TEXAS INSTRUMENTS

Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C203 DSP

Application Report

February 1999 *Mixed-Signal Products*

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

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, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

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

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

List of Figures

List of Tables

Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C203 DSP

David Quach and Thomas Kugelstadt

ABSTRACT

This application report presents two hardware solutions for interfacing the TLV1544 10-bit low-power analog-to-digital converter (ADC) to the TMS320C203 16-bit fixed-point digital signal processor (DSP). The report describes the interface hardware and shows a C-callable software routine which supports the data transfer. In addition, it provides useful hints on the design of a typical system power supply and shows two input buffers for the analog inputs of the ADC.

1 Introduction

The TLV1544/48 is a 10-bit, low-power, successive approximation analog-to-digital converter (ADC) with a conversion time of $t_{conv} \leq 10\ \mu s$. The TLV1544 has four analog inputs; the TLV1548 has eight.

The device operates from a maximum supply voltage of V_{DD} = 5.5 V down to a minimum supply voltage of V_{DD} = 2.7 V, thus making it suitable for portable, low-power applications.

With a 5.5-V supply, a maximum ADC interface clock (I/OCLK) of 10 MHz is possible. With a 2.7-V supply, the maximum clock is 2.89 MHz.

The DSP serial interface port standard configuration provides a 10-MHz clock that is applied to the I/O CLK pin of the ADC to allow for a high data throughput.

2 The System

While this report focuses on the interface between the ADC and DSP (see Figure 1), it includes informative hints on the power supply section and on the buffering of the ADC inputs.

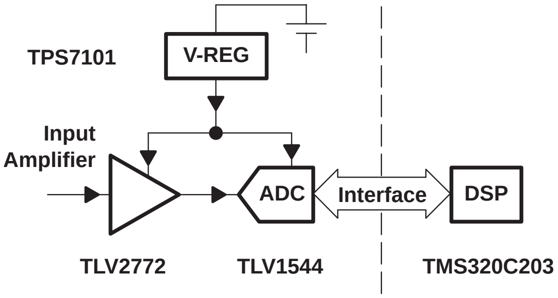


Figure 1. Data Acquisition System Using the TLV1544 ADC

2.1 Standard ADC-DSP Interface

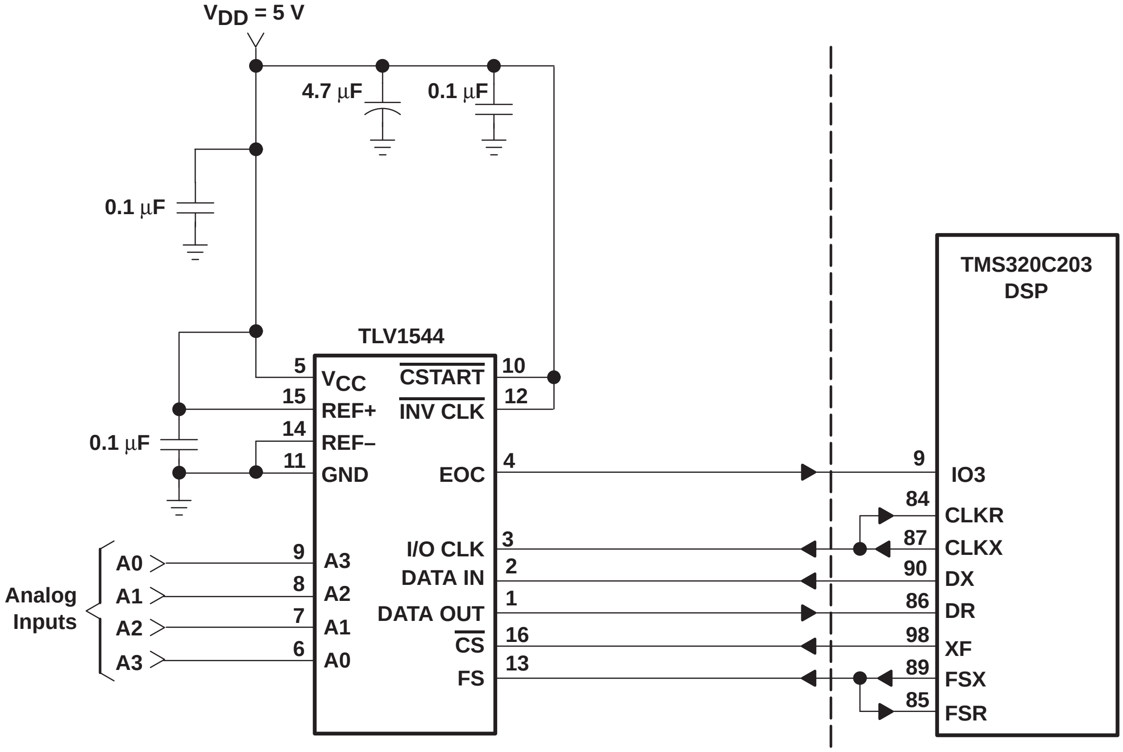
Figure 2 shows the standard ADC-DSP interface. The regulated supply voltage of V_{DD} = 5 V is applied to the ADC V_{CC} pin. A 4.7-µF bulk capacitor keeps the entire circuit supply stable against any significant current changes during ADC operation. The 0.1-µF bypass capacitors keep the ADC supply, as well as the positive reference voltage and the unused digital inputs, ripple-free. The bypass capacitors should be as close as possible to the individual pins.

The positive reference voltage (REF+), is tied to V_{CC}; the negative reference voltage, REF–, is connected to GND. This defines the analog conversion range of the ADC and specifies the maximum input signal level at the analog inputs, A0 to A3.

The unused, low active, digital control pins are tied to V_{CC}.

In the application with fixed data clock, the interface between the TLV1544 ADC and the TMS320C203 DSP requires no additional control logic.

- The DSP sends an initial chip-enable signal via the XF pin to the $\overline{\text{CS}}$ pin of the ADC.
- The transmit clock output of the DSP, CLKX, provides a fixed data clock into the I/O CLK input of the ADC and into the receive clock input, CLKR.
- 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 4-bit control words from the DX output into the DATA IN input of the ADC.
- The ADC clocks digital conversion results out at DATA OUT into the DR pin of the DSP.
- The EOC output of the ADC indicates the end of a conversion with a low-to-high transition. This signals the DSP, through the IO3 input, to start a new data transfer.



NOTE: Fixed data clock is only possible with V_{DD} = 5 V.

Figure 2. Standard ADC/DSP Interface

2.2 Power Supply Circuit

Figure 3 shows a typical voltage regulator schematic using the adjustable low-dropout (LDO) regulator TPS7101. The LDO regulator has a very low dropout voltage of 32 mV at an output current of I_{OUT} = 100 mA.

The very low typical quiescent current of 285 µA remains independent of output loading over the full range of output current, 0 mA to 500 mA.

The TPS7101 regulates an input voltage in the range of 6 to 10 Vdc down to the adjusted output level. In this application the output is adjusted to 5.1 V through the voltage divider R1 and R2.

The equation governing the output voltage is:

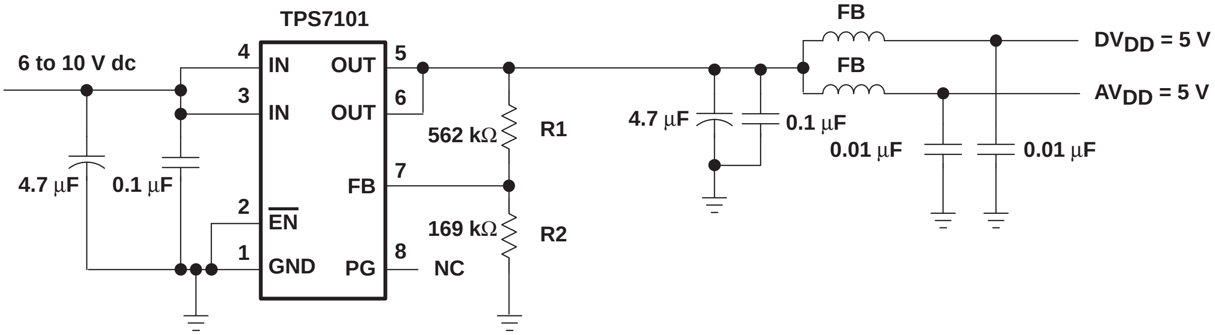
$$V_O = V_{REF} \times \left(1 + \frac{R1}{R2}\right) \text{ with Vref } = \text{ reference voltage of 1.178 V typ.} \quad (1)$$

Resistors R1 and R2 should be chosen for approximately 7 µA divider current. The recommended value for R2 is 169 kΩ with R1 adjusted for the desired output voltage. Smaller resistors can be used, but offer no inherent advantage and consume more power. Larger values of R1 and R2 should be avoided, as leakage currents at the FB pin will introduce an error. Solving Equation 1 for R1 yields a more useful equation for choosing the appropriate resistance:

$$R1 = \left(\frac{V_O}{V_{ref}} - 1 \right) \times R2 \tag{2}$$

Calculating R1 for V_{OUT} = 5.1 V and R2 = 169 kΩ results in:

$$R1 = \left(\frac{5.1\ V}{1.178\ V} - 1 \right) \times 169\ k\Omega = 562\ k\Omega$$



NOTES: A. Bypass capacitors should be placed as close to the device pins as possible. The total ESR of the regulator output capacitors must maintain between 0.7 Ω min and 2.5 Ω max.
B. FB is the Fair-Rite™ #27-44-044447 or equivalent.
Fair-Rite is a trademark of Fair–Rite Products Corporation.
C. NC = no internal connection.

Figure 3. Typical Voltage Regulator Schematic

The input capacitors shown are usually not required; however, the 0.1-µF ceramic bypass capacitor improves load transient response and noise rejection if the TPS7101 is located more than a few inches from the power supply. A higher capacitance electrolytic capacitor may be necessary if large load transients with fast rise times are anticipated.

The chosen output capacitors are required for stability. A low-ESR 4.7-µF solid tantalum capacitor and a 0.1-µF high-frequency ceramic capacitor, connected from the regulator output to ground, are sufficient to ensure stability, provided that the total ESR is maintained between 0.7 Ω and 2.5 Ω.

