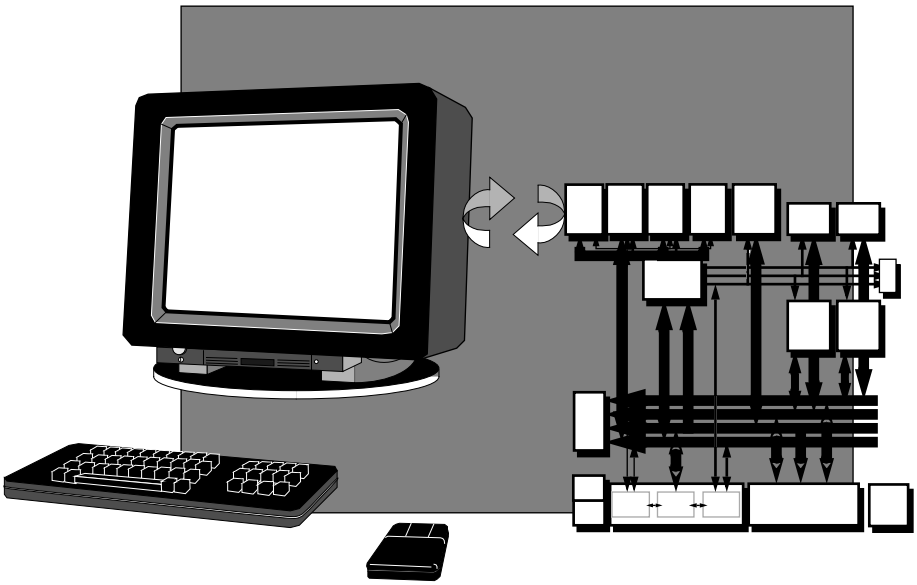
SECTION 5

HOST INTERFACE



SECTION CONTENTS

5.1 INTRODUCTION

The Host Interface (HI) is a byte-wide parallel slave port which may be connected directly to the data bus of a host processor. The host processor may be any of a number of popular microcomputers or microprocessors, another DSP or DMA hardware. The DSP56156 has an 8-bit bidirectional data bus H0-H7 (PB0-PB7) and 7 control lines HR/$\overline{W}$, $\overline{HEN}$, $\overline{HREQ}$, HA0-HA2, and $\overline{HACK}$ (PB8-PB14) to control data transfers. The HI pin functions are described in Section 2. The HI appears as a memory mapped peripheral, occupying 8 bytes in the host processor's address space and three words in the DSP processor's address space. Figure 5-1 shows the HI block diagram. Separate transmit and receive data registers are double-buffered to allow the DSP56156 and host processor to efficiently transfer data at high speed. Host processor communication with the HI registers is accomplished using standard host processor instructions and addressing modes. Host processors may use byte move instructions to communicate with the HI registers. The host registers are addressed so that 8-bit MC6801-type host processors can use 16-bit load (LDD) and store (STD) instructions for data transfers. The 16-bit MC68000/10 host processor can address the HI using the special MOVEP instruction for word (16-bit) or long word (32-bit) transfers. The 32-bit MC68020 host processor can use its dynamic bus sizing feature to address the HI using standard MOVE word (16-bit), or long word (32-bit) instructions.

Handshake flags are provided for polled or interrupt-driven data transfers. The DSP56156 interrupt response is sufficiently fast that most host microprocessors can load or store data at their maximum programmed I/O (non-DMA) instruction rate without testing the handshake flags for each transfer. If the full handshake is not needed, the host processor can treat the DSP56156 as fast memory and data can be transferred between the host and DSP56156 at the fastest host processor rate. DMA hardware may be used with the external Host Request and Host Acknowledge pins to transfer data at the maximum DSP56156 interrupt rate.

The host processor can also issue vectored exception requests to the DSP56156 with the host command feature. The host may select any of the 32 DSP exception routines to be executed by writing a vector address register. This flexibility allows the host programmer to execute a wide number of preprogrammed functions inside the DSP56156. Host exceptions can allow the host processor to read or write DSP56156 registers, Data memory or Program memory locations and perform control and debugging operations if exception routines are implemented in the DSP to do these tasks.

The DSP56156 CPU views the HI as a memory mapped peripheral occupying three 16-bit words in data memory space. The DSP56156 may access the HI as a normal memory-mapped peripheral using standard polled or interrupt programming techniques.

