TEXAS INSTRUMENTS

Application Report
SPRA477A - August 2001

TMS320C6201/6701 EVM: TMS320C6000 McBSP to Multimedia Audio Codec Interface

Brian G. Carlson and Vassos Soteriou *DNA Enterprises, Inc. – Texas Instruments*

ABSTRACT

This application note describes how a multimedia audio codec can be interfaced to the TMS320C6201/C6701 DSPs. Although this application report uses the CS4231A audio codec as an example, a part that is obsolete, this application note can be used as a reference guide in interfacing similar audio codecs to the TMS320C6000™ McBSP. Cirrus Logic offers the CS4235 CrystalClear™ ISA audio device that provides similar functionality to the CS4231A. Specifically, this application report addresses the digital interface between these devices and the TMS320C6000 DSP, using the serial interface for audio data transfer and the parallel interface for control and status access, referencing the CS4231A as an example device. The CS4231A codec's digital audio data can be directly interfaced to the DSP's McBSP for efficient data transfers that do not contend for the DSP's EMIF. The codec can be controlled and monitored by the DSP via the codec's parallel interface which can be memory-mapped directly to the EMIF's asynchronous interface with no glue logic. EMIF access to the codec allows an external processor to also control and monitor it via the DSP's host port interface. The application note also identifies other codec digital interface signals that can be useful to the DSP in some applications.

List of Figures

List of Tables

1 Introduction

The TMS320C6000 DSP can interface to the CS4231A multimedia audio codec in several ways since the codec provides both parallell, serial, and interrupt support. The most efficient interface method takes advantage of the CS4231A's serial audio data port which is compatible with the TMS320C6000 multichannel buffered serial port (McBSP). This interface method provides a dedicated path for the serial audio data so that it does not continuously contend for bandwidth on the DSP's parallel, external memory interface (EMIF).

The codec's parallel control interface can be memory–mapped into the DSP's EMIF for control and status access, which is used during initialization or infrequently during normal operation. The use of a separate control interface to the codec simplifies software driver support. The codec driver does not have to deal with the complexities and additional memory requirements associated with data and control/status information multiplexed in a single stream. As an added benefit, a memory-mapped codec interface enables an external processor to control and monitor the codec through the DSP's host port interface (HPI). This direct access to the codec from a host processor provides flexibility that could be useful in some applications.

This application note specifically addresses the digital interfaces between the DSP and the codec using the serial interface for audio data and the parallel interface for control and status. Other parallel interface methods are possible, but they are not optimal for the TMS320C6000. The analog interface of the CS4231A, and other functional aspects not directly related to the digital interface to the DSP, are not within the scope of this application note. The CS4231A data sheet[4] provides extensive information about the device and should be referenced for further details.

Figure 1 shows the digital interfaces between a TMS320C6000™ DSP and the CS4231A codec.

