

# ***TMS320VC5509 DSP Host Port Interface (HPI) Reference Guide***

Literature Number: SPRU619  
April 2003



## **IMPORTANT NOTICE**

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of that third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments  
Post Office Box 655303  
Dallas, Texas 75265

## Read This First

### About This Manual

This manual describes the features and operation of the host port interface (HPI) that is available on the TMS320VC5509 digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation.

### Notational Conventions

This document uses the following conventions:

- ❑ In most cases, hexadecimal numbers are shown with the suffix h. For example, the following number is a hexadecimal 40 (decimal 64):

40h

Similarly, binary numbers often are shown with the suffix b. For example, the following number is the decimal number 4 shown in binary form:

0100b

- ❑ If a signal or pin is active low, it has an overbar. For example, the  $\overline{\text{RESET}}$  signal is active low.
- ❑ Bits and signals are sometimes referenced with the following notations:

Notation	Description	Example
Register(n-m)	Bits n through m of Register	R(15-0) represents the 16 least significant bits of register R.
Bus[n:m]	Signals n through m of Bus	A[21:1] represents signals 21 through 1 of bus A.

- ❑ The following terms are used to name portions of data:

Term	Description	Example
LSB	Least significant bit	In R(15-0), bit 0 is the LSB of register R.
MSB	Most significant bit	In R(15-0), bit 15 is the MSB of register R.

## **Related Documentation From Texas Instruments**

The following documents describe the C55x devices and related support tools. Copies of these documents are available on the Internet at [www.ti.com](http://www.ti.com).  
*Tip:* Enter the literature number in the search box provided at [www.ti.com](http://www.ti.com).

***TMS320VC5509 Fixed-Point Digital Signal Processor Data Manual*** (literature number SPRS163) describes the features of the TMS320VC5509 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

***TMS320VC5509A Fixed-Point Digital Signal Processor Data Manual*** (literature number SPRS205) describes the features of the TMS320VC5509A fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

***TMS320C55x Technical Overview*** (literature number SPRU393) introduces the TMS320C55x DSPs, the latest generation of fixed-point DSPs in the TMS320C5000™ DSP platform. Like the previous generations, this processor is optimized for high performance and low-power operation. This book describes the CPU architecture, low-power enhancements, and embedded emulation features.

***TMS320C55x DSP CPU Reference Guide*** (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x DSPs.

***TMS320C55x DSP Peripherals Overview Reference Guide*** (literature number SPRU317) introduces the peripherals, interfaces, and related hardware that are available on TMS320C55x DSPs.

***TMS320C55x DSP Algebraic Instruction Set Reference Guide*** (literature number SPRU375) describes the TMS320C55x DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

***TMS320C55x DSP Mnemonic Instruction Set Reference Guide*** (literature number SPRU374) describes the TMS320C55x DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

***TMS320C55x Optimizing C/C++ Compiler User's Guide*** (literature number SPRU281) describes the TMS320C55x C/C++ Compiler. This C/C++ compiler accepts ISO standard C and C++ source code and produces assembly language source code for TMS320C55x devices.

***TMS320C55x Assembly Language Tools User's Guide*** (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x devices.

***TMS320C55x DSP Programmer's Guide*** (literature number SPRU376) describes ways to optimize C and assembly code for the TMS320C55x DSPs and explains how to write code that uses special features and instructions of the DSPs.

## ***Trademarks***

TMS320, TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

Trademarks are the property of their respective owners.

# Contents

<b>1</b>	<b>Introduction to the HPI</b>	<b>2</b>
<b>2</b>	<b>DSP Memory Accessible Through the HPI</b>	<b>3</b>
<b>3</b>	<b>HPI-DMA Interaction</b>	<b>4</b>
<b>4</b>	<b>HPI Signals</b>	<b>5</b>
4.1	HPI and EMIF Sharing Pins	5
4.2	HPI Signal Summary	5
4.3	HDS2, HDS1, and HCS: Data Strobing and Chip Selection	8
4.4	HBE[1:0]: Indicating Which Byte or Bytes to Access	8
<b>5</b>	<b>Nonmultiplexed Mode</b>	<b>10</b>
5.1	Signal Connections in the Nonmultiplexed Mode	10
5.2	Indicating the Cycle Type in the Nonmultiplexed Mode	11
<b>6</b>	<b>Multiplexed Mode</b>	<b>13</b>
6.1	Signal Connections in the Multiplexed Mode	13
6.2	Indicating the Cycle Type in the Multiplexed Mode	15
6.3	Loading HPIA With an Address	17
6.4	Auto-Increment Option: Automatic Address Increment Between Transfers	17
<b>7</b>	<b>Interrupts Between the Host and the DSP</b>	<b>18</b>
7.1	Sending an Interrupt Request From the Host to the DSP	18
7.2	Sending an Interrupt Request From the DSP to the Host	18
<b>8</b>	<b>Boot Loading With the HPI</b>	<b>18</b>
<b>9</b>	<b>Power, Emulation, and Reset Considerations</b>	<b>19</b>
9.1	HPI and the IDLE Instruction	19
9.2	HPI Emulation Modes	19
9.3	Effects of a DSP Reset on the HPI	19
<b>10</b>	<b>HPI Registers</b>	<b>20</b>
10.1	Data Register (HPID)	20
10.2	Address Register (HPIA)	20
10.3	Control Register (HPIC)	21

# Figures

1	The Position of the HPI in a Host-DSP System .....	2
2	DSP Memory Accessible Through the HPI .....	3
3	HPI Strobe and Select Logic .....	8
4	Example of Host-DSP Signal Connections in the Nonmultiplexed Mode .....	11
5	Example of Host-DSP Signal Connections When Using the $\overline{\text{HAS}}$ Signal in the Multiplexed Mode .....	14
6	Example of Host-DSP Signal Connections When the $\overline{\text{HAS}}$ Signal is Tied High in the Multiplexed Mode .....	15
7	Data Register (HPID) .....	20
8	Address Register (HPIA) .....	20
9	Control Register (HPIC) .....	21

# Tables

1	Signals of the HPI .....	5
2	Effect of Driving the HPI Byte-Enable Signals in the Nonmultiplexed Mode .....	9
3	Examples of Using the $\overline{\text{HBE}}$ Signals for Host Write Cycles .....	9
4	Examples of Using the $\overline{\text{HBE}}$ Signals for Host Read Cycles .....	9
5	Access Types Selectable With the HCNTL0 Signal in the Nonmultiplexed Mode .....	12
6	Cycle Types Selectable With the HCNTL0 and $\text{HR}/\overline{\text{W}}$ Signals in the Nonmultiplexed Mode .....	12
7	Access Types Selectable With the HCNTL[1:0] Signals in the Multiplexed Mode .....	16
8	Cycle Types Selectable With the HCNTL[1:0] and $\text{HR}/\overline{\text{W}}$ Signals in the Multiplexed Mode .....	16
9	Control Register (HPIC) Field Descriptions .....	21

