BURR-BROWN®
APPLICATION BULLETIN
Mailing Address: PO Box 11400, Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706 • Tel: (520) 746-1111
Telex: 066-6491 • FAX (520) 889-1510 • Product Info: (800) 548-6132 • Internet: www.burr-brown.com/ • FAXLine: (800) 548-6133

INTERFACING THE ADS7822 TO THE SYNCHRONOUS SERIAL PORT OF THE 80x51 MICROCONTROLLER

By Gebhard Haug and Bonnie C. Baker

Analog-to-digital converters can be controlled by a normal I/O port or with the synchronous serial port of the 80x51 microcontrollers. The synchronous serial port is more efficient, however, there are some pitfalls to be aware of when configuring the 8-pin A/D converters (ADS1286, ADS7816, ADS7817 and ADS7822) from Burr-Brown. This application bulletin describes how to get around these pitfalls and get the best performance out of this combination.

The ADS1286, ADS7816, ADS7817, and ADS7822 are all 12-bit converters that are available in a variety of 8-pin packages. These devices are classical successive approximation register (SAR) A/D converters. Their architecture is based on capacitive redistribution, which inherently includes a sample/hold function. All four of these converters have three digital communication lines in their interface. These communication lines are $\overline{CS}$/SHDN, D_{OUT} and DCLOCK. The $\overline{CS}$/SHDN pin provides a chip select function when LOW. When this pin is pulled HIGH, the A/D converter goes into its shutdown mode. The basic timing diagram for these A/D converters is shown in Figure 1. The timing specifications for the individual A/D converters are called out in Table 1.

SYMBOL	DESCRIPTION	ADS1286	ADS7816	ADS7817	ADS7822	UNITS
t_{SMPL}	Analog Input Sample Time	1.5 (typ)	1.5 (min) 2.0 (max)	1.5 (min) 2.0 (max)	1.5 (min) 2.0 (max)	Clk Cycles
t_{CONV}	Conversion Time	12	12	12	12	Clk Cycles
t_{CYC}	Throughput Rate	20 (max)	200 (max)	200 (max)	75 (max)	kHz
t_{CSD}	$\overline{CS}$ Falling to DCLOCK LOW	0 (max)	0 (max)	0 (max)	0 (max)	ns
t_{SUCS}	$\overline{CS}$ Falling to DCLOCK Rising	30 (min)	30 (min)	30 (min)	30 (min)	ns
t_{hDO}	DCLOCK Falling to Current D_{OUT} not Valid	15 (min)	15 (min)	15 (min)	15 (min)	ns
t_{dDO}	DCLOCK Falling to Next D_{OUT} Valid	150 (max)	150 (max)	150 (max)	200 (max)	ns

TABLE I. Timing Specifications for the ADS1286, ADS7816, ADS7817, and ADS7822.