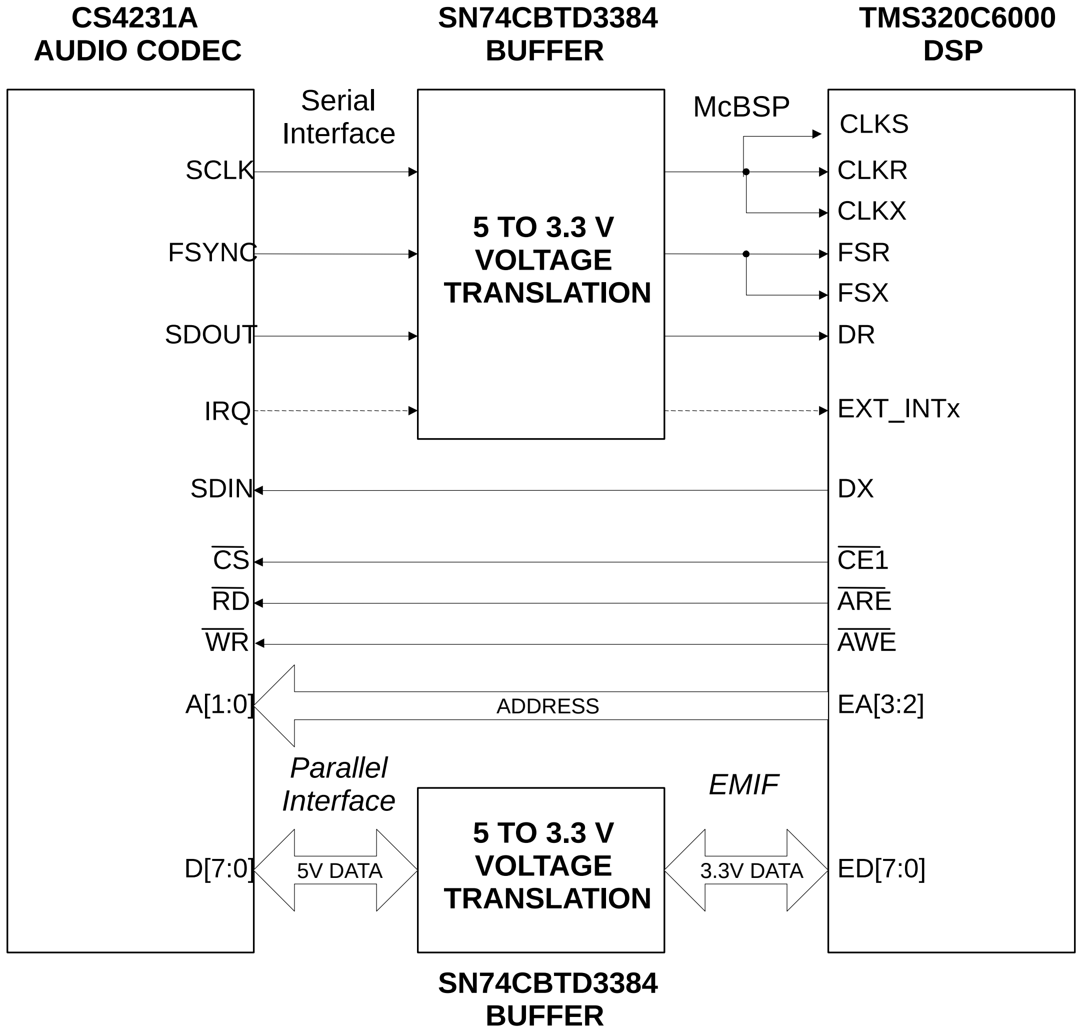

Figure 1. TMS320C6000 Interface to the CS4231A Multimedia Audio Codec

The digital interfaces between the codec and the DSP consist of a serial interface that connects the codec to the DSP’s McBSP and a parallel interface that connects the codec to the EMIF’s asynchronous mode. It is important to note that the CS4231A is a 5V device, so any signals that it provides to the 3.3V I/O DSP must be translated using devices such as the TI SN74CBTD3384 bus switches. Signals originating from the DSP do not need to be translated since their V_{OH} and V_{OL} values are compatible with the codec.

The codec always generates the serial data clock and frame sync signals. After voltage translation, the serial data clock is connected to all three of the DSP’s McBSP clock pins, which should be programmed as inputs. Similarly, the frame sync signal is connected to both FSR and FSX, which should also be programmed as inputs. This configuration means that both transmit and receive data are synchronized with phase alignment.

The codec’s parallel interface consists of an 8-bit data bus, 2-bit address bus, chip select, read strobe, and write strobe. The EMIF’s lower eight bits (ED[7:0]) are connected to the codec’s data bus using a voltage translation buffer. The DSP’s EA[3:2] address signals are mapped directly to the codec’s A[1:0] address signals, so the codec registers are mapped on 32–bit word boundaries with only the lower eight data bits being valid. The EMIF asynchronous control signals are directly connected to the codec’s chip select and read/write strobes since the EMIF memory space control register can be programmed with timing characteristics that match the requirements of the codec.