Two additional low-pass filters, each consisting of a ferrite bead, FB, and a 0.01-µF capacitor, block the digital noise and transients on the digital supply line, DV_{DD} , from the analog supply, AV_{DD}.

For more information on the type of ferrite beads and the selection and type of low-ESR capacitors, refer to the TPS7101 Data Sheet, literature number SLVS092F and the *TLV1544 EVM User's Guide*, literature number SLAU014.

2.3 Analog Input Buffer Circuit

Figure 4 and Figure 5 show the schematics of typical analog input buffers using the TLV2772 dual operational amplifier.

The TLV2772 combines high slew rate and bandwidth, rail-to-rail output swing, high output drive and excellent dc precision. The device provides 10.5 V/ms of slew rate and 5.1 MHz of bandwidth while only consuming 1 mA of supply current per channel.

This ac performance is much higher than current competitive CMOS amplifiers. The rail-to-rail output swing and high output drive make it a good choice for driving the analog input or reference of analog-to-digital converters. The device also has low distortion while driving a 600-W load for use in telecom systems.

This amplifier has a 360-mV input offset voltage, a 17-nV/Hz input noise voltage, and a 2-pA input bias current for measurement, medical, and industrial applications.

The device operates from a 2.5-V to 5.5-V single supply voltage. Its low power consumption makes it a good solution for portable applications.

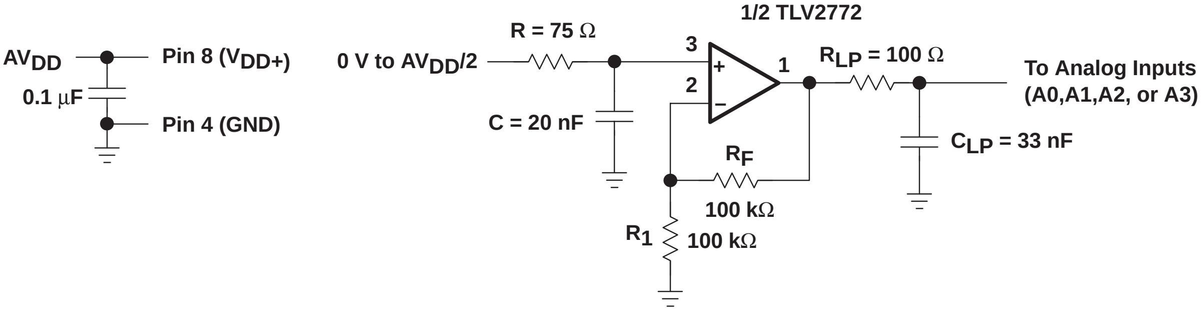


Figure 4. Schematic of a Noninverting Analog Input Buffer

In this configuration the TLV2772 works as a noninverting amplifier with a closed loop gain of two (as shown in Equation 3).

$$Gain = \frac{V_{OUT}}{V_{IN}} = 1 + \frac{R_F}{R_1} = 1 + \frac{10\ k\Omega}{10\ k\Omega} = 2 \qquad (3)$$

The input voltage range is therefore limited to between 0 V and $AV_{DD}/2$. The input low-pass filter consists of an RC circuit with a corner frequency of 106 kHz. While the filter is optional, it can be useful when operating in a noisy environment.

The low-pass filter at the op-amp output (RLP and CLP), has a corner frequency of

$$fc = \frac{1}{2\pi \times R_{LP} \times C_{LP}} = \frac{1}{2\pi \times 100\ \Omega \times 33\ nF} = 48\ kHz \qquad (4)$$

This filter limits the input signal bandwidth, and with it the operational amplifier inherent noise level which is fed into the ADC, thus improving the signal-to-noise ratio of the system significantly.

An additional 0.1-µF bypass capacitor, connected between V_{DD+} (pin 8) and GND (pin 4) of the device, ensures a noise-free supply for the operational amplifier.

Figure 5 shows another input buffer circuit that allows ±10-V input signals into the 5-V analog inputs of the ADC.

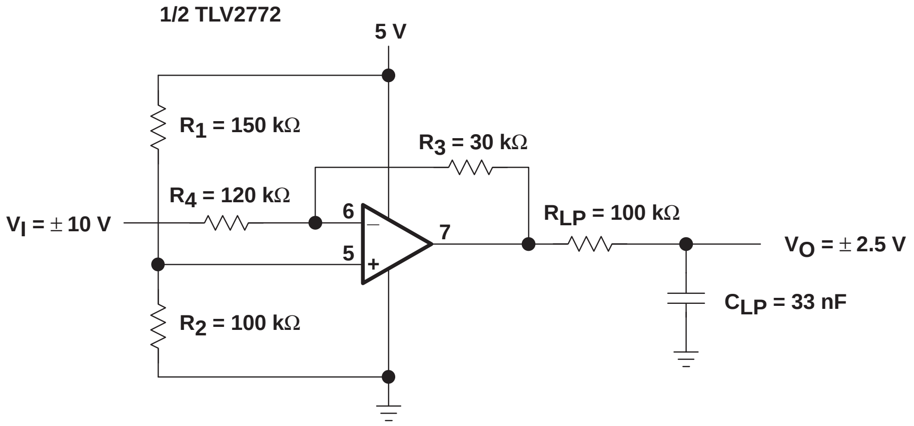


Figure 5. Schematic of an Inverting Input Buffer for Larger Input Signals

In this configuration the TLV2772 operates as an inverting amplifier with a closed loop gain of 0.25 (as shown in Equation 4).

$$Gain = \frac{V_{OUT}}{V_{IN}} = -\frac{R_3}{R_4} = -\frac{30\ k\Omega}{120\ k\Omega} = -0.25 \tag{5}$$

Resistors R_1 and R_2 bias the output of the operational amplifier to a 2.5-V operating point. A maximum input voltage of ±10-V is amplified by –0.25 and results in an output voltage of ±2.5 V. The negative sign in the gain factor represents a phase shift of 180° between input and output signal.

3 ADC Overview

The TLV1544 and TLV1548 are CMOS 10-bit switched capacitor successive approximation (SAR) ADCs. The TLV1544 has four analog inputs; the TLV1548 has eight. Figure 6 shows a functional block diagram of the devices.

The TLV1544 operates from a minimum supply of 2.7 V to a maximum supply of 5.5 V and allows high-speed data transfer from the host of up to 10 MHz maximum. In addition to the on-chip multiplexer that can select any one of the analog inputs or any one of the three internal self-test voltages, the ADC provides a versatile control capability. Through the DATA IN pin, 4-bit control words initialize the device for:

- Any one of the analog input channels
- Power-down mode
- Slow or fast conversion rate
- Any one of the self-test voltages.

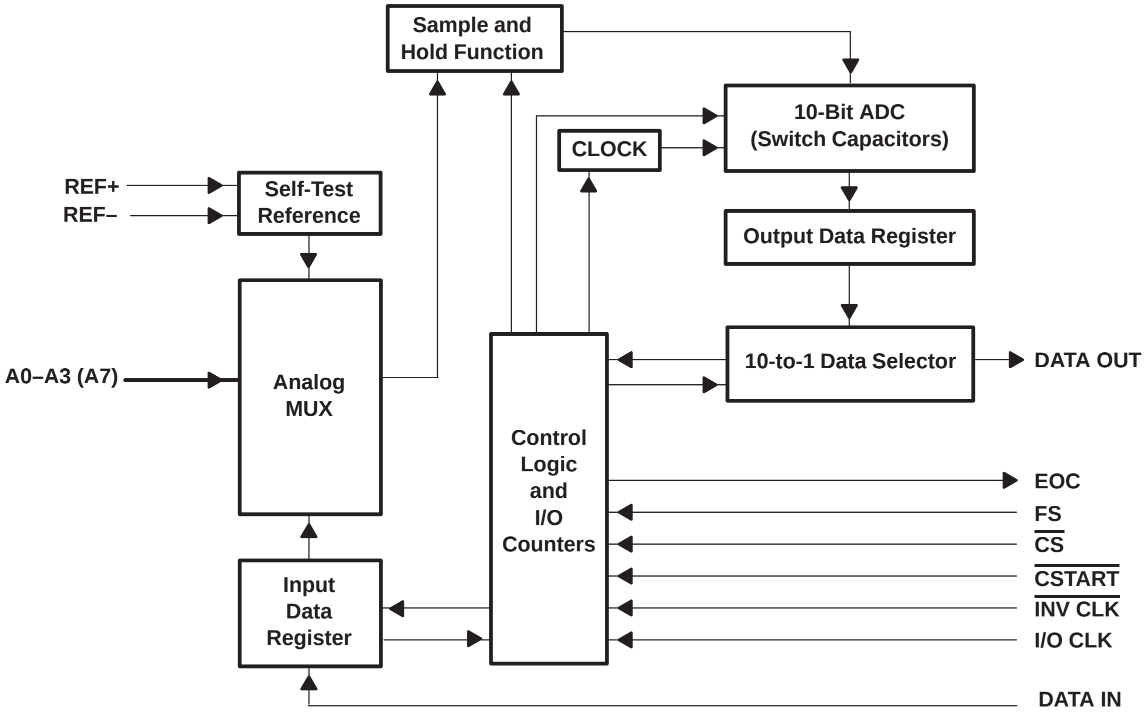


Figure 6. Functional Block Diagram of the TLV1544 and TLV1548

A 4-wire serial interface (SPI ™, QSPI ™) allows data transfer to a microprocessor or DSP. When interfacing to a TMS320 DSP, an additional frame sync (FS) signal indicates the start of a serial data frame. A high at chip-select pin $\overline{\text{CS}}$ activates the device. The data clock at I/O CLK determines the data rate between ADC and host. Through DATA IN, 4-bit control words initialize the ADC for the desired operation mode and select the analog input. The EOC pin indicates the end of a conversion and DATA OUT provides the conversion results in a 10-bit serial format.

SPI and QSPI are trademarks of Motorola, Inc.

A high at the $\overline{\text{INV CLK}}$ pin allows for I/O clock phase adjustment of 180°.When operating in the extended sampling mode, the $\overline{\text{CSTART}}$ pin controls the sampling period of the sample-and-hold circuit and starts the conversion. In this application, $\overline{\text{INV CLK}}$ and $\overline{\text{CSTART}}$ are not used and are therefore tied high.