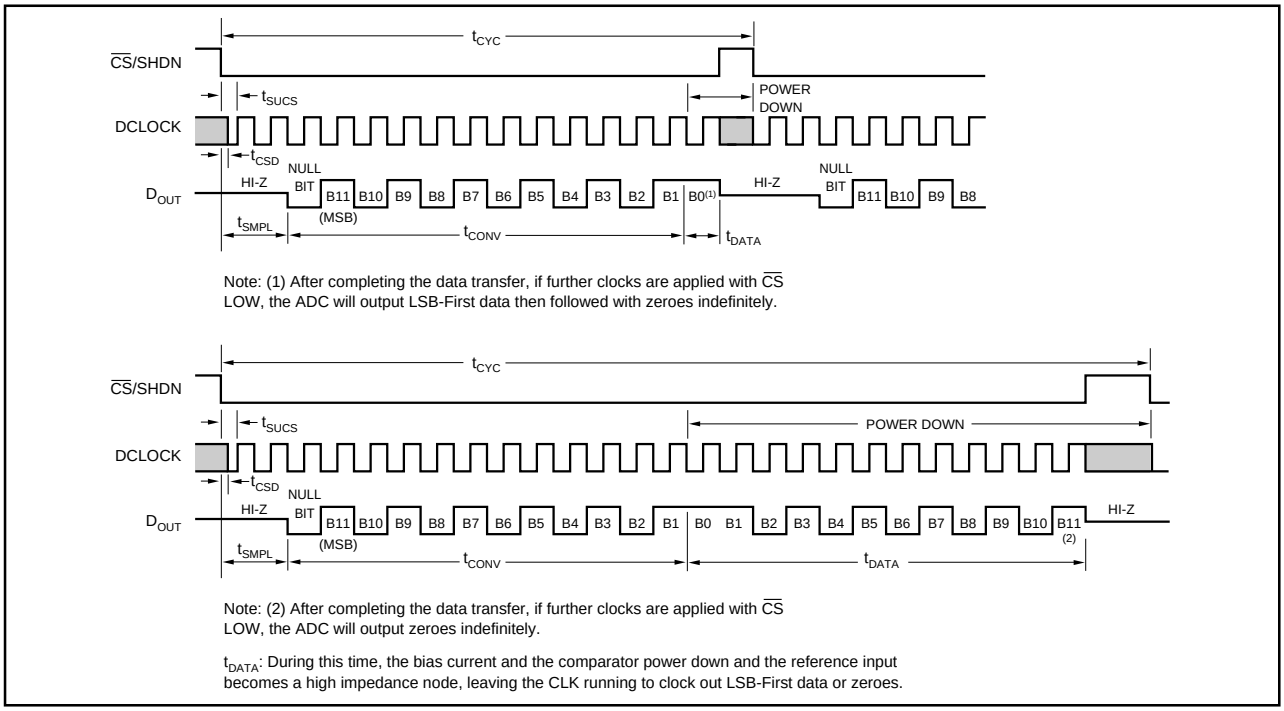

FIGURE 1. Timing Diagram for the ADS1286, ADS7816, ADS7817, and ADS7822.

Printed in U.S.A. September, 1997

As shown in Figure 1, a falling $\overline{CS}$ signal initiates the conversion and data transfer. In addition, any conversion data from previous conversions is erased with the falling edge of $\overline{CS}$. The first 1.5 to 2.0 clock periods of the conversion cycle are used to sample the input signal. After the second falling DCLOCK edge, D_{OUT} is enabled and will output a LOW value for one clock period. For the next 12 DCLOCK periods, D_{OUT} will output the conversion result, most significant bit (MSB) first. After the least significant bit (B0) has been transmitted, subsequent clocks will repeat the output data but in a least significant bit (LSB) first format.

The suggested circuit for these A/D converters and the 80x51 is shown in Figure 2. The serial port of the 80x51 is configured in the normal mode (or mode 0). In this mode, the serial data enters and exits through RxD. Eight bits are transmitted or received (LSB first) through RxD. In this circuit, the RxD/P1.2 port is used exclusively in the serial port interface mode and configured to receive serial data from the A/D converter. TxD outputs the shift clock of the microcontroller. The baud rate of TxD is fixed at 1/12 the oscillator frequency. For this circuit, TxD/P1.3 is used in the serial port interface mode as well as the I/O mode providing the DCLOCK signal that is required by the A/D converter during the conversion process. Finally, the P1.1 pin of the 80x51 microcontroller is chosen to drive the chip select ($\overline{CS}$) pin of the A/D converter. The selection of this microcontroller pin is arbitrary.

The serial port timing diagram for the 80x51 type microcontroller is shown in Figure 3. Note that the serial port clock that is generated by the 80x51 device normally idles HIGH. This produces a potential problem with these A/D converters as the $\overline{CS}$/SHDN signal is pulled LOW to start a conversion. In some instances, a conversion will occur with this timing arrangement by requiring an additional clock cycle in order to complete the data transfer. In other cases, a conversion will not occur unless the DCLOCK is LOW at the time the $\overline{CS}$/SHDN goes LOW.

