

TMS320VC5510 HPI Throughput and Optimization

Michelle Richardson

DSP Systems/Software Applications

ABSTRACT

This application report summarizes some characteristics relevant to understanding TMS320VC5510 host port interface (HPI), and proposes recommendations to optimize throughput on the HPI peripheral.

Contents

1	Introduction	2
2	Understanding HPI Throughput	2
2.1	Best-Case Timing Review	2
2.1.1	Multiplexed Read Data Transfer Timing	2
2.1.2	Multiplexed Write Data Transfer Timing	3
3	Factors That Influence HPI Throughput	4
3.1	Characteristics of CPU, DMA HPI Accesses	4
3.1.1	Effects of DMA and CPU Activity on HPI Throughput	5
3.1.2	Host Considerations	5
4	Recommendations for HPI Throughput Optimization	5
5	Performing Your Own System Analysis/Optimization	6
6	References	6

List of Figures

Figure 1	Multiplexed Read Timing With Auto-Increment	3
Figure 2	Multiplexed Write Timing With Auto-Increment	4

1 Introduction

The C5510 host port interface (HPI) is a 16-bit wide parallel interface that allows a host processor to have direct access to portions of the DSP memory map. Timing diagrams contained in the data manual address best-case timing, but give little insight into system timing when latencies are introduced.

HPI latency is dependent on several factors, including other DMA channel and CPU activity, memory utilization, and latencies introduced by the host processor. This can pose a challenge for users who want to estimate HPI average throughput or worst-case latency for a system with significant DMA or activity. Since DMA is typically used to transfer HPI data to and from other memory areas for processing, or to and from other peripherals, it is necessary to understand this interaction when HPI throughput is a high priority. In systems where maximizing HPI throughput is less of a concern, an understanding of this interaction will help the user avoid unwanted system performance degradation.

This application note reviews HPI best-case timing, summarizes some characteristics of CPU, HPI and DMA accesses that are relevant to understanding how to optimize HPI throughput and suggests some ways to maximize HPI throughput.

2 Understanding HPI Throughput

2.1 Best-Case Timing Review

The timing diagrams contained in the data sheet indicate best-case throughput of the C5510 HPI. This timing assumes that there is no DMA or CPU activity on the DSP to that same port, and assumes no host wait states or latencies beyond the timing requirements as specified in the data sheet. For example, in the case of a write access, there may be a delay between rising HRDY and when the host processor drives the data strobe low for the next transfer.

This section provides a review of HPI timing based on simplified versions of the timing diagrams contained in the data manual. Only multiplexed mode, in which the data and address information is multiplexed on the data pins, is covered here for simplicity, but the concept is the same. The user should consult the *TMS320VC5510 Fixed-Point Digital Signal Processor* (SPRS076) for the complete set of HPI timing diagrams.

2.1.1 Multiplexed Read Data Transfer Timing

There are two main signals relevant to understanding HPI throughput, $\overline{\text{HDS}}$ and HRDY. These signals are shown in Figure 1 below, along with the data bus for reference. Minimum timing requirements or characteristics are shown, along with the reference designator given in the data manual.

The $\overline{\text{HDS}}$ signal is the data strobe output from the host. It has a minimum high and low pulse width requirement of four cycles each, designated E15, E16 in Figure 1.

An additional HPI timing characteristic, which is designated E2 in the figure, gives the minimum time from $\overline{\text{HDS}}$ low to read data valid. Since E2 overlaps with E15, we can add E2 and E16 to calculate the minimum time between when the data strobe goes low and HRDY goes high, or the time it takes to complete a word transfer. HRDY is the output signal from the DSP that indicates its internal status to the host, and when it returns to the high state, the DSP data is ready to be read by the host.

Best-case read timing can be calculated as follows:

$$\begin{aligned} \text{Transfer time per data word} &= E2 + E16 \\ &= 14P + 10\text{ns} + 4P, \text{ where } P \text{ is } 1/\text{CPU frequency in seconds} \\ &= 18P + 10\text{ns} \end{aligned}$$

At 200MHz (P=5ns) the time it would take 20 cycles per word, not including overhead required to set up the HPIC and HPIA registers, or the timing of the address strobe signal, $\overline{\text{HAS}}$. If $\overline{\text{HAS}}$ is used, an additional delay of 4ns is required. In auto-increment mode, the four-cycle delay due to $\overline{\text{HAS}}$ impacts only the first word transfer, as it overlaps with the high pulse of HDS during subsequent transfers.