An optional interrupt connection is shown in Figure 1. This interrupt connection is independent of both the serial and parallel interfaces and is not required for codec operation. The codec supports an active, high-interrupt output that can be driven from an internal 16-bit timer. This interrupt may be useful in some applications as an independent watchdog timer that can be connected to one of the DSP’s EXT_INTx inputs. If this interrupt is used, the DSP’s default rising-edge interrupt polarity should be used.

2 Serial Interface

2.1 Timing

The codec’s serial port timing is directly compatible with the TMS320C6000 McBSP. The McBSP is designed to directly interface with devices such as the CS4231A audio codec that present a serial clock, frame sync and data. The codec generates the serial data clock that operates at a rate of 64 times the sample rate. For the maximum sample rate of 48KHz, the serial clock is 3.072 MHz, which is far below the maximum of 100 MHz for a 200 MHz DSP. The codec transitions SDOUT data on the rising edge of the SCLK and samples SDIN data on the falling edge, which is the default operation of the McBSP. In the serial data streams, the codec’s left channel data is always before the right channel data, and the most–significant bit of each element is transmitted first which is compatible with the McBSP. Both left and right channels’ elements are always 16 bits wide with the actual audio data being left justified in the element. For example, 8-bit companded data would occupy the first eight bits. Unused bits are output as zeros after the least-significant bit.

The codec supports three types of serial data formats that can all be used with the flexible McBSP. However, the codec’s default 64-bit enhanced mode serial format is recommended, since it provides the most flexibility without incurring additional overhead. The 64-bit enhanced mode serial format is directly compatible with the McBSP, and is configured for positive frame sync pulses with a 1-bit data delay. In this mode, each frame has 64 data bits. The first 16 bits of data represent the left channel and the second 16 bits represent the right channel. The last 32 bits can be optionally used to monitor the codec’s interrupt, capture enable, playback enable, and over-range indicators. If an application does not need the extra status information, then the number of elements per phase (frame) can be set to just one 32-bit word. For mono applications, the number of elements could be set to just one 16-bit word. If the status information is not needed, the 32-bit mode serial format could be used. In this mode, the clock still runs at 64 times the sample rate, and the frame consists of 64 bits times. However, the SCLK and the SDOUT are held low during the last 32 bits of the frame. Stopping the SCLK is compatible with the McBSP since the clock is stopped after the 32 bits are already transferred. The 32-bit mode is used by the example applications for the TMS320C6201/C6701 EVMs since the additional status information is not required.

The serial port timing requirements are met by one or two orders of magnitude for all parameters at the maximum sample rate of 48 kHz (3.072 MHz SCLK), so there are no critical timing parameters for the serial interface. The timing analysis is shown in Table 1.

Table 1. TMS320C6201 Serial Port and CS4231A Codec Timing Analysis

CS4231A Timing Requirements		**McBSP Switching Characteristics**		**UNIT**
tS1 (min) SDIN valid to SCLK falling	30	(tSCLK/2) – td(CKXH–DXV)max= 162 – 16	146	ns
tH1 (min) SDIN hold after SCLK falling	30	(tSCLK/2) + td(CKXH–DXV)min= 162 + 3	165	ns
McBSP Timing Requirements		**CS4231A Switching Characteristics**		
tsu(DRV–CKRL)min Setup Time, DR valid before CLKR(ext) low	1	(tSCLK/2) – tPD1(max)= 162 – 30	132	ns
th(CKXL–DRV)min Hold Time, DR valid after CLKR(ext) low	4	(tSCLK/2) + tPD1(min)= 162 + 0 ; tpD1(min) not available	162	ns
tsu(FRH–CKRL)min Setup Time, ext. FSR high before CLKR(ext) low	4	(tSCLK/2) – tPD2(max)= 162 – 20	142	ns
th(CKRL–FRH)min Hold Time, ext. FSR high after CLKR(ext) low	3	(tSCLK/2) + tPD2(min)= 162 + (–20)	142	ns
tsu(FXH–CKXL)min Setup Time, ext. FSX high before CLKX(ext) low	4	(tSCLK/2) – tPD2(max)= 162 – 20	142	ns
th(CKXL–FXH)min Hold Time, ext. FSX high after CLKX(ext) low	3	(tSCLK/2) + tPD2(min)= 162 + (–20)	142	ns

Figure 2 shows the codec's 64–bit enhanced mode serial format.

