TEXAS INSTRUMENTS

Application Report
SCPA030 - June 2000

Interfacing PCI2040 to TMS320VC5420 DSP

Mike Campbell *PCIBus Solutions*

ABSTRACT

This application note describes how system designers can interface the TMS320VC5420 16-bit HPI (C5420) to the PCI bus, using the TMS320C6x™ HPI (C6x) mode interface on the PCI2040 PCI-to-DSP bridge. The note specifies two modes for mapping the C5420 to host memory, depending on the number of C5420s. The note also describes both hardware and software considerations, provides an example schematic, and includes other illustrative figures.

Contents

List of Figures

1 Introduction

PCI2040 is a PCI-to-DSP bridge designed to interface to either the 8-bit TMS320C54x HPI port or the 16-bit TMS320C6x HPI (C6x) port. This application note describes a method for interfacing the TMS320VC5420 16-bit HPI (C5420) to the PCI bus, using the 16-bit C6x mode interface on the PCI2040.

2 Mapping C5420 to Host Memory

There are two different ways of mapping the C5420 into host or PC memory space, depending on the number of C5420s.

- Use memory map 0 with two or less C5420s. This method maps each subsystem or core of the c5420 into a dedicated memory space. Figure 1 illustrates memory map 0.

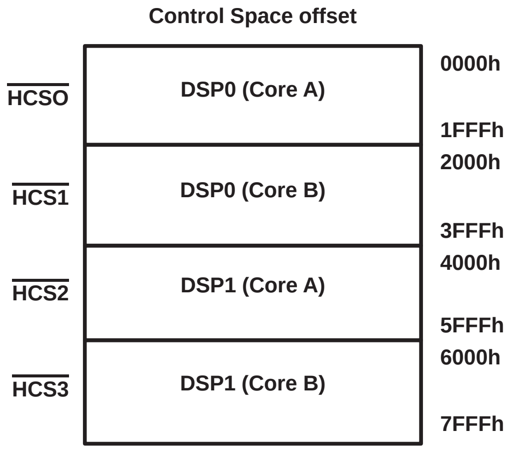


Figure 1. Memory Map 0

- Use memory map 1 with more than two C5420s. This method maps each DSP to a specific memory space, and each core or subsystem of the DSP shares the same memory space. Figure 2 illustrates memory map 1.

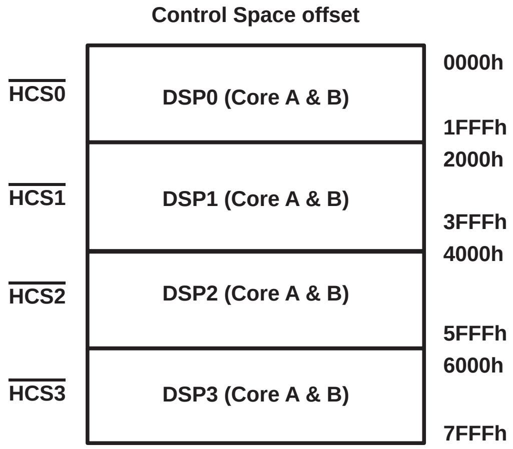


Figure 2. Memory Map 1

NOTE: The Control space offset is the offset from address represented in the control space base address register located at PCI offset 14h in the PCI2040.

3 Hardware Considerations

The C5420 is a dual CPU device with two independent c54x subsystems capable of core-to-core communication. Although the PCI2040 was not designed specifically to interface to a dual CPU device, it functions properly with such a device with special considerations:

- Converting a c6201 HPI protocol to a c5420 HPI-16 protocol
- Switching between the two DSP cores
- Handling multiple resets and interrupts

3.1 C6x HPI Protocol to C5420 HPI-16 Protocol

The C5420 has two modes of operation for the HPI bus:

- Non-multiplexed mode (HMODE=1): the C5420 HPI interface has a 18-bit address bus used to access all internal memory.
- Multiplexed mode (HMODE=0): the address and data are shared or multiplexed on the same pins. The PCI2040 only implements multiplexed mode, so the HMODE pin on the C5420 must be tied or pulled down to GND.

You must add additional logic for compatibility to use the PCI2040’s 16-bit HPI to connect the C5420 to the PCI bus. The C6x HPI has an active low HRDY signal, and the C5420 has an active high HRDY signal, so you must use an inverter on all implemented C5420s. Select a 3.3V inverter like the SN74ALCV04 so VCC_H can be tied directly to VCC.