Clock Issues

There are two possible solutions to these problems: 1. Use an external inverter to invert the controller's clock (the clock will idle LOW) or 2. Implement a software modification with the 80x51 controller.

With the first solution, timing becomes critical. The 80x51 must latch the data into its internal shift register with the same clock edge that the A/D converter uses to advance its serial output stream. The relationship of the 80x51 timing and the A/D converter timing is illustrated in Figure 4. There are two timing issues that make this possible. The first is the time delay of the inverter that has been placed between RxD and DCLOCK or the A/D converter. The second is the hold time of the A/D converter. The set-up and hold times of this series of converters are shown in Figure 5.

A better approach is to modify the software and not include additional inverters in the clock line. In this second solution, the controller's CLK, in this case P1.3, is brought LOW before setting $\overline{CS}$/SHDN LOW. As the next instruction, P1.3 is brought HIGH again to allow the serial port to operate normally. All future clock signals needed for the conversion are done internally by the controller's serial port (TxD/P1.3). Figure 6 gives the code for this algorithm.

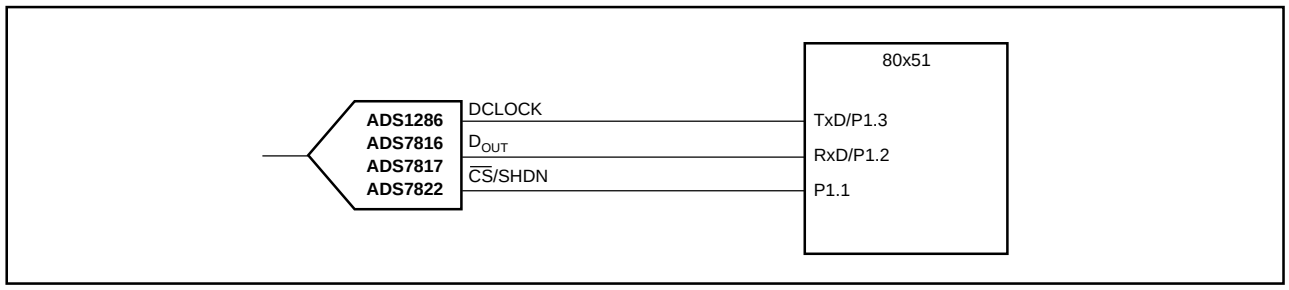

FIGURE 2. Circuit Diagram for the ADS7816, ADS7817, or ADS7822 and a Generic 80x51 Microcontroller.

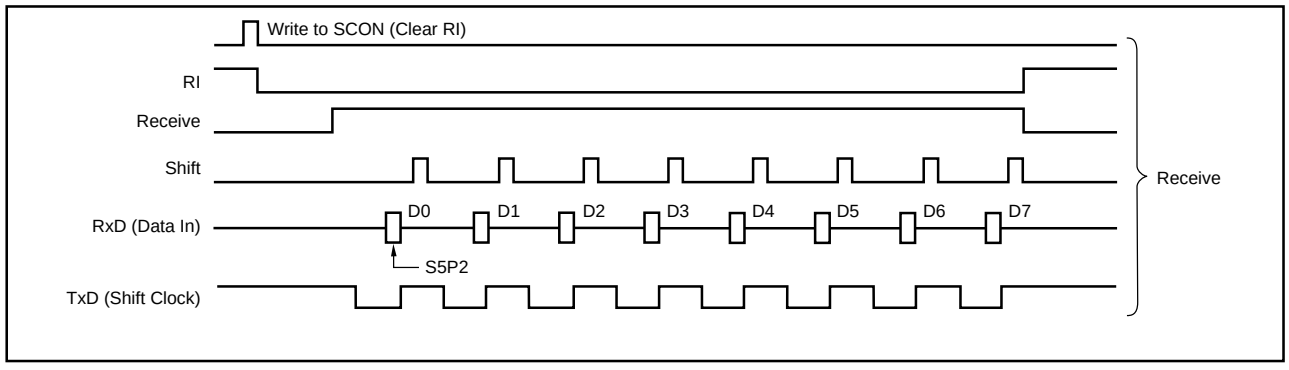

FIGURE 3. Serial Port Timing Diagram for the 80x51 Type Microcontroller.