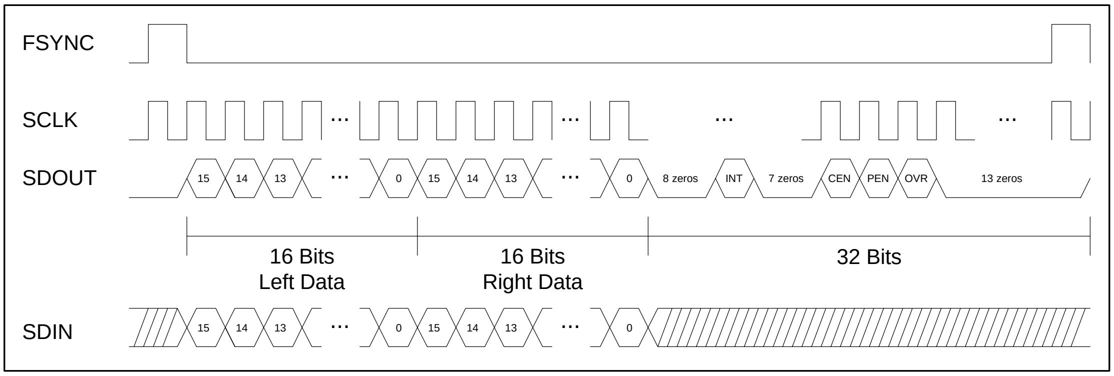

Figure 2. CS4231A 64-bit Enhanced Mode Serial Timing

2.2 Codec Registers

The CS4231A includes four direct access (R0–R3) and 32 indirect access (I0–I31) registers that are used by the DSP to initialize, control, and monitor the codec. Only a few register bits must be initialized in a certain manner in order to support the digital serial interface between the codec and the DSP. The following paragraphs in this section highlight these specific register bits. Refer to the CS4231A data sheet for details on its registers and how to use them.

To enable the CS4231A codec's serial interface, which is one of its expanded mode features, the MODE2 bit in the MODE and ID register (I12) must be initialized to 1. This bit allows access to indirect registers 16–31, that are required to enable and configure the serial interface.

The Alternate Feature Enable register (I16) must be initialized to enable the serial port and to select the serial data format. The Serial Port Enable (SPE) bit must be set to 1 to enable the serial interface. When the serial port is enabled, digital audio data from the ADCs is sent out on SDOUT, and audio data from SDIN is sent to the DACs. The Serial Format bits (SF1, SF0) should both be set to 0 to select the default, 64-bit enhanced serial data format. The MCE bit in the Index Address Register (R0) must be set before the SPE, SF1 and SF0 register bits can be changed.

The Playback Data Format (I8) and the Capture Data Format (I28) registers are used to select the digital audio data format. The S/M bit in these registers selects either Mono (0) or Stereo (1) data streams. In mono mode, the left and right channels have the same data. In stereo mode, alternating samples represent the left and right audio channels. The C/L, FMT1 and FMT0 bits set the audio data format as shown in Table 1. In all data formats, the data are always sent MSB first, and is left-justified in each 16-bit element. For example, 4-bit ADPCM data would occupy the first four bits and would be followed by 12 zeros. The MCE bit in the Index Address Register (R0) must be set before the C/L, FMT1 and FMT1 register bits can be changed.

The playback and capture bits (PEN/PPIO/CEN/CPIO) in the Interface Configuration (I9) register should be set to 1 to enable codec operation. Other codec registers can be initialized as needed for the particular application.

