Application Report
SPRAAJ2–August 2008

Interfacing the TMS320F2833x to the AIC23B Stereo Audio Codec

Christine Peng

ABSTRACT

This application report describes the implementation of an audio application using the TLV320AIC23B stereo audio codec interfaced to the TMS320F28335 digital signal controller and the setup of the multi-channel buffered serial port (McBSP) and direct memory access (DMA) module to support the application.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/SPRAAJ2.

Contents

List of Figures

List of Tables

1 Introduction

With the addition of the direct memory access (DMA) bus on the TMS320F2833x, the multi-channel buffered serial port (McBSP) peripheral benefits from the DMA's ability to move data efficiently between buffers in SARAM without tying up precious central processing unit (CPU) resources. To demonstrate the flexibility of the DMA-McBSP interface, this application report discusses an application example which transfers audio data between the F2833x and the TLV320AIC23B audio stereo codec.

Prior to reading this application report and using the example code, it is important to refer to the following documents to understand how to properly set up and configure the McBSP and DMA modules on the F2833x for interfacing with the AIC23B audio codec.

- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controller (DSC) Data Manual* (SPRS439)
- *TLV320AIC23B Stereo Audio CODEC, 8- to 96-kHz, With Integrated Headphone Amplifier Data Manual* (SLWS106)
- *TMS320x2833x Direct Memory Access (DMA) Reference Guide* (SPRUFB8)
- *TMS320F2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7)

2 McBSP Interface to AIC23B Audio Codec

The TLV320AIC23B is a high-performance stereo audio codec from Texas Instruments with integrated digital-to-analog converter (DAC) and analog-to-digital converter (ADC) circuits for converting audio data between the digital and analog domains. The codec inputs audio data from a microphone or audio player via line inputs and converts it to digital data, which can then be received via the McBSP in either normal *DSP* or inter-IC sound (I2S) mode and processed by the F2833x device.

The processed digital data can then be transmitted from the McBSP back to the AIC23B. It is converted by the integrated DAC back to analog audio data, which is output to speakers or headphones. The McBSP is interfaced to the DMA to expedite this transfer process in the F2833x. A serial peripheral interface (SPI) is also required to send initialization and control signals to the AIC23B codec.

In the example software accompanying this application report, McBSP-A is configured to handle the audio stream and McBSP-B is configured for clock-stop (SPI) mode to handle the AIC23B control signals.

Note: In the F28x Peripheral Explorer implementation of this software, McBSP-A is configured to handle the audio stream and SPI-A (instead of McBSP-B) is used to configure the AIC23B control registers. The remainder of this document will *not* discuss this particular implementation.

XDS510, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

2.1 System Hardware Connections

Figure 1 illustrates the interface between the AIC23B codec and the F2833x peripherals required for this application. A reference design schematic of an AIC23B codec printed circuit-board layout can be found in the Appendix.

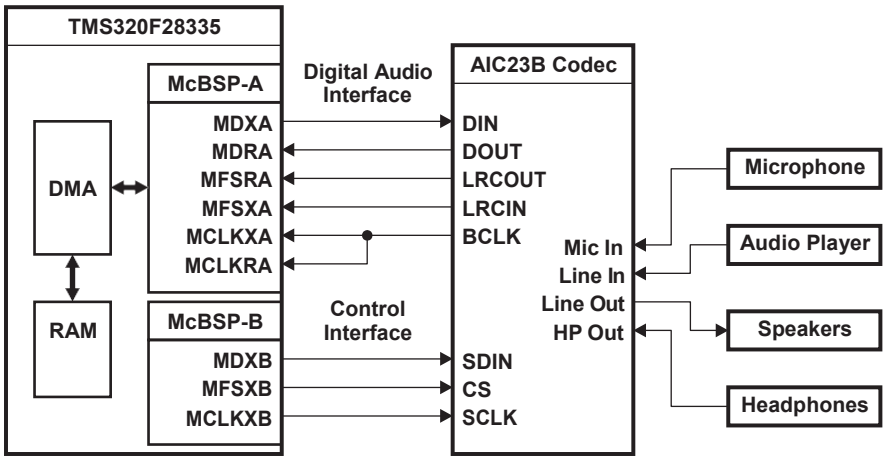


Figure 1. TMS320F28335 Interface With AIC23B Codec

2.1.1 Audio Interface

For the digital audio interface, the AIC23B is the master and the F2833x is the slave. In master mode, the AIC23B can transmit either its internal 12 MHz clock frequency or an external crystal frequency out of the BCLK pin to the McBSP-A CLKXA and CLKRA pins (shorted), clocking the audio data transfers. Additionally, the AIC23B codec transmits a frame sync or word select (I2S mode) via the LRCOUT and LRCIN pins to the McBSP-A MFSRA and MFSXA pins, respectively.

