TMS320C6415/6416: Using PCI EEPROM Interface and McBSP2 in a Single System

Jamon Bowen *C6000 Applications*

ABSTRACT

Some of the pins that the TMS320C6415 and TMS320C6416 digital signal processors (DSPs) use for their third multichannel buffered serial port (McBSP2) are multiplexed with the pins of the PCI serial EEPROM interface. In many applications, the only use for the PCI EEPROM is for auto-initialization of the PCI configuration space registers. Once PCI is configured, the PCI EEPROM interface can be disabled, and McBSP2 can be turned on. This application report describes how this is achieved.

Contents

List of Figures

List of Tables

List of Examples

1 TMS320C64x

The TMS320C64x™ has a number of peripherals available for a developer to use. Refer to the *TMS320C6000 Peripherals Reference Guide* (SPRU190) for more details. To maximize flexibility and minimize pin out footprint, several of the peripherals are multiplexed at the device pins and the system designer chooses which of the peripherals to use in the end system. In general, the peripheral set that the DSP will support is determined at reset via device configuration inputs, and cannot be changed after the device is released from reset. The PCI EEPROM interface, however, is often only used to initialize the PCI configuration registers and is idle the rest of the time. To use the pins of this interface more efficiently, they are multiplexed with the pins of a multichannel buffered serial port (McBSP2). Once the PCI configuration registers have been loaded, the EEPROM can be disabled, and McBSP2 can be enabled. This allows a system to use EEPROM for PCI auto-initialization and still use McBSP2 for serial communication.

1.1 PCI Support

The PCI port for the TMS320C64x supports connection of the DSP to a PCI host via the integrated PCI master/slave bus interface. The PCI port conforms to PCI specification revision 2.2. This port allows 32-bit address/data transfers at 33 MHz to other devices using a well-established bus standard. The TMS320C64x provides a built-in interface to external EEPROM to allow the PCI configuration registers to load with predetermined values. Once this configuration completes, the EEPROM remains in an idle state, unless it is accessed by the application.

2 PCI EEPROM Interface

The TMS320C64x DSP supports a 4-wire serial interface to a 4KB EEPROM. The DSP requires a specific format for the data stored in the serial EEPROM because a portion of this memory is used for PCI auto-initialization. The remaining locations are not used for auto-initialization and can be used for storing other data. This provides a convenient place for applications to store non-volatile data between power ups. The first 28 bytes of the EEPROM are reserved for the auto-initialization of PCI configuration registers. The memory map of the first 28 bytes of data is given in Table 1.

Table 1. EEPROM Memory Map

Address	Contents (msb lsb)
0h	Vendor ID
1h	Device ID
2h	Class Code [7:0]/Revision ID
3h	Class Code [23:8]
4h	Subsystem Vendor ID
5h	Subsystem ID
6h	Max_Latency/Min_Grant
7h	PC_D1/PC_D0 (power consumed D1, D0)
8h	PC_D3/PC_D2 (power consumed D3, D2)
9h	PC_D1/PC_D0 (power dissipated D1, D0)
Ah	PC_D3/PC_D2 (power dissipated D3, D2)
Bh	Data_scale (PD_D3….PC_D0)

TMS320C64x is a trademark of Texas Instruments.

Ch	0000 0000 PMC[14:9], PMC[5], PMC[3]
Dh	Checksum

3 McBSP

The multichannel buffered serial port for the TMS320C6000™ is based on the standard serial port interface found on the TMS320C2000™ and TMS320C5000™ platform devices. In addition, the port can buffer serial samples in memory automatically with the aid of the EDMA controller. It also has multichannel capability compatible with the T1, E1, SCSA, and MVIP networking standards. Like its predecessors, it provides these capabilities:

- Full-duplex communication
- Double-buffered data registers that allow a continuous data stream
- Independent framing and clocking for receive and transmit
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected analog-to-digital (A/D) and digital-to-analog (D/A) devices.

The TMS320C64x McBSP has many additional features that are described in the *TMS320C6000 Peripherals Reference Guide* (SPRU190).

4 Using EEPROM and McBSP2 in a Single System

Up to three McBSPs are available for use on the TMS320C64x DSP. McBSP0 is always available, McBSP1 is multiplexed with the universal test and operations interface for autosynchronous transfer mode (UTOPIA™), and McBSP2 is multiplexed with the PCI EEPROM. Using the setup demonstrated in Figure 2, the PCI EEPROM can be used for PCI auto-initialization, then the McBSP2 port can be enabled for use by the DSP after reset. The PCI EEPROM and McBSP2 internal connections are illustrated in Figure 1.