Table 2. CS4231A Digital Audio Format Selection

FMT1	FMT0	C/L	Data Format
0	0	0	Linear, 8–bit unsigned
0	0	1	μ–Law, 8–bit companded
0	1	0	Linear, 16–bit, 2's comp.
0	1	1	A–Law, 8–bit companded
1	0	0	RESERVED
1	0	1	ADPCM, 4–bit IMA
1	1	0	RESERVED
1	1	1	RESERVED

2.3 DSP Registers

McBSP configuration registers must be initialized to support the serial interface with the CS4231A audio codec. There are several McBSP register bits, but this section only addresses the ones that are directly related to the serial interface to the codec. The following serial interface characteristics, relative to the DSP, must be addressed by the McBSP register bits initialization:

- External frame syncs (FSXM=FSRM=0)
- External serial clocks (CLKRM=CLKXM=0)
- Positive frame sync polarities (FSXP=FSRP=0)
- Positive serial clock polarities (CLKXP=0, CLKRP=1)
- Single phase frames (RPHASE=XPHASE=0)
- One element per phase (RFRLEN1=XFRLEN1=0000000b)
- 16 bits per audio channel
- Most significant bit first
- Left-justified with right zero fill (RJUST=10b)
- 16-bit (Mono) or 32-bit (Stereo) element length (For 16-bit mono, RWDLEN1=010b, XWDLEN1=010b; for 32-bit stereo, RWDLEN1=101b, XWDLEN1=101b)
- Companding to match selected codec data format (RCOMPAND, XCOMPAND)
- 1-bit data delay (RDATDLY=01b, XDATDLY=01b)

The following steps describe the setup of the DMA, McBSP, and interrupts in the required order. The McBSP initialization procedure that uses the CPU or DMA is also discussed in the application report literature number SPRA488, *TMS320C6000 McBSP Initialization*.

1. The McBSP of the C6000 DSP should be properly initialized before it is enabled. If the McBSP is not already in the reset state, the /XRST and /RRST in the SPCR should be set to be equal to 0.
2. The McBSP configuration registers XCR, RCR, SRGR, PCR, and SPCR should then be initialized with the required parameters as shown in the bulleted points above. The Serial Port Control Register (SPCR) should be initialized to disable the frame sync and sample rate generators (/FRST=/GRST=0) since these are provided by the codec.
3. The DMA channels should be hooked up to the interrupt service routines. One DMA channel should be configured to transfer data from the codec to the McBSP. A second DMA channel should be set up for data transfers from the McBSP to the codec. The CPU interrupts that correspond to the DMA channels that will be used to service the McBSP should then be enabled. The default mapping of DMA channel-complete interrupts to the CPU is the following:
   - DMA channel 0 à CPU interrupt 8
   - DMA channel 1 à CPU interrupt 9
   - DMA channel 2 à CPU interrupt 11
   - DMA channel 3 à CPU interrupt 12

4. Either a or b should be followed:

   a) This step should be performed if the CPU is used to service the McBSP. The XRST and RRST signals should be set to 1 to enable the serial port. Note that the value written to the SPCR at this time should only have the reset bits changed to 1 and the remaining bit fields should have the same value as in Step 2 above.

   b) If the DMA is used to perform data transfers, it should first be initialized with the appropriate read/write syncs, src/dst addresses and their update modes, transfer complete interrupt, and any other feature suitable for the application. Lastly, the START bit should be set. The DMA is now in the START state and waits for the synchronization events to occur. Then, the McBSP should be pulled out of reset. For details on DMA initialization for servicing the McBSP, refer to *TMS320C6000 McBSP Initialization* (literature number SPRA488) and *TMS320C6000 DMA Applications* (literature number SPRA529). The McBSP transmitter and receiver should then be enabled (/RRST=/XRST=1) in order for the McBSP to reinitialize.

All of the above McBSP registers and their bit–field values are shown in Figure 3 through Figure 7.

