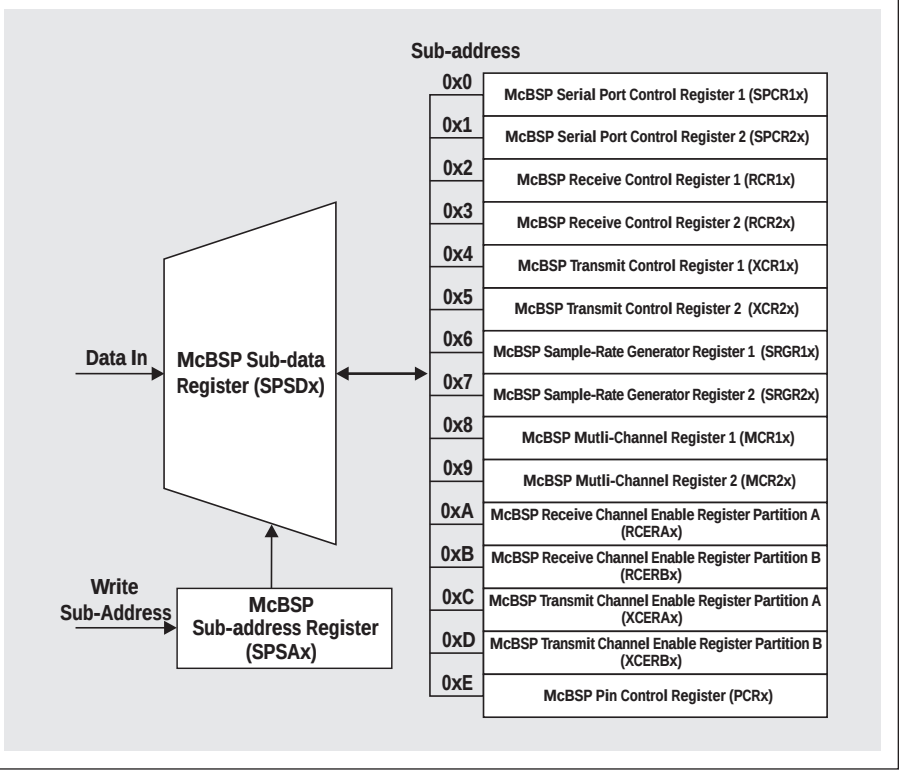
Efficiently interfacing serial data converters to high-speed DSPs

By Lijoy Philipose
Application Specialist, Data Converters

Introduction

Efficiently interfacing serial analog-to-digital data converters to TMS320C54xx digital signal processors (DSPs) can seem like a daunting task. Typically, these data converters are ten times slower than the high-speed DSPs interfaced to them. In high-performance, time-critical applications, asking the DSP to service these slower off-chip devices is a waste of processing power. The desirable solution is to transfer the task to an off-chip peripheral device. The direct memory access (DMA) controller used in conjunction with the multi-channel buffered serial port (McBSP) peripheral provides such a solution. This system will interface to the slower data converter by reading and storing samples into memory for DSP processing. The goal here is to introduce the McBSP and the DMA-enhanced peripherals and to provide an outline for interfacing serial data converters to the McBSP of the C5402 DSK. This article highlights how to use the McBSP and DMA to read samples from a serial analog-to-digital converter (ADC), specifically the TLV2548 ADC. The advantage of this method is that it frees the DSP for work more deserving of its power and speed.

Figure 1. McBSP sub-address scheme



McBSP

The multi-channel buffered serial port is a superset of the standard serial ports found on TI's DSPs. The McBSP not only has features found on previous serial port interfaces but also can direct interfacing to TI/E1 framers, IOM-2 compliant devices, MVIP-switching compatible and ST-BUS compliant devices, AC97 compliant devices, IIS compliant devices, and SPI devices. It provides a wide selection of transmit/receive data sizes, µ-law and A-law companding, programmable polarity for both frame synchronization and data clocks, and a highly programmable internal clock and frame generation. These features and programming requirements are described in Reference 6.

