Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP

Literature Number: SLAA025
December 1997

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

List of Tables

Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP

ABSTRACT

This application report provides a design solution for the interface between the TLV1544 10-bit serial-output analog-to-digital converter and the TMS320C50 digital signal processor. The report includes hardware requirements and three software application examples.

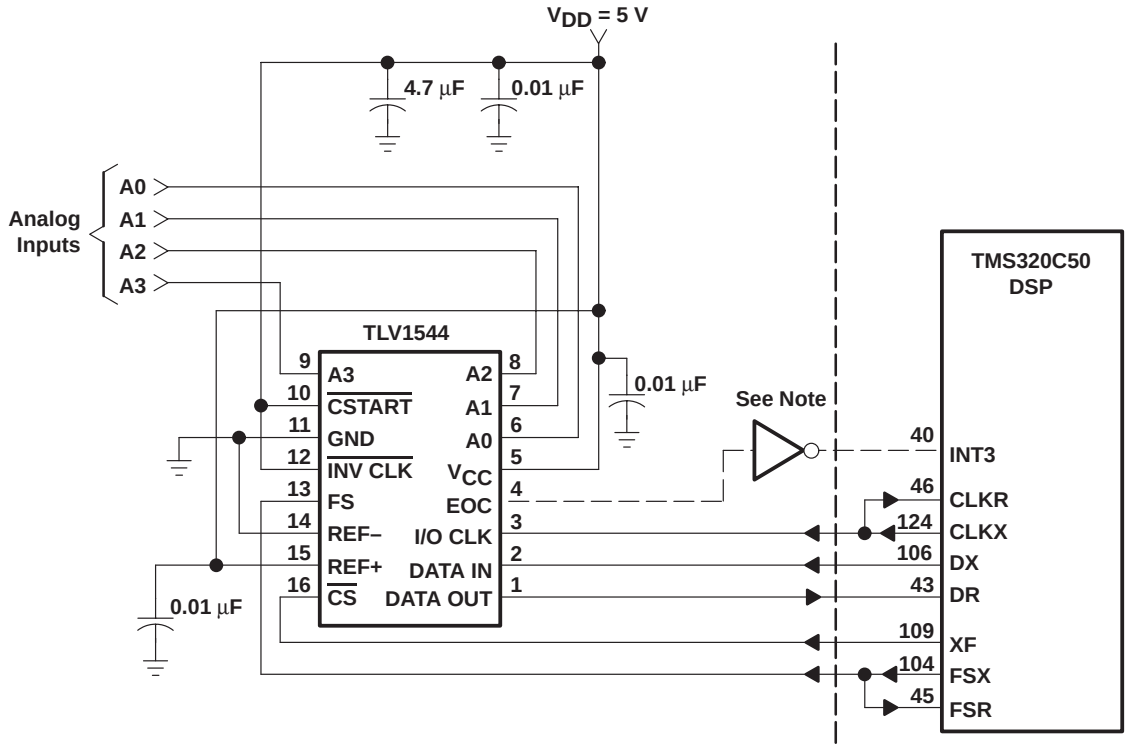
1 Introduction

The analog-to-digital (A/D) interface can present a significant design problem because hardware and software must work together across the interface to produce a usable, complete design. This application report provides a design solution for the interface between the TLV1544 10-bit serial-output analog-to-digital converter (ADC) and the TMS320C50 digital signal processor (DSP). It also provides reference information for further development of hardware and software, especially when using the TLV1548 companion device to the TLV1544.

The information presented describes the hardware and software needed to bridge between the system analog signals and the digital signal processing. The first six sections describe the basic operation of the TLV1544, the TLV1548, and the DSP interface. Sections 7 through 10 contain three example software applications. For additional information and for reference, see the following related documents:

- TMS320C5x User's Guide, literature number SPRU056
- TMS320C5x Data Sheet, literature number SPRS030
- TLV1544 Data Sheet, literature number SLAS139

Figure 1 is a schematic diagram of the hardware.



NOTE: Only example program 1 uses the inverter and the connection shown in dotted lines. In programs 2 and 3, EOC is not required.

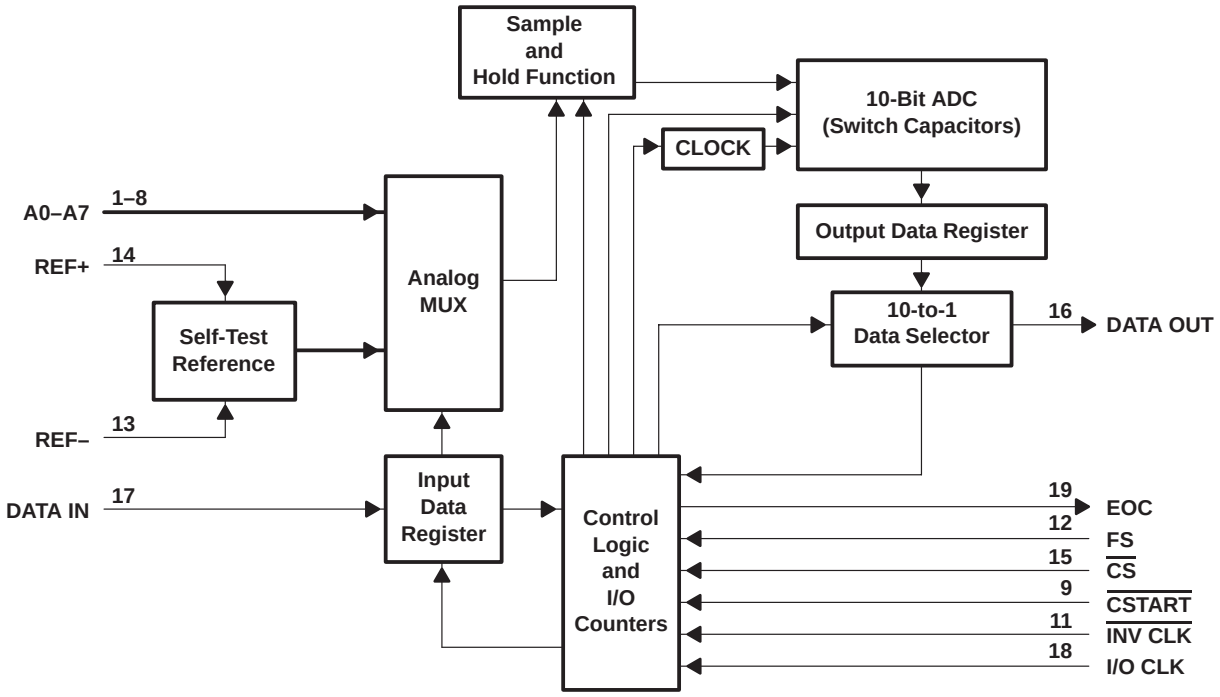
Figure 1. Schematic Diagram

2 TLV1544 and TLV1548 Overview

The TLV1544 and TLV1548 are CMOS 10-bit switched-capacitor successive approximation ADCs. Figure 2 shows the functional block diagram for the devices. The only functional difference is that the TLV1544 has four multiplexed analog inputs and the TLV1548 has 8 multiplexed analog inputs.