# Host Port Interface (HPI)

---

---

---

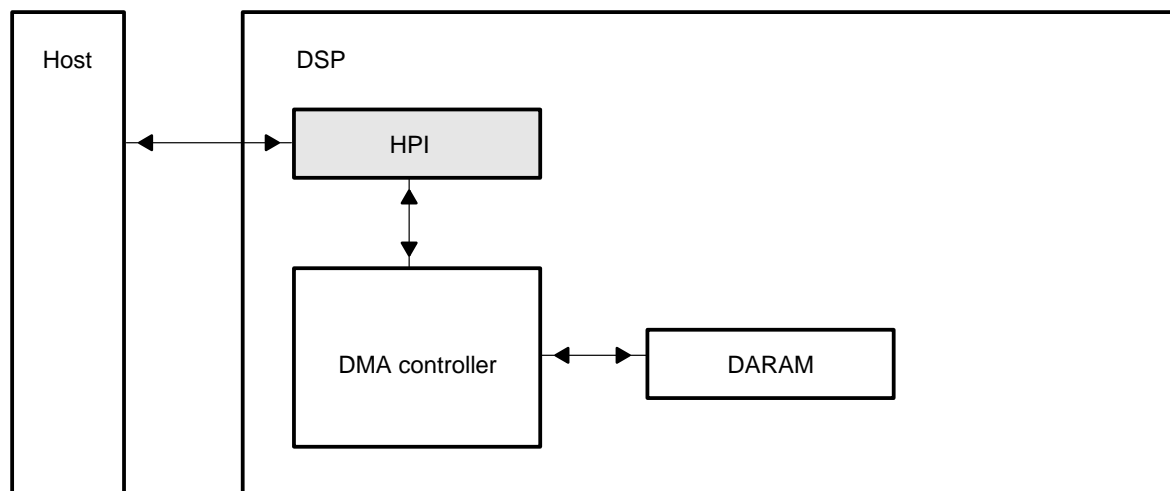
This chapter describes the features and operation of the host port interface (HPI) on TMS320VC5509 devices in the TMS320C55x™ (C55x™) DSP generation. Through the HPI, an external host processor (host) can directly access a portion of the dual-access RAM (DARAM) inside the DSP.



## 1 Introduction to the HPI

The host port interface (HPI) provides a 16-bit-wide parallel port through which an external host processor (host) can directly access part of the dual-access RAM (DARAM) inside the DSP. The HPI uses 14-bit addresses, where each address is assigned to a 16-bit word in memory. Figure 1 is a conceptual diagram of the connections between the HPI and other components of a host-DSP system.

Figure 1. The Position of the HPI in a Host-DSP System



The DMA controller handles all HPI accesses. Through the DMA controller, you can choose one of two HPI access configurations. In one configuration, the HPI shares the DARAM with the DMA channels. In the other configuration, the HPI has exclusive access to the DARAM.

The HPI cannot directly access other peripherals' registers. If the host requires data from other peripherals, that data must be moved to the DARAM first, either by the CPU or by activity in one of the six DMA channels. Likewise, data from the host must be transferred to the DARAM before being transferred to other peripherals.

To provide flexibility in the choice of a host, the HPI allows two modes for passing data and addresses. The nonmultiplexed mode (see section 5 on page 10) provides the host processor with separate address and data buses. The multiplexed mode (see section 6 on page 13) provides a single bus to transport address and data information. The different modes require some different connections to HPI signals. There are three HPI registers for data, addresses, and control information (see section 10 on page 20).

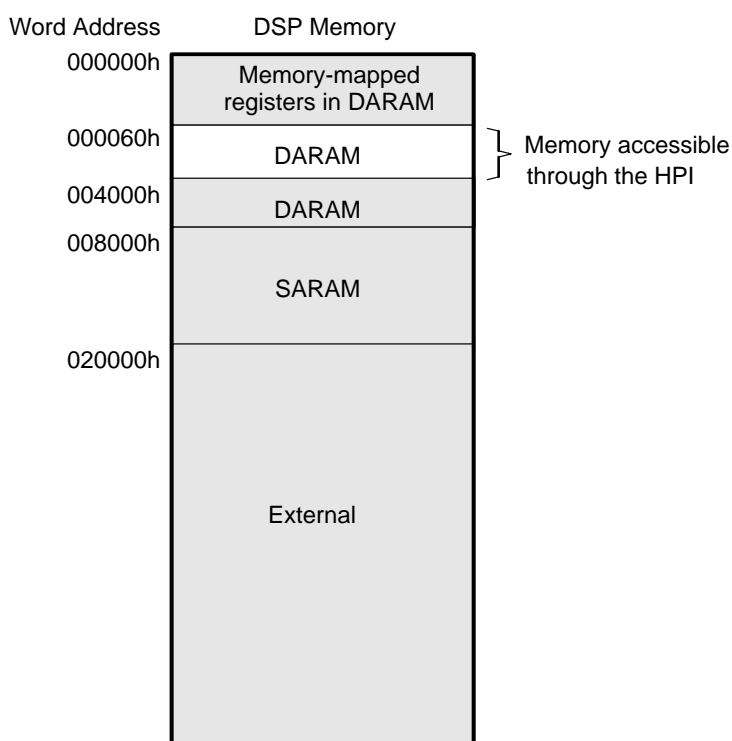
## 2 DSP Memory Accessible Through the HPI

Addresses driven by a host on the external address lines of the HPI are treated as word addresses, not byte addresses. Each HPI address corresponds to a 16-bit word in data space.

Figure 2 highlights the portion of the DSP data memory map that is accessible to a host through the HPI. The shaded areas in the memory map are not accessible through the HPI.

The 14 address lines of the HPI enable the host to access the internal dual-access RAM (DARAM) at addresses 000060h-003FFFh. Addresses 000000h-00005Fh are reserved for the memory-mapped registers (MMRs) of the CPU and are not accessible through the HPI.

Figure 2. DSP Memory Accessible Through the HPI



**Note:** The shaded areas of the memory map are not accessible through the HPI.

### 3 HPI-DMA Interaction

The HPI uses the DMA controller to move data into and out of DSP memory. The DMA controller services the HPI (via a dedicated port) and six programmable DMA channels. Activity in the channels is controlled by such factors as their priority, the DMA port resources they use, and whether they are triggered by synchronization events. The HPI and the six channels are serviced by the DMA controller in a round-robin manner. Because of this structure, the activity in the enabled channels affects the latency of HPI transactions and vice versa.

Two programmable options can affect the latencies of HPI-DMA interaction:

- ❑ The EHPIPRIO bit in the DMA global control register (DMAGCR) controls the priority of the HPI requests in the DMA service chain. When EHPIPRIO = 0, HPI requests are considered low priority and are serviced after all high priority channels. When EHPIPRIO = 1, HPI requests are considered high priority and are serviced before low priority channels. If the HPI and any channels are at the same priority level, they are serviced in a round-robin manner.
- ❑ The EHPIEXCL bit in DMAGCR controls whether the HPI has exclusive access to the internal memory of the DSP. In the case of the TMS320VC5509 HPI, this means exclusive access to the internal DARAM within its address reach. When EHPIEXCL = 0 (not exclusive), the DMA channels can use any DMA port. When EHPIEXCL = 1 (exclusive), the DMA channels cannot access the DARAM and SARAM ports; they can access only the EMIF port (for external memory) and the peripheral port. Any channels configured to use internal memory are suspended until the HPI-exclusive condition is released. This capability provides the host a minimum latency operation condition at the expense of suspending the channels.

For detailed information on the service chain and the DMA global control register, see the *TMS320VC5509/5510 DSP Direct Memory Access (DMA) Reference Guide* (SPRU587).

## 4 HPI Signals

Section 4.1 explains how the HPI and the external memory interface (EMIF) share pins. Section 4.2 provides a summary description of each HPI signal.

### 4.1 HPI and EMIF Sharing Pins

On all TMS320VC5509 devices, the HPI shares a parallel port with the external memory interface (EMIF). Parallel port mode bits (bits 1-0) in the external bus selection register (EBSR) determine whether the port is used for data EMIF mode (00b), full EMIF mode (01b), nonmultiplexed HPI mode (10b), or multiplexed HPI mode (11b).

The reset value of the parallel port mode bits is determined by the state of the GPIO0 pin at reset. If GPIO0 is high at reset, the full EMIF mode is enabled. If GPIO0 is low at reset, the multiplexed HPI mode is enabled. After reset, your software can modify the EBSR to select a different mode.

See the device-specific data manual for more details about EBSR.