3.2 Switching Between Two DSP Cores

Use the SELA/B pin on the C5420 to select which DSP subsystem to access through the HPI interface. Connect the PCI2040’s GPIO like GPIO2 to the SELA/B pin. You can toggle this pin either low or high to select core A or core B.

3.3 Multiple HRST

There are two C5420 methods for controlling the reset function of each subsystem.

- Holding A_$\overline{\text{RS}}$ and B_$\overline{\text{RS}}$ low while the $\overline{\text{HPIRS}}$ pin transitions from low to high keeps both cores in reset while allowing you to download application code into the DSP. After the download, you can drive A_$\overline{\text{RS}}$ and B_$\overline{\text{RS}}$ high to take the two subsystems out of reset.
- Holding A_$\overline{\text{RS}}$ and B_$\overline{\text{RS}}$ high while the $\overline{\text{HPIRS}}$ pins transitions from low to high holds both cores in reset while allowing you to download code in the DSP. You can take a specific subsystem out of reset by performing HPI write to address 2Fh to release that subsystem from reset. You can change the value of the SELA/B pin and write to the other subsystem to release that subsystem from reset. This method is recommended when using the PCI2040.

You can use power-on reset logic to reset each subsystem (A_$\overline{\text{RS}}$ and B_$\overline{\text{RS}}$). Connect the $\overline{\text{HPIRS}}$ pin to the $\overline{\text{HRST}}$ pin on the PCI2040. You must ensure that the poweron reset logic deasserts reset to each subsystem before host software deasserts $\overline{\text{HRST}}$ to the DSP. Remember to take each subsystem of the DSP out of reset by performing a HPI write to address 2Fh after the application is downloaded.

3.4 Handling Two $\overline{\text{HINT}}$ Interrupts From One DSP

Handling multiple interrupts depends on how many C5420s are located behind the PCI2040.

- If there are two or less C5420s, you can connect $\overline{\text{HINT0}}$ to A_$\overline{\text{HINT}}$ and $\overline{\text{HINT1}}$ to B_$\overline{\text{HINT}}$ to the first C5420. You can connect the second C5420 interrupts (A_$\overline{\text{HINT}}$ and B_$\overline{\text{HINT}}$) to PCI2040's $\overline{\text{HINT2}}$ and $\overline{\text{HINT3}}$, respectively (see Figure 3). This method corresponds with memory map 0.

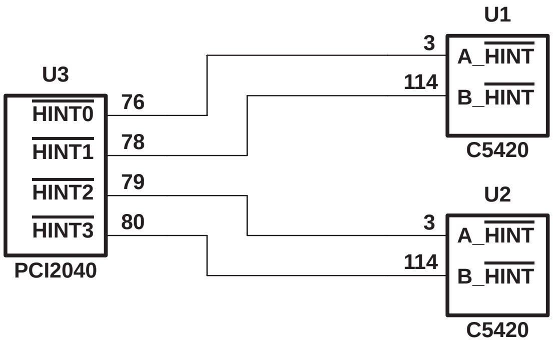


Figure 3. Two or Less C5420s Behind PCI2040

- If there are more than two C5420s, you need to AND the two interrupts on the DSP together as illustrated in Figure 4. Each interrupt on the DSP shares the same interrupt line on the PCI2040, so you have no way of knowing which subsystem asserted $\overline{\text{HINT}}$, and you must write a *1* to the $\overline{\text{HINT}}$ bit of the HPIC register for each subsystem. This method corresponds to memory map 1.