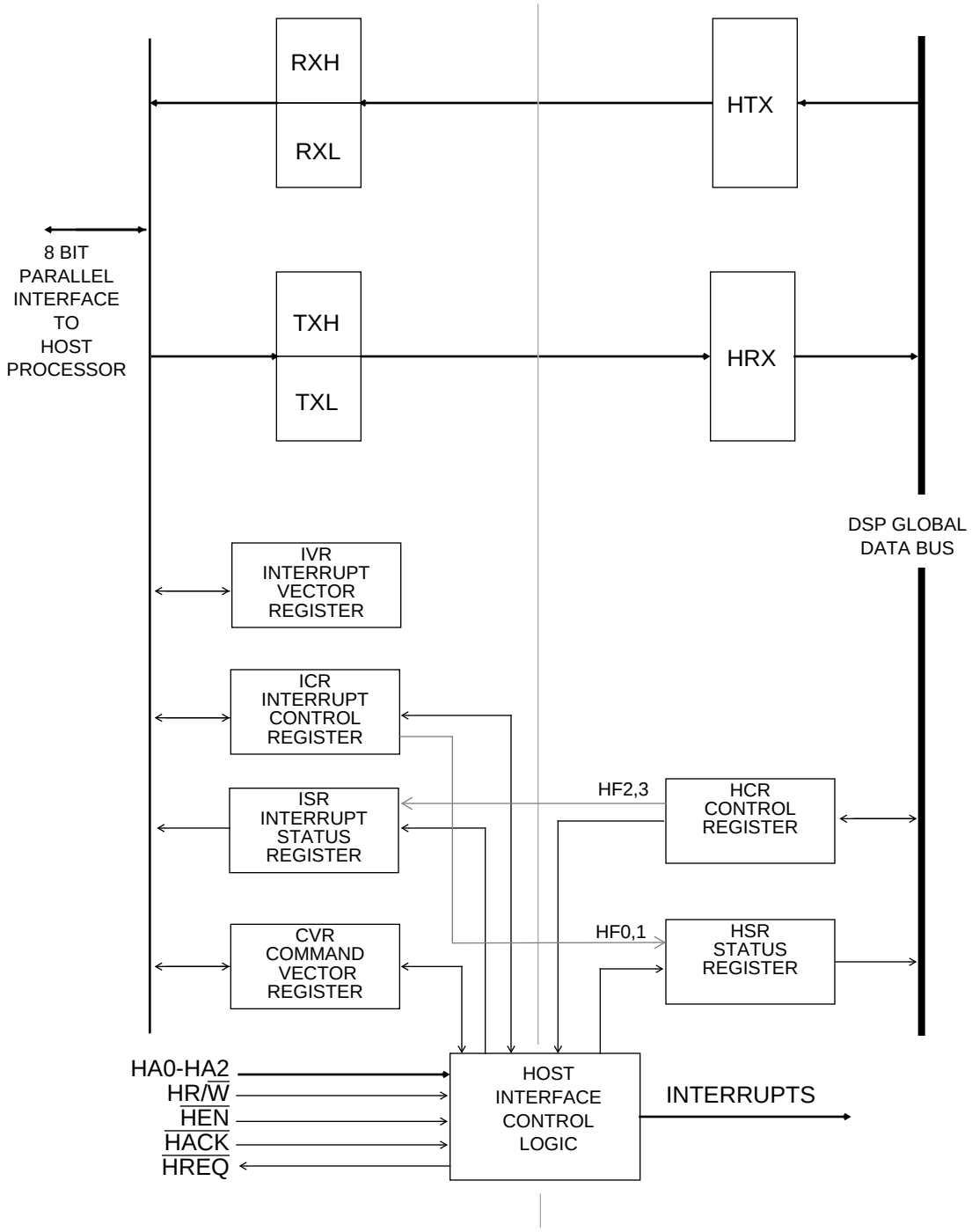


Figure 5-1 Host Interface Block Diagram

5.2 HOST INTERFACE PROGRAMMING MODEL

The HI has two programming models - one for the DSP56156 programmer and one for the host processor programmer. In most cases, the notation used in this manual reflects the DSP56156 perspective. The Host Interface - DSP56156 Programming Model is shown in Figure 5-2. The programming model register names on the DSP CPU side of the HI begin with the letter “H”. The Host Interface - Host Processor Programming Model is shown in Figure 5-3. The HI Interrupt Structure is shown in Table 5-2.

5.3 HOST TRANSMIT DATA REGISTER (HTX)

The Host Transmit register (HTX) is used for DSP to host processor data transfers. The HTX register is viewed as a 16-bit write-only register by the DSP. Writing the HTX register clears HTDE. The DSP may program the HTIE bit to cause a Host Transmit Data interrupt when HTDE is set. The HTX register is transferred as 16-bit data to the Receive Byte Registers RXH:RXL if both the HTDE bit and the Receive Data Full, RXDF, status bit are cleared. This transfer operation sets RXDF and HTDE.

5.4 RECEIVE BYTE REGISTERS (RXH, RXL)

The Receive Byte Registers are viewed as two 8-bit read-only registers by the host processor called Receive High (RXH) and Receive Low (RXL). These two registers receive data from the high byte and low byte respectively of the Host Transmit Data register HTX and are selected by three external Host Address inputs HA2, HA1 and HA0 during a host processor read operation or by an on-chip address counter in DMA operations. The Receive Byte Registers (at least RXL) contain valid data when the Receive Data Register Full RXDF bit is set. The host processor may program the RREQ bit to assert the external Host Request $\overline{\text{HREQ}}$ pin when RXDF is set. This informs the host processor or DMA controller that the Receive Byte Registers are full. These registers may be read in any order to transfer 8- or 16-bit data. However, reading the Receive Low register RXL clears the Receive Data Full RXDF bit. Because reading RXL clears the RXDF status bit, it is normally the last register read during a 16-bit data transfer.

5.5 TRANSMIT BYTE REGISTERS (TXH, TXL)

The Transmit Byte Registers are viewed as two 8-bit write-only registers by the host processor called Transmit High (TXH) and Transmit Low (TXL). These two registers send data to the high byte and low byte respectively of the Host Receive Data register (HRX) and are selected by three external Host Address inputs HA2, HA1 and HA0 during a host processor write operation. Data may be written into the Transmit Byte Registers when the Transmit Data Register Empty TXDE bit is set. The host processor may program the TREQ bit to assert the external Host Request $\overline{\text{HREQ}}$ pin when TXDE is set. This informs

the host processor or DMA controller that the Transmit Byte Registers are empty. These registers may be written in any order to transfer 8 or 16-bit data. However, writing the Transmit Low register TXL clears the TXDE bit. Because writing the TXL register clears the TXDE status bit, TXL is normally the last register written during a 16-bit data transfer. The Transmit Byte Registers TXH:TXL are transferred as 16-bit data to the Host Receive Data Register HRX when both TXDE bit and the Host Receive Data Full, HRDF, bit are cleared. This transfer operation sets TXDE and HRDF.