Each device has a chip-select ($\overline{CS}$), input-output clock (I/O CLK), data input (DATA IN) and serial data output (DATA OUT). An additional frame sync (FS) input initiates data transfer when using a DSP and connects to the DSP serial port FSX pin.

The $\overline{CSTART}$ input is not used in this application and is tied high. Since the DSP transmit clock, CLKX, already provides the proper phase for the ADC I/O CLK input, the INV CLK terminal is tied high.



Terminals shown are for the DB package.

Figure 2. Functional Block Diagram of the TLV1544 and TLV1548

3 Operational Overview

This application report discusses the software and hardware interface for the TLV1544. The software, however, remains basically the same for the TLV1548 except for the addressing of eight analog inputs instead of four.

This section describes the overall operational sequence of the A/D interface.

3.1 Signal Sequence

As shown in the timing sequence in Figure 3, a high level on the $\overline{CS}$ pin disables the DATA IN and takes DATA OUT to a high impedance state. When taken low, $\overline{CS}$ enables the device inputs, but no data is transferred until the falling edge of FSX is received from the DSP at the FS input of the TLV1544/TLV1548.

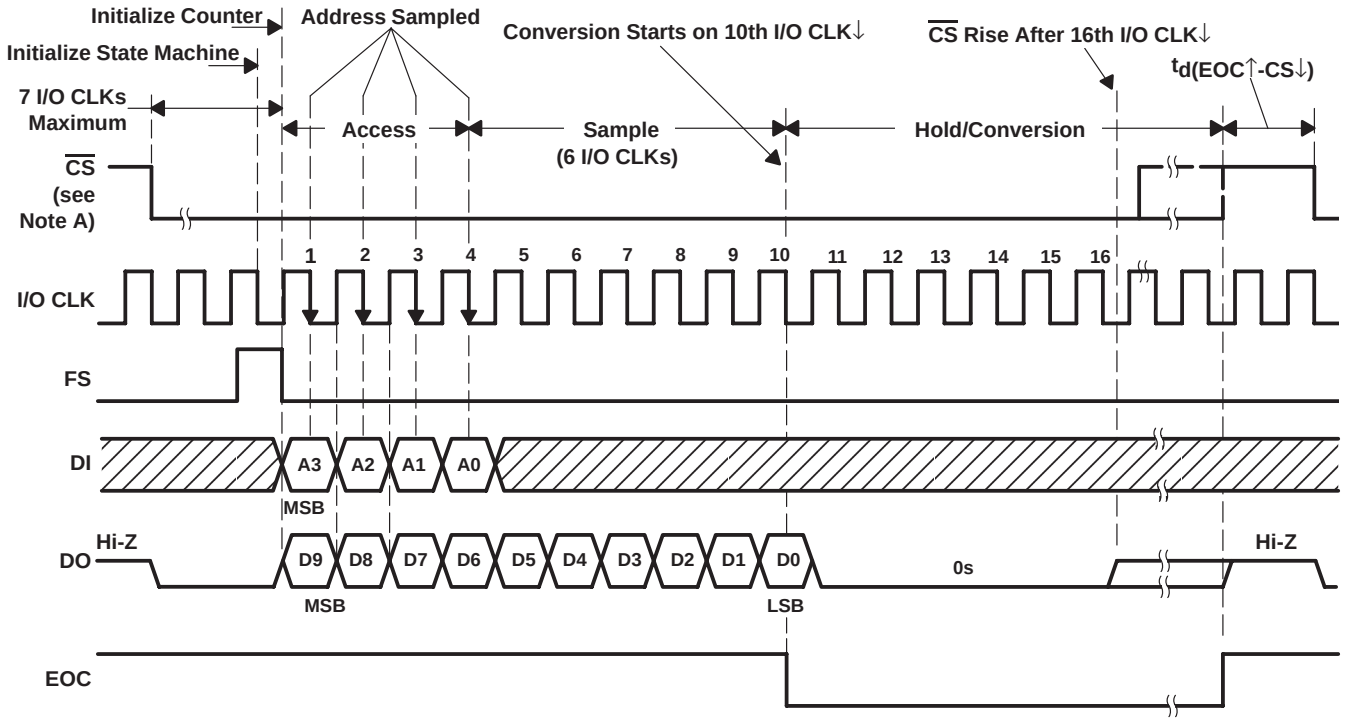
Due to internal counter timing, the maximum allowable time between the falling edge of $\overline{CS}$ and the falling edge of FSX is seven I/O CLK periods. In terms of the software, this means that the high-to-low transition of the DSP XF signal must be at or within 28 instruction cycles of the load DXR command (See the ADC/DSP section). This maximum time is shown as FSD_{max} in Figure 6.

After the falling edge of the DSP FSX, the TLV1544 recognizes the data clock, CLKX, as I/O CLK and receives the 4-bit analog channel address at DATA IN on the first four falling edges of the data clock. DATA IN ignores the remaining 12 clocks.

The channel address selects which analog input channel is converted.

The following six clock periods are the TLV1544 analog sampling time.

During the same 16-clock interval, DATA OUT provides the 10-bit conversion result, padded with 6 zeroes, on the 16 valid clock rising edges after FS goes low. The 10th falling edge of the I/O CLK holds the analog input signal, begins conversion, and takes EOC low indicating conversion start. When conversion is complete, the EOC output goes high indicating conversion is complete and the digital conversion result is in the output register. The DSP XF takes CS back to a high level and the sequence can be repeated.



