Building a simple data acquisition system using the TMS320C31 DSP

By Joselito Parguian
Application Specialist

Introduction

When design engineers are faced with challenging design issues with emerging technology, they almost always think the design solution has to be complex. This is not always the case, especially with Texas Instruments products. One of TI's objectives is to develop products that are technically competitive and easy for designers to implement. This article demonstrates this simplicity while describing the interfacing of the TMS320C31 DSP to the TLV2548 ADC and TLV5618A DAC. This hardware interface solution is efficient, inexpensive, and easily accomplished. It provides an effective and very simple system for data acquisition applications.

Serial analog-to-digital converter

The TLV2548 is a high-performance, 12-bit, low-power, CMOS analog-to-digital converter (ADC). Conversion is accomplished by successive approximation with a charge-redistribution digital-to-analog converter (DAC). This conversion process is initiated by an external trigger signal, and once conversion is complete, a 16-bit output code is generated.

Data transfer in normal operation consists of a digital signal processor (DSP) initiating a command for the ADC to start sampling and converting the signal. A specified time after sampling and conversion are completed, the ADC interrupts the DSP and notifies it that the data is ready to be read.

The DSP and ADC logic interface is described below:

- The DSP selects the ADC by asserting the $\overline{CS}$ pin of the ADC through the XF0 output of the DSP.
- The transmit clock output of the DSP, CLKX, provides a fixed data clock into the SCLK input of the ADC and into the receive clock, CLKR, of the DSP as well.
- The transmit frame-sync output, FSX, initializes every data transfer by sending a frame-sync pulse to the ADC FS input, as well as to the receive frame-sync input, FSR.
- The DSP initializes the ADC by transferring a 4-bit control word and 12 bits of data from the DX output into the SDI input of the ADC. A configuration cycle follows after initialization is complete, and then a select cycle.
- After the conversion process is complete, the ADC generates an interrupt for the DSP to read its conversion data. The falling edge of $\overline{CS}$ or FS, whichever comes first, cancels the interrupt and clears the conversion data.
- The ADC clocks the digital conversion result out at the SDO pin, and into the DR input of the DSP.

Serial digital-to-analog converter

The TLV5618A is a low-power, dual 12-bit voltage output DAC, fabricated in CMOS technology, that can operate from a wide range of single-supply voltages. The resistor string output voltage is buffered by a 2x gain rail-to-rail output buffer. The buffer features a Class-AB output stage to improve stability and reduce settling time. The programmable settling time of the DAC allows for speed optimization versus power dissipation. The DAC is programmed with a 16-bit serial string containing 4 control and 12 data bits as shown in Figure 1.

Figure 1. DAC data format

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
R1	SPD	PWR	RO	MSB					12 Data bits						LSB

The DSP and DAC logic interface is described below:

- The DSP selects the DAC by asserting the $\overline{CS}$ pin of the DAC through the XF0 output gated along with FSX/FSR signals of the DSP. This is necessary because the DSP serial port is shared with another serial I/O device. In addition, the DAC does not support frame-sync pulse for data transfers, and the DSP initializes every data transfer by sending a frame-sync pulse.
- The transmit clock output of the DSP, CLKX, provides a fixed data clock into the SCLK input of the DAC and into the receive clock, CLKR, of the DSP as well.
- The DSP initializes the DAC and writes the data by transferring a 4-bit control word and 12 bits of data from the DX output of the DSP into the DIN input of the DAC.

Before the operation of the system is described, the reader should understand the internal operation of the TMS320C31 DSP standard bi-directional serial port interface.

DSP standard serial interface

The TMS320C31 DSP supports a standard bi-directional serial port interface that provides direct communication with any serial I/O devices. Before any system processes can be performed, initialization and configuration are required for all devices to be working properly. The DSP serial port requires a little more time and understanding for initializing and configuring, compared to the data converter devices. Therefore, this article explains the serial interface in some detail.