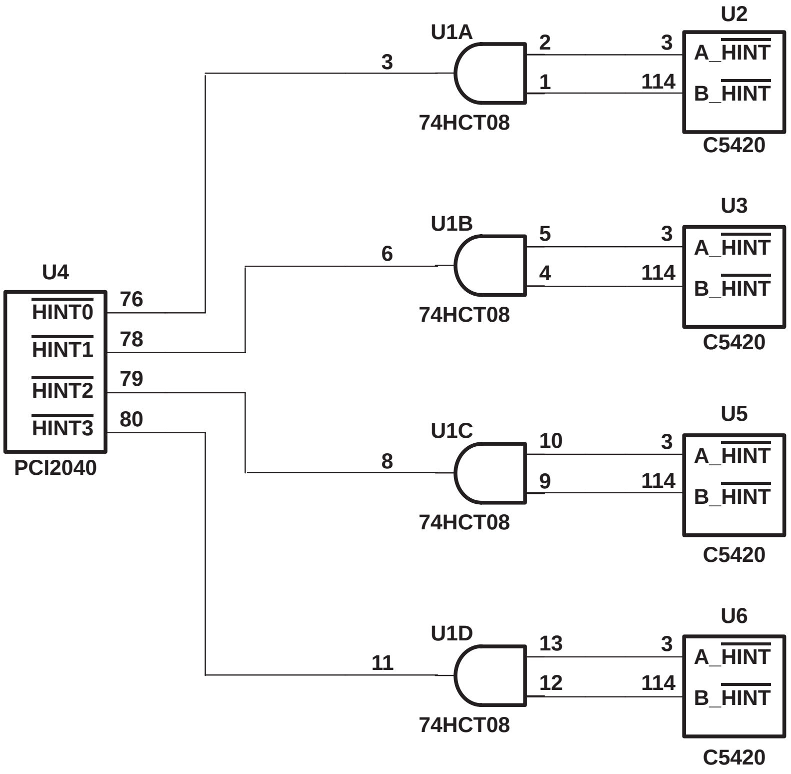


Figure 4. More Than Two C5420s Behind PCI2040

4 Software Considerations

The C6x has a 16-bit HPI, but it uses words that are 32 bits wide, so all HPI transactions consist of two back to back HPI cycles. Since the PCI2040 16-bit interface was designed to interface to C6x(s), the PCI2040 will also do back-to-back HPI cycles. Unlike the C6x, the C5420 uses 16-bit words that require only one HPI cycle to complete a transaction. Programmers writing software to access a C5420 through the PCI2040 need to be aware that reading and writing to the DSP require special access routines.

4.1 Writing From HPIA, HPIC, and HPID Registers

When writing to the HPIA (HPI address) register, the HPIC (HPI control) register, and the HPID (HPI data) register (in non-autoincrement mode), remember that the PCI2040 will do two back-to-back cycles to the HPI. Both cycles are to the same register, so the data written during the first cycle is over written during the second cycle. You should always transmit data written to these registers in bytes two and three on the PCI bus to ensure the correct data is written (see Figure 5).

31 24	23 16	15 8	7 0
Byte 1	Byte 0	Don’t Care	Don’t Care

Figure 5. PCI Data During a Write to C5420

4.2 Reading From HPIA, HPIC, and HPID Registers

When reading to the HPIA register, the HPIC register, and the HPID register (in non-autoincrement mode), the PCI2040 read these registers twice and returns the value in both the upper and lower half of the PCI doubleword (see Figure 6).

31 24	23 16	15 8	7 0
Byte 1	Byte 0	Byte 1	Byte 0

Figure 6. PCI Data During a Read From C5420

4.3 Using the Autoincrement Feature

To maximize the HPI performance, make all reads and writes to the HPID register using the autoincrement feature so the DSP increments the HPIA register between the first and second HPI cycle. This incrementation allows the transfer of two consecutive words to or from PCI space, enabling reading or writing 32-bits of data at a time, which effectively doubles the data transfer rate.

5 Summary

You can use the PCI2040 C6x to interface to the C5420 with minimal external logic, but you must be aware of how the PCI2040 passes data to and from the HPI interface.

6 Example Schematic

Figure 7, Figure 8, and Figure 9 illustrate a schematic of an implementation of a single C5420 behind the PCI2040. This schematic shows only the connections between the PCI2040 and the C5420 and does not contain any detail of any C5420 implementation concerns. This example uses the memory map 0 method.