NOTE A: To minimize errors caused by noise at $\overline{CS}$, the internal circuitry waits for a setup time after $\overline{CS}\downarrow$ before responding to control input signals. No attempt should be made to clock in input data until the minimum $\overline{CS}$ setup time elapses.

Figure 3. DSP Interface Timing (16-Clock Transfer, Normal Sample Mode, $\overline{INVCLK}$=High)

3.2 Reference Voltage Inputs

The voltage difference between the REF+ and REF– terminals determines the analog input range, i.e., the upper and lower limits of the analog inputs that produce the full-scale (output data all 1s) and zero-scale (output data all 0s) readings, respectively. The absolute voltage values applied to REF+, REF–, and the analog input should not be higher than the positive supply or lower than ground. The digital output is full scale when the analog input is equal to or higher than the voltage on REF+ and is zero scale when the input signal is equal to or lower than REF–. In this application, the REF– terminal is connected to ground.

3.3 Input Data Bits

DATA IN is internally connected to a 4-bit serial input data register. The input data for the ADC are control data which select a different analog input channel or different mode of operation. The DSP provides the data word with the MSB first. Each data bit clocks in on the falling edge of the I/O CLK sequence. The software program modes are shown in Table 1.

Table 1. TLV1544/1548 Software Programmed Operation Modes

FUNCTION SELECT	INPUT DATA BYTE A3 – A0		COMMENT†
	BINARY	HEX	
Analog channel A0 for TLV1548 Selected	0000b	0h	Analog channel 0 for TLV1544 Selected
Analog channel A1 for TLV1548 Selected	0001b	1h	
Analog channel A2 for TLV1548 Selected	0010b	2h	Analog channel 1 for TLV1544 Selected
Analog channel A3 for TLV1548 Selected	0011b	3h	
Analog channel A4 for TLV1548 Selected	0100b	4h	Analog channel 2 for TLV1544 Selected
Analog channel A5 for TLV1548 Selected	0101b	5h	
Analog channel A6 for TLV1548 Selected	0110b	6h	Analog channel 3 for TLV1544 Selected
Analog channel A7 for TLV1548 Selected	0111b	7h	1001b
Software power down set	1000b	8h	No conversion result (cleared by any access)
Fast conversion rate (10 μs) set	1001b	9h	No conversion result (cleared by setting to fast)
Slow conversion rate (40 μs) set	1010b	Ah	No conversion result (cleared by setting to slow)
Self-test voltage ($V_{ref+} – V_{ref–}$)/2 selected	1011b	Bh	Output result = 200h
Self-test voltage $V_{ref–}$ selected	1100b	Ch	Output result = 000h
Self-test voltage V_{ref+} selected	1101b	Dh	Output result = 3FFh
Reserved	1110b	Eh	No conversion result
Reserved	1111b	Fh	No conversion result

†Except as noted in the Comments column, the Input Function Byte codes select the same function in the TLV1548 and the TLV1544 ADCs.

4 The ADC/DSP System

The software configures the DSP serial port to the 16-bit master mode so that the DSP generates the frame sync signal at FSX and the data clock at CLKX serial port terminals. From the hardware schematic, the connections between the DSP and the ADC are as follows.

Table 2. DSP/ADC Interconnection

FROM DSP	FROM ADC	TO DSP	TO ADC
FSX		FSR	FS
CLKX		CLKR	I/O CLK
DX			DATA IN
	DATA OUT	DR	
XF			CS

The following statements describe the generation and application of the configuration and control signals. (See Figure 1 and the TLV1544 functional block diagram).

1. The DSP CLKX output provides a 5-MHz data clock, which is a divide-by-4 of the DSP master clock.
2. The DSP DX output supplies the 16-bit control to the TLV1544 at DATA IN.
3. The TLV1544 DATA OUT serial output provides the digital conversion results to the DSP DR terminal.
4. The falling edge of the frame sync signal (FSX) of the DSP serial port initiates the data transfer between the DSP and the ADC.

After the falling edge of FSX, the next 16 data clocks transfer data into the DSP DR terminal and out of the DX terminal. Since this DSP/ADC interface is synchronous, the FSX signal is sent to the FSR terminal and the CLKX is sent to the CLKR terminal.

5 The DSP Serial Port

The serial port provides direct communication with serial I/O devices and consists of six basic signals and five registers. The DSP internal serial port operation section discusses the registers.

The six signals are:

- CLKX – The serial transmit clock. This signal clocks the transmitted data from the DX terminal to the DATA IN terminal of the TLV1544.
- CLKR – The serial receive clock. This signal clocks data into the DSP DR terminal provided from the TLV1544 DATA OUT terminal.
- DX – Data transmit. From this terminal the DSP transmits 16-bit data to the DATA IN terminal of the TLV1544.
- DR – Data receive. The DSP receives 16-bit data from the DATA OUT terminal of the TLV544 into this terminal
- FSX – Frame sync transmit. This signal frames the transmit data. The DSP begins to transmit data from DX on the falling edge of FSX and continues to transmit data for the next 16 clock cycles from the CLKX terminal. The FSX signal is applied to the TLV1544 FS terminal.
- FSR – Frame sync receive, This signal frames the receive data. The DSP begins to receive data on the falling edge of FSR and continues to recognize valid data for the following 16 clocks from CLKR.

Table 3 lists the serial port pins and registers.

Table 3. DSP Serial Port Signals and Registers

PINS	DESCRIPTION	REGISTERS	DESCRIPTION
CLKX	Transmit clock signal	SPC	Serial port control register
CLKR	Receive clock signal	DXR	Data transmit register
DX	Transmitted serial data signal	DRR	Data receive register
DR	Received serial data signal	XSR	Transmit shift register
FSX	Transmit frame synchronization signal	RSR	Receive shift register
FSR	Receive frame synchronization signal		

For this application the DSP serial port is programmed as the master, so the CLKX output is fed to the CLKR terminal and the FSX output is fed to the FSR terminal.

6 Other DSP/TLV1544 Signals

The DSP XF terminal generates the chip select signal applied to the $\overline{CS}$ terminal of the TLV1544. The inverted TLV1544 end-of-conversion (EOC) output signal drives the DSP INT3 terminal. The rising edge of EOC determines the time that the current digital conversion result is available to be read by the DSP.