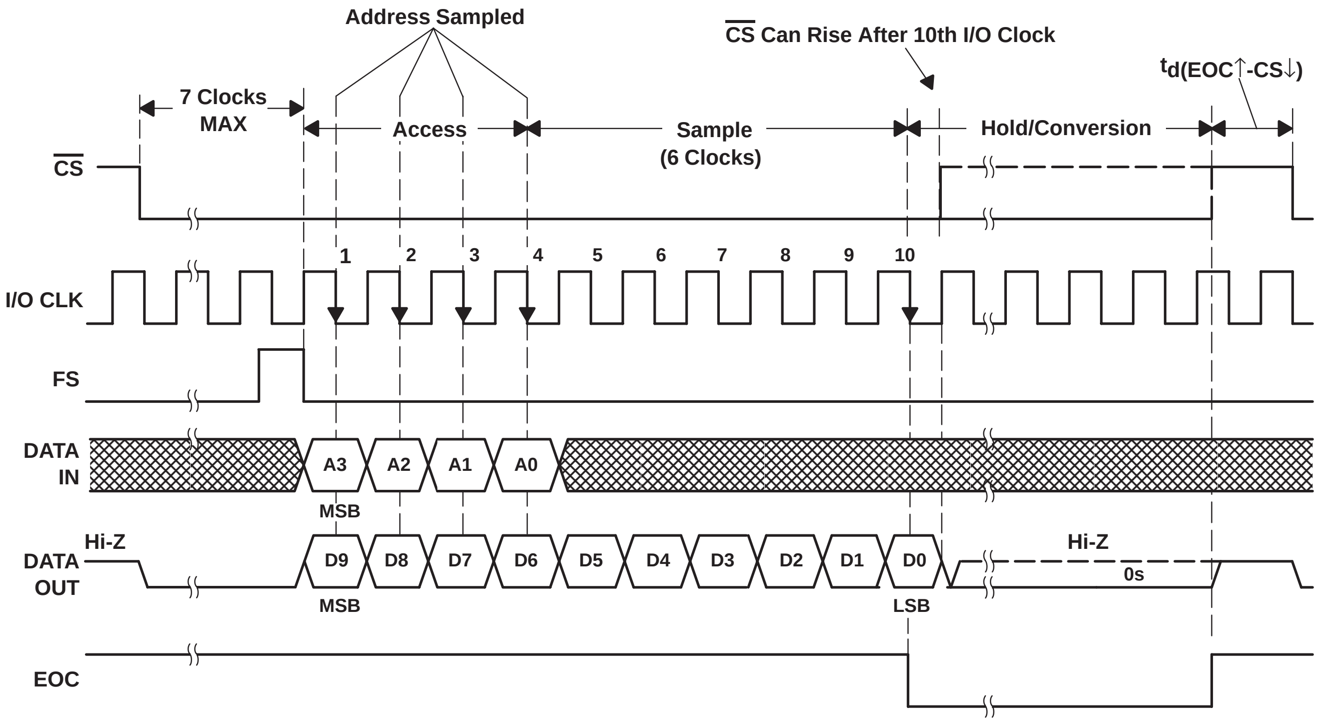
3.1 Signal Sequence

Figure 7 shows the timing diagram of a data transfer between the TLV1544 ADC and the TMS320C203 DSP through the DSP serial port. The ADC serves as a slave device and the DSP operates as the master, supplying the frame-sync signal, FS, and the data transfer clock, I/O CLK. Initially, with $\overline{\text{CS}}$ high and the ADC being inactive, the inputs, DATA IN and I/O CLK, are disabled. DATA OUT is in the high-impedance state and EOC is high.

When the DSP activates the ADC by taking $\overline{\text{CS}}$ low and providing the data clock to I/O CLK, the data transfer sequence begins. I/O CLK and DATA IN are enabled and DATA OUT is removed from the high-impedance state to logic low. The DSP then sends out an FS pulse on the FS line, indicating the start of a data frame.

With the falling edge of FS, the DSP provides the 4-bit control word to DATA IN (see Table 1 for ADC control words) starting with the most significant bit (MSB). At the same time, the ADC provides a 10-bit conversion result (from the previous conversion) at DATA OUT, beginning with MSB.

The input data selects a different mode or selects different analog input channels. In the case of the control word being a channel address, the selected analog input channel is accessed during the first four I/O clock cycles after the falling edge of FS. Starting with the falling edge of the fourth I/O cycle, the sample-and-hold (S&H) circuit samples the selected analog input.



NOTES: D. The falling edge of FS has to appear within 7 I/O clock cycles from the falling edge of $\overline{CS}$.

E. The TLV1544/48 data sheet specifies a minimum time of t_d = 100-ns after the rising edge of EOC before the next falling edge of $\overline{CS}$.

Figure 7. DSP Interface Timing (16-Clock Transfer, Normal Sample Mode, $\overline{\text{INV CLK}}$ = High)

At the tenth falling edge of I/O CLK, the sample is held and the analog-to-digital conversion starts. At the same time the EOC output changes from high to low, indicating the conversion start. The DSP initiates the end of a data transfer sequence by taking $\overline{CS}$ high, which disables DATA IN, I/O CLK and DATA OUT. This can happen between the end of the tenth I/O clock and the rising edge at EOC, indicating end-of-conversion.

If $\overline{CS}$ is taken high immediately after the tenth clock, DATA OUT goes into high-impedance state and, following the 10-bit conversion result, random signal levels are clocked into the DSP for the next six clock cycles. If $\overline{CS}$ changes to high somewhere between the end of the tenth clock and EOC going high, the 10-bit result is padded with a maximum of six zeros to complement the 16-clock cycle of the DSP.

In any case, the six bits following the conversion result are useless information and should be ignored by the DSP interface software. The entire data transfer sequence is complete when EOC returns to high.

A delay time of $t_{d(EOC\uparrow - CS\downarrow)}$ = 100 ns after the rising edge of EOC is required before $\overline{CS}$ can change to low again to start a new data transfer.

Table 1. TLV1544/TLV1548 ADC Control Words

FUNCTION SELECT	INPUT DATA BYTE		COMMENT
	A3 – A0		
	BINARY	HEX	
Analog channel A0 for TLV1548 Selected	0000b	0h	Analog channel 0 for TLV1544
Analog channel A1 for TLV1548 Selected	0001b	1h	
Analog channel A2 for TLV1548 Selected	0010b	2h	Analog channel 1 for TLV1544
Analog channel A3 for TLV1548 Selected	0011b	3h	
Analog channel A4 for TLV1548 Selected	0100b	4h	Analog channel 2 for TLV1544
Analog channel A5 for TLV1548 Selected	0101b	5h	
Analog channel A6 for TLV1548 Selected	0110b	6h	Analog channel 3 for TLV1544
Analog channel A7 for TLV1548 Selected	0111b	7h	
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

4 The TMS320C203 DSP

The TMS320C203 DSP is a 16-bit fixed-point, static CMOS digital signal processor. The combination of an advanced Harvard architecture (separate buses for program memory and data memory), on-chip peripherals, on-chip memory, and a highly specialized instruction set is the basis of the operational flexibility of this device.

The synchronous and asynchronous serial ports are the two on-chip peripherals used mainly in this application. Their block diagrams and operation are explained in more detail in the following sections.

4.1 The DSP Serial Port

The serial port provides communication with serial devices such as codecs and serial ADC converters. The synchronous serial port offers these features:

- Two four-word-deep FIFO buffers
- Interrupts generated by the FIFO buffers
- A wide range of speeds of operation
- Burst and continuous modes of operation

4.1.1 Signals and Registers

The serial port consists of the following six basic signals:

CLKX *Transmit clock input or output*. This signal clocks data from the transmit shift register (XSR) to the DX pin. The serial port can be configured for either generating an internal clock, or accepting an external clock.

If the port is configured for generating an internal clock, CLKX becomes an output, transmitting a maximum frequency equal to one fourth of the CPU clock. If the port is configured to accept an external clock, CLKX changes to an input, receiving the external clock signal.

FSX *Transmit frame synchronization*. FSX indicates the start of a transmission. If the port is configured for generating an internal frame sync pulse, the FSX pin transmits the pulse.

If the port is configured for accepting an external frame sync pulse, this pin receives the pulse.

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

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

FSR *Receive frame synchronization*. FSR initiates the reception of data at the beginning of the packet.

DR *Serial data receive*. DR receives serial data, transferring it into the receive shift register (RSR).

The synchronous serial port (SSPCR) has two four-level transmit and receive FIFO buffers shown in Figure 8.

Two on-chip registers allow access to the FIFO buffers and control the operation of the port:

Synchronous data transmit and receive register (SDTR). The SDTR, at I/O address FFF0h, is used for the top of both FIFO buffers (transmit and receive) and is the only visible part of the FIFO buffers.

Synchronous serial port control register (SSPCR). The SSPCR, at I/O address FFF1h, contains bits for setting port modes, indicating the status of a data transfer, setting trigger conditions for interrupts, indicating error conditions, accepting bit input, and resetting the port.

4.1.2 Serial Port Operation

Figure 8 shows how the pins and registers are configured on the serial port and how the FIFO buffering is implemented.

