Application Report
SLAA214–October 2004

Interfacing TI's Touch Screen Controllers Through McBSP

Wendy X. Fang, Ph.D.

Data Acquisition

ABSTRACT

This application report develops and presents the methodology to interface TI's touch screen controller products and multichannel buffered serial ports (McBSP) through SPI and I2S buses. The interface software code is designed and tested using the TMS320VC5416 DSK and the TSC2100 EVM.

1 Introduction

TI's touch screen controller (TSC) devices, many with audio codec built in, are widely used in handheld and mobile digital applications, such as personal digital assistant devices, personal media players, smart cellular phones, MP3 players, etc. Additional information about TI's TSC products can be found by browsing the analog product tree on the following TI Web site:

http://focus.ti.com/analog/docs/analogprodhome.tsp?templateId=4&familyId=82

To control and access a TSC device requires a host processor with the corresponding digital interface. A TI OMAP™ processor can be used as this host. The interface includes the serial digital port(s) and the corresponding software drivers. For information on TI OMAP processors, visit this Web site:

www.omap.com

Many of TI's TSC devices use SPI protocol for the control/data port and I2S protocol for the audio data port. To interface with a TSC, the host processor needs to support the corresponding buses. Both SPI and I2S buses can be designed and adapted using TI's multichannel buffered serial ports (McBSP). The McBSP is based on the standard serial port interface found on TI TMS320C2000, TMS320C5000, and TMS320C6000 digital signal processors (DSP). Inside many TI OMAP processors is either a TMS320C5400 or TMS320C5500 DSP core. Therefore, in many cases, the TSC and OMAP interface is becoming the TSC and McBSP interface

This application report develops the TSC and McBSP interface on both SPI and I2S buses. The code was designed and developed using a TMS320VC5416 DSK and TSC2100 EVM system. The details of the EVM and DSK boards can be found in References [2] and [5], respectively.

2 Hardware Connection

In developing the interface of this application report, the hardware connections between the 'C5416 DSP and the TSC2100 were made as shown in Table 1 and Table 2. The first McBSP port, named McBSP0, is used for the SPI interface, and the second port, named McBSP1, is used for the I2S interface. Inside McBSP0, the lines BCLKX and BCLKR and the lines BFSX and BFSR will be shorted because McBSP0 will run at stop mode for the SPI function. Therefore, they do not need to be connected externally. On McBSP1, the lines BCLKR and BCLKX and the lines BFSR and BFSX are shorted to each other externally.

Table 1. 'C5416 and TSC2100 SPI Hardware Interface

	HOST PROCESSOR PIN NAME	TSC2100 PIN NAME
SPI Clock	BCLKX0 (BGA N5 or PGE Pin 48)	SCLK (QFN Pin 4 or TSSOP Pin 8)
SPI Slave Select	BFSX0 (BGA M7 or PGE Pin 53)	$\overline{SS}$ (QFN Pin 7 or TSSOP Pin 11)
SPI MOSI Data	BDX0 (BGA L8 or PGE Pin 59)	MOSI (QFN Pin 6 or TSSOP Pin 10)
SPI MISO Data	BDR0 (BGA K5 or PGE Pin 45)	MISO (QFN Pin 5 or TSSOP Pin 9)

Table 2. 'C5416 and TSC2100 I2S Hardware Interface

	HOST PROCESSOR PIN NAME	TSC2100 PIN NAME
I2S Bit Clock	BCLKX1 (BGA N12 or PGE Pin 71) BCLKR1 (BGA N2 or PGE Pin 38)	BCLK (QFN Pin 31 or TSSOP Pin 3)
I2S Left/Right Word Clock	BFSX1 (BGA N13 or PGE Pin 73) BFSR1 (BGA M2 or PGE Pin 36)	LRCK (QFN Pin 27 or TSSOP Pin 31)
I2S Data to DAC	BDX1 (BGA M13 or PGE Pin 74)	DIN (QFN Pin 29 or TSSOP Pin 1)
I2S Data from ADC	BDR1 (BGA M1 or PGE Pin 35)	DOUT (QFN Pin 30 or TSSOP Pin 2)

In developing this interface, the hardware configuration on the 'VC5416 DSK board was kept at its default settings. On the TSC2100 EVM board, however, the SW1-3(SPI) and SW1-1 (I2S) were turned off so as to open the two ports for the external connection from the McBSPs.