6.1 DSP Internal Serial Port Operation

Three signals are necessary to connect the transmit pins of the transmitting device with the receive pins of the receiving device for data transmission. The transmitted serial data signal (DX) sends the actual data. FSX initiates the transfer (at the beginning of the packet), and CLKX clocks the bit transfer. The corresponding pins on the receive device are DR, FSR and CLKR, respectively.

Referring to Figure 4, the transmit data is written to the DXR, while received data is read from the DRR. A transmit is executed by writing data to the DXR, which copies the data to the XSR when the XSR is empty. The XSR manages the shifting of the data to the DX pin, thus allowing another write to DXR as soon as the DXR-to-XSR copy is completed. Upon completion of the DXR-to-XSR copy, a 0-to-1 transition occurs on the XRDY bit in the SPC and generates an XINT. The process is similar on the receiving end. Upon completion of the RSR-to-DRR copy, a 0-to-1 transition occurs on the RRDY bit in the SPC and generates a RINT.

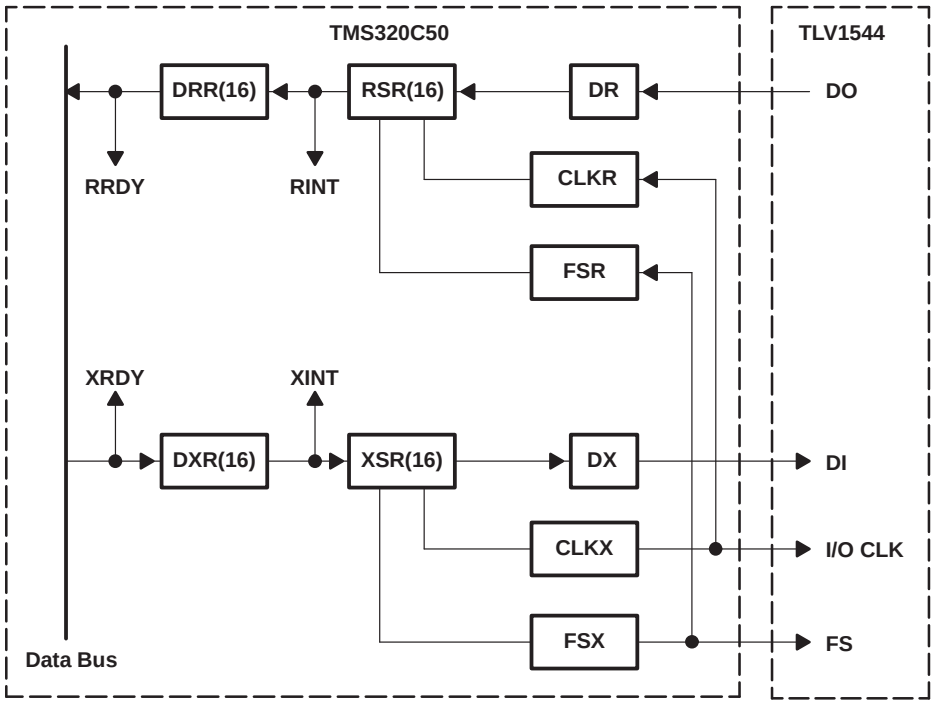


Figure 4. Internal DSP Serial Port Block Diagram

7 Software Overview

The first software application uses interrupt processing and stores 256 conversion-result data-points for each of the four analog input channels, A0-A3. The second application uses wait instructions to allow for the ADC maximum conversion time before reading analog channel A3 data and does not store a sequence of results. The third application is similar to the second application but, additionally, stores 4096 data points from channel A3 in memory for observation or further processing if desired. The CPU CLK for the TMS320C50 DSP is 20 MHz (50ns). The TLV1544 I/O CLK is driven by CLKX, which is 5 MHz (200ns).

The second and third applications require no external logic and therefore provide a direct interface to the TMS320C50 serial port. Figure 1 shows the hardware schematic with the associated software shown in the Program 1, Program 2, and Program 3 sections. Figures 5, 7, and 8 show the flow charts.

All three programs start with a common initialization procedure for the DSP followed by the initialization of the serial port.

The following steps initialize the DSP:

1. Disable the global interrupts.
2. Set the data page pointer to 0h.
3. Set the Processor Mode Status Register to 0834h (PMST=0834h).

The individual bits within the PMST have the following functions:

- The interrupt vector pointer bits are set to 00001 (IPTR=00001), which re-maps the Interrupt vectors to the on-chip SRAM from address 800h to 82Fh.
- The RAM overlay bit (OVLY) and the RAM bit (RAM) are set to 1 (OVLY = RAM = 1) which maps the on-chip single-access RAM (SARAM) into both the data memory and the program memory space.
- The microprocessor/microcomputer bit is set to 1 (MP=1) so that the on-chip ROM is not addressable in program memory space.

The next two steps initialize the serial port:

1. Set the serial port control register to 038h (SPC = 038H)

The individual bits within the SPC now contain the following functions:

- The frame sync mode bit is set to 1 (FSM = 1) to allow burst mode operation.
- The clock mode bit is set to 1 (MCM = 1) to set the transmit clock CLKX to 1/4 of the DSP master clock of 20 MHz (CLKX frequency = 5 MHz)
- The transmit mode bit is set to 1 (TSM = 1) to generate the frame sync internally as required for the data transfer initiation. FSX is now programmed as an output.

2. Set the serial port control register to 0f8h (SPC = 0F8h)

- Writing 1's to the transmitter reset field ($\overline{\text{XRST}}$) and to the receiver reset field ($\overline{\text{RRST}}$) activates the serial port transmitter and receiver.

The following sections describe the individual programs.

8 Program 1

This program uses one of the DSP interrupts ($\overline{\text{INT3}}$) to read and to store 256 data samples. Figure 5 shows its flowchart.

The program package includes:

- TLV1544.ASM Main Routine
- TLV1544.CMD Command File
- VECTORS.ASM Interrupts Vectors
- VALUES.ASM Variable Values
- SPINI.ASM Serial Port Initialization Routine
- CHANNELS.ASM Channels Selection Routine
- TLV1544.PPT Program Flow Chart
- TLV1544.OUT COFF File Output

The CPU CLK for the C50 DSP is 20 MHz. The I/O CLK for the TLV1544 is driven by CLKX, which is 5 MHz.