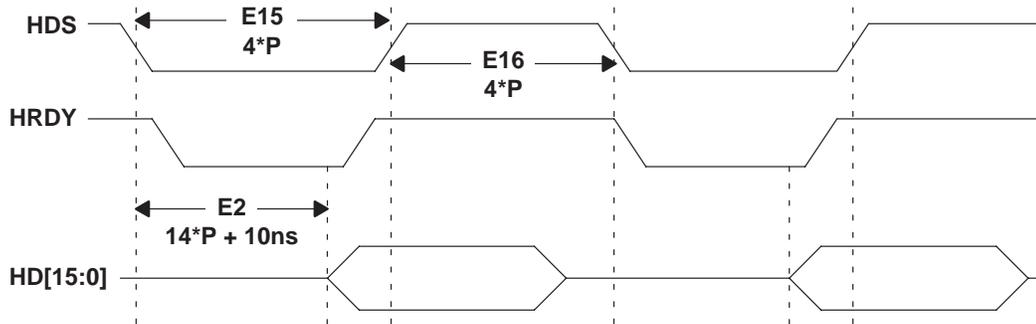


Figure 1. Multiplexed Read Timing With Auto-Increment

DMA activity on the DSP affects throughput by increasing the latency of HRDY.

2.1.2 Multiplexed Write Data Transfer Timing

For write timing, the HDS and HRDY signals are also relevant to understanding best-case throughput. The $\overline{\text{HDS}}$ signal retains the minimum high and low pulse width requirement of four cycles each, designated E15, E16 in Figure 2. An additional HPI timing characteristic, which is designated E10 in the figure, gives the minimum time from $\overline{\text{HDS}}$ high HRDY high. Since E10 overlaps with E16, we add E10 and E15 to calculate the time between when HDS goes low to when HRDY goes high, or the time to complete one write transfer.

We can calculate the time to perform a one-word write transfer as follows:

$$\begin{aligned} E15 + E10 &= 4P + 14P + 10\text{ns}, \text{ where } P \text{ is } 1/\text{CPU frequency in seconds} \\ &= 18P + 10\text{ns} \end{aligned}$$

So, at 200MHz (P=5ns) a write transfer would take 20 cycles per word. As with read transfers, if the $\overline{\text{HAS}}$ signal is used, it will introduce an additional 4ns to the total transfer time.

In a write transfer with auto-increment, subsequent transfers can begin before the HRDY signal returns high from the previous write. Since the next HDS cycle can begin while HRDY is still low, the write transfer time may be reduced by up to four cycles per word after the initial transfer, as highlighted in Figure 2.

Again, DMA or CPU activity can extend the latency of HRDY.

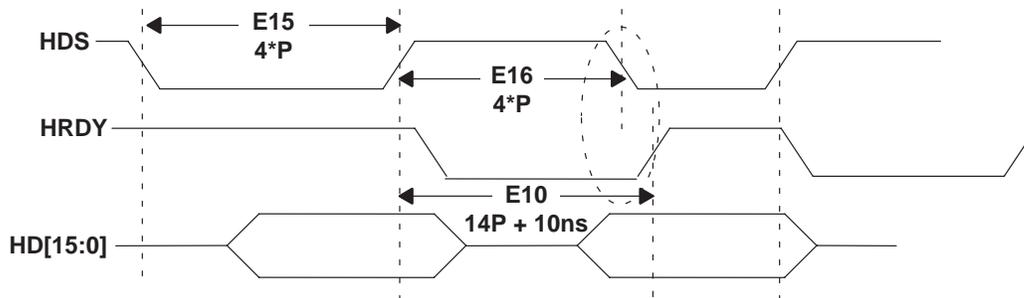


Figure 2. Multiplexed Write Timing With Auto-Increment

3 Factors That Influence HPI Throughput

3.1 Characteristics of CPU, DMA HPI Accesses

A summary of some characteristics of HPI, DMA and CPU accesses is useful in order to understand how the CPU and DMA activity may increase HRDY latency, thereby impacting HPI throughput.

For more detailed information on the DMA and the HPI, please consult the *TMS320VC5510 DSP Host Port Interface (HPI) Reference Guide* (SPRU588) and the *TMS320VC5503/5507/5509/5510 Direct Memory Access(DMA) Controller Reference Guide* (SPRU587). The internal memory architecture is described in the *TMS320VC5510 Fixed-Point Digital Signal Processor* (SPRS076).