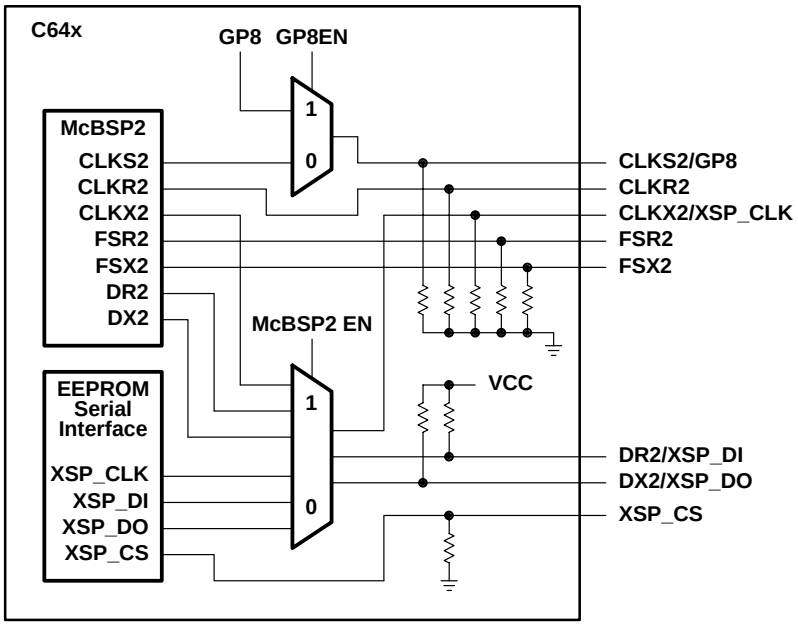

Figure 1. PCI EEPROM and McBSP2 Internal Connections

TMS320C6000, TMS320C2000, TMS320C5000, and UTOPIA are trademarks of Texas Instruments.

The McBSP pin, CLKS2, is also multiplexed with the general-purpose input/output (GPIO) pin, GP8. This should not interfere with the interface as long as this pin is not enabled by the DSP program.

4.1 Peripheral Selection

PCI support in the TMS320C64x is selected for an application by setting the PCI_EN pin high. This pin must remain valid at all times, and is not permitted to change its value throughout device operation. If PCI support is selected in an application, then the MCBSP2_EN pin is used to determine whether to support McBSP2 or EEPROM. This pin is required to be driven valid at all times, but the user can change its value dynamically after device reset. To allow user selection of McBSP2/EEPROM support, the following interface (shown in Figure 2) can be used:

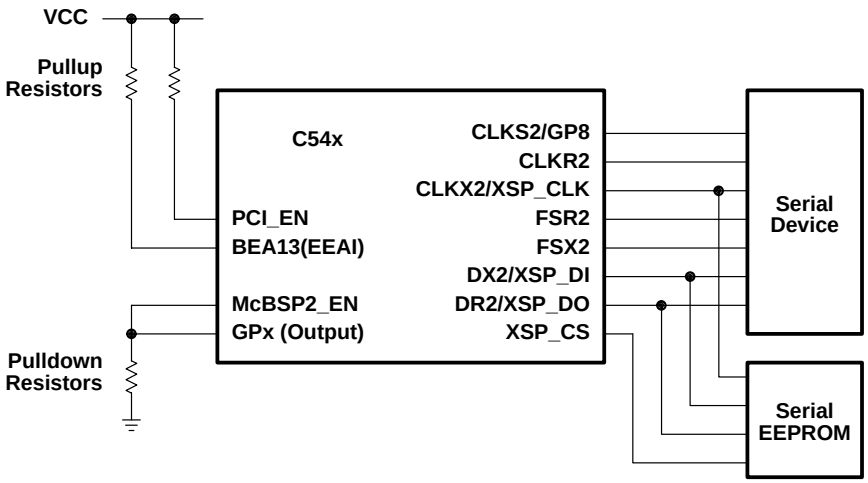

Figure 2. McBSP2/EEPROM Selection Interface

McBSP2_EN contains an internal pulldown resistor and GP0–GP3 also have internal pulldown resistors. If one those GPIO pins are used in this interface, an external pulldown resistor is not necessary, but recommended. Although internal pulldown resistors exist on these pins, providing external connectivity adds convenience in debugging, and flexibility in switching operating modes. GP4–GP7 contain an internal pullup resistor, so eliminating the external pulldown is not an option if one of those pins is used.