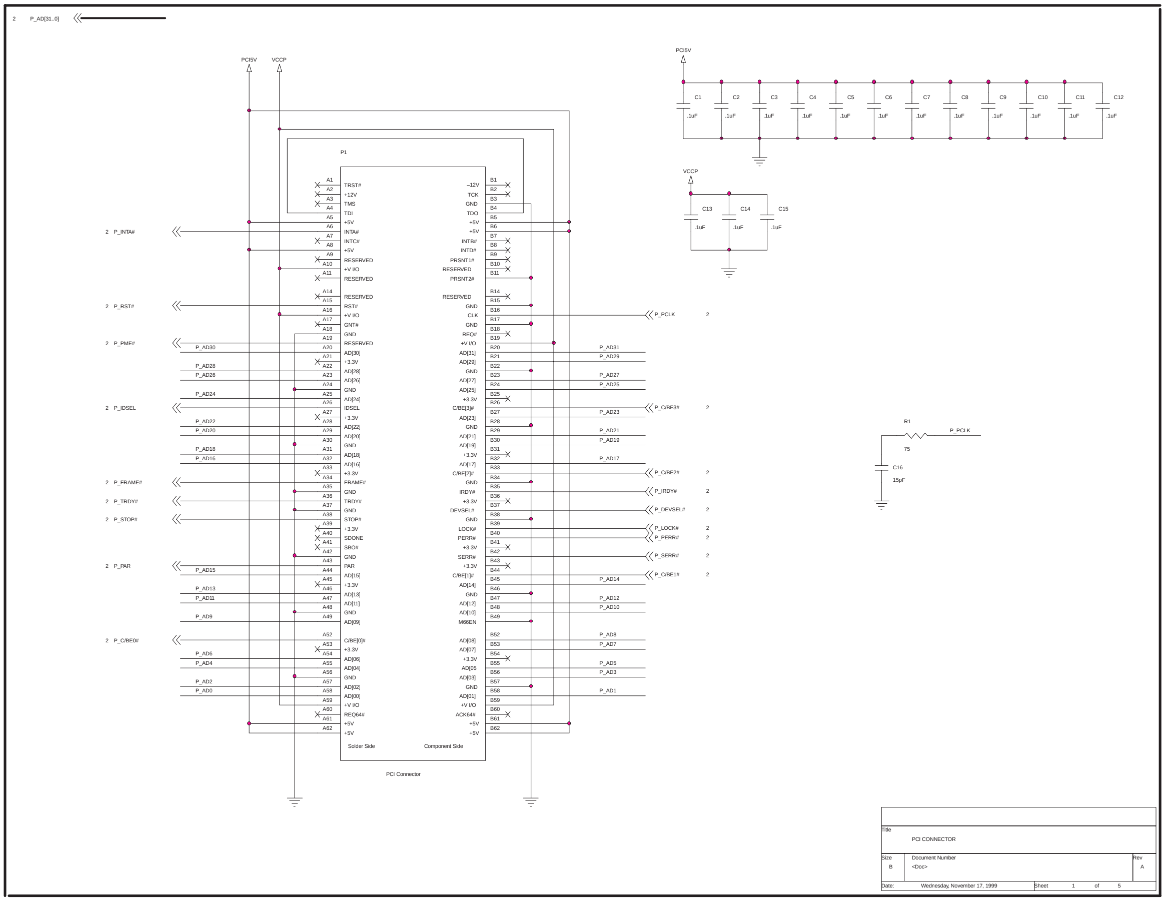


Figure 7. Schematic (1 of 3)