- The HPI is supported by an auxiliary port in the DMA controller. The HPI is an additional channel on the DMA service chain, and subject to the same arbitration as other DMA channels. The HPI is the last channel in the sequence on the service chain.
- HPI latency varies due to the number of DMA channels that are active on the same port, their priorities, and activity level.
- The HRDY signal is an output from the DSP that indicates when the internal DMA portion of the current transaction has been completed. For read transfers, HRDY indicates when the read data is ready to be read by the host. When HRDY goes high you still have to terminate the read transaction (HDS high) and wait four cycles (E16) before starting the next read.] Use of the HRDY signal is highly recommended for optimum HPI throughput.
- If the HPI and DMA channels request access to the same memory port (DARAM, SARAM, EMIF) at the same time, those accesses will be subject to arbitration on the service chain.
- Internal DARAM and SARAM are both are divided into 8K-byte blocks. Each 8K-byte block of internal DARAM can support two accesses from the CPU or the DMA per clock cycle. Each 8K-byte block of SARAM can support one access per clock cycle.
- The CPU has higher priority than the DMA/HPI, in the event of simultaneous accesses to the same 8K-byte block of internal RAM, if the memory cannot support both in the same clock cycle.
- The priority of the HPI in the DMA service chain is programmable via the EHPI_PRIO bit in the DMA_GCR register. A high priority setting does not elevate the HPI above other high priority DMA channels.

- The HPI may be set to have exclusive access to DSP internal memory by setting the EHPI_EXCL bit in the DMA_GCR register. This locks out the other DMA channels by denying them access to that memory.
- Each DMA channel has an internal FIFO that it uses to buffer transfer data between the source and destination port. This can serve to minimize latencies due to accesses to the same port or block of memory.
- Relative clock speeds of the host and DSP will affect impact of DMA activity on HPI accesses.

3.1.1 *Effects of DMA and CPU Activity on HPI Throughput*

Based on the characteristics described in the previous section, it is easier to understand the impact of the DMA on HPI throughput. For example, if both a host and a DMA channel access the same 8K byte block of DARAM, CPU accesses to that block, impact on HPI throughput will be minimal because DARAM allows two accesses per clock cycle. Additional DMA accesses to the same block of internal DARAM would increase latency. The amount of the delay would be dependent on the frequency of the HPI and DMA accesses, and the priorities of the respective channels relative to the HPI priority.

Since the CPU has priority over any DMA or HPI accesses, continuous CPU data or program accesses to this block can have the effect of starving HPI and/or DMA activity. This is usually observed in terms of the HRDY signal staying low for long periods of time, even when DMA activity has been disabled.

SARAM can satisfy a single access per clock cycle, so concurrent HPI and DMA accesses to the same block would have a greater affect on system throughput relevant to the HPI.

3.1.2 *Host Considerations*

Internal host latencies, with respect to processing or driving any of the pins at the interface, may affect system throughput via the C5510 HPI. For example, the DSP uses the HRDY signal as an indicator to the host that it is ready for the next transfer. Any delay in the host processing and acting upon that signal will be reflected in overall throughput. Wait states imposed on the host processors parallel interface may also affect HPI throughput.

4 **Recommendations for HPI Throughput Optimization**

The following recommendations should be considered in optimizing HPI throughput:

- Do not place program code in same 8K byte block of memory as the HPI buffer(s), since the CPU will take priority until those accesses are complete.
- Use the HRDY signal to gate transfers to avoid the need to assume worst-case latency.
- Minimize CPU or DMA accesses to the same 8K byte internal memory block containing an HPI buffer, at the same time the HPI access are occurring.
- If DMA is used to move data to or from an HPI buffer, use ping-pong buffers to minimize concurrent HPI and DMA accesses to the same block.
- When HPI throughput is a priority, and simultaneous HPI and DMA activity is required of the same 8K byte block, place the HPI buffer in DARAM to allow two accesses per cycle. Note

that the trade-off will be a reduction in DARAM space available for time-critical code and data.

- Place other DMA buffers in different type of internal memory, to minimize DMA accesses to the same port.
- Use the host interrupt (HINT) signal as an alternative to the host polling a status flag in DSP memory. If polling must be used to test status, such as available data, allow time between accesses.
- Use smaller HPI transfer burst sizes to allow DMA to use the overhead time to complete its transfers. This will also take best advantage of the internal DMA FIFOs.
- Set the HPI to a higher priority than as many DMA channels as possible.
- Plan the placement of buffers, code and other data carefully. Don't rely on the linker to do the placement for you.

5 Performing Your Own System Analysis/Optimization

Since overall HPI throughput is dependent on host characteristics, as well as DMA and CPU activity, users concerned with optimizing HPI throughput beyond the scope of this document should evaluate it by modeling and characterizing their system using hardware.

6 References

1. *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076)
2. *TMS320VC5510 DSP Host Port Interface (HPI) Reference Guide* (SPRU588)
3. *TMS320VC5503/5507/5509/5510 Direct Memory Access(DMA) Controller Reference Guide* (SPRU587)
4. *Using the TMS320VC5509/5510 Enhanced HPI* (SPRA741)

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 the 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.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265