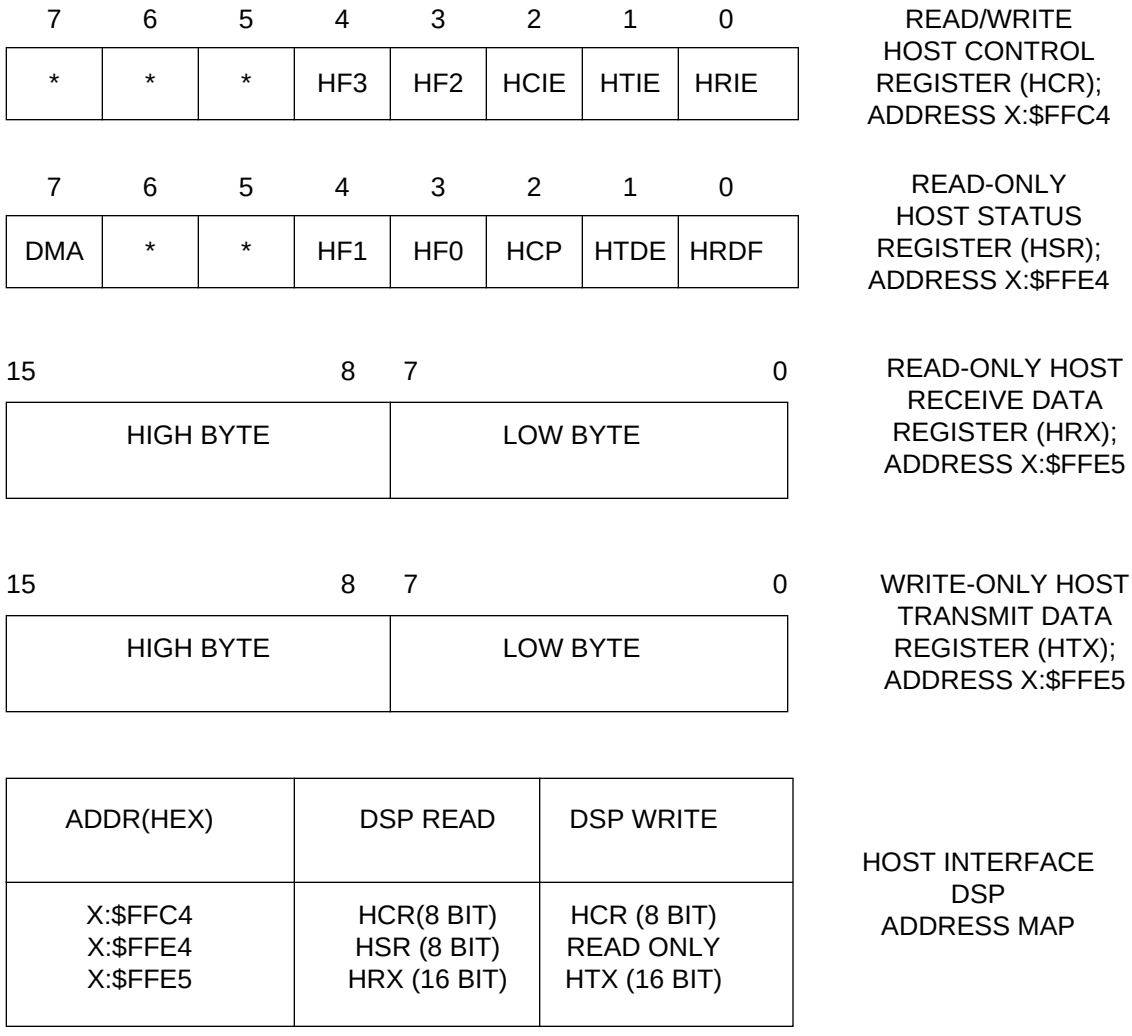


Figure 5-2 Host Interface - DSP Programming Model

5.6 HOST RECEIVE DATA REGISTER (HRX)

The Host Receive Data register (HRX) is used for host processor to DSP data transfers. The HRX register is viewed as a 16-bit read-only register by the DSP. The HRX register is loaded with 16-bit data from the Transmit Data Registers TXH: TXL when both the

Transmit Data Register Empty, TXDE, and Host Receive Data Full HRDF, bits are cleared. This transfer operation sets TXDE and HRDF. The HRX register contains valid data when the HRDF bit is set. Reading HRX clears HRDF. The DSP may program the HRIE bit to cause a Host Receive Data interrupt when HRDF is set.

5.7 COMMAND VECTOR REGISTER (CVR)

The Host Command Vector Register (CVR) is used by the host to request vectored exception service from the DSP of any exception routine in the DSP. The Host Command feature is independent of any of the data transfer mechanisms in the HI but can be used to initialize the DSP for data transfer by triggering the appropriate preprogrammed software routine.

7	6	5	4	3	2	1	0
HC	*	*	HV				

READ/WRITE
COMMAND VECTOR REGISTER (CVR); ADDRESS $1

5.7.1 CVR Host Vector (HV) Bits 0 through 4

The 5-bit Host Vector (HV) selects the host command exception address to access the host command exception routine. When the Host Command Exception is recognized by the DSP interrupt control logic, the starting address of the exception taken is 2*HV. This allows the host processor to provide the exception starting address for the Host Command Exception. The host processor can select any of the 32 possible exception routine starting addresses in the DSP by writing the exception routine starting address (divided by 2) into HV. This means that the host processor can force any of the existing exception handlers (SSI, TIMER, IRQA, IRQB, etc.) and can use any of the reserved or otherwise unused starting addresses provided they are pre-programmed in the DSP. HV is set to $16 (vector location $002C) by DSP reset.

CAUTION:

The HV should not be used with a value of zero because the reset location is normally programmed with a JMP instruction. This will cause an improper fast exception.

5.7.2 CVR Reserved bits – Bits 5 and 6

Reserved bits are unused and are read by the host as zeros. Reserved bits should be written as zero for future compatibility.

7	6	5	4	3	2	1	0	
INIT	HM1	HM0	HF1	HF0	*	TREQ	RREQ	READ/WRITE INTERRUPT CONTROL REGISTER (ICR); ADDRESS $0
HC	*	*	HV					READ/WRITE COMMAND VECTOR REGISTER (CVR); ADDRESS $1
HREQ	DMA	*	HF3	HF2	TRDY	TXDE	RXDF	READ-ONLY INTERRUPT STATUS REGISTER (ISR); ADDRESS $2
IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0	READ/WRITE INTERRUPT VECTOR REGISTER (IVR); ADDRESS $3

15 8	7 0	
RXH (ADDR $6) HIGH BYTE	RXL (ADDR $7) LOW BYTE	READ-ONLY RECEIVE BYTE REGISTERS; ADDRESSES $6,7
TXH (ADDR $6) HIGH BYTE	TXL (ADDR $7) LOW BYTE	WRITE-ONLY TRANSMIT BYTE REGISTERS; ADDRESSES $6,7

ADDR (BIN) HA2	HA1	HA0	HOST READ	HOST WRITE
0	0	0	ICR(8 BIT)	ICR (8 BIT)
0	0	1	CVR (8 BIT)	CVR (8 BIT)
0	1	0	ISR (8 BIT)	not used
0	1	1	IVR (8 BIT)	IVR (8 BIT)
1	0	0	read zeros	not used
1	0	1	reserved	reserved
1	1	0	RXH (8 BIT)	TXH (8 BIT)
1	1	1	RXL (8 BIT)	TXL (8 BIT)

HOST INTERFACE HOST PROCESSOR ADDRESS MAP

Figure 5-3 Host Interface - Host Processor Programming Model

Table 5-1 Host Interface Interrupt Structure

DSP INTERRUPT STRUCTURE

INTERRUPT SOURCE	STATUS	MASK	EXCEPTION STARTING ADDRESS
RECEIVE DATA FULL TRANSMIT DATA EMPTY HOST COMMAND	HRDF HTDE HCP	HRIE HTIE HCIE	$0028 $002A 2*HV($0000-003E)

HOST PROCESSOR $\overline{\text{HREQ}}$ STRUCTURE

$\overline{\text{HREQ}}$ SOURCE	STATUS	MASK
RECEIVE DATA FULL TRANSMIT DATA EMPTY	RXDF TXDE	RREQ TREQ

5.7.3 CVR Host Command Bit (HC) Bit 7

The Host Command bit (HC) is used by the host to handshake the execution of host command exceptions. Normally the host processor sets HC=1 to request the host command exception from the DSP. When the host command exception is taken by the DSP, the HC bit is cleared by the HI hardware. The host processor can read the state of HC to determine when execution of the host command has started. The host processor may elect to clear the HC bit, cancelling the Host Command Exception request at any time before it is recognized by the DSP.