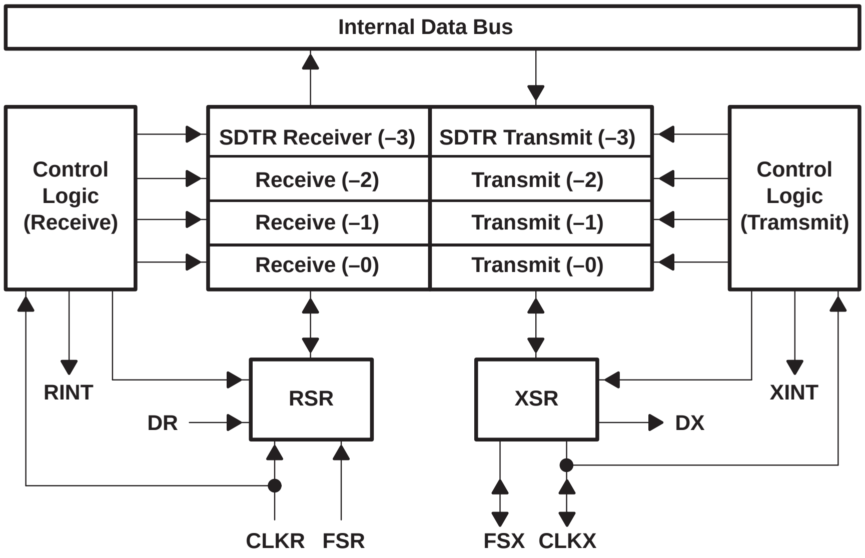


Figure 8. Synchronous Serial Port Block Diagram

4.1.3 Synchronous Serial Port Configuration

While the data registers are mainly responsible for shifting and buffering data, the SSPCR configures the entire serial port and is therefore the most important register used in the interface programs. Figure 9 shows the 16-bit memory-mapped SSPCR. Some of the bits are read-only while others are read/write.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE	SOFT	TCOMP	RFNE	FT1	FT0	FR1	FR0	OVF	IN0	XRST	RRST	TXM	MCM	FSM	DLB
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Figure 9. Serial Port Control Register

Table 2 gives an overview of the bit definition of the SPC. Only the bits which are important to this application are explained. For detailed information on the bit functions refer to the TMS320C2xx User’s Guide.

Table 2. Serial Port Control Register Bits Summary

BIT	NAME	FUNCTION
0	DLB	The digital loop-back mode bit allows to connect the transmitter output (DX and FSX) to the receiver input (DR and FSR). *(This bit is not used in the application and set to zero.)*
1	FSM	The frame synch mode bit specifies when a frame sync pulse is needed. FSM = 1, burst mode is selected (an FS-pulse is used for each word) FSM = 0, continuous mode is selected (only one start pulse is required)
2	MCM	The clock mode bit specifies the clock source for CLKX. MCM = 1, on-chip clock source is used with CLKX = 1/4 CLKOUT1 MCM = 0, external clock source is chosen
3	TXM	The transmit mode bit specifies the source for FSX-pulse generation TXM = 1, FSX is generated on–chip and synchronized to CLKX TXM = 0, FSX needs to be applied from extern
4 5	RRST XRST	The transmit and receive reset signals activate and deactivate the transmitter and receiver of the serial port. XRST/ RRST = 1, transmitter and receiver are active XRST/ RRST = 0, activity halts
6	IN0	The input bit reflect the levels of the CLKR pins. IN0 is read only bits.

7	OVF	OVF = 1, receive FIFO buffer is full.
		OVF = 0, receive FIFO buffer is read.

8 9	FR0 FR1	FR0/FR1 = 0/0, generate RINT when receive FIFO buffer is not empty. FR0/FR1 = 0/1, generate RINT when receive FIFO buffer holds at least two words. FR0/FR1 = 1/0, generate RINT when receive FIFO buffer holds at least three words. FR0/FR1 = 1/1, generate RINT when receive FIFO buffer is full (holds four words).
10 11	FT0 FT1	FT0/FT1 = 0/0, generate XINT when transmit FIFO buffer can accept one or more words. FT0/FT1 = 0/1, generate XINT when transmit FIFO buffer can accept two or more words. FT0/FT1 = 1/0, generate XINT when transmit FIFO buffer can accept three or four words. FT0/FT1 = 1/1, generate XINT when transmit FIFO buffer is empty (can accept four words).

12	RFNE	RFNE = 1, transmit FIFO is not empty.
		RFNE = 0, transmit FIFO is empty.

13	TCOMP	TCOMP = 1, transmit FIFO is not empty.
		TCOMP = 0, transmit FIFO is empty.

14	SOFT	The SOFT bit is an emulation bit that aborts transmission when a breakpoint is encountered in the high-level language debugger. It is enabled when the FREE bit is 0. SOFT = 1, stop after word completion SOFT = 0, immediate stop *(This bit is effective only in emulation mode, otherwise it is set to zero.)*

15	FREE	The FREE bit selects the free run of CLKX.
		FREE = 1, CLKX runs free
		FREE = 0, STOP bit is enabled
		(This bit is effective only in emulation mode, otherwise it is set to zero.)

When interfacing to the TLV1544/48 ADC, the serial port of the TMS320C203 DSP must be configured as shown in Table 3.

Table 3. SSPCR Configuration for Interfacing to the TLV1544/TLV1548 ADC

REQUIRED CONFIGURATION	AFFECTED BITS
The DSP needs to be set-up as the master device, generating the necessary frame sync pulse to start a data transfer.	TXM = 1
Because of serial port inactivity during the conversion process of the ADC, the serial port must operate in burst mode.	FSM = 1
The DSP on-chip clock source is selected to provide CLKX as the data transfer clock to the I/O CLK input of the ADC.	MCM = 1
When the SSPCR is to be modified to reconfigure the serial port, transmitter and receiver need to be reset. After the modification is complete, transmitter and receiver need to be activated.	XRST/RRST = 0 XRST/RRST = 1

When reconfiguring the SPPCR, two instructions are required. The first instruction resets the transmitter and receiver and configures the SSPCR. The second instruction reactivates the transmitter and receiver. Figure 10 shows the binary format and the hex code of the assembler instructions used to configure and to activate the serial port.

		FREE	SOFT	TCOMP	RFNE	FT1	FT0	FR1	FR0	OVF	IN0	XRST	RRST	TXM	MCM	FSM	DLB
Configure Port	SSPCR = #000Eh	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
		0				0				0				E			
Activate Port	SSPCR = #003Eh	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
		0				0				3				E			

Figure 10. Configuring and Activating the Serial Port for 5-V Operation of the ADC

4.1.4 Transmit and Receive Operations in Burst Mode

In burst mode operation, there is a period of serial port inactivity between packet transmits. Therefore each data packet needs to be marked by a frame sync pulse.

In the transmit direction, after a write to SDTR, a frame sync pulse (at FSX) is generated on the next rising edge of CLKX. The Frame Sync pulse remains high for one CLKX cycle. A transmit interrupt, XINT, is generating on the falling edge of CLKX. On the next rising edge of CLKX after FSX goes high, XSR is loaded with the value from the FIFO buffer. On the next rising edge of the CLKX cycle, the first data bit (MSB first) is driven on the DX-pin. The remaining bits are shifted out at each consecutive rising edge of CLKX.

In the receive direction, the shifting into RSR begins on the falling edge of the CLKX cycle after the frame sync has gone low. After all bits have been received, the content of the RSR is transferred to the SDTR on the falling edge of CLKX. and a receive interrupt, RINT, is generated.

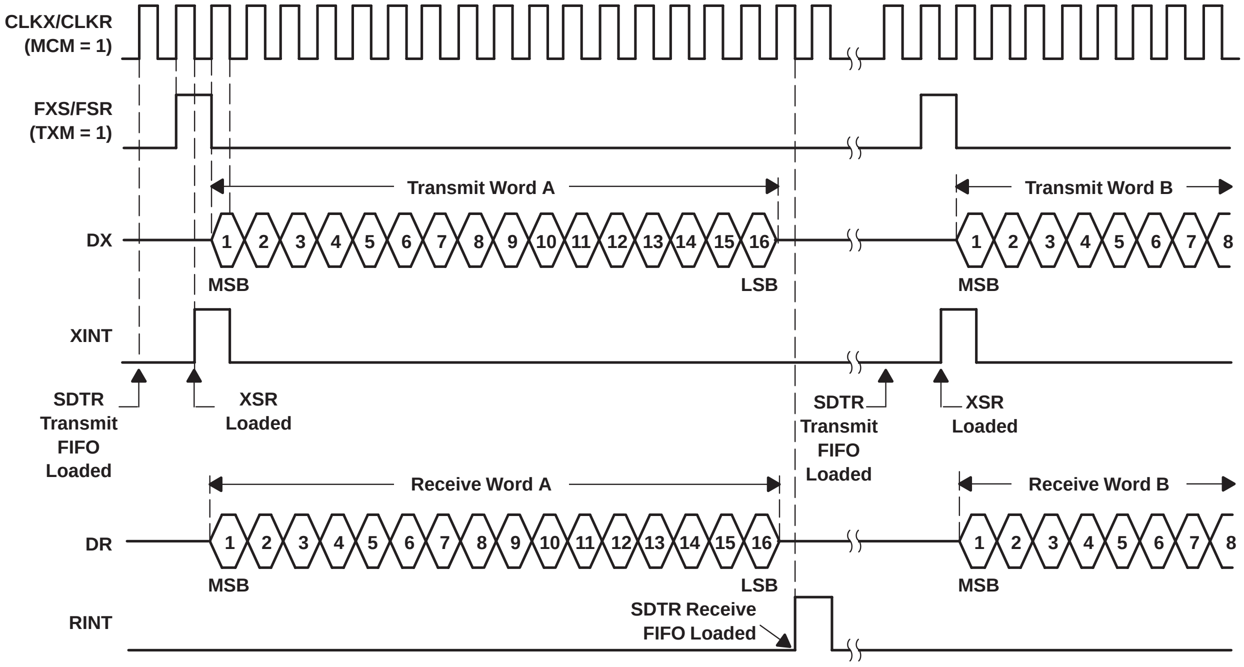


Figure 11. Transmit- and Receive-Operation in Burst Mode

4.2 The DSP Asynchronous Serial Port

The asynchronous serial port can be used to transfer data to and from other devices.

The asynchronous serial port offers these features:

- Full-duplex transmit and receive operations at the maximum transfer rate
- Data-word length of eight bits for both transmit and receive
- Double buffering in all modes to transmit and receive data
- Adjustable baud rate of up to 250,000 10-bit characters per second
- Automatic baud-rate detection logic

4.2.1 Interface Pins and Registers

The asynchronous serial port consists of the following interface pins:

TX TX transmits serial data from the asynchronous serial port transmit shift register (AXSR).

RX RX transmits serial data into the asynchronous serial port receive shift register (ARSR).

IO0–IO3 General-purpose I/O pin 0–3. Can be used for general purpose I/O or for handshaking by the UART.

Two on-chip registers transmit and receive data and control the operation of the port:

Asynchronous serial port control register (ASPCR). The ASPCR, at I/O address FFF5h, contains bits for setting port modes, enabling or disabling the automatic baud-rate detection logic, selecting the number of stop bits, enabling or disabling interrupts, configuring pins IO3–IO0, and resetting the port.