The program executes the following steps:

1. Initializes the C50 DSP.
2. Initializes the serial port.
3. Sends the operation mode (here for fast conversion rate) to the TLV1544 ADC.
4. Selects a specific analog input channel. (This program selects all four input channels sequentially.)
5. Upon the transfer of data from RSR to DRR, a RINT occurs which causes the DSP to load the content of DRR into ACC. The data is then shifted to the right six times then stored into the memory.
6. A transition of the ADC's EOC output from low to high causes, through the inverter, a valid interrupt signal at $\overline{\text{INT3}}$. The following interrupt routine initiates a low-to-high transition at the DSP XF terminal, which is connected to the ADC chip-select pin ($\overline{\text{CS}}$) and de-activates the TLV1544.
7. After obtaining all data, the DSP powers the ADC down via a separate power-down routine.
8. The data is now available for use in customer-defined functions (algorithms).

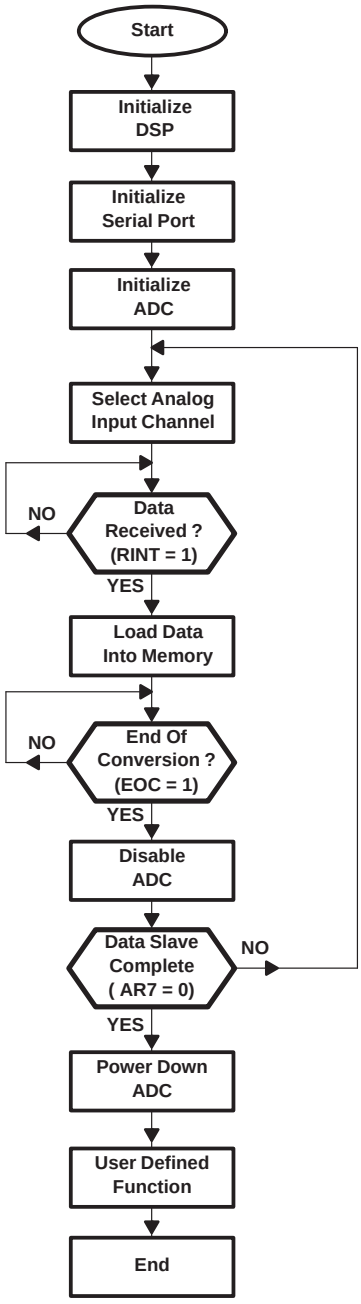


Figure 5. TLV1544 to TMS320C50 DSP Interface Program Flow Chart: Program 1

8.1 Program 1 Listing

```
; *File: TLV1544.ASM Main routine for the DSP C50 –TLV1544 INTERFACE *
   .title    ”TLV1544C ADC Interface routine”
;*****************************************************************
; This routine interfaces the ’C50 DSP to the TLV1544 ADC via the
;     DSP serial port.
        .mmregs
        .sect ”.vectors”
        .copy ”vectors.asm”
        .sect ”.data”
        .copy ”values.asm”
        .sect ”.text”
Start:
; ** DSP INITIALIZATION
        SETC  INTM                ; DISABLE GLOBAL INTERRUPTS
        LDP   #0                  ; SET DATA PAGE POINTER
        OPL   #0834h, PMST        ; Configure PMST
; ** SERIAL PORT INITIALIZATION
        CALL  SP_INI
; ** SEND OPERATION MODE TO TLV1544
        LAR   AR1,#1h             ; AR1=1 when EOC = 1
        LAR   AR6,#1000h          ; Set start address of memory
                        ;   range to 1000h
        CLRC  XF                  ; Enable Chip Select
        SPLK  #fast_conv,DXR      ; Send fast conversion mode to
                                  ; TLV1544
        CALL  Wait                ; Wait for EOC and RINT interrupts
; ** SCAN AND COLLECT DATA FROM ADC CHANNELS
        CALL  Channels
; ** TLV1544 POWER DOWN AFTER TRANSITIONS
        CLRC  XF                  ; Enable Chip Select
        SPLK  #power_down,DXR  ; Send power down mode to TLV1544
; ** APPLICATION FUNCTION(S)
; Function:
;
;
End_Loop:
        NOP
        NOP
        NOP
       B  End_Loop
; **    REPEAT ROUTINE FOR NUM_DATA TIMES
Loop:
       CLRC   XF              ; Enable Chip Select
       SAR    AR5,DXR         ; Send analog channel mode to TLV1544
       CALL   Wait            ; Wait for EOC and RINT interrupts
       MAR    *,AR7           ; Select AR7 (number of data)
       BANZ   Loop            ; Return to loop until AR7=0
        RET
; **    WAIT FOR RINT AND EOC
```

```
Wait:
        BIT   AR1,15        ; If AR1 L.S.B. = 1, then TC =1
        BCND  Wait,TC          ; Go back to Wait if AR1=1
        LAR   AR1,#1h          ; Reset AR1 to 1
        RET                    ; Return from Call
; **    EOC FROM LO TO HI, INT3 Active
End_Conv:
        SETC  XF               ; Disable Chip Select
        LAR   AR1,#0h          ; Set AR1 = 0
        RETE                   ; Return to Wait loop
; **    RSR TO DRR COMPLETE, RINT Active
Load_Data:
; Data collection from ADC
        LAMM  DRR              ; Load data from DRR to ACC
        CLRC  SXM              ; Clear sign bit
        RPT   #5               ; Shift right 6 times
        SFR
        MAR   *,AR6            ; Select AR6 (start address of memory)
        SACL  *+               ; Save ACC into memory
        RETE                   ; Return to Wait loop
; END OF RECEIVING
        .copy ”spini.asm”
        .copy ”channels.asm”
        .end


/* File: TLV1544.CMD COMMAND FILE                                    */
/*        .title ”COMMAND FILE FOR TLV1544.ASM”                      */
/*********************************************************************/
/*   This CMD file allocates the memory area for the TLV1544         */
/* interface Program                                                 */
/*                                                                   */
 –M TLV1544.MAP
 –O TLV1544.OUT
 –v0
 –e Start
 MEMORY
 {
   PAGE 0:   VECT:     origin = 0800h, length = 0030h
