Using the TMS370 A/D Converter Module

Henry Kwan
Microcontroller Applications Engineering
Semiconductor Group
Texas Instruments Incorporated

TEXAS INSTRUMENTS

IMPORTANT NOTICE

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

TI warrants performance of its semiconductor products to current specifications 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. Unless mandated by government requirements, specific testing of all parameters of each device is not necessarily performed.

TI assumes no liability for TI applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that 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

Illustrations

Tables

Section 1

Introduction

To provide advanced performance and a cost-effective system solution to complex control applications, the TMS370 family combines an 8-bit CPU containing powerful peripherals such as A/D converter, timers, SPI and SCI with on-chip memory RAM, ROM, EEPROM and EPROM. Many applications involve the determination of values of physical parameters such as temperature, position, and pressure, which must be transformed into electrical (analog) signals and then converted to digital codes for the controller. With the on-chip Analog-to-Digital (A/D) Converter, the TMS370 microcontrollers greatly simplify interactions between the analog world and a digital system. This application report illustrates the operation of the on-chip A/D converter and provides some application examples for analog-to-digital conversions with the TMS370 family microcontrollers.

Section 2

Module Description

The A/D Converter module is an 8-bit successive approximation converter with internal sample-and-hold circuitry. The module has eight multiplexed analog input channels, which allow the processor to convert the voltage level of up to eight different sources. The A/D Converter contains three major blocks: an analog (input and reference) multiplexer, successive approximation A/D converter with internal sample-and-hold circuitry, and interrupt logic.

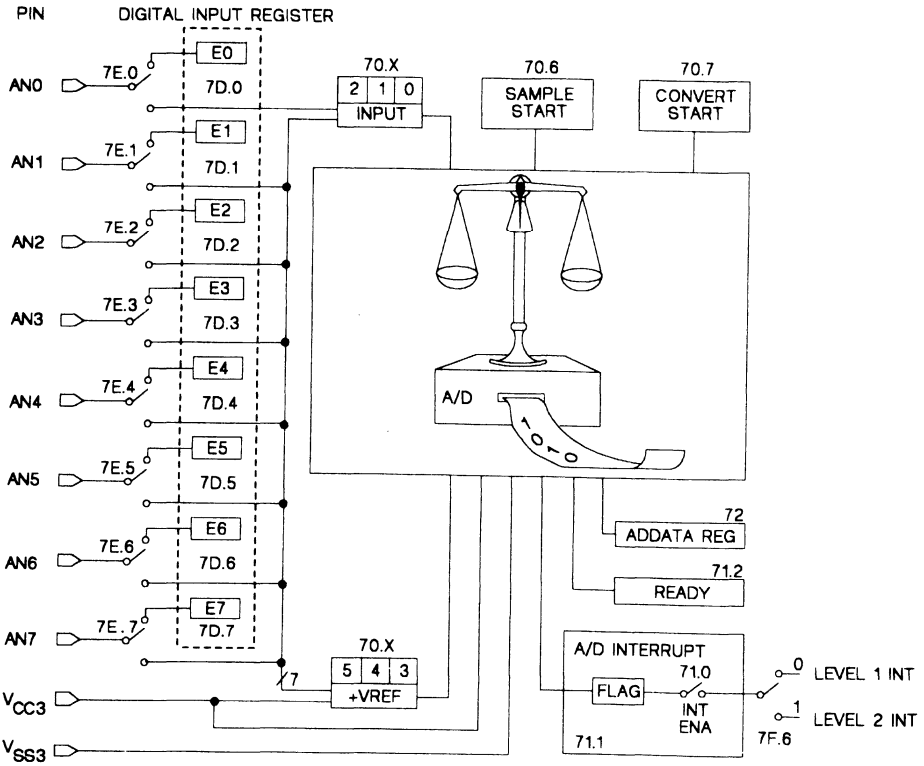


Figure 2-1. A/D Converter Block Diagram

2.1 Principles of Operation

Successive approximation is one of the most common techniques used in A/D conversion. The technique generates each bit of the digital code sequentially, starting with the MSB, and compares the analog input with binary-weighted values to produce the output in a fixed number of steps. Successive approximation provides an excellent trade-off between resolution, speed, accuracy and cost.

Figure 2-2 shows a simplified diagram of the successive approximation A/D converter.

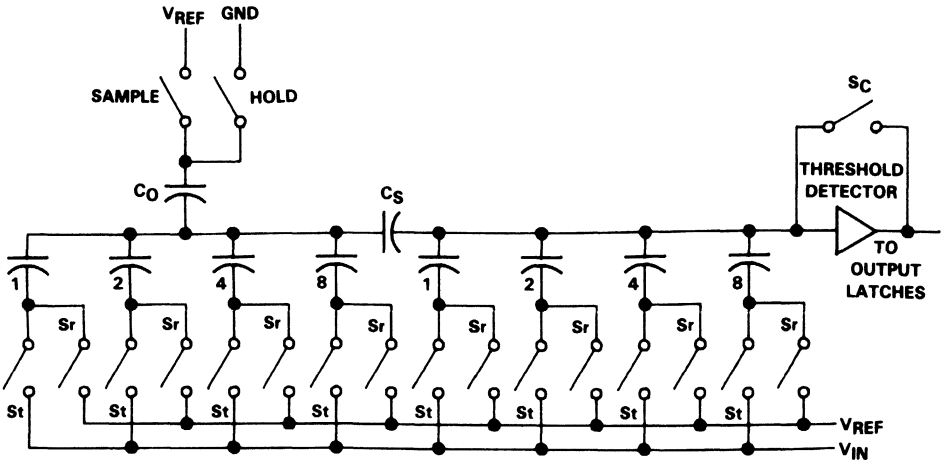


Figure 2-2. Simplified Model of the Successive Approximation Converter

The series capacitor Cs effectively divides the value of the lefthand side capacitors by 16 to form a binary-weighted capacitor array. The conversion process is accomplished by a sequence of three operations. In the first sequence, the sampled mode, the analog input is sampled by connecting V_{IN} to the analog input, closing switch Sc and all St switches. All capacitors charge up to the input voltage simultaneously during the sampling time. Capacitor Co is switched to V_{REF} during sampling mode. In the second sequence, the hold mode, capacitor Co is switched to GND; Sc switch is opened; and V_{IN} is connected to GND. The third sequence, the redistribution mode begins by identifying the charge on each capacitor relative to the reference voltage.

All eight capacitors are examined separately until all eight bits are determined. The rightmost capacitor (corresponding to MSB) is first switched to the reference voltage, and all of the other capacitors are switched to GND. If the voltage at the summing node is greater than the trip point of the threshold

detector, a bit is set in the output register and the capacitor is switched back to GND. If the voltage at the summing node is less than the trip point of the threshold detector, the capacitor remains connected to V_{REF} throughout the remainder of the conversion process. This process is repeated for all eight capacitors.

2.2 Functional Description

The A/D module has ten input pins. Two pins are used for analog voltage supply V_{CC3} and V_{SS3}, which isolates the A/D module from digital switching noise. The other eight pins (AN0 - AN7) are used for analog input channels and can be configured as general purpose input pins if not needed. The analog reference can be either V_{CC3} or one of the analog input channels AN1 to AN7. This allows for ratio measurement of one analog signal to another.