I/O status register (IOSR). The IOSR, at I/O address FFF6h, contains bits for detection of the incoming baud rate, various error conditions, the status of data transfers, detection of a break on the RX pin, the status of pins IO3–IO0, and detection of changes on pins IO3–IO0.

4.2.2 Asynchronous Serial Port Configuration

The ASPCR controls the operation of the asynchronous serial port. Figure 12 shows the 16-bit memory-mapped ASPCR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE	SOFT	URST				DIM	TIM	RIM	STB	CAD	SETBRK	CIO3	CIO2	CIO1	CIO0
R/W	R/W	R/W	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 12. Aynchronous Serial Port Control Register

Table 4 gives an overview of the bit definition of the ASPCR. Only the bits important to this application are explained. For detailed information on the bit functions refer to the *TMS320C2xx User's Guide.*

Table 4. Asynchronous Serial Port Control Register Bits Summary

BIT	NAME	FUNCTION
0 – 3	CIO0–CIO3	CIOn = 0, IOn is configured as an input. CIOn = 1, IOn is configured as an output. n = 0, 1, 2 3
4	SETBRK	SETBRK = 0, The TX output is forced high when the port is not transmitting. SETBRK = 1, The TX output is forced low when the port is not transmitting.

5	CAD	CAD = 0, Disables auto-baud alignment.
		CAD = 1, Enables auto-baud alignment.

6	STB	STB = 0, One stop bit is used in transmission and reception. STB = 1, Two stop bits are used in transmission and reception.

7	RIM	RIM = 0, Disables receive interrupts.
		RIM = 1, Enables receive interrupts.

8	TIM	TIM = 0, Disables transmit interrupts.
		TIM = 1, Enables transmit interrupts.

9	DIM	DIM = 0, Disables delta interrupts.
		DIM = 1, Enables delta interrupts.

10–12	Reserved	Always read as 0s.
13	URST	URST = 0, The port is in reset. URST = 1, The port is enabled.
14	SOFT	The SOFT bit is an emulation bit that aborts transmission when a breakpoint is encountered in the high-level language debugger. It is enabled when the FREE bit is 0. SOFT = 1, stop after word completion SOFT = 0, immediate stop *(This bit is effective only in emulation mode, otherwise it is set to zero.)*

15	FREE	The FREE bit selects the free run of CLKX.
		FREE = 1, CLKX runs free
		FREE = 0, STOP bit is enabled
		(This bit is effective only in emulation mode, otherwise it is set to zero.)

When interfacing to the TLV1544/48 ADC, the serial port of the TMS320C203 DSP must be configured as shown in Table 5.

Table 5. ASPCR Configuration for Interfacing to the TLV1544/TLV1548 ADC

REQUIRED CONFIGURATION	AFFECTED BITS
The DSP set IO0-IO3 as input.	IO0-IO3 = 0

The Delta Interrupt Mask is enabled when the EOC signal of TLV1544/48 is low.
The Delta Interrupt mask is disabled when the EOC signal of TLV1544/48 is high.
	DIM = 1, enable
	DIM = 0, disable

When the ASPCR is to be modified to reconfigure the serial port, transmitter and receiver need to be reset.
After the modification is complete, transmitter and receiver need to be activated.
	URST = 0
	URST = 1

When reconfiguring the ASPCR, two instructions are required. The first instruction resets the transmitter and receiver and configures the ASPCR. The second instruction reactivates the transmitter and receiver. Figure 13 shows the binary format and the hex code of the assembler instructions used to configure and to activate the serial port.

		FREE	SOFT	URST				DIM	TO,	RIM	STB	CAD	SETBRK	CIO3	CIO2	CIO1	CIO0
Configure Port	SSPCR = #0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0				0				0				8			
Activate Port	SSPCR = #2000h	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		2				0				0				0			

Figure 13. Configuring and Activating the Asynchronous Serial Port

4.2.3 I/O Status Register (IOSR) Configuration

The IOSR returns the status of the asynchronous serial port and of I/O pins IO0–IO3. Figure 14 shows the 16-bit I/O address FF6h, memory-mapped IOSR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ADC	BI	TEMT	THRE	FE	OE	DR	DIO3	DIO2	DIO1	DIO0	IO3	IO2	IO1	IO0
0	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 14. I/O Status Register (IOSR)

When interfacing to the TLV1544/48 ADC, use the input pin IO3 of the TMS320C203 DSP to connect to the EOC pin of the ADC. When the logic level changes in EOC, the DIO3 bit is set to 1, then the delta interrupt is generated. The DIO3 bit must be cleared before leaving the Interrupt Service Routine (ISR).

Figure 15 shows the delta interrupt logic operation diagram:

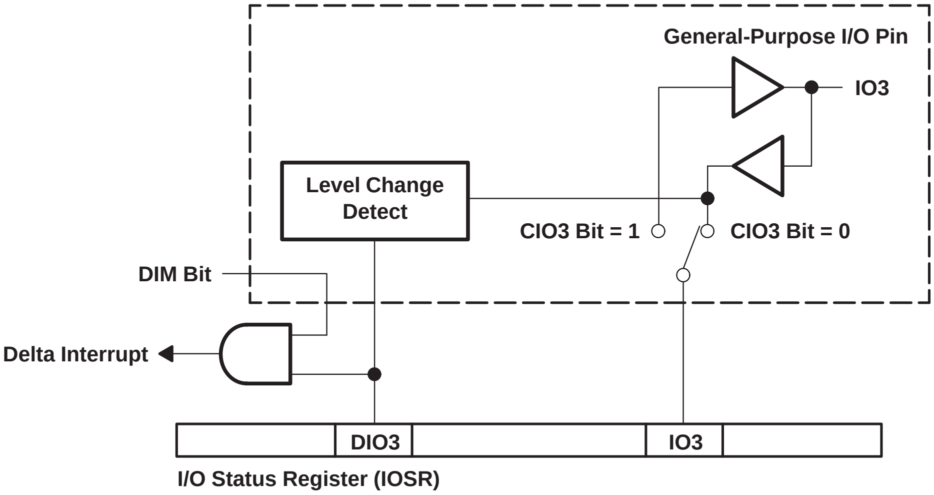


Figure 15. Delta Interrupt Logic Operation

5 Software Description

The interface software consists of one C-callable assembler program and one corresponding C-program. The user can specify certain data transfer parameters —such as the analog input channel, the conversion mode, the memory start address, and the number of samples—through the global variables of the .C file without modifying the corresponding assembler program.

When a .C interface routine is called, the global variables in the C file are loaded into the local variables of the assembler program.

Table 6 shows the global and local variables.

Table 6. Local and Global Variables and Corresponding Programs

PROGRAM	TYPE	VARIABLE	DESCRIPTION
All C–files	Global	_Samples	User defined number of samples to be acquired
	Global	_MemStart	User defined memory pointer start
	Global	_Channel	User defined channel number to be sampled
	Global	_ControlMode	User defined conversion mode
All Assembler files	Local	ADWORD	ADC control word
	Local	ADCOUNT	Sample counter
	Local	ADMEM	Memory pointer
	Local	RINT_COUNT	Number of receive interrupts
	Local	END_BIT	End-of-program bit
	Local	TEMP	Temporary register
	Local	TEMP1	Temporary register

The assembler programs execute the following steps:

1. Initialize the DSP, the synchronous serial port, and the asynchronous serial port
2. Load the user defined values
3. Activate and initialize the ADC
4. Acquire the specified number of data
5. Disable the ADC
6. Return to the C program

The above sequence shows that the assembler routines are used solely to acquire data. It also shows that every time the interface routine is called, the ADC is enabled for the data acquisition process and is disabled before the routine returns to the C program. Therefore, neither of the self-test modes nor the power-down instruction, which are available in the list of TLV1544 control words, are used in these routines.

The following sections explain the main assembler program in detail. To review the individual file listings, refer to Appendix A.

5.1 Program-1 *(Filename: C1544IO3.asm ⇒ IO3 as Interrupt-Source)*

This program supports the glueless DSP-to-ADC interface in Figure 16. In this application the ADC operates from a 5-V supply and can be selected for fast or slow conversion mode. The transfer clock at I/OCLK connected to the DSP at CLKX is 10-MHz (1/2 of the CPU clock).

The EOC signal of the ADC is fed into the external interrupt input, IO3, of the DSP.

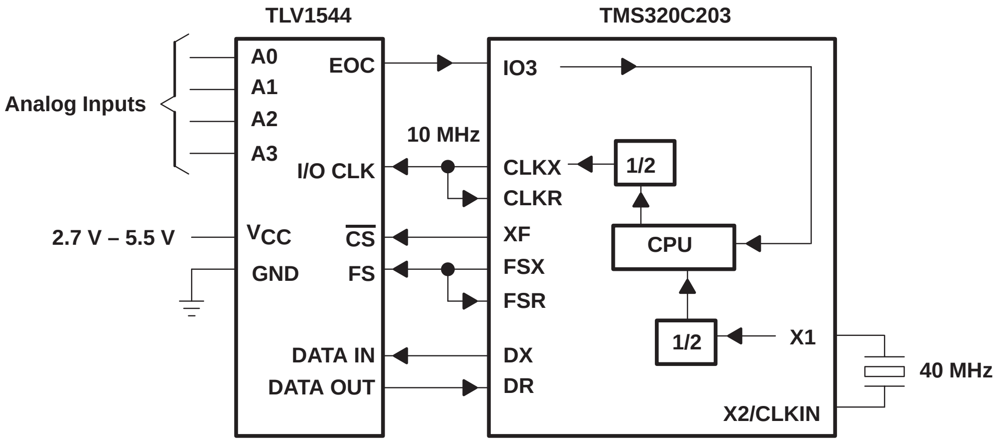


Figure 16. External Interrupt, IO3, Detects EOC via External Inverter

The timing diagram in Figure 17 divides a data transfer sequence into four steps:

1. The DSP activates the ADC by taking $\overline{\text{CS}}$ low. Then the control word to initialize the ADC is loaded into the DSP serial port.
2. The actual data transfer happens. While the DSP sends control data to configure the ADC operation mode, the ADC transmits conversion results to the serial port receiver.
3. On the 16th clock cycle of I/O CLK (CLKX) after FS has gone low, a receive interrupt, RINT, is generated. The following interrupt service routine, RINT-ISR, disables the ADC, stores the received data into memory, and returns from interrupt to idle mode.
4. The CPU idles for the remaining ADC conversion time..