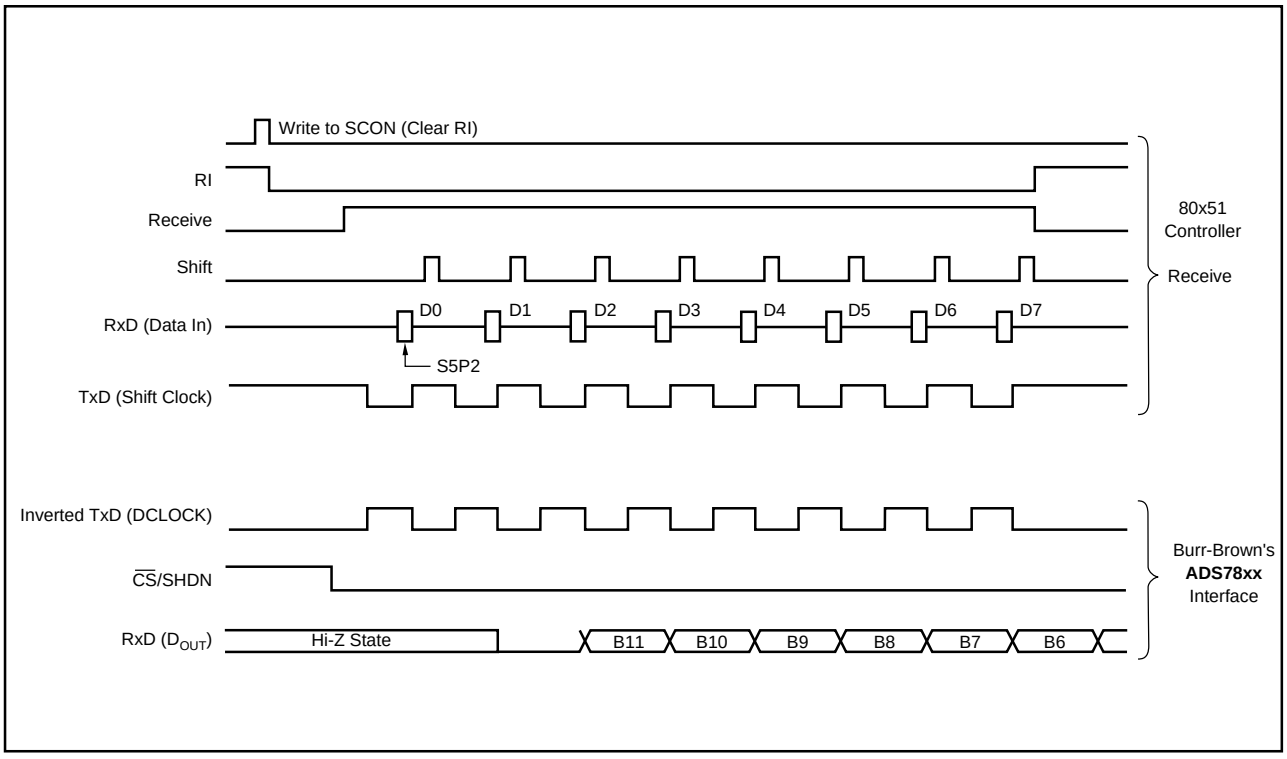

FIGURE 4. Timing Diagram for the ADS7816, ADS7817, or ADS7822 and a Generic 80x51 Microcontroller. This Timing Operation Requires that the TxD Output be Inverted with an External Gate.