31	30 24	23 21	20 19	18	17 16
RPHASE	RFRLEN2	RWDLEN2	RCOMPAND	RFIG	RDATDLY
0	0	0	0	0	01

15	14 8	7 5	4	3 0
reserved	RFRLEN1	RWDLEN1	RWDREVRS	reserved
0	0	010	0	0

Figure 3. Receive Control Register (RCR)

31	30 24	23 21	20 19	18	17 16
XPHASE	XFRLEN2	XWDLEN2	XCOMPAND	XFIG	XDATDLY
0	0	0	0	0	01

15	14 8	7 5	4	3 0
reserved	XFRLEN1	XWDLEN1	XWDREVRS	reserved
0	0	010	0	0

Figure 4. Transmit Control Register (XCR)

31	30	29	28	27	26 16
GSYNC	CLKSP	CLKSM	FSGM	FPER	FPER
0	0	0	0		0

15 10	0
FWID	CLKGDV
0	0

Figure 5. Sample Rate Generator Register (SRGR)

31 16
reserved
0x0000

15	13	12	11	10	9	8
rsv	XIOEN	RIOEN	RSXM	FSRM	CLKXM	CLKRM
0	0	0	0	0	0	0

7	6	5	4	3	2	1
rsv	CLKS_STAT	DX_STAT	DR_STAT	FSRP	CLKXP	CLKRP
0	0	0	0	0	0	1

Figure 6. Pin Contrl Register (PCR)

31 24	23	22	21 20	19	18	17	16
0x00	FRST	GRST–	XINTM	XSYNCERR	XEMPTY–	XRDY	XRST
R, +0	0	0	0	0	0	0	0

15	14 13	12 11	10 8	7	6	5 4	3	2	1	0
DLB	RJUST	CLKSTP	rsv	DXENA	rsv	RINTM	RSYNCERR	RFULL0	RRDY	RRST
0	10	0	0	0	0	0	0	0	0	0

Figure 7. Serial Port Control Register (SPCR)

2.4 Parallel Interface

The codec's parallel interface is directly compatible with the EMIF's asynchronous mode. When an EMIF's CE space is configured for asynchronous operation, the DSP provides a high degree of programmability for shaping accesses. The programmable parameters include setup, strobe, and hold times. Setup time is the time between the beginning of a memory cycle and the activation of the read (ARE) or write (AWE) strobe. Strobe time is the time between the activation and deactivation of the read or write strobe. Hold time is the time between the deactivation of the read or write strobe and the end of the cycle. The codec's parallel interface timing is defined by the read cycle timing diagram shown in Figure 3 and the write cycle timing diagram shown in Figure 4. Table 2 provides the parametric timing indicated in the timing diagrams.