At the end of a conversion, the EOC signal of the ADC causes an external interrupt through the DSP IO3 input. The following interrupt service routine, IO3–ISR, enables the ADC for a new data transfer and loads the latest control word into the serial port.

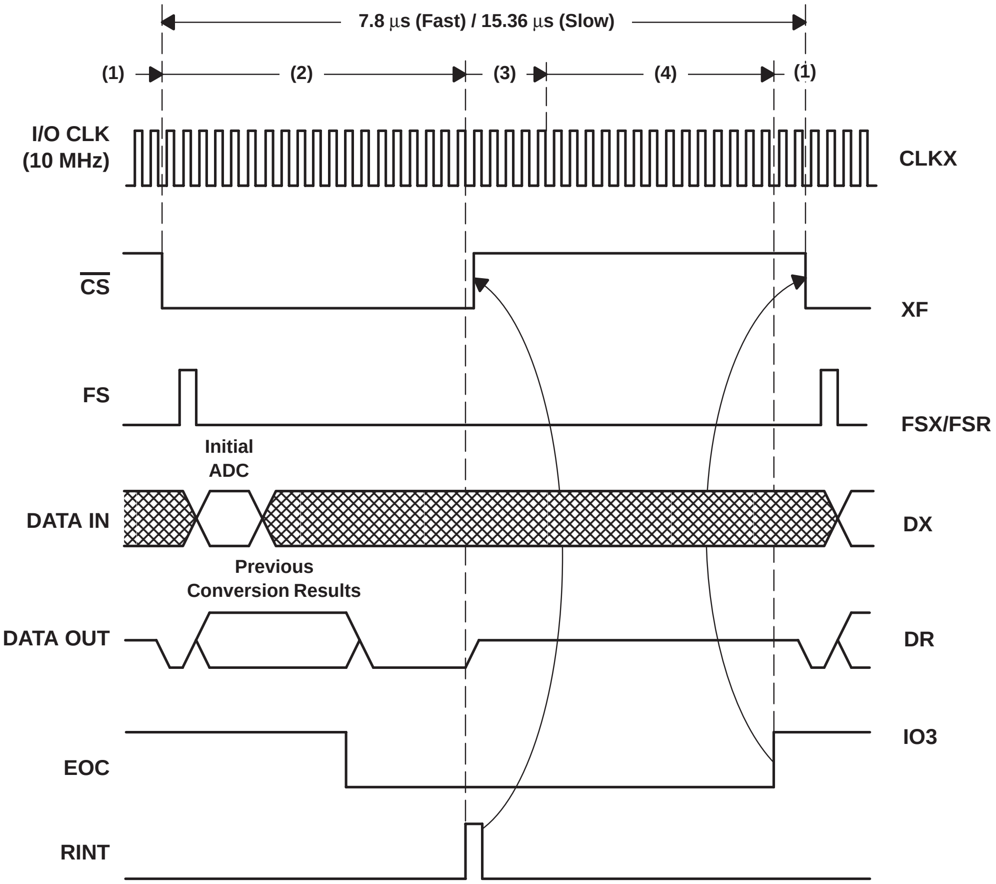


Figure 17. Data Transfer Sequence

5.1.1 Assembler Program Description

Using the flowchart in Figure 18, this section explains the assembler program in detail.

> **NOTE:** Each task box in the flowchart appears as a header in the assembler program listing in Appendix A.

TLV1544START

The program starts calling the main routine, _C1544IO3, from the C program. All previously used pointers and registers are saved. These include the following registers:

- The frame and the stack pointer, FP and SP
- The status registers, ST0 and ST1
- The auxiliary registers, AR6 and AR7
- The wait-state registers, WSGR

During the DSP initialization all interrupts are disabled and the wait states are set to zero. Then the serial port is configured for burst-mode operation. The FS signal and the CLKX are programmed to be generated on-chip. Finally the transmitter and receiver stages are activated.

Next, initialization of the asynchronous serial port configures IO0–IO3 as input pins.

Now, the user-defined values (global variables in the C program) are loaded into the local variables of the assembler routine. The channel number is stored into TEMP1. The memory start address is loaded into ADMEM, and the number of samples to be acquired is saved into ADCOUNT. The control mode, which can be fast or slow conversion, is stored into ADWORD.

Then both interrupts, RINT and TXRXINT, are enabled and the ADC is initialized.

Before the initialization of the ADC begins, the variable RINT_COUNT is set to 0. RINT_COUNT specifies the number of receive interrupts that must occur before the ADC can provide valid conversion results.

The local variable, END_BIT, which defines the end of the entire program, is set to 1. The general output port (XF), of the DSP is driven low to enable the ADC through the chip-select pin, $\overline{\text{CS}}$.

The variable ADWORD, which contains the conversion mode, is then copied into the serial port data transmit register (DXR), and sent to the ADC. Subsequently the content of TEMP1, which specifies the channel number, is loaded into ADWORD.

The CPU then resides in idle mode and waits for a receive interrupt (RINT), to occur.

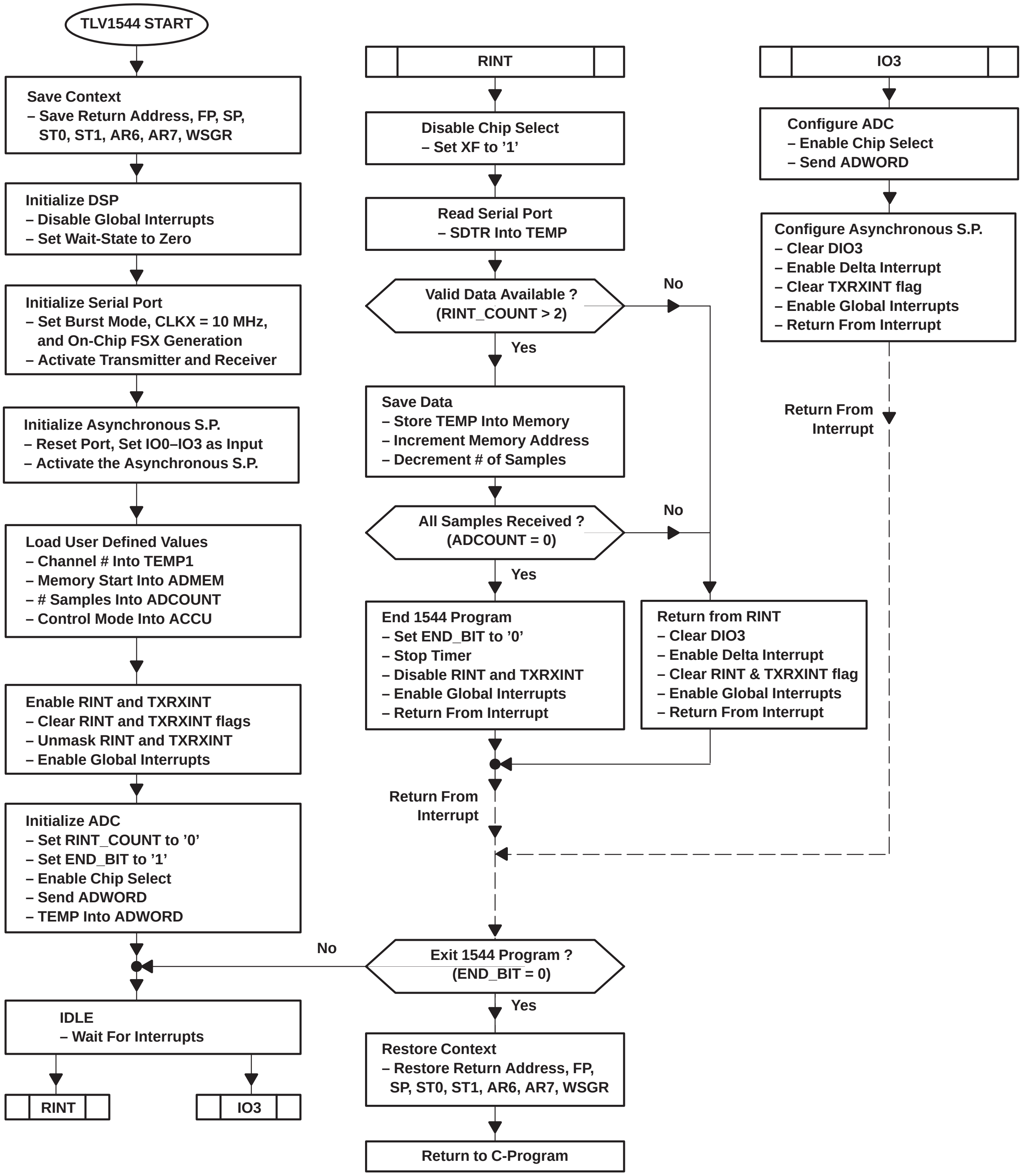


Figure 18. Flowchart of C1544IO3.asm

RINT (Receive Interrupt Routine)

With the ADC initialized into fast-conversion mode, the DSP starts the first data transfer of control data in a 16-bit data frame. Therefore, upon the 16th clock cycle of CLKX, RINT is generated that forces the CPU to execute the RINT service routine (RINT-ISR). At the beginning of the RINT routine, the XF output of the DSP is taken high, disabling the ADC. After that, the content of the data receive FIFO in SDTR is stored into the temporary register, TEMP.

The following decision box investigates the receive data for validity by checking the content of RINT_COUNT for a value higher than two. Since it takes two data transfers to configure the ADC completely, one to specify the conversion mode and a second one to select the analog input channel, the conversion results from the ADC are invalid for both transfers. With each data transfer generating a receive interrupt, the number of occurring RINTs (which is stored in RINT_COUNT) must be three to indicate that valid data are available.

If no valid data are available, the CPU leaves the RINT-ISR to configure the ADC channel number through the IO3 interrupt service routine, IO3-ISR.