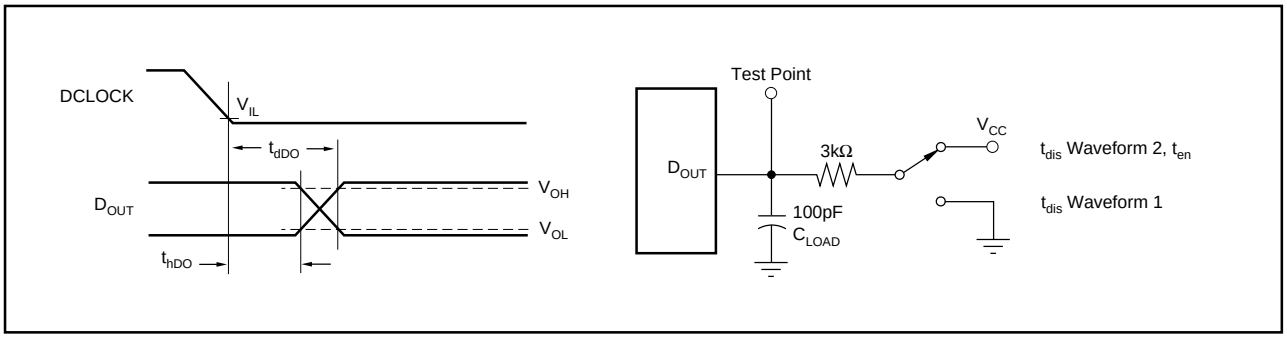

FIGURE 5. Set-up (t_{dDO}) and Hold (t_{hDO}) Times of the ADS1286, ADS7816, ADS7817 and ADS7822. See Table I for Timing Details.

```
    clr     ads_clk             ; make sure dclock is low (P1.3) when cs falls
    clr     ads_cs              ; P1.1
    setb    ads_clk             ; set bit (P1.3) to enable serial port
    setb    –REN                ; enable serial port to receive data
    clr     –RI                 ; clear receive register
                                ; to start transfer
GetAD10:
    Jnb     _RI, Get AD10       ; wait until done
```

FIGURE 6. Code Required to Implement the Software Solution for Clock Conflict (shown in Figure 4) Between the 80x51 Type Microcontroller and Burr-Brown's 8-Pin A/D Converter.

Adjusting the Bit Order for the Controller

These SAR converters always output the MSB first when a new sample is converted. On the other hand, the 80x51 devices expect data to be transferred to the serial port with LSB first. If the first two bytes (one byte = 8 bits) are transferred, the data bits must be re-ordered by the software of the controller.

An alternative to this approach is to transfer a third byte from the A/D converter to the controller. As mentioned earlier in this Application Bulletin, the A/D converter transfers the 12 bits of the conversion MSB first and then follows it by the same 12 bits of the conversion LSB first. This "third byte approach" timing between the microcontroller and A/D converter is shown in Figure 7. (Refer to Figure 1 for the A/D converter's timing diagram for these details.)

If one looks closely at the timing shown in Figure 7, they will notice that bit 11 (MSB) is missing. This is due to the fact that the LSB does not start at the byte or nibble boundary. It appears as if the controller will still have to perform bit shifting to finally capture the data of the converter. On the contrary, the software can create one more clock cycle at the beginning of the conversion process. In this manner, the serial system will produce the desired results as shown in Figure 8. The code for this enhancement is given in Figure 9.

A short program in 80x51 assembly language is shown in Figure 10. This program can also be downloaded from Burr-Brown's WEB site (http://www.burr-brown.com/). It is titled "ADS78xx vs 80x51 Interface." This short program demonstrates how to interface the ADS7822 to the synchronous serial port of the 80x51 devices.