CAUTION:

The command exception might be recognized by the DSP and executed before it can be canceled by the host, even if the host clears the HC bit.

Setting HC causes HCP (Host Command Pending) to be set in the HSR register. The host can write HC and HV in the same write cycle if desired. HC is cleared by DSP reset.

5.8 HOST CONTROL REGISTER (HCR)

The Host Control Register (HCR) is an 8-bit read/write control register used by the DSP to control the HI interrupts and flags. HCR cannot be accessed by the host processor. The HCR register occupies the low order byte of the internal data bus - the high order portion

is zero filled. HCR is a read/write register which can be accessed using bit manipulation instructions on control register bits. Any reserved bits are read as zeros and should be programmed as zeros for future compatibility. The contents of HCR are cleared on DSP reset. The control bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
*	*	*	HF3	HF2	HCIE	HTIE	HRIE

READ/WRITE
HOST CONTROL
REGISTER (HCR);
ADDRESS X:$FFC4

5.8.1 HCR Host Receive Interrupt Enable (HRIE) Bit 0

The Host Receive Interrupt Enable (HRIE) bit is used to enable a DSP interrupt when the Host Receive Data Full (HRDF) status bit in the Host Status register (HSR) is set. When HRIE is cleared, HRDF interrupts are disabled. When HRIE is set, a Host Receive Data interrupt request will occur if HRDF is set.

5.8.2 HCR Host Transmit Interrupt Enable (HTIE) Bit 1

The Host Transmit Interrupt Enable (HTIE) bit is used to enable a DSP interrupt when the Host Transmit Data Empty (HTDE) status bit in the Host Status Register (HSR) is set. When HTIE is cleared, HTDE interrupts are disabled. When HTIE is set, a Host Transmit Data interrupt request will occur if HTDE is set.

5.8.3 HCR Host Command Interrupt Enable (HCIE) Bit 2

The Host Command Interrupt Enable (HCIE) bit is used to enable a vectored DSP interrupt when the Host Command Pending (HCP) status bit in the Host Status Register (HSR) is set. When HCIE is cleared, HCP interrupts are disabled. When HCIE is set, a Host Command interrupt request will occur if HCP is set. The starting address of this interrupt is determined by the Host Vector (HV).

5.8.4 HCR Host Flag 2 (HF2) Bit 3

The Host Flag 2 (HF2) bit is used as a general purpose flag for DSP to host processor communication. Changing HF2 will change the Host Flag 2 (HF2) bit of the Interrupt Status Register ISR on the host processor side of the host interface. HF2 may be set or cleared by the DSP.

5.8.5 HCR Host Flag 3 (HF3) Bit 4

The Host Flag 3 (HF3) bit is used as a general purpose flag for DSP to host processor communication. Changing HF3 will change the Host Flag 3 (HF3) bit of the Interrupt Status Register ISR on the host processor side of the host interface. HF3 may be set or cleared by the DSP.

5.8.6 HCR Reserved Control – Bits 5, 6 and 7

These unused bits are reserved for future expansion and should be written with zeros for future compatibility.

5.9 HOST STATUS REGISTER (HSR)

The Host Status register (HSR) is an 8-bit read-only status register used by the DSP to interrogate status and flags of the HI. It cannot be directly accessed by the host processor. When the HSR register is read to the internal data bus, the register contents occupy the low order byte of the data bus - the high order portion is zero filled. The status bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
DMA	*	*	HF1	HF0	HCP	HTDE	HRDF

READ-ONLY
HOST STATUS
REGISTER (HSR)
ADDRESS X:$FFE4

5.9.1 HSR Host Receive Data Full (HRDF) Bit 0

The Host Receive Data Full (HRDF) bit indicates that the Host Receive Data register (HRX) contains data from the host processor. HRDF is set when data is transferred from the TXH:TXL registers to the HRX register. HRDF is cleared when the Receive Data register HRX is read by the DSP. HRDF can also be cleared by the host processor using the Initialize function. HRDF is also cleared by a DSP reset. This bit is typically used for polling operations.

5.9.2 HSR Host Transmit Data Empty (HTDE) Bit 1

The Host Transmit Data Empty (HTDE) bit indicates that the Host Transmit Data register (HTX) is empty and can be written by the DSP. HTDE is set when the HTX register is transferred to the RXH:RXL registers. HTDE is cleared when the Transmit Data register HTX is written by the DSP. HTDE can also be set by the host processor using the Initialize function. HTDE is also set by a DSP reset. This bit is typically used for polling operations.

5.9.3 HSR Host Command Pending (HCP) Bit 2

The Host Command Pending (HCP) bit indicates that the host processor has set the HC bit and that a Host Command Interrupt is pending. The HCP bit reflects the status of the HC bit in the Command Vector Register (CVR) on the host processor side of the host interface. HC and HCP are cleared by the DSP exception hardware when the exception is taken. The host processor can clear HC which also clears HCP. The HCP is cleared by DSP reset. This bit is typically used for polling operations.

5.9.4 HSR Host Flag 0 (HF0) Bit 3

The Host Flag 0 (HF0) bit indicates the state of Host Flag 0 (HF0) in the Interrupt Control Register ICR on the host processor side of the host interface. HF0 can only be changed by the host processor. HF0 is cleared by a DSP reset.

5.9.5 HSR Host Flag 1 (HF1) Bit 4

The Host Flag 1 (HF1) bit indicates the state of Host Flag 1 (HF1) in the Interrupt Control Register ICR on the host processor side of the host interface. HF1 can only be changed by the host processor. HF1 is cleared by a DSP reset.