Let's look at how the McBSP registers are accessed. The McBSP registers are memory-mapped using a register sub-addressing scheme. Figure 1 shows a visual representation of this scheme. Register sub-addressing involves multiplexing a set of registers to a single location in the memory map. A sub-bank address register is used to control the multiplexer. A sub-data register (SPSDx) is used to read or write data to the desired sub-addressed register. To access a specific sub-addressed register, the register's sub-address location is written into the sub-address register (SPSAx). This directs the multiplexer to connect to the desired physical location in memory. When a write access occurs, data written to the sub-data register is moved to the embedded data register specified in the sub-address register. Similarly, for a read access, the contents of the register specified in the sub-address register are moved to the sub-data register.

Take, for example, McBSP0. The sub-data register (SPSD) is at location 0x039, and the sub-address register (SPSAx) is at location 0x038 in physical memory. The following sample code writes 0x000 to Serial Port Control Register 1 of McBSP0.

```
SPSA0        .set   038h   ;McBSP0 Sub-Address Register
SPSD0        .set   039h   ;McBSP0 Sub-Data Register
SPCR10_SUB   .set   000h   ;McBSP0 Serial Port Control
                             Register 1 sub-address

mmr(#SPSA0) = #SPCR10_SUB
mmr(#SPSD0) = #0000h
```

There are 16 registers associated with the McBSP. Interfacing a single TLV2548 ADC to the McBSP requires the proper configuration of only the following 9 registers. The serial port is configured by properly initializing the following registers:

1. *Serial Port Control Register 1 (SPCR1)* contains the McBSP receiver status bits and the main switch to enable or disable the receiver. This register includes the clock stop mode bit, which sets the serial port for various clocking modes for SPI and non-SPI schemes. Also included in SPCR1 is the ABIS mode bit and receiver interrupt mode bit.
2. *Serial Port Control Register 2 (SPCR2)* contains the McBSP transmitter status bits and the main switch to enable or disable the transmitter. This register also contains the bits to reset the frame-sync generator and the sample-rate generator.
3. The *Pin Control Register (PCR)* contains the bits to configure the McBSP pin as inputs or outputs during normal serial port operation. This is used to reconfigure the serial port pins as general-purpose inputs or outputs when the receiver or transmitter is disabled. PCR configures the transmitter and receiver clock and frame sync modes. For example, these bits determine whether CLKX/R and FSX/R are input/output pins and what their polarity is.
4. *Receive Control Register 1 (RCR1)* contains the bits to configure various options of the receiver. The value of this register determines the receiver word size (between 8 and 32 bits) and the number of words per frame (1 to 128) expected per receiver event.
5. *Receive Control Register 2 (RCR2)* determines the size of the word received and the bit delay after the frame-sync pulse. This register configuration plays an essential role if transfers greater than 16 bits and multiple phases are necessary. In this application the register bit of interest is the data-bit delay. The RCR2 register bits also select between µ-law, A-law companding, or whether the MSB or the LSB is transferred first for non-companding 8-bit transfers.
6. *Transmit Control Register 1 (XCR1)* contains the bits that determine the transmit word length and frame size. The transfer can be 8 to 32 bits wide and anywhere from 1 to 128 words long.
7. *Transmit Control Register 2 (XCR2)* contains the bits that determine the transmit data delay and select between µ-law, A-law companding, or whether the MSB or the LSB is transferred first for non-companding 8-bit transfers.

8. & 9. *Sample-Rate Generator Register 1 (SRGR1)* and *Sample-Rate Generator Register 2 (SRGR2)* control the sample-rate generator. The sample-rate generator is composed of a three-stage clock divider that allows programmable data clocks (CLKG) and framing signals (FSG). These two McBSP internal signals can be programmed to drive receive/transmit clock (CLKR/X) and receive/transmit data framing (FSR/X). Sample-rate generator registers (SRGR[1,2]) control the operation of the various features of the sample-rate generator. These registers are used to control the width of the frame-sync pulse and to determine whether frame-sync is an external input driven by a sample-rate generator or a signal indicating that a data copy from DXR(1,2) to XSR(1,2) has been made. These registers control whether the sample-rate generator clock is derived from the CPU clock or CLKS pin, and by what value to divide the CPU clock to produce the desired serial clock (CLKX/R).

Describing all 9 registers in detail is not appropriate in this forum. Detailed descriptions of all the register bits are provided in Reference 6.