These connections between the TSC and DSP, as shown in Table 1 and Table 2, were physically made through a DSK/TSC adapter card, connected to the 80-pin peripheral interface header (P2) on the 'C5416 DSK board and to the two headers for external SPI (J12) and I2S (J11) on the TSC2100 EVM board. For more details about the DSK/TSC adapter card, contact TI at the following e-mail address:

dataconvapps@list.ti.com

3 SPI Interface

The TSC2100 is entirely controlled by registers. Reading from and writing to these registers is accomplished through the SPI interface by an SPI master, which is called McBSP0 in this application report. Note that the TSC is always an SPI slave.

3.1 SPI at TSC2100

Figure 1 displays the TSC SPI timing diagram, in which data is valid at the first rising edge of the SCLK signal after the $\overline{SS}$ signal becomes active (goes low); data on the MOSI and MISO bus lines are changing at the rising edge of the SCLK and are read or sampled at the falling edge. Therefore, the data must remain stable around the falling edge of the SCLK to ensure the reliability of the SPI interface.

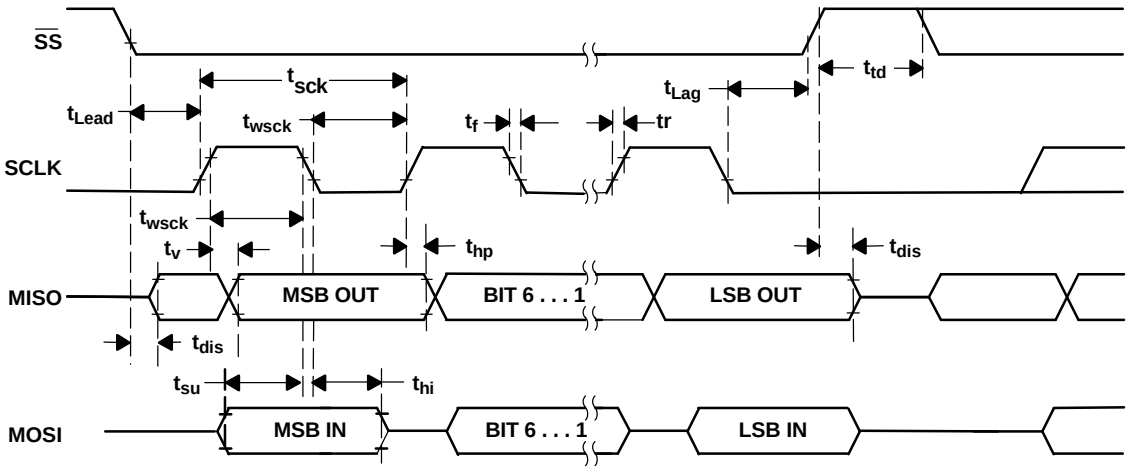

Figure 1. SPI Timing Diagram

The TSC2100 SPI operation sequence is shown in Figure 2 and Figure 3. Figure 2 illustrates the *Writing to the TSC* operation, in which a 16-bit control word is followed by one or more 16-bit data blocks that are sent from the host processor to the TSC (through the MOSI line) so as to control, set, and configure the TSC. Figure 3 shows the *Reading from the TSC* operation, in which a 16-bit command word is written to the TSC (through the MOSI line), requesting the reading; the data output from the TSC appears on the MISO line, starting from the second 16-bit word within the frame.