5.9.6 HSR Reserved Status – Bits 5 and 6

These status bits are reserved for future expansion and read as zero during DSP read operations. Reserved bits should be written as zero for future compatibility.

5.9.7 HSR DMA Status (DMA) Bit 7

The DMA status bit (DMA) indicates that the host processor has enabled the DMA mode of the HI by setting HM1 or HM0 to a one. When the DMA status bit is a zero, it indicates that the DMA mode is disabled by the Host Mode bits HM0 and HM1(both are cleared) in the Interrupt Control Register ICR and no DMA operations are pending. When the DMA status bit is set, the DMA mode is enabled by the Host Mode bits HM0 and HM1. The channel not in use (i.e., the transmit channel or receive channel) can be used for polled or interrupt operation by the DSP. DMA is cleared by reset.

5.10 INTERRUPT CONTROL REGISTER (ICR)

The Interrupt Control Register (ICR) is an 8-bit read/write control register used by the host processor to control the HI interrupts and flags. ICR cannot be accessed by the DSP. ICR is a read/write register which can be accessed using bit manipulation instructions on control register bits. The control bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
INIT	HM1	HM0	HF1	HF0	*	TREQ	RREQ

READ/WRITE
INTERRUPT CONTROL
REGISTER (ICR)
ADDRESS $0

5.10.1 ICR Receive Request Enable (RREQ) Bit 0

The Receive Request enable (RREQ) bit is used to control the $\overline{\text{HREQ}}$ pin for host receive data transfers. In the Interrupt Mode (DMA off), RREQ is used to enable interrupt requests via the external Host Request $\overline{\text{HREQ}}$ pin when the Receive Data Register Full (RXDF) status bit in the Interrupt Status register (ISR) is set. When RREQ is cleared, RXDF interrupts are disabled. When RREQ is set, the external Host Request $\overline{\text{HREQ}}$ pin will be asserted if RXDF is set.

In DMA modes, RREQ must be set or cleared by software to select the direction of DMA transfers. Setting RREQ sets the direction of the DMA transfer to be from DSP to HOST, and enables the $\overline{\text{HREQ}}$ pin to request these data transfers. RREQ is cleared by DSP reset.

5.10.2 ICR Transmit Request Enable (TREQ) Bit 1

The Transmit Request enable (TREQ) bit is used to control the $\overline{\text{HREQ}}$ pin for host transmit data transfers. In the Interrupt Mode (DMA off), TREQ is used to enable interrupt requests via the external Host Request $\overline{\text{HREQ}}$ pin when the Transmit Data Register Empty (TXDE) status bit in the Interrupt Status register (ISR) is set. When TREQ is cleared, TXDE interrupts are disabled. When TREQ is set, the external Host Request $\overline{\text{HREQ}}$ pin will be asserted if TXDE is set.

In DMA modes, TREQ must be set or cleared by software to select the direction of DMA transfers. Setting TREQ sets the direction of the DMA transfer to be from HOST to DSP, and enables the $\overline{\text{HREQ}}$ pin to request these data transfers.

Table 5-2 and Table 5-3 summarize the effect of RREQ and TREQ on the $\overline{\text{HREQ}}$ pin. TREQ is cleared by DSP reset.

5.10.3 ICR Reserved bit – Bit 2

This bit is reserved and unused. It reads as a logic zero. Reserved bits should be written as zero for future compatibility.

Table 5-2 HREQ Pin Definition - Interrupt Mode

TREQ	RREQ	$\overline{\text{HREQ}}$ Pin
0	0	No Interrupts (Polling)
0	1	RXDF Request (Interrupt)
1	0	TXDE Request (Interrupt)
1	1	RXDF and TXDE Request (Interrupt)

Table 5-3 HREQ Pin Definition - DMA Mode

TREQ	RREQ	$\overline{\text{HREQ}}$ Pin
0	0	DMA Transfers Disabled
0	1	DSP→ HOST Request (RX)
1	0	HOST→ DSP Request (TX)
1	1	Undefined (Illegal)

5.10.4 ICR Host Flag 0 (HF0) Bit 3

The Host Flag 0 (HF0) bit is used as a general purpose flag for host processor to DSP communication. HF0 may be set or cleared by the host processor and cannot be changed by the DSP. Changing HF0 also changes the Host Flag bit 0 (HF0) of the Host Status register HSR on the DSP side of the HI. HF0 is cleared by DSP reset.

5.10.5 ICR Host Flag 1 (HF1) Bit 4

The Host Flag 1 (HF1) bit is used as a general purpose flag for host processor to DSP communication. HF1 may be set or cleared by the host processor and cannot be changed by the DSP. Changing HF1 also changes the Host Flag bit 1 (HF1) of the Host Status register HSR on the DSP side of the HI. HF1 is cleared by a DSP reset.

5.10.6 ICR Host Mode Control (HM1, HM0) Bits 5 and 6

The Host Mode control bits HM0 and HM1 select the transfer mode of the HI. HM1 and HM0 enable the DMA mode of operation or interrupt (non-DMA) mode of operation.

When the DMA mode is enabled, the $\overline{\text{HREQ}}$ pin is used as a DMA Transfer Request output to a DMA controller and the $\overline{\text{HACK}}$ pin is used as a DMA Transfer Acknowledge input from a DMA controller. The DMA Control bits HM0 and HM1 select the size of the DMA word to be transferred as shown in Table 5-4. The direction of the DMA transfer is selected by the TREQ and RREQ bits.

Table 5-4 Host Mode (HM1, HM0) Bit Definition

HM1	HM0	Mode
0	0	Interrupt Mode (DMA off)
0	1	Illegal
1	0	DMA mode - 16 bit
1	1	DMA mode - 8 bit

When both HM1 and HM0 are cleared, the DMA mode is disabled and the TREQ and RREQ control bits are used for host processor interrupting via the external Host Request $\overline{\text{HREQ}}$ output pin. In the interrupt mode, the Host Acknowledge $\overline{\text{HACK}}$ input pin is used for the MC68000 family vectored Interrupt Acknowledge input.