The internal sample-and-hold circuitry is used to maintain the analog input during conversion. This minimizes inaccuracies in the converted value of an analog signal due to changes in the signal's value during the conversion process. The input sampling begins when the SAMPLE START bit (bit 6 of the ADCTL) is set. The application program should allow 1 µs for each kilohm of source output impedance or a minimum of 1 µs for the low-impedance source to sample the analog signal. This allows time to charge the internal capacitor array. When the sampling time is completed, the SAMPLE START and the CONVERT START bit (bit 7 of the ADCTL) are set. The analog signal's value will be held by the A/D module for 18 cycles after the CONVERT START bit is set. By that time, the A/D module has cleared both the SAMPLE START and CONVERT START bit to signify the end of the internal sampling phase.

After the internal sampling phase, the program can change the input channel without affecting the conversion. The reference voltage should remain constant throughout the conversion.

The conversion process takes 164 system clock cycles after the CONVERT START bit is set. Upon completion, the AD INT FLAG will be set. If the AD INT ENA bit is set, the module will generate an interrupt request. Section 4 illustrates software examples of A/D initialization and operation.

Section 3

Design Considerations

The following section provides a starting point for the digital designer by offering some hints for the analog interface. For a more thorough discussion of additional analog devices (such as op-amp and filter circuits), refer to additional analog applications literature.

3.1 A/D Input Pin Model

A model of the A/D input pin (shown in Figure 3-1) is intended to facilitate the user's understanding of the effects of interface circuitry on the A/D conversion.

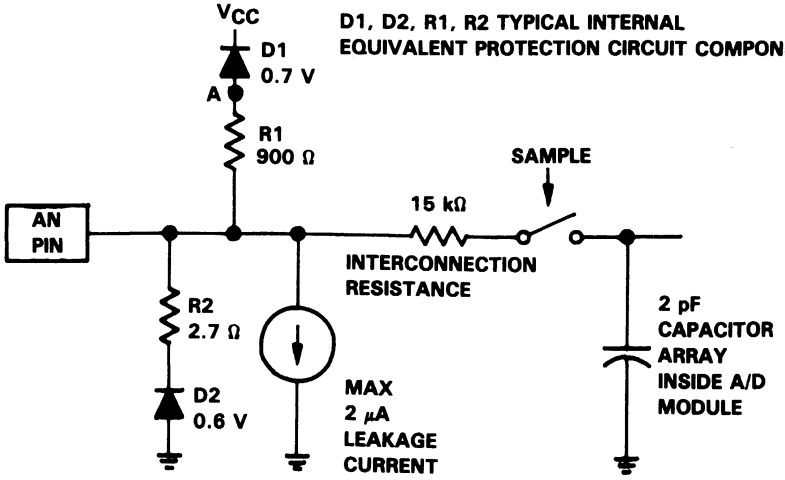


Figure 3-1. A/D Input Pin Model

3.2 Analog Input Pin Connection

The external pin connection can greatly affect the performance and accuracy of the A/D conversion. Since the A/D converter uses the charge redistribution technique to sample the analog signal, there is no need to use external sample-and-hold circuitry. Using an external low-pass filter to reduce system noise may help to prevent errors. Simple noise filtering can be accomplished by adding a resistor and capacitor across the A/D inputs as shown in Figure 3-3 and Figure 3-4. For inexpensive filtering Cx acts with Rx to form a first-order low-pass network. However, the capacitor and resistor size should be chosen carefully to preclude additional system errors.

One of the most common A/D application errors is inappropriate source impedance. Too much source impedance might introduce unexpected system errors, and too little source impedance might cause permanent damage to the A/D input pins because of a possible latch-up problem. In practice, minimum source impedance should be used to limit the error as well as minimize the required sampling time; however, source impedance should be large enough to limit the current sufficiently to protect against an over-voltage condition.

When the reference voltage, V_{REF}, is at 5.1 V, one LSB corresponds to 20 mV. From the input pin model, the maximum leakage current is 2 μA. (See Note[1].) That is, for the worst case of 2 μA leakage current flow through a 1K external resistor will result in a 2 mV voltage drop or induce 0.1 LSB error. If the source impedance induces an error higher than can be tolerated by the system, a buffering device (e.g., op-amp) might be considered.

Latch-up poses a different problem for the input pin connection. Latch-up is the uncontrolled flow of current through the parasitic SCR inherent in all CMOS devices. This SCR might be triggered into a low-impedance state, resulting in excessive supply current. Once the SCR is triggered, the current flow is limited only by the impedance of the power supply and the forward resistance of the SCR. An external resistance should be used to limit the current flow through the A/D pin so that the current is never high enough to cause CMOS latch-up. The source resistance will depend on the total system.

The absolute maximum rating of the analog pin should not exceed the values specified in the electrical specification. The input voltage range should be within -0.3 V to 7 V and the input current should be within ±10 μA.

Suppose for example, the worst case for an application, the analog input signal is shorted to 12 V. An external resistor will be required to limit the input voltage below 7 V in order to protect the input pin from damaging. Also the internal diode to V_{CC} (5 V) would clamp the voltage at node A (see Figure 3-1) to 5.7 V. Let X be the resistance of the external resistor. Therefore,

$$\frac{12 - 7}{X} = \frac{12 - 5.7}{900 + X}$$

or $$X = 3.46\ k$$

It is suggested that the designer add in some guard band for tolerance of the internal resistance and fluctuations of the external power supplies. The designer may also consider using external clamping diodes to limit the analog voltage range between V_{SS3} and 7 V. However, if clamping diodes are used, the leakage current induced by the diodes should be kept as low as possible.

If an external capacitor is added to form a low-pass filter, the capacitance value should be chosen carefully. The capacitor size mainly depends on the frequency of the analog input signal and the sampling time allowed. Obviously the RC time constant needs to be large enough to filter any undesirable noise signal, but it must be expected that the external filter also introduces a delay between the analog source and the A/D input pin. It is important to make sure that the RC time constant is much smaller (say 10 times smaller) than the sample time to allow the internal capacitor array to become fully charged within the sampling window. Adding an external capacitor can also increase protection in case an over-voltage condition occurs. In combination with the external resistor, the external capacitor limits the rise time of large spikes so that the diode can clip them before they do any damage.

---

1 Note: Refer to TMS370Cx5x 8-Bit Microcontrollers Data Sheet for updated maximum leakage current.

3.3 Analog Input Conditioning

For applications dealing with stringent conditions, one might consider adding in op-amps or related devices for signal conditioning, e.g., buffering, amplification, level translation, linearization or current-to-voltage conversion. The following figure and table show the op-amp symbol and some key op-amp parameters.

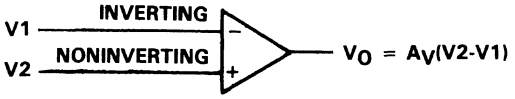


Figure 3-2. Operational Amplifier

Table 3-1. Key Op-Amp Parameters

Key Parameters	Description	Ideal Op-amp
Input Resistance	Resistance at either input of the op-amp (load of the source)	Infinity
Output Resistance	Source impedance of the output stage	0
Differential voltage gain or open-loop voltage gain (Av)	The ratio of the input voltage to output voltage without external feedback	Infinity
Slew Rate (V/μs)	Response time of the op-amp's output (rise and fall time)	Infinity
Common Mode Rejection	Ability to limit a response to a common mode voltage (noise rejection)	Infinity
Bandwidth	Frequency response of the op-amp	Infinity

Operational amplifiers can be configured to perform a large number of functions. Because of their variable characteristics and wide range of adaptability, they are very handy for analog signal interfacing. Two popular input buffer configurations for the op-amp are shown in Figure 3-3 and Figure 3-4.