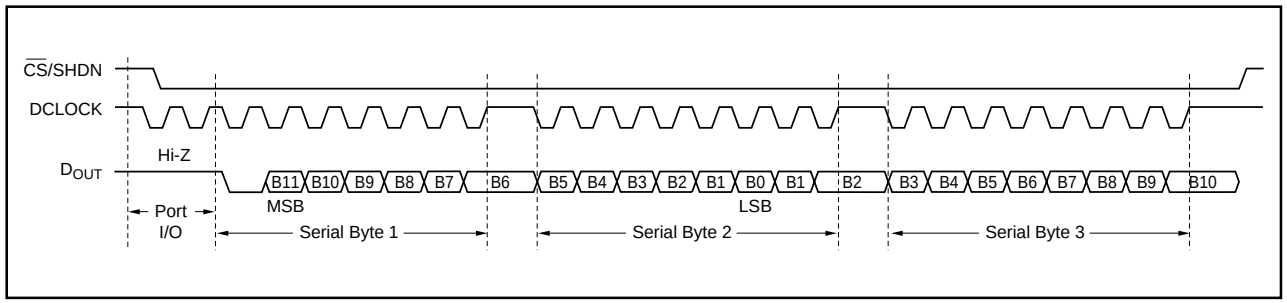

FIGURE 7. Since the 80x51 Type Controller Requires LSB First, the A/D Converter Must Output Its First 12 Bits (MSB First) and the 12 More Bits (LSB First). Notice that Bit 11 is Missing at the End of the Transfer.

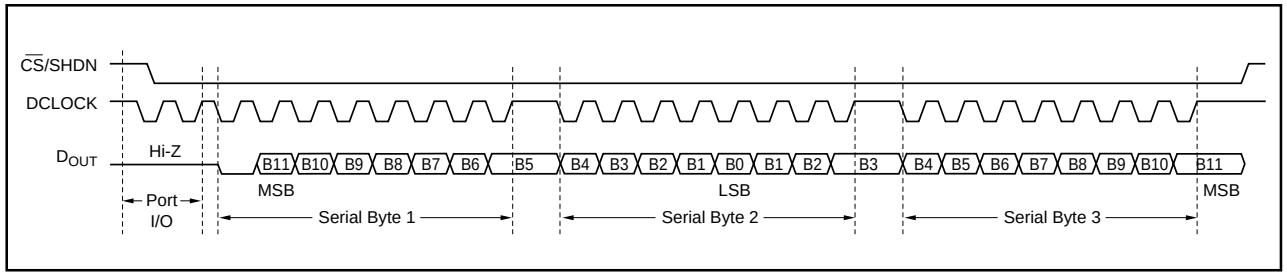

FIGURE 8. A Minor Software Change Corrects the Problem Illustrated in Figure 7.

```
    clr     ads_clk          ; make sure dclock is low (P1.3) when cs falls
    clr     ads_cs           ; P1.1
    setb    ads_clk          ; send one additional dclock cycle (P1.3)
    clr     ads_clk
    setb    ads_clk          ; set bit (P1.3) to enable serial port
    setb    _REN             ; enable serial port to receive data
    clr     _RI              ; clear receive register to start transfer
GetAD10:
```

