Publication Number 29-399R02

MODEL 7/32
REFERENCE MANUAL

CONSISTS OF:

Functional Characteristics of Model 7/32	29-399R01A12
Hexadecimal Display Programming Specification	09-065R01A22
Memory Access Controller Programming Specification	02-348R01A22
Extended Selector Channel Programming Specification	02-328R01A22

INTERDATA®
2 Crescent Place, Oceanport, New Jersey 07757

September 1974

29-399R01A12
September 1974

FUNCTIONAL CHARACTERISTICS OF MODEL 7/32

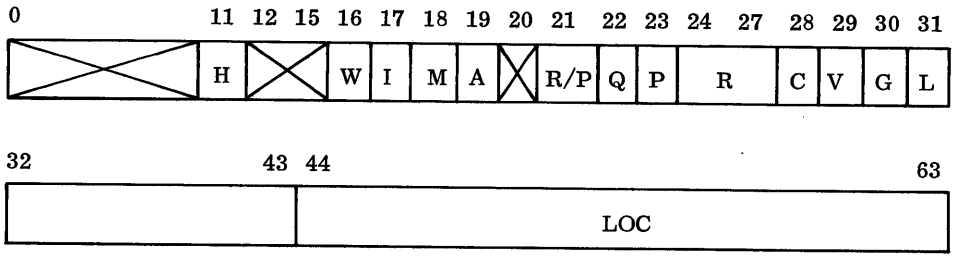
1. INTRODUCTION

Model 7/32 is one implementation of the 32 Bit Series Architecture described in 32 Bit Series Reference Manual (29-365). The 7/32 implementation does not include all the features described in the 32 Bit Series Reference Manual, and is therefore a logical subset of the complete system. The purpose of this manual is to describe all programming features specific to the Model 7/32. Therefore, this manual does not repeat all the descriptions in 29-365, but supplements it.

2. MODEL 7/32 PROCESSOR AND MEMORY

2.1 Program Status Word

The 64 bit Program Status Word (PSW) for Model 7/32 is shown below:



Bits 44:63 contain the Location Counter. Only twenty bits are used for Location Counter. The maximum memory available in Model 7/32 is 2^{20} bytes (i.e., 1,048,576 bytes or approximately 1 Megabyte). Program references to memories beyond maximum possible (i.e., beyond 1 Mbyte) result in wraparound. For example: if an attempt is made to address memory location X'100000', the instruction addresses location X'00000'.

If an attempt is made to read from a non-existing memory location in the local memory (Bank 0, first 256KB of memory), the data read is all zeros. If an attempt is made to read from a non-existing memory using the Memory Access Controller interface, the data read is undefined. (Also, see Section 2.4.4).

Bits 0:31 are reserved for status information, interrupt masks and machine mode (halfword or fullword). These bits are defined as follows:

Bit 11 - Halfword Mode
Bit 16 - Wait State
Bit 17 - Immediate Interrupt Mask
Bit 18 - Machine Malfunction Interrupt Mask
Bit 19 - Arithmetic Fault Interrupt Mask
Bit 21 - Relocation/Protection Interrupt Mask
Bit 22 - System Queue Service Interrupt Mask
Bit 23 - Protect Mode
Bits 24:27 - Register Set Select Bits
Bits 28:31 - Condition Code

Unassigned PSW bits must not be used and must always be zero. PSW bit 20 is not to be used by any program that runs on Model 7/32. Bits 17 and 20 together define the Priority Interrupt Enable Mechanism for 32 Bit Series machines as described below:

Bit 17	Bit 20	
0	0	Disable I/O Interrupts at all levels
0	1	Enable at higher level; disable at current and lower levels
1	0	Enable I/O Interrupts at all levels
1	1	Illegal combination

Note that when bit 20 is a zero, as it must be on Model 7/32, priority interrupt system is not enabled. In that case, interrupts are unconditionally enabled or disabled by bit 17 of the PSW.

PSW bits 24:27 are used to designate the current register set. In Model 7/32, there are only two register sets - set 0 and set 15. Register set 0 is the I/O and Interrupt register set and register set 15 is the user register set. In order to maintain upward compatibility with future INTERDATA 32-Bit machines, it is necessary to select the I/O register set by loading 0 into bits 24:27 and select the user register set by loading X'F' into bits 24:27.

Bit 11 of the PSW selects a special mode called Halfword Mode. This bit is used only for certain special applications and is normally not used. The Halfword Mode and its implications are described in detail in Section 2.3.

2.2 Reserved Memory Locations

The following memory locations are reserved for interrupt pointers, Program Status Words and system constants.

Location	Use
X'000000'-X'00001F'	Floating point registers
X'000020'-X'000027'	Machine malfunction interrupt old PSW
X'000028'-X'000029'	Used by microprogram to save console status
X'00002A'-X'00002F'	Not used, must be zero
X'000030'-X'000037'	Illegal instruction interrupt new PSW
X'000038'-X'00003F'	Machine malfunction interrupt new PSW
*X'000040'-X'000047'	Old, New PSW Ext. Interrupt Halfword Mode
X'000048'-X'00004F'	Arithmetic fault interrupt new PSW
X'000050'-X'00007F'	Bootstrap loader and device definition table
X'000080'-X'000083'	System queue pointer
X'000084'-X'000085'	Current PSW save pointer
X'000086'-X'000087'	Register save pointer
X'000088'-X'00008F'	System queue service interrupt new PSW
X'000090'-X'000097'	Relocation/protection interrupt new PSW
X'000098'-X'00009B'	Supervisor call new status
X'00009C'-X'0000BB'	Supervisor call interrupt new location counters (32 Bytes)
X'0000BC'-X0000CF'	Not used, must be zero
X'0000D0'-X'0002CF'	Interrupt service pointer table (256 Halfwords)
X'0002D0'-X'0004CF'	Expanded interrupt service pointer table (256 Halfwords)
X'0004D0'-X'0008CF'	Expanded interrupt service pointer table (256 Halfwords)

* Used only in Halfword Mode. See Section 2.3.4 for explanation.

These reserved locations play an important role in both interrupt and I/O processing. They are explained in detail in Section 2.4. It should be noted here that the new PSW Location for each external interrupt and Supervisor Call is a halfword (16 bits). Therefore, all the interrupt processing routines (or at least the first instruction of such routines) must reside in first 64K bytes of memory.

The basic machine is set up for handling 255 device addresses. Device numbers may range from X'001' through X'3FF' (Device number X'000' is not used). Dedicated numbers are X'001' for the console and X'007' for the LSU.

The Interrupt Service Pointer table at memory locations X'D0'-X'2CF' is used for vectoring the device interrupts. This table can be expanded optionally to handle either 511 or 1023 device interrupts. This is accomplished by expanding the bus buffer through the use of Sub Channel Controllers to provide two additional bits of address decoding, in which case, the service pointer table is expanded either to X'4CF' or to X'8CF'.

All the memory addresses stored in these memory locations for Old or New PSW locations are absolute values and are not translated by the Memory Access Controller (MAC), if present. Refer to Section 4 for the detailed description of the MAC.

If the configuration has provision for no more than 255 external devices, the Memory Access Controller (MAC) segmentation registers are assigned locations X'000300' thru X'00033F'. The MAC status register is assigned the fullword location at X'000340'. If the configuration has provision for more than 255 external devices, the block of memory locations assigned to the controller starts at the nearest multiple of X'000100' above the expanded interrupt service pointer table. For example, if the Interrupt Service Pointer Table is expanded to X'0004CF', the MAC registers would be assigned locations X'000500 through X'000540'.

Bit 21 of the current program status word enables the segmentation, relocation and protection features of MAC. If this bit is reset, all memory references are absolute and all protection is disabled.

When bit 21 is reset, the controller is still active in that it traps memory references to the location assigned to its registers. If the processor references any location trapped, but not used, by the Controller, the results are undefined. The Controller traps at least a 256 byte block of memory beginning at X'000300'. It is therefore recommended that all programs reside in memory above memory location X'000A00'.

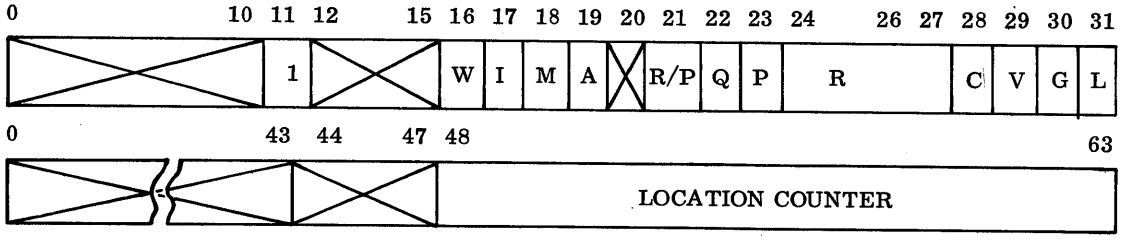
2.3 Machine Modes

The INTERDATA 7/32 Processor is a microprogrammed computer which is equipped with additional Read Only Memory (ROM) which enables it to emulate INTERDATA Models 7/16, 70, 74, and 80. Program compatibility at the machine language level is achieved by providing a special compatibility mode called Halfword Mode.

NOTE

The Halfword Mode, as described below, is included in the 7/32 to allow the execution of certain 16-bit series programs, such as some test programs. The Halfword mode feature will not necessarily be included in future members of the 32-bit series of machines.

The 7/32 Processor can be made to operate in halfword mode by setting a bit (bit 11) in the Program Status Word. The 64 bit program status word for Model 7/32 (in halfword mode) is shown below:



The PSW bits from 16 to 31 have the same meaning as explained in 2.1. Bits 48:63 contain the Location Counter.

Figure 1 shows the instruction formats in halfword mode. These formats are identical to the formats of a 16 bit processor.

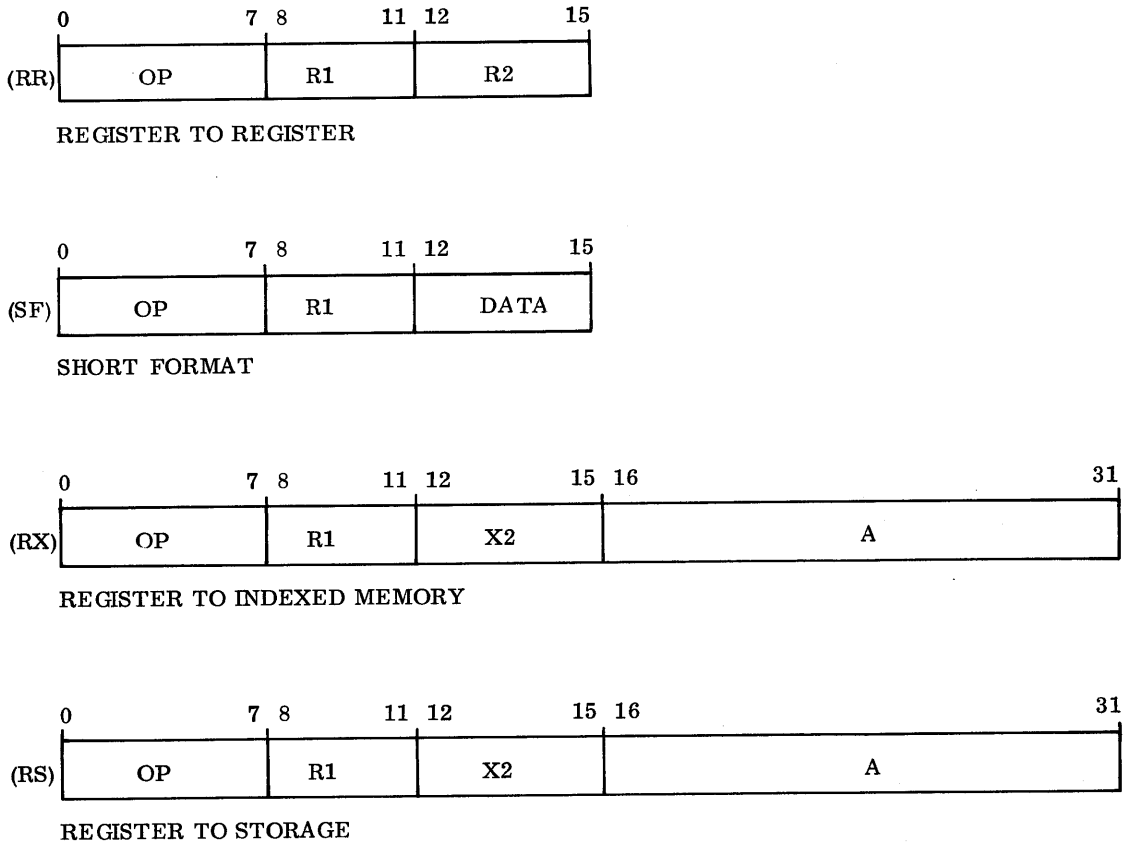


Figure 1. Instruction Formats (Halfword Mode)

The eight-bit OP field in all formats specifies the machine operation to be performed. Operation codes are represented as two hexadecimal characters.

The four-bit R1 field in the instruction formats specifies the address of the first operand. The R1 field is normally the address of a General Register.

The four-bit R2 field in the RR instruction format specifies the address of the second operand, which is normally a register address.

The four-bit DATA field of the SF instructions supplies data in the case of Fixed-Point Arithmetic instructions, or a displacement from the current Location Counter in the case of Branch instructions.

A non-zero X2 field in the RX and RS formats specifies a General Register whose contents is used as an index value. The index value (X2) may be positive or negative. If X2 is zero, no address modification takes place. General Registers 1 through 15 can optionally be used for indexing, but General Register 0 can never be used for indexing.

The 16-bit Address field specifies a memory address in the RX format, or contains a value (data) to be used as an immediate operand in the RS format.

The instructions shown in Appendix 2 can be executed in halfword mode. This set of instructions corresponds to a Model 70, 80, or 7/16 with HSALU option. The machine, when operating in halfword mode, executes all non-privileged instructions (except SVC) exactly the same way they are executed on 16-bit INTERDATA processors. The SVC instruction, in halfword mode, is executed as follows:

The SVC instruction has RX format. The second operand address, A+(X2) replaces bits 16:31 of Register 13 of Register set 0. Bits 0:15 of this register are undefined and should be ignored. The current Program Status Word replaces the contents of Registers 14 and 15 of Register set 0. The fullword quantity located at '000098' in memory replaces bits 0:31 of the current Program Status Word. The R1 field is doubled and added to X'00009C'. The halfword quantity located at this address becomes the current Location Counter.