Analog audio data from a microphone or audio player line input is sampled and converted to digital data by the AIC23B ADC. The AIC23B then transmits the digital audio data out of DOUT to the MDRA pin in McBSP-A. In the F2833x, the DMA pulls the data from the McBSP data result registers and stores it in a ping-pong buffer in random access memory (RAM). Once processing on the data is complete, the DMA pulls the processed data from RAM and feeds it to the McBSP data transmit registers for transmission. The data is transmitted back out of the MDXA pin and into the AIC23B codec via DIN. The AIC23B DAC then converts the digital data back to analog audio data and sends it to external speakers or headphones.

2.1.2 Control Interface

The control interface is a uni-directional SPI interface over which the F2833x sends initialization commands to set up the AIC23B registers. In the example included with this application report, McBSP-B configured for clock-stop (SPI) mode is the master and the AIC23B codec is the slave. McBSP-B transmits a clock divided down from the F2833x LSPCLK out of the MCLKXB pin to the AIC23B SCLK pin. It then transmits serial commands via the MDXB pin to the AIC23B SDIN pin to setup the AIC23B registers. For AIC23B register settings, see the *TLV320AIC23B Stereo Audio CODEC, 8-to 96-kHz, With Integrated Headphone Amplifier Data Manual* (SLWS106). Finally, MFSXB sends a chip select signal to the CS pin on the AIC23B.

2.2 Software

The software package included with this application report is divided into three directories: two for the AIC23B driver that is included (and source directories) and one for the audio project. All necessary software is located in the DSP2833x_Audio_App folder. The directory structure is shown in Figure 2. Table 1 lists the software files associated with these directories.

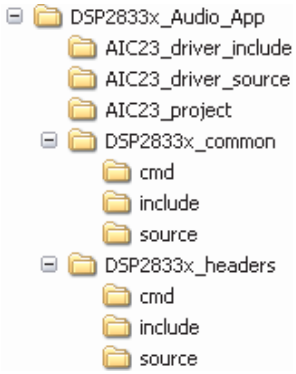


Figure 2. DSP2833x_Audio_App Directory Structure

Table 1. Software File Listing

File	Description
../AIC23_project/	Directory - AIC23B in DSP/I2S mode
AIC23_project.pjt	Main project file for F2833x-AIC23B audio project
AIC23_project.c	Project main C-source file – Configures McBSP to interface to AIC23B in either normal DSP McBSP mode using the internal 12 MHz *USB* clock or I2S mode using an external 12.288 MHz clock. In both cases, audio data from the source is sampled at 48 kHz. #define directives allow you to select I2S mode or DSP mode and line input or microphone input.
AIC23_SPI_control.c	McBSP in SPI mode AIC23B control source code
AIC23_project.wks	Workspace loads project, ping and pong buffer memory windows, and graphs illustrating audio data in buffers. This workspace was generated while using an XDS510™ USB emulator and Code Composer Studio™ IDE V3.3.
../AIC23_driver_include/	Directory – AIC23B Driver Include Files
AIC23.h	AIC23B register and bit definitions and function prototypes
../AIC23_driver_source/	Directory – AIC23B Driver Source Files
AIC23.c	AIC23B driver functions
../DSP2833x_headers/	Directory – DSP2833x header files (includes files extracted from C2833x/C2823x C/C++ Header Files and Peripheral Examples)
../DSP2833x_common/	Directory – DSP2833x common files (includes files extracted from C2833x/C2823x C/C++ Header Files and Peripheral Examples with minor modifications to remove unnecessary files)

2.2.1 AIC23B Driver Software

Because the AIC23B audio interface is controlled by serial commands transmitted from the F2833x, the driver software has been included for the F2833x to generate these commands.