When HM1 or HM0 are set, the DMA mode is enabled and the $\overline{\text{HREQ}}$ pin is not available for host processor interrupts. When the DMA mode is enabled, the TREQ and RREQ bits selects the direction of DMA transfers; the Host Acknowledge $\overline{\text{HACK}}$ input pin is used as a DMA Transfer Acknowledge input. If the DMA direction is from DSP to Host, the contents of the selected register are enabled onto the Host Data Bus when $\overline{\text{HACK}}$ is asserted. If the DMA direction is from Host to DSP, the selected register is written to TXH or TXL from the Host Data Bus when $\overline{\text{HACK}}$ is asserted. The size of the DMA word to be transferred is determined by the DMA control bits HM0 and HM1. The HI register selected during a DMA transfer is determined by a 2-bit address counter which is preloaded with the value in HM1 and HM0. The address counter substitutes for the Host Address bits HA1 and HA0 of the HI during a DMA transfer. The Host Address bit HA2 is forced to one during each DMA transfer. The address counter can be initialized with the INIT bit feature. After each DMA transfer on the Host Data Bus, the address counter is incremented to the

next register. When the address counter reaches the highest register (RXL or TXL), the address counter is not incremented but is loaded with the value in HM1 and HM0. This allows 8- or 16-bit data to be transferred in a circular fashion and eliminates the need for the DMA controller to supply the Host Address HA2, HA1 and HA0 pins. For 16-bit data transfers, the DSP interrupt rate is reduced by a factor of 2 from the Host Request rate.

HM1 and HM0 are cleared by DSP reset.

5.10.7 ICR Initialize Bit (INIT) Bit 7

The INIT bit is used by the host to force initialization of the HI hardware. This may or may not be necessary, depending on the software design of the interface. The type of initialization done depends on the state of TREQ and RREQ. The INIT command is designed to conveniently convert into the desired data transfer mode after the INIT is completed. The commands are described below and in Table 5-5a. The host sets INIT which causes the HI hardware to execute the command. The interface hardware clears INIT when the command is complete. INIT is cleared by DSP reset.

Note that INIT execution always loads the DMA address counter and clears the channel according to TREQ and RREQ. INIT execution is not affected by HM1 and HM0.

Table 5-5a INIT Execution Definition

TREQ	RREQ	After INIT Execution — Interrupt Mode (HM1=0, HM0=0)
0	0	INIT=0; “address counter = 00”
0	1	INIT=0; RXDF=0; HTDE=1; “address counter = 00”
1	0	INIT=0; TXDE=1; HRDF=0; “address counter = 00”
1	1	INIT=0; RXDF=0; HTDE=1; TXDE=1; HRDF=0; “address counter = 00”

Table 5-5ab INIT Execution Definition

TREQ	RREQ	After INIT Execution — DMA Mode (HM1 or HM0 = 1)
0	0	INIT=0; address counter = HM1,HM0
0	1	INIT=0; RXDF=0; HTDE=1; address counter = HM1,HM0
1	0	INIT=0; TXDE=1; HRDF=0; address counter = HM1,HM0
1	1	Undefined (illegal)

5.11 INTERRUPT STATUS REGISTER (ISR)

The Interrupt Status register (ISR) is an 8-bit read-only status register used by the host processor to interrogate the status and flags of the HI. The ISR can not be accessed by the DSP. The status bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
HREQ	DMA	*	HF3	HF2	TRDY	TXDE	RXDF

READ-ONLY
INTERRUPT STATUS
REGISTER (ISR)
ADDRESS $2

5.11.1 ISR Receive Data Register Full (RXDF) Bit 0

The Receive Data Register Full (RXDF) bit indicates that both the Receive Byte Registers, RXH and RXL, contain data from the DSP and may be read by the host processor. RXDF is set when the contents of the Host Transmit Data Register HTX is transferred to the Receive Byte Registers RXH:RXL. RXDF is cleared when the Receive Data Low (RXL) register is read by the host processor. RXL is normally the last byte of the Receive Byte Registers to be read by the host processor. RXDF can be cleared by the host processor using the Initialize function. RXDF is cleared by a DSP reset. RXDF may be used to assert the external Host Request $\overline{\text{HREQ}}$ pin if the Receive Request enable RREQ bit is set. RXDF provides valid status regardless of whether the RXDF interrupt is enabled or not so that polling techniques may be used by the host processor.

5.11.2 ISR Transmit Data Register Empty (TXDE) Bit 1

The Transmit Data Register Empty (TXDE) bit indicates that the Transmit Byte Registers TXH and TXL are both empty and can be written by the host processor. TXDE is set when the content of the Transmit Byte Registers TXH:TXL are transferred to the Host Receive Data Register (HRX). TXDE is cleared when the Transmit Byte Low (TXL) register is written by the host processor. TXL is normally the last byte of the Transmit Byte Registers to be written by the host processor. TXDE can be set by the host processor using the Initialize feature. TXDE is set by a DSP reset. TXDE may be used to assert the external Host Request $\overline{\text{HREQ}}$ pin if the Transmit Request Enable TREQ bit is set. TXDE provides valid status regardless of whether the TXDE interrupt is enabled or not so that polling techniques may be used by the host.

5.11.3 ISR Transmitter Ready (TRDY) Bit 2

The Transmitter Ready (TRDY) status bit indicates that both the Transmit Byte Registers and Host Receive Data register are empty, i.e., the channel from the host processor through the HI to the DSP CPU is clear. By testing TRDY, the host processor programmer can be assured that the first word received by the DSP will be the first word the host processor transmits.

TRDY = TXDE ^ HRDF

The DSP reset will set TRDY.

5.11.4 ISR Host Flag 2 (HF2) Bit 3

The Host Flag 2 (HF2) bit indicates the state of Host Flag 2 (HF2) in the Host Control Register (HCR). HF2 can only be changed by the DSP. HF2 is cleared by a DSP reset.

5.11.5 ISR Host Flag 3 (HF3) Bit 4

The Host Flag 3 (HF3) bit indicates the state of Host Flag 3 (HF3) in the Host Control Register HCR. HF3 can only be changed by the DSP. HF3 is cleared by a DSP reset.

5.11.6 ISR (Reserved Status) Bit 5

This status bit is reserved for future expansion and will read as zero during host processor read operations. Reserved bits should be written as zero for future compatibility.

5.11.7 ISR DMA Status (DMA) Bit 6

The DMA status bit (DMA) indicates that the host processor has enabled the DMA mode of the HI (HM1 or HM0 =1). When the DMA status bit is clear, it indicates that the DMA mode is disabled by the Host Mode bits HM0 and HM1 in the Interrupt Control Register ICR and no DMA operations are pending. When DMA is set, it indicates that the DMA mode is enabled and the host processor should not use the active DMA channel (RXH:RXL or TXH:TXL depending on DMA direction) to avoid conflicts with the DMA data transfers.