### 4.2 HPI Signal Summary

Table 1 summarizes the HPI signals. In the Type column, Z refers to the high-impedance state. There are some differences in the signal connections between the two modes of the HPI: Nonmultiplexed mode (see section 5 on page 10) and multiplexed mode (see section 6 on page 13). For timing information on the HPI signals, see the device-specific data manual: *TMS320VC5509 Fixed-Point Digital Signal Processor Data Manual* (SPRS163) or *TMS320VC5509A Fixed-Point Digital Signal Processor Data Manual* (SPRS205).

Table 1. Signals of the HPI

Signal(s)	Type	Description
HD[15:0]	Input/Output/Z	<p>HPI data bus. HD is a parallel, bidirectional, 3-state bus.</p> <p>In the nonmultiplexed mode: These 16 signals are used to carry data only.</p> <p>In the multiplexed mode: These 16 signals are used to carry both addresses and data.</p> <p>Between data transfers: The HPI does not drive HD. If the bus holders are enabled, HD retains the last driven state. If the bus holders are disabled, HD enters the high-impedance state. For information about the bus holders, see the device-specific data manual.</p>
HA[13:0]	Input	<p>HPI address bus. HA is a parallel, unidirectional address bus that is used only in the nonmultiplexed mode. HA carries 14-bit addresses from the host processor to the HPI. The 14 lines of this bus allow the addressing of 16K words of the DSP memory.</p>

Table 1. Signals of the HPI (Continued)

Signal(s)	Type	Description										
HBE[1:0]	Input	<p>Host byte-enable signals. These two signals determine whether the host processor is accessing the whole word, the least significant byte (LSByte), or the most significant byte (MSByte) of HPIA, HPIC, or the addressed memory location (via HPID).</p> <p>The effect of the HBE pins is shown in the following table (0 = low, 1 = high). For more details about these options, see section 4.4 on page 8.</p> <table><tr><th>HBE[1:0]</th><th>Access</th></tr><tr><td>00</td><td>Word</td></tr><tr><td>01</td><td>MSByte</td></tr><tr><td>10</td><td>LSByte</td></tr><tr><td>11</td><td>Reserved (do not use)</td></tr></table> <p><b>Note:</b> The HBE signals and their associated function are not supported on the original TMS320VC5509 devices, but they are supported on the TMS320VC5509A devices.</p>	HBE[1:0]	Access	00	Word	01	MSByte	10	LSByte	11	Reserved (do not use)
HBE[1:0]	Access											
00	Word											
01	MSByte											
10	LSByte											
11	Reserved (do not use)											
HCS	Input	HPI chip-select signal. HCS serves as the enable input of the HPI, and must be low during an access. For the relationships among the signals HDS1, HDS2, HCS, and HRDY, see section 4.3 on page 8.										
HR/W	Input	HPI read/write signal. This input indicates the direction of the host access. When high, HR/W indicates a read from the DSP memory. When low, HR/W indicates a write to the DSP memory.										
HDS1, HDS2	Input	<p>HPI data strobe signals. The exclusive-NOR of HDS1 and HDS2 forms a strobe signal for controlling data transfers during host-access cycles. For the relationships among the signals HDS1, HDS2, HCS, and HRDY, see section 4.3 on page 8. Connections to HDS1 and HDS2 depend on the host's strobe signal(s):</p> <table><tr><th>Available Host Data Strobe Pins</th><th>Connections to HPI Data Strobe Pins</th></tr><tr><td>Host has separate read and write strobe pins, both active-low</td><td>Connect one strobe to HDS1 and the other to HDS2.</td></tr><tr><td>Host has one active-low strobe pin</td><td>Connect the strobe to HDS1 or HDS2, and connect the remaining HDS pin to logic level 1.</td></tr><tr><td>Host has one active-high strobe pin</td><td>Connect the strobe to HDS1 or HDS2 and connect the remaining HDS pin to logic level 0.</td></tr></table>	Available Host Data Strobe Pins	Connections to HPI Data Strobe Pins	Host has separate read and write strobe pins, both active-low	Connect one strobe to HDS1 and the other to HDS2.	Host has one active-low strobe pin	Connect the strobe to HDS1 or HDS2, and connect the remaining HDS pin to logic level 1.	Host has one active-high strobe pin	Connect the strobe to HDS1 or HDS2 and connect the remaining HDS pin to logic level 0.		
Available Host Data Strobe Pins	Connections to HPI Data Strobe Pins											
Host has separate read and write strobe pins, both active-low	Connect one strobe to HDS1 and the other to HDS2.											
Host has one active-low strobe pin	Connect the strobe to HDS1 or HDS2, and connect the remaining HDS pin to logic level 1.											
Host has one active-high strobe pin	Connect the strobe to HDS1 or HDS2 and connect the remaining HDS pin to logic level 0.											

Table 1. Signals of the HPI (Continued)

Signal(s)	Type	Description																
HRDY	Output	<p>HPI-ready signal. This signal tells the host whether the HPI is ready for an access. When low, HRDY indicates that the HPI is busy and the host should extend the current transfer cycle. When high, HRDY indicates that the HPI has completed the data transfer and is ready for the host to continue.</p> <p>When <math>\overline{\text{HCS}}</math> goes high (inactive), HRDY is always driven high, regardless of the internal status of the HPI. For the relationships among the signals <math>\overline{\text{HDS1}}</math>, <math>\overline{\text{HDS2}}</math>, <math>\overline{\text{HCS}}</math>, and HRDY, see section 4.3 on page 8.</p>																
HCNTL0, HCNTL1	Input	<p>HPI access control signals.</p> <p>In the nonmultiplexed mode, HCNTL0 determines whether the HPI accesses the control register (HPIC) or the data register (HPID), as shown in the following table (0 = low, 1 = high). HCNTL1 is not used.</p> <table><tr><th>HCNTL0</th><th>Access Type (Nonmultiplexed Mode)</th></tr><tr><td>0</td><td>HPIC read/write</td></tr><tr><td>1</td><td>HPID read/write</td></tr></table> <p>In the multiplexed mode, HCNTL1 and HCNTL0 together select the type of register access, as shown in the following table (0 = low, 1 = high).</p> <table><tr><th>HCNTL[1:0]</th><th>Access Type (Multiplexed Mode)</th></tr><tr><td>00</td><td>HPIC read/write</td></tr><tr><td>01</td><td>HPID read/write with address auto-increment by 1</td></tr><tr><td>10</td><td>HPIA read/write</td></tr><tr><td>11</td><td>HPID read/write without address auto-increment</td></tr></table>	HCNTL0	Access Type (Nonmultiplexed Mode)	0	HPIC read/write	1	HPID read/write	HCNTL[1:0]	Access Type (Multiplexed Mode)	00	HPIC read/write	01	HPID read/write with address auto-increment by 1	10	HPIA read/write	11	HPID read/write without address auto-increment
HCNTL0	Access Type (Nonmultiplexed Mode)																	
0	HPIC read/write																	
1	HPID read/write																	
HCNTL[1:0]	Access Type (Multiplexed Mode)																	
00	HPIC read/write																	
01	HPID read/write with address auto-increment by 1																	
10	HPIA read/write																	
11	HPID read/write without address auto-increment																	
$\overline{\text{HAS}}$	Input	<p>Address strobe signal. This signal is used only in the multiplexed mode. This address strobe signal allows HCNTL[1:0], <math>\overline{\text{HBE}}[1:0]</math>, and <math>\overline{\text{HR}}/\overline{\text{W}}</math> to be removed earlier in an access cycle, which allows more time to switch bus states from address to data information. <math>\overline{\text{HAS}}</math> facilitates interfacing to multiplexed address and data type buses. Typically, an address latch enable (ALE) signal of a host is connected to <math>\overline{\text{HAS}}</math>. <math>\overline{\text{HAS}}</math> must be kept high if it is not used.</p>																
$\overline{\text{HINT}}$	Output	<p>DSP-to-host interrupt signal. <math>\overline{\text{HINT}}</math> enables the DSP to send an interrupt pulse to the host processor. The signal level is controlled by the HINT bit in status register ST3_55 of the C55x CPU (HINT = 0 means <math>\overline{\text{HINT}}</math> low; HINT = 1 means <math>\overline{\text{HINT}}</math> high).</p>																