Six control lines from the DSP are used to interface to the data converters, and they are as follows:

CLKX Transmit clock input or output

This signal clocks data from the transmit shift register (XSR) to the data transmit (DX) pin. The serial port can be configured for internal clock generation

Continued on next page

Continued from previous page

or to accept an external clock. If the port is configured to generate the data clock on-chip, CLKX becomes an output, providing the data clock for the serial interface. If the port is configured to accept an external clock, CLKX changes to an input, receiving the external clock signal.

FSX Transmit frame synchronization input or output
FSX indicates the start of a data transfer. The serial port can be configured for internal frame-sync generation or to accept an external frame-sync signal. If the port is configured to generate the frame-sync pulse on-chip, FSX becomes an output. If the port is configured to accept an external frame-sync pulse, this pin becomes an input.

DX Serial data transmit
DX transmits the actual data from the transmit shift register (XSR).

CLKR Receive clock input
CLKR always receives an external clock for clocking the data from the data receive (DR) pin into the receive shift register (RSR).

FSR Receive frame synchronization input
FSR always receives an external frame-sync pulse to initiate the reception of data at the beginning of a frame.

DR Serial data receive
DR receives the actual data, which is clocked into the receive shift register (RSR).

In addition to these six control lines, there are control, shift, and buffer registers that are all 32 bits wide. These 32-bit-wide registers support the standard serial port interface operation of the DSP. Each of these registers, except for the two shift registers (XSR and RSR), is located in its specific address mapped in memory. Table 1 shows the register names, their respective addresses, and the values used for configuration applicable to this article.

Table 1. Registers associated with the serial port

REGISTER NAME	ADDRESS (HEX)	VALUE (HEX)
Serial port global control	808040	0C140044
FSX/DX/CLKX port control	808042	00000111
FSR/DR/CLKR port control	808043	00000111
R/X timer-control	808044	000001CF
R/X timer-counter	808045	00000000
R/X timer-period	808046	00000000
Data-transmit (DXR)	808048	Variable
Data-receive (DRR)	80804C	Variable

The control registers contain the control bits, set by the CPU, to configure the operation of the standard serial port. The addressable buffer registers, DXR and DRR, are discussed further under “Serial port operation,” which follows. See References 1 and 3 for more information about the TMS320C31 DSP serial port.

Serial port operation

During the transmit operation, the CPU loads the data transmit register (DXR), which then loads the word into the transmit-shift register (XSR), and the bits are shifted out. The word does not get loaded into the XSR until it is empty. Once the DXR is loaded into the XSR, the XRDY status bit is set, specifying that the buffer (DXR) is available to receive the next word, thus providing a double buffering function. The rising edge of the XRDY signal sets the XINT0 bit in the interrupt flag (IF) register, provided that the transmit interrupt enable (XINT) bit, in the serial port global control register, and the CPU serial port 0 transmit interrupt enable (EXINT0) bit, in the interrupt enable (IE) register, are set.

During the receive operation, the CPU reads the received data from the data receive register (DRR), which is double-buffered as well. When the serial data (such as the ADC data) is input, the receive shift register (RSR) receives the data. When the specified number of bits is shifted in, the DRR is loaded from RSR, and the RRDY status bit is set, specifying that there is new data ready to be read from the DRR. If the DRR is not yet read and the RSR is full, the receiver is frozen. Any new data coming into the DR pin is ignored until the DRR is read. RSR does not write over the DRR; therefore, DRR must be read to allow new data in the RSR to be transferred to the DRR. The rising edge of the RRDY signal sets the RINT0 bit in the interrupt flag (IF) register, provided that the receive interrupt enable (RINT) bit, in the serial port global control register, and the CPU serial port 0 receive interrupt enable (ERINT0) bit, in the interrupt enable (IE) register, are set.

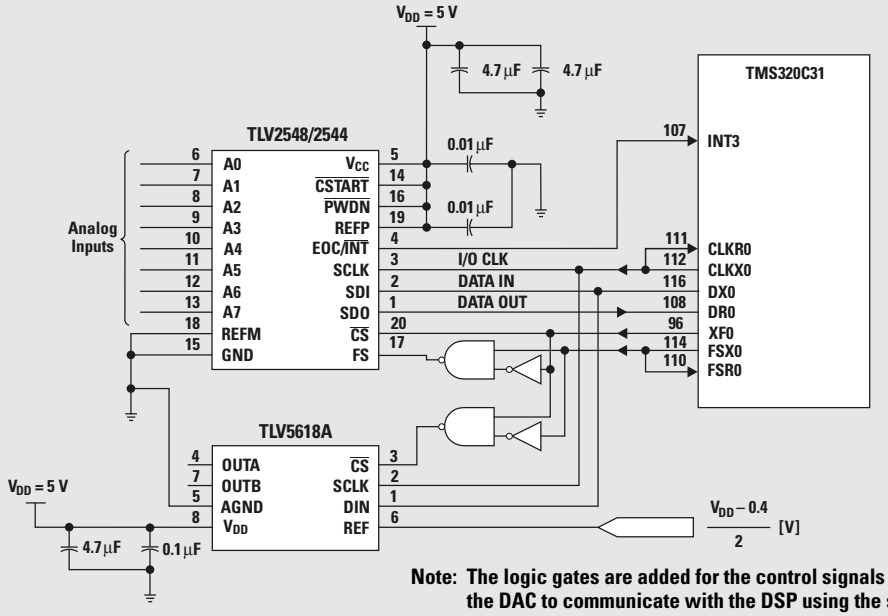
The interface operation

When only one data converter device is involved, interfacing to the DSP is a cinch. But if more than one serial I/O device is used to interface with the DSP serial port, some challenging design issues arise. The situation that needs to be addressed is that the DSP should assert only one data converter at a time, specifically with the ADC. This is because the ADC interrupt signal is canceled by the falling-edge transition of FS. If the ADC interrupt signal is canceled, the corresponding ADC conversion data is also cleared. On the other hand, the DAC write cycle starts when the device detects a falling-edge transition from its $\overline{CS}$ pin. The MSB is expected by the DAC after the falling edge of $\overline{CS}$; therefore, the DSP has to send the data instantaneously when a write to the DAC is performed. Since the DSP sends a frame-sync signal for every word transfer, the DSP has to be able to discern with which device to communicate. By gating, the required signals correctly achieve the complete isolation of the devices and ensure correct data transfer to each device.

The hardware solution in Figure 2 shows the use of multiple-gate logic to control the selection of data converters sharing the same serial port of the DSP.

The falling edge of the FS signal from the DSP is the start of the ADC cycle. The ADC cycle is normally started with an initialization cycle followed by a configuration cycle. Once the ADC is initialized and configured, the normal sample-and-convert process can be executed continuously. The ADC transmits and receives data with a 16-bit string, and the first 4 bits of data (MSB) received from the

Figure 2. Hardware configuration for DSP to ADC and DAC interface



Note: The logic gates are added for the control signals to allow the ADC and the DAC to communicate with the DSP using the same serial port.

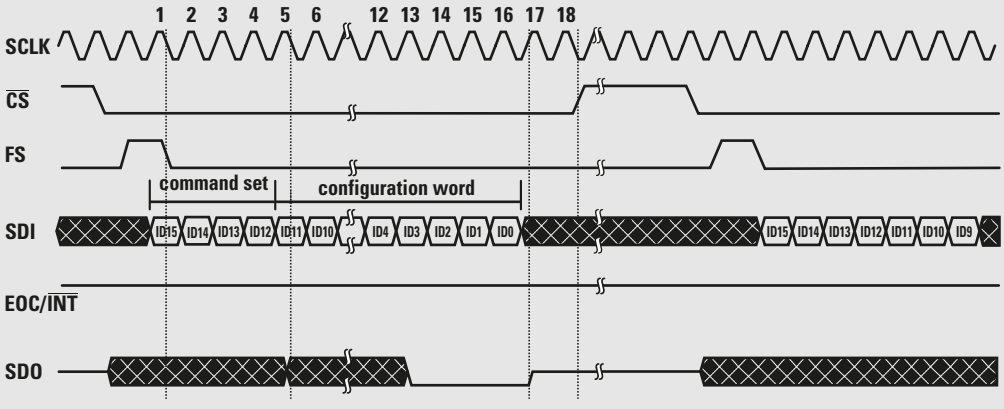
DSP are interpreted as a command set to determine the mode of the ADC. The remaining 12 bits of data make up the configuration code. If the command set initiated is A hex for bits D[15:12], then the following 12 bits of data, D[11:0], are used to configure the ADC. Figure 3 shows the timing diagram of the ADC write cycle initiated by the DSP to configure the ADC. During the initialization and configuration cycle, no conversion cycle is initiated on either read or write. The conversion cycle is initiated only after the ADC is configured as one of the four different modes of conversion (modes 00, 01, 10, 11). These four conversion modes operate slightly differently from each other, depending on how the converter performs sampling. Although the ADC has different options to select its trigger for conversion, this article discusses only the use of $\overline{CS}$ and FS to trigger the start of conversion for the mode discussed here, mode 00 (single-shot mode). See Reference 6 for more information.

Figure 3 shows the typical data transfer sequence for the ADC input data. The first bit (MSB) of data is expected after the falling edge of FS, and all following bits are shifted in on the falling edges of SCLK. For the output data, the first bit (MSB) of data is presented to the SDO pin after the falling edge of $\overline{CS}$ and FS LOW is detected. Successive output data changes on the rising edge of SCLK and is available at the falling edges of SCLK.

The data collected by the DSP from the ADC is presented to the DAC at some certain time by the DSP. A slight adjustment to the ADC data needs to be made before it is presented to the DAC. Since the data received from the ADC has only 12 bits of usable data, the DSP has to

Continued on next page

Figure 3. ADC configuration (write CFR)



Continued from previous page

perform a shift of 4 bit positions to the right, starting from the MSB. The 4-bit control word for the DAC is then padded to the most significant nibble to make a good 16-bit string of data for the DAC. The DSP data manipulation flow, to format the ADC data to suit the DAC data format, is shown in Figure 4.

The DAC 4-bit control word, as shown in the 4 MSBs in Figure 1, is described as follows:

- R0 and R1 select all possible combinations of register-select bits (see Table 2).
- SPD controls the speed mode of the DAC.
- PWR controls the DAC to be in power-down or normal operation.

  SPD = 1 fast mode SPD = 0 slow mode
  PWR = 1 power down PWR = 0 normal operation

The DAC timing requirement is shown in Figure 5. See Reference 7 for more detailed information.

Figure 6 shows a typical timing diagram of the ADC and DAC data transfer through and from the DSP, respectively. The LOW time of ADC_CS indicates that the ADC is active, and the HIGH time indicates its inactivity. On the other hand, the LOW time of DAC_CS (HIGH time of ADC_CS) indicates the DAC active state. The specific assertion between the DAC and the ADC is made possible by the logic gates added in the hardware to share the DSP serial port.

A simple illustration of the complete data acquisition system is shown in Figure 7. The functional operation of the illustration is performed using the single-shot mode (mode 00) of the ADC for the data acquisition. Therefore, the single-shot mode will be discussed further to explain its operation.

Table 2. Register-select bits

R1	R0	REGISTER
0	0	Write data to DAC B and buffer
0	1	Write data to buffer
1	0	Write data to DAC A and update DAC B with buffer content
1	1	Reserved

Figure 4. DSP data manipulation of ADC data for DAC data format

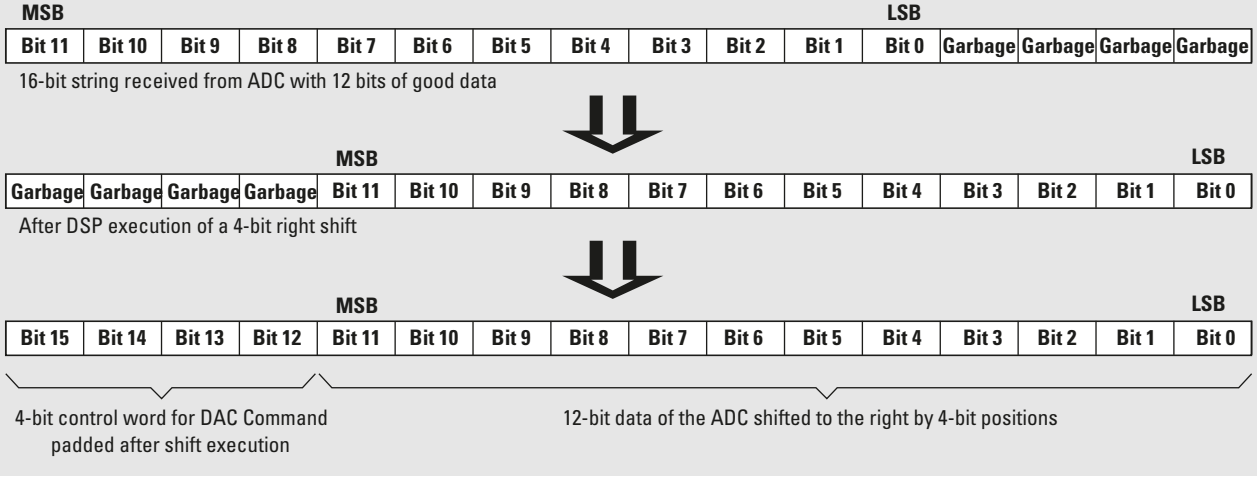


Figure 5. DAC timing diagram

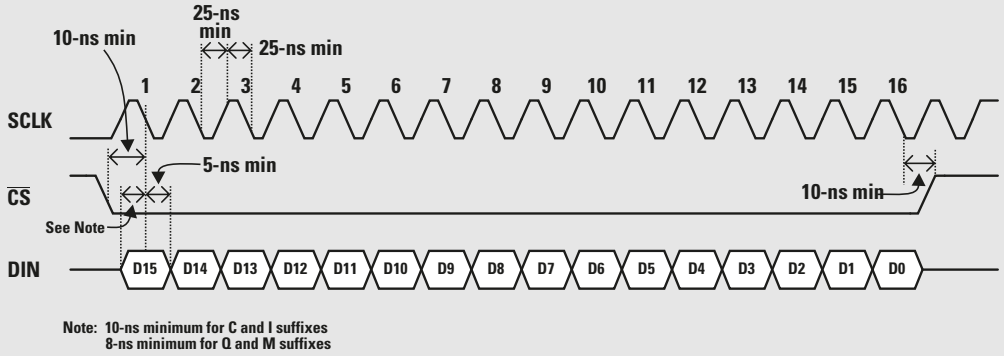
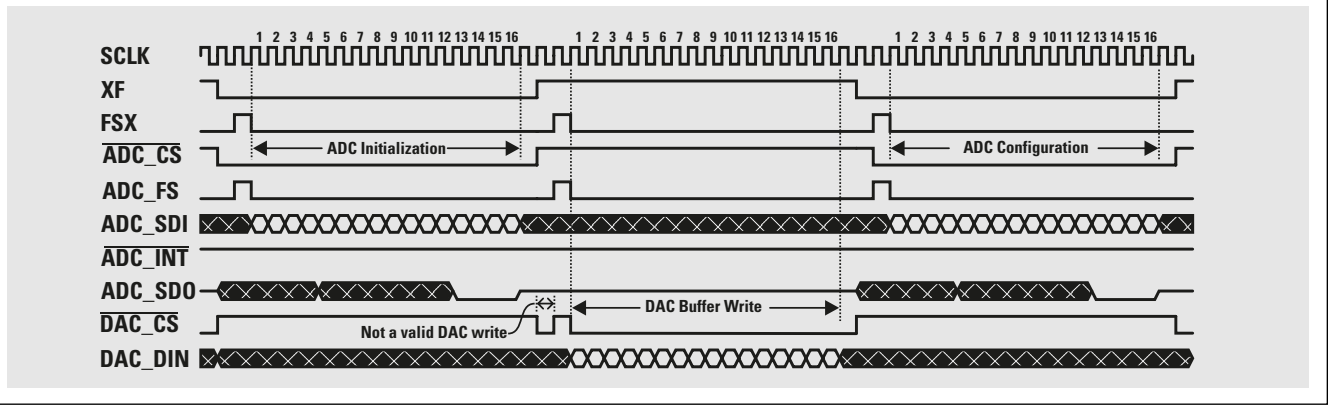


Figure 6. ADC and DAC timing diagram

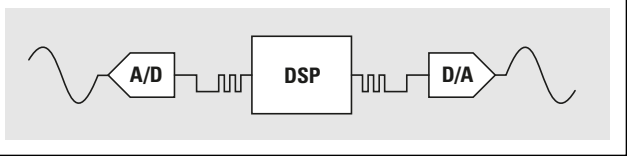


Mode 00 (single-shot mode)

The single-shot mode is the common method used for data conversion because it is easy to implement and a fairly simple process. The DSP initiates the conversion process by asserting the $\overline{CS}$ pin of the ADC through the XF0 pin and issuing the FS signal. This starts the ADC cycle. Figures 8-10 show a complete typical conversion cycle of the ADC in single-shot mode.

Figure 8 shows the initialization and configuration cycles. As discussed earlier, the initialization cycle is necessary for device performance stability. If the device is not initialized properly, it may not perform correctly. To initialize the device properly, write the configuration word A000 hex once upon power-up of the device. After initialization the device can then be configured. A000 hex was chosen for the configuration as well, since external reference, short sampling, internal oscillator for the conversion clock, single-shot mode, and interrupt pin function were selected. The DAC buffer is also initialized to zero to avoid any spurious data being written out from the DAC. This is specifically done when the first write to channel A of the DAC is initiated. The buffer content is written simultaneously to channel B of the DAC as well.

Figure 7. Illustration of the complete data acquisition system



In this article, channel selection (channel 1) is arbitrarily chosen for the ADC to sample the analog input. The channel is decoded in the first 4 SCLKs. Sampling is started at the beginning of the 5th SCLK and continues until the end of the 16th SCLK. The conversion cycle follows after the 16th SCLK, and an interrupt ($\overline{ADC\_INT}$) is generated by the ADC after the conversion is complete (or EOC is generated as the conversion is in progress), but it does not use the FIFO to store the data. The DSP must read the data out of the data converter before beginning another conversion cycle. Notice that as soon as the device is selected ($\downarrow\overline{CS}$ when FS is LOW), the previous conversion data is presented at the SDO pin and is held until the ADC sees a valid FS ($\downarrow$FS) signal. Succeeding data is available at the falling edge of SCLK. Also note that, for the first conversion cycle, the data presented at the SDO pin is garbage data. The DSP can either ignore this or perform a dummy read cycle, depending on the user's preference. If the garbage data is not read, it is simply overwritten with

Continued on next page

Figure 8. ADC initialization and configuration and DAC buffer write cycle for mode 00 (non-conversion cycle)

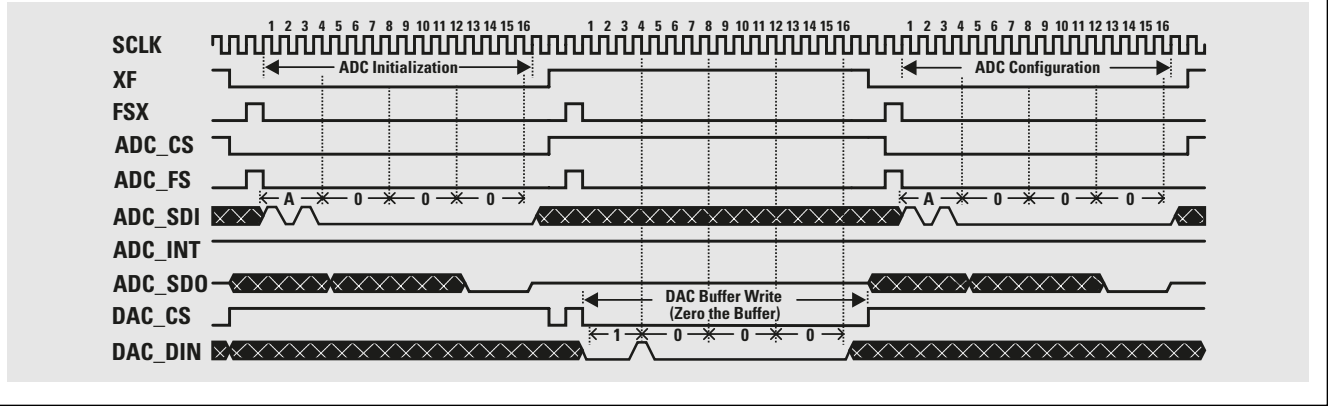


Figure 9. ADC channel select, first conversion cycle, and DAC write

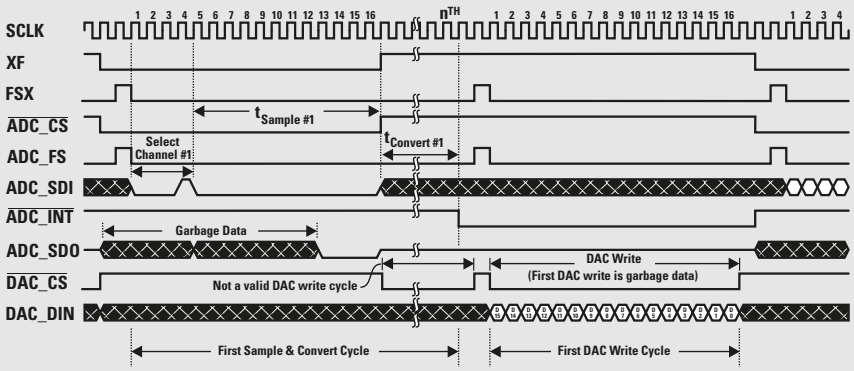
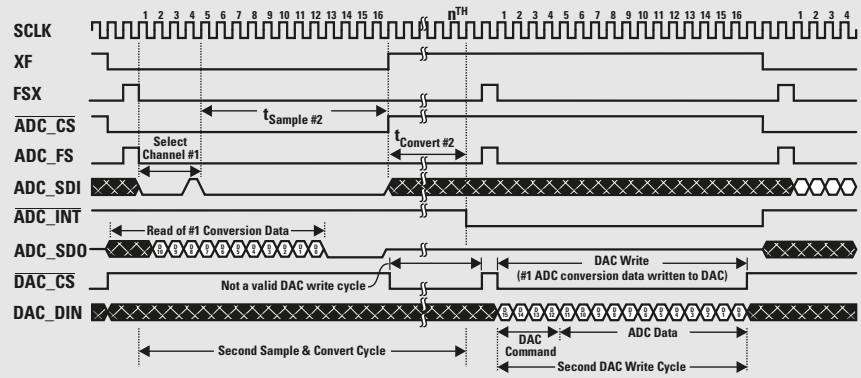


Figure 10. ADC channel select, second generation cycle, and DAC write



Continued from previous page

the result of the simultaneous conversion process. Be aware, though, that this is also true for good data if another conversion is initiated without reading the ADC.

As soon as the DSP reads the ADC data, it adjusts the data to fit the DAC data format as discussed earlier. When the DSP has completed writing the adjusted data to the DAC, the ADC cycle is again repeated. This cycle continues until the program is aborted. A routine for the single-shot mode can be downloaded from the TI Web site at www.ti.com/sc/docs/psheets/abstract/apps/slaa101.htm

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. Joselito Parguian, "Interfacing the TLV2544/TLV2548 ADC to the TMS320C31 DSP," Application Report slaa101
2. Thomas Kugelstadt, "A Methodology of Interfacing Serial A-to-D Converters to DSPs," *Analog Applications Journal* (February 2000), pp. 1-9 . . . . . .slyt175
3. "TMS320C3X DSP User's Guide" spru031
4. "TMS320C3X DSP Starter Kit User's Guide" . . .spru163
5. "TMS320C3X/C4X Assembly Language Tools User's Guide" spru035
6. "TLV2544, TLV2548 2.7-V to 5.5-V, 12-Bit, 200-kSPS, 4-/8-Channel, Low-Power Serial Analog-to-Digital Converters with Autopower-down," Data Sheet slas198
7. "TLV5618A 2.7-V to 5.5-V Low-Power Dual 12-Bit Digital-to-Analog Converter with Power Down," Data Sheet slas230
8. "TLV1504/1508/1544/2544/2548, TLC1514/1518/2554/2558 10-Bit and 12-Bit ADC EVM User's Guide" slau029
9. "Code Composer Studio User's Guide" spru328

Related Web sites

www.dataconverter.com

Get product data sheets at:
www.ti.com/sc/docs/products/analog/*device***.html**
Replace *device* with tlv2544, tlv2548, or tlv5618a

www.ti.com/sc/docs/products/dsp/tms320c31.html

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	
Amplifiers	**amplifier.ti.com**
Data Converters	**dataconverter.ti.com**
DSP	**dsp.ti.com**
Interface	**interface.ti.com**
Logic	**logic.ti.com**
Power Mgmt	**power.ti.com**
Microcontrollers	**microcontroller.ti.com**
Applications	
Audio	**www.ti.com/audio**
Automotive	**www.ti.com/automotive**
Broadband	**www.ti.com/broadband**
Digital control	**www.ti.com/digitalcontrol**
Military	**www.ti.com/military**
Optical Networking	**www.ti.com/opticalnetwork**
Security	**www.ti.com/security**
Telephony	**www.ti.com/telephony**
Video & Imaging	**www.ti.com/video**
Wireless	**www.ti.com/wireless**

TI Worldwide Technical Support

Internet

TI Semiconductor Product Information Center Home Page
support.ti.com

TI Semiconductor KnowledgeBase Home Page
support.ti.com/sc/knowledgebase

Product Information Centers

Americas

Phone	+1(972) 644-5580	Fax	+1(972) 927-6377
Internet/Email	support.ti.com/sc/pic/americas.htm		

Europe, Middle East, and Africa

Phone			
Belgium (English)	+32 (0) 27 45 54 32	Netherlands (English)	+31 (0) 546 87 95 45
Finland (English)	+358 (0) 9 25173948	Russia	+7 (0) 95 7850415
France	+33 (0) 1 30 70 11 64	Spain	+34 902 35 40 28
Germany	+49 (0) 8161 80 33 11	Sweden (English)	+46 (0) 8587 555 22
Israel (English)	1800 949 0107	United Kingdom	+44 (0) 1604 66 33 99
Italy	800 79 11 37		
Fax	+(49) (0) 8161 80 2045		
Internet	support.ti.com/sc/pic/euro.htm		

Japan

Fax			
International	+81-3-3344-5317	Domestic	0120-81-0036
Internet/Email			
International	support.ti.com/sc/pic/japan.htm		
Domestic	www.tij.co.jp/pic		

Asia

Phone			
International	+886-2-23786800		
Domestic	Toll-Free Number		Toll-Free Number
Australia	1-800-999-084	New Zealand	0800-446-934
China	800-820-8682	Philippines	1-800-765-7404
Hong Kong	800-96-5941	Singapore	800-886-1028
Indonesia	001-803-8861-1006	Taiwan	0800-006800
Korea	080-551-2804	Thailand	001-800-886-0010
Malaysia	1-800-80-3973		
Fax	886-2-2378-6808	Email	tiasia@ti.com
Internet	support.ti.com/sc/pic/asia.htm		ti-china@ti.com

C011905

Safe Harbor Statement: This publication may contain forward-looking statements that involve a number of risks and uncertainties. These "forward-looking statements" are intended to qualify for the safe harbor from liability established by the Private Securities Litigation Reform Act of 1995. These forward-looking statements generally can be identified by phrases such as TI or its management "believes," "expects," "anticipates," "foresees," "forecasts," "estimates" or other words or phrases of similar import. Similarly, such statements herein that describe the company's products, business strategy, outlook, objectives, plans, intentions or goals also are forward-looking statements. All such forward-looking statements are subject to certain risks and uncertainties that could cause actual results to differ materially from those in forward-looking statements. Please refer to TI's most recent Form 10-K for more information on the risks and uncertainties that could materially affect future results of operations. We disclaim any intention or obligation to update any forward-looking statements as a result of developments occurring after the date of this publication.

Trademarks: All trademarks are the property of their respective owners.

Mailing Address: Texas Instruments
Post Office Box 655303
Dallas, Texas 75265