In the interface shown in Figure 2, MCBSP2_EN is connected to one of the general-purpose pins (GP0–GP7). At reset, PCI_EN is pulled up to 1, and McBSP2_EN is pulled down to zero, to enable the PCI EEPROM interface. After reset has completed, the PCI is configured from the EEPROM (assuming EEAI is enabled), and McBSP2 can be enabled. This is accomplished by setting the GPx as an output where ‘x’ can take on values 0–7, depending on which GPIO pin is used, and then by sending a 1 to MCBSP2_EN. This configuration satisfies all the requirements of the MCBSP2_EN pin by doing the following:

- Automatically enabling PCI EEPROM whenever the DSP is reset
- Always presenting valid data to the MCBSP2_EN pin
- Switching between the peripheral functional modes without any glue logic.

Other interface designs would need to provide support for each of these properties. A external solution could satisfy this requirement, but it would require a pulldown resistor on MCBSP2_EN (to ensure that MCBSP2_EN is always active) or actively driving the pin low before reset is inactive, and then driving the pin high at some time in the future. The external solution would need a signal from the DSP once PCI initialization completed. This signal could be sent with a GPIO pin. At this point, however, the current interface would be effectively duplicated with some extra logic. Another possible solution is to have the PCI host hold the pin low during reset, then enable the pin as soon as the DSP's PCI had initialized.

5 Switching Modes

To ensure proper operation of the McBSP, EEPROM, and the external serial device the switch from PCI EEPROM to McBSP2 needs to be a smooth transition. To ensure that the switch causes no errors there are a few requirements for the EEPROM and the serial device.

5.1 EEPROM

The EEPROM that is hooked to this interface is selected via XSP_CS high. This pin has an internal pulldown resistor to ensure that the serial EEPROM stays inactive when it is not in use. The EEPROM that is used with this interface must be compatible with the interface that TMS320C64x provides. See the *TMS320C6415 Fixed-Point DSP* (SPRS146) or *TMS320C6416 Fix-Point DSP* (SPRS164) data sheets for details.

The major requirements for this interface include the following:

- The EEPROM must ignore the clock between accesses.
- The EEPROM data out must go into a high impedance state when chip select (XSP_CS) is inactive.

These requirements are necessary to avoid bus contention with MCBSP2 and the serial device. If the EEPROM's data output does not have a high impedance state then an external buffer could be added.

5.2 Serial Device

The McBSP is a very versatile interface and can work with a wide variety of serial devices. A interface between PCI EEPROM and a serial device imposes restrictions on the type of serial device that can be used. To get around these restrictions and allow any serial device access to McBSP2, the following problems need to be solved (depending on the serial device):

- Bus contention on CLKX2/XSP_CLK (only a concern if the serial device is the clock master of McBSP2's transmissions)
- Bus contention on DR2/XSP_DI (EEPROM data out and the serial devices data out share the same connection.)
- How to activate/wakeup the serial device
- Without addressing these problems, McBSP2 will not work with all the serial devices that it normally could. Modifying the interface to accommodate other serial devices, however, is not difficult. The particular modifications will be different depending of the serial device that is used, but the following suggestions may be helpful:

- Add external buffers to the serial device pins that need to be in a high-impedance state when PCI EEPROM is in use. The buffer(s) can be taken out of high impedance mode by running the GPIO signal that enables McBSP2 to the buffer's enable. This will only work with the serial device's output, *not* the clock. A GPIO pin would be needed to enable the clock after McBSP2 is enabled; otherwise, bus contention could result (see Figure 3 for timing details).
- An additional GPIO signal can be used to keep the serial device in reset until McBSP2 is enabled.
- Another clock source can be used as a free-running clock to the serial device. This clock source could be enabled by a GPIO controlled buffer.