5.11.8 ISR Host Request (HREQ) Bit 7

The Host Request (HREQ) bit indicates the status of the external Host Request $\overline{\text{HREQ}}$ output pin. When the HREQ status bit is cleared, it indicates that the external $\overline{\text{HREQ}}$ pin is deasserted and no host interrupts or DMA transfers are being requested. When the HREQ status bit is set, it indicates that the external $\overline{\text{HREQ}}$ pin is asserted indicating that the DSP is interrupting the host processor or that a DMA Transfer Request is being made. The HREQ interrupt request may originate from one or more of 2 sources - the Receive Byte Registers are full or the Transmit Byte Registers are empty. These conditions are indicated by the Interrupt Status register (ISR) RXDF and TXDE status bits, respectively. If the interrupt source has been enabled by the associated request enable bit in the Interrupt Control Register ICR, HREQ will be set if one or more of the 2 enabled interrupt sources is set. DSP reset will clear HREQ.

5.12 INTERRUPT VECTOR REGISTER (IVR)

The Interrupt Vector Register (IVR) is an 8-bit read/write register which contains the exception vector number for use with MC68000 processor family vectored interrupts. This

register is accessible only to the host processor. The contents of the IVR register are placed on the Host Data Bus, H0-H7, when $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ pins both are asserted and the DMA mode is disabled. The content of this register is initialized to $0F by a DSP reset. This corresponds to the un-initialized exception vector in the MC68000 family.

7	6	5	4	3	2	1	0
IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0

READ/WRITE
INTERRUPT CONTROL
REGISTER (IVR);
ADDRESS $3

5.13 IVR HOST INTERFACE INTERRUPTS

The HI may request interrupt service from either the DSP or host processor. The DSP interrupts are internal and do not require the use of an external interrupt pin. The DSP acknowledges host interrupts by jumping to the appropriate interrupt service routine. The DSP interrupt service routine must read or write the appropriate HI register (i.e. clearing HRDF or HTDE for example) to clear the interrupt. In the case of Host Command interrupts, the interrupt acknowledge from the program controller will clear the pending interrupt condition.

The host processor interrupts are external and use the Host Request $\overline{\text{HREQ}}$ pin. H$\overline{\text{REQ}}$ is normally connected to the host processor maskable interrupt (IRQ) input. The host processor acknowledges host interrupts by executing an interrupt service routine. The MC68000 processor family will assert the $\overline{\text{HACK}}$ pin to read the exception vector number from the Interrupt Vector Register (IVR) of the HI. The most significant bit (HREQ) of the Interrupt Status Register (ISR) may be tested to determine if the DSP is the interrupting device and the two least significant bits (RXDF and TXDE) may be tested to determine the interrupt source. The host processor interrupt service routine must read or write the appropriate HI register to clear the interrupt.

5.14 DMA MODE OPERATION

The DMA mode allows the transfer of 8- or 16-bit data between the DSP HI and an external DMA controller. The HI provides the pipeline data registers and the synchronization logic between the two asynchronous processor systems. The DSP Host Exceptions provide cycle-stealing data transfers with the DSP internal or external memory. This allows the DSP memory address to be generated using any of the DSP addressing modes and modifiers. Queues and circular sample buffers are easily created for DMA transfer regions. The DSP Host Exceptions appear as high priority fast or long exception service routines. The external DMA controller provides the transfers between the DSP HI registers and the external DMA memory. The external DMA controller must provide the address to the external DMA memory. The address of the selected HI register is provided by a DMA address counter in the DSP HI.

5.14.1 Host to DSP – Host Interface Action

The following procedure outlines the steps that the HI hardware takes to transfer DMA data from the Host Data Bus to DSP memory.

1. Assert the Host Request $\overline{\text{HREQ}}$ output pin when the transmit byte registers TXH:TXL are empty (This always occurs in HOST to DSP DMA mode when TXDE=1).
2. Write the selected Transmit Byte register from the Host Data Bus when the $\overline{\text{HACK}}$ input pin is asserted by the DMA controller. Deassert the $\overline{\text{HREQ}}$ pin.
3. If the highest register address has not been reached (i.e., TXDE=1), postincrement the DMA address counter to select the next register. Wait until $\overline{\text{HACK}}$ is deasserted then go to step 1.
4. If the highest register address has been reached ((i.e., TXDE=0), load the DMA address counter with the value in HM1 and HM0 and transfer the Transmit Byte Registers TXH:TXL to the Host Receive Data Register HRX when HRDF=0. This will set HRDF=1. Wait until $\overline{\text{HACK}}$ is deasserted then go to step 1.

NOTES:

- The DSP to HOST data transfers can occur normally in the channel not used for DMA except that the HOST must use polling and not interrupts.
- The transfer of data from the TXH:TXL register to the HRX register automatically loads the DMA address counter from the HM1 and HM0 bits in the DMA HOST to DSP mode.

The host exception is triggered when HRDF=1. The host exception routine must read the Host Receive Data Register HRX to clear HRDF. The transfer from step 4 to step 1 will automatically occur if TXDE=1. Note that the execution of the host exception on HRDF=1 condition will occur after the transfer to step 1 and is independent of the handshake since it is only dependent on HRDF=1.

5.14.2 Host To DSP – Host Processor Procedure

The following procedure outlines the typical steps that the host processor must take to set-up and terminate a host to DSP DMA transfer.

1. Setup the external DMA controller source address, direction, byte count and other control registers. Enable the DMA controller channel.
2. Set TXDE and clear HRDF. This can be done with the appropriate Initialize function. The host must also initialize the DMA counter in the HI using the Initialize feature.

3. The DSP's destination pointer used in the DMA exception handler (an address register for example) must be initialized and HRIE must be set to enable the HRDF interrupt. This could be done with a separate Host Command exception routine in the DSP. $\overline{\text{HREQ}}$ output pin will be asserted immediately by the DSP hardware which begins the DMA transfer.

4. Perform other tasks until interrupted by the DMA controller DMA Complete interrupt. The DSP Interrupt Control Register (ICR), the Interrupt Status Register (ISR), and RXH:RXL may be accessed at any time by the host processor (using HA0-HA2, HR/$\overline{\text{W}}$, and $\overline{\text{HEN}}$) but the Transmit Byte Registers (TXH:TXL) may not be accessed until the DMA mode is disabled.

5. Terminate the DMA controller channel to disable DMA transfers.

6. Terminate the DSP HI DMA mode by clearing the HM1 and HM0 bits and clearing TREQ in the Interrupt Control Register (ICR).