The McBSP registers are initialized to the same value regardless of whether the CPU or the DMA acts as interface controller. Table 1 is a summary of the McBSP register values typically used in most serial interfaces. These same settings can be used for most serial data converters that provide less than 16 bits of resolution.

Continued on next page

Table 1. McBSP register settings

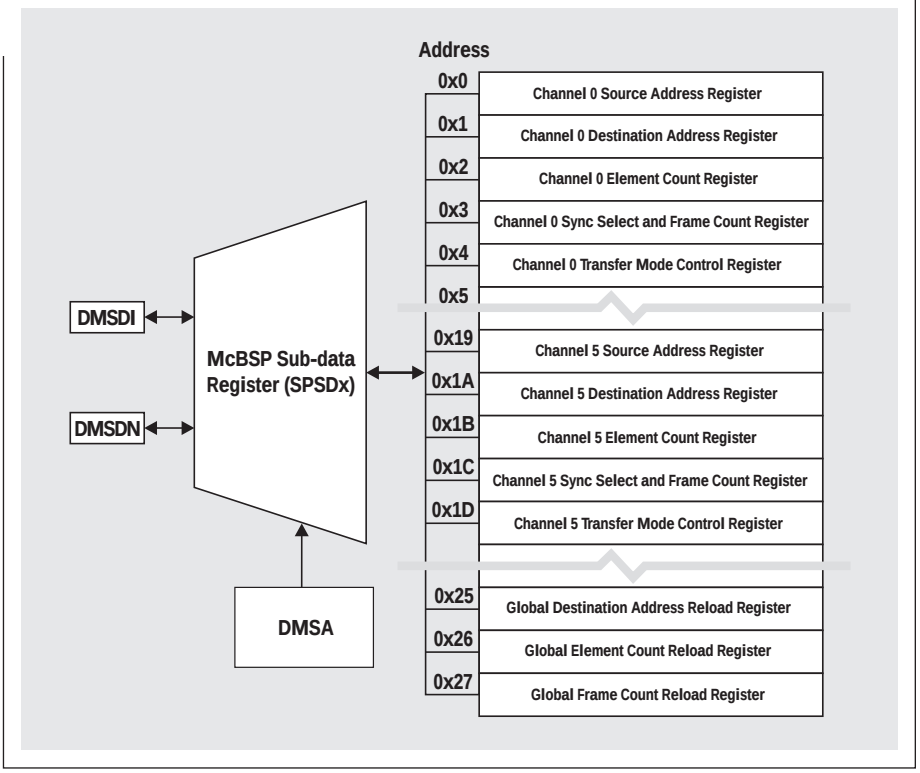
McBSP1 ADDRESS	McBSP1 SUB-ADDRESS	ACRONYM	REGISTER INITIALIZED	COMMENT
0041		DRR11		Receiver data register.
0043		DXR11		Transmit data register.
0048		SPSA1		McBSP1 sub-addressing register.
0049	0x0000	SPCR11	0x0001	While configuring McBSP transmitter-related registers, the LSB bit must be 0. This disables the transmitter.
	0x0001	SPCR21	0x02C1	While configuring McBSP receiver-related registers, the LSB bit must be 0. This disables the receiver.
	0x0002	RCR11	0x0040	Selects one 16-bit word transfer per frame.
	0x0003	RCR21	0x0001	Set 1-bit delay on receiver. Receiver assumes first MSB bit to arrive during clock cycle following FSR pulse.
	0x0004	XCR11	0x0040	Selects one 16-bit word transfer per frame.
	0x0005	XCR21	0x0001	Transmitter shifts out data immediately following falling edge of FSX.
	0x0006	SRGR11	0x0009	Assuming 100-MHz CPU clock, the CLKX has a frequency of 10 MHz. CLKX = CPU clock/(CLKGDV + 1), where CLKGDV = SRGR1(7,0).
	0x0007	SRGR21	0x2000	The sample-rate generator clock is derived from the CPU clock.
	0x000E	PCR1	0x0A00	FSX is determined from sample-rate generator frame-synchronization mode bit SRGR2.FSGM. CLKX output is driven by the sample-rate generator.