AIC23.h, which can be found in the AIC23_include directory, includes definitions for all the AIC23B register and bit names. AIC23.c, which can be found in the AIC23_source directory, includes driver control functions that return the necessary command values to be transmitted to the AIC23B codec via McBSP-B in AIC23_SPI_control.c. Some of the included driver functions require input parameters for options such as volume control or sample rate generation. Currently, the driver software supports only the I2S and DSP (normal McBSP) audio data formats. To support other data formats or miscellaneous settings, the functions in AIC23B.c can be modified accordingly for a specific application.

2.3 DSP Mode vs. I2S Mode

There are several minor differences to consider when configuring the AIC23B driver software and setting up the F2833x McBSP to support AIC23B DSP mode versus I2S mode.

In AIC23B DSP mode, the LRCIN and LRCOUT pins generate a McBSP-compatible frame sync signal to start a data transfer. The left-channel data is transferred first, immediately followed by the right-channel data. Figure 3 shows waveforms illustrating the DSP-mode data format.

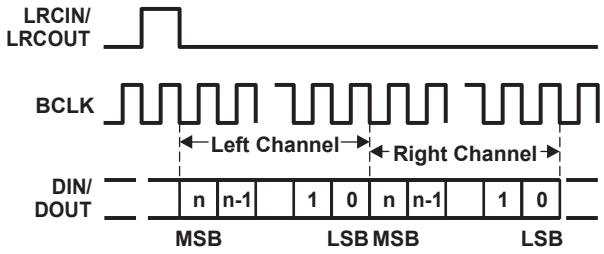


Figure 3. AIC23B DSP-Mode Data Format

In AIC23B I2S mode, the LRCIN and LRCOUT pins transmit low for left-channel data and high for right-channel data. The MSB of the 32-bit word is available on the second rising edge of BCLK, after the falling edge on LRCIN or LRCOUT (see Figure 4). The McBSP module must read two consecutive words, one after the other, after detecting an active low on the MFSR/MFSX pins. Therefore, to receive a 32-bit left-channel word and a 32-bit right-channel word properly, the AIC23B sample rate must be configured so that there are exactly 64 BCLK cycles in one LRCIN/LRCOUT cycle.

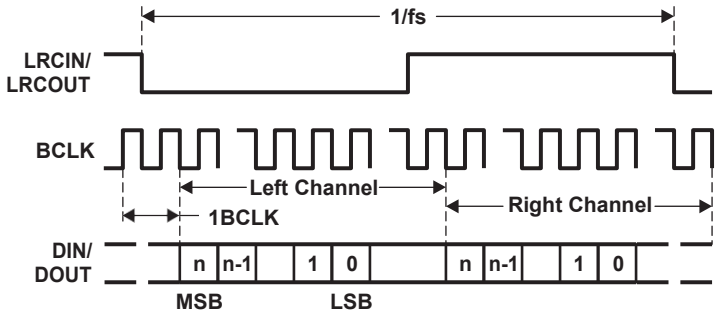


Figure 4. AIC23B I2S Mode Data Format

In the example software packaged with this application report, the configuration differences listed in Table 2 must be taken into consideration for AIC23B DSP versus I2S audio formats.

Table 2. Register Configuration Differences for AIC23B DSP vs I2S Formats

Register [bit]	Bit-Field Name	Binary Value	Description
DAUDINTF[1:0]	FOR	11	DSP format, frame sync + 2 words
		10	I2S format
DAUDINTF[4]	LRP	0	DSP - MSB ready on first BCLK rising edge after LRCIN rising edge I2S – Right channel on, LRCIN/LRCOUT high
		1	DSP – MSB ready on second BCLK rising edge after LRCIN rising edge I2S – Do NOT use LRP = 1
SMPLRCTL[0]	USB	0	I2S mode – external clock to achieve 64 clocks per active low on frame sync
		1	DSP mode – USB clock or external clock because number of clocks per frame sync is don't care
SMPLRCTL[5:1]	SR:BOSR	XXXXX	Depends on USB vs external clock frequencies and desired sampling rate

Due to the different requirements of AIC23B DSP and I2S audio formats, the differences in McBSP bit configurations are outlined in the register settings tables in the next section.