The non-inverting configuration provides amplification of small input signals and provides low source impedance for the A/D converter. The inverting amplifier configuration affords convenient scaling of negative input for the A/D converter (the A/D module does not convert input below the value of V_{SS3}). Resistors R1 and R2 determine the transfer function (gain) of the amplifier circuitry. Resistor R3 (equivalent to R1 in parallel with R2) is included to

correct the DC offset caused by internal input offset or input bias current. Some op-amps like LinCMOS (TLC272) provide extremely low input bias performance, thus eliminating the need for bias compensation resistors and thereby simplifying the interface circuits. Some op-amps also provide additional terminals for input offset or frequency compensation.

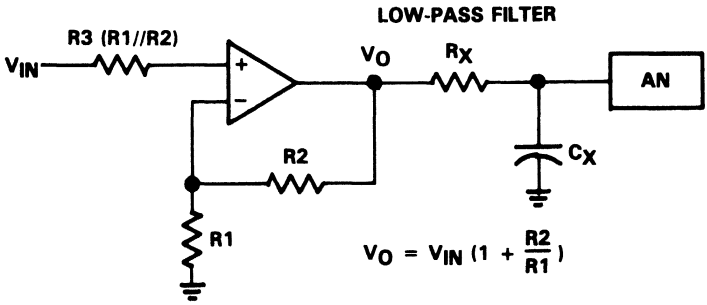


Figure 3-3. Non-inverting Buffer for Analog Input Pin

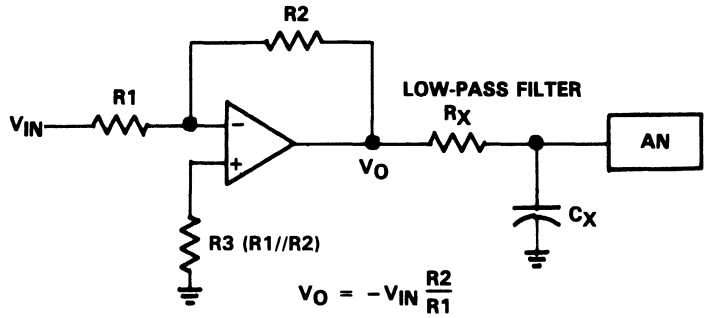


Figure 3-4. Inverting Buffer for Analog Input Pin

With these two basic configurations, the resistance value and reference can be manipulated to provide optimal scaling and range offsetting of the input signal for A/D conversion. For example, in Figure 3-5 the output of a transducer, with an output of range 2.5 V to 12.5 V, might be offset by 2.5 V [(2.5 V to 12.5 V) - 2.5 V], and then scaled down 0.5 (R/2R) by the amplifier to provide 0 V to 5 V input signals to the A/D converter.

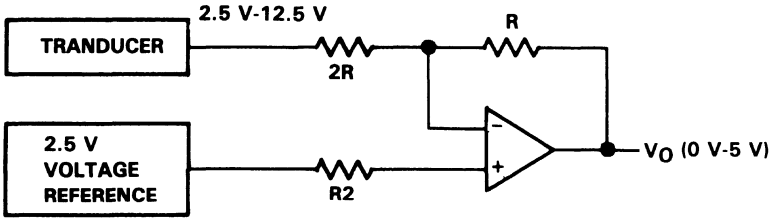


Figure 3-5. Range Offsetting and Scaling

The bridge amplifier is another very popular interfacing circuit especially applicable with input transducers. Transducers, like strain gauges and thermistors, simply produce a varying resistance over a range of parameter (pressure or temperature) changes. Figure 3-6 shows a typical bridge amplifier circuit. A bridge consists of four terminal elements, one of them (resistance) is variable by a factor of (1+X), where X is a fraction as a function of other parameters (e.g., temperature, pressure). The bridge amplifier measures the deviation of the resistance (good common mode rejection) from the initial value as an indication of change of the parameter (temperature). A more detailed example is discussed in Section 5.

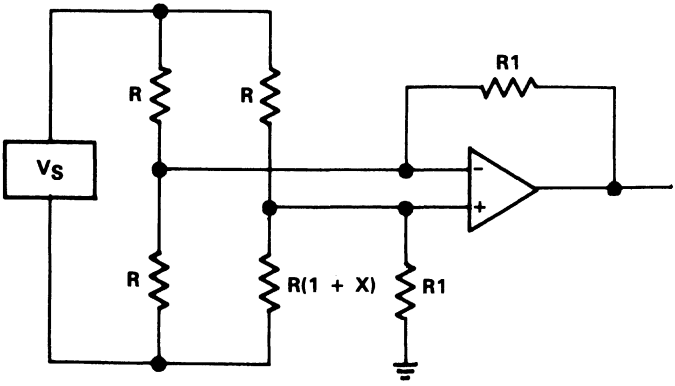


Figure 3-6. Bridge Amplifier

Other basic operational amplifier circuits which might be configured with the A/D module can provide different types of signal conditioning for different applications. For examples, unit-gain voltage follower can be used as an input buffer to the A/D converter, current amplifier can provide current to voltage converter, low pass filter can reduce system noise to achieve better A/D con-

version accuracy, logarithmic-amp can compress the input signal from several orders-of-magnitude to a non-linear input signal with fix percent of relative accuracy throughout the required range. For more information, refer to linear circuits application manuals and literature.

3.4 Resolution

Some applications may need more resolution than an 8-bit A/D can provide. One way to get around this problem is to apply scaling and offsetting in order to manipulate the input signal and use more than one channel for conversion as shown in Figure 3-7.

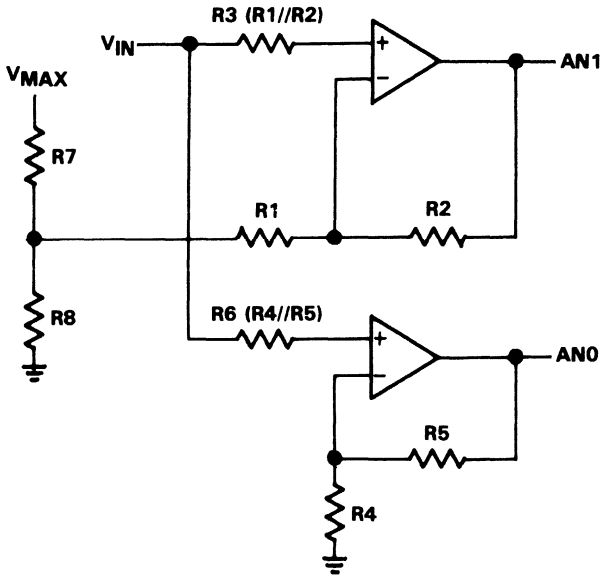


Figure 3-7. Example of Interface Circuit to Increase Resolution to 9-Bits

The input signal is split into two ranges: one channel converts the input signal from 0 to $V_{MAX}/2$, while the other channel converts the input signal from $V_{MAX}/2$ to V_{MAX}. The following discussion describes an application that requires the conversion of an input signal from 0 to 5 V with 10 mV resolution per step.