    PROG:              origin = 0830h, length = 0200h
   PAGE 1:   RAMB0:    origin = 0100h, length = 0200h
    RAMB1:             origin = 0300h, length = 0200h
    RAMB2:             origin = 0060h, length = 0020h
 }
```

```
; ***************************************************************
; This file contains the control data for the TLV1544 and the starting
; address and the range of the used memory space.
channel_0 .set    0000h    ; Select TLV1544 channel 0
channel_1 .set    2000h    ; Select TLV1544 channel 1
channel_2 .set    4000h    ; Select TLV1544 channel 2
channel_3 .set    6000h    ; Select TLV1544 channel 3
power_down  set   8000h    ; Software Power Down
fast_conv .set    9000h    ; Fast Conversion Rate
slow_conv .set    0A000h   ; Slow Conversion Rate
test_200  .set    0B000h   ; (Vreg+ – Vreg–)/2
test_000  .set    0C000h   ; Vreg–
tets_3FF  .set    0D000h   ; Vreg+
num_data_0  set    100h    ; Number of Data from channel 0
num_data_1  set    100h    ; Number of Data from channel 1
num_data_2  set    100h    ; Number of Data from channel 2
num_data_3  set    100h    ; Number of Data from channel 3
data_loc_0   .set    1000h    ; Start data location for channel 0
data_loc_1   .set    1200h    ; Start data location for channel 1
data_loc_2   .set    1400h    ; Start data location for channel 2
data_loc_3   .set    1600h    ; Start data location for channel 3


; * File: SPINI.ASM Serial Port initialization code for the DSP ’C50 *
         .title ”Serial Port Initialization Routine”
; ***************************************************************
; This routine initializes the Serial Port.
; The various bits are set to interface the TLV1544 ADC to the
; TMS320C50 DSP via its serial port.

; ** SERIAL PORT INITIALIZATION
SP_INI:
        SETC    INTM             ; Disable interrupts
        SPLK    #038h,SPC        ; Configure SPC
        SPLK    #0F8h,SPC        ; Activate Transmitter and Receiver
        SPLK    #14h,IMR         ; Unmask RINT and INT3
        CLRC    INTM             ; Enable interrupts
        RET                      ; Return from Call


;*File: CHANNELS.ASM Serial Port initialization code for the DSP C50*
       .title ”Channels Selection Routine”
; ***************************************************************
```

```
; This routine selects the individual analog input channels of the
; TLV1544. The channel number, the data location, and the number of
; data samples can be changed in the file values.asm.
Channels:
        LAR     AR5,#channel_0   ; SELECT CHANNEL 0
        LAR     AR6,#data_loc_0  ; DATA LOCATION 0
        LAR     AR7,#num_data_0  ; NUMBER OF DATA 0
        CALL    Loop
        LAR     AR5,#channel_1   ; SELECT CHANNEL 1
        LAR     AR6,#data_loc_1  ; DATA LOCATION 1
        LAR     AR7,#num_data_1  ; NUMBER OF DATA 1
        CALL    Loop

        LAR     AR5,#channel_2      ; SELECT CHANNEL 2
        LAR     AR6,#data_loc_2  ; DATA LOCATION 2
        LAR     AR7,#num_data_2  ; NUMBER OF DATA 2
        CALL    Loop

        LAR     AR5,#channel_3   ; SELECT CHANNEL 3
        LAR     AR6,#data_loc_3  ; DATA LOCATION 3
        LAR     AR7,#num_data_3  ; NUMBER OF DATA 3
        CALL    Loop
         RET
```

9 Program 2 and Program 3 Wait-Cycles

In program 1 the DSP receives data upon an Interrupt signal from the ADC. In programs 2 and 3 the DSP waits for the maximum conversion time of the TLV1544 to pass before retrieving data.

This wait time is achieved by enabling and disabling the ADC via the $\overline{CS}$ signal for a defined number of wait cycles.

The number of wait cycles the chip-select signal ($\overline{CS}$) is kept low, by the DSP, is defined as CSL (chip select low), and likewise the number of cycles $\overline{CS}$ is kept high is defined as CSH (chip select high).

A wait cycle is executed by two instructions:

- the ”Repeat Next” instruction (RPT #k),
- a ”No Operation” instruction (NOP).

The instruction following the RPT instruction, in this case NOP, is repeated k times.

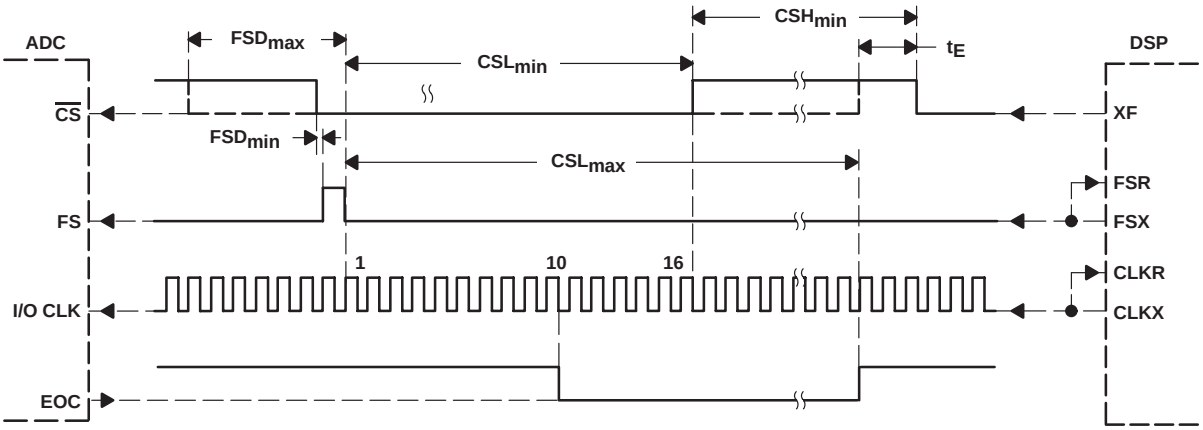


Figure 6. Timing diagram of the control signals between ADC and DSP

The interface timing for the TLV1544 shown in Figure 6 requires $\overline{CS}$ to stay low for the minimum time of 16 I/O clocks after the falling edge of FS. The software achieves this by taking XF low and then immediately initiating a data transfer which generates the required frame synch (FS) pulse.