In AIC23_project.c, I2S-mode or DSP-mode software configurations can be selected along with microphone or line inputs using the following directives:

```
//=============================================================
// SELECT AUDIO INPUT AND DIGITAL AUDIO INTERFACE OPTIONS HERE:
//=============================================================

#define MIC 0      // 0 = line input, 1 = microphone input
#define I2S_SEL 0 // 0 = normal DSP McBSP dig. interface,
                   // 1 = I2S interface

//=============================================================
```

2.3.1 McBSP Setup

All data is transferred between the AIC23B and the F2833x via two McBSP modules: digital audio data is transferred to and from McBSP-A and command data is transferred from McBSP-B to the AIC23B.

2.4 McBSP-A Configuration

To see how McBSP-A is configured to support the AIC23B digital audio interface, see the init_mcbspa() function in AIC23_project.c. Because McBSP-A is interfaced to the DMA, and because both left-channel and right-channel data must be transferred between the AIC23B and the McBSP for every frame sync, McBSP-A must be configured as follows to handle audio data for this example:

- Interrupts - McBSP interrupts must be disabled because the DMA automatically handles data transfers between the McBSP data registers and SARAM. Although interrupts are disabled, the McBSP must be configured to generate an interrupt signal to the DMA module for every word transmitted (XRDY flag) and every word received (RRDY flag).

Table 3. McBSP Interrupt Enable Register (MFFINT) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
MFFINT[2]	RINTENA	0	Disable receive interrupts
MFFINT[0]	XINTENA	0	Disable transmit interrupts

Table 4. Serial Port Control 1 Register (SPCR1) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
SPCR1[14:13]	RJUST	10	Left-justify and zero-fill LSBs in DRR
SPCR1[5:4]	RINTM	00	RINTM flag driven by RRDY

Table 5. Serial Port Control 2 Register (SPCR2) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
SPCR2[5:4]	XINTM	00	XINTM flag driven by XRDY

- Receive/Transmit Format - The McBSP Receive Control Registers 1/2 (RCR1/RCR2) and McBSP Transmit Control Registers 1/2 (XCR1/XCR2) must be configured for a dual-phase frame with one 32-bit word in each phase (one left-channel word in phase 1 and one right-channel word in phase 2). The receive/transmit data delay is dependent on whether the McBSP is configured to transfer digital audio data in I2S mode or normal DSP mode.

Table 6. Receive/Transmit Control 1 Registers (R/XCR1) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
R/XCR1[7:5]	R/XWDLEN	101	32-bit word in phase 1 (L-channel)

Table 7. Receive/Transmit Control 2 Registers (R/XCR2) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
R/XCR2[7:5]	R/XWDLEN	101	32-bit word in phase 2 (R-channel)
R/XCR2[1:0]	R/TXDATDLY	0	DSP mode: AIC23B DAUDINTF[LRP]=0
		1	DSP mode: AIC23B DAUDINTF[LRP]=1
			I2S mode: data available on second rising edge of BCLK

- Clocking and Frame Sync: Because McBSP-A is slave to the AIC23B, the frame syncs and clocks are generated externally. To support the I2S and normal DSP audio data formats from the AIC23B, the McBSP must be configured such that receive data is sampled on the rising edge of CLKR and transmit data is sampled on the falling edge of CLKX. The CLKG divider in slave mode should be set to 1. The frame sync active polarity is dependent on whether the McBSP is configured to transfer digital audio data in I2S mode or normal DSP mode.

Table 8. Pin Control Register (PCR) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
PCR[11]	FSXM	0	Tx frame sync generated by AIC23B
PCR[10]	FSRM	0	Rx frame sync generated by AIC23B
PCR[9]	CLKXM	0	Tx driven by external clock on CLKX
PCR[8]	CLKRM	0	Rx driven by external clock on CLKR
PCR[7]	SCLKME	0	With SPCR2[CLKSM]=1, McBSP sample rate generator internally clocked by LSPCLK.
PCR[3]	FSXP	0	DSP mode – active high frame sync
		1	I2S mode – active low frame sync
PCR[2]	FSRP	0	DSP mode – active high frame sync
		1	I2S mode – active low frame sync
PCR[1]	CLKXP	0	Tx data sampled on falling edge of clock
PCR[0]	CLKRP	1	Rx data sampled on rising edge of clock

Table 9. Sample Rate Generator Register 1 (SRGR1) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
SPCR1[7:0]	CLKGDV	00000001	Sample rate generator clock divide by 1

Table 10. Sample Rate Generator Register 2 (SRGR2) Settings

Register [bit]	Bit-Field Name	Binary Value	Description
SPCR2[5:4]	CLKSM	1	With PCR[SCLKME] = 0, McBSP sample rate generator internally clocked by LSPCLK.