Resistors R1, R2, R4, and R5 are set to provide a gain of two for the amplifier. Resistors R7 and R8 form a voltage divider to provide an offset of 2.5 V ($V_{MAX}/2$) for the op-amp. When the input signal is within the range 0 V -

2.5 V, channel AN0 provides the conversion result (8 bits digital output) with the MSB (bit 8, the extra bit) equal to 0. The output of channel AN1 will be zero because of the offset. When the input signal is within the range 2.5 V - 5 V, channel AN1 provides the conversion result (8 bits digital output) with the MSB (bit 8, the extra bit) equal to 1. The output of channel AN0 will be FF (its full-scale value). The user should note that when the input signal is within the range 2.5 V to 5 V, the output of channel AN0 can be clamped to V_{CC}+0.3 V by using a protection diode.

Usually additional variable resistors are needed to adjust the gain and offset of the amplifiers. However, with on-chip EEPROM, the gain error can be compensated without adjusting the external resistor. The precise value of the resistor is not important. The amplifier can be calibrated with known input values, and the actual gain of the circuit is calculated and stored in the EEPROM. The actual value of the conversion result can be calculated based on this gain factor.

The user can also avoid adjusting the offset of the amplifier by sacrificing the resolution. Resistors R7 and R8 are chosen so that the ranges are overlapping. In that case, the exact values of the resistors (offset of the op-amp) are not important. You can also use an additional op-amp or increase the gain of the amplifier to compensate for overlapping.

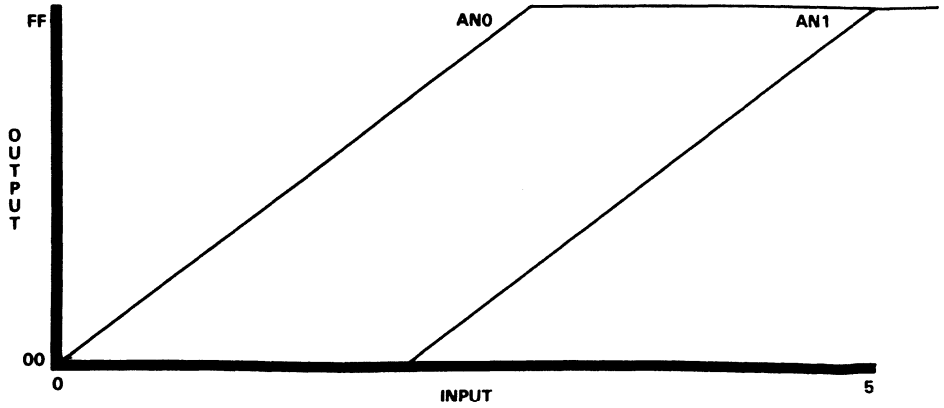


Figure 3-8. Transfer Characteristics of the Interface Circuit

Another technique used to increase the effective resolution is over-sampling. The digital output is determined by averaging several conversion results. The transition noise or uncertainty can be greatly reduced. For some applications, pseudo-random noise might be injected into the input and the average of many conversions computed to determine the digital output. The integral of the pseudo random noise is zero over a long period of time. When the pseudo noise is injected, the conversion result varies by some number of LSBs from a

nominal value (see Figure 3-9). The final average value depends on where the original input signal lies within the code width of the converter. If the input signal is not at the center of a code, the computed average will show either a negative or positive offset from the center.

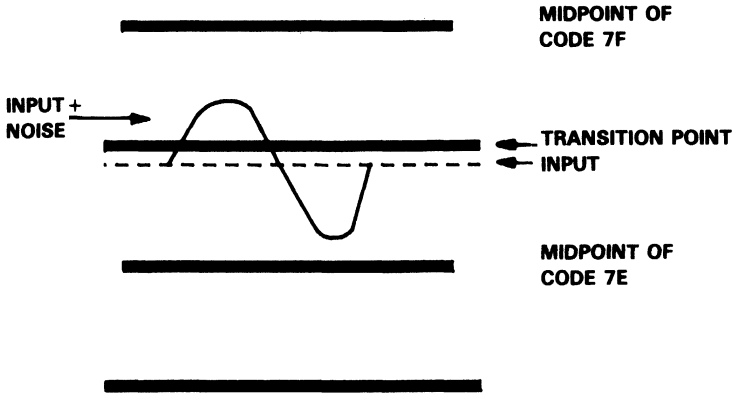


Figure 3-9. Injecting Noise into the Input Signal

Another technique used to increase effective resolution is the two-step subranging conversion. The A/D converter first generates the most significant eight bits of the digital value of the input signal. A fast, very-high-accuracy D/A converter uses the most significant six bits result (with the least significant bits set to zero) to generate a precise analog signal, which is then subtracted from the input. The difference is then amplified and digitized to provide the additional least significant bits. The accuracy of the result depends on the accuracy of the generated analog signal. (For more information about external A/D converters, refer to Section 5.3 and Appendix C.)

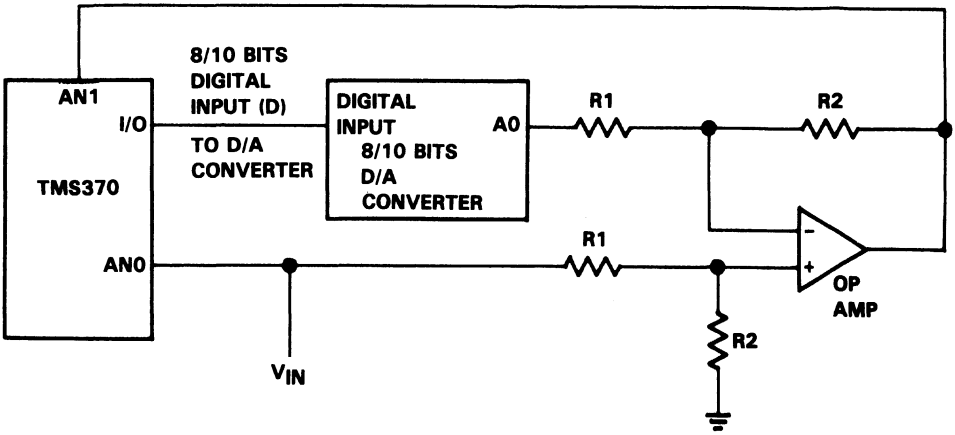


Figure 3-10. Block Diagram of Two-Step Subranging Conversion

3.5 Ratiometric Conversion

Ratiometric conversion is another way to obtain greater output resolution if the maximum of the input signal is less than V_{CC3}. In ratiometric conversions, the conversion result is the ratio of the reference voltage V_{REF} to the analog input signal. In other words, the absolute value of the analog input is of no particular concern but the ratio of the output to the full-scale value is important. The analog reference (maximum of the input signal) can be one of the analog input channels AN1 to AN7. This allows maximum full-scale utilization of the A/D converter. However, the absolute accuracy of the A/D converter is tested at V_{REF} equal to 5.1 V. The absolute accuracy will decrease when V_{REF} is below 5.1 V in the ratiometric conversion.

3.6 Sampling Frequency

Sampling frequency is the rate at which the conversions take place. This factor can greatly affect system performance. The application or ultimate use of the converted data determines the required sampling frequency.

Consider the following example of a case in which an analog input signal is sampled at a frequency much lower than the frequency of the actual signal. The resultant frequency is the "alias" of the original. Figure 3-1 illustrates the aliasing error caused from an insufficient number of samples.