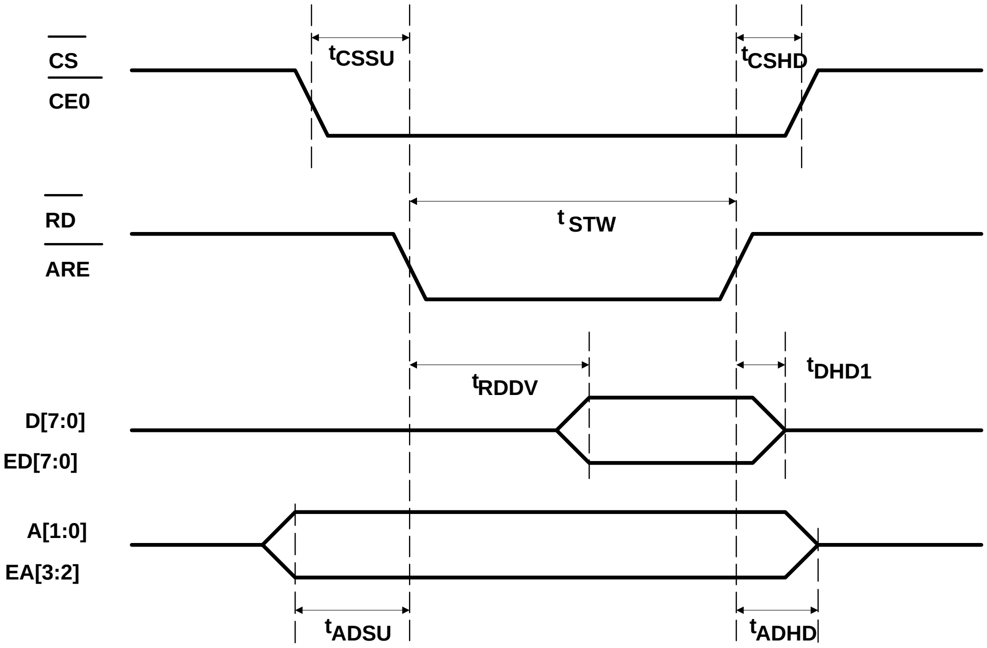

Figure 8. CS4231A Parallel Interface Read Timing

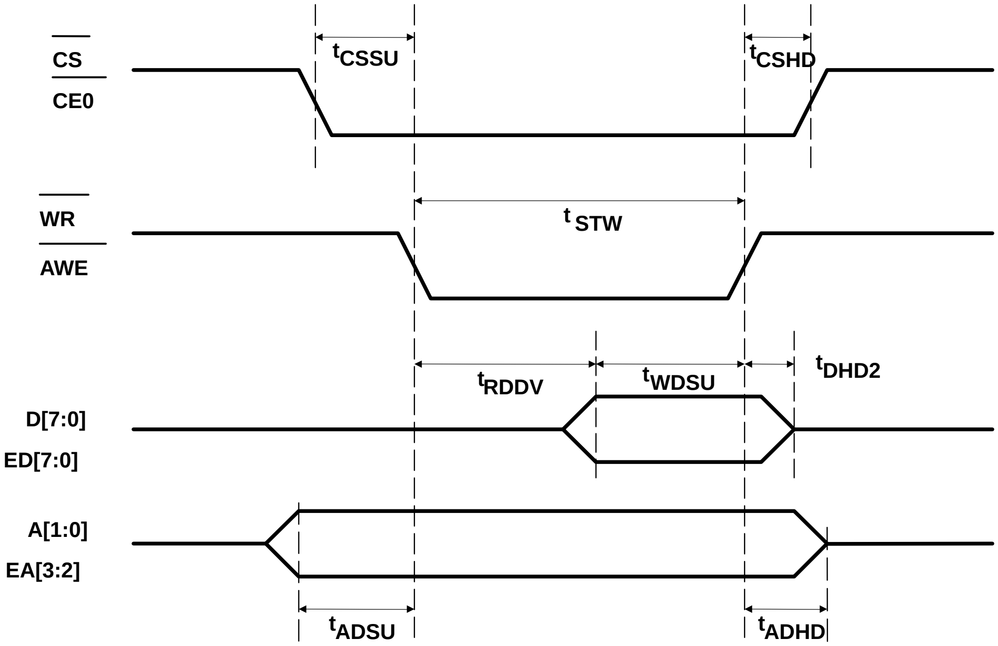

Figure 9. CS4231A Parallel Interface Write Timing

Table 3. CS4231A Parallel Interface Parametric Timing

Parameter	Symbol	Min (ns)	Max (ns)
CS setup to WR or RD falling edge	t_{CSSU}	10	
CS hold from WR or RD rising edge	t_{CSHD}	0	
WR or RD strobe width	t_{STW}	90	
RD falling edge to data valid	t_{RDDV}		60
Data hold from RD rising edge	t_{DHD1}	0	20
Data valid to WR rising edge	t_{WDSU}	22	
Data hold from WR rising edge	t_{DHD2}	15	
Address setup to RD or WR falling edge	t_{ADSU}	22	
Address hold from WR or RD rising edge	t_{ADHD}	10	

31 28	27 22	21 20	19 16
WRITE SETUP	WRITE STROBE 10010	WRITE HOLD	READ SETUP
0101	10010	011	0101

15 14	13 8	7	6 4	3 2	1 0
rsv	READ STROBE	rsv	MTYPE010	reserved	READ HOLD
R, +11	10010	R, +0	010	R, +0	011

Figure 10. EMIF CE0 Space Control Register Diagram

The start of the asynchronous EMIF access to the codec's parallel interface begins with asserting chip enable (CE0#) for the memory space. The codec has two setup requirements prior to the falling edge of the read or write data strobe. The chip select must be set up at least 10 ns, and the address must be setup at least 22 ns, before the data strobe's falling edge. Since the EMIF CE0# and address signals transition on the same CLKOUT1 edge, the worst case setup time that has to be met is 22 ns. This time period defines the asynchronous setup time that should be programmed in the CE0# memory space control register which is located at address 0x01800008. Assuming a DSP clock of 200 MHz, with a CLKOUT1 period of 5 ns, the read and write setup fields of this control register should be set to 5. The codec requires a minimum read or write data strobe period of at least 90 ns. This means that the read and write strobe fields of the control register should be set to 0x12 (10010b).