Figure 3 illustrates the switch from PCI EEPROM support to McBSP2 support, when all of the requirements for the EEPROM and the serial device are met:

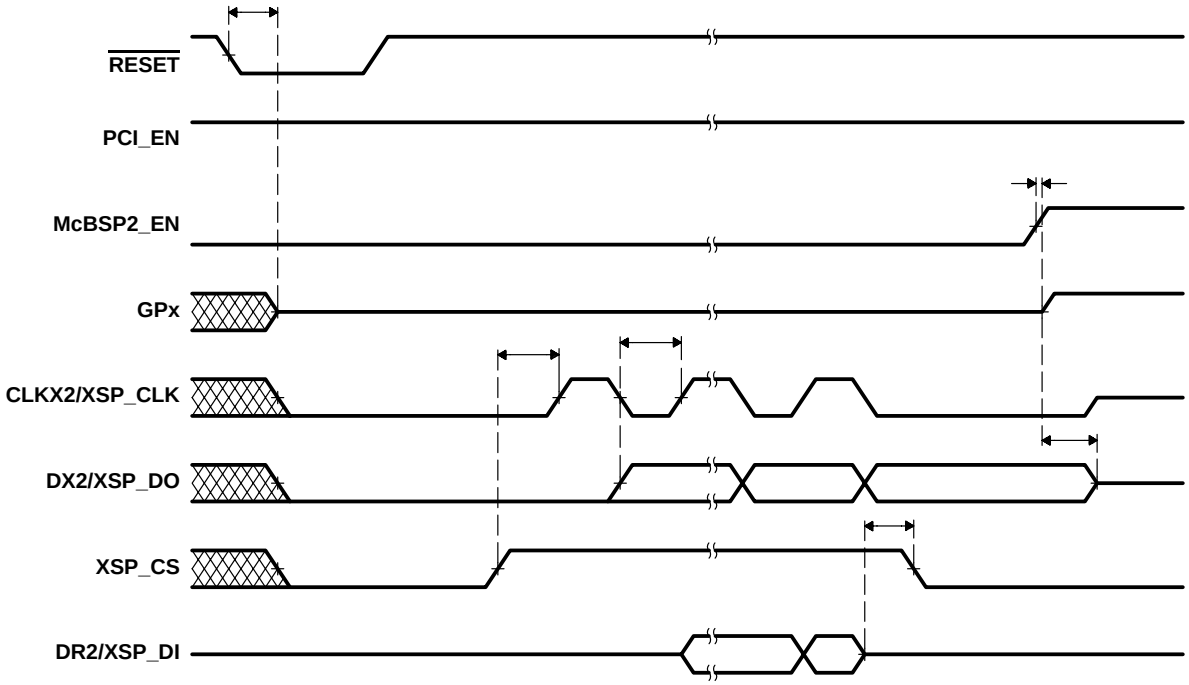

Figure 3. Interface Timing

Once the interface in Figure 2 is implemented, the software to support switching between EEPROM and McBSP2 can be implemented with relative ease. Example 1 shows GP1 being used to drive the MCBSP2_EN input. Any GPx pin can be used to drive MCBSP2_EN pin, but depending on the application, one might be limited to use of specific GPx pins.

The actual switch between PCI EEPROM mode and McBSP2 mode is simple and is shown in Figure 3. The first step of the program is to wait until the EEPROM signals that auto-initialization has completed. Once this signal is sent, the GPIO pin of the interface (in this case, GP1) is enabled. Next, GP1 is switched to output mode from its default setting. Lastly, a 1 is written to the GPVAL register in the bit location for GP1, to enable McBSP2.

Example 1. Code to Support McBSP2/EEPROM Selection Interface

```
/*********************************************************/
/* Title  Using PCI EEPROM and McBSP2 in a Single System   */
/* Author:  Jamon Bowen                                    */
/* Description: Allows PCI to Autoinitialize using PCI     */
/*   EEPROM, then EEPROM is disabled, and McBSP2 is        */
/*   enabled.                                              */
/*********************************************************/

#define CHIP_6400

#include<csl.h>
#include<csl_pci.h>
#include<csl_gpio.h>

void main() {
 /*test to see if configuration is done*/
 while(PCI_FGET(EECTL,CFGDONE) == 0)
 {}
 /*Enable GP1*/
 GPIO_enablePin(GPIO_PIN1);

 /*Set GPDIR of GP1 to output */
 GPIO_pinDirection(GPIO_PIN1, GPIO_OUTPUT);

 /*Write a one to GPVAL of GP1*/
 GPIO_pinValueSet(GPIO_PIN1,1);

}
```

This portion of code should be inserted at the beginning of an application. By using this simple code and interface, the use of McBSP2 and PCI EEPROM in a single system can be achieved.

6 References

1. *TMS320C6000 Peripherals Reference Guide* (SPRU190).
2. *TMS320C6415 Fixed-Point DSP* (SPRS146).
3. *TMS320C6416 Fixed-Point DSP* (SPRS164).
4. *TMS320C6000 Chip Support Library API Reference Guide* (SPRU401).

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.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265