Continued from previous page

Direct memory access (DMA)

The DMA controller uses the same sub-addressing scheme as the McBSP, so the same procedure used in writing and reading from McBSP registers must be used here. The DMA provides one additional register, which makes programming the registers easier. Figure 2 shows two registers at the input to the multiplexer. DMSDI is a sub-bank access register (DMSA) that increments the sub-bank address register after each read/write, whereas the DMSDN register does not increment the sub-bank address register after read/write operations. The obvious advantage to using this DMSDI is that the user no longer has to change the contents of DMSA to point to the next register. For example, programming DMA channel 0 registers involves writing 0x00 to DMSA. The first value written to DMSDI is moved to DMSRC0, the second to DMDST0, the third to DMCTR0, etc. The DMSDN register can be used to write to the DMA sub-register if that particular register is the only one to be modified. Five channel-specific registers and the Channel Priority and Enable Control Register (DMPREC) need to be configured for each DMA channel used. They are outlined as follows:

Figure 2. DMA sub-address scheme



1. *Source (DMSRCn) and Destination (DMDSTn) Registers* (where n is the DMA channel). The DMSRC register stores the address of the data to be read out or moved from. Likewise, the DMDST register contains the address to where the data will be written in memory.
2. The *Element Count Register (DMCTRn)* is a 16-bit counter that keeps track of the number of DMA transfers to be completed. This register is always initialized to one less than the number of elements to be stored.
3. The *DMA Sync Event and Frame Count (DMSFCn) Register* controls three services: 1) the synchronization event used to trigger a DMA transfer; 2) the word size for each transfer, specified as either 16-bit or 32-bit words; and 3) the number of frames to be transferred. A frame can be specified from as low as 1 up to 256. For example, if only one frame is desired, this register field should be written as zero (the desired value minus one).
4. The *Transfer Mode Control Register (DMMCRn)* controls the channel transfer mode. This register determines whether the source/destination address will be post-increment or post-decrement after each transfer, whether the channel is operating in auto-buffering mode (ABU) or multi-frame mode, when the DMA will interrupt the DSP, and what address space the source/destination addresses.
5. The *Channel Priority and Enable Control Register (DMPREC)* controls the high-level function of the DMA channels. Due to the limited number of interrupts available in the C54xx family, some DMA interrupts are multiplexed with other peripheral interrupts; the register bits in this register determine which interrupts are assigned to the Interrupt Flag Register. The DMPREC also sets the priority given to each channel, either low or high. Channels with high priority are serviced before those with low priority.

The DMA register values used in this article are presented in Table 2.

Now that we have a basic understanding of the McBSP and DMA peripherals, let's look at the TLV2548 analog-to-digital converter.

TLV2548

This ADC features 8 analog input channels, an 8-level FIFO, and various conversion modes. The TLV2548 12-bit ADC is interfaced easily to the C5402 DSK. The hardware connections to the McBSP1 are shown in Figure 3. The DSP/McBSP drives serial lines CLKX and FSX, and accepts as input CLKR and FSR. The serial lines CLKX and CLKR, as well as FSX and FSR, are tied together on the EVM provided by Texas Instruments. The TLV2548 interrupt line is tied to INT3# of the DSP. Note that, since the FSX and FSR signals are identical, whenever a transmit operation occurs, a receiver transfer operation also occurs.

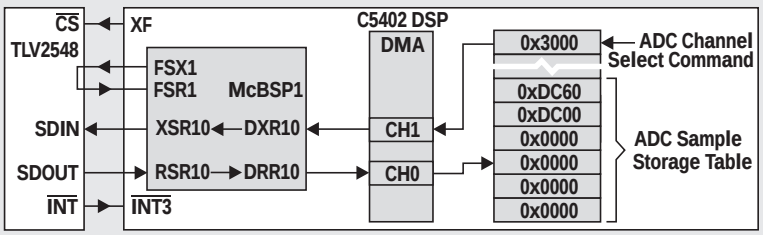
The TLV2548 offers four conversion modes, but only one is of interest—the single-shot conversion mode. In