If valid data are available, the latest receive data, stored in TEMP, are saved into memory. Subsequently the memory address is increased by incrementing the content of ADMEM, and the number of samples is decreased by decrementing the content of ADCOUNT.

A second decision box checks whether all samples have been received. If all samples were received, the END_BIT is set to zero and both interrupts, RINT and TINT, are disabled. The program leaves the RINT-ISR and, through the EXIT routine, returns to the C program.

If more samples need to be acquired, the program clears the DIO3 bit, enables Delta interrupt, clears the RINT and TXRXINT flags, and returns from RINT into Idle-mode, where it resides until the external interrupt, IO3 occurs.

IO3 (External Interrupt Routine)

The IO3-ISR loads the serial port with transmit data and initiates a new data transfer. Once, the EOC output of the ADC changes from low to high, an external interrupt is generated that causes the CPU to enter the IO3 service routine. Immediately the DSP initiates a new data transfer by enabling the ADC through the XF pin. Afterwards the channel number, stored in ADWORD, is sent to the ADC as the new control word. Then the program clear DIO3 bit, disables Delta interrupt, and clears TXRXINT flag. The CPU returns from interrupt into idle mode and waits until the next receive interrupt, RINT, occurs.

Exit 1544 program ?

As long as END-BIT is set to one, the CPU diverts to the idle mode to continue acquiring data. Once END_BIT has been set to zero, all previously saved registers in the Save-Context box are restored. The CPU now exits the interface routine to return to the C-Program

6 Summary

This report discusses all components of a low-cost, low-power data acquisition system as shown in Figure 19. The low-dropout voltage regulator, TPS7101, is programmable through an external voltage divider to provide the required supply voltage of 5 V.

The low-noise, dual operational amplifier, TLV2772, builds the signal conditioning interface to the TLV1544 ADC. One stage works as a noninverting amplifier with a closed-loop gain of two to interface small input signals between 0 and 2.5 V to the ADC. The second stage operates as an inverting amplifier with a gain of 0.25, allowing large input signals of ±10 V to be interfaced to the ADC.

The TLV1544 is a 10-bit, ADC with four analog inputs. The device interfaces easily with the TMS320C203 DSP to build a simple data acquisition system used in battery powered applications as well as in industrial control systems.

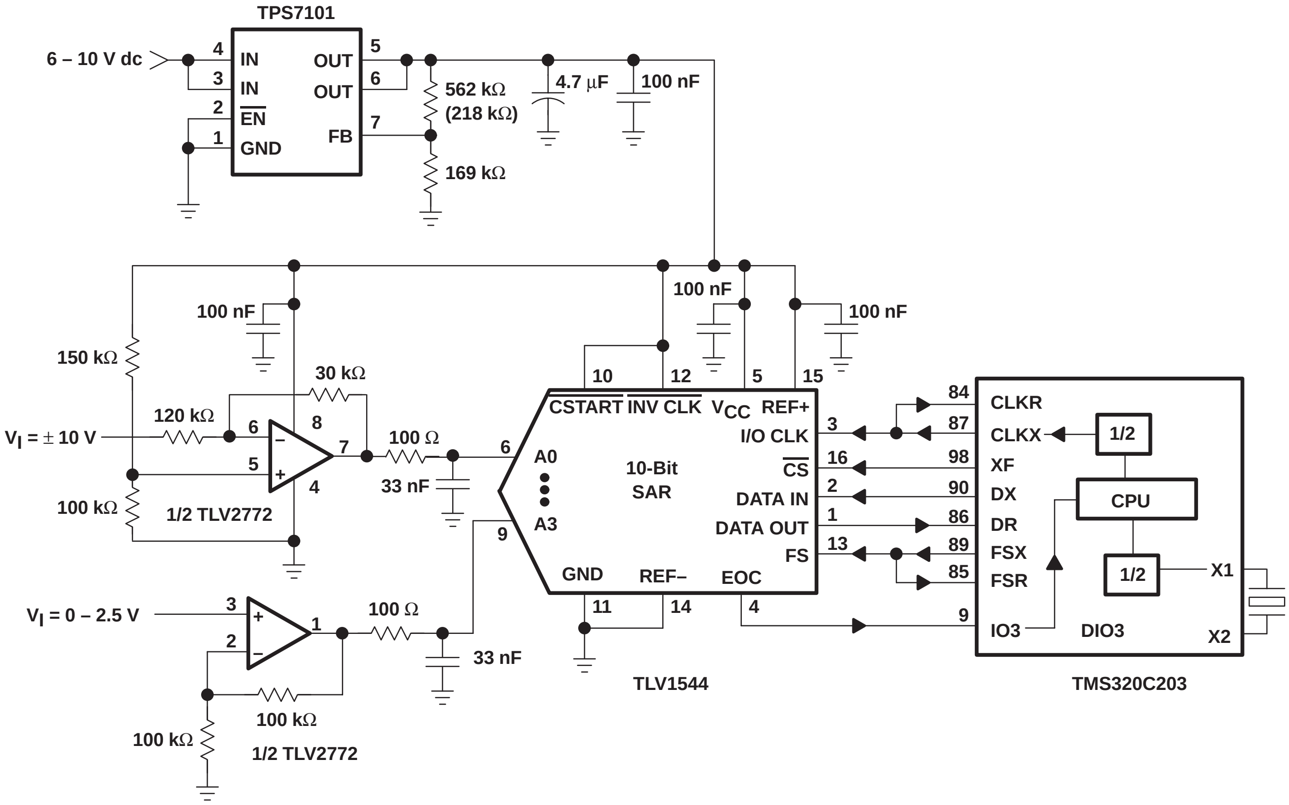


Figure 19. Interfacing the TLV1544 ADC with the TMS320C50 DSP

The report includes a C-callable interface program, which supports the data transfer between ADC and DSP.

7 References

For additional information and 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
- *TLV1544 EVM Manual*, literature number SLAU014
- *TLV2432 Data Sheet*, literature number SLOS168B
- *TPS7101 Data Sheet*, literature number SLVS092F
- *SN74AHC1G04 Data Sheet*, literature number SCLS318G
- *Switched-Capacitor Analog Input Calculations Application Report*, literature number SLAA036
- *Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP*, literature number SLAA025A.

Appendix A TLV1544 Program Files: IO3 as Interrupt Source

A.1 Boot Routine: BOOTIO3.ASM

```
************************************************************************
* TITLE         : TLV1544 ADC boot routine (IO3 as interrupt source)  *
* FILE          : BOOTIO3.ASM                                         *
* DESCRIPTION   : Boot routine to initialize the DSP and run the      *
*                 c–program ’main()’ in the C1544T.C file.            *
* AUTHOR        : AAP Application Group, Dallas                       *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.  *
* REFERENCE     : TMS320C2xx User’s Guide, TI 1997                    *
*               : Data Acquisition Circuits, TI 1998                  *
************************************************************************
   .mmregs
************************************************************************
* Declare the stack.
************************************************************************
__stack:        .usect  ”.stack”,0
       .def   _c_int0
       .ref   _main
       .text
_c_int0:
************************************************************************
* INITIALIZATION BODY                                                 *
************************************************************************
*
* DSP INITIALIZATION
       SETC    INTM            ; disable global interrupts
       LDP     #0              ; load data page 0
************************************************************************
* SET UP INITIAL STACK AND FRAME POINTERS                             *
************************************************************************
      LRLK    AR0,__stack      ; set up frame pointer for main routine
      LRLK    AR1,__stack      ; set up stack pointer for main routine
      MAR     *,AR1            ; pointer select to AR1
      B       _main            ; jump to C program at main
```

A.2 C-Program: C1544I.C

```
/* TITLE:       TLV1544 ADC C program main routine          */
/*              (IO3 as interrupt source)                   */
/* File:        C1544I.C                                    */
/* Description: In this c–program file the user selects     */
/*              the input channel, the conversion mode, the*/
/*              memory start address, and the number        */
/*              of samples.This c–program then calls the    */
/*              C1544IO3() interface program to execute it.*/
/* ––––––––––––––––––––––––––––––––––––––––––––––––––––––– */
/* Programmed Operation Mode:                               */
/*                                                          */
/* Channel:                   Value:                        */
/* Select TLV1544 channel 0      0x0000                     */
/* Select TLV1544 channel 1      0x2000                     */
/* Select TLV1544 channel 2      0x4000                     */
/* Select TLV1544 channel 3      0x6000                     */
/* (Vreg+ – Vreg–)/2             0x0B000                    */
/* Vreg–                         0x0C000                    */
/* Vreg+                         0x0D000                    */
/*                                                          */
/* ControlMode:                                             */
/* Software Power Down           0x08000                    */
/* Fast Conversion Rate          0x09000                    */
/* Slow Conversion Rate          0x0A000                    */
/*––––––––––––––––––––––––––––––––––––––––––––––––––––––––– */
extern ControlMode, Channel, Samples, MemStart;
extern void C1544IO3(void);
main()
{
/* Fast Conversion Mode                                     */
   ControlMode = 0x09000;                                   */
/* Select channel 1                                         */
   Channel = 0x2000;

/* Take 256 samples                                         */
   Samples = 0x100;

/* Memory Start Address                                     */
   MemStart = 0x1000;
/* Call TLV1544 Interface Program                           */
   C1544IO3();
}
```