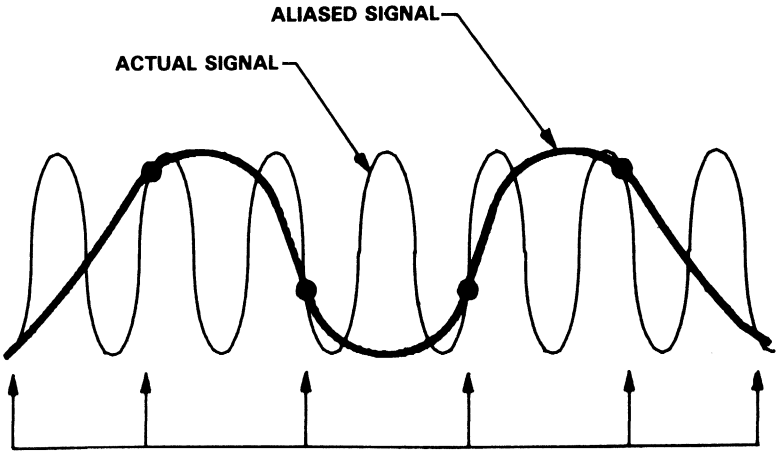


Figure 3-11. Aliasing Signal Caused by Inadequate Sampling Rate

When sampling an analog signal, the Nyquist criterion must be followed in order to reproduce the sampled data with no loss of information. The Nyquist criterion states that the sampling frequency must be greater than twice that of the highest frequency to be sampled.

On the other hand, sampling the input signal at much higher rate than its input frequency can reduce the system throughput due to poor CPU utilization. Users should choose the sampling frequency carefully to obtain an optimal solution.

The A/D converter takes 164 cycles to convert the analog input to a digital result. If the controller operates using a 20 MHz crystal (system clock frequency at 5 MHz), the conversion will take 32.8 μs. The A/D module allows a programmable sampling time depending on the system application. The user should allow 1 μs sampling time for each kilohm of source impedance or a minimum of 1 μs for a low impedance source. Assuming the analog source impedance is less than or equal to 1 kilohm for minimum sampling time (the sampling time is limited by the instruction cycle time to set up the SAMPLE START bit, the minimum sampling time is 1.6 μs using a 20 MHz crystal). In that case, the A/D converter can convert an analog input in every 34.4 μs or the maximum conversion rate would be 29069 conversions per second.

To meet the Nyquist criterion, the maximum frequency of the input signal must be limited to approximately 14 kHz.

In multi-sensor systems, the A/D converter uses time-multiplexing techniques to scan between inputs from various sensors. When these techniques are used the scan frequency must take into account of the number of channels that will

be acquiring, so that the A/D converter captures changes occurring at the fastest rate of interest for a given signal.

3.7 Analog Reference and Layout Considerations

We have discussed various techniques using signal conditioning and filtering to improve system accuracy. It is important to observe that no filter is justifiable as a substitute for proper attention to layout and shielding techniques. Rather, it is adjunct to them. Every effort should be made to keep noise out of the system. Filtering is added to the system only if it becomes necessary to clean up the remaining undesirable noise, especially that present in the original signal.

To minimize noise and digital clock coupling to an input which might be causing conversion errors, the lead to the analog input should be kept as short as possible. Furthermore, a low impedance shield between the noisy signals and the analog input signal can be used to block out the capacitor coupling effect.

Digital ground lines are usually quite noisy and have a large current spike. All analog grounds should be run separately from the digital ground line to make sure that there are no common impedance earth paths with digital ground or other circuits (as shown in Figure 3-12 and Figure 3-13). Analog ground should be connected to a low-impedance point near the power supply. During the conversion, current flow into the analog ground can be changed with a high-impedance in the ground line. Such changes can cause changes in voltage at the analog ground pin (V_{SS3}), and might cause conversion errors near the transition point.

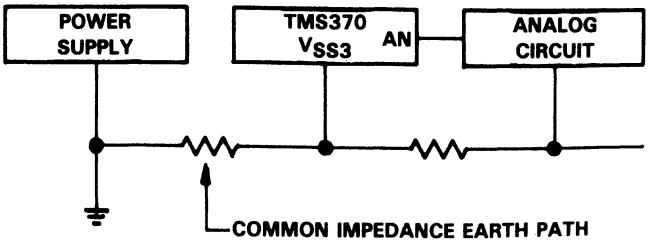


Figure 3-12. Circuit with Common Impedance Earth Path

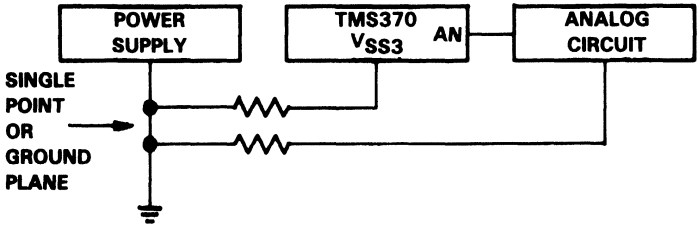


Figure 3-13. Circuit with No Common Impedance Earth Path

Supply transients should be prevented by good decoupling practice; that is, by having a decoupling capacitor close to the V_{CC3} and V_{SS3} pins. The reference voltage (V_{REF}) can also affect the conversion accuracy. It should be kept clean, well filtered, and used only by the A/D converter if possible. V_{REF} can be from 2.5 V to V_{CC3} + 0.1. However, it is important to note that the absolute accuracy is only tested at V_{REF} equal to 5.1 V, and as V_{REF} decreases, the LSB size decreases and the absolute error in term of the LSB may increase.

The source impedance (Z_{REF}) of V_{REF} (Figure 3-14) should not exceed the value specified in the electrical specification (24K ohms for CLKIN less than 12 MHz and 10K ohms for CLKIN higher than 12 MHz). During the conversion process, the reference voltage charges and discharges the capacitor array to determine the conversion value. If the reference voltage source impedance is too high, it will limit the current appropriately charging/discharging the capacitor array, and this will cause conversion errors.

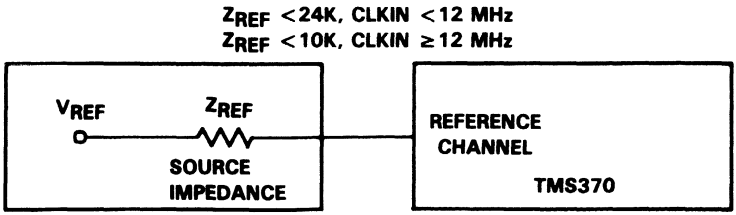


Figure 3-14. Reference Voltage Source Impedance

Section 4

Software Routines

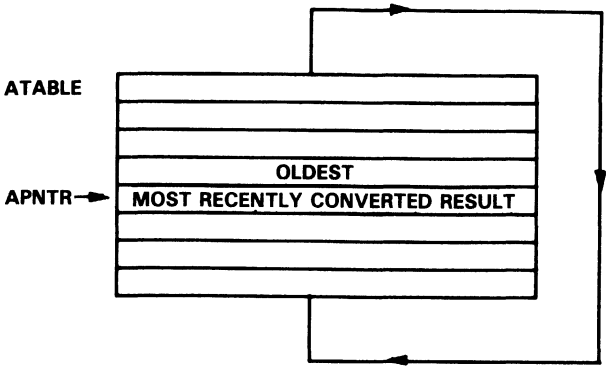
The following TMS370 software routine examples show various uses of the A/D module. The register-equate directives shown in Table 4-1 are common for all examples.