Table 2. DMA register settings

REGISTER NAME	REGISTER VALUE	COMMENT
DMSRC0	&DRR11	*DMA Channel 0: Source Address Register* Source memory-mapped address to read from
DMDST0	&DataTable	*DMA Channel 0: Destination Address Rregister* Destination address to store sample to
DMCTR0	NSAMPLES-1	*DMA Channel 0: Element Count Register* Number of samples to store minus one
DMSFC0	0x5000	*DMA Channel 0: Sync Select and Frame Count Register* Synchronization transfers with McBSP1 receive event. When McBSP REVT occurs, CH0 reads data out of DRR1; then CH1 moves channel command word to DXR1.
DMMCR0	0xC004	*DMA Channel 0: Transfer Mode Control Register* Channel-context registers are re-initialized upon completion of block transfer. DMAC0 interrupt is generated after block transfer. No modification is made to Source Address Register after each transfer. Destination Address Register is post-incremented after transfer.
DMSRC1	&ADC_Cmd	*DMA Channel 1: Source Address Register* Address in memory where conversion command word is located.
DMDST1	&DXR11	*DMA Channel 1: Destination Address Register* Destination address to store conversion command—i.e., memory-mapped address of McBSP1 transmit data register.
DMCTR1	NSAMPLES -1	*DMA Channel 1: Element Count Register* Number of times to transfer command word to transmitter minus one.
DMSFC1	0xE000	*DMA Channel 1: Sync Select and Frame Count Register* Synchronization transfers with INT3# receive event. When INT3# occurs, CH0 reads data out of DRR1; then CH1 moves channel command word to DXR1.
DMMCR1	0x8000	*DMA Channel 1: Transfer Mode Control Register* The contents of the registers are re-initialized upon completion of block transfer. No modification is made to Source Address Register after each transfer. No modification is made to Destination Address Register after each transfer.
DMAPREC	0x0103	*DMA Priority and Enable Control Register* Enable DMA CH0 and CH1. CH0 has high priority. CH0 reads data out of the DRR1 and stores it before CH1 triggers another conversion cycle.

this mode, the digital code from the previous conversion must be read out before triggering another conversion cycle. This mode requires the CPU to service it often, preventing the CPU from performing other tasks. For most applications, it is undesirable to tie up an ultra-fast DSP by having it baby-sit a slow off-chip device. The solution is to use an off-chip peripheral, the DMA controller. This controller transfers data between regions in memory without intervention of the CPU. One DMA channel can be programmed to take a command word (for ADC) from memory and place it in the McBSP transmit data register, while another independent DMA channel reads the data out of the McBSP receive data register and stores it at a specified memory location (data table). Figure 3 shows the block diagram of this method. The source register of DMA Channel 1 points to the address of the next conversion command word. This command word tells an ADC which mode to enter, or what channel to convert from next. The destination register contains the address of the McBSP transmit data register. Likewise, the DMA Channel 0 source register contains the address of the McBSP receive data register, and its destination register contains the memory address to store the ADC digital code. Each DMA channel's transfer mode control register contains the bits that determine whether the source/destination addresses decrement, increment, or remain unchanged. If a number of samples is desired from only one channel, then the DMA channel transfer mode register is set so that no modification is made to the source and destination registers. The receiver DMA Channel 0 Transfer Mode Control Register is configured so the destination register is incremented after every transfer. This ensures that every read sample from the ADC is stored in a separate location in memory.