CAUTION

When an interrupt is generated that causes a mode change from the halfword mode to the fullword mode, higher-order bits of all registers are undefined with the exception of Register 0 or 14 (register set 0) which contains the old PSW status (See Figure 2.)

INTERRUPT	OLD PSW	NEW PSW	OTHER PARAMETERS
MACHINE MALFUNCTION	MEM. LOC. 20-27	MEM. LOC. 38-3F	
ILLEGAL INSTRUCTION	REG. 14,15	MEM. LOC. 30-37	
ARITHMETIC FAULT	REG. 14,15	MEM. LOC. 48-4F	
MAC INTERRUPT	REG. 14,15	MEM. LOC. 90-97	
SVC INTERRUPT	REG. 14,15	MEM. LOC. 98-9B (STATUS) ONE OF 9C...BB	REG. 13 = ADDRESS OF SVC PARAMETER BLOCK
IMMEDIATE INTERRUPT	REG. 0,1	NEW PSW = X'00002800'	REG. 2 = DEVICE ADDR. REG. 3 = DEVICE STATUS REG. 4 = INTRPT ADDR.
SYSTEM Q SERVICE	REG. 14,15	MEM. LOC. 80-8F	REG. 13 = ADDR. OF SYSTEM Q.

Figure 2. Register Set 0 Modifications By Interrupts

Except SINT, LPSW, and EPSR, all privileged instructions are also executed as defined by 16 bit architecture (refer to the 7/16 Reference Manual, 29-398, or the User's Manual, 29-261). Supervisor call (SVC), and Simulate Interrupt (SINT) instructions are executed as defined in the 32-bit series architecture manual. Load Program Status Word (LPSW) and Exchange Program Status Register (EPSR) instructions are executed as described below:

LSPW:

PSW(16:31) ← A+(X2)
PSW(48:63) ← A+(X2)+2

EPSR:

PSW(16:31) → (R1)
PSW(16:31) ← (R2)

where: A - 16-bit address specified by RX instruction
X2 - The address of a general register, the contents of which is used as an index value
R1 - First operand register
R2 - Second operand register

It should be noted that in halfword mode, all the instructions shown in Appendix 2 use 16-bit General Registers. These 16-bit General Registers are defined by the low-order portion (Bits 16-31) of the 32-Bit General Registers. The high-order portion (bits 0-15) of all the 32-bit General Registers are undefined in halfword mode.

The interrupt system in halfword mode is different from the interrupt system on 16-bit processors, as described in the following sections.

Bit 23 of the Program Status Word controls the protect mode of the 7/32 Processor. If this bit is set, the processor is in protect mode, and only non-privileged instructions may be executed. The halfword mode bit (Bit 11) and protect mode bit define a total of four machine modes for the 7/32 processor. The following paragraphs define these machine modes. Refer to Figures 3 and 4.

Mode #	PSW Bit 11	PSW Bit 23	Machine Mode	Normal Usage	Legal Instructions	External Interrupt Handling
00	0	0	Fullword Non-protect	O. S., 32-bit Stand-alone Programs	All Model 7/32 Instructions (see Appendix 1)	Interrupt Service Pointer Table LOC. X'D0 thru X'2CE'
01	0	1	Fullword Protect	Fullword User Prog.	All Model 7/32 Non-privileged Instructions (see Appendix 1)	Interrupt Service Pointer Table LOC. X'D0 thru X'2CE'
11	1	1	Halfword Protect	Halfword User Prog.	All Model 7/32 Halfword Mode Non-privileged Instructions (see Appendix 2)	Interrupt Service pointer table LOC. X'D0' thru X'2CE'
10	1	0	Halfword Non-Protect	Halfword Peripheral Test Prog. (see Appendix 4)	All Model 7/32 Halfword Mode instructions with certain restrictions	LOC. X'40'-'43' = Old PSW X'44'-'47' - New PSW

Figure 3. Machine Modes

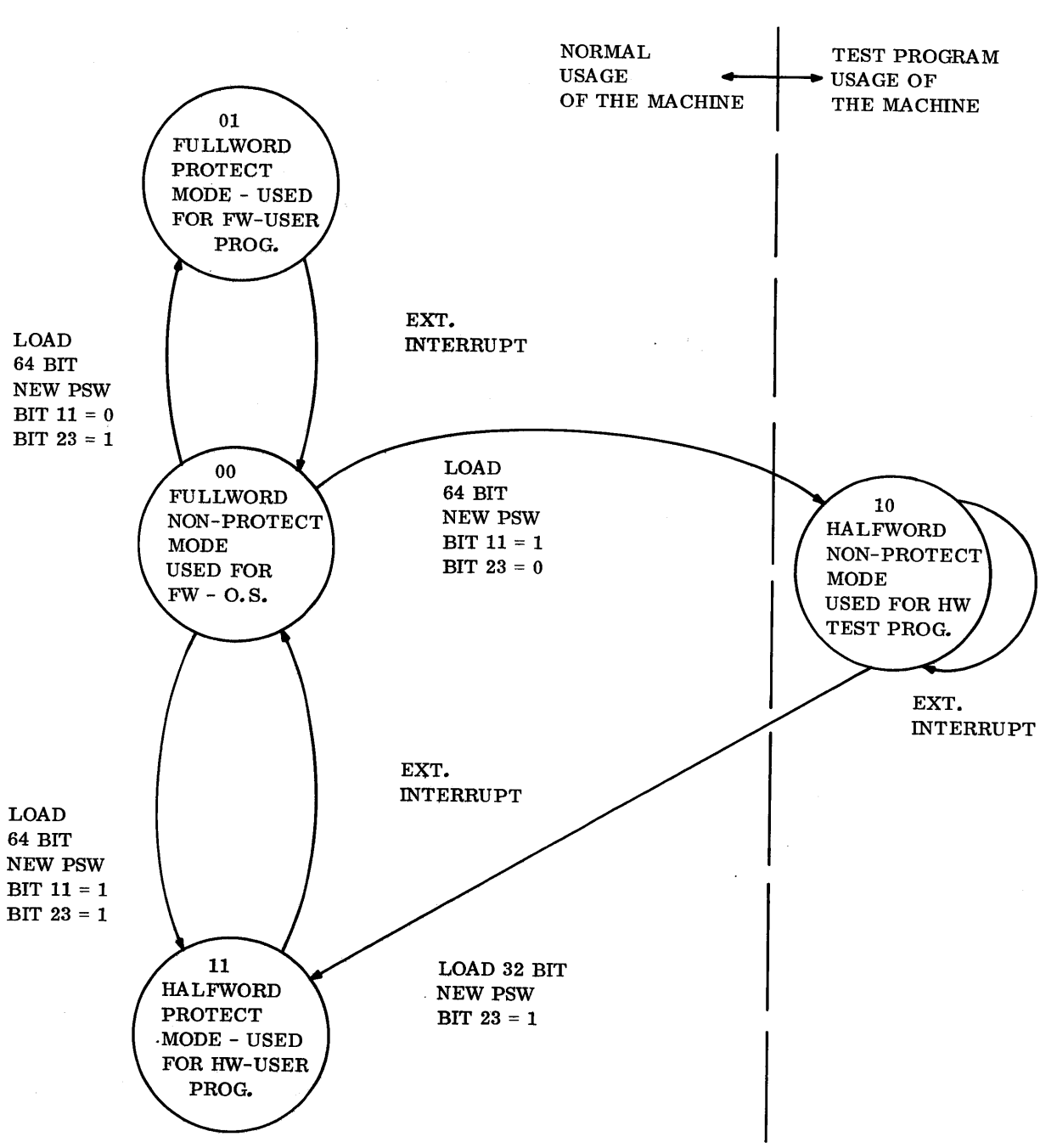


Figure 4. Model 7/32 Machine Mode Transition by an External Interrupt

2.3.1 Mode 00: Bit 11 = 0, Bit 23 = 0. This is the Fullword Non-protect mode. All the instructions in Appendix 1 - privileged and non-privileged - can be executed in this mode. Note that the following instructions, which are legal in the halfword mode, are illegal in this mode:

Mnemonic	Op-Code	Name
ACHR	OE	Add With Carry Halfword RR
SCHR	OF	Subtract With Carry Halfword RR
ACH	4E	Add With Carry Halfword
SCH	4F	Subtract With Carry Halfword
MHUR	9C	Multiply Halfword Unsigned RR
AIR	9F	Acknowledge Interrupt RR
MHU	DC	Multiply Halfword Unsigned
AI	DF	Acknowledge Interrupt

When an external interrupt occurs in this mode and if it is enabled by bit 17 of the current PSW, the interrupt Service pointer table (memory locations X'D0' through X'2CE') is used to perform an immediate interrupt or an Auto Driver channel operation. This mode is normally used by the Operating System and other stand-alone programs for Model 7/32 for privileged instruction execution. Refer to 29-365 for details.

2.3.2 Mode 01: Bit 11 = 0, Bit 23 = 1. This is the Fullword Protect mode. All the nonprivileged instructions in the Model 7/32 instruction repertoire are executable in this mode. External interrupts are treated in the same manner as in Mode 00. The machine will be in this mode when Fullword user programs are run under an operating system.

2.3.3 Mode 11: Bit 11 = 1, Bit 23 = 1. This is the Halfword Protect Mode. All the nonprivileged instructions shown in Appendix 2 can be executed in this mode and (except SVC) they work exactly the same way as they work on 16-bit processors. The SVC instruction is executed as in the Fullword Protect mode. The External Interrupts are treated in the same manner as in Mode 00.

The machine will be in this mode when Halfword user programs are run under an operating system. An operating system for Model 7/32 can use this mode to run user programs written for 16-bit processors.

2.3.4 Mode 10: Bit 11 = 1, Bit 23 = 0. This is the Halfword Non-protect Mode. It is a special mode. Most programs written for INTERDATA 16-bit processors can be run in this mode. The external interrupts are handled as follows:

CURRENT PSW	(16:31)	——→	MEM. LOC. X'40'-X'41'
CURRENT PSW	(48:63)	——→	MEM. LOC. X'42'-X'43'
NEW PSW	(16:31)	←——	MEM. LOC. X'44'-X'45'
NEW PSW	(48:63)	←——	MEM. LOC. X'46'-X'47'

Note that only 16-bit new PSW status and 16-bit new PSW Location Counter are loaded. The program must acknowledge the interrupt by using either an Acknowledge Interrupt (AI) or Acknowledge Interrupt Register (AIR) instruction.

In Mode 10, all other interrupts are handled the same as in Fullword Modes as explained in Section 2.4. Mode 10 can be used to run peripheral test program software. This is explained in detail later in Section 3. Note that not all old stand alone software runs in this mode because SVC and SINT instructions, which generate internal interrupts, alter the 64 bit new PSW as defined in Section 2.4.

2.3.5 Change of Mode in Model 7/32. In general, a machine mode can be changed by executing an instruction which loads a new PSW to change the state of bits 11 and/or 23 by taking an internal interrupt.

The machine mode can be changed from Mode 00 to any other mode by loading a new 64 bit PSW.

Example 1:

```
        LPSW    PSW1
PSW1    DC      X'00000100'    P=1
        DC      A(LOC1)
LOC1    EQU     *
```

In this example, the new machine mode (at location LOC1) is mode 01, i.e., Fullword Protect mode.

Example 2:

```
         LPSW    PSW2
PSW2     DC      X'00100100'
         DC      A(LOC2)
LOC2     EQU     *
```

Here, the new machine mode is 11, i.e., Halfword Protect mode.

When an external interrupt occurs in mode 01 or 11, the mode is changed to 00. This is done by the hardware (microprogram) and a new PSW of X'00002800' is loaded. A mode change from Mode 01 (or 11) to 00 can occur only by an interrupt since no privileged instruction can be executed, when bit 23 is set, to load a new PSW.

When an external interrupt occurs in Mode 10, a new PSW is loaded from memory locations X'44' - X'45' into PSW bits 16 through 31. Bit 11 remains set. Normally PSW bit 23 will be reset in the new PSW. Contents of memory locations X'46'-X'47' replace the program location counter. Thus, a mode change from 10 to 10 (no change), or 10 to 11 can occur.

The above are the normal mode changes and that is how the machine is expected to be used. However, the following mode changes can occur:

While in mode 10, a new PSW of X'0100' can be loaded to change the mode to 11. However, this is done only by the Processor Test to check this mode.

Finally, the memory locations in the low core can be modified to select any other mode by selecting the appropriate new PSW on an internal interrupt.

The following sequence of instructions will force the fullword non-protect mode from any other mode:

```
2400          LIS     R0,0
4000          STH     R0, X'30'     STORE NEW PSW STATUS
0030
4000          STH     R0, X'32'     TO SELECT MODE 00
0032
4000          STH     R0, X'34'
0034
C810          LHI     R1, ILGL      STORE NEW PSW LOC.
XXXX
4010          STH     R1, X'36'
0060
0000          DC      X'0'          ILLEGAL INSTRUCTION
XXXX ILGL     EQU     *             MACHINE NOW IN MODE 00
```

2.4 Interrupt System

When bit 11 of the current PSW is set, the processor is in the halfword mode. Refer to Section 2.3.4 for the description of external interrupt mechanism in the halfword mode. The interrupt mechanism for all other interrupts does not depend on the state of bit 11. Refer to Figure 2.

With the exception of the machine malfunction interrupt, when the current PSW becomes the old PSW it is saved in a pair of registers belonging to register set 0. The machine malfunction old PSW is stored in a reserved memory location. Again with one exception, when a new PSW becomes the current PSW, it is loaded from a reserved memory location. The exception is the immediate interrupt. On an immediate interrupt, the current status is forced to a pre-determined value (X'00002800'). The current Location Counter is loaded from the interrupt service pointer table.

The new Program Status Word for any interrupt should, if possible, disable interrupts of its own class, and should specify register set 0 to avoid the overhead of saving registers.

2.4.1 Immediate Interrupt. The immediate interrupt is used for I/O control. Through this mechanism, external devices can request and obtain Processor service. Bit 17 of the current PSW controls the immediate interrupt. If this bit is set, the Processor is responsive to device requests. If this bit is reset, requests are queued until the Processor is able to recognize them. When the Processor recognizes a request from a device it:

Saves the current PSW in Registers 0 and 1 of general register set 0.

Loads the status portion (bits 0:31) of the current PSW with a value of X'00002800'.

Acknowledges the request and obtains the device number and status from the device. The device number is placed in Register 2 of the register set 0. The status is placed in Register 3.

Adds two times the device number to X'0000D0' (the starting location of the interrupt service pointer table) to obtain the halfword address within the table that corresponds to the interrupting device. For the immediate interrupt, the value in the table must be even. The halfword value in the table becomes the current Location Counter.