A.3 C-Callable Interface Program: C1544IO3.ASM

```
* TITLE         : TLV1544 ADC C–Callable Interface routine            *
*                 (IO3 as interrupt source)                         *
* FILE          : C1544IO3.ASM                                        *
* FUNCTION      : _C1544IO3                                           *
* DESCRIPTION   : Main routine to transfer data between the C203 DSP  *
*                 and TLV1544 ADC via the DSP serial interface. At    *
*                 first it initializes the DSP and the ADC, then      *
*                 transfers data from the ADC to the DSP and stores   *
*                 them within a pre-defined memory table. This        *
*                 program uses the timer as EOC signal of the TLV1544 *
*                 ADC. The data transfer procedure is supported       *
*                 by the two interrupt routines, RINT and TXRXINT.    *
* AUTHOR        : AAP Application Group, Dallas.                      *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.  *
* REFERENCE     : TMS320C2xx User’s Guide, TI 1997
*
*               : Data Acquisition Circuits, TI 1998                  *
***********************************************************************
    .mmregs
    .sect ”.vectors”
    .copy ”VECIO3.asm”

LOCALS    .SET   0
       .sect “.io_reg”
ICR       .set      0ffech
SDTR      .set      0fff0h
SSPCR     .set      0fff1h
ASPCR     .set      0fff5h
IOSR      .set      0fff6h
TCR       .set      0fff8h
PRD       .set      0fff9h
TIM       .set      0fffah
WSGR      .set      0fffch
* Global Variables
    .global      _C1544IO3
    .global       _Samples
    .global       _MemStart
    .global       _Channel
    .global       _ControlMode
    .global       _c_int0
* Local Variables
    .def          ADWORD
    .def          ADCOUNT
    .def          ADMEM
    .def          RINT_COUNT
    .def          END_BIT
    .def          TEMP
    .def          TEMP1
AD_DP           .usect ”.variabl”, 0    ; AD data page for local variable
ADWORD          .usect ”.variabl”, 1    ; control word for the ADC
ADCOUNT         .usect ”.variabl”, 1    ; counter for the samples
ADMEM           .usect ”.variabl”, 1    ; memory pointer during interrupts
RINT_COUNT      .usect ”.variabl”, 1    ; the number of receive interrupts
END_BIT         .usect ”.variabl”, 1    ; end–of–data–transfer–program flag
TEMP            .usect ”.variabl”, 1    ; temporary memory
TEMP1            .usect ”.variabl”, 1    ; temporary memory
_Samples        .usect ”.variabl”, 1    ; user defined number of sample
_MemStart       .usect ”.variabl”, 1    ; user defined memory pointer
_Channel        .usect ”.variabl”, 1    ; user defined channel
_ControlMode    .usect ”.variabl”, 1    ; user defined control mode
    .sect ”.text”
* TLV1544 START
_C1544IO3:
* Save Context
    POPD    *+               ; save return address
    SAR   AR0, *+            ; save Frame Pointer (ar0)
    SAR   AR1, *             ; save Stack Pointer (ar1)
    LAR   ar0,*+,ar1         ; set–up new Frame Pointer
    ADRK  #LOCALS            ; set–up new Stack Pointer
    SST   #0, *+             ; save status register 0
    SST   #1, *+             ; save status register 1
    SAR   AR6, *+            ; save AR6
    SAR   AR7, *+            ; save AR7
    IN    *+, WSGR           ; save wait-sate register
* Initialize DSP
    SETC  INTM               ; disable global interrupts
    LDP   #AD_DP             ; load data page AD_DP
    SPLK  #0h, TEMP          ;
    OUT   TEMP, WSGR         ; set wait–states to zero
* Initialize Serial Port
    SPLK  #000Eh,TEMP        ; set Burst Mode, CLKX = 1/2 CLKOUT1 = 10 MHz
    OUT   TEMP, SSPCR        ; FSX generated by DSP
    SPLK  #003Eh,TEMP        ; activate transmitter and receiver
    OUT   TEMP, SSPCR        ;
* Initialize Asynchronous Serial Port
    SPLK  #0000h,TEMP        ; Reset port, IO0–IO3 seta s input pin,
    OUT   TEMP, ASPCR        ;
    SPLK  #2000h,TEMP        ; activate the Asynchronous Serial Port
    OUT   TEMP, ASPCR        ;
* Load User defined Values
    LACL  _Channel           ; LOAD _Channel into TEMP1
    SACL  TEMP1               ;
    LACL  _MemStart          ; LOAD _MemStart into ADMEM
    SACL  ADMEM              ;
    LACL  _Samples           ; LOAD _Samples into ADCOUNT
    SACL  ADCOUNT            ;
    LACL  _ControlMode       ; LOAD _ControlMode into ARWORD
    SACL  ADWORD             ;
* Enable RINT and TXRXINT
    LDP   #0                 ;
    SPLK   #28h,IFR          ; clear RINT and TXRXINT interrupt flags
    LACL  IMR                ;
    OR    #0028h             ;
    SACL  IMR                ; unmask RINT and TXRXINT
    CLRC  INTM               ; enable global interrupts
* INITIALIZE ADC
    LDP   #AD_DP             ; load data page AD_DP
    SPLK  #0h, RINT_COUNT    ; RINT_COUNT = 0
    SPLK  #1h,END_BIT        ; END_BIT = 1
    CLRC  XF                 ; enable Chip Select
    OUT   ADWORD, SDTR       ; move ADWORD (Conv.Mode) into SDTR
    LACL  TEMP1               ; load TEMP1 (Channel #) into ADWORD
    SACL  ADWORD             ;
* Idle
Idle_Mode:
    IDLE                     ; power down (IDLE), waiting for interrupts
* Exit 1544 program ?
    LDP   #AD_DP
    LACL  END_BIT
    BCND  Idle_Mode,NEQ      ; If END_BIT <> 0, go to Idle_Mode
* Restore Context
    MAR   *, AR1             ; make Stack Pointer (ar1) active
    MAR   *–                 ; decrement Stack Pointer (ar1)
    OUT   *–, WSGR           ; restore WSGR
    LAR   AR7, *–            ; restore AR7
    LAR   AR6, *–            ; restore AR6
    LST   #1, *–             ; restore st1
    LST   #0, *–             ; restore st0
    SBRK  #(LOCALS+1)        ; de–allocate frame size
    LAR   AR0, *–            ; restore old Frame Pointer (ar0)
    PSHD  *                  ; push return address on hardware stack
```

```
* Return to C–program
    RET                     ; return to C–program
**********************************************************************
* RINT
* The receive interrupt routine (RINT) disables the Chip Select signal
* and stores the received data into the memory location specified by
* ADMEM. It increments the memory pointer and decrements the number of
* samples being transferred.
**********************************************************************
RINT_IRQ:
* Disable Chip Select
    SETC  XF                ; disable Chip Select (XF = 1)
* Read Serial Port
    LDP   #AD_DP            ; load data page #AD_DP
    IN    TEMP, SDTR        ; load SDTR into TEMP
    LACC  TEMP, 10          ; load content of temp into ACC ;and shift 10
    SACH  TEMP              ; save (ACH) into TEMP
* Valid data available ?
    LACL  RINT_COUNT        ; ACCL = RINT_COUNT
    ADD   #1                ; increment 1
    SACL  RINT_COUNT        ; RINT_COUNT += 1
    SUB   #2                ; ACCL –= 2
    BCND  RINT_END,LEQ      ; skip saving data until RINT_COUNT > 2
* Save data
    LDP   #AD_DP            ; load data page #AD_DP
    MAR   *,AR7             ; select AR7
    LAR   AR7, ADMEM        ; AR7 = ADMEM
    LACL  TEMP              ; load TEMP into (ACL)
    SACL  *+                ; save sample in the memory
    SAR   AR7, ADMEM        ; increment memory address (ADMEM)
    LACL  ADCOUNT           ; load ADCOUNT into (ACL)
    SUB   #1                ; decrement ADCOUNT
    SACL  ADCOUNT           ; ADCOUNT –= 1
* All samples received ?
    BCND  RINT_END,NEQ      ; if ADCOUNT not 0 jump to DISABLE_CS
* End 1544 program
    LDP   #END_BIT          ; load data page #END_BIT
    SPLK  #0h,END_BIT       ; set END_BIT = 0
    LDP   #0                ; load data page 0
    SPLK  #28h, IFR         ; clear RINT and TXRXINT flags
    LACL  #IMR              ;
    AND   #0FFD7h           ; mask RINT and TXRXINT
    SACL  IMR               ; save ACL into IMR
    CLRC   INTM             ; enable global interrupt
    RET                     ; return from RINT-ISR
* Return from RINT
RINT_END:
    LDP    #AD_DP
    SPLK   #0080h, TEMP     ; clear DIO3
    OUT    TEMP, IOSR
    SPLK   #2200h, TEMP     ; enable Delta interrupt
    OUT    TEMP, ASPCR
    LDP    #0               ; load data page 0
    SPLK   #28h, IFR        ; clear RINT flag and TXRXINT flag
    RET                     ; return from RINT–ISR
****************************************************************************
* IO3
* The interrupt routine for the internal interrupt, IO3, is initiated
* by the end-of-conversion signal sent from the ADC. This routine enables
* the Chip Select signal and sends the current control word, defined
* in ADWORD, to the ADC.
****************************************************************************
IO3_IRQ:
* Configure ADC
    LDP   #AD_DP            ; load data page #AD_DP
    CLRC  XF                 ; enable Chip Select
    OUT   ADWORD, SDTR       ; move ADWORD into SDTR
* Configure the Asynchronous Serial Port
    SPLK  #0080h, TEMP       ; clear DIO3
    OUT   TEMP, IOSR
    SPLK  #2000h, TEMP       ; disable delta interrupt
    OUT   TEMP, ASPCR
    KDO   #0
    SPLK  #20h, IFR          ; clear TXRXINT flag
    CLR   INTM               ; enable global interrupts
    RET                      ; Return from interrupt to “Exit 1544 program /”
   .copy  ”BOOTIO3.asm”
```

A.4 Vector Table: VECIO3.ASM

```
*************************************************************************
* TITLE       : TLV1544 ADC Interface Vector routine                    *
*               (IO3 as interrupt source)                               *
* FILE        : VECIO3.ASM                                              *
* DESCRIPTION : This file defines the interrupt vector table.           *
*               If RINT occurs: vector points to RINT_IRQ subroutine.   *
*               If IO3 occurs: vector points to IO3_IRQ subroutine.     *
* AUTHOR      : AAP Application Group, Dallas                           *
*               CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.      *
* REFERENCE   : TMS320C2xx User’s Guide, TI 1997                        *
*************************************************************************
         .global     _main
RS       b   _c_int0           ; 0x00; RESET
INT1     b   INT1              ; 0x02; external user interrupt #1
INT2     b   INT3              ; 0x04; external user interrupt #2 and #3
TINT     b   TINT              ; 0x06; internal timer interrupt
RINT     b   RINT_IRQ          ; 0x08; Serial Port receive interrupt
XINT     b   XINT              ; 0x0A; Serial Port transmit interrupt
TXRXINT  b   IO3_IRQ           ; 0x0C; Asynchronous Serial Port interrupt
```