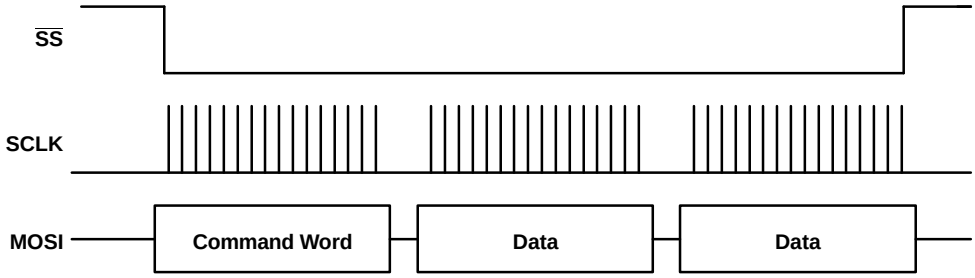

Figure 2. Write Operation for TSC2100 SPI Interface

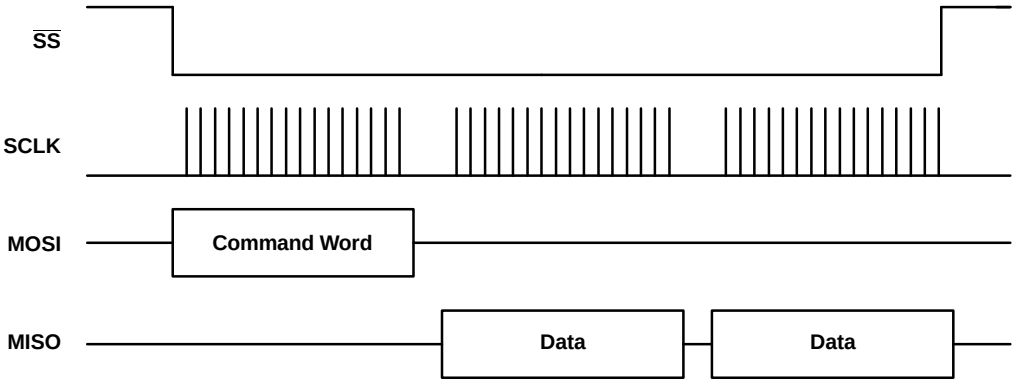

Figure 3. Read Operation for TSC2100 Operation Sequence

For more details on the TSC2100 SPI timing and operation, see Reference [1].

3.2 SPI at McBSP

There are many different ways to configure McBSP0 by programming the frame and clock so that the McBSP0 can interface with the SPI on the TSC2100. This application report discusses only one, as Table 3 to Table 5 illustrate. Table 3 lists the McBSP0 initialization sequence and example values; Table 4 outlines the SPI data receive (RX) and transmit (TX) sequence; and Table 5 gives several examples for reading and writing operations.

Table 3. McBSP0 Initialization

PROGRAM STEP	PROGRAMMING VALUE	EXPLANATION
1	SPCR1 = 0x0000 SPCR2 = 0x0000	Stop and reset receive, transmit, frame sync and sample rate generator : • $\overline{RRST}$ = $\overline{XRST}$ = $\overline{FRST}$ = $\overline{GRST}$ = 0b
2	SPCR1 = 0x1000 SPCR2 = 0x0000	Set sample rate generator at clock stop mode: • CLKSTP = 10b
3	PCR = 0x0A0C	Set McBSP as master to generate the $\overline{SS}$ (FSX) and SCLK (CLKX); and set data being changed at rising edge and read at falling edge of SCLK: • FSXM = CLKXM = FSXP = 1b • CLKXP = 0b
4	XCR1 = 0x00A0 XCR2 = 0x0001 RCR1 = 0x00A0 RCR2 = 0x0001	Set the interface to single-phased frame; and one 32-bit long data length with 1 SCLK delay: • RFRLEN = XFRLEN = 000000b • RWDLEN = XWDLEN = 101b • RDATDLY = WDATDLY = 01b
5	SRGR2 = 0x2000 SRGR1 = 0x00C7	Set sample rate generator using CPU clock / 200 • CLKSM = 1b, FSGM = 0b, CLKGDV=199
6	SPCR2 = SPCR2 | 0x0040	Enable sample rate generator ($\overline{GRST}$ = 1)
7	Wait(400 CPU clock)	Wait for at lease 2 SCLKs for sample rate generator to stabilize