In setting up the registers for the immediate interrupt service routine, the Processor loads the device number into the least significant 10 bits and status into the least significant 8 bits of Registers 2 and 3 respectively. The most significant bits are forced to zero.

Note that the current PSW for immediate interrupts disables the immediate interrupt and specifies register set 0. If it is desired to run the interrupt routine with interrupts enabled, the routine must save the information contained in Registers 0:4, and should switch to register set 15.

2.4.2 Console Interrupt. The console interrupt is a special case of the immediate interrupt. It too is controlled by Bit 17 of the current PSW. If this bit is set, a console interrupt is generated by:

Depressing the Function key on the console and
Depressing 0

The effect of the console interrupt is to cause an immediate interrupt, as described previously from device X'001'.

2.4.3 Simulated Interrupt. The Simulate Interrupt instruction simulates an immediate interrupt. When this instruction is executed, the Processor goes through the immediate interrupt procedure as if a request for service had been received from an external device. The current PSW is saved, and the new PSW loaded just as for the immediate interrupt. The device is addressed, and the status returned in Register 3. The halfword address from the interrupt service pointer table is placed in Register 4. The state of Bit 17, immediate interrupt enable, has no effect on this interrupt, as it is always enabled.

2.4.4 Machine Malfunction Interrupt. Bit 18 of the current PSW controls the machine malfunction interrupt. This interrupt, if enabled, occurs on a memory parity error, following the detection of primary power failure, and during the restart procedure after power has been restored. When a machine malfunction interrupt occurs, the current PSW is saved in memory location X'000020'. The new PSW from memory location X'000038' becomes the current PSW. The Condition Code of the new PSW as stored in memory must contain zeros. After the interrupt is taken, the state of the Condition Code indicates the specific cause of the interrupt.

Condition Code states are:

C	V	G	L	
0	0	0	0	Power restore
X	0	0	1	Power failure
0	0	1	0	Parity error on instruction fetch
X	1	0	0	Parity error on data fetch
1	X	0	X	Parity error or power failure during auto driver channel operation

Power failure micro code sequence is executed when the primary power fail detector senses a low voltage, when the Initialize key (INT) of the display console is depressed, when the initialize switch (INT) of a Loader Storage Unit (LSU) is placed in the UP position, or when the key operated Power switch is turned to the OFF position. Following the PSW exchange, the software has approximately one millisecond to perform any necessary operations before the automatic shut down micro code sequence is activated. During automatic shut down, the Processor saves the current PSW at the memory location specified by the contents of location X'000084'. This Halfword save address must exist within the first 65KB of memory. Both sets of general registers, starting with register set 0, are saved at the location specified by the contents of memory location X'000086'. Pointers contained in memory locations X'000084' and X'000086' are absolute addresses. Note that the register save requires 128 consecutive bytes of memory.

When power returns, actual or simulated, the Processor first senses the configuration on the multiplexor bus. If the LSU is not present, it restores the PSW and general registers from their respective store areas. If one is present, the Processor obtains the following information from the user PROMs:

(1) PSW status
(2) PSW location
(3) Addresses where loading from user PROMs is to commence and to be completed
(4) Address to which control is to be transferred following the loading process

In either case, if Bit 18 of the restored PSW is set, the Processor takes another machine malfunction interrupt, this time with no bits set in the Condition Code of the current PSW.

During write operations to memory, the Parity bit of each memory word is set by the hardware to maintain odd parity if the parity option is installed. The Parity bit is recomputed on each memory read. If the computed bit is not equal to the parity bit read out of memory, the Processor generates a machine malfunction interrupt, setting the V or the G flag to indicate error on data fetch or instruction fetch.

If a machine malfunction interrupt condition arises during an auto driver channel operation, the PSW, current at the time the channel was activated, becomes the old machine malfunction PSW. Register 4 of register set 0 contains the address of the Channel Command Block. The C flag of the current PSW is set along with either the L flag or the V flag to indicate either power failure or parity error on a data fetch.

Programming Note:

> If an attempt is made to access a nonexisting memory location, a parity error will be detected if the parity option is installed and machine malfunction is enabled. At this time, a machine malfunction interrupt will be generated, with condition code set to indicate parity error on data fetch. If the machine malfunction interrupt is disabled, no interrupt is generated. In each case the data read consists of all zeros.

2.4.5 Arithmetic Fault Interrupt. Bit 19 of the current PSW controls the arithmetic fault interrupt. This interrupt, if enabled, can occur for any of the following reasons:

Fixed point division by zero
Fixed point quotient overflow
Floating point division by zero
Floating point overflow or underflow

When this interrupt occurs, the current PSW is saved in Registers 14 and 15 of register set 0. The new PSW, from memory location X'000048', becomes the current PSW. All Condition Code bits in the new PSW as stored in memory must be zero. Before going to the interrupt service routine, the Processor sets the carry flag in the Condition Code if the interrupt is the result of a floating point operation. If the interrupt is the result of a fixed point operation, the carry flag is not set.

Any of the following conditions cause fixed point quotient overflow:

A halfword divide operation produces a result greater than 32,767.
A halfword divide operation produces a result less than -32,768.
A fullword divide operation produces a result greater than 2,147,483,647.
A fullword divide operation produces a result less than -2,147,483,648.

When a fixed point division by zero or a fixed point quotient overflow occurs, the operand registers remain unchanged.

Floating point overflow occurs when in a floating point operation, the value of the exponent exceeds 63. Floating point underflow occurs when, during the execution of a Floating Point instruction, the value of the exponent becomes less than 64. Following floating point overflow, the result is forced to plus or minus X'7FFF FFFF'. Following a floating point underflow, the result is forced to true zero X'0000 0000'. After a floating point division by zero, the operand register remains unchanged.

After any arithmetic fault interrupt, the Location Counter of the old PSW contains the address of the instruction immediately following the one that caused the interrupt.

2.4.6 Relocation/Protection Interrupt. Bit 21 of the current PSW controls the relocation/protection interrupt. If this bit is set, and the currently running program violates any of the relocation and protection conditions available in the Memory Access Controller, the Processor saves the current PSW in Registers 14 and 15 of register set 0. The new PSW at memory location X'000090' becomes the current PSW.

2.4.7 System Queue Service Interrupt. Memory location X'000080' contains the address of the system queue. In the course of executing any of the following instructions:

Load Program Status Word
Load Program Status Word Register
Exchange Program Status

the Processor tests Bit 22 of the new status being loaded. If this bit is set, the Processor checks the state of the system queue. If there is an entry in the queue, the just loaded PSW becomes the old PSW. It is saved in Registers 14 and 15 of register set 0. The address of the queue, taken from location X'000080' is placed in Register 13 of register set 0. The new PSW from location X'000088' becomes the current PSW.

2.4.8 Protect Mode Violation Interrupt. Bit 23 of the current PSW controls the execution of Privileged instructions. When this bit is set, the Processor is in the Protect mode. Programs running in the Protect mode are not allowed to execute Privileged instructions. Privileged instructions are:

All I/O instructions
Load Program Status Word
Load Program Status Word Register
Exchange Program Status Register
Simulate Interrupt
Simulate Channel Program

If a program running in the protect mode attempts to execute a Privileged instruction, the instruction is not executed. The Processor saves the current PSW in Registers 14 (PSW Bits 0:31) and 15 (PSW Bits 32:63) of register set 0. The new PSW at location X'000030' becomes the current PSW. The Location Counter of the old PSW (register 15, set 0) contains the address of the privileged instruction.

Programming Note: Refer to Appendix 1 for the privileged instructions in the fullword mode (PSW Bit 11 = 0). Refer to Appendix 2 for the privileged instructions in the halfword mode (PSW bit 11 = 1).

2.4.9 Illegal Instruction Interrupt. The illegal instruction interrupt cannot be disabled. The interrupt occurs whenever the Processor fetches an instruction word containing an operation code that is not one of those permitted by the system. The Processor saves the current PSW in Registers 14 and 15 of register set 0. The illegal instruction new PSW from memory location X'000030' becomes the current PSW.

When the Processor encounters an illegal instruction, it makes no attempt to execute it. The Location Counter of the old PSW (register 15 set 0) contains the address of the illegal instruction.

Refer to Appendix 1 (or Appendix 2) for the legal op-codes in the fullword (or halfword) mode.

2.4.10 Supervisor Call Interrupt. This interrupt occurs as the result of the execution of a Supervisor Call instruction. This instruction provides a means for user level programs to communicate with system programs. The supervisor call interrupt is always enabled. When the Processor executes a Supervisor Call instruction, it:

- Saves the current PSW in Registers 14 and 15 of register set 0.
- Places the address of the supervisor call parameter block (address of the second operand) in Register 13 of register set 0.
- Loads the current PSW status with the value contained at memory location X'00098', supervisor call new status.
- Loads the current PSW Location Counter from one of the supervisor call new PSW Location Counter locations.

3. PERIPHERAL TEST PROGRAMS FOR MODEL 7/32 IN THE HALFWORD MODE

A number of test programs are furnished with Model 7/32 to help the user in trouble-shooting. There are two types of test programs: those which run in the fullword mode and those which run in the halfword mode. The test programs which run on all 32 Bit Series machines are designed to run in the fullword mode (Bit 11 of the PSW = 0). These include Series 32 Processor Tests, Series 32 Memory Tests, Extended Selector Channel test, Series 32 System Exerciser, etc. The corresponding program descriptions include detailed operating instructions for running these programs.

Some test programs are written using Model 7/16 Instruction repertoire (e.g., Single Address 360/370 Interface Test (06-166). They run on Model 7/16 as well as on Model 7/32. It is necessary for the user to change certain parameters to run these programs on Model 7/32. Refer to the corresponding program descriptions for these modifications.

Some of the peripheral test programs which are used on INTERDATA 16 Bit machines (e.g., Card Reader Test 06-038) also run on the Model 7/32 in the halfword mode with certain limitations. Refer to Appendix 4 for a list of such programs.

The corresponding program descriptions explain how to load and execute these programs on 16-Bit processors.

In order to run these programs on Model 7/32, the following procedure should be followed:

1. Load the special 50 Sequence described in Figure 5. Read each memory location to verify that the data is loaded correctly.

2. Address memory location X'30' using the Console Panel switches. Depress the SGL switch. The processor will try to execute the illegal instruction X'00'. An illegal instruction interrupt is generated and the new PSW from memory location X'30' is loaded. This new PSW has PSW bit 11 set and therefore the processor is in the halfword mode. The new location counter at this time is X'50'.

3. Place the program paper tape for the General Loader (06-025) into the paper tape reader. Depress RUN switch on the Console Panel. The 50 Sequence loads the General Loader at the top of memory if it is less than 64KB or at memory location X'FA00' if more than 64KB of memory is in the machine.

4. Refer to the General Loader description and set the bias value to load the test program. The bias value must be selected so that the entire test program can be loaded in the first 64KB of memory. Also if the Memory Access Controller (MAC) is present, the bias value must be above the memory locations assigned to the MAC registers. This can be achieved by selecting a bias value larger than X'A00'. Using the Console Panel write data X'4000' into memory location X'80' and X'400C' into location X'84'.

5. Load the test program using the General Loader.

6. The test program can then be executed referring to the Operating procedures in the test program description.

7. It must be noted that only the external interrupt handling in Model 7/32 is the same as in 16 Bit processors. No other interrupts are executed the same. Therefore, if a program has any other interrupts enabled, and if one occurs, it will be handled as explained in Section 2. Since 16-bit oriented software would not have low memory locations set up for the 7/32 Old and New PSW locations, this action may change the processor mode to a fullword mode depending on the low core set up. The results are totally unpredictable in such a situation and it is recommended that the user reload the program.

MEMORY LOC	DATA	MEANING
0030	0010	Illegal Instruction Interrupt
0032	00F0	New PSW (STATUS and LOC)
0034	0000	
0036	0050	
0038	0010	Machine Malfunction New PSW
003A	00F0	
003C	0000	
003E	0050	
0048	0010	Arithmetic Fault New PSW
004A	00F0	
004C	0000	
004E	0050	
0050	D500	AL X'CF'
0052	00CF	
0054	4300	B X'80'
0056	0080	
0078	XX YY	XXYY = 0294 for a Teletype Paper Tape Reader = 0399 for a High Speed Paper Tape Reader = 1399 for a High Speed Paper Reader/Punch

50 Sequence to load the General Loader (06-025) on Model 7/32 in the halfword mode.

Figure 5. Special 50 Sequence

APPENDIX 1
MODEL 7/32
LEGAL OP-CODES IN FULLWORD MODE
PSW BIT 11 = 0

	0	1	2	3	4	5	6	7	9	C	D	E	F
0		SRLS	BTBS		STH	ST	STE		SRHLS	BXH	STM	TS	
1	BALR	SLLS	BTFS		BAL	AM	AHM	STME	SLHLS	BXLE	LM	SVC	
2	BTCR	CHVR	BFBS		BTC			LME	STBR	*LPSW	STB	*SINT	
3	BFCR		BFFS		BFC			LHL	LBR	THI	LB	*SCP	TI
4	NR		LIS	EXHR	NH	N	ATL	TBT	EXBR	NHI	CLB		NI
5	CLR		LCS		CLH	CL	ABL	SBT	*EPSR	CLHI	*AL		CLI
6	OR		AIS		OH	O	RTL	RBT	*WBR	OHI	*WB	LA	OI
7	XR		SIS		XH	X	RBL	CBT	*RBR	XHI	*RB	TLATE	XI
8	LR	*LPSWR	LER		LH	L	LE		*WHR	LHI	*WH		LI
9	CR		CER		CH	C	CE		*RHR	CHI	*RH		CI
A	AR		AER		AH	A	AE		*WDR	AHI	*WD	RRL	AI
B	SR		SER		SH	S	SE		*RDR	SHI	*RD	RLL	SI
C	MHR	MR	MER		MH	M	ME			SRHL		SRL	
D	DHR	DR	DER		DH	D	DE		*SSR	SLHL	*SS	SLL	
E			FXR			CRC12			*OCR	SRHA	*OC	SRA	
F			FLR			CRC16				SLHA		SLA	

* Privileged Instructions

APPENDIX 2
MODEL 7/32
LEGAL OP-CODES IN HALFWORD MODE
(PSW BIT 11 = 1)