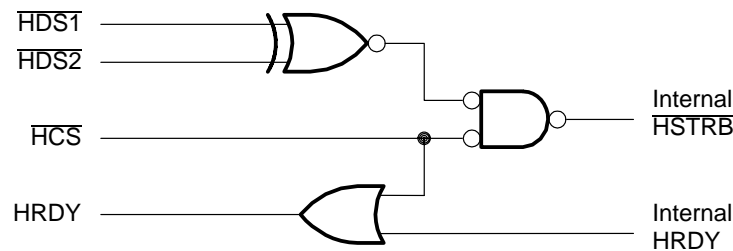
### 4.3 $\overline{\text{HDS2}}$ , $\overline{\text{HDS1}}$ , and $\overline{\text{HCS}}$ : Data Strobing and Chip Selection

As illustrated in Figure 3, the strobing logic is a function of three key inputs: the chip select pin ( $\overline{\text{HCS}}$ ) and two data strobe signals ( $\overline{\text{HDS1}}$  and  $\overline{\text{HDS2}}$ ). The internal strobe signal,  $\overline{\text{HSTRB}}$ , functions as the actual strobe signal inside the HPI.  $\overline{\text{HCS}}$  must be low (HPI selected) during strobe activity on the  $\overline{\text{HDS}}$  pins. If  $\overline{\text{HCS}}$  remains high (HPI not selected), activity on the  $\overline{\text{HDS}}$  pins is ignored.

Strobe connections between the host and the HPI depend in part on the number and types of strobe pins available on the host. Table 1 (in section 4.2) describes some options for connecting to the  $\overline{\text{HDS}}$  pins.

The  $\overline{\text{HCS}}$  input and one  $\overline{\text{HDS}}$  strobe input can be tied together and driven with a single strobe signal from the host. This technique selects the HPI and provides the strobe simultaneously. However, since  $\overline{\text{HRDY}}$  is also gated by  $\overline{\text{HCS}}$  (see Figure 3) using  $\overline{\text{HCS}}$  as a strobe limits the ability to use  $\overline{\text{HRDY}}$ . If  $\overline{\text{HCS}}$  goes high (HPI not selected),  $\overline{\text{HRDY}}$  goes high regardless of whether the current transfer is completed.

Figure 3. HPI Strobe and Select Logic



### 4.4 $\overline{\text{HBE}}[1:0]$ : Indicating Which Byte or Bytes to Access

The HPI is capable of limiting access to only one byte of a 16-bit word in memory, in HPIA, or in HPIC. Two active-low byte enable signals ( $\overline{\text{HBE}}[1:0]$ ) tell the HPI which byte or bytes should be sent to the host or modified in the DSP. Table 2 describes the effects of using the  $\overline{\text{HBE}}$  signals. Table 3 and Table 4 provide examples.

**Note:**

The  $\overline{\text{HBE}}$  signals and their associated function are not supported on the original TMS320VC5509 devices, but they are supported on the TMS320VC5509A devices.

Table 2. Effect of Driving the HPI Byte-Enable Signals in the Nonmultiplexed Mode

HBE[1:0]	Access	For Reading ...	For Writing ...
00	Word	The HPI reads a word at the specified location <sup>†</sup> and sends the word out on HD[15:0].	The HPI accepts a word from HD[15:0] and writes the word to the specified location <sup>†</sup> .
01	MSByte	The HPI reads the 8 MSBs at the specified location <sup>†</sup> and sends the byte out on HD[15:8]. HD[7:0] remain in a high-impedance state.	The HPI accepts a byte from HD[15:8] and writes the byte to the 8 MSBs at the specified location <sup>†</sup> . The 8 LSBs are not modified.
10	LSByte	The HPI reads the 8 LSBs at the specified location <sup>†</sup> and sends the byte out on HD[7:0]. HD[15:8] remain in a high-impedance state.	The HPI accepts a byte from HD[7:0] and writes the byte to the 8 LSBs at the specified location <sup>†</sup> . The 8 MSBs are not modified.
11	Reserved	-	-

<sup>†</sup> The *specified location* is any of the following: (1) a memory location at the address specified by the host, (2) the address register (HPIA), or (3) the control register (HPIC).

Table 3. Examples of Using the  $\overline{HBE}$  Signals for Host Write Cycles

16-Bit Value From Host	Byte Enable Levels		DSP Memory/Register Before Write	DSP Memory/Register After Write
	HBE1	HBE0		
B3C9h	0	0	1212h	B3C9h
2187h	0	1	0000h	2100h
4072h	1	0	BBBBh	BB72h

Table 4. Examples of Using the  $\overline{HBE}$  Signals for Host Read Cycles

16-Bit Value In DSP	Byte Enable Levels		Value Driven to Host <sup>†</sup>
	HBE1	HBE0	
B3C9h	0	0	B3C9h
2187h	0	1	21ZZh
4072h	1	0	ZZ72h

<sup>†</sup> ZZ indicates that the corresponding data bus lines are in the high-impedance state.



## 5 Nonmultiplexed Mode

As explained in section 4.1 (page 5), the HPI shares a parallel port with the EMIF, and either the full EMIF mode or the multiplexed HPI mode is automatically enabled after a DSP reset. To change the mode to the nonmultiplexed HPI mode, write 10b to bits 1-0 of the external bus selection register (EBSR).

In the nonmultiplexed mode:

- ☐ The HPI uses **separate buses for addresses and data**.
- ☐ The HPI receives 14-bit addresses from the host via the address bus (HA). For each data transfer, an address must be driven on HA. The HPI address register (HPIA) is not used.
- ☐ The HPI data register (HPID) acts as a temporary holding place for data to be transferred through the HPI. If the current access is a read, HPID contains the data that was read from the DSP memory. If the current access is a write, HPID contains the data that is to be written to the DSP memory. The DSP CPU cannot access HPID.
- ☐ HPIC contains the DSPINT bit, which enables the host to send interrupt requests to the DSP. The DSP CPU cannot access HPIC. More details about HPIC are in section 10 (page 20).

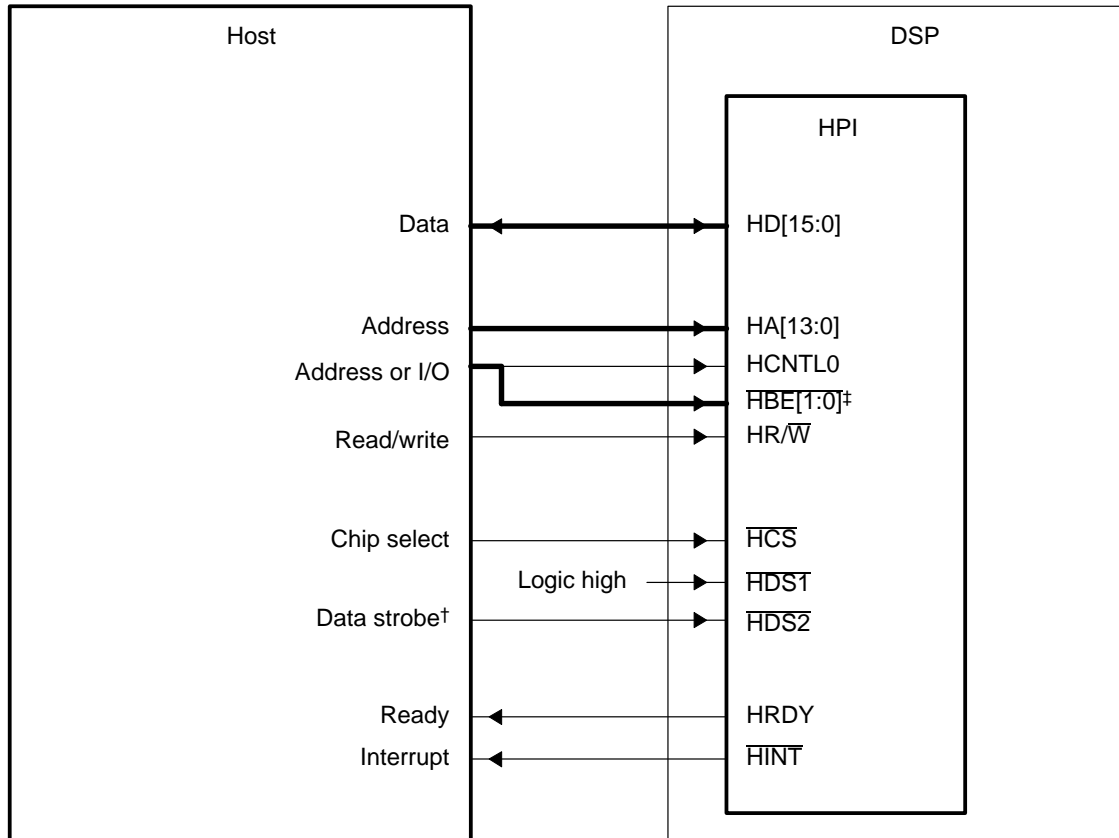
Section 5.1 provides an example of signal connections. Section 5.2 explains how the host must differentiate various accesses to HPID and HPIC. For timing diagrams of host-HPI activity, see the device-specific data manual.