Table 4-1. Common Equate Table

```
ADCTL   .EQU    P070        ;ANALOG CONTROL REGISTER
ADSTAT  .EQU    P071        ;ANALOG STATUS AND INTERRUPT REGISTER
ADDATA  .EQU    P072        ;ANALOG CONVERSION DATA REGISTER
ADIN    .EQU    P07D        ;ANALOG PORT E DATA INPUT REGISTER
ADENA   .EQU    P07E        ;ANALOG PORT E INPUT ENABLE REGISTER
ADPRI   .EQU    P07F        ;ANALOG INTERRUPT PRIORITY REGISTER
```

4.1 Single Channel Continuous Conversion

The first program example performs a single channel conversion. The sampling frequency is controlled by using the on-chip timer, and the digital results are stored in a table beginning at ATABLE (8 bytes long). The conversions continue with the data updated in a round-robin fashion. APNTR is the pointer to the most recently converted result.



Channel Assignments:

ANALOG INPUT CHANNEL = AN0
REF CHANNEL = V_{CC3}

We have shown that the maximum sampling frequency is limited by the conversion rate of the A/D converter and the Nyquist criterion. With a crystal operating at 20 MHz, the maximum conversion rate is 29069 conversions per second, or the maximum frequency of the input signal according to Nyquist criterion is limited to approximately 14 kHz. However, this only shows the maximum conversions that the A/D module can handle. You should also consider the software overhead required to initiate a conversion and any processor loading that might affect how fast the conversion data will be processed.

This example routine sets up the timer to generate an interrupt at a rate of 10 kHz. The interrupt routine initiates an A/D conversion. That is, 1 conversion occurs for every 100 µs. Assuming the system clock cycle is 200 ns, the timer will be set to a period of 500 (01F4h) counts.

The following section sets up the table (ATABLE) and the control registers for the A/D converter.

```
          .REG    ATABLE,8          ;8 BYTE TABLE THAT STORES CONVERTED DATA
          .REG    APNTR             ;POINTER TO MOST RECENTLY CONVERTED DATA
T1C       .EQU    P043              ;LSB TIMER COMPARE REGISTER
T1CTL1    .EQU    P049              ;TIMER COUNTER CONTROL REG 1
T1CTL2    .EQU    P04A              ;TIMER COUNTER CONTROL REG 2
T1CTL3    .EQU    P04B              ;TIMER INTERRUPT CONTROL REG
T1CTL4    .EQU    P04C              ;TIMER COUNTER CONTROL REG 4
;
INIT      MOV     #0FEH,ADENA       ;ENABLE AN0 AS ANALOG CHANNEL
          MOV     #01H,ADSTAT       ;SET THE INTERRUPT ENABLE AND
                                    ;CLEAR FLAG
          MOV     #0A0H,B
          LDSP                      ;INITIALIZE STACK POINTER TO 0A0H
                                    ;CLEAR THE TABLE BEFORE CONVERSION
          MOV     #08H,B
          MOV     B,APNTR           ;SET POINTER TO FIRST BYTE
          CLR     A
INIT0     MOV     A,ATABLE-1(B)     ;CLEAR ALL EIGHT BYTES
          DJNZ    B,INIT0
```

The following section sets up the on-chip timer to control the sampling frequency. The conversion period is loaded into the Timer Compare register (T1C). When the counter (T1CNTR) matches the compare register, an interrupt request will be generated. The timer interrupt service routine will initiate an A/D conversion and set up the time for the next conversion in the compare register. For more detailed information about the on-chip Timer, the user is referred to the TMS370 Data Manual or the Timer application report.

```
;
;  SET UP THE TIMER COMPARE FUNCTION TO CONTROL THE SAMPLING FREQ
;
   MOV    #00H,T1CTL1          ;SET TIMER CLOCK TO SYSTEM CLOCK
   MOV    #090H,T1CTL4         ;SET TIMER TO CAPTURE/COMPARE MODE
                               ;SET COMPARE RESET ENABLE
   MOV    #HI(500-1),T1C-1     ;SETUP THE SAMPLING TIME IN COMPARE REGISTER
   MOV    #LO(500-1),T1C
   MOV    #01,T1CTL2           ;RESET TIMER TO ZERO
   MOV    #01,T1CTL3           ;ENABLE COMPARE 1 INTERRUPT
   EINT
;   .
;  MAIN PROGRAM
;   .
;  THE ANALOG INPUT SIGNAL IS SAMPLED AND CONVERTED
;  CONTINUOUSLY AT A RATE OF 10 KHZ.
;   .
;   .
```

The following section is the timer interrupt routine, it sets up the time for the next conversion in the compare register and initiates the A/D conversion. The address of the label T1SERV must be placed in the interrupt vector table located at 7FF4h and 7FF5h.

```
;
;         INTERRUPT ROUTINE FOR TIMER COMPARE
;
T1CINT    .DBIT   5,T1CTL3        ;NAMED TIMER1 COMPARE INTERRUPT FLAG
T1SERV    SBIT0   T1CINT          ;CLEAR INTERRUPT FLAG
SAMPLE    MOV     #040H,ADCTL     ;START SAMPLING (APPROX. 2uS DELAY
                                  ;FOR CLOCKIN = 20 MHZ)
          MOV     #0C0H,ADCTL     ;START CONVERSION
          RTI
```

The following section is the A/D interrupt routine, it saves the conversion results in the ATABLE and sets the pointer to the next available location. The address of the label ATOD must be placed in the interrupt vector table located at 7FECh and 7FEDh.

```
;
;         INTERRUPT ROUTINE FOR A/D CONVERTER
;
ADFLAG    .DBIT   1,ADSTAT        ;NAMED THE INTERRUPT FLAG AS ADFLAG
ATOD      PUSH    A               ;SAVE THE REGISTERS
          PUSH    B
          SBIT0   ADFLAG          ;CLEAR THE INTERRUPT FLAG
          MOV     APNTR,B         ;GET THE CURRENT POINTER
          MOV     ADDATA,A        ;GET THE CONVERSION RESULTS
          MOV     A,ATABLE-1(B)   ;SAVE THE RESULT IN THE TABLE
          DJNZ    APNTR,EXITAD    ;CHECK FOR WRAP AROUND
          MOV     #08H,APNTR      ;START FROM LOCATION ATABLE(7)
EXITAD    POP     B               ;RESTORE REGISTER
          POP     A
          RTI
;
;         INIT INTERRUPT VECTORS
          .SECT   "vect",7FECH
          .WORD   ATOD,0,0,0,T1SERV,0,0,0,0,INIT
```

4.2 Multiple Channels Conversion

The second example program samples and converts data from four channels, each of which uses a different channel for reference input. The program stores the results in a table beginning at ATABLE. The routine stops interrupting the main program after it finishes all four channels. If the main program wants more recent data, it only needs to execute the code SAMPLE and the routine will again sample and convert all four channels of data. The A/D interrupt enable bit is cleared by the A/D interrupt routine as a signal to the main program that all four channels have been processed. The address of the label ATOD must be placed into the interrupt vector table located at 7FECh and 7FEDh.