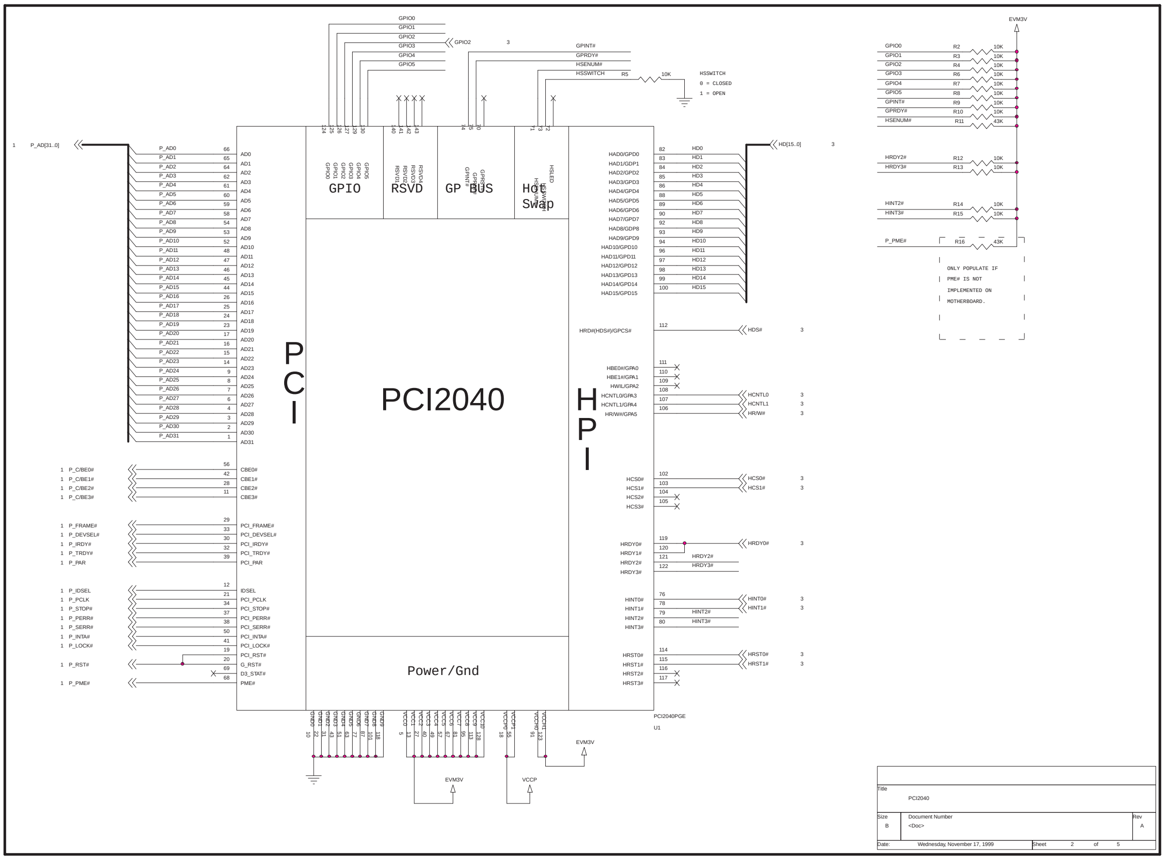
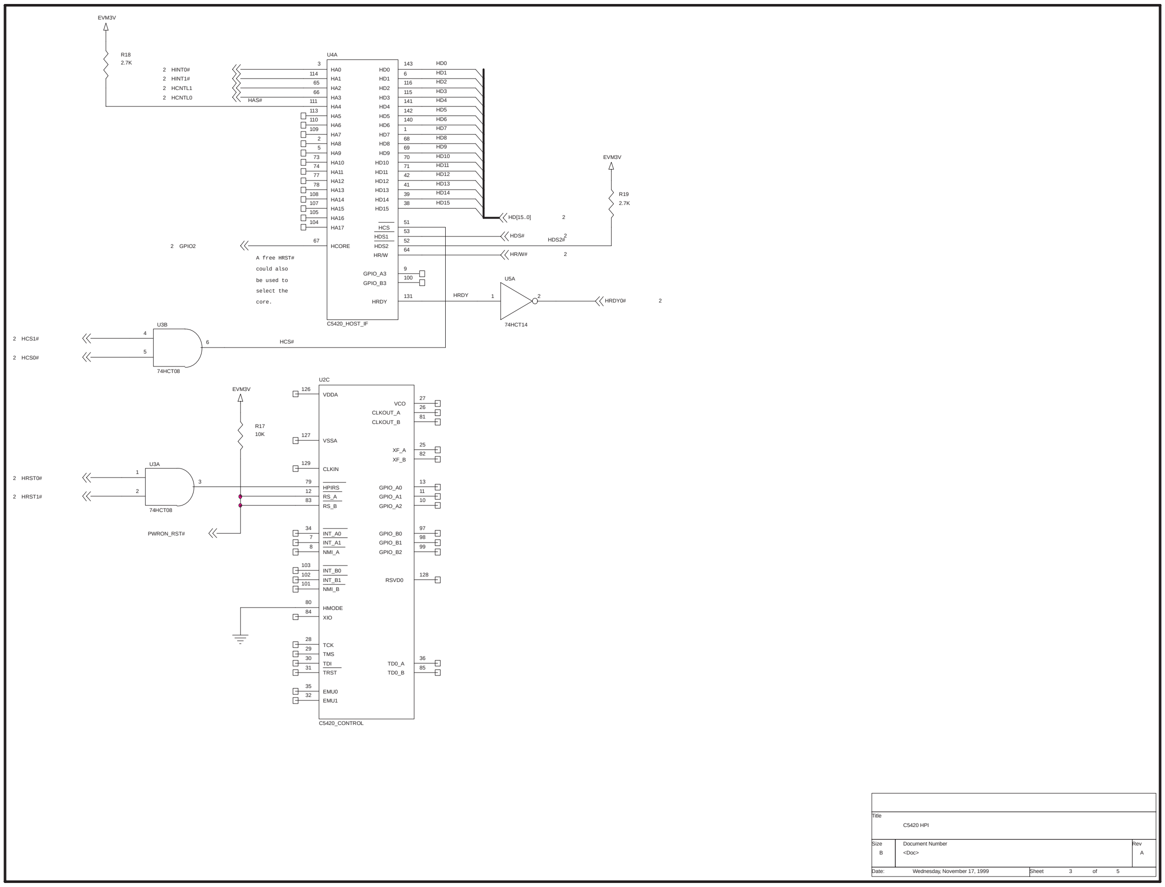


Figure 8. Schematic (2 of 3)



A free HRST# could also be used to select the core.

Figure 9. Schematic (3 of 3)

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

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

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.