During a read access, the codec register data is available after a maximum of 60 ns into the 90 ns strobe period. This means that there is a data setup of 30 ns for the DSP, which easily meets the 5 ns setup requirement. The codec's data hold time is 0 ns minimum after the rising edge of the read strobe. Since the rising edge of the read strobe is after the edge that the DSP samples the data, this is not a problem.

During a write access, the codec requires that the data be set up at least 22 ns before the rising edge of the write strobe. Since the DSP provides valid data at the beginning of the memory cycle for writes, there is approximately a 115 ns setup time. The codec requires that write data be held for at least 15 ns after the rising edge of the write strobe. Since the DSP holds the data valid until the end of the hold period, the length of the hold period must be at least 15 ns.

The codec's chip select does not have to be held past the rising edge of the read or write data strobe. However, the address signals must be held for at least 10 ns after the data strobe's rising edge. The worst-case hold time is, therefore, defined by the write data hold time of 15 ns. At 200 MHz, this means that the read and write hold fields of the control register should be set to 0x3 (011b).

One codec timing parameter, not shown in the timing diagrams but important to note, is that there must be at least 80 ns between the rising edge of a read or write strobe and the next falling edge of a read or write strobe. This means that codec accesses need to be controlled, either by hardware or software, if the application would ever attempt to perform sequential register accesses to the codec. A hardware approach can be as simple as increasing the setup and strobe periods in the EMIF memory space control register to 13 and 3 CPU clocks respectively. At 200 MHz, this total of 16 clocks between the end of one data strobe and the beginning of the next would meet the 80 ns requirement. A more elaborate solution would use programmable logic to manage the codec interface signals and ensure that back-to-back accesses meet the minimum time period. This solution may be required anyway, if additional asynchronous devices were required in the same memory space. A software approach would require a delay to be inserted between codec accesses to ensure that the 80 ns requirement is met.

If the codec cannot be allocated to its own asynchronous EMIF memory space, such as CE0#, it can coexist with other asynchronous devices using programmable logic, which manages the codec interface signal timing and provides the required ready (ARDY) signal generation. This approach was taken on the TMS320C6201/6701 EVM since multiple devices, including a daughter board interface, had to coexist on the asynchronous interface. When multiple asynchronous devices with different timing requirements are in a design, programmable logic is required to handle these timing differences through the use of the ARDY signal.

2.5 Other Interface Signals

There are a few other codec signals independent from the serial and parallel interfaces that could be interfaced to the DSP in some applications. As mentioned previously, the codec's IRQ interrupt output can be interfaced to one of the DSP's external interrupt inputs. The codec provides an on-chip timer that can be used as an independent timing source or as a watchdog timer. The codec's power-down (/PDWN) input can be controlled by the DSP to disable the device and put it into a low-power mode. The codec provides two general-purpose output signals (XCTL1/XCTL0) that can be controlled by the DSP via writes to the codec's Pin Control (I10) register.

3 References

1. *TMS320C6201 Data Sheet*, Texas Instruments, SPRS051G January 1997, revised November 2000.
2. *TMS320C6000 Peripherals Reference Guide*, Literature Number SPRU190D, February 2001.
3. *CS4231A Parallel Interface, Multimedia Audio Codec Data Sheet*, Crystal Semiconductor, September 1994.
4. *TMS320C6201/C6701 Evaluation Module Reference Guide*, SPRU269D, December 1998.

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 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 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, license, warranty or endorsement thereof.

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. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's 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, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265