The number of repetitions of the NOP instruction in order to keep $\overline{CS}$ low for the minimum of 16 I/O clocks is calculated as:

$$CSL = \frac{16}{I/O\ CLK} \times MCLK \tag{1}$$

where MCLK is the master clock of the DSP in MHz, and I/O CLK is the data clock of the TLV1544. In this application MCLK is 20 MHz and I/O CLK is 5 MHz.

With these values, CSL results in:

$$CSL = \frac{16}{5\ MHz} \times 20\ MHz = 64 = CSL_{min} \tag{2}$$

The minimum number of NOP repetitions the RTP instruction has to perform is 64.

The maximum low time for $\overline{CS}$ ends when the EOC signal changes from low to high. Since the maximum conversion time of the TLV1544 is 10 μs, starting with the 10th I/O clock after the falling edge of FS, the formula for CSL is:

$$CSL = \left(\frac{10}{I/O\ CLK} + 10\mu s\right) \times MCLK \quad (3)$$

With the previous values for MCLK and I/O CLK CSL results in:

$$CSL = \left(\frac{10}{5\ MHz} + 10\mu s\right) \times 20\ MHz = 240 = CSL_{max} \quad (4)$$

The maximum number of NOP operations allowed to keep $\overline{CS}$ low is 240.

Once XF or $\overline{CS}$ has been taken high there is a minimum time required for $\overline{CS}$ to stay high before it is taken low again to start the next conversion cycle.

This minimum high time is composed of two values:

- the time difference between the rising edge of $\overline{CS}$ and the rising edge of EOC, and
- the minimum time required for $\overline{CS}$ to stay high after the rising edge of EOC.

The latter value is specified in the TLV1544 data sheet with $t_{d(EOC\uparrow-CS\downarrow)}$ = 100 ns. In this application an additional 100-ns guard-band has been added. The sum of both, $t_{d(EOC\uparrow-CS\downarrow)}$ and the guard-band is defined as $t_E(t_{Extended})$ and is 200 ns wide.

The minimum number of NOP repetitions required to achieve the minimum required high time is defined as CSH (chip select high) and, using the results of equation (4), is calculated as:

$$CSH = (CSL_{max}–CSL) + t_E \times MCLK \quad (5)$$

In programs 2 and 3, CSL is set to 100 CPU cycles and therefore CSH results in:

$$CSH = (240–100) + 200\ ns \times 20\ MHz = 144 = CSH_{min} \quad (6)$$

$\overline{CS}$ needs to be high for at least 144 CPU cycles before initiating a new conversion cycle via a high-to-low transition.

> **NOTE:** All calculated repetition numbers are in fact CPU cycles. Since this application is not time critical the same numbers have been used for the repetition of NOP operations. Finally the best way to use the interface for maximum throughput is to perform the data transfer per Interrupt control. In this case the inverted EOC output signal of the TLV1544 is applied to the $\overline{INT3}$ input of the TMS320C50 DSP (as shown in program 1).

9.1 Program 2

This program waits for the maximum TLV1544 conversion time to pass before retrieving data and does not store any data values. Figure 7 shows the flow chart.

This program package includes:

- P1544.ASM Main Routine

- TLV1544.CMD Command File

The CPU CLK for the C50 DSP is 20 MHz. The I/O CLK for the TLV1544 is driven by CLKX, which is 5 MHz.

The program executes the following steps :

1. Initializes the C50 DSP.
2. Initializes the serial port.
3. Sends the operation mode (here for fast conversion rate) to the TLV1544 ADC.
4. Selects analog input channels A3.
5. Receives data from the TLV1544, perform a right-shift of 6 bits, and save the data in the accumulator.
6. Continues to receive data in an endless loop.

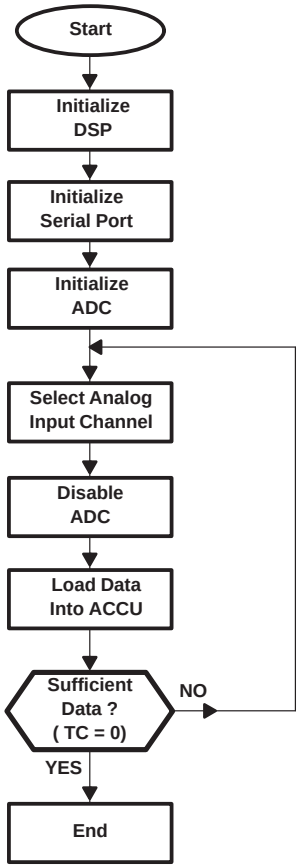


Figure 7. TLV1544 to TMS320C50 DSP Interface Program Flow Chart: Program 2

9.1.1 Program 2 listing

```
; * File: P1544.ASM Main routine for the C50 EVM
*
         .title ”TLV1544 ADC Interface routine”
;
*********************************************************************
; This routine interfaces the ’C5O DSP to the TLV1544 ADC via
;   the DSP serial port.
          .mmregs
         .sect ”.vectors”         ; reserved for interrupt vectors
        .sect ”.data”             ; reserved for values
          .sect ”.text”
start:
;    DSP Initialization
        SETC  INTM             ; disable global interrupts
        LDP   #0               ; set data page pointer
        OPL   #0834h, PMST  ; configure PMST
;    Serial Port Initialization
        SPLK  #038h,SPC        ; Configure SPC
        SPLK  #0F8h,SPC        ; Activate Transmitter and Receiver
;    ADC Initialization
        SETC  TC               ; Set test condition for endless loop
        CLRC  XF               ; Enable Chip Select
        SPLK  #9000h,DXR       ; Send Fast Conversion mode to TLV1544
        RPT   #100             ; Wait 100 cycles
        NOP
        SETC  XF               ; Disable Chip Select
        RPT   #144             ; Wait 144 cycles
        NOP
loop:
        CLRC  XF               ; Enable Chip Select
        SPLK  #6000h,DXR       ; Select TLV1544’s Analog  Channel_3
        RPT   #100             ; Wait 100 cycles
        NOP
        SETC  XF               ; Disable Chip Select
        RPT   #144             ; Wait 144 cycles
        NOP
        LAMM  DRR              ; Load Data from DRR to ACC
        CLRC  SXM              ; Clear Sign bit
        RPT   #5               ; Shift right 6 times
        SFR
        BCND   loop,TC         ; Endless loop as long as TC = 1
        .end
```

```
/* File:  TLV1544.CMD COMMAND FILE
*/
/*        .title "COMMAND FILE FOR P1544.ASM"                          *
/
/*****************************************************************/
/*This CMD file allocates the memory area for the TLV1544          */
/*  interface Program.                                             *
/
/*                                                              */
 -M TLV1544.MAP
 -O TLV1544.OUT
 -v0
 -e Start
 MEMORY
 {
   PAGE 0:   VECT:        origin = 0800h, length = 0030h
             PROG:        origin = 0830h, length = 0200h
   PAGE 1:   RAMB0:       origin = 0100h, length = 0200h
             RAMB1:       origin = 0300h, length = 0200h
             RAMB2:       origin = 0060h, length = 0020h
 }
 SECTIONS
 {
   .text    :{} > PROG PAGE 0
  .vectors : {} > VECT PAGE 0
   .data    :{} > RAMB1 PAGE 1
   .bss     :{} > RAMB2 PAGE 1
 }
```

