STATUS REQUEST
              B  VGL, ERROR            BAD STATUS
              B  C, SENSE1
              L  SRH, IO, DR
              •
              •
```

After processing the data, return to SENSE1 to read next byte

```
              •
              •
```

Stop the HSPTR

```
              L  SRH, 'A9'             COMMAND BYTE
              L  IO, SRH, CMD          DSBL, STOP, INCR, READ
```

HIGH SPEED PAPER TAPE PUNCH

```
           L  SRH, '03'
           L  IO, SRH, ADR
           L  SRH, '92'
           L  IO, SRH, CMD
SENSE1     L  FLR, IO, STAT
           B  VGL, ERROR
           B  C, SENSE1
```

Load data to output into SRH

```
           .
           .
           .
           L  IO, SRH, DA
```

Loop back to SENSE to output subsequent data bytes:

```
           .
           .
           .
SENSE2     L  FLR, IO, STAT          WAIT FOR LAST
           B  VGL, ERROR             CHARACTER TO BE
           B  C, SENSE 2             PROCESSED
           L  SRH, 'A9'
           L  IO, SRH, CMD           DSBL, STOP, INCR, READ
```

8. INTERRUPT SYSTEM

The interrupt structure provides rapid response to external and internal events that require special software attention. The descriptions that follow are oriented towards the emulator.

8.1 Internal Interrupts

Seven different internal interrupts may be generated. Of these, the Fixed-Point Divide Fault, Floating-Point Arithmetic Fault, Queue Service and Supervisor Call Interrupts are created by the emulator and the Illegal Instruction, Protect Mode Violation and Machine Malfunction Interrupts are generated in the hardware.

8.1.1 Illegal Instruction Interrupt. The Illegal Instruction Interrupt occurs when an instruction not in the user repertoire is attempted. When Phase Two is entered, the Decoder Read-Only Memory (DROM) is interrogated. DROM is addressed by the Operation Code Field (Bits 0:7) of IR. Each legal user instruction has an associated 12-bit word in DROM. This word is the starting address of the micro-routine that will execute the user's instruction. Illegal op-codes have an associated DROM word of all zeros. The DROM read-out is jammed into RAH, RAS, and RAL as Phase Two is entered.

ROM location X'000' contains all zeros. When ROM location X'000' is read-out, the zero condition of RD forces the Processor to Phase Three and sets the ROM Address Registers to X'100'. The micro-routine at X'100' decrements the Location Counter (LOC) by 2 if the attempted instruction was RR or by 4 if RX or RS. Then the current program status word (PSW and LOC) is stored in the old PSW save area for the Illegal Instruction Interrupt, and the New PSW for the Illegal Instruction Interrupt becomes the current PSW.

8.1.2 Protect Mode Violation Interrupt. When PSW Bit 7 is reset, the Processor is in Supervisor Mode and all user instructions are executable.

The Protect Mode Violation Interrupt occurs when execution of a Priviledged instruction is attemped while the Processor is in the Protect Mode (PSW Bit 7=1). An instruction is priviledged if it performs Input/Output or changes the Status Field (Bits 0:11) of PSW. The high order bit of DROM read-out, Bit 4, must be a one for the priviledged instructions. If PSW Bit 7 is set, RAH Bit 4 is allowed to set on DROM read-out Bit 4. The ROM Address Registers then specify non-existent ROM outside of the 2K allowed. Reading non-existent ROM yields a zero RD, resulting in an Illegal Instruction Interrupt.

8.1.3 Machine Malfunction Interrupt. The Machine Malfunction Interrupt occurs on Memory Parity errors or on Primary Power Fail if PSW Bit 2 is set. The emulator will also generate a Machine Malfunction Interrupt on Power Up if PSW Bit 2 is set.

Testable Primary Power Fail occurs if the Power Fail Detector circuit determines that primary line voltage is low or the Initialize Switch is depressed or the Power Switch is turned off. When any of the above events occur, a one millisecond timer is started and, if PSW Bit 2 is set, the Power Fail bit of the Machine Malfunction Alarm Register is set. See Figure 3. After the one millisecond time-out, the signal PPF is generated. The user may use the one millisecond interval to do any necessary system shut-down procedures.

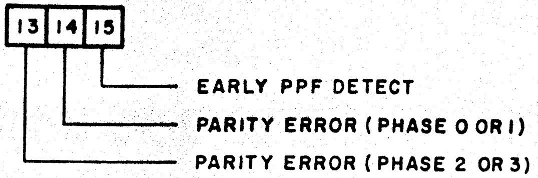

Figure 3. Machine Malfunction Alarm Register

During this interval, PSW Bit 2 may be set again as the Machine Malfunction Interrupt is only taken once. After the time-out, the signal PPF occurs causing the micro-program to save the PSW and LOC and the General Registers and then initialize the system.

If the Memory Parity option is present, the Parity Bit of each halfword in memory is set or reset to maintain odd parity. The Parity Bit is generated on every Memory Write or Priviledged Write and checked on every Memory Read.

If a parity error occurs and PSW Bit 2 is set, the parity error flags in the Alarm Register are set to indicate the Phase the Processor was in when the error occurred.

When Phase 2 is exited, the state of the Machine Malfunction Alarm Register, along with other interrupts, is tested. If any alarm bit is set, the Processor is forced to Phase 3 and the ROM Address Registers are set to X'010'.

The Interrupt Service Subroutine sorts interrupts by priority. Machine Malfunction is second to Primary Power Fail. If PPF is not active, the Machine Malfunction Interrupt is tested. If any bit in the Alarm Register is set, a PSW swap is performed and the bits that are set in the Alarm Register are ORed into the New PSW condition code. The Alarm Register is then cleared. This is done by the ALRM function of the DO Micro-instruction.