Table 4. SPI Writing/Reading Sequence

PROGRAM STEP	PROGRAMMING VALUE	EXPLANATION
1	SPCR1 = SPCR1 | 0x0001 SPCR2 = SPCR2 | 0x0001	Enable receive and transmit
2	Wait(400 CPU clock)	Wait for at lease 2 SCLKs to stabilize
3	Writing or Reading Operation (Examples in Table 5)	A writing or reading frame.
4	… …	Continuous on more writing or reading frame
5	Wait for TX registers to be empty SPCR1 = SPCR1 & 0xFFFE SPCR2 = SPCR2 & 0xFFFE	Disable receive and transmit, if preferred.

Table 5. TSC 16-Bit Data Writing/Reading SPI Operation

	PROGRAMMING VALUE	EXPLANATION
A Single Writing Frame	While (!(SPCR2&0x0002)); DXR2 = (TSC command) DXR1 = (Data to Write)	• Wait for transmitter ready (empty) • Write 16-bit command into McBSP0_DXR2 • Write 16-bit data into McBSP0_DXR1

Table 5. TSC 16-Bit Data Writing/Reading SPI Operation (continued)

	PROGRAMMING VALUE	EXPLANATION

A Single Reading Frame	While (!(SPCR2&0x0002));
	DXR2 = (TSC command)
	DXR1 = 0x0000 (dummy)
	While (!(SPCR1&0x0002));
	Read_Data = DRR1
		• Wait for transmitter ready (empty)
		• Write 16-bit command into McBSP0_DXR2
		• Write 16-bit dummy data into McBSP0_DXR1
		• Wait for receiver ready (full)
		• Read TSC data from McBSP0_DRR1

A Multiple Writing Frame(1)	While (!(SPCR2&0x0002)); DXR1 = (TSC command) While (!(SPCR2&0x0002)); DXR1 = (Data to Write)	• Wait for transmitter ready (empty) • Write 16-bit command into McBSP0_DXR1 • Wait for transmitter ready (empty) • Write 16-bit data into McBSP0_DXR1
	While (!(SPCR2&0x0002)); DXR1 = (Data to next addr)	• Wait for transmitter ready (empty) • Write data to the next address
	…	• Repeat the preceding two lines until all data is transferred.
A Multiple Reading Frame(1)	While (!(SPCR2&0x0002)); DXR1 = (TSC command)	• Wait for transmitter ready (empty) • Write 16-bit command into McBSP0_DXR1

	While (!(SPCR2&0x0002));
	DXR1 = 0x0000 (dummy)
	While (!(SPCR1&0x0002));
	Read_Data = DRR1
		• Wait for transmitter ready (empty)
		• Write 16-bit dummy data into McBSP0_DXR1
		• Wait for receiver ready (full)
		• Read TSC data from McBSP0_DRR1

	While (!(SPCR2&0x0002));
	DXR1 = 0x0000 (dummy)
	While (!(SPCR1&0x0002));
	Read_Data = DRR1
		• Wait for transmitter ready (empty)
		• Write 16-bit dummy data into McBSP0_DXR1
		• Wait for receiver ready (full)
		• Read next addressed data

	…	• Repeat the preceding four lines until all data is transferred.

(1) For multiple SPI reading/writng, the McBSP0 control registers' settings can be obtained by sending an e-mail to dataconvapps@list.ti.com.

Figure 4 shows the measured SPI timing diagram of a single writing frame. Channel 1 is the SPI /SS signal; channel 2 is SCLK; channel 3 MOSI (where the command is 1000h and the data is 0200h); and channel 4 is MISO (for writing to the TSC, it should be ignored).