9.2 Program 3

This program is the same as program 2, except 4096 data points are saved in the memory space from 1000h to 2000h. Figure 8 shows the flow chart.

This interface program package includes:

- S1544.ASM Main Routine
- TLV1544.CMD Command File

The CPU CLK for the C50 DSP is 20 MHz. The I/O CLK for the TLV1544 is driven by CLKX, which is 5 MHz.

The program executes the following steps:

1. Initialize the C50 DSP.
2. Initialize the serial port.
3. Send the operation mode (here for fast conversion rate) to the TLV1544 ADC.
4. Select analog input channels A3.
5. Receive data from the TLV1544 and save data in memory.
6. Continue to receive data for 4096 samples.

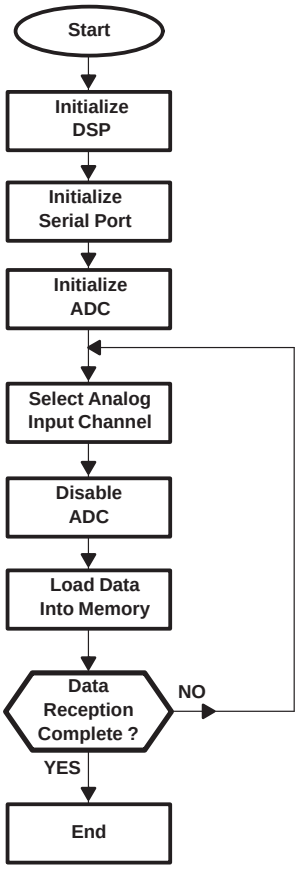


Figure 8. TLV1544 to TMS320C50 DSP Interface Program Flow Chart: Program 3

9.2.1 Program 3 Listing

```
; * File: S1544.ASM Main routine for the C50
*
         .title ”TLV1544 ADC Interface routine”
;
;**********************************************************************
; This routine interfaces the ’C5O DSP to the TLV1544 ADC via the DSP
;   serial port and saves data in the specified memory block.
          .mmregs
        .sect ”.vectors”         ; reserved for Interrupt Vectors
        .sect ”.data”            ; reserved for values
          .sect ”.text”
start:
;   DSP Initialization
        SETC     INTM            ; disable global interrupts
        LDP      #0              ; set data page pointer
        OPL      #0834h, PMST    ; configure PMST
;   Serial Port Initialization
        SPLK     #038h,SPC       ; Configure SPC
        SPLK     #0F8h,SPC       ; Activate Transmitter and Receiver
```

```
                                ; ADC Initialization
        LAR     AR6,#1000h      ; set start address of memory range
                                ; to 1000h
        CLRC    XF              ; Enable Chip Select
        SPLK    #9000h,DXR      ; Send Fast Conversion mode to
                                ;TLV1544
        RPT     #100            ; Wait 100 cycles
        NOP
        SETC    XF              ; Disable Chip Select
        RPT     #144            ; Wait 144 cycles
        NOP
        LAR     AR7,#1000h      ; store #1000h ADC samples
loop:
        CLRC    XF              ; Enable Chip Select
        SPLK    #6000h,DXR      ; Select TLV1544's Analog Channel_3
        RPT     #100            ; Wait 100 cycles
        NOP
        SETC    XF              ; Disable Chip Select
        RPT     #144            ; Wait 144 cycles
        NOP
        LAMM DRR                ; Load Data from DRR to ACC
        CLRC    SXM             ; Clear Sign bit
        RPT     #5              ; Shift right 6 times
        SFR
        CALL load_data
        MAR  *,AR7              ; select AR7
        BANZ loop               ; Return to loop until AR7 = 0
end_loop:
         NOP
         NOP
         NOP
        B    end_loop
load_data:
;   Data collection from ADC
        MAR  *,AR6              ; select AR6
        SACL *+                 ; Save ACC into memory
        RET                     ; Return to loop
;   End of collection
        .end
```

```
/* File: TLV1544.CMD COMMAND FILE                                    *
/
/*        .title "COMMAND FILE FOR S1544.ASM"                        *
/
/*****************************************************************/
/* This CMD file allocates the memory area for the TLV1544        */
/*  interface Program                                             */
/*                                                                */
 -M TLV1544.MAP
 -O TLV1544.OUT
 -v0
 -e Start
 MEMORY
 {
   PAGE 0:    VECT:        origin = 0800h, length = 0030h
              PROG:        origin = 0830h, length = 0200h
   PAGE 1:    RAMB0:       origin = 0100h, length = 0200h
              RAMB1:       origin = 0300h, length = 0200h
              RAMB2:       origin = 0060h, length = 0020h
 }
 SECTIONS
 {
   .text    :{} > PROG PAGE 0
   .vectors :{} > VECT PAGE 0
   .data    :{} > RAMB1 PAGE 1
   .bss     :{} > RAMB2 PAGE 1
 }
```

10 Summary

This application report demonstrates a method for bridging the analog-to-digital interface using the TLV1544 ADC and the TMS320C50 DSP.