ANALOG INPUT CHANNEL	REF CHANNEL
AN3	AN7
AN2	AN6
AN1	AN5
AN0	AN4

```
.REG    ADCHANL                 ;KEEP CURRENT CHANNEL NUMBER
.REG    ATABLE,4                ;4 BYTE TABLE THAT STORES CHANNEL DATA

INIT    MOV     #00H,ADENA      ;ENABLE AN0 - AN7 AS ANALOG CHANNEL
        MOV     #CA0H,B
        LDSP                    ;INITIALIZE STACK POINTER
;
;   INITIALIZE THE TABLE FOR CONVERSION RESULTS
;
        CLR     A
        MOV     #04,B           ;INIT THE TABLE
INIT0   MOV     A,ATABLE-1(B)
        DJNZ    B,INIT0
        EINT                    ;ENABLE INTERRUPTS
        CALL    SAMPLE          ;SAMPLE ALL THE DATA
;
;          .
;          .
;       MAIN PROGRAM
;          .
;          .
;       CHECK THE CONVERSION COMPLETED BEFORE USING THE DATA
;
WAITC   BTJO    #01H,ADSTAT,WAITC
;
;       ALL CONVERSIONS HAVE BEEN DONE, RESULTS ARE READY
;       READ DATA HERE
;          .
;          .
        CALL    SAMPLE          ;SAMPLE ANOTHER SET OF DATA
;          .
;          .
```

The following section is the subroutine to initiate the first A/D conversion. When the conversion is completed, an interrupt request will be generated. Subsequent conversions will be driven by the interrupt routine.

```
;
;   SUBROUTINE SECTION
;
SAMPLE  MOV     #3BH,ADCHANL        ;RESET THE CHANNEL SELECTION FOR
                                    ;NEW SET OF CONVERSION
        MOV     #01H,ADSTAT         ;ENABLE THE INTERRUPT AND CLEAR
                                    ;ANY FLAGS
        MOV     #07BH,ADCTL         ;START SAMPLING (APPROX. 2uS DELAY
                                    ;FOR CLOCKIN = 20 MHZ)
        MOV     #0FBH,ADCTL         ;START CONVERSION
        RTS
```

The following section is the A/D interrupt routine. It saves the conversion result in the ATABLE and initiates another conversion if not all four channels have been processed.

```
;
;   INTERRUPT ROUTINE FOR A/D CONVERTER
;
ATOD    PUSH    A                   ;SAVE THE REGISTERS
        PUSH    B
        MOV     #01,ADSTAT          ;CLEAR THE INTERRUPT FLAG
        MOV     ADCHANL,B           ;GET THE CURRENT CHANNEL NUMBER
        AND     #07H,B              ;GET ANALOG INPUT CHANNEL ONLY
        INC     B
        MOV     ADDATA,A            ;GET THE CONVERSION RESULTS
        MOV     A,ATABLE-1(B)       ;SAVE THE RESULT IN THE TABLE
        DJNZ    B,NEXTCON           ;GO TO NEXT CONVERT
ENDCON  AND     #0FEH,ADSTAT        ;CLEAR THE INTERRUPT ENABLE
                                    ;TO SIGNAL THE END OF 4 CONVERSION
        JMP     EXITAD
NEXTCON SUB     #09H,ADCHANL        ;SET THE NEXT REFERENCE CHANNEL AND
                                    ;ANALOG INPUT CHANNEL
        MOV     ADCHANL,ADCTL       ;SET UP INPUT AND REF CHANNEL
        OR      #40H,ADCTL          ;START SAMPLE DATA
        OR      #0E0H,ADCTL         ;START CONVERSION
EXITAD  POP     B                   ;RESTORE REGISTER
        POP     A
        RTI
;
;   INIT INTERRUPT VECTORS
        .SECT   "vect",7FECH
        .WORD   ATOD,0,0,0,0,0,0,0,0,INIT
```

The above examples illustrate two basic operations of analog-to-digital conversion. The first uses the TMS370 timer to control the sampling frequency of conversions, and the second example illustrates multiple channels conversion, that is, multiple input and reference sources.

The routines can be easily extended to multiple channels conversion with the on-chip timer controlling the sampling frequency. In some cases, the user may even want different sampling frequencies for different channels to account for any disparity in the frequencies of the input signals.

One way to achieve this is to set the time base (output compare function) to the period of the fastest sampling frequency. The sampling frequency of slower input signals will be a multiple of this time base. Additional registers may be allocated to indicate the number of timer interrupts that might have occurred since the last conversion of a particular signal (slow input signal). The interrupt routine will determine whether single or multiple conversions will be initiated.

Section 5

Application Examples

The following section shows some A/D conversion applications using the TMS370 family microcontrollers. All hardware is tested only under specific conditions. The user should take all standard precautions when using these circuits in their respective applications.

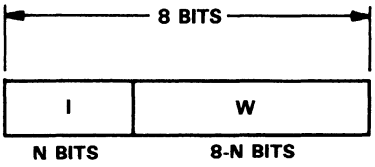
5.1 Data Translation

Many applications involve monitoring physical parameters. Temperature, force, pressure, position and other physical parameters must be translated before they can be processed by the microcontroller. Physical parameters are first transformed to analog signals (voltage, current) by transducers. These analog signals are then converted to digital data. However, one quickly discovers that most of the transfer functions between the physical parameters and the digital output are nonlinear. Calculating the value of the physical parameters from the digital output may be time consuming and would severely limit the system throughput.

One way to simplify the interpretation of the converted data is to linearize the analog input before the conversion. Signal conditioning amplifiers, log amplifiers and other linear circuit techniques can be used. However, analog linearization may not be cost-effective or possible for certain applications. Also, analog components suffer aging (gain, offset drift over time) and tolerance problems that can affect system accuracy. Alternatives such as table lookup techniques or linearization algorithms might reduce the need for expensive hardware linearization.

The values of physical parameters can be calculated beforehand and stored in a table. Upon conversion completion, the application software would simply retrieve the value of the parameter by using the conversion result as the index to the table.

Instead of code-by-code conversion, it is also possible to interpret all 256 discrete values (00-FF) with a table of fewer than 256 entries. Values of the function between table values can be determined by interpolation techniques. For example, the conversion output can be split into two fields, the upper N bits are used as an offset to retrieve data from the table, the lower 8-N bits are used as the weighting factor for interpolation. The value of any conversion result can be expressed as:



$$F(I.W) = F(I) + \frac{W}{2^{8-N}}[F(I+1) - F(I)]$$

The following program example uses the result of the conversion and the interpolation technique to calculate the value of the physical parameter. The table is 33 bytes long starting at location ATABLE. The most significant five bits of the conversion result are used as the index to the table, whereas the least significant three bits are used as the weighting factor.