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

* Privileged Instructions

APPENDIX 3

INSTRUCTION EXECUTION TIMES

INSTRUCTION	EXECUTION TIME IN MICROSECONDS (FOR 750 NANOSECOND MEMORY)					COMMENTS
	RR or SF FORMAT	RI 1 FORMAT	RI2 FORMAT	RX1 or RX2 FORMAT	RX3 FORMAT	
L				3.25	3.75	
LR	1.0					
LI			2.5			
LIS	1.0					
LCS	1.5					
LH				2.75	3.25	
LHI		1.75				
LHS				3.5	4.0	
LHL				2.75	3.25	
LA				2.25	2.75	
LM				3.5+1.5n	4.0+1.5n	n=no. of registers to be loaded
ST				3.5	4.0	
STH				2.5	3.0	
STM				2.75+1.5n	3.25+1.5n	n=no. of registers to be stored
EXHR	1.0					
A				3.25	3.75	
AR	1.0					
AI			2.5			
AIS	1.25					
AH				2.75	3.25	
AHI		1.75				
AM				5	5.5	
AHM				3.5	4	
S				3.25	3.75	
SR	1.0					
SI			2.5			
SIS	1.25					
SH				2.75	3.25	
SHI		1.75				
C				4.25/4.5	4.75/5	SIGNS ALIKE/ SIGNS DIFFER
CR	2.0/2.25					SIGNS ALIKE/ SIGNS DIFFER
CI			3.5/3.75			SIGNS ALIKE/ SIGNS DIFFER
CH				3.75/4.0	4.25/4.5	SIGNS ALIKE/ SIGNS DIFFER

INSTRUCTION	EXECUTION TIME IN MICROSECONDS (FOR 750 NANOSECOND MEMORY)					COMMENTS
	RR or SF FORMAT	RI1 FORMAT	R12 FORMAT	RX1 or RX2 FORMAT	RX3 FORMAT	
CHI		2.75/3.0				SIGNS ALIKE/ SIGNS DIFFER
CHVR	2.75/4.0					NO OVF/OVF
M				24.0/26/ 26.5/25	24.5/26.5 27/25.5	++,+-,-+,--
MR	21.75/23.75/ 24.25/22.75					++,+-,-+,--
MH				6.25	6.75	
MHR	4.25					
D				82.5/88.25/ 96.25	83/88.75/ 96.75	MINIMUM/AVERAGE/ MAXIMUM
DR	80.25/86/94					MINIMUM/AVERAGE/ MAXIMUM
DH				13.0/13.0 14.25/13.0	13.5/13.5/ 14.75/13.5	++,+- (AVE),-+ (AVE),--
DHR	11/11/12.25					++,+- (AVE),-+ (AVE),--
	11					
N				3.25	3.75	
NR	1.0					
NI			2.5			
NH				2.75	3.25	
NHI		1.75				
O				3.25	3.75	
OR	1.0					
OI			2.5			
OH				2.75	3.25	
OHI		1.75				
X				3.25	3.75	
XR	1.0					
XI			2.5			
XH				2.75	3.25	
XHI		1.75				
CL				3.25	3.75	
CLR	1.0					
CLI			2.5			
CLH				2.75	3.25	
CLHI		1.75				
TI			2.5			
THI		1.75				
SRL		3.25+[n-2]* .25				n=No. of SHIFTS

INSTRUCTION	(FOR 750 NANOSECOND MEMORY)					
	RR or SF FORMAT	RI1 FORMAT	R12 FORMAT	RX1 or RX2 FORMAT	RX3 FORMAT	COMMENTS
SRHLS		$1.75+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
SRHL		$2.25+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
SRLS		$2.75+\lceil \frac{n-2}{2} \rceil^*$				n=No. of SHIFTS
SLL		$3.25+\lceil \frac{n-2}{2} \rceil^*$ .25				n=No. of SHIFTS
SRA		$3.75+\lceil \frac{n-2}{2} \rceil$	*.25			n=No. of SHIFTS
SLHLS		$1.75+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
SLHL		$2.25+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
SLLS		$2.75+\lceil \frac{n-2}{2} \rceil^*$ .25				n=No. of SHIFTS
SRHA		$2.5+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
SLA		$3.75+\lceil \frac{n-2}{2} \rceil^*$ .25				n=No. of SHIFTS
SLHA		$2.75+\lceil n-1\rceil^*$ .25				n=No. of SHIFTS
RRL		2.25/1.75+ 1.0n				n=0/n>0 (n=No. of SHIFTS)
RLL		2.25/1.75+ 1.0n				n=0/n>0 (n=No. of SHIFTS)
LB				2.75	3.25	
LBR	1.25					
STB				3.25	3.75	
STBR	2.0					
EXBR	1.0					
CLB				3.0	3.5	
BTC				2.0/2.0	2.0/2.5	NO BR/BR
BTCR	1.5					
BTFS	1.5/2.0					NO BR/BR
BTBS	1.5/2.0					NO BR/BR
BFC				2.0/2.0	2.0/2.5	NO BR/BR
BFCR	1.5					
BFFS	1.5/2.0					NO BR/BR
BFBS	1.5/2.0					NO BR/BR
BAL				2.0	2.5	
BALR	1.5					
BXH				4.75/4.25	5.25/4.75	NO BR/BR

INSTRUCTION	EXECUTION TIME IN MICROSECONDS (FOR 750 NANOSECOND MEMORY)					COMMENTS
	RR or SF FORMAT	RI1 FORMAT	R12 FORMAT	RX1 or RX2 FORMAT	RX3 FORMAT	
BXLE				4.75/4.25	5.25/4.75	NO BR/BR
TBT				6.0/6.75/ 7.5	6.5/7.25/ 8.0	MIN/AVE/MAX
SBT				6.25/7.0/ 7.75	6.75/7.5/ 8.25	MIN/AVE/MAX
RBT				6.25/7.0/ 7.75	6.75/7.5/ 8.25	MIN/AVE/MAX
CBT				6.25/7.0 7.75	6.75/7.5 8.25	MIN/AVE/MAX
ATL				4.75/9.0	5.25/9.5	OVF/NO OVF
ABL				4.75/9.0	5.25/9.5	OVF/NO OVF
RTL				4.75/9.75	5.25/10.25	EMPTY/NOT EMPTY
RBL				4.75/9.25	5.25/9.75	EMPTY/NOT EMPTY
AE				12/17/22.5	12.5/17.5/ 23	MIN/AVE/MAX
AER	11/16/21.5					MIN/AVE/MAX
SE				12.5/17.5/23	13/18/23.5	MIN/AVE/MAX
SER	11.5/16.5/22					MIN/AVE/MAX
CE				6.5/7.5/8.75	7/8/9.25	MIN/AVE/MAX
CER	5.5/6.5/7.75					MIN/AVE/MAX
ME				29.25/29.5/ 31.25	29.75/30/ 31.75	MIN/AVE/MAX
MER	28.25/28.5/ 30.25					MIN/AVE/MAX
DE				48.5/48.5/ 50.25	49/49/ 50.75	MIN/AVE/MAX
DER	47.5/47.5/ 49.25					MIN/AVE/MAX
LE				7.5/10.5/ 14.25	8.0/11/ 14.75	MIN/AVE/MAX
LER	6.5/9.5/ 13.25					MIN/AVE/MAX
LME				7.25+[n-1] * 3.0	7.75+[n-1] * 3.0	n=No. of Float. Reg. to be loaded
STE				5.25	5.75	
STME				7.5+[n-1] * 3.0	8.0+[n-1] * 3.0	n=No. of Float. Reg to be stored
FXR	8/10/16					MIN/AVE/MAX
FLR	10.5/13.75/ 18					MIN/AVE/MAX
LPSW				6.0	6.5	
LPSWR	3.0					
EPSR	3.25					
SVC				6.25	6.75	

INSTRUCTION	EXECUTION TIME IN MICROSECONDS (FOR 750 NANOSECOND MEMORY)					COMMENTS
	RR or SF FORMAT	RI1 FORMAT	R12 FORMAT	RX1 or RX2 FORMAT	RX3 FORMAT	
AL				8.75+2.5 L+ 2.75n	9.25+2.5 L +2.75n	L=Leader Bytes; n=data bytes
SINT		1.0+Interrupt processing time				
RD				3.75	4.25	
RDR	2.25					
RH				4.75/4.0	5.25/4.5	Byte Dev/Halfword Dev
RHR	3.0/2.25					Byte Dev/Halfword Dev
RB				6.75+2.75n	7.25+2.75n	n=No. of Bytes
RBR	4.75+2.75n					n=No. of Bytes
WD				3.75	4.25	
WDR	2.25					
WH				4.25/3.5	4.75/4.0	Byte Dev/Halfword Dev
WHR	3.25/2.5					Byte Dev/Halfword Dev
WB				6.25+3.0n	6.75+3.0n	n=No. of Bytes
WBR	4.25+3.0n					n=No. of Bytes
SS				4.0	4.5	
SSR	3.0					
OC				4.0	4.5	
OCR	2.25					
CRC12				11.5/13.25/ 15	12/13.75/ 15.5	MIN/AVE/MAX
CRC16				13/15.25/ 17.5	13.5/15.75 18	MIN/AVE/MAX
TLATE				4.5/5.25	5.0/5.75	Translation/ Special Char.
SCP				5.5/9.25/	6.0/9.75	Byte Count > 0/ Inc. Byte Count
				9.75/11.0	10.25/11.5	Incr. Byte Count ≤ 0 (Fast Mode)/ Incr. Byte Count > 0 (Normal Mode)

NOTE: [n−1] is zero if n = 0

$\left[\frac{n-2}{2}\right]$ is maximum positive integer value less than or equal to $\frac{n-2}{2}$

(for n = 1,2,or 3 $\left[\frac{n-2}{2}\right] = 0$)

AUTO DRIVER CHANNEL EXECUTION TIMES (IN MICROSECONDS)

FAST MODE (F BIT = 1)

FUNCTION	EXE-CUTE BIT RESET	BAD DEVICE STATUS	BUFFER BYTE COUNT IS > 0	INCR. BUFFER BYTE COUN IS NOT >0	INCR. BUFFER BYTE CT. IS > 0
READ (BYTE DEV.)	9.5	10.75	12.0	15.75	16.0
READ (HALFWORD	9.5	10.75	12.0	16.5	16.75
WRITE (BYTE DEV.)	9.5	10.75	12.0	15.75	16.0
WRITE (HALFWORD DEV.)	9.5	10.75	12.0	17.0	17.25

NORMAL MODE (F BIT = 0)

FUNCTION	EXECUTE BIT RESET	BAD DEVICE STATUS	BUFFER BYTE COUNT > 0	INCR. BUFFER BYTE COUNT IS NOT>0	INCR. BUFFER BYTE COUNT > 0	NOTES
LRC, BUFFER 0, READ	9.5	10.75	13.0	20.25	20.75	#
LRC, BUFFER 0, READ, TRANSLATE	9.5	10.75	13.0	23.5/21	24.0	#
LRC, BUFFER 0, WRITE	9.5	10.75	13.0	20.5	21.0	#
LRC, BUFFER 0, WRITE, TRANSLATE	9.5	10.75	13.0	24.5/19.5	25.0	#
LRC, BUFFER 1, READ	9.5	10.75	13.75	21.0	21.5	
LRC, BUFFER 1, READ, TRANSLATE	9.5	10.75	13.75	24.25/21.75	24.75	#
LRC, BUFFER 1, WRITE	9.5	10.75	13.75	21.25	21.75	
LRC, BUFFER 1, WRITE, TRANSLATE	9.5	10.75	13.75	25.25/20.25	25.75	#
CRC, BUFFER 0, READ	9.5	10.75	13.0	30.5/32.75/35	31/33.25/35.5	*
CRC, BUFFER 0, READ, TRANSLATE	9.5	10.75	13.0	33.75/36/38.25	34.25/36.5/38.75	*
CRC, BUFFER 0, WRITE	9.5	10.75	13.0	30.75/33/35.25	31.25/33.5/35.75	*
CRC, BUFFER 0, WRITE, TRANSLATE	9.5	10.75	13.0	34.75/37/39.25	35.25/37.5/39.75	*
CRC, BUFFER 1, READ	9.5	10.75	13.75	31.25/33.5/35.75	31.75/34/36.25	*
CRC, BUFFER 1, READ, TRANSLATE	9.5	10.75	13.75	34.5/36.75/39	35/37.25/39.5	*
CRC, BUFFER 1, WRITE	9.5	10.75	13.75	31.5/33.75/36	32/34.25/36.5	*
CRC, BUFFER 1, WRITE, TRANSLATE	9.5	10.75	13.75	35.5/37.75/40	36/38.25/40.5	*

– NORM TRANS/SPECIAL CHAR.

* – MIN/AVE/MAX

INTERRUPT RESPONSE TIMES (IN MICROSECONDS)		COMMENTS
IMMEDIATE INTERRUPT	6.0	
MACHINE MALFUNCTION INT.	9.0	QUEUE EMPTY/QUEUE NOT EMPTY
SYSTEM QUEUE INTERRUPT	4.5/11.5	
NORMAL INTERRUPT LATENCY TIME	4	

NOTE: FOLLOWING USER INSTRUCTIONS CAN BE ABORTED BY I/O INTERRUPTS.

LE

LER

AE

AER

SE

SER

CE

CER

ME

MER

DE

DER

FLR

M

MR

D

DR

APPENDIX 4
PERPHERAL TEST PROGRAMS WHICH RUN ON MODEL 7/32 IN THE HALFWORD MODE

1.	06–038	Card Reader Test
2.	06–071	Auto Call Unit Test
3.	06–101	Digital MPX Test
4.	06–102	Line Printer Test
5.	06–122	Disc Test/Formatter
6.	06–127	PALS Off-Line Test
7.	06–129	Universal Logic Interface Test
8.	06–131	Magnetic Tape and Cassette Test
9.	06–132	201 Data Set Adapter Test
10.	06–133	Universal Clock Module Test
11.	06–137	High Speed Paper Tape Reader/Punch Test
12.	06–146	CRT Test Program
13.	06–147	Conversion Equipment Test Program
14.	06–149	MUX Bus Switch Test
15.	06–150	Sense Contact Module Test
16.	06–151	Relay Driver Test
17.	06–152	Decision Card Reader Test
18.	06–163	Tektronix Display Test
19.	06–164	20 Surface Disc Test

09-065R01A22
September 1974

M71-102 HEXADECIMAL DISPLAY PANEL AND M71-101 BINARY DISPLAY PANEL PROGRAMMING SPECIFICATION

1. INTRODUCTION

The M71-102 Hexadecimal Display Panel and M71-101 Binary Display Panel provide a means to manually control the Processor, interrogate and display various Processor registers and machine status, set and display Processor memory locations, and may be programmed as an I/O device by the user. The Hexadecimal Display Panel and Binary Display Panel are identical in operation. For convenience of the operator the Hexadecimal Display is equipped with a Hexadecimal readout in addition to the standard Binary readout.