Figure 3. C5402 DMA interfacing to TLV2548

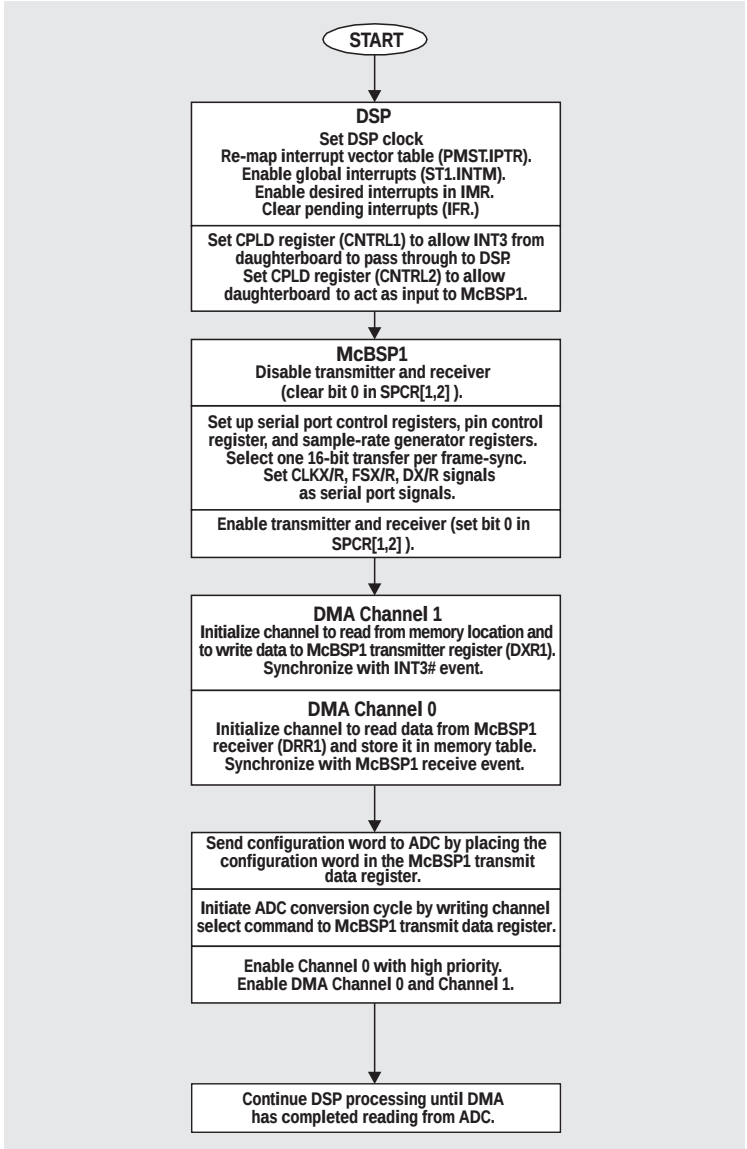


Continued on next page

Continued from previous page

The software flowchart for this interface method is presented in Figure 4. The CPU begins by initializing the respective DSP, McBSP, and DMA registers. Afterward, the only two remaining tasks for the CPU to perform are to send the configuration word and conversion channel command word to the ADC. Once the command word is placed in the data transmit register, it is automatically transferred through the transmit shift transfer onto the DX serial line. The McBSP sends out an FSX pulse immediately before the MSB bit is shifted out on the DX line. During this time, the ADC begins to shift out data on the SDO line. These first two receiver events must be read out of the receiver and discarded. The McBSP will store the digital code until it is read out, so it is important that the McBSP buffer registers be cleared before the DMA is allowed to collect samples. To understand the flowchart and how the hardware behaves, please refer to Figures 5 and 6. The first two cycles in Figure 6 show the configuration word followed by the first channel select command word. The data that is sent out of the ADC during these initial conversion cycles should be trashed because it is from an unknown configuration. The DMA Channel 1 is programmed to synchronize each transfer with an INT3# event. When the ADC de-asserts INT3#, DMA Channel 1 will be prompted to trigger another conversion cycle. When a McBSP1 receive event occurs, DMA Channel 0 reads the receive data register. Figure 5 is a close-up of a single conversion cycle in this single-shot conversion mode. These events will continue until the DMA has completed the number of transfers specified in the element count register for the channel. For example, if 256 samples were desired, the element count register, both for DMA

Figure 4. Single-shot conversion flowchart



Begin by setting the CPU clock speed. On the C5402 DSK the maximum CPU clock is 100 MHz. If there are any pending interrupts, they are cleared by writing all ones to IFR. Enable DMAC0 interrupt in IMR. The interrupt service routine (ISR) associated with this interrupt can be used to initiate DSP processing of the samples. The interrupt vector table (IVT) needs to be re-mapped so that it will point to the user ISR. The vector table should be placed at the beginning of a data page. The C5402 DSK I/O lines are managed by a CPLD. The CPLD therefore must be initialized to allow the external interrupt 3 (INT3#) signal from the expansion bus to pass through to the DSP. Similarly, the input to the McBSP must be set to arrive from the expansion bus. Before the McBSP registers are configured, it is important that the transmitter and receiver portions be disabled. Once that is accomplished, the user needs to configure the device for the desired operation. Once register values are set, then the transmitter and receiver may be enabled. For this interface the McBSP needs to be configured as in Table 1.