For more information on McBSP register and bit settings, see the *TMS320F2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7).

2.5 McBSP-B Configuration

To understand how McBSP-B is configured to support clock-stop/SPI mode to transmit command data via the AIC23B serial control interface, see the init_mcbsp_spi() function in AIC23_SPI_control.c. Alternatively, the SPI module on the F2833x could have been used to perform the same function. Some main considerations when configuring the McBSP/SPI are:

- McBSP-B must be configured for clock-stop mode with a polarity of 1 and phase of 0 (transmit on falling edge without delay and receive on rising edge)
- McBSP-B (as SPI) is the master and AIC23B is the slave. So, the clock and frame sync (chip select) are generated by the F2833x in this case, and XDATDLY and RDATDLY must be set to 1.

For detailed instructions on how to configure the McBSP registers for SPI/clock-stop mode, see the *TMS320F2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7).

2.5.1 DMA Setup

To expedite data transfers between the McBSP module and buffers in SARAM, the DMA must be configured to transfer data between McBSP Data Receive Registers 1/2 (DRR1/2) and McBSP Data Transmit Registers 1/2 (DXR1/2), and ping-pong buffers in SARAM. Also, because left-channel and right-channel data must be extracted separately, the ping and pong buffers are each split into a left-channel sub-buffer (first 512 32-bit address locations in each buffer) and a right-channel sub-buffer (second 512 32-bit address locations in each buffer). To see how the DMA is configured to interface with the McBSP for transferring audio digital data, see the init_dma() function in the AIC23_project.c file. With the help of the DMA, the data is easily sorted between buffers while the CPU handles the audio data processing with the following considerations:

- The DMA must be configured to accept peripheral interrupt signals and to trigger on McBSP receive and transmit event flags. Additionally, channel interrupts must be enabled. Enabling continuous mode allows the DMA to continuously expedite receive and transmit audio data transfers.
- When receiving data from the AIC23B via DMA channel 1, the McBSP data receive registers (DRR2/DRR1) are the DMA source and the SARAM buffer location (ping or pong buffer) is the DMA destination. When transmitting data to the via DMA channel 2, the SARAM buffer location (ping or pong buffer) is the DMA source and the McBSP transmit registers (DXR2/DXR1) are the DMA destination.
- One burst of data is two 16-bit words (or one 32-bit word). For instance, when receiving the first data word from the AIC23B, one 16-bit word is transmitted from DRR2 (MSB) and transferred by the DMA to ping buffer 16-bit address offset 0x0001. Then another 16-bit word is transmitted from DRR1 (LSB) and transferred by the DMA to ping buffer 16-bit address offset 0x0000. This is because a 32-bit read results in the larger address location being read first as the MSB, and the smaller address location being read second as the LSB.
- One transfer is 1024 bursts (1024 32-bit words – the size of one ping buffer or one pong buffer).
- After two bursts (left-channel, then right-channel data), a wrap must be performed to return to left-channel data. This time, the source or destination pointer handling the buffer must wrap to two 16-bit address locations, after the previous address location, to avoid over-writing (if receiving into buffer) or re-reading (if transmitting out of buffer) the previous 32-bit word processed.

Figure 5 through Figure 8 illustrate how the DMA handles data received from the McBSP into first the ping buffer, and then the pong buffer.

First, one burst of data is received. The source address increments by 1 and the destination address, which began at 16-bit address offset 0x0001 of the ping buffer, is decremented by 1 in the burst.

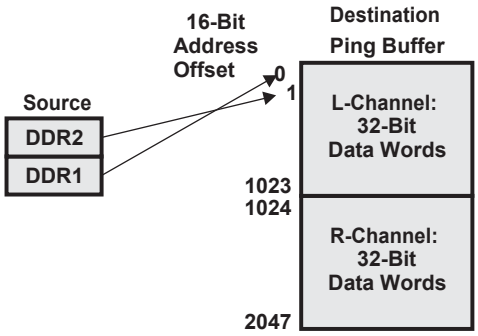


Figure 5. First Burst of Data (left channel) Received

Now, one word of left-channel data has been received into the ping buffer. The source address pointer has decremented by 1 (transfer step = -1) to return to DRR2, and the destination address pointer has incremented by 1025 16-bit address locations to reach the appropriate offset for the MSB of the first right-channel 32-bit word. The second burst is processed.