2. CONFIGURATION

The Hexadecimal Display Panel is available as an option on the Model 7/16 and 32 Bit Series Processors. The Hexadecimal Display Panel provides the system operator with visual indications of the state of the Processor, as well as manual control over the system.

The Hexadecimal Display Panel, shown in Figure 1, is a RETMA standard 5 $\frac{1}{4}$" x 19" panel which is plug removable from the Processor. It displays the current state of the Processor and provides all necessary manual control over the system. The following paragraphs describe the control and display elements of the Hexadecimal Display Panel.

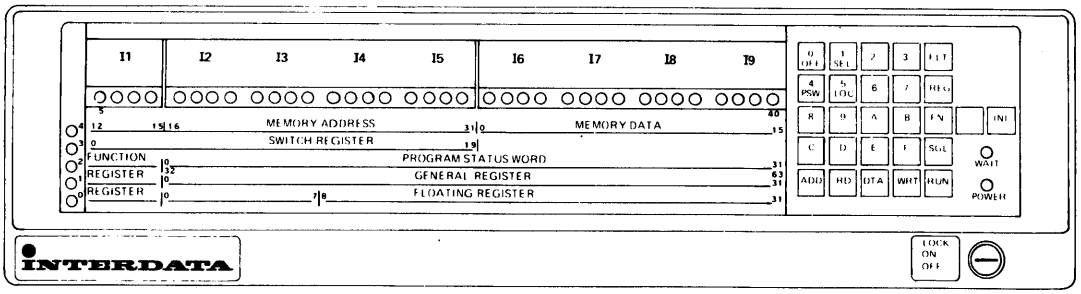


Figure 1. Hexadecimal Display Panel

2.1 Display Registers and Indicators

Internal to the Hexadecimal Display Panel are five eight-bit byte Display Registers, D1 through D5, that hold data output from the Processor, and a 20-bit Switch Register that holds data input from the Hexadecimal Keyboard. Refer to Figure 2.

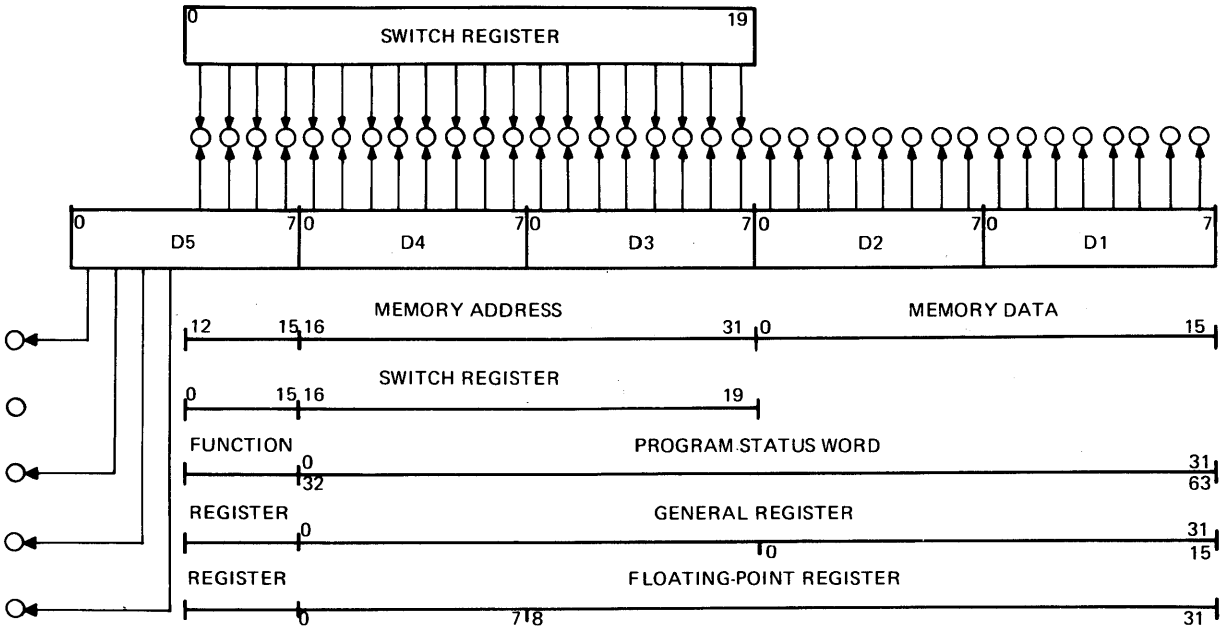


Figure 2. Display Registers And Indicators

Associated with each one of Display Registers D1 through D4 are eight indicator lamps that provide a binary read-out and two optional hexadecimal read-out indicators. Associated with the least significant four bits of Display Register D5 are four indicator lamps for binary display and one optional hexadecimal read-out indicator.

The most significant four bits of Display Register D5 (Bits 0:3) control four of the five indicator lamps along the left edge of the Hexadecimal Display Panel. The fifth indicator lamp is controlled by logic internal to the Hexadecimal Display Panel. To the right of each of these five lamps is a diagram that defines what is being displayed. In general, only one of the diagram lamps is on at a time. If none of the diagram lamps are on, a user program has written data to the display registers.

As seen in Figure 2, the most significant 20-bits of the display show the contents of Display Registers D3 and D4 and the least significant four bits of Display Register D5 (Bits 4:7); or the contents of the 20-bit Switch Register. When the Switch Register is being displayed, the lamp next to the Switch Register diagram is illuminated. Any other diagram lamp that may have been on, remains on. When the Switch Register is no longer displayed, its diagram lamp goes out and the most significant 20-bits of the display again show the contents of Display Registers D3 and D4 and the least significant four bits of Display Register D5 (Bits 4:7).

The methods of displaying the Switch Register and the other diagrammed items are discussed later.

2.2 Key Operated Security Lock

This is a three-position, OFF-ON-LOCK, key-operated locking switch, which controls the primary power to the system. This switch can also disable the Hexadecimal Display Panel, thereby preventing any accidentical manual input to the system. The power indicator lamp (PWR) associated with the key lock is located in the lower right corner of the Hexadecimal Display Panel. The PWR lamp is on when the key lock is in the ON or LOCK position. The relationship between the key lock switch positions, primary power, the Control keys, and the Hexadecimal keys is:

OFF	The primary power is OFF.
ON	The primary power is ON and the Control keys and Hexadecimal keys are enabled.
LOCK	The primary power is ON and the Control keys and Hexadecimal keys are disabled.

2.3 Control Keys

The momentary contact Control keys are only active when the key-operated locking switch is in the ON position.

INITIALIZE (INT) — The Initialize (INT) key causes the system to be initialized. After the initialize operation, all device controllers on the system Multiplexor Bus are cleared and certain other functions in the Processor are reset.

DATA (DTA) — The Data (DTA) key clears the Switch Register and connects it to the most significant 20 display indicators. The Switch Register diagram lamp illuminates. Hexadecimal data may now be entered into the Switch Register from the Hexadecimal Keyboard. As each Hexadecimal key is depressed, the data shifts into the Switch Register from the right. If more than five hexadecimal digits are entered, data shifted out of the Switch Register is lost.

Depressing any non-hexadecimal key disconnects the Switch Register from the high order 20 display lamps and extinguishes the Switch Register diagram lamp.

ADDRESS (ADD) — The Address (ADD) key causes the Processor to halt and copy the contents of the Switch Register into the Location Counter field of the Program Status Word. The new value of the Location Counter is then output to Display Registers D1, D2, D3, and D4. The function diagram lamp is illuminated and a Hexadecimal 5 is output to the top four display lamps.

MEMORY READ (RD) — The Memory Read (RD) key causes the Processor to halt and read the halfword contents of the memory location presently pointed to by the Location Counter. (If the Memory Access Controller is enabled then the relocated value of the Location Counter is the effective address of the memory location.) The halfword data read is output to Display Registers D1 and D2. The Location Counter is incremented by two and output to Display Registers D3 and D4 and the least significant four bits of Display Register D5 (a 20-bit value). The lamp next to the Memory Address/Memory Data diagram is illuminated.

MEMORY WRITE (WRT) — The Memory Write (WRT) key causes the Processor to halt and read in the least significant 16 bits of the 20 bit Switch Register. The halfword of data is written into the memory location presently pointed to by the Location Counter. (If the Memory Access Controller is enabled then the relocated value of the Location Counter is the effective address of the memory location.) The data written is then output to Display Registers D1 and D2. The Location Counter is incremented by two and output to Display Registers D3 and D4 and the least significant four bits of Display Register D5. The lamp next to the Memory Address/Memory Data diagram is illuminated.

EXAMINE REGISTER (REG) — The Examine Register (REG) key sets up the Hexadecimal Display Panel to interpret the next Hexadecimal key depressed as a General Register number. When the hexadecimal register number key is depressed, the Processor halts and the content of the selected General Register is output to Display Registers D1, D2, D3, and D4. The General Register diagram lamp is illuminated and the number of the displayed register is output to the top four display lamps.

EXAMINE FLOATING-POINT REGISTER (FLT)

The Examine Floating-Point Register (FLT) key sets up the Hexadecimal Display Panel to interpret the next hexadecimal key depressed as the number of a Floating-Point Register. When the hexadecimal register number key is depressed, the Processor halts and the content of the selected Floating-Point Register is output to Display Registers D1, D2, D3, and D4. The Floating-Point Register diagram lamp is illuminated and the number of the displayed register is output to the top four display lamps. If an odd numbered register had been selected and the processor is not equipped with double precision option, the register number is forced to the next lower even value before being used. On Processors not equipped with floating-point, the result of this operation is undefined.

FUNCTION (FN)

The Function (FN) key sets up the Hexadecimal Display Panel to interpret the next hexadecimal key depressed as the number of one of sixteen functions. When the hexadecimal key is depressed, the Processor halts to interpret the selected function. If the function is undefined, the Processor remains halted with no change to the display indicators. The defined functions are covered in Section 3.0.

SINGLE STEP (SGL)

The Single Step (SLG) key causes the Processor to execute one user level instruction and then halt. The register that was selected (PSW, LOC, General Register, etc.) is displayed.

RUN (RUN)

The Run (RUN) key causes the Processor to begin program execution at the address pointed to by the Location Counter (LOC).

3. OPERATING PROCEDURES

3.1 Power Up

To power up the system, turn the key-operated security lock clockwise from the OFF position to the ON position. This action provides electrical power to the system and leaves all device controllers on the Multiplexor Bus in an initialized state.

3.2 Power Down

To shut down power to the system:

1. Halt the Processor.
2. Turn the key-operated security lock to the OFF position.

This removes primary power from the system and forces a Primary Power Fail (PPF) interrupt to the Processor. When power is re-applied, the Processor displays the contents of the Location Counter (LOC) and then assumes the Halt mode. If the Processor had been running when power was turned off, the mode assumed when power is re-applied depends upon the presence or absence of the Automatic Restart option. See Section 3.12.

3.3 Address the Processor

To select an address:

1. Depress the Data (DTA) key. The Switch Register is cleared and displayed.
2. Enter the desired address from the Hexadecimal Keyboard.
3. Depress the Address (ADD) key. The Processor halts and copies the contents of the Switch Register into the Location Counter field of the PSW. The new value of the Location Counter is then displayed.

NOTE

The Model 7/16 Processor only uses the least significant 16 bits of the Switch Register.

3.4 Memory Read

To display the contents of memory locations:

1. Select the memory read start address as in 3.3
2. Depress the Read (RD) key. The address read from, plus two, and the data read from memory are displayed.
3. Repeat from Step 2 to read successive memory locations. The Location Counter is automatically incremented by two each time RD is depressed.

3.5 Memory Write

To write data from the Switch Register into memory:

1. Select the memory write start address as in 3.3.
2. Depress the Data (DTA) key. The Switch Register is cleared and displayed.
3. Enter the data to be written from the Hexadecimal Keyboard.
4. Depress the Write (WRT) key. The address written into, plus two, and the data written are displayed.
5. Repeat from Step 2 to write different data into successive locations or from Step 4 to write the same data into successive locations. The Location Counter is automatically incremented by two each time WRT is depressed.

3.6 General Register Display

To examine the contents of a General Register:

1. Depress the Register (REG) key.
2. Depress the hexadecimal register number. The Processor halts and the contents of the selected General Register is displayed.

NOTE

For 32 Bit Series machines, only, the General Register displayed is from the register set specified by the current Program Status Word.

3.7 Floating-Point Register Display

To examine the contents of a Floating-Point Register:

1. Depress the Floating-Point Register (FLT) key.
2. Depress the hexadecimal register number. If the Processor is not equipped with floating-point the result of this operation is undefined. If the Processor is equipped with floating-point, the selected register number is forced even and the Floating-Point Register is displayed. The Processor is left in the Halt mode.

3.8 Program Status Word Display and Modification

To examine the Status field (most significant half) of the current PSW:

1. Depress the Function (FN) key.
2. Depress Hexadecimal key 4. The Processor halts and the status field (most significant half) of PSW is displayed.

To examine the Location Counter field (least significant half) of the current PSW:

1. Depress the Function (FN) key.
2. Depress Hexadecimal key 5. The Processor halts and the Location Counter field (least significant half) of PSW is displayed.

To modify the least significant 16 bits (Bits 16-31) of the Status field:

1. Depress the Data (DATA) key.
2. Enter the data (to be written into bits 16-31 of the PSW) from the Hexidecimal keyboard.
3. Depress the Function (FN) key.
4. Depress Hexadecimal key 1. The Processor halts and copies the 16 bits of the Switch register in bits 16-31 of the PSW. The modified PSW is then displayed.

3.9 Program Execution

To begin execution of a program:

1. Select the program start address as in 3.3.
2. Depress the Run (RUN) key.

To execute a program in the Single-Step mode:

1. Select the program start address as in 3.3.
2. Select the register to be displayed.
3. Depress the Single-Step (SGL) key. One instruction is executed, the last selected register (PSW, LOC, General Register, etc.) is displayed and the Processor halts.
4. Repeat Step 3 to execute successive instructions. Return to Step 2 to display different registers.

3.10 Program Termination

To manually halt the execution of a program, display any register or depress the Single-Step (SGL) key. In the latter case, the last selected register is displayed.

3.11 Console Interrupt

To generate an interrupt from the Hexadecimal Display Panel:

1. Depress the Function (FN) key.
2. Depress Hexadecimal key 0. If enabled by the current PSW, an interrupt from device number 1 is simulated. If not enabled, the Processor enters the Run mode. Hexadecimal Display Panel interrupts are not queued.