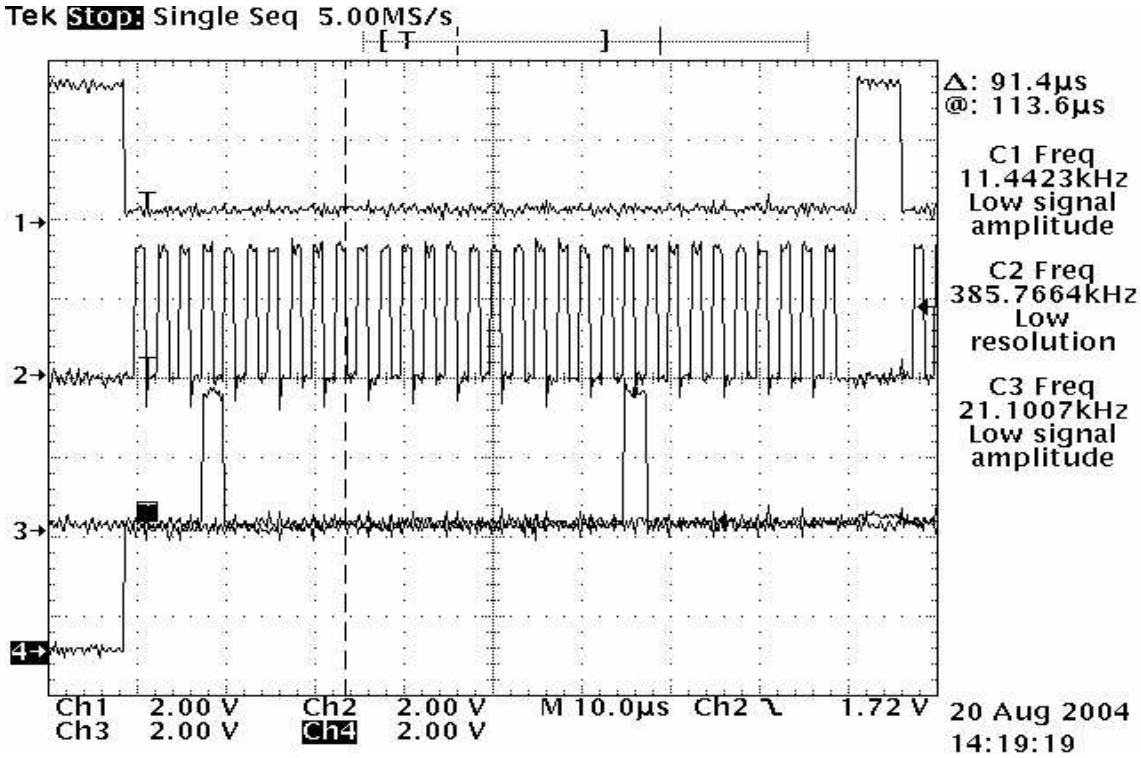

Figure 4. McBSP and TSC SPI Interface (Writing)

4 I2S Interface

I2S was designed specifically for transmitting digital audio data between processors and digital devices. Most TI audio codec devices support the I2S interface.

4.1 I2S at TSC2100

The TSC2100 supports various 4-wire, digital audio interfaces, including the I2S, Left-Adjusted, Right-Adjusted, and DSP modes. For details, see Reference [1]. The TSC2100 audio interface can work as either a master or a slave. Owing to the similarity of these audio interfaces, this application report focuses on the I2S interface only.

Figure 5 shows the I2S timing diagram at TSC2100. Note that the I2S left/right-channel selection, LRCK, changes at the falling edge of the I2S bit clock, BCLK, and the codec's ADC data DOUT and DAC data DIN change also at the BLCK's falling edge and are read at the rising edge. Thus, the DOUT and DIN must be kept stable around the BCLK rising edges.