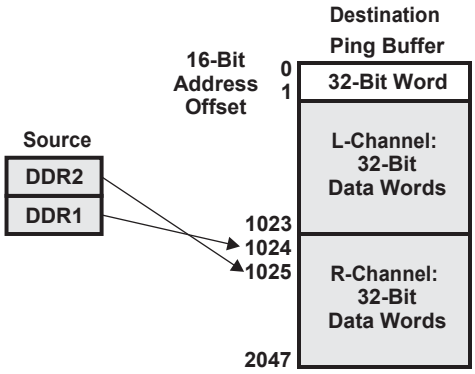


Figure 6. Second Burst of Data (right channel) Received

Next, one 32-bit word of left-channel data and one 32-bit word of right-channel data have been received into the ping buffer. The DMA is configured to wrap the destination address pointer back to the starting address (16-bit address offset 0x0001 in ping buffer), plus a wrap step size of 2 (16-bit address offset 0x0003 in ping buffer). Now the next 32-bit left-channel word can be received in a burst.

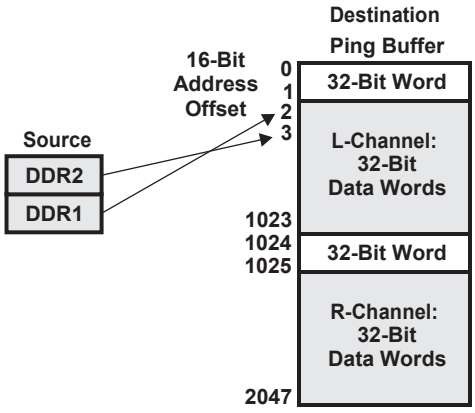


Figure 7. Address Pointer Wrap Back to Left Channel

Finally, 1024 bursts (one transfer of 1024 32-bit words or 2048 16-bit words) have been processed. Because a transfer has completed, DMA channel 1 will now interrupt. The interrupt service routine sets the destination address pointer to the pong-buffer and the DMA begins transferring data in bursts into the pong buffer, the same way it did for the ping buffer. Likewise, after the pong buffer transfer completes, the interrupt service routine sets the destination address back to the ping buffer.

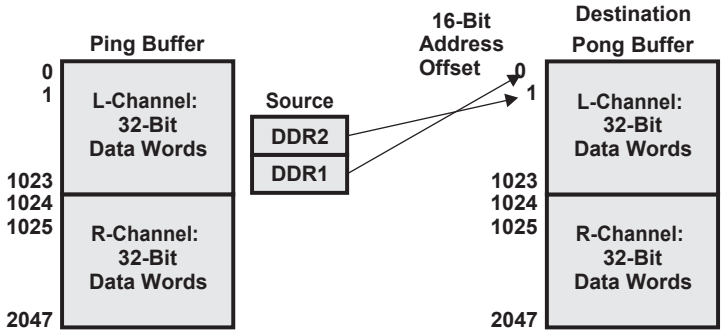


Figure 8. Ping Buffer Transfer Complete

3 Running the Application

After properly connecting the external hardware and configuring the McBSP and DMA modules appropriately in software, the audio application is ready to run.

1. Go to Workspace → Load Workspace. Navigate to the AIC23_project.wks file to load the workspace and run the included audio example project as-is from the Code Composer Studio File menu. Alternatively, you can open the audio example project directly by going to Project → Open, then navigate to the AIC23_project.pjt file.
2. Set the appropriate #define selections for the microphone or line input and I2S- or DSP-mode audio format after the workspace has loaded in the AIC23_project.c file.
3. Build and load the code. To see the audio data changing in real-time, set the debug mode for *Real-time Mode* (Debug → Real-time Mode), right click on the memory windows and/or graphs and select *Continuous Refresh*.
4. Run the project (Debug → Run). You should be able to hear the audio from headphones or speakers.

Figure 9 shows the AIC23B audio example project workspace while the program is running with a sine wave audio signal fed via line input to the AIC23B. The 32-bit left- and right-channel digital audio data are displayed on graphs in the middle of the screen, and the corresponding data values appear in the memory windows to the right of the screen.