8.2 External Interrupts

If individually enabled by the user, a peripheral Device Controller is allowed to request Processor service when the device is ready to transfer data. The Processor has two separate classes of interrupts directly related to peripheral device handling. These are Data Channel Requests and Normal I/O Interrupts over the Multiplexed I/O Bus. If PSW Bit 1 is reset, normal I/O device interrupt signals are ignored. The signal remains pending, however, until PSW Bit 1 is set and the interrupt acknowledged.

8.2.1 Data Channel. The Data Channel feature of the Processor provides a means for a customer designed Device Controller to gain rapid access to core memory. When a memory access is required, the Controller activates the Data Channel Request Line (DC). This line is separate from the normal I/O ATN line and is not masked by PSW Bit 1. The micro-program tests the DC line before testing ATN, thereby placing it higher in priority.

The Processor acknowledges the Data Channel Interrupt by addressing Device Number 0 and also by activating the DCAK line.

The Data Channel Acknowledge (DCAK) line is daisy-chained through all devices connected to the Data Channel. The Device Controller closest to the Processor that originated a Data Channel interrupt captures the DCAK signal. That device becomes the On-Line device and controls the DRD Line. The DRD Line tells the micro-program whether a Memory Read (DRD = 1) or Memory Write (DRD = 0) is to be performed. If a Memory Read is requested, the micro-program does a data request. The Device Controller should then present a 16-bit memory address to the Data Lines. This address is copied into the Memory Address Register, memory is read and the memory data is gated out to the 16 Data Lines. If a memory write is requested, the micro-program does two data requests. The first collects the 16-bit address and the second brings in the 16-bits of data. The data is then written into core. If another memory cycle is desired, the device re-activates the DC line. Data Channel device controllers should not activate the HW line.

8.2.2 Normal I/O Interrupt. The Processor may service a normal I/O Interrupt in a variety of ways depending on the state of PSW Bit 4 and in-core tables. The interrupt handling is a function of the emulator and is described in the Reference Manual, Publication Number 29-219.

9. ADDITIONAL SPECIFICATIONS

The following is a list of Micro-programming constraints and general information:

1. On power up or following a C POW Micro-instruction, the Initialize relay is closed until voltages are at nominal levels. **While the Initialize relay is closed, the signal** SCLR (System Clear) **sets up the Initialized state:**

BANK	=	0
UT	=	0
RUN	=	0
RAH	=	0
RAS	=	0
RAL	=	0
RD	=	0
CTR	=	16
FLR	=	0
Phase	=	3

2. When Micro-programming arithmetic operations, remember that the A Register is the second operand and the Source Register is the first operand. This is important in subtract; the A Register is the subtrahend.

3. When micro-programming multiple-precision operations, the low order halfword(s) of the operands are considered to be all magnitude. The sign bit of a multiple-precision operand is Bit 0 of the most significant halfword.

   The hardware provides a cumulative flag affect to facilitate multiple-precision operations. Once the G or L flag becomes set, the G and L flags will never again be both zero unless the flag register is explicitly cleared. For this reason, prior to examining the least significant 16 bits of an operand, a D CLR must have occurred or the Flag Register loaded with zero. If G and L remain reset, the halfword was zero. If either G or L set, the halfword was not zero. As successively more significant halfwords are examined, if G and L stay reset, the partial operand is zero. Otherwise, G or L set indicates non-zero. When the most significant 16 bits are examined, G and L are still reset if the entire operand is zero or, if not zero, the G and L flags reflect the sign of the entire operand.

BEFORE		RESULT OF ARITHMETIC	AFTER	
G	L	OR LOGICAL OPERATION	G	L
0	0	0000 0000 0000 0000	0	0
0	0	0000 0011 0101 0000	1	0
0	0	1000 1110 0100 1100	0	1
0	1	0000 0000 0000 0000	1	0
0	1	0000 0011 0101 0000	1	0
0	1	1000 1110 0100 1100	0	1
1	0	0000 0000 0000 0000	1	0
1	0	0000 0011 0101 0000	1	0
1	0	1000 1110 0100 1100	0	1

4. Flags are not changed in any immediate operation unless the FLR is the explicit destination.

5. **If the FLR is the destination in an instruction that normally sets flags, the FLR will contain the OR of the low order 4-bits of the result and the result flags.**

6. **Specifying Carry Out (CO) is reasonable only on shifts or Add or Subtract. If Carry Out is specified on the logical operations (N, O, X), Carry will reset. Specifying Carry In (CI) has no affect on logical operations.**

7. **To implement a transfer within a micro-program, special care should be taken in the loading of the ROM address registers (RAH, RAL). The ROM decodes the address from the RAL and the page from the inner rank RAS. RAS is loaded from RAH whenever RAL is loaded or a Branch is taken. Loading RAH has no immediate affect on the address for the ROM. Before loading RAL, RAH must contain the correct page number. Addresses generated by the D0 micro-instruction set both RAH and RAS.**

8. **RAL should not be the destination in an I/O operation.**

9. **The instruction L MAR, MDR, CS can not be reliably executed.**