Figure 6 displays the I2S data receive and transmit operation timing. The I2S data have a bit delay after LRCK changed; the data length, n, in Figure 6 can be 16, 20, 24, or 32 bits

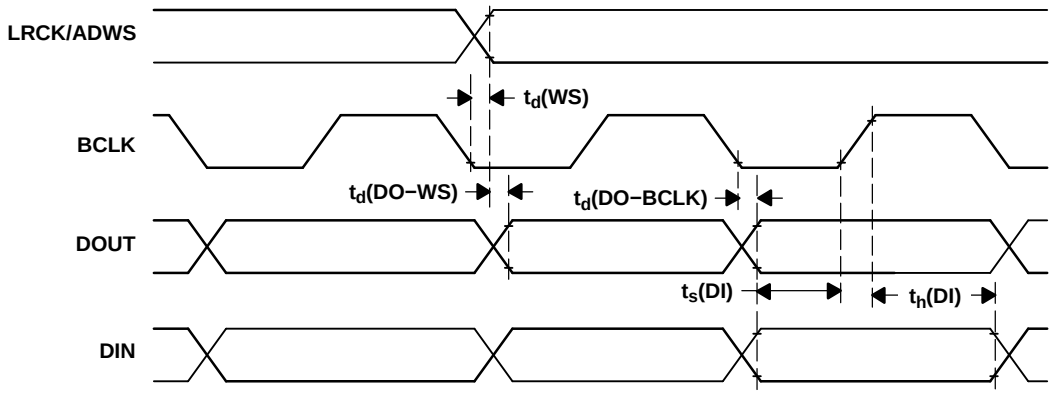

Figure 5. TSC2100 I2S Timing Diagram

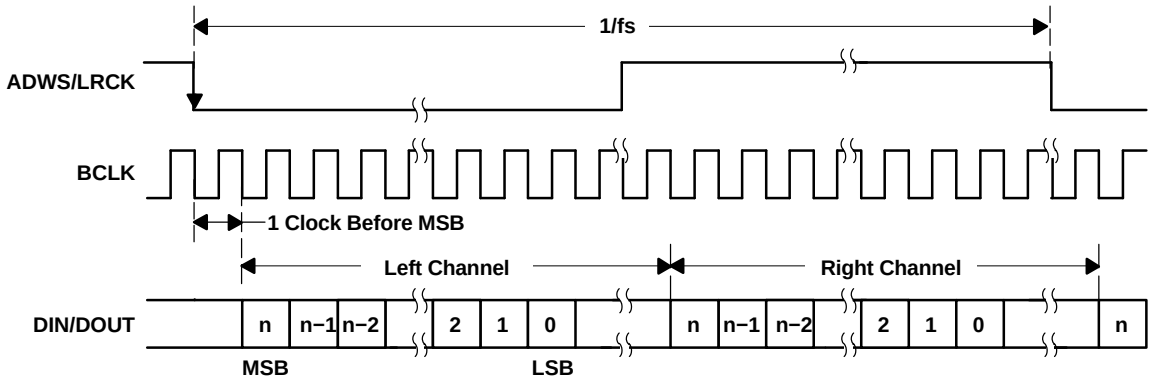

Figure 6. Operation for I2S Interface

To have the TSC running in I2S mode and cooperating with the McBSP, the TSC registers must be set up through the SPI interface, as discussed previously. The TSC registers settings are specifically based on the operation modes and applications. An example of the TSC2100 settings is given in Table 6, where the TSC2100 audio control registers are set up to run as a master in continuous I2S mode, with a 16-bit word length, and 44.1 kHz. For more on using SPI, see Table 4 and Table 5 .

Table 6. Using SPI to Program TSC2100 Audio Function

PROGRAM STEP	PROGRAMMING VALUE	EXPLANATION
1	SPCR1 = SPCR1 | 0x0001 SPCR2 = SPCR2 | 0x0001	Enable receive and transmit
2	Wait(400 CPU clock)	Wait for at least 2 SCLKs to stabilize
3	While (!(SPCR2&0x0002)); DXR2 = 0x10C0 DXR1=0x2800	Program TSC2100 Page 2 0x06 register
4	While (!(SPCR2&0x0002)); DXR2 = 0x1360 DXR1 = 0x1120	Program TSC2100 Page 2 0x1B register
5	While (!(SPCR2&0x0002)); DXR2 = 0x10A0 DXR1 = 0x3800	Program TSC2100 Page 2 0x05 register
6	While (!(SPCR2&0x0002)); DXR2 = 0x1020 DXR1 = 0x0000	Program TSC2100 Page 2 0x01 register