The Hexadecimal Display Panel interrupt feature allows an operator to inform the running program that some operator service or function is needed. No acknowledgement of the interrupt is required of the running program.

3.12 Switch Register

To examine the Switch Register at any time during execution of a program, depress any hexadecimal key. The Switch Register is displayed for as long as the key is depressed. No information enters the Switch Register. When the hexadecimal key is released, the top 20 display lamps return to their previous state.

The Switch Register can be modified without interrupting the Processor as follows:

1. Depress the Data (DTA) key. The Switch Register is cleared and displayed.
2. Enter the desired hexadecimal data.

3.13 Power Fail

When the Processor detects a power failure, the micro-program senses the Hexadecimal Display Panel status. The present status of the display is stored in main memory at a dedicated area by the micro-program. The current Program Status Word, Location Counter and the programmable registers are then saved in dedicated main memory locations and the micro-program deactivates the System Clear (SCLR) relay.

On power up, after the system clear relay has re-activated, the Program Status Word, Location Counter, and programmable registers are restored from their main memory save locations. The status of the display prior to the power failure is retrieved and interegated by the micro-program.

If the Hexadecimal Display Panel was in the Run mode, and the Automatic Restart option is present and if the Machine Malfunction Interrupt Enable bit of the PSW is set, a Machine Malfunction Interrupt is taken. If Machine Malfunction Interrupts are not enabled, the Processor enters the Run mode beginning at the instruction pointed to by the Location Counter.

If the Hexadecimal Display Panel was not in the Run mode, or if the Automatic Restart option is not present, the value of the Location Counter is output to the display registers, the WAIT lamp on the console is illuminated and the Halt mode is entered.

Power failure and operation of the Initialize key are indistinguishable to the Micro-Program. Consequently, operation of the Initialize key should be considered carefully when the Machine Malfunction Interrupt is enabled.

Care should also be taken when using the Hexadecimal Display Panel as an input device (testing Switch Register bits) due to the volatility of the Switch Register in a power fail situation.

After a power up, the contents of the Switch Register are undefined. The display status byte is forced to X'40' on power up or initialize.

4. DATA FORMAT

A byte on a halfword can be transferred to and from the Display using WD, WH, WDR, or RD, RH, RHR, RDR instruction. Refer to Figure 3.

REGISTER DISPLAY	D5	D4	D3	D2	D1

SWITCH REGISTER		S2	S1

INSTRUCTIONS EXECUTED	DATA TRANSFERRED	
	NORMAL MODE	INCREMENTAL MODE
RD (R)	S1	S1
RD (R)	S1	S2
RD (R)	S1	S1
RD (R)	S1	S2
RH (R)	S1,S2	S1,S2
RB (R) *	S1,S2,S1.S2	S1,S2,S1,S2
WD (R)	D1	D1
WD (R)	D1	D2
WD (R)	D1	D3
WD (R)	D1	D4
WD (R)	D1	D5
WH (R)	D1,D2	D1,D2
WH (R)	D1,D2	D3,D4
WH (R)	D1,D2	D5,NOTE 1
WB (R) * *	D1,D2,D3,D4,D5	D1,D2,D3,D4,D5

* BLOCK LENGTH = 4 BYTES ** BLOCK LENGTH = 5 BYTES

NOTE 1. SUBSEQUENT BYTES OUTPUT ARE LOST.

Figure 3. Hexadecimal Display Panel Data Transfers

5. PROGRAMMING INSTRUCTIONS

5.1 Input/Output Programming

The Hexadecimal Display Panel is available to any running program as an I/O device with device address 01. The status and command bytes for the Hexadecimal Display Panel are summarized in Table 1. The status byte indicates the mode of the Hexadecimal Display Panel and is of little interest to a running program as it always indicates Run mode or Hexadecimal Display Panel Interrupt (Function 0). The command byte selects Normal or Incremental mode, which pertains to data Transfers. The selection logic which determines the Switch Register byte or register display byte to transfer is reset every time the Hexadecimal Display Panel is addressed when in the Normal mode. When an Output Command Incremental mode is issued to the Hexadecimal Display Panel, the byte selection logic is initially reset. Subsequent Read or Write instructions transfer bytes as shown in Figure 3.

Block I/O with the Hexadecimal Display Panel is only feasible when the least significant four status bits are reset.

NOTE

After an initialize sequence or after any manual Hexadecimal Display Panel operation that results in anything being displayed, the Display Device Controller is automatically placed in the Normal mode.

When programming the Hexadecimal Display Panel in the Incremental mode, the Output Command Incremental mode must be issued before each set of data transfers to guarantee that the byte selection logic is reset.

The most significant four bits of the Switch Register are only available to the micro-program. These four bits are transferred as Bits 5, 6, 7, and 0 of the status when the Hexadecimal Display Panel status is Address (i.e., Display Status = X011XXXX'.

TABLE 1. DISPLAY STATUS AND COMMAND

STATUS

	0	1	2	3	4	5	6	7	
Run	X	0	0	0	X	X	X	X	
Memory write	X	0	0	1	X	X	X	X	
Memory read	X	0	1	0	X	X	X	X	
Address	X	0	1	1	X	X	X	X	
Fixed Register	X	1	0	0	X	X	X	X	} Single or Halt
Floating Register	X	1	0	1	X	X	X	X	} Single or Halt
Function	X	1	0	0	X	X	X	X	} Single or Halt

General Register	0	0	X	X	X	1	0	0	0
	1	1	X	X	X	1	0	0	0
	2	0	X	X	X	1	0	0	1
	3	1	X	X	X	1	0	0	1
	4	0	X	X	X	1	0	1	0
	5	1	X	X	X	1	0	1	0
	6	0	X	X	X	1	0	1	1
	7	1	X	X	X	1	0	1	1
	8	0	X	X	X	1	1	0	0
	9	1	X	X	X	1	1	0	0
	A	0	X	X	X	1	1	0	1
	B	1	X	X	X	1	1	0	1
	C	0	X	X	X	1	1	1	0
	D	1	X	X	X	1	1	1	0
	E	0	X	X	X	1	1	1	1
General Register	F	1	X	X	X	1	1	1	1

Floating Register 0, 2, 4, 6, 8, A, C, E

Function	0	0	X	X	X	0	0	0	0
	1	1	X	X	X	0	0	0	0
	2	0	X	X	X	0	0	0	1
	3	1	X	X	X	0	0	0	1
	4	0	X	X	X	0	0	1	0
	5	1	X	X	X	0	0	1	0
	6	0	X	X	X	0	0	1	1
	7	1	X	X	X	0	0	1	1
	8	0	X	X	X	0	1	0	0
	9	1	X	X	X	0	1	0	0
	A	0	X	X	X	0	1	0	1
	B	1	X	X	X	0	1	0	1
	C	0	X	X	X	0	1	1	0
	D	1	X	X	X	0	1	1	0
	E	0	X	X	X	0	1	1	1
Function	F	1	X	X	X	0	1	1	1

Function 0: Console Interrupt
Function 1: Modify PSW Bits
Function 4: PSW
Function 5: LOC

COMMAND

Normal	1	0	0	0	0	0	0	0
Incremental	0	1	0	0	0	0	0	0

5.2 Wait State

The running program can place the Processor into the Wait state by setting the Wait bit of the current PSW. The WAIT indicator on the lower right of the panel illuminates to inform the operator of the Wait state. The Processor can leave the Wait state and resume execution in two ways:

1. An Interrupt can occur causing the Processor to jump to an interrupt service routine. When the routine restores the original PSW, the Wait state is re-established.
2. The operator can depress the RUN key which causes the Wait bit in the PSW and the WAIT lamp to be reset and execution to resume at the address specified by LOC.

6. PROGRAMMING SEQUENCES

The Hexadecimal Display has a device address of X'01'.

6.1 This device can be used to output up to five bytes of data to the Console Panel Indicators. The following program sequence outputs four bytes of data starting from the memory location BUF:

```
LIS     R1, 1       R1 = Display Adv.
LHI     R3, X'40'   Display in Increment Mode
OCR     R1, R3
WD      R1, BUF
WD      R1, BUF+1
WD      R1, BUF+2
WD      R1, BUF+3
```

At this time the Console Panel Indicators are on as shown below:

D5	D4	D3	D2	D1
	(BUF+3	(BUF + 2)	(BUF +1)	(BUF)

6.2 In order to light indicators D1 and D2, the Console can be in the normal mode and one halfword can be output. The following programming sequence can be used:

```
LIS     R1, 1       Console in Normal Mode
LHI     R3, X'80'
OCR     R1, R3
WH      R1, BUF
```

The Console Panel Indicators will be on as shown below:

D5	D4	D3	D2	D1
			BUF+1	BUF

Note that when a halfword of data is output to the Console Panel, the most significant byte loads in indicator D1 and the least significant byte loads in D2.

6.3 The Console Panel Switch Register can be read by using the read instructions as shown below:

```
LIS     R1, 1       R1 = Console Address
LHI     R3, X'80'   R3 = 80 = Normal Mode
OCR     R1, R3
RHR     R1, R4      Read 1 Halfword
EXBR    R4, R4      Exchange Bytes
```

At this time Register 4 has the 16 data switches.

Programming Note:

If more than five bytes are output to the Display Panel, the data is lost after five bytes. The Console must then be initialized by giving an output command to it before outputting any data.

02-348R01A22
September 1974

MEMORY ACCESS CONTROLLER PROGRAMMING SPECIFICATION

1. INTRODUCTION

The 02-348 Memory Access Controller (MAC) is an auxiliary module available with the INTERDATA Series 32 Processors.

The function of the Memory Access Controller is two-fold. It provides memory relocation and protection and the interface logic for the Multiplexed DMA Bus, Local Memory and the CPU.

The throughput between the CPU and local memory or between the selector channel and local memory is not affected by use of the Memory Access Controller.

In an operating system environment, the operation of the Memory Access Controller is completely transparent to most programs. It is very similar to a peripheral device in that only the operating system modules directly responsible for its operation need to be aware of its existence.

To enable (disable) the MAC directly from the Hexadecimal Display Panel, Bit 21 of the current program status word is set (reset) using the following procedure:

1. Depress the Data (DATA) key.
2. Enter the data (bits 16-31 for status word) from the Hexadecimal keyboard.
3. Depress the Function (FN) key.
4. Depress Hexadecimal key 1. The modified PSW is then displayed.

2. CONFIGURATION

Refer to Memory Access Controller Installation Specification, 02-348A20.

3. GENERAL DESCRIPTION

The Memory Access Controller operates on a twenty bit program (or virtual) address produced by the processor. It converts this address, through an addition process, into a real address, again twenty bits, in memory. At the same time, it checks the program address against a preset limit which the program address is not allowed to exceed. It verifies that the item referenced by the program address is actually in memory. It also checks the type of memory access, that is, instruction fetch, memory read, or memory write, against the allowable operations. If the program address exceeds the limit, or if the item is not in memory, or if the operation is not allowed, the Memory Access Controller notifies the processor by generating a Memory Access Controller Interrupt.

4. ADDRESS CONVENTION

Although the address coming into the Memory Access Controller is a 20 bit address in program space, and the relocated address produced by the controller is a 20 bit address in memory, internal to the controller, addresses are artificially broken down into two parts. These are the Block Address and the displacement within the block. An incoming address is shown in Figure 1.

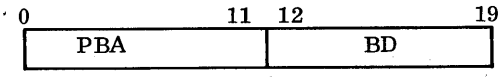


Figure 1. Program Address

PBA is the program block address, the address of a 256 byte block in program space. BD is the byte displacement within the block. Similarly, the real address is shown in Figure 2.

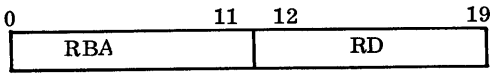


Figure 2. Program Address

RBA is the block address of a 256 byte block of real memory, and BD is the byte displacement within that block. The block address convention allows this Memory Access Controller to operate on 12 bit addresses. It also forces program segmentation to begin and end on 256 byte boundaries.

5. REGISTER ALLOCATIONS

The memory access controller contains seventeen hardware registers. They are: sixteen segmentation registers and one interrupt status register. If the memory access controller is disabled, these registers are accessed as if they were memory locations.

The segmentation registers are assigned locations X'00300' through X'0033F'. The interrupt status register is assigned to the fullword location X'00340'.

If the processor configuration has provision for more than 255 external devices, the block of memory locations assigned to the registers starts at the nearest multiple of X'00100' above the expanded interrupt service pointer table (i.e., X'00500' or X'00900').

Although only 68 bytes of address space (i.e., X'00300' through X'00343') are used as register addresses, the controller is set up to trap 256 bytes of address space (i.e., X'00300' through X'003FF'). If the processor references any location trapped, but not used by the controller, the results are undefined.

If the memory access controller is enabled and a memory reference is relocated to one of the locations reserved for the controller, the data is read from or written into the corresponding memory locations.

6. PROGRAMMING INSTRUCTIONS

Bit 21 of the current program status word controls the segmentation, relocation, and protection features of the memory access controller.

If bit 21 is set, the segmentation, relocation, protection and relocation/protection interrupts are enabled.

If bit 21 is reset, all memory references are absolute, and all protection is disabled.

When disabled the memory access controller is still active in that it traps all references to memory locations assigned to its register. Refer to Section 5 for register allocations.

Bit 21 must be reset to reference the memory access controller registers.

Memory reference instructions may be used to load data into the segmentation registers and to read the least significant byte of the status register (i.e.,, ST, STH, STB, L, LH, LB).

7. REGISTER SELECTION

Bit 11 of the current program status word controls the method of segmentation register selection.

If Bit 11 is reset, the high order four bits (bits 0:3) of the 20 bit program address are used to select the segmentation register (0:15). Refer to Figure 3.

If Bit 11 is set, the processor is in the halfword mode. Refer to Appendix 2 for a description of the memory access controllers halfword mode operation.

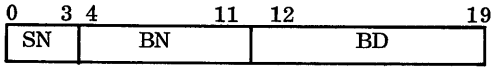


Figure 3. Program Address

SN = The segment number
BN = The block number within the segment
BD = The byte displacement within the block.

8. SEGMENTATION REGISTERS

Each of the 16 segmentation registers is 32 bits wide, and is divided into three fields as shown in Figure 4.

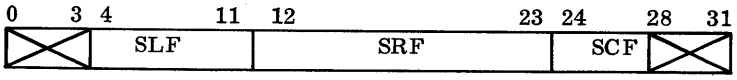


Figure 4. Segmentation Register

SLF = The segment limit field
SRF = The segment relocation field
SCF = The segment control field
Bits 0:3 = are not used and must be zero
Bits 28:31 = are not used and must be zero

8.1 Segment Relocation Field

The relocation field contains the block address of the first 256 byte blocks of memory controlled by the register.