5.14.3 DSP to Host Interface Action

The following procedure outlines the steps that the HI hardware takes to transfer DMA data from DSP memory to the Host Data Bus.

1. The transmit exception will be triggered when HTIE=1 and HTDE=1. The exception routine software will write the data word into HTX.

2. Transfer the HTX register to the Receive Byte Registers RXH:RXL when they are empty (RXDF=0). This will automatically occur. Load the DMA address counter from HM1 and HM0. This action will set HTDE=1 and trigger another DSP transmit exception to write HTX (i.e., HTDE=0).

3. Assert the Host Request $\overline{\text{HREQ}}$ pin when the Receive Byte Registers are full.

4. Enable the selected Receive Byte register on the Host Data Bus when $\overline{\text{HACK}}$ is asserted. Deassert the Host Request $\overline{\text{HREQ}}$ pin.

5. If the highest register address has not been reached (i.e., RXDF=1), postincrement the DMA address counter to select the next register. Wait until $\overline{\text{HACK}}$ is deasserted then go to step 3.

6. If the highest register address has been reached (i.e., RXDF=0), wait until $\overline{\text{HACK}}$ is deasserted then go to step 2. The DSP transmit exception must have written HTX (i.e., HTDE=0) before Step 2 will be executed.

NOTES:

- The HOST→ DSP data transfers can occur normally in the channel not used for DMA except that the HOST must use polling and not interrupts.

- The transfer of data from the HTX register to the RXH:RXL registers automatically loads the DMA address counter from the HM1 and HM0 bits when in DMA DSP to HOST mode.

5.14.4 DSP To Host – Host Processor Procedure

The following procedure outlines the typical steps that the host processor must take to set-up and terminate a DSP to Host DMA transfer.

1. Setup the DMA controller destination address, direction, byte count and other control registers. Enable the DMA controller channel.
2. Set HTDE and clear RXDF. This can be done with the appropriate INIT function.
3. The DSP's source pointer used in the DMA exception handler (an address register for example) must be initialized and HTIE must be set to enable the DSP host transmit interrupt. This could be done by the host with a Host Command exception routine. The DSP host transmit exception will be activated immediately by DSP hardware which begins the DMA transfer.
4. Perform other tasks until interrupted by the DMA controller DMA Complete interrupt. The DSP Interrupt Control Register (ICR), the Interrupt Status Register (ISR), and TXH:TXL may be accessed at any time by the host processor (using HA0-HA2, HR/$\overline{W}$, and $\overline{HEN}$) but the Receive Byte Registers (RXH and RXL) may not be accessed until the DMA mode is disabled.
5. Terminate the DMA controller channel to disable DMA transfers.
6. Terminate the DSP HI DMA mode by clearing the HM1 and HM0 bits and clearing RREQ in the Interrupt Control Register (ICR).

5.15 HOST PORT USAGE – GENERAL CONSIDERATIONS

Careful synchronization is required when reading multi-bit registers that are written by another asynchronous system. This is a common problem when two asynchronous systems are connected. The situation exists in the host port. However, if the port is used in the way it was designed, proper operation is guaranteed. The considerations for proper operation are discussed below.

5.15.1 Host Programmer Considerations

1. Unsynchronized Reading of Receive Byte Registers.

   When reading receive byte registers, RXH or RXL, the host programmer should use interrupts or poll the RXDF flag which indicates that data is available. This guarantees that the data in the receive byte registers will be stable.

2. Overwriting Transmit Byte Registers.

   The host programmer should not write to the transmit byte registers, TXH or TXL, unless the TXDE bit is set, indicating that the transmit byte registers are empty. This guarantees that the DSP will read stable data when it reads the HRX register.

3. Synchronization of Status Bits from DSP to Host.

   HC, HREQ, DMA, HF3, HF2, TRDY, TXDE, and RXDF status bits are set or cleared from inside the DSP and read by the host processor. The host can read these status bits very quickly without regard to the clock rate used by the DSP, but there is a chance that the state of the bit could be changing during the read operation. This is generally not a system problem, since the bit will be read correctly in the next pass of any host polling routine. However, if the host holds the $\overline{\text{HEN}}$ input pin for the minimum assert time plus 1.5 Ccyc, the Status data is guaranteed to be stable. The 1.5 Ccyc is first used to synchronize the $\overline{\text{HEN}}$ signal and then to block internal updates of the status bits. There is no other minimum $\overline{\text{HEN}}$ assert time relationship to DSP clocks. There is a minimum $\overline{\text{HEN}}$ deassert time of 1.5 Ccyc so that the blocking latch can be deasserted to allow updates if the host is in a tight polling loop. This only applies to reading status bits.

   The only potential system problem with the uncertainty of reading any status bits by the host is HF3 and HF2 as an encoded pair. For example, if the DSP changes HF3 and HF2 from “00” to “11” there is a very small probability that the host could read the bits during the transition and receive “01” or “10” instead of “11”. If the combination of HF3 and HF2 has significance, the host would potentially read the wrong combination.

   Solutions:

   a. Read the bits twice and check for consensus.

   b. Assert $\overline{\text{HEN}}$ access for $\overline{\text{HEN}}$ + 1.5 Ccyc so that status bit transitions are stabilized.

4. Overwriting the Host Vector

   The host programmer should change the Host Vector register only when the Host Command bit (HC) is clear. This will guarantee that the DSP interrupt control logic will receive a stable vector.

5. Cancelling a pending Host Command Exception

   The host processor may elect to clear the HC bit to cancel the Host Command Exception request at any time before it is recognized by the DSP. Because the host does not know exactly when the exception will be recognized, because of synchronization, and because pipelining of exception processing, the DSP may execute the host exception

after the HC bit is cleared. For this reason, the HV must not be changed at the same time the HC bit is cleared. In this way, if the exception was taken, the vector will be known.

5.15.2 DSP programmer considerations

1. Reading HF1 and HF0 as an encoded pair.

   DMA, HF1, HF0, HCP, HTDE, and HRDF status bits are set or cleared by the host processor side of the interface. These bits are individually synchronized to the DSP clock.

   The only system problem with reading status is HF1 and HF0 if they are encoded as a pair, e.g. the four combinations 00, 01, 10, and 11 each have significance. This is because there is a very small probability that the DSP will read the status bits that were synchronized during transition. The solution to this potential problem is to read the bits twice for consensus.