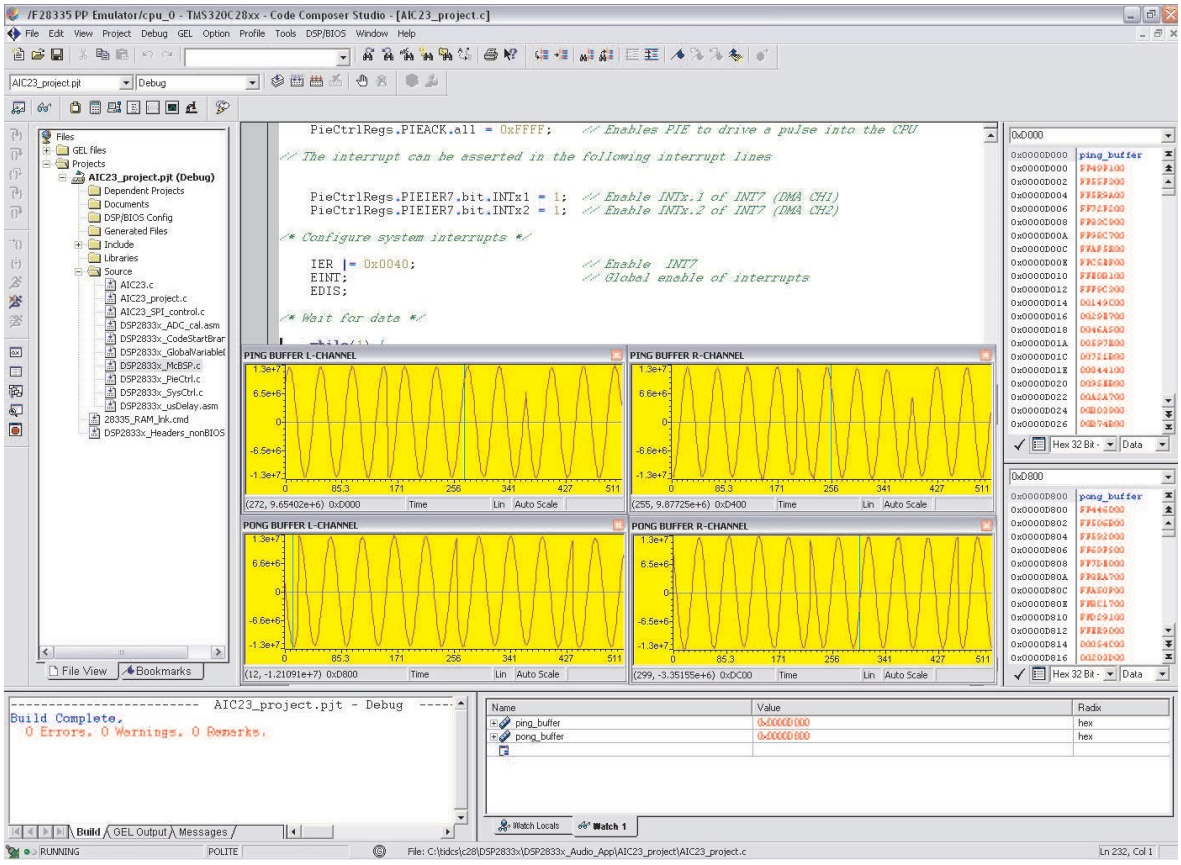


Figure 9. AIC23B Project Workspace – Sine Wave Digital Data

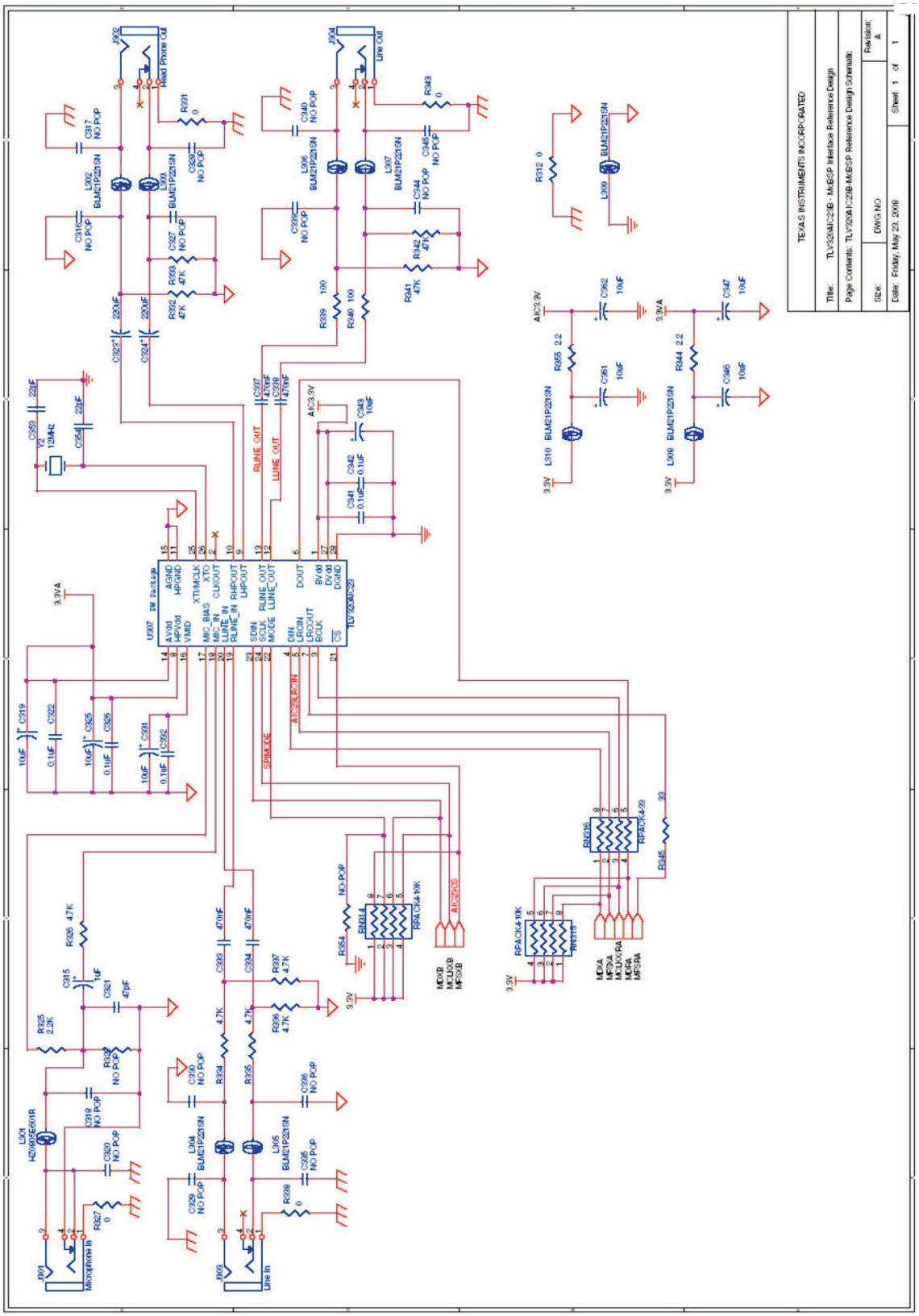
4 Conclusion

By utilizing the DMA on the F2833x devices with the McBSP module, audio data transfers are easily handled without tying up precious CPU resources, leaving the CPU free to process the converted digital data and perform other tasks. The TLV320AIC23B audio stereo codec is ideal for interfacing with the F2833x due to its DSP mode, which is 100% compatible with the McBSP module. Because of the McBSP module's flexibility, with the appropriate clock settings, the F2833x also supports the AIC23B's I2S audio format mode. Using the software package included with this application report, which configures the McBSP and DMA modules on the F2833x and utilizes AIC23B drivers to transmit commands over a serial interface, you can immediately set up and run a simple audio solution on the F2833x in either DSP or I2S audio format modes.

5 References

- *TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232 Digital Signal Controller (DSC) Data Manual* (SPRS439)
- *TLV320AIC23B Stereo Audio CODEC, 8- to 96-kHz, With Integrated Headphone Amplifier Data Manual* (SLWS106)
- *TMS320x2833x Direct Memory Access (DMA) Reference Guide* (SPRUFB8)
- *TMS320F2833x Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUFB7)
- TI F28x Development Tools Web Page: http://www.ti.com/f28xkits

Appendix A TLV320AIC23B to McBSP Interface Reference Design Schematic



IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated