Interfacing the TLC2543 ADC to the TMS320C25 DSP

SLAA017
February 1997

Texas Instruments

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software 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.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI 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 semiconductor products or services might be or are used.

Contents

List of Figures

Interfacing the TLC2543 ADC to the TMS320C25 DSP

ABSTRACT

This document describes the requirements for interfacing the TLC2543 ADC to a TMS320C25 DSP.

1 Introduction

This application report details the implementation of an interface between a TMS320C25[1] digital signal processor (DSP) and a TLC2543[2] 12-bit, 11-channel analog-to-digital converter (ADC). The results also can be interpreted more generally to be valid for interfacing a TMS320C5x DSP to the same device.

The interface to the TLC2543 is a standard serial peripheral interface (SPI) that is resident on many standard microcontrollers. This interface can be adapted for use in interfacing DSPs and microcontrollers across the SPI.

1.1 Description of the TLC2543

The TLC2543 is a 12-bit, 11-channel ADC that uses a switched-capacitor successive approximation technique to perform the conversion process (see Figure 1 for a block diagram of the TLC2543). The TLC2543 has a low supply current of 1 mA typically with a power-down mode of 4 μA typically when activated by software. The unadjusted error is +/– 1 LSB maximum.

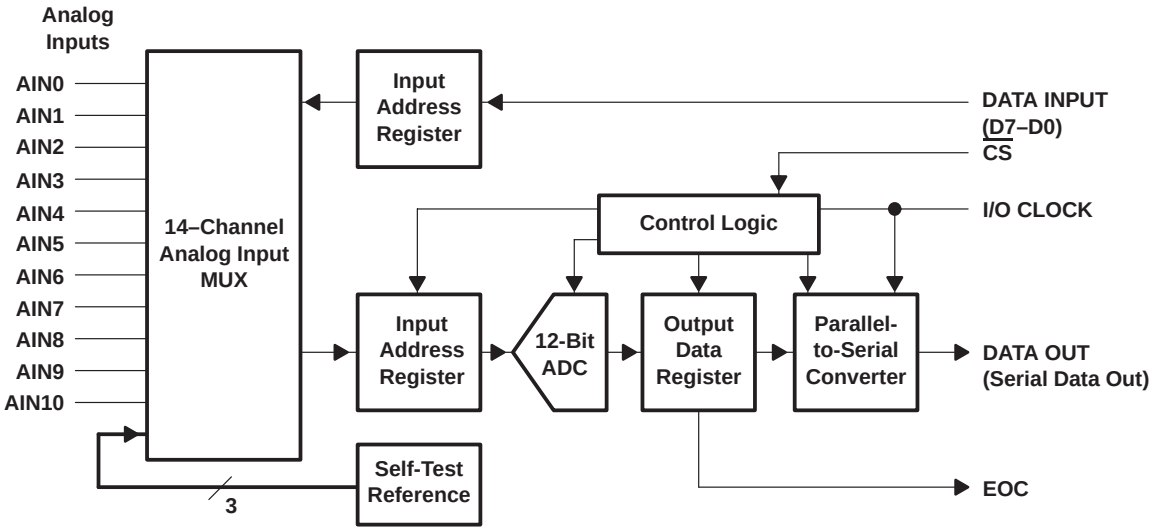


Figure 1. TLC2543 Block Diagram

The digital interface is based on a standard bidirectional SPI port. Both data and control information are sent referenced to an externally-sourced bit clock. Concurrent with data transfers coming from the device, control information can be sent to the device. This control information selects the analog channel number, test modes, data output format, and power-down mode.

Data output formats can be selected as either 8, 12, or 16 bits in either binary or twos-complement notation. Data can be output as LSB or MSB first.

1.2 Hardware Interfacing

For the hardware interface circuit, a standard 40-MHz TMS320C25 (100 ns cycle time) with a TLC2543IN (industrial temperature) were used (see Figure 2).

The TMS320C25 contains one synchronous serial port that functions from an external clock source (bit clock) and can operate in either burst mode (requiring frame syncs) or continuous mode (only one frame sync required for entire transmission). This serial port is suitable for interfacing to standard combo codecs and analog interface circuits (AICs) without glue logic. To adapt its operation to that of an SPI requires the ability to control the clock asynchronously and to operate entirely without frame syncs.

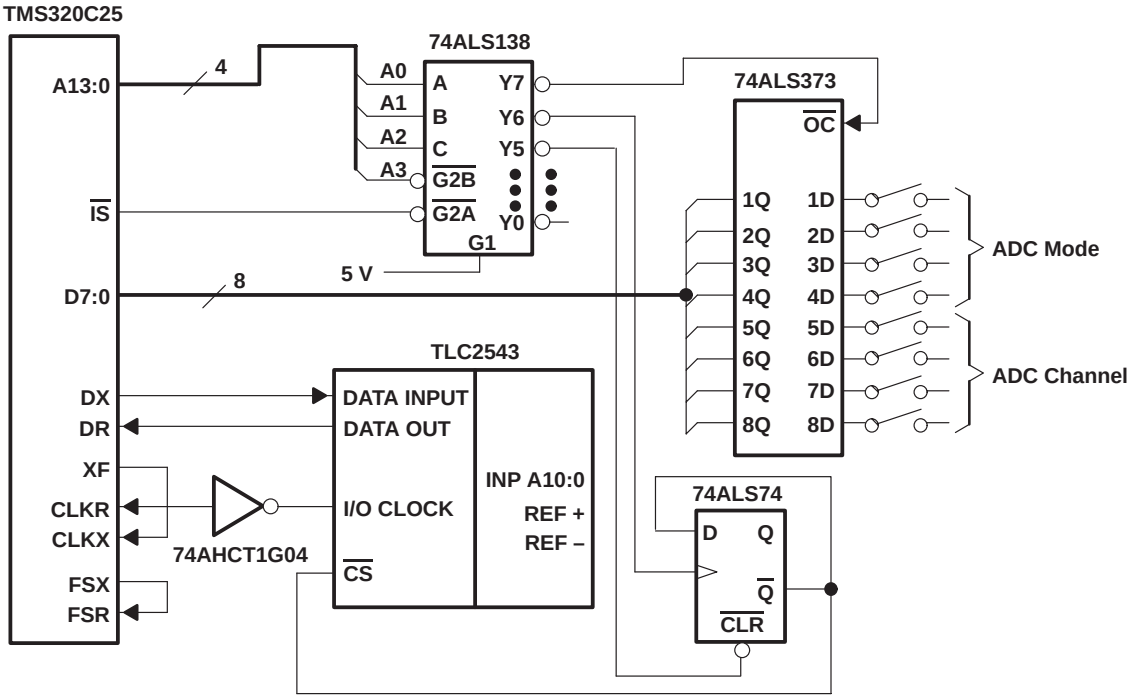


Figure 2. TLC2543/TMS320C25 Interface Circuit

For a minimum glue logic configuration, XF (external flag) on the TMS320C25 is used as the clock source and toggled accordingly by software. XF provides the clock for both the transmit (TX) and receive (RX) operation of the TMS320C25 serial port as well as the I/O clock for the TLC2543. While the TMS320C25 serial port data is referenced from the positive edge of the bit clock, the TLC2543 I/O clock is referenced from the negative edge, therefore, an inverter such as TI part 74AHCT1G04 is used.

The TLC2543 can operate in two modes of data acquisition, using $\overline{CS}$ (chip select) on each transfer or always keeping $\overline{CS}$ low with the sample rate based on the end-of-conversion (EOC) signal going high (active). This interface uses $\overline{CS}$ for each transfer to ensure that the DSP has accurate control over the sampling rate. $\overline{CS}$ is brought low (active) by a DSP access to its I/O port address 0x6 that in turn toggles the register connected to the $\overline{CS}$ input. On reset, the $\overline{CS}$ signal is high. Another port address can also ensure that the D flip-flop is cleared ($\overline{CLR}$ input).

The information for configuring the TLC2543 control byte is accessed from the external latch read from the DSP port address 0x7. The configuration information is supplied by a host or by dip-switch settings.

The interface timing diagram is shown in Figure 3. A delay of 1.425 μs is required from $\overline{CS}$ going low to the first I/O CLOCK going high (active). This delay ensures that the TLC2543 internal circuits have initialized correctly. Control data is sent to the TLC2543 on the first eight clock cycles of the I/O CLOCK; the remaining eight clock cycles of the transfer are ignored. The 12-bit output from the TLC2543 conversion are left-justified in the 16-bit word received by the DSP. In access cycle B, the channel address for the analog sample is determined. In sample cycle B, the appropriate analog channel is sampled. Conversion takes place starting on the last transition (negative edge) of I/O CLOCK. When the $\overline{CS}$ signal is brought low, the data-out bus from the TLC2543 goes to a high-impedance state allowing other devices to share the input channel of the DSP serial port.

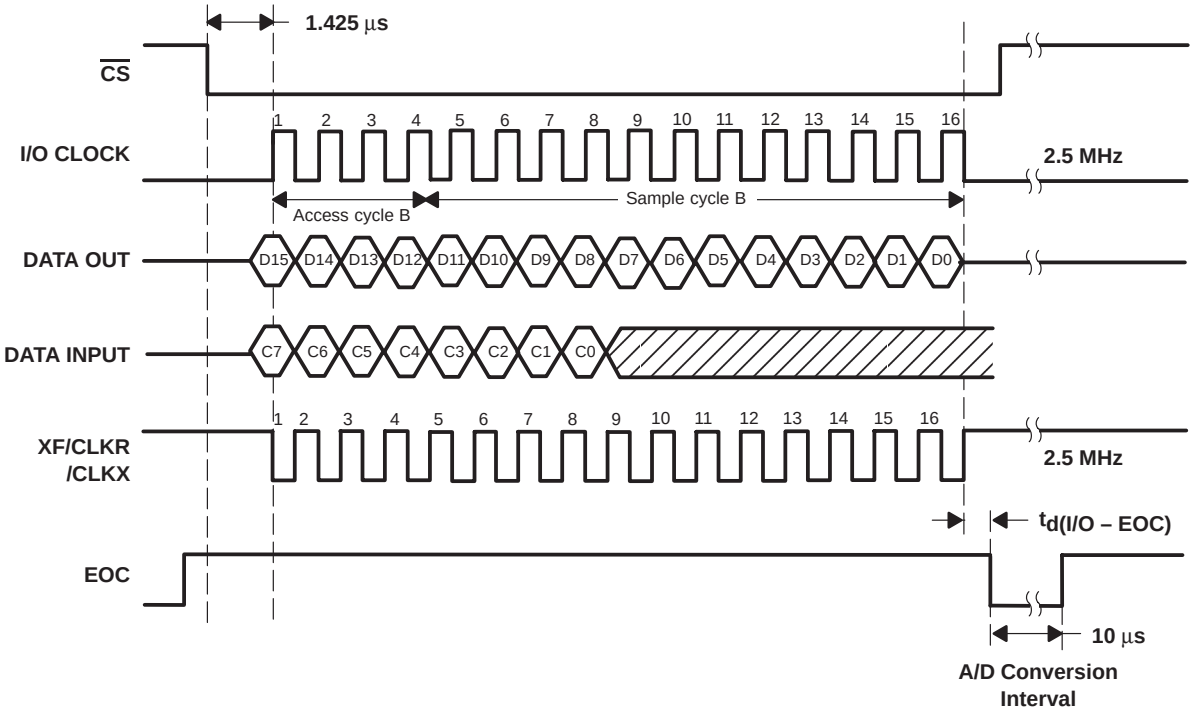


Figure 3. Interface Timing Diagram

1.3 Software Description

The TMS320C25 controlling software consists of two modules: the T2543.ASM and T2543VEC.ASM (see Figure 4), and the associated linker command file describing the TMS320C25 memory map.

The T2543.ASM module is the initialization program that configures the processor, the serial port, and the interrupts, and performs the first transfer to program the TLC2543 for the first time. The first sample captured from the device is considered invalid. The TMS320C25 serial port is configured for continuous mode; therefore, one frame sync is required to start the transaction and no syncs are required thereafter. The FSX pulse is configured as an output and is connected to FSR to provide the initial frame sync for both transmit and receive of the serial port. An additional clock is required to clock the frame sync, and that is performed before $\overline{CS}$ is taken low on the TLC2543.

It is necessary to maintain the correct amount of I/O clocks for the ADC; otherwise, sample timing becomes corrupted. The XF flag is toggled 16 times (using the NOP instruction to make 2.5 MHz) and simulates the first transfer of 16 bits of sample data and 8 bits of control data. The TMS320C25 timer is then set up and interrupts are enabled so that data transfers take place with a sample period based on the timer-period register.

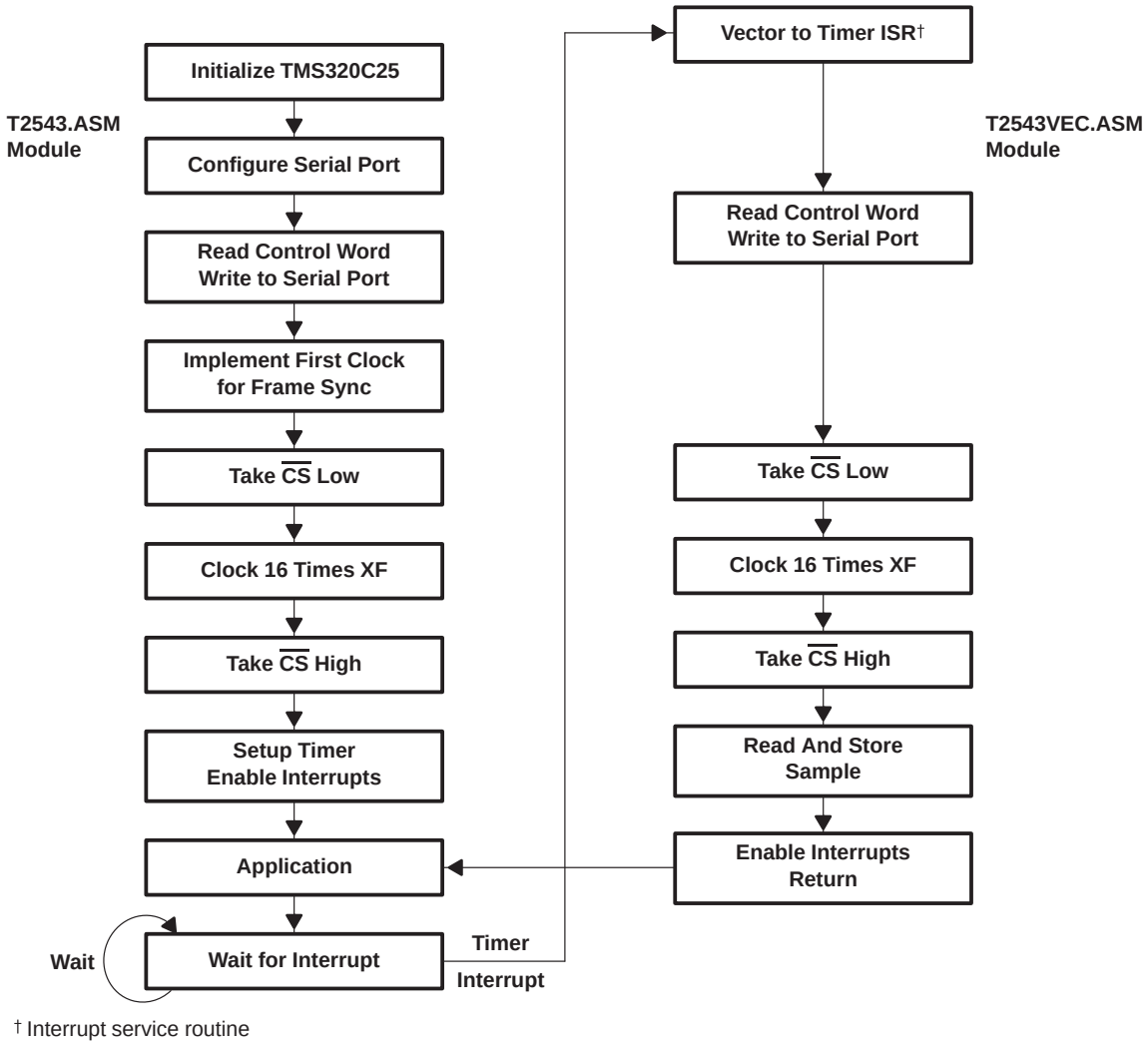


Figure 4. Software Flow Chart

The T2534VEC.ASM module contains the TMS320C25 interrupt vector table and the timer interrupt service routine. For each timer interrupt, a new control word is read from the external latch, $\overline{CS}$ is brought low, 16 clock cycles are issued, $\overline{CS}$ is brought high again, and the sample is read and stored. A delay of 1.425 μs is implemented with a loop using the RPT instruction for 15 counts, which should give at least a 1.6-μs delay.

2 Benchmark Information

2.1 Sample Rates

The $\overline{CS}$-activated mode of transfer was selected for a more deterministic sampling approach where the DSP has control over when a sample is read and inherently when the next sample is converted. Another approach is to use an inverted version of the EOC signal to trigger an external DSP interrupt; then the sampling is controlled more by the ADC.

The maximum sample rate of the TLC2543 for different sample lengths is shown in Table 1.

Table 1. Sample Rates of TLC2543 (When Using 2.5 MHz I/O Clock)

SAMPLE LENGTH (bits)	ACCESS TIME, T_{access} (μs)	SAMPLE RATE (k samples/s)
8-bit interface	14.9	67.3
12-bit interface	16.5	60.7
16-bit interface	18.1	55.4

NOTES: 1. For 8-bit mode, two transfers are needed for full 12-bit samples.
2. The 16-bit mode is used in this application report.
3. A 1.425-μs delay is used only for $\overline{CS}$-initiated transfers.

Access to conversion time for sampled data can be calculated using equation 1.

Access time:

$$T_{access} = 1.425\ \mu s + sample\ length \times T(I/O\ CLOCK) + t_{d(I/O\text{-}EOC)} + t_{con}$$

Where:

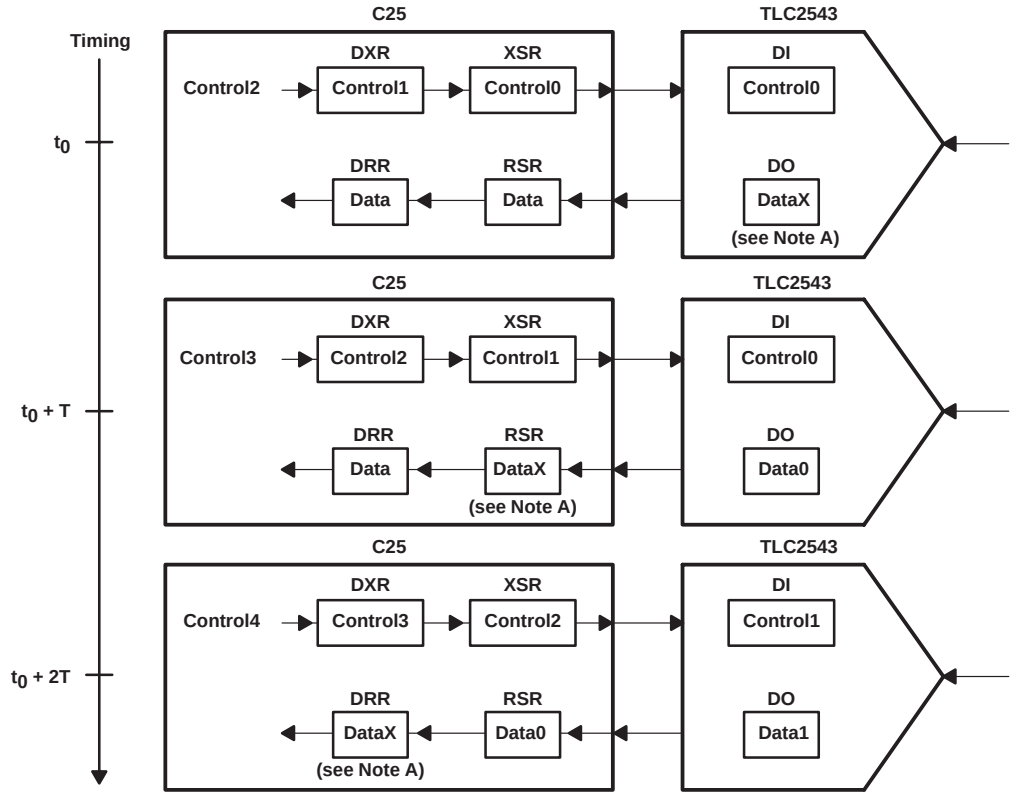
1.425 μs = time for TLC2543 initialization after $\overline{CS}$ low
T(I/O CLOCK) = 1/2.5 MHz = 400 ns
$t_{d(I/O\text{-}EOC)}$ = 240 ns (maximum)
t_{conv} = 10 μs (maximum)

2.2 Data and Control Information Timing

The following delays must be considered for both control-word programming and the reception of samples. These delays are due to:

1. The TMS320C25/C50 serial ports are double buffered; therefore, there is a delay between when the control byte is written to the serial port and when the byte is transmitted to the TLC2543.
2. There is one sample delay between the selection of an analog channel and when its associated data is transferred to the DSP. (The digital output from the converter is available in the next cycle.)
3. The last delay is implicit in the manual clocking of the XF terminal by the TMS320C25 in order to receive the sample. For this application, it is 16 clock cycles per transfer with the sample being transferred from the DSP receive shift register (RSR) to the DSP data receive register (DRR) on the last sample clock.

Figure 5 shows delay timings that can be expected from the input of the control 1 byte to the storage of the associated data1 word in a DSP memory location.



NOTE A: DataX means undetermined data.

Figure 5. Control and Data Timing

Equation 2 shows the components that comprise the total time delay.

Time delay = 2T + clocking time + data storage

Where:

T = sampling period (derived from TMS320C25 timer with a minimum of 18.1 μs for a 16-bit access)

Clocking time + data storage = 9.8 μs (derived from TMS320C25 code)

2.3 Circuit Verification

Using the calibration modes of the TLC2543, the circuit can be verified functionally. Table 2 shows the results that should be obtained when using this application circuit and software.

Table 2. Verification Table Of Results Using TLC2543 Test Modes

TEST MODE (D7–D4)	BINARY (D0 = 0)	TWOS COMPLEMENT (D0 = 1)	DESCRIPTION
1011b	7ffh	000h	$(V_{ref+}–V_{ref–})/2$
1100b	000h	800h	$V_{ref–}$
1101b	fffh	7ffh	V_{ref+}
1110b	000h	800h	Software power down

3 Application Information

DSPs have distinct advantages over traditional microcontrollers when considering feed forward and compensator control systems[3]. The main advantages are:

- Computing speed: DSPs have the ability to implement complex control algorithms while maintaining high sampling rates. A DSP can sample greater than four-fifths times (80%) faster than the average microcontroller. The hardware multiplier/accumulate function is a distinct advantage in the evaluation of modern control theory.
- Cost: DSPs have the ability to handle more channels with one device, or compute on-the-fly look-up tables rather than needing external ROM often gives DSPs a cost and performance advantage. Integration of ROM and RAM on DSPs also allows operation as a microcomputer requiring few external resources.
- Word length: DSPs are typically 16-bit or 32-bit processors. This factor has advantages over 8-bit controllers in the areas of dynamic range and sampling rate. For example, an 8-bit microcontroller has to sample twice to read a 12-bit value from an ADC, whereas a DSP must sample only once to read a 12-bit value.

When considering a typical control system, the need for a mutlichannel ADC becomes quite evident. Figure 6 shows a typical feedback or compensator diagram. The calculation of the error signal as well as the digital controller function is performed by the DSP. The sensor monitors a device (in this case a servomotor). A correction signal is generated by the A/D *y(n)* and input to the DSP. The connection signal changes the value of *u(n)*. Depending on the complexity of the control system, there may be a number of these feedback paths each monitoring a different system variable. There also may be a number of input references that control different aspects of the system, which creates the need for a multichannel ADC.

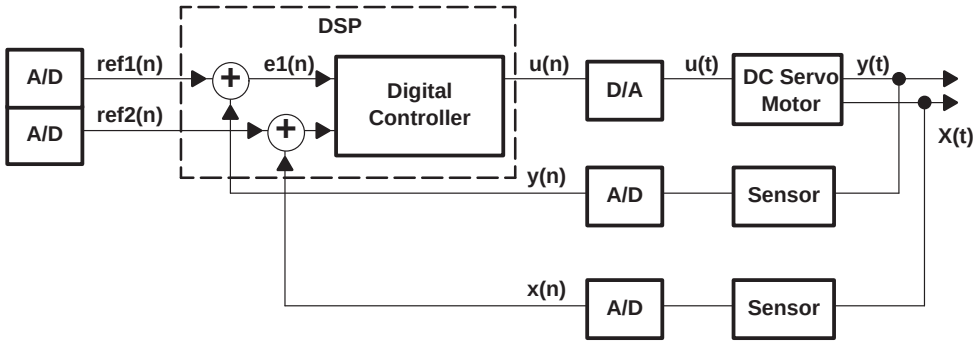


Figure 6. Compensator Block Diagram

Figure 7 shows the control and feedback system for a dc brushless servomotor with some of the variables that can be monitored. The typical word length of an ADC in such a control system is 12-bits, so the TLC2543 is suitable for this application. The sampling rates of the different variables would, however, be different and also indirectly related to the maximum sampling rate of the ADC. If there were 11 inputs for the TLC2543 and the 16-bit mode is used, each input would be sampled at approximately 5 kHz. Due to the ease of programming the sampled channel with the TLC2543, each variable could be sampled at a different frequency depending on the loop time-constant containing that variable.

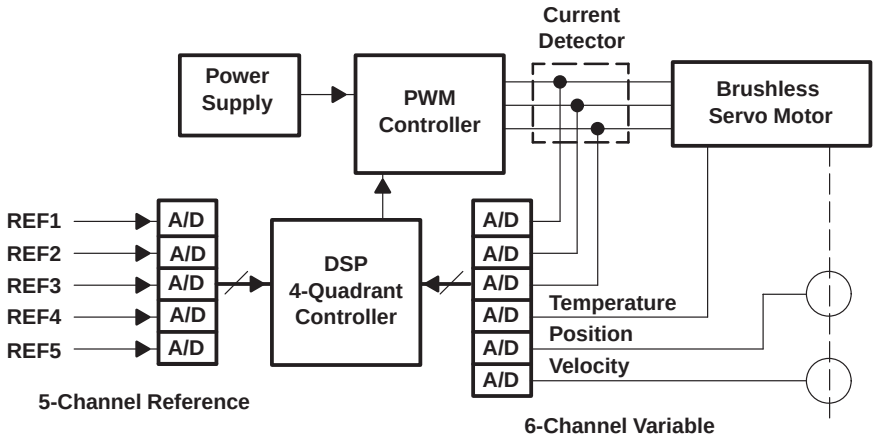


Figure 7. Brushless DC Servomotor Feedback System

References

1. *TMS320C2x User's Guide*, SPRU014C, Texas Instruments Incorporated, 1993.
2. *TLC2543 Data Sheet*, SLAS079A, Texas Instruments Incorporated, 1993.
3. Yasuhiko Dote, *Servo Motor and Motion Control Using Digital Signal Processors,* Prentice Hall, 1990.

Appendix A Software Listing

```
T2543.ASM
;***********************************************************************
;** Module:    tlc2543
;**
;** File Name: T2543.ASM
;**
;**-----------------------------------------------------------------------
;**
;**      Project:          TLC2543 Application note
;**
;**      Operating System: DOS
;**
;**      Language:         TMS320C2x Assembler
;**
;***********************************************************************
   .mmregs
   .def    Entry            ; entry point
   .def    TLCConfig, Temp
   .ref    Input
   .sect   ”program”
Entry
   dint                     ; disable interrupts
   ldpk    0                ; Data page = 0
   rsxm                     ; no sign extension
   spm     0                ; no PREG shift
   cnfd                     ; configure B0 as data (default)
;***********************************************************************
;**      Setup analogue interface and serial port
;***********************************************************************
   fort    0                ; 16 bit transfers
   stxm                     ; FSX is output
   rfsm                     ; FSR/FSX needed for 1st transfer
    in     TLCConfig, PA7   ; read control word
    lac    TLCConfig
    and    #0ffh            ; mask important part
    sfl                     ; shift to ACCH
    sfl;
    sacl   DXR,6            ; for sending on next transfer
    in     Temp, PA5        ; /CLR external flip flop
                            ; hi on /CS
    rxf                     ; 1 Clock for frame sync of DSP
    nop
```

```
    sxf
    nop
    in      Temp, PA6          ; Hi to lo on /CS
    rpt     #15                ; Implement 1.5us delay
    nop
    .loop   16                 ; generate 16 clocks
       rxf
       nop
       sxf
       nop
       .endloop
       in   Temp, PA6          ; Lo to hi on /CS
       lacl DRR                ; read data input
       sac  Input              ; store in memory location
;***********************************************************************
;**         Enable Interrupts
;***********************************************************************
      lalk  #07d0h
      sacl  PRD                ; set timer for 200us interrupt
      lack  008h               ;
      sacl  IMR                ; Enable TINT only
;***********************************************************************
;**         Background task
;***********************************************************************
loop      idle                 ; Wait for interrupt
          nop                  ; < application runs here >
          b     loop
;***********************************************************************
;**       Data allocation
;***********************************************************************
TLCConfig .usect ”scratch”,1
Temp      .usect ”scratch”,1
     .end
```

T2543VEC.ASM

```
;***********************************************************************
;**
;** Module:     2543VEC
;**
;** File Name:  T2543VEC.ASM
;**
;**-----------------------------------------------------------------------
;**
;**      Project:          TLC2543 Application note
;**
;**      Operating System: DOS
;**
;**      Language:         TMS320C2x Assembler
;**
;**-----------------------------------------------------------------------
;**
;**     Description :
;**
;**            Interrupt vectors and ISR's
;**
;**
;***********************************************************************
            .mmregs
            .refEntry, TLCConfig, Temp
            .defInput
            .sect   "inttable"
            .title  "Init Interrupt Table"

start       B       Entry,*       ; Boot vector
Int0        B       Int0,*        ; Int 0 vector
Int1        B       Int1          ; Int 1 vector
Int2        B       Int2          ; Int 2 vector
            .space  16*16
            B       Tint          ; Timer Interrupt
Rint        B       Rint          ; RINT
Xint        B       Xint          ; XINT
Trap        B       Trap          ; Trap

            .sect   "isr"
            .title  "Interrupt Service Routines"
```

```
;***********************************************************************
;**        Timer ISR
;***********************************************************************
; Entry : Timer period has expired.
*
* Read in TLC control word from port 7
* Format :
;
;         d76543210  0/1
;          |||||||+------ Binary/2s complement
;          ||||||+------- MSB / LSB first
;          ||||++-------- 01b (8bit), x0b (12bit), 11b (16bit) format
;          ++++---------- channel address 0–10
;          test voltage   11–14
;          soft powerdown 15
;
Tint
      in     TLCConfig, PA7     ; read control word
      lac    TLCConfig
      and    #0ffh              ; mask important bit
      sfl                       ; shift into ACCH
      sfl                       ;
      sacl   DXR,6              ; for sending on next transfer
      in     Temp, PA6          ; Hi to lo on /CS
      rpt    #15                ; implement 1.5µs delay
      nop
      .loop  16                 ; generate 16 clocks
      rxf                       ; low
      nop                       ; dummy cycle
      sxf                       ; high
      nop                       ; dummy cycle
      .endloop
      in     Temp, PA6          ; Lo to hi on /CS
      lacl   DRR                ; Read sample
      sacl   Input              ; store sample
      eint
      ret
      .bss    Input,1           ; Location of input
      .end
```

```
T2543.CMD
/*********************************************************************
**
**     Module:     Linker command file for T2543
**
**     File Name:  T2543.CMD
**
**-------------------------------------------------------------------
**
**     Project:          TLC2543
**
**     Operating System: DOS
**
**     Language:         TMS320C25
**
*********************************************************************
T2543.obj
T2543vec.obj
MEMORY
{
    PAGE 0 : prg_ext  (RW)   : origin =  0000h  length =  400h
    PAGE 1 : B2_mem   (RW)   : origin =  0060h  length =  01fh
    PAGE 1 : B0_mem   (RW)   : origin =  0200h  length =  0ffh
    PAGE 1 : B1_mem   (RW)   : origin =  0300h  length =  0ffh
}
SECTIONS
{
    program : {
                 *(inttable)   /* first store the interrupt-table  */
                 *(program)    /* prg sections of all object files */
                 *(isr)/* interrupt service routines */
              }  > prg_ext PAGE 0
    B2_data : {
                 *(.bss)
                 *(scratch)
              }  > B2_mem PAGE 1
    .bss    : {                  /* dummy declaration*/
              }  > B2_mem PAGE 1
}
```