### 5.1 Signal Connections in the Nonmultiplexed Mode

Figure 4 shows an example of signal connections for the nonmultiplexed mode. Details about all the HPI signals are in section 4 (page 5). Here are key points that are specific to the nonmultiplexed mode:

- ☐ Data and addresses travel on separate buses (HD and HA, respectively).
- ☐ The host indicates the cycle type with the HCNTL0 and HR/ $\overline{W}$  signals as described in section 5.2.

Figure 4. Example of Host-DSP Signal Connections in the Nonmultiplexed Mode



† For data strobing options, see section 4.3 on page 8.

‡ HBE[1:0] and their associated function are not supported on the original TMS320VC5509 devices, but they are supported on the TMS320VC5509A devices.

**Note:** The HPI shares a parallel port with the EMIF. To select the nonmultiplexed HPI mode for the port, you must write 10b to bits 1-0 of the external bus selection register (EBSR). For more information, see section 4.1 on page 5.

## 5.2 Indicating the Cycle Type in the Nonmultiplexed Mode

The host uses the HCNTL0 and HR/W pins of the HPI to indicate the cycle type. The cycle type consists of:

- ☐ The access type that the host selects by driving the appropriate level on the HCNTL0 pin. Table 5 describes the available access types for the nonmultiplexed mode. In this mode, HPIA is not used and, therefore, only HPIC accesses and HPID accesses without auto-incrementing are valid. Any HPI access with HCNTL0 driven low is an HPIC access; the value on the address bus is ignored.
- ☐ The transfer direction that the host selects with the HR/W pin. The host must drive the HR/W signal high (read) or low (write).

A summary of cycle types is in Table 6. The HPI samples the HCNTL0 and  $\overline{HR/W}$  levels at the falling edge of the internal strobe signal,  $\overline{HSTRB}$ .

**Table 5.** Access Types Selectable With the HCNTL0 Signal in the Nonmultiplexed Mode

HCNTL0	Access Type
0	HPIC access The host requests to access the control register (HPIC).
1	HPID access The host requests to access the data register (HPID).

**Table 6.** Cycle Types Selectable With the HCNTL0 and  $\overline{HR/W}$  Signals in the Nonmultiplexed Mode

HCNTL0	$\overline{HR/W}$	Cycle Type
0	0	HPIC write cycle
0	1	HPIC read cycle
1	0	HPID write cycle
1	1	HPID read cycle

## 6 Multiplexed Mode

The HPI shares a parallel port with the EMIF. To select the multiplexed HPI mode for the port, you can drive the GPIO0 pin low at the time of a DSP reset, or you can write 11b to bits 1-0 of the external bus selection register (EBSR) after reset. For more information, see section 4.1 on page 5.

In the multiplexed mode:

- ☐ **Addresses and data are carried on the same bus** (the HPI data bus, HD[15:0]). Thus, an address register (HPIA) is needed to store an address while the bus is carrying data. The HPI supports access to approximately 16K words of DARAM, each of which can be identified by a 14-bit address. However, HPIA is a 16-bit register. The host must write a 16-bit value to HPIA with the address in bits 13-0 and 0s in bits 15 and 14. The multiplexing of addresses and data on HD[15:0] means that the host must load HPIA *before* performing reads and writes to the DSP memory.
- ☐ When reading from or writing to the DSP memory, the HPI uses its data register (HPID) as a temporary holding place for the data. HPID contains the data that was read from the DSP memory (for a host read operation) or the data that is to be written to the DSP memory (for a host write operation). The DSP CPU cannot access HPID.
- ☐ HPIC contains the DSPINT bit, which enables the host to send interrupt requests to the DSP. The DSP CPU cannot access HPIC. For more details about HPIC, see section 10 on page 20.

Section 6.1 provides examples of signal connections. Section 6.2 explains how the host must differentiate various accesses to HPIA, HPID, and HPIC. Section 6.3 gives a procedure for the host to follow when loading an address to HPIA, and section 6.4 describes how that address can be automatically incremented between data transfers. For timing diagrams of host-HPI activity, see the device-specific data manual.

### 6.1 Signal Connections in the Multiplexed Mode

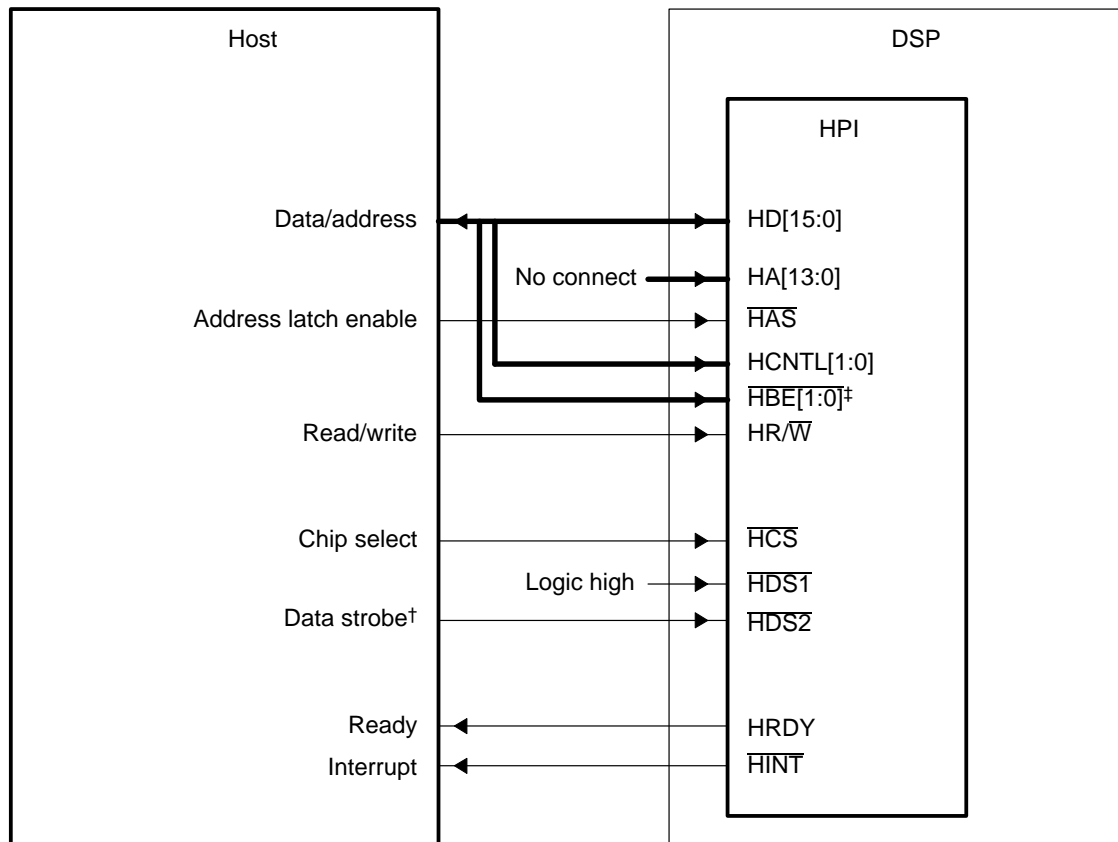
Figure 5 and Figure 6 show examples of signal connections for the multiplexed mode. In Figure 5, the address strobe signal ( $\overline{\text{HAS}}$ ) is used. In Figure 6,  $\overline{\text{HAS}}$  is tied high (not used).