Program addresses are relocated by adding the block number taken from the program address to the value contained in the relocation field. The byte displacement from the program address is then appended to the result to produce the 20 bit address in memory. Refer to Figure 5.

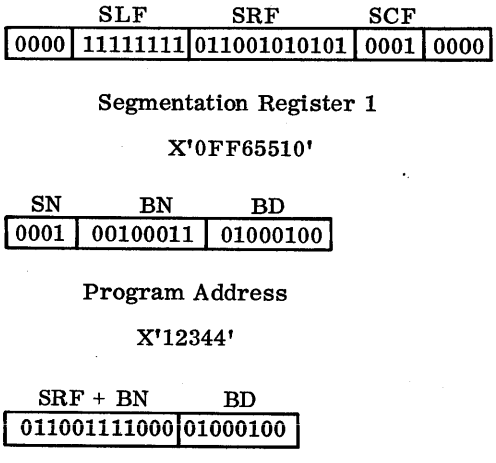


Figure 5. Segment Relocation Fields

8.2 Segment Limit Field

The limit field contains the maximum block number allowed for this segment (although segments have a fixed maximum length of 64KB, the actual segment size may be any multiple of 256 bytes up to the limit).

The block number is compared to the value contained in the segment limit field.

If the comparison indicates that the block number is less than or equal to the limit, the relocated address is valid.

If the comparison indicates that the block number is greater than the limit, the relocated address is invalid.

8.3 Segment Control Field

The control field determines the type of protection to be provided by the memory access controller.

The control field is divided into three parts as shown in Figure 6.

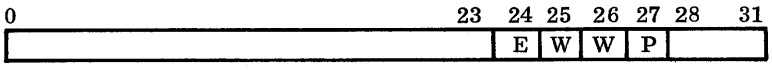


Figure 6. Segment Control Field

E = Execution Protect (Bit 24)
W = Write Protect (Bits 25, 26)
P = Present Bit (Bit 27)

8.3.1 Execution Protect. If Bit 24 is set, the area of memory described by the segmentation register is protected against execution (instruction fetches from the area are not allowed).

If Bit 24 is reset, execution of instructions from the area of memory described by the segmentation register is allowed.

8.3.2 Write Protect. Bits 25 and 26 are encoded to provide two types of write protection. The interpretation of these bits is:

BIT	25	26	
	0	0	= UNPROTECTED
	0	1	= WRITE PROTECTED
	1	0	= WRITE/INTERRUPT PROTECTED
	1	1	= WRITE PROTECTED (same as 01)

If the area controlled by the segmentation register is unprotected, all writes are allowed.

If the area controlled by the segmentation register is write protected, all write operations are aborted.

If the area controlled by the segmentation register is write/interrupt protected, the memory access controller allows the write operation to proceed, and then interrupts the processor.

8.3.3 Present Bit. If bit 27 is set, it means that the area of memory described by the segmentation register has been loaded with the correct program segment.

If Bit 27 is reset, it means that the program segment does not exist or has not yet been loaded.

9. INTERRUPTS

The Memory Access Controller generates interrupts for the following reasons:

Invalid Address
Non-Present Address
Write Protect Violation
Write/Interrupt Condition
Execute Protect Violation

It can generate interrupts to the processor only when relocation and protection is enabled (Bit 21 of the current program status word is set).

If the processor cannot service an interrupt immediately, the controller continues to operate, however, it does not allow memory to be changed. All writes are converted to reads until the processor clears the interrupt.

9.1 Invalid Address Interrupt

This occurs when the program block number is greater than the value in the limit field.

9.2 Non-Present Address Interrupt

This occurs when the program accesses a segmentation register in which the present bit (27) is zero.

9.3 Write Protect Violation Interrupt

This occurs when the program attempts to write into an area controlled by a segmentation register in which the write protect bit (26) is set.

9.4 Write/Interrupt Condition Interrupt

This occurs when the program writes into an area controlled by a segmentation register in which the write/interrupt code appears (Bit 25 = 1 and Bit 26 = 0).

9.5 Execute Protect Violation Interrupt

This occurs when the program attempts to execute an instruction from an area controlled by a segmentation register in which the execute protect bit (24) is set.

10. INTERRUPT STATUS REGISTER

When the memory access controller generates an interrupt, it sets a bit in the interrupt status register. The interrupt status register is 32 bits wide. Bits 0:26 are undefined. The significance of the remaining bits is:

Bit	Meaning
27	Invalid Address Interrupt
28	Non-Present Address Interrupt
29	Write Protect Violation Interrupt
30	Write/Interrupt Condition Interrupt
31	Execute Protect Violation Interrupt

The first reference, either read or write, to the interrupt status register clears the interrupt condition within the controller, however, only a write instruction will clear the interrupt status register.

11. INTERRUPT HANDLER

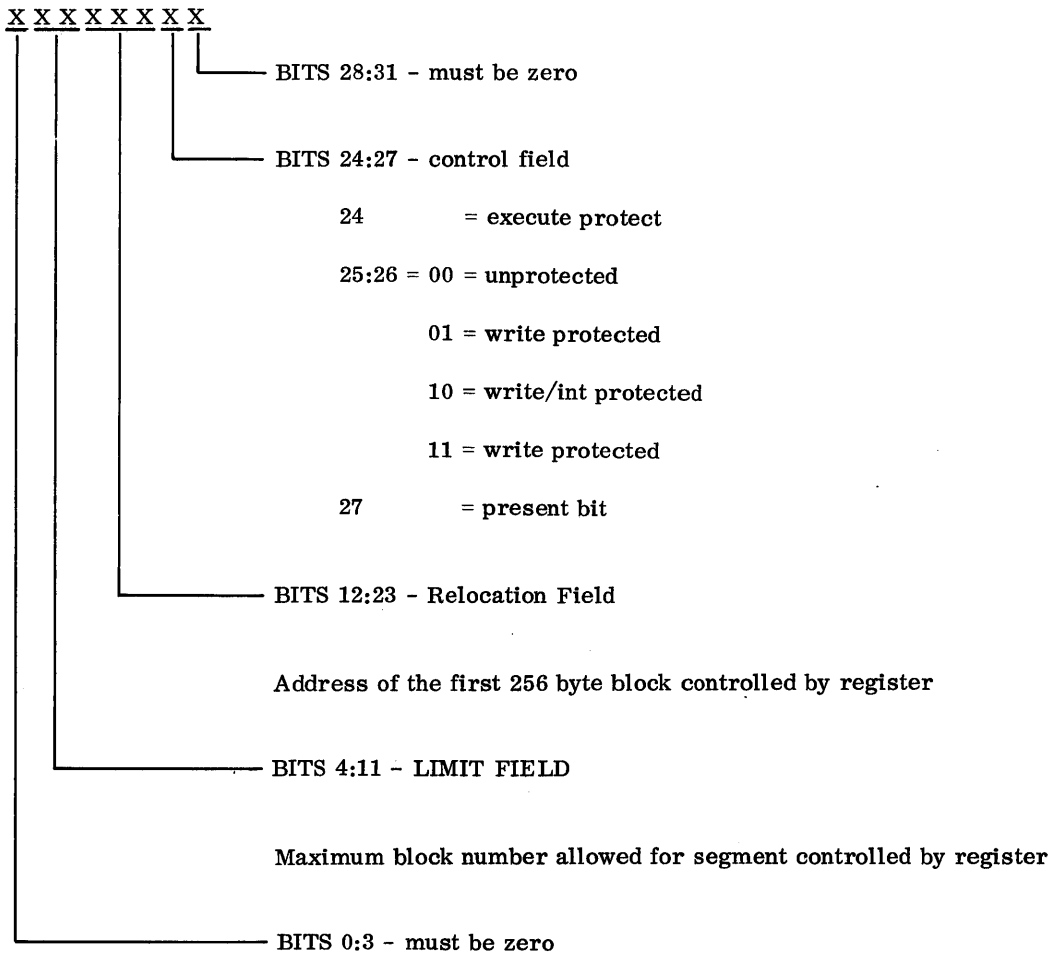
The new program status word for the memory access controller interrupt handler should disable relocation and protection (bit 21 of the new program status word is reset). The handler can then determine the cause of the interrupt by testing the memory location assigned to the interrupt status register.

12. INITIALIZATION

Whenever the Initialize Switch (INT) on the display panel is depressed, or the processor is powered up, all segmentation, relocation, protection and relocation/protection interrupts are disabled regardless of the state of Bit 21 in the current program status word. The memory access controller will remain disabled until a memory reference instruction is issued to the area trapped by the memory access controller. At this time, the memory access controller will be enabled or disabled depending on the state of Bit 21 of the current program status word.

APPENDIX 1

SEGMENTATION REGISTER VALUE



APPENDIX 2

MEMORY ACCESS CONTROLLER

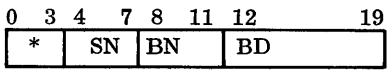
HALFWORD MODE OPERATION

FOR MODEL 7/32

The halfword mode operation of the Memory Access Controller is identical to the fullword mode operation with the exception of segmentation register selection and the operation of the segment limit field.

Segmentation Register Selection

Bits 4:7 of the 20 bit program address are used to select the segmentation register (0:15).



Halfword Mode Program Address

SN = the Segment Number
BN = the block number within the segment
BD = the byte displacement within the block
* In this mode bits 0:3 of the program address must be zero.

SEGMENT LIMIT FIELD

The Limit Field contains the maximum block number allowed for this segment (although segments have a fixed maximum length of 4KB, the actual segment size may be any multiple of 256 bytes up to the limit).

The block number is compared to the value contained in the segment limit field.

If the comparison indicates that the block number is less than or equal to the limit, the relocated address is valid.

If the comparison indicates that the block number is greater than the limit, the relocated address is invalid.

02-328R01A22
September, 1974

EXTENDED SELECTOR CHANNEL PROGRAMMING SPECIFICATION

1. INTRODUCTION

The 02-328 Extended Selector Channel (ESELCH) is upward program compatible with the current 16 bit address Selector Channel. It controls the transfer of data between I/O devices and local or extended memories at rates of up to 2,000,000 bytes per second. Up to 16 I/O devices can be connected to the Extended Selector Channel, but only one device can transfer data at a time. Data transfer to the device may be either Byte or Halfword oriented.

2. CONFIGURATION

Refer to the Extended Selector Channel Installation Specification 02-328A20.

3. PROGRAMMING INSTRUCTIONS

A sense status instruction (SS or SSR) is used to transfer the status byte from the extended selector channel device controller to the processor. Refer to Table 1 for status and command byte. This instruction should not be used under interrupt control as it could cause the selector channel to become idle and reset the interrupt condition.

The output command instruction (OC or OCR) causes a command byte to be sent to the extended selector channel controller.

The Write Data (WD or WDR) or Write Halfword (WH or WHR) instructions may be used to send the starting and final addresses to the ESELCH.

The Read Data (RD or RDR) or Read Halfword (RH or RHR) instructions may be used to obtain the last processor memory location either written into or read from memory.

The Write Block (WB or WBR) instruction or Read Block (RB or RBR) instruction should <u>not</u> be used since the status byte returned to an idle SELCH, by these instructions, is the status of any active device on the SELCH bus with the busy bit forced to a zero.

In the halfword mode on Model 7/32 (PSW bit 11 = 1), an Acknowledge Interrupt (AI or AIR) instruction clears a pending interrupt and causes the device number of the ESELCH and the status of the peripheral device to be sent to the processor.

TABLE 1. EXTENDED SELECTOR CHANNEL STATUS AND COMMAND BYTE DATA

BIT NUMBER	8	9	10	11	12	13	14	15
STATUS BYTE			MEMORY MALFTN.	MEMORY PARITY FAIL	BUSY			
COMMAND BYTE		EXTENDED ADDRESS READ	READ	GO	STOP	SELCH STATUS		

STATUS

BUSY — This bit is set when the extended selector channel is in the process of transferring data.

Memory Parity Fail — This bit is set when the memory interface recognizes a parity failure. This bit is stored in the extended selector channel for subsequent evaluation by the processor, however, the transfer is not interrupted. This bit is reset by initialization of the processor or by an output command Go.

Memory Malfunction — This bit is set when the memory interface recognizes a malfunction other than data parity failure. This bit is stored in the extended selector channel for subsequent evaluation by the processor. However, the transfer is not interrupted. This bit is reset by initialization of the processor by an output command Go.

COMMAND

Extended Address Read — This command specifies whether a 2-byte or 3-byte final address is returned to the processor.

When this bit is set, a 3-byte final address is returned.

When this bit is reset, a 2-byte final address is returned.

This command can be issued at the same time the Read Mode is established.

Read — This command changes the mode of the extended selector channel from write to read.

In the read mode, data is transmitted from the active device on the extended selector channel and written into memory.

Whenever a data transmission has been completed, the extended selector channel is placed in the write mode. Each time a read operation is required, a read command must be issued.

GO — This command initiates a data transmission. It can be issued at the same time the read/write mode is established.

STOP — This command halts any data transmission in progress, and initializes the ESELCH for starting a new operation. It should be given when the extended selector channel terminates.

SELCH Status — When this bit is set, the ESELCH status will be returned at all times on SSR or SS instructions to the ESELCH.

When this bit is reset, the ESELCH is busy, only the busy bit is present in the status byte and all other bits are zero.

When this bit is reset, and the ESELCH is not busy, the status of the device is presented in the status byte, except for the busy bit which is forced to zero.

4. PROGRAMMING SEQUENCES

Programming a device on the selector channel consists of setting up the device, setting up the selector channel, and sending a GO command to the selector channel. Refer to Table 1 and Section 3 for a description of the selector channel status and command byte.

The setting up or the initialization of a device on the selector channel bus is accomplished by executing an Output command (OC or OCR) instruction. Refer to the appropriate device programming manual for a description of the command byte. No I/O instruction can be issued to any device on the selector channel when the selector channel is busy (not idle). Only Output commands with the STOP, EXTENDED ADDRESS READ, or SELCH STATUS BIT set can be issued to the selector channel when it is busy. Note that the selector channel has a unique device number just like all other I/O devices. Output commands, as with all Input/Output instructions, affect only the device addressed. If a device on the selector channel bus is referenced while the selector channel is busy, the False Sync. bit is set (V condition code) indicating that the command was ignored.