Once all five registers of each DMA channel are configured, the channels are enabled. DMA Channel 0, the channel that stores the received data, is given the highest priority and is synchronized with the McBSP1 receive event. DMA Channel 1, the channel that writes the command word, is synchronized with the INT3# event. The DMA should be programmed as described in Table 2.

The remaining steps are to configure the ADC and trigger the first conversion. This must be done manually by the CPU. The CPU puts the configuration word (for single-channel conversion) in the data transmit register, then places the channel select command on DXR1. Finally, DMA Channel 0 and Channel 1 are enabled. The resulting conversion is communicated to the DSP/DMA with INT3# going low. When INT3# is low, it's time for the DMA to send the next channel select command and read out the received data. The result of this conversion cycle is presented in the next cycle read. The CPU can be off doing its tasks until the DMAC0 interrupt occurs, at which point all the desired samples have been collected, and processing on those samples can begin.

receive and transmit channels, would be initialized to 255. Once the DMA channels have transferred 256 times, the conversion stops. The DMA can be programmed to interrupt the CPU once it is done. Again, the advantage of using the DMA to control the interface to this slow ADC is that the CPU is free to do other tasks.

This interface method and these register settings can be adapted for use with many of TI's current serial data converters that provide an interrupt signal and/or resolutions of ≤16 bits.

References

For more information related to this article, you can download an Acrobat Reader file at www-s.ti.com/sc/techlit/*litnumber* and replace "*litnumber*" with the **TI Lit. #** for the materials listed below.

Document Title **TI Lit. #**

1. Interfacing the TLV2544/TLV2548 ADC to the TMS320C5402 DSP slaa093
2. TLV2548/TLV2544 Datasheet slas198
3. TLVX544/2548EVM Evaluation Module for the TLVX544/TLV2548 10-bit and 12-bit ADC slau029
4. TMS320C54X Assembly Language Tools User's Guide spru102
5. TMS320C54X DSP CPU and Peripherals Reference Set, Vol. 1 . . . . . .spru131
6. TMS320C54XX DSP Enhanced Peripherals Reference Set, Vol. 5 . . . . . .spru302
7. TMS320C54X DSP Reference Set, Vol. 3: Algebraic Instruction spru179
8. TMS320C54X Optimizing C Compiler UG . .spru103
9. Using TMS320C5402 DMA Channels to Read from the TLV2548 . . . . . .slaa095

Related Web sites

www.dataconverter.com
www.ti.com/sc/select
www.ti.com/sc/docs/products/analog/tlv2548.html

Figure 5. Single-shot mode: One conversion cycle

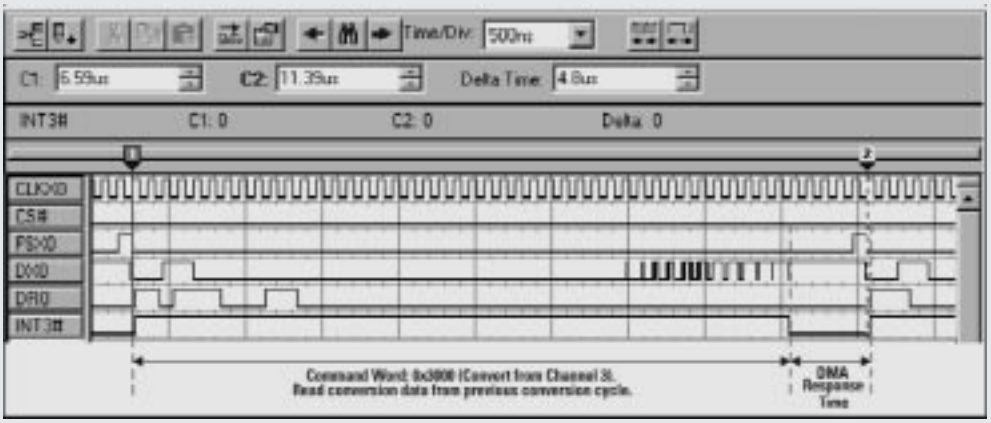


Figure 6. Single-shot conversion: Multiple cycles