Details about all the HPI signals are in section 4 (page 5). The following are key points about signals used in the multiplexed mode:

- ☐ Addresses must share the HD lines with data.
- ☐ The host indicates the cycle type with the HCNTL[1:0] and  $\text{HR}/\overline{\text{W}}$  signals as described in section 6.2.

- $\overline{HAS}$  allows HCNTL[1:0] and HR/ $\overline{W}$  to be removed earlier in an access cycle, which allows more time to switch bus states from address to data information.  $\overline{HAS}$  is an optional signal available for hosts that carry both data and addresses on a single bus. The HPI can be used without  $\overline{HAS}$ . If  $\overline{HAS}$  is not going to be used, the host must drive it high.

Figure 5. Example of Host-DSP Signal Connections When Using the  $\overline{HAS}$  Signal in the Multiplexed Mode

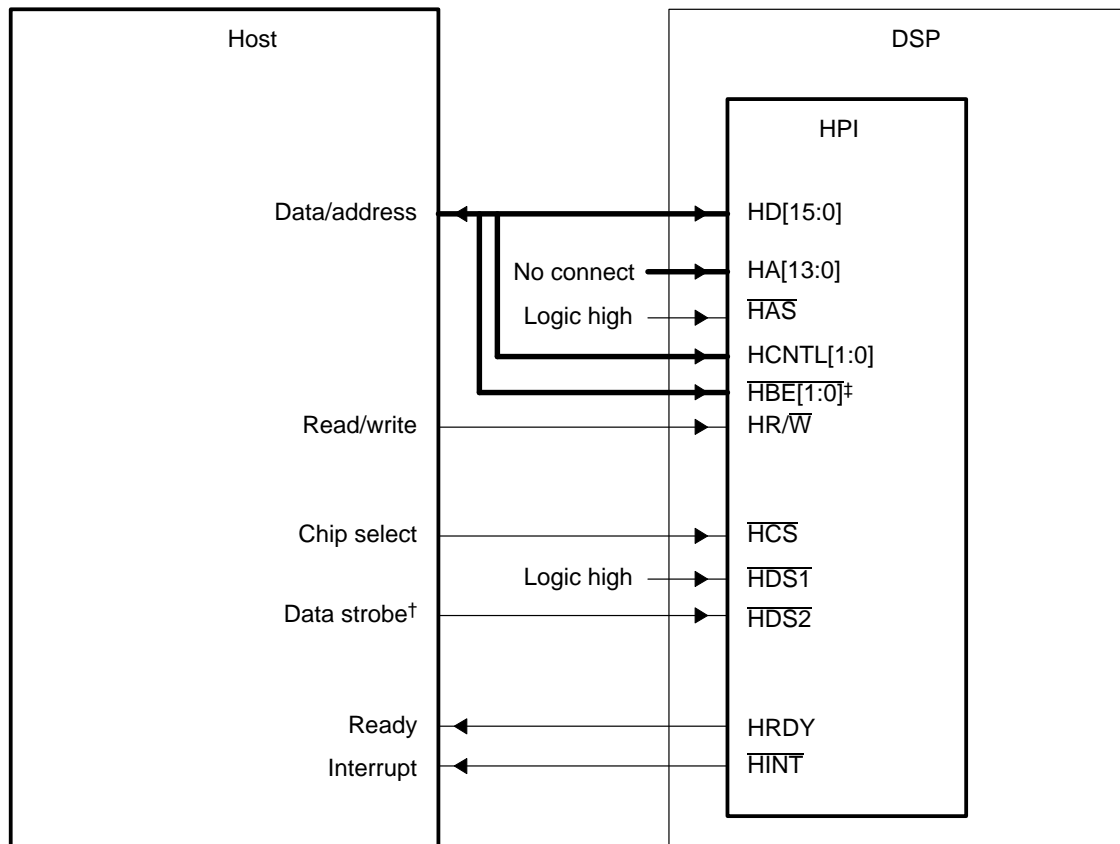


† For data strobing options, see section 4.3 on page 8.

‡  $\overline{HBE}[1:0]$  and their associated function are not supported on the original TMS320VC5509 devices, but they are supported on the TMS320VC5509A devices.

**Note:** The HPI shares a parallel port with the EMIF. To select the multiplexed HPI mode for the port, you can drive the GPIO0 pin low at the time of a DSP reset, or you can write 11b to bits 1-0 of the external bus selection register (EBSR) after reset. For more information, see section 4.1 on page 5.

Figure 6. Example of Host-DSP Signal Connections When the  $\overline{HAS}$  Signal is Tied High in the Multiplexed Mode



<sup>†</sup> For data strobing options, see section 4.3 on page 8.

<sup>‡</sup> HBE[1:0] and their associated function are not supported on the original TMS320VC5509 devices, but they are supported on the TMS320VC5509A devices.

**Note:** The HPI shares a parallel port with the EMIF. To select the multiplexed HPI mode for the port, you can drive the GPIO0 pin low at the time of a DSP reset, or you can write 11b to bits 1-0 of the external bus selection register (EBSR) after reset. For more information, see section 4.1 on page 5.

## 6.2 Indicating the Cycle Type in the Multiplexed Mode

The host uses the HCNTL[1:0] and HR/W pins of the HPI to indicate the cycle type. The cycle type consists of:

- ☐ The access type that the host selects by driving the appropriate levels on the HCNTL[1:0] pins. Table 7 describes the available access types for the multiplexed mode.
- ☐ The transfer direction that the host selects with the HR/W pin. The host must drive the HR/W signal high (read) or low (write).

A summary of cycle types is in Table 8. The HPI samples the HCNTL levels either at the falling edge of  $\overline{\text{HAS}}$  (if  $\overline{\text{HAS}}$  is used in the multiplexed mode) or at the falling edge of the internal strobe signal,  $\overline{\text{HSTRB}}$  (if  $\overline{\text{HAS}}$  is not used and is tied high).

Table 7. Access Types Selectable With the HCNTL[1:0] Signals in the Multiplexed Mode

HCNTL1	HCNTL0	Access Type
0	0	HPIC access The host requests to access the control register (HPIC).
0	1	HPID access with auto-incrementing The host requests to access the data register (HPID) and to have the memory address automatically incremented by 1 after the access.
1	0	HPIA access The host requests to access the address register (HPIA).
1	1	HPID access without auto-incrementing The host requests to access the data register (HPID) but requests no automatic post-increment of the memory address.