The selector channel is idle only after initialization (refer to Section 9) or after an I/O instruction is issued to the selector channel (e.g. WH, RD, OC, SSR,...) when it is not busy. Therefore, prior to issuing any I/O instructions to the selector channel or to any device on the selector channel bus, the program must either:

1. Wait until the selector channel becomes not busy (i.e. completes any I/O transfer), or
2. Issue an Output command to the selector channel with the STOP bit in the command byte set.

In order to perform a Read or Write operation on a device using a selector channel, these four steps are required in the following order:

1. Wait for the selector channel to become not busy, if it is transferring data, then give an Output command to the selector channel with the STOP bit set in the command byte.

2. Set up the device on the selector channel by giving the appropriate I/O commands to the device.

3. Send the starting and ending address to the selector channel (refer to Section 5).

4. Send an Output command to the selector channel with the GO bit set in the command byte. The READ bit should also be set for a Read operation. The READ bit should be reset for a Write operation.

PROGRAMMING NOTES

No I/O instruction can be issued to any other device during the execution of the program instructions in Steps (2) through (4) above. When the GO command is issued, the Selector Channel does the following:

1. Controls the last device addressed by the processor (i.e. by the user program or by the microprogram) if that device is connected to the Selector Channel.

2. Hangs if the last device addressed by the processor is not attached to the Selector Channel. This condition must be cleared by issuing a command with the STOP bit set to the Selector Channel.

For this reason, the program instructions in Steps (2) through (4) above must be executed with external interrupts disabled (PSW bit 17 = 0). Otherwise, it is possible to detect an external interrupt and the interrupt driver in the user program (or the microprogram in the case of an immediate interrupt) can address any other device. Also the single mode may not be used since the display panel is addressed in this mode.

The Selector Channel has a 20-bit incrementing Address Register and a 20-bit Final Address Register. The user program loads the starting address into the incrementing Address Register and the final address into the Final Address Register. Transfer is completed when the incrementing Address register matches the Final Address Register. The address limits are expressed inclusively; transfers begin and end on the addresses placed in the Starting and Final Address Registers.

The memory is addressed on halfword boundaries; that is, each time memory is accessed, two bytes or a halfword are obtained. 20-bits addressing is used in memory system, with the least significant bit, Bit 19, determining the byte desired. Refer to Figure 1.

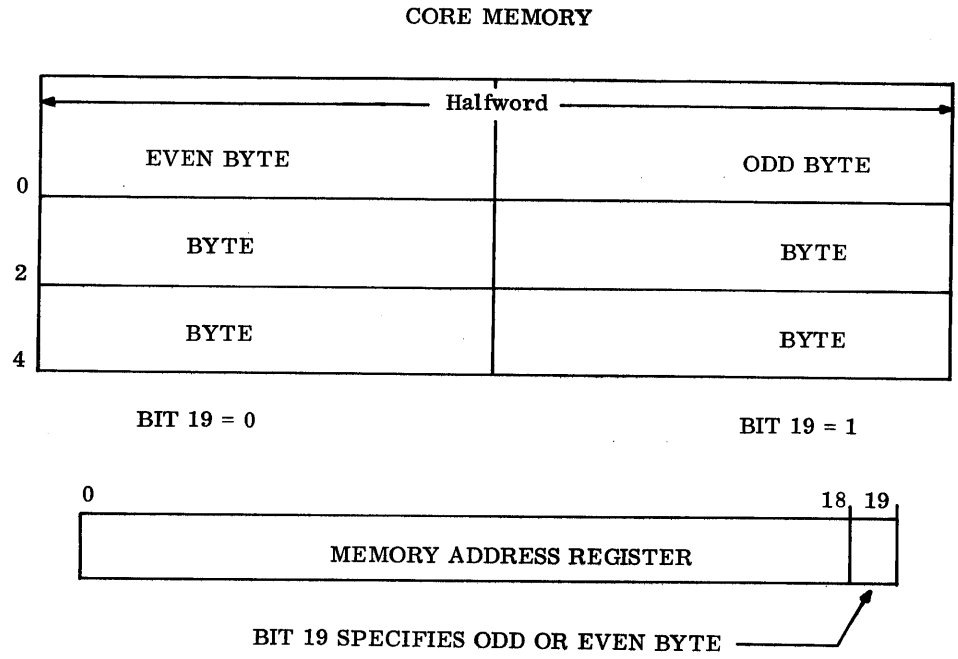


Figure 1. Memory Addressing

Each time the Selector Channel accesses memory, two bytes (halfword) are transferred. It is mandatory that data transfers begin on a halfword boundary.

The following results if data transfers are ended on the byte boundaries:

1. Write Mode (Memory to device) - end on byte boundary (bit 19 = 0) no effect.

2. Read Mode (device to memory) - end of byte boundary (BIT 19 = 0) The previous contents of the last odd byte in memory is written into the current odd byte in memory. Refer to Figure 2.

Data transfers across a 256K boundary may be executed provided the memory is contiguous.

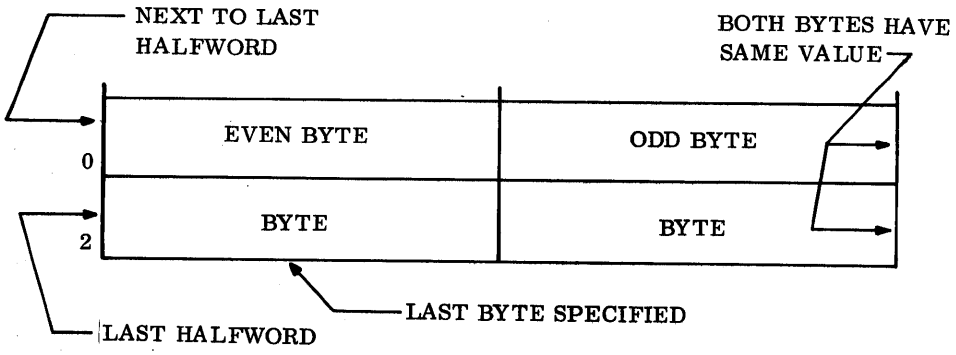
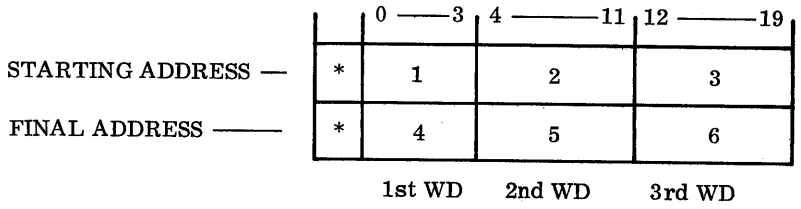


Figure 2. Core Memory Configuration, End On Byte Boundary

5. TRANSMISSION OF STARTING AND FINAL ADDRESSES

An Output Command with the STOP bit set should be issued prior to starting any operation on the Selector Channel to clear any preceding conditions. Normally, six successive bytes are required to specify the starting and final addresses of the user's buffer area. If the final address of the buffer area is no greater than 65536, either four or six (with most significant byte zero) successive bytes may be used to specify the starting and final addresses. Figure 3 illustrates the meaning of six bytes used for addressing.



* Must be zero

1. STARTING ADDRESS (BITS 0-3)
2. STARTING ADDRESS (BITS 4-11)
3. STARTING ADDRESS (BITS 12-19)
4. FINAL ADDRESS (BITS 0-3)
5. FINAL ADDRESS (BITS 4-11)
6. FINAL ADDRESS (BITS 12-19)

Figure 3. Meaning Of The Data Bytes When Setting Start And Final Addresses

Either the Write Data (WD or WDR) or Write Halfword (WH or WHR) instructions may be used to send the starting and final addresses to the Selector Channel Controller.

For example, if the starting address is X'000000' and the ending address is X')ABCDE', the following addressing sequences are all correct:

A.
```
WD (or WDR)  SELCH, LARX (00)
WD (or WDR)  SELCH, LARH (00)
WD (or WDR)  SELCH, LARL (00)
WD (or WDR)  SELCH, LFAX (0A)
WD (or WDR)  SELCH, LFAH (BC)
WD (or WDR)  SELCH, LFAL (DE)
```

B.
```
WD (or WDR)  SELCH, LARX (00)
WH (or WHR)  SELCH, LARHL (0000)
WD (or WDR)  SELCH, LFAX (0A)
WH (or WHR)  SELCH, LFAHL (BCDE)
```

C.
```
WH (or WHR)  SELCH, LARXH (0000)
WD (or WDR)  SELCH, LARL (00)
WH (or WHR)  SELCH, LRAXH (0ABC)
WD (or WDR)  SELCH, LFAL (DE)
```

If the starting address is X'001234' and the ending address is X'00ABCD', the following addressing sequences are all correct:

A.
```
WD (or WDR)  SELCH, LARH (12)
WD (or WDR)  SELCH, LARL (34)
WD (or WDR)  SELCH, LFAH (AB)
WD (or WDR)  SELCH, LFAL (CD)
```

B.
```
WH (or WHR)  SELCH, LARHL (1234)
WH (or WHR)  SELCH, LAFAHL (ABCD)
```

C.
```
WD (or WDR)  SELCH, LARX (00)
WD (or WDR)  SELCH, LARH (12)
WD (or WDR)  SELCH, LARL (34)
WD (or WDR)  SELCH, LFAX (00)
WD (or WDR)  SELCH, LFAH (AB)
WD (or WDR)  SELCH, LFAL (CD)
```

D.
```
WD (or WDR)  SELCH, LARX (00)
WH (or WHR)  SELCH, LARHL (1234)
WD (or WDR)  SELCH, LFAX (00)
WH (or WHR)  SELCH, LFAHL (ABCD)
```

E.
```
WH (or WHR)  SELCH, LARXH (0012)
WD (or WDR)  SELCH, LARL (34)
WH (or WHR)  SELCH, LFAXH (00AB)
WD (or WDR)  SELCH, LFAL (CD)
```

6. TERMINATION

Data transmission between the Selector Channel and the device presently connected to it is halted if any of the following conditions occur:

1. The starting address matches the final address. This denotes a normal termination.

2. The starting (incrementing) address goes from all Ones to all Zeros (maximum count). In this case, a match has not occurred and is considered as abnormal termination.

3. Any of DU, EOM, or EX status bits of the device presently connected to the Selector Channel changes to a One. This is also an abnormal termination.

4. A Stop command is sent to the Selector Channel via a user program. The termination condition is determined in one of two ways: by a status scan, or by the interrupt method. An Output Command Stop should be issued to the Selector Channel following its termination.

5. Address a non-existent memory. This is an abnormal termination.

Status Scan

The status of the Selector Channel Controller may be examined by issuing a Sense Status (SS or SSR) instruction. The Busy Bit (Bit 12) is a One while transmission is in progress, and it is a Zero when transmission is terminated. One method of testing for termination would be to continually or periodically test the Busy Bit of the Selector Channel Controller. The change from One to Zero would then indicate the termination of a data transfer. In the status scan method of programming, it is possible for the Busy Bit to change from One to Zero during a Sense Status instruction without returning the SELCH to Idle. To guarantee the Idle Mode after Busy = 0 on a Sense Status instruction, a Stop Command should be sent to the SELCH.

Interrupt Method

When data transmission is initiated on the Selector Channel, the interrupt is always enabled. If external device interrupts are enabled, the Processor is interrupted when the Selector Channel terminates. In the Halfword mode, an interrupt can be serviced via Immediate Interrupt or Acknowledge Interrupt instruction (AI, AIR), which clears the interrupt. When this occurs, the device number of the Selector Channel (X'0F0' preferred) and status of the peripheral device are sent to the Processor if SELCH Status Command bit is reset. Status of the Selector Channel is sent to the processor if SELCH Status Command is set. In the Fullword mode, the interrupt causes program control to transfer to the start address of a user's termination service routine via a pointer located at an Interrupt Service Pointer table address by the formula: 2X Device Address + X'D0'.

7. READING THE FINAL ADDRESS

The last processor memory location either written into or read from may be obtained by executing a pair of Read Data (RD or RDR) instructions or a Read Halfword (RH or RHR) instruction provided the final address is no greater than 65536. Three successive Read Data (RD or RDR) instructions or one Read Data and one Read Halfword instruction are required if the final address is greater than 65536.

The command bit, extended address read (Bit 9) specifies whether a 2 byte or 3 byte final address is being used. Before issuing RD or RH instruction, a Command Stop should be issued to insure that the SELCH is initialized.

This information permits a user program to verify a successful data transmission or determine at what address termination occurred.

Table 2 illustrates various sequences in which the final address can be read.

8. DEVICE NUMBER

The Selector Channel is normally assigned device number X'0F0' (10 bit address), but may easily be changed by a minor wiring modification on the Selector Channel device Controller Board. Refer to the Installation Specification for specific details (02-328A20).

9. INITIALIZATION

Whenever the Initialize switch (INT) on the Display Panel is depressed, or a Stop Command is issued, the following action occurs:

1. Any data transmission in process is halted and the Stop Mode is affected.
2. The Selector Channel is placed in the Write Mode.
3. The Selector Channel is made idle.
4. The Selector Channel interrupt is reset.

TABLE 2. SEQUENCES USED TO READ FINAL ADDRESS

1. Final Address is X'00 ABCD'

Table 2 - 1

Instruction Sequence	Answers	
	Bit 9 Set	Bit 9 Reset
RD (or RDR) RD (or RDR)	00 AB	AB CD
RH (or RHR)	00AB	ABCD
RD (or RDR) RD (or RDR) RD (or RDR)	00 AB CD	AB CD Undefined
RD (or RDR) RH (or RHR)	00 ABCD	AB CDXX
RH (or RHR) RD (or RDR)	00AB CD	ABCD Undefined

2. Final Address is X'0ABCDE'

Table 2 - 2

RD (or RDR) RD (or RDR)	0A BC	BC DE
RH (or RHR)	0ABC	BCDE
RD (or RDR) RD (or RDR) RD (or RDR)	0A BC DE	BC DE Undefined
RD (or RDR) RH (or RHR)	0A BCDE	BC DEXX
RH (or RHR) RD (or RDR)	0ABC DE	BCDE Undefined

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

Error (Page No. ___, Drawing No. _____)
Addition (Page No. ___, Drawing No. _____)
Other (Page No. ___, Drawing No. _____)

Explanation:

CUT ALONG LINE

FOLD FOLD

Fold and Staple
No postage necessary if mailed in U. S. A.

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT No. 22
OCEANPORT, N.J.

BUSINESS REPLY MAIL

NO POSTAGE NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY:

INTERDATA®

2 Crescent Place, Oceanport, New Jersey 07757

TECH PUBLICATIONS DEPT. MS 53

FOLD

FOLD

STAPLE

STAPLE