Table 6. Using SPI to Program TSC2100 Audio Function (continued)

PROGRAM STEP	PROGRAMMING VALUE	EXPLANATION
7	While (!(SPCR2&0x0002)); DXR2 = 0x1040 DXR1 = 0x0000	Program TSC2100 Page 2 0x02 register
8	While (!(SPCR2&0x0002)); DXR2 = 0x0000 DXR1 = 0x0000	Write a dummy frame to put all data off from the McBSP's TX buffer
9	Wait for TX registers to be empty SPCR1 = SPCR1 & 0xFFFE SPCR2 = SPCR2 & 0xFFFE	Disable receive and transmit, if preferred.

4.2 I2S at McBSP

Table 7 and Table 8 describe how to program the McBSP1 interfacing with an I2S device (in this example, the TSC2100), using the settings with the McBSP1 control registers. Table 7 lists the McBSP1 initialization sequence and example values; Table 8 outlines the I2S data receive (RX) and transmit (TX) sequence if performance is at ISR or other normal routine. The most common way to move audio streaming data from or to the I2S bus is to use the DMA controller, which requests configuration and setup for the DMAC registers on the host processor. DMAC settings and operation are not discussed in this application report.

Table 7. McBSP1 Initialization for 16-Bit Audio Streaming

PROGRAM STEP	PROGRAMMING VALUE		EXPLANATION
	McBSP1 as MASTER(1)	McBSP1 as SLAVE(1)	
1	SPCR1 = 0x0000 SPCR2 = 0x0000	SPCR1 = 0x0000 SPCR2 = 0x0000	Stop and disable receive, transmit, and sample-rate generator
2	SPCR1 = 0x4000 SPCR2 = 0x0020	SPCR1 = 0x4000 SPCR2 = 0x0020	Set to left-adjusted and RINT/XINT interrupt triggering mode
3	PCR = 0x0A03	PCR = 0x0003	Set McBSP as master or slave with proper polarities
4	XCR1 = 0x00A0 XCR2 = 0x0001 RCR1 = 0x00A0 RCR2 = 0x0001	XCR1 = 0x00A0 XCR2 = 0x0001 RCR1 = 0x00A0 RCR2 = 0x0001	Set the interface to single-phased frame; and 32-bit long data length with 1 SCLK delay.
5	SRGR2 = 0x101F SRGR1 = 0x0F07	N/A	Set sample rate generator Clock source : CLKS (MCLK) CLKGDV = 7 (0x07) FWID = 15 (0x0F) FPER = 31(0x01F)
6	SPCR2 = SPCR2 | 0x0040	N/A	Enable sample rate generator ($\overline{\text{GRST}}$ = 1)
7	Wait(400 CPU clock)	N/A	Wait at lease 2 SCLKs for sample rate generator to stabilize
8	SPCR1 = SPCR1 | 0x0001 SPCR2 = SPCR2 | 0x0001	SPCR1 = SPCR1 | 0x0001 SPCR2 = SPCR2 | 0x0001	Enable receive and transmit
9	Wait(400 CPU clock)	Wait(400 CPU clock)	Wait at least 2 SCLKs to stabilize

(1) The McBSP1 should run as a slave when TSC2100 is an I2S master, as shown in Table 6.

To read or write audio data from or to the TSC2100 by using ISR, the McBSP1 transmit interrupt vector (XINT) is set and enabled, and an ISR is called when an XINT event occurs (that is, the XINT occurs when a new frame sync occurs). Table 8 gives the ISR programming sequence, and Figure 7 shows the result of the implementation, where the channels 1 to 4 are BCLK, LRCK, DIN, and DOUT, respectively.