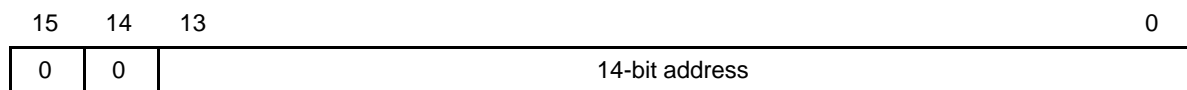
Table 8. Cycle Types Selectable With the HCNTL[1:0] and  $\overline{\text{HR/W}}$  Signals in the Multiplexed Mode

HCNTL1	HCNTL0	$\overline{\text{HR/W}}$	Cycle Type
0	0	0	HPIC write cycle
0	0	1	HPIC read cycle
0	1	0	HPID write cycle with auto-incrementing
0	1	1	HPID read cycle with auto-incrementing
1	0	0	HPIA write cycle
1	0	1	HPIA read cycle
1	1	0	HPID write cycle without auto-incrementing
1	1	1	HPID read cycle without auto-incrementing

### 6.3 Loading HPIA With an Address

The HPI supports access to approximately 16K words of DARAM (see section 2 on page 3). Each word is identifiable with a 14-bit address. When the host writes a 14-bit address to the 16-bit address register (HPIA), it must be done as follows:

- 1) Drive HCNTL1 high and drive HCNTL0 low to indicate an HPIA access.
- 2) Send a 16-bit value with the following format:



### 6.4 Auto-Increment Option: Automatic Address Increment Between Transfers

If the host is reading and/or writing at random addresses, it must write to HPIA before each data transfer. However, if the host performs accesses at sequential addresses and the HPI is in its multiplexed mode, it can reduce the required number of cycles by using the address auto-increment option: Drive HCNTL1 low and HCNTL0 high for auto-incremented data accesses. When auto-incrementing is used, the host needs to write only the start address to HPIA; for each subsequent HPID access, the address in HPIA is automatically incremented by 1.

When using the auto-increment option, keep the following important points in mind:

- ☐ The address increment occurs only after a host cycle in which the  $\overline{\text{HBE1}}$  signal is asserted (that is, when a full word is accessed or the high byte of a word is accessed). If reading or writing a word one byte at a time, the host must access the low byte first. Accessing the high byte second ensures that the increment occurs after both bytes have been transferred.
- ☐ Although the internal address is automatically incremented, a host read of HPIA always returns the last address written to HPIA by the host.



## 7 Interrupts Between the Host and the DSP

By modifying special interrupt bits, the host and the DSP can send interrupt requests to each other.

### 7.1 Sending an Interrupt Request From the Host to the DSP

You can have the host send an interrupt request to the DSP as follows:

- 1) Make sure the HPI is configured to write to the HPI control register (HPIC).  
In the nonmultiplexed mode of the HPI, the HCNTL0 signal must be held low to select HPIC. In the multiplexed mode of the HPI, HCNTL1 and HCNTL0 must both be held low to select HPIC.

- 2) Write a 1 to bit 1 (DSPINT) of HPIC.

Setting this bit causes the DSP to set the DSPINT flag bit in the CPU. If this maskable interrupt is properly enabled in the CPU, the CPU fetches the DSPINT interrupt vector and branches to the corresponding interrupt service routine.

Whenever DSPINT is read, 0 is returned.

### 7.2 Sending an Interrupt Request From the DSP to the Host

The DSP can send an interrupt request to the host by clearing and then setting the HINT bit in status register ST3\_55 of the CPU. A change to the HINT bit changes the level of the  $\overline{\text{HINT}}$  output signal of the HPI. If the DSP writes a 0 to the HINT bit,  $\overline{\text{HINT}}$  goes low (active). If the DSP writes a 1 to the HINT bit,  $\overline{\text{HINT}}$  goes high (inactive). Thus, the interrupt pulse width is managed by software.

Although the  $\overline{\text{HINT}}$  signal can be used to interrupt the host, there is no direct acknowledgement path from the host back to the DSP. If desired, the host can acknowledge the interrupt by writing to a memory location that is common to the host and the DSP.

During a DSP reset, the CPU sets the HINT bit and  $\overline{\text{HINT}}$  goes high (inactive).

## 8 Boot Loading With the HPI

The HPI can be used to load application code to the internal DARAM of the DSP. After reset, the host can load the desired application code into the memory space of the DSP through the HPI. After the code has been loaded, the host can cause the DSP to begin execution of the loaded code. For details, see the application report *Using the TMS320VC5509/C5509A Bootloader* (SPRA375).

## 9 Power, Emulation, and Reset Considerations

### 9.1 HPI and the IDLE Instruction

The DSP is divided into idle domains that can be programmed to be idle or active upon execution of the IDLE instruction in CPU. To reduce power consumption, you might want to turn off certain idle domains. The HPI does not belong to any of the idle domains and cannot be placed in an idle mode. However:

- ☐ If the clock generator idle domain or the DMA idle domain is idle, the host cannot access the DSP memory.
- ☐ The DSP contains a host mode idle (HIDL) bit in the external bus selection register (EBSR). If the HIDL bit is 0, the clock generator cannot be placed into its idle mode; it remains active for the sake of the HPI. If the HIDL bit is 1, the clock generator can be made idle. For more details about EBSR and the use of the HIDL bit, see the device-specific data manual.

### 9.2 HPI Emulation Modes

The HPI relies on the operation of the DMA controller to move data to/from the memory of the DSP. As a result, when emulation activity affects the DMA controller, it also affects the ability of the HPI to access memory. The FREE bit of the DMA controller determines how the DMA controller reacts to an emulation breakpoint or other emulation halt:

- ☐ If FREE = 0 (the reset value), a breakpoint or other emulation halt suspends DMA transfers.
- ☐ If FREE = 1, DMA transfers are not interrupted by a breakpoint or other emulation halt.

The FREE bit is bit 2 of the DMA global control register. This register is described in the *TMS320VC5509/5510 Direct Memory Access (DMA) Controller Reference Guide* (SPRU587).

### 9.3 Effects of a DSP Reset on the HPI

If you drive the DSP reset signal low, the DSP undergoes a reset. The control register (HPIC) is forced to its default value (see Figure 9 in section 10). The address register (HPIA) and the data register (HPID) are not initialized by a DSP reset.

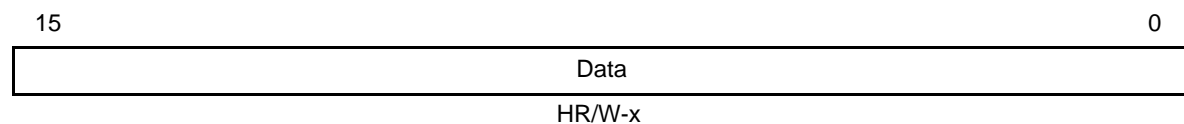
## 10 HPI Registers

The host port interface (HPI) contains three registers (see the following sections) that a host can use to access the memory of the DSP. The registers share a data bus; therefore, the host must drive the signals HCNTL1 and/or HCNTL0 to the appropriate levels to indicate which HPI register the host is to access. The DSP can neither read from nor write to these registers.

### 10.1 Data Register (HPID)

As shown in Figure 7, HPID is a 16-bit register. This register acts as a temporary holding place for data to be transferred through the HPI. HPID contains the data that was read from the DSP memory if the current access is a read, or the data that is to be written to the DSP memory if the current access is a write.

Figure 7. Data Register (HPID)

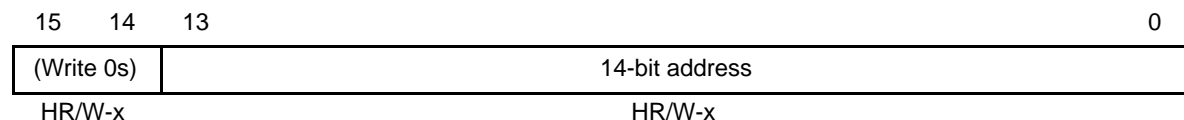


**Legend:** HR/W = Host read/write access; -x = Value not defined after DSP reset  
(The DSP cannot access HPID)

### 10.2 Address Register (HPIA)

In the multiplexed mode of the HPI (see section 6 on page 13), this 16-bit register acts as a temporary holding place for a 14-bit address for a read or write operation. As shown in Figure 8, when the host writes to HPIA, it must write 0s to bits 15-14 and must write the 14-bit address to bits 13-0. In the nonmultiplexed mode of the HPI (see section 5 on page 10), HPIA is not needed because the address is directly available on input signals HA[13:0].