FIGURE 9. This Code is Required to Implement the Software Solution that Provides the LSB First to the 80x51 Type Microcontroller from Burr-Brown's 8-Pin A/D Converters.

```
; 80x51.ASM
;
; [purpose]
; interfaces ADS7822 to DALLAS Microcontroller via the sync. serial port
;*********************************************************************
; serial port used
       _SCON     EQU       0c0h          ; serial port config register
       _SBUF     EQU       0c1h          ; serial port data buffer
       _RI       BIT       _SCON.0
       _TI       BIT       _SCON.1
       _REN      BIT       _SCON.4
ads_cs  bit p1.1                         ; chip select bit
ads_dta bit p1.2
ads_clk bit p1.3                         ; dclock bit
       org       08100h
Reset:
       mov       sp, #07fh               ; init stack pointer first
       call      Init                    ; do global init stuff
Loop:
       call      GetAD                   ; get a reading
       call      ADDisplay               ; display it
       call      CRLF                    ; and goto the next line
       jmp       Loop
;*********************************************************************
Init:
       mov       _SCON, #00h             ; init serial port for synchronous I/O
       ret
;*********************************************************************
; [GetAD]
; Gets a reading from an ADS7822
; [Parameters]
; none
; [returns]
; Hibyte in 020h
; Lobyte in 021h
GetAD:
       clr       ads_clk                 ; make sure clk is low when cs falls
       clr       ads_cs
       setb      ads_clk                 ; one additional clock tick to make
       clr       ads_clk                 ; bit 0 as bit 3 in the second data byte
       setb      ads_clk
       setb      _REN                    ; enable serial port receive data
       clr       _RI                     ; clear receive register to start transfer
GetAD10:
       jnb       _RI, GetAD10            ; wait until done
       clr       _RI                     ; start another transfer
GetAD20:
       jnb       _RI, GetAD20            ; wait until done
       mov       a, _SBUF
       clr       _RI                     ; start last transfer
       anl       a, #0f0h                ; mask unwanted bits
       mov       021h, a                 ; save low byte
GetAD30:
       jnb       _RI, GetAD30
       mov       a, _SBUF                ; get high byte
       mo        020h, a                 ; save high byte
       setb      ads_cs                  ; done converting, reset CS
       ret
```

FIGURE 10. This Program Will Interface the ADS7822 to the Synchronous Serial Port of the 80x51 Type Device.

```
;***********************************************************************
; [ADDisplay]
; sends contents of 020, 021h to the serial port
; [Parameters]
; none
; [returns]
; -
ADDisplay:
       mov      a, 020h              ; high byte
       call     BinToHex             ; high byte to hex digits
       call     WriteSerial
       mov      a, b
       call     WriteSerial
       mov      a, 021h
       call     BinToHex             ; same with low byte
       call     WriteSerial
       mov      a, b
       call     WriteSerial
       ret
;***********************************************************************
; [CRLF]
; sends a cr, lf sequence to the serial port
; [Parameters]
; none
; [returns]
; -
CRLF:
       mov      a, #0dh              ; write CR
       call     WriteSerial
       mov      a, #0ah              ; write LF
       call     WriteSerial
       ret
;***********************************************************************
; [WriteSerial]
; sends the byte contained in a to the serial port
; [Parameters]
; Accu: byte to send
; [returns]
; -
WriteSerial:
       clr      TI                   ; clear transmit register
       mov      SBUF, A              ; write data
WriteSerial10:
       jnb      TI, WriteSerial10    ; wait until done
       ret
;***********************************************************************
; [WriteSerial]
; converts binary number to two hex digits
; [Parameters]
; Accu: byte to convert
; [returns]
; Accu: first hex digit
; B: second hex digit
BinToHex:
       push     dph
       push     dpl
       mov      dptr,#HexTable
       push     acc
       anl      a,#00fh
```

FIGURE 10 (Cont). This Program will Interface the ADS7822 to the Synchronous Serial Port of the 80x51 Type Device.

```
        movc    a,@a+dptr
        mov     b,a
        pop     acc
        anl     a,#0f0h
        swap    a
        movc    a,@a+dptr
        pop     dpl
        pop     dph
        ret
HexTable:
        db ‘0’,’1',’2',’3',’4',’5',’6',’7'
        db ‘8’,’9',’A’,’B’,’C’,’D’,’E’,’F’
        end
```

FIGURE 10 (Cont). This Program will Interface the ADS7822 to the Synchronous Serial Port of the 80x51 Type Device.

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 acknowledgment, 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.

Customers are responsible for their applications using TI components.

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.