Table 8. SPI Writing/Reading Sequence at XINT ISR

PROGRAM STEP	PROGRAMMING VALUE	EXPLANATION
1	While (!(SPCR2 & 0x0002));	Wait for transmitter ready (empty)
2	DXR2 = Left Audio Data DXR1 = Right Audio Data While (!(SPCR1 & 0x0002)); AudioData = DRR1	Writing/reading stereo/2-channel data through I2S
3	IFR |= 0x0800	Reset BXINT1 interrupt

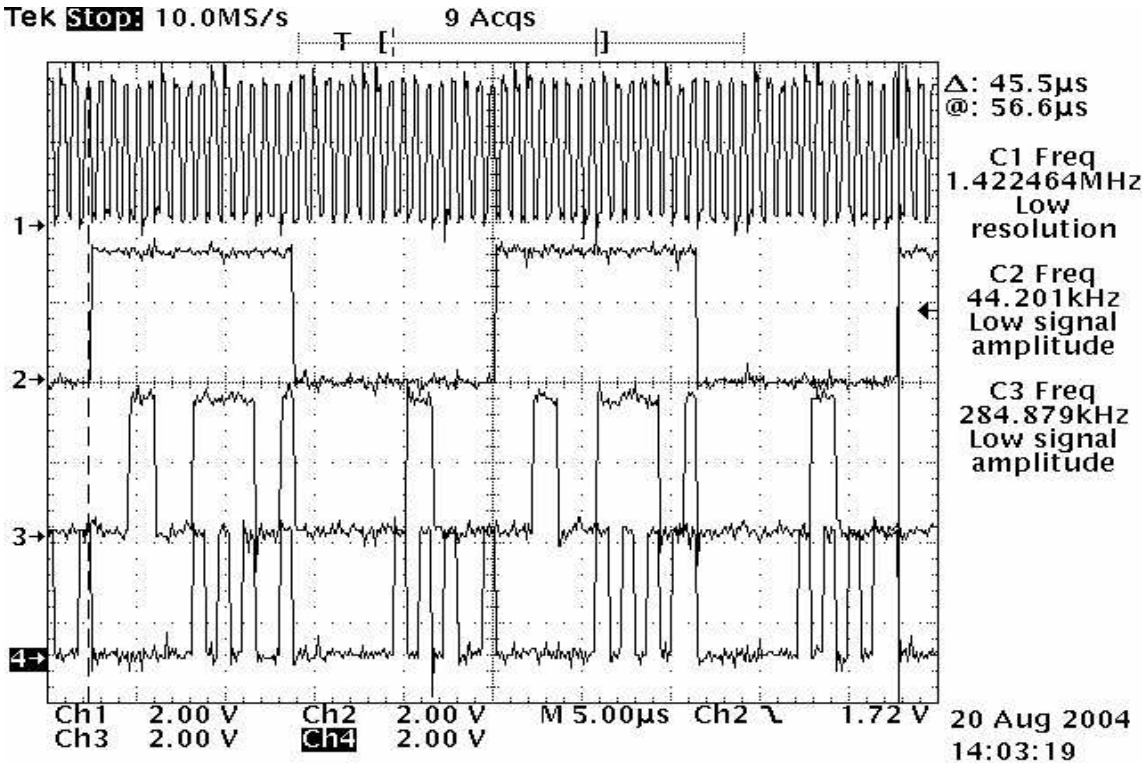

Figure 7. McBSP and TSC I2S Interface

5 Conclusion

This application report discussed the design and development of the SPI interface for McBSPs communicating with TSC2100 control and data registers, and the I2S interface for McBSP/TSC2100 audio data streaming. The corresponding code can be found by e-mail at:

dataconvapps@list.ti.com

6 Reference

1. *TSC2100,Programmable Touch Screen Controller with Integrated Stereo Audio CODEC and Headphone/Speaker Amplifier* data sheet (SLAS378)
2. *TSC2100 EVM, Touch Screen Controller Evaluation Module* user's guide (SLAU100)
3. *TMS320VC5416 Fixed-Point Digital Signal Processor* data manual (SPRS095)
4. *TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals* (SPRU302)
5. TMS320VC5416 DSK, Technical Reference, Spectrum Digital

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