Figure 8. Address Register (HPIA)

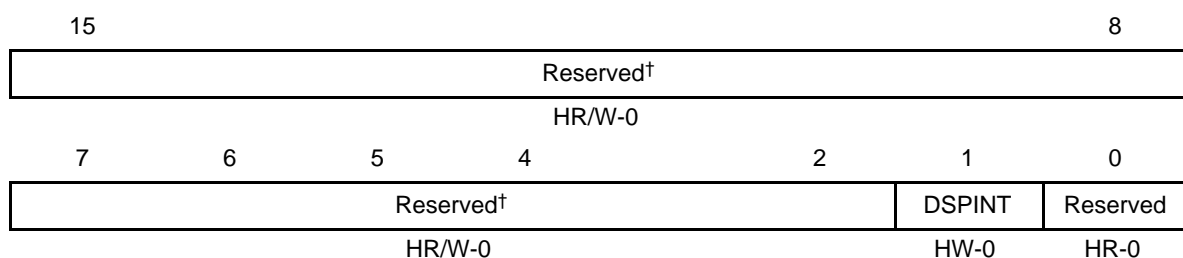


**Legend:** HR/W = Host read/write access; -x = Value not defined after DSP reset  
(The DSP cannot access HPID)

### 10.3 Control Register (HPIC)

HPIC provides the DSPINT bit, which enables the host to interrupt the DSP by generating an interrupt request to the DSP CPU. The fields of HPIC are shown in Figure 9 and described in Table 9.

Figure 9. Control Register (HPIC)



**Legend:** HR/W = Host read/write access; HW = Host write-only access; -n = Value after DSP reset  
(The DSP cannot access HPIC)

<sup>†</sup> Always write 0s to these reserved bits. The read states of these reserved bits are not defined.

Table 9. Control Register (HPIC) Field Descriptions

Bit	Field	Value	Description
15-2	Reserved	0	Always write 0s to these bits. The read states of these bits are not defined.
1	DSPINT	0	Writing 0 to DSPINT has no effect. This bit always reads as 0.
		1	Writing 1 to DSPINT causes the HPI to send an interrupt request to the DSP CPU.
0	Reserved		The read state of this read-only bit is not defined.

## A

- access control signals (HCNTL0 and HCNTL1)
  - summary description 7
  - use of HCNTL[1:0] in multiplexed mode 15
  - use of HCNTL0 in nonmultiplexed mode 11
- access types in multiplexed mode (table) 16
- access types in nonmultiplexed mode (table) 12
- address bus (HA) 5
- address register (HPIA)
  - description 20
  - load procedure 17
  - use in multiplexed mode 13
- address strobe signal ( $\overline{HAS}$ )
  - summary description 7
  - use in multiplexed mode 13
- auto-increment option 17

## B

- block diagram showing role of HPI 2
- boot loading with HPI 18
- byte-enable signals ( $\overline{HBE}[1:0]$ )
  - summary description 6
  - used to indicate which byte(s) to access 8

## C

- chip select signal ( $\overline{HCS}$ )
  - how to use for chip selection 8
  - summary description 6
- chip selection 8
- control register (HPIC)
  - description 21
  - use in multiplexed mode 13
  - use in nonmultiplexed mode 10

- control signals HCNTL0 and HCNTL1
  - summary description 7
  - use of HCNTL[1:0] in multiplexed mode 15
  - use of HCNTL0 in nonmultiplexed mode 11
- cycle types in multiplexed mode 15
- cycle types in nonmultiplexed mode 11

## D

- data bus (HD) 5
- data register (HPID)
  - description 20
  - use in multiplexed mode 13
  - use in nonmultiplexed mode 10
- data strobe signals ( $\overline{HDS1}$  and  $\overline{HDS2}$ )
  - how to use for data strobing 8
  - summary description 6
- data strobing and chip selection 8
- diagram showing role of HPI 2
- DSP memory accessible through HPI 3
- DSP reset, effects on HPI 19
- DSP sending/receiving interrupts via HPI 18
- DSP-to-host interrupt signal ( $\overline{HINT}$ )
  - summary description 7
  - used for DSP-to-host interrupts 18
- DSPINT bit of HPIC
  - described in table 21
  - shown in figure 21

## E

- EMIF and HPI sharing pins 5
- emulation modes 19

## F

- figure showing role of HPI 2

**H**

HA bus 5

$\overline{\text{HAS}}$  signal

- summary description 7
- use in multiplexed mode 13

$\overline{\text{HBE0}}$  and  $\overline{\text{HBE1}}$  signals

- summary description 6
- used to indicate which byte(s) to access 8

HCNTL0 and HCNTL1 signals

- summary description 7
- use of HCNTL[1:0] in multiplexed mode 15
- use of HCNTL0 in nonmultiplexed mode 11

$\overline{\text{HCS}}$  signal

- how to use for chip selection 8
- summary description 6

HD bus 5

$\overline{\text{HDS1}}$  and  $\overline{\text{HDS2}}$  signals

- how to use for data strobing 8
- summary description 6

$\overline{\text{HINT}}$  signal

- summary description 7
- used for DSP-to-host interrupts 18

host processor sending/receiving interrupts via HPI 18

host-to-DSP interrupt request bit (DSPINT)

- described in table 21
- shown in figure 21

HPI affected by the states of idle domains 19

HPI and EMIF sharing pins 5

HPI in host-DSP block diagram 2

HPI-DMA interaction 4

HPI-ready signal (HRDY)

- relationship to chip select signal ( $\overline{\text{HCS}}$ ) 8
- summary description 7

HPIA

- description 20
- load procedure 17
- use in multiplexed mode 13

HPIC

- description 21
- use in multiplexed mode 13
- use in nonmultiplexed mode 10

HPID

- description 20
- use in multiplexed mode 13
- use in nonmultiplexed mode 10

HR/ $\overline{\text{W}}$  signal

- summary description 6
- use in multiplexed mode 15
- use in nonmultiplexed mode 11

HRDY signal

- relationship to chip select signal ( $\overline{\text{HCS}}$ ) 8
- summary description 7

$\overline{\text{HSTRB}}$  signal 8

**I**

idle domains, effects on HPI 19

indicating cycle type in multiplexed mode 15

indicating cycle type in nonmultiplexed mode 11

interaction between DMA controller and HPI 4

internal  $\overline{\text{HSTRB}}$  signal 8

interrupt signal ( $\overline{\text{HINT}}$ )

- summary description 7
- used for DSP-to-host interrupts 18

interrupts between host and DSP 18

introduction to HPI 2

**L**

loading HPIA 17

**M**

memory accessible through HPI 3

multiplexed mode 13

**N**

nonmultiplexed mode 10

**P**

pins/signals of HPI 5

power reduction, effects on HPI 19

**R**

read/write signal (HR/ $\overline{\text{W}}$ )

- summary description 6
- use in multiplexed mode 15
- use in nonmultiplexed mode 11

ready signal (HRDY)  
    relationship to chip select signal ( $\overline{HCS}$ ) 8  
    summary description 7  
reducing power consumption, effects on HPI 19  
registers of HPI 20  
reset, effects on HPI 19

## **S**

sharing pins with EMIF 5  
signal connections in multiplexed mode 13  
signal connections in nonmultiplexed mode 10

signals/pins of HPI 5  
strobe and select logic (figure) 8  
strobe signal  $\overline{HAS}$   
    summary description 7  
    use in multiplexed mode 13  
strobe signals  $\overline{HDS1}$  and  $\overline{HDS2}$   
    how to use for data strobing 8  
    summary description 6