$$F(I.W) = F(I) + W/8[F(I+1) - F(I)]$$

Assuming the conversion result is 01100010 (98), the value of the physical parameter can be calculated by the following equation:

$$F(01100.010) = F(01100) + 2/8[F(01101) - F(01100)]$$

```
        .REG    ATABLE,33           ;33 BYTES TABLE
        .REG    RESULT              ;REGISTER FOR FINAL RESULT
        .REG    ATPNT               ;TEMPORARY REGISTER
;
;
BEGIN   PUSH    A                   ;SAVE REG A
        PUSH    B                   ;SAVE REG B
        MOV     ADDATA,ATPNT        ;SAVE THE CONVERSION RESULT
        MOV     ATPNT,B
        SWAP    B                   ;GET THE INDEX FIELD
        RL      B
        AND     #1FH,B
;
;  GET THE VALUE FROM THE TABLE
;
        MOV     ATABLE(B),A         ;GET F(I)
        MOV     A,RESULT
;
;  CHECK IF INTERPOLATION NECESSARY
;  IF THE MOST LEAST SIGNIFICANT THREE BITS ARE ZERO, NO
;  INTERPOLATION IS NECESSARY

        BTJO    #07H,ATPNT,INTERP
        JMP     FINISH              ;
INTERP  INC     B                   ;SET INDEX POINT TO NEXT ENTRY
        MOV     ATABLE(B),A         ;GET F(I+1)
        SUB     RESULT,A            ;CALCULATE THE DIFFERENCE
                                    ;F(I+1) - F(I)
        AND     #07H,ATPNT          ;GET THE WEIGHTING FACTOR
        MPY     ATPNT,A             ;W * [F(I+1) - F(I)]
                                    ;RESULT STORE IN A:B
        MOV     #08,ATPNT
        DIV     ATPNT,A             ;DIVIDE A:B BY 8
        ADD     A,RESULT            ;F(I) + INTERPOLATION VALUE
FINISH  POP     B                   ;RESTORE REGISTERS A AND B
        POP     A
        RTS
```

TMS370 microcontrollers contain on-chip Data EEPROM which provides an excellent area to implement the translation table. With the on-chip EEPROM capability, the translation table can be ad, ;ted for correction as environmental conditions change. Also, the Write Protection feature of the Data EEPROM can be used to protect the translation table from inadvertent overwriting by the application software. For more detailed information about the on-chip Data EEPROM, refer to Section 6 of the TMS370 Data Manual or the EEPROM Application Report.

5.2 Temperature Sensor Interface

A typical temperature measurement application is shown in Figure 5-1. The main principle of this example applies to most other input transducers. The interfacing circuitry consists of a bridge amplifier detecting the resistance variation over the temperature range.

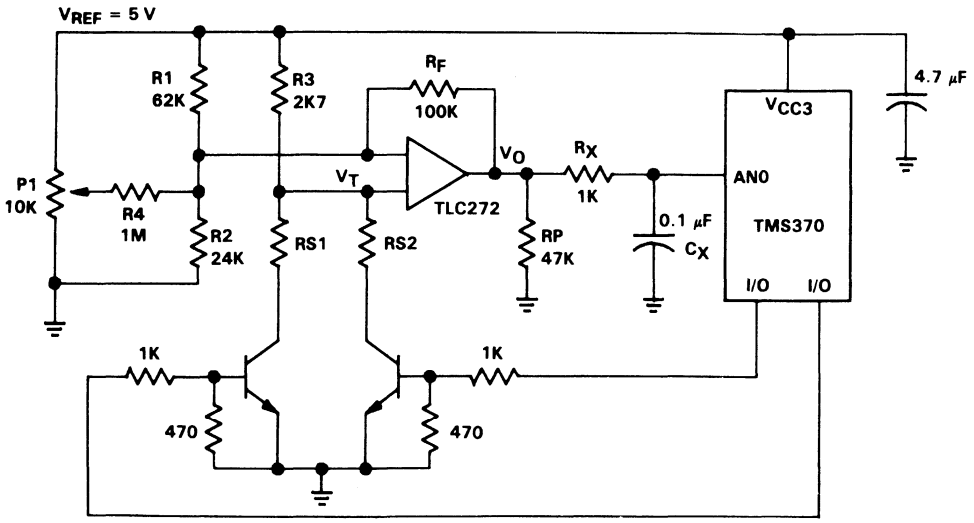


Figure 5-1. Temperature Sensor Interface

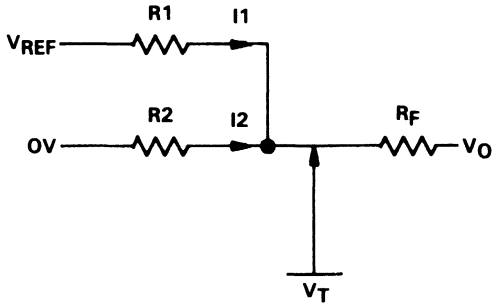
The bridge is comprised of resistors R1, R2, R3, and the temperature sensor (either RS1 or RS2). The differential output voltage of the bridge is forced to zero by the feedback connection. The circuit is configured as a current amplifier.

Potentiometer P1 and resistor R4 are used to adjust any offset present in the components.

Assuming the transistor turn-on resistance is negligible compared to Rs, then

$$Vt = V_{REF}\,[R_s \div (R_3 + R_S)] \text{-------- (1)}$$

The circuit can be analyzed using the virtual ground technique.



$I_1 = [V_{REF} - V_T] \div R_1$;

$I_2 = -V_T \div R_2$;

$I_1 + I_2 = -[V_O - V_T] \div R_F$

Therefore

$V_O = V_T - R_F(I_1 + I_2)$

$V_O = V_T + R_F[(V_T \div R_2) - (V_{REF} - V_T) \div R_1]$ -------(2)

Rs is a positive temperature coefficient silicon sensor approximately 0.8 % per degree C at 25 degrees C. Its nominal resistance at 25 degrees C is 1K. Resistor R3 is chosen to linearize the exponential temperature coefficient of Rs.

The temperature sensor interface is required to convert the temperature from 0 - 100 degrees C (Rs=850 ohm to 1700 ohm) to output ranging from 0 V to 5 V. A reasonable value of Rf (100K) is chosen. R1 and R2 are then determined by substituting the conditions of temperature at 0 and 100 degrees C to equations (1) and (2).

Rp is a non-critical pull-down resistor; it is used at the output of the op-amp for best amplifier linearity near 0 V. R_X and C_X form a low pass filter for inexpensive noise filtering.

5.3 Automatic Ranging Interface

The following case is an example of auto-ranging interface circuitry. The circuit has a total of four gain ranges which can be easily extended to more if desired. The gain ranges are 1, 2, 4 and 8. A/D resolution is effectively improved at lower voltage ranges.

The ranging is done by changing the amplification (resistance at the non-inverting terminal) of the non-inverting amplifier (TLC272). The actual gain of the amplifier is greatly depending on the accuracy of the resistors. Usually additional variable resistors are used to adjust the gain of the amplifier. However, if the exact gain of the amplifier at each range is calibrated and stored in the Data EEPROM, these manual adjustments can be avoided. The conversion

result is then based on the calibration gain to calculate its actual value. For applications requiring high accuracy, the application program can calibrate the gain value at multiple locations in each range.

Two voltage comparators (LM339) are used to provide the lower and higher trip points for ranging. Two analog input pins (AN6, AN7) are configured as general purpose input pins to determine whether the input signal is within the trip points. It is important to leave some margin between the lower (higher) trip points and the minimum (maximum) of the output of the amplifier, such that the amplifier output will not clip at its mimimum (maximum) value during the A/D sampling phase. For cost sensitive applications, the user may use the A/D module itself instead of the voltage comparators to determine the input signal range. However, three additional conversions (98.4 us at 20 MHz) may be required in the worst case.

Two output pins (INT2, INT3) are used to select the desired gain factor of the amplifier.

INT2	INT3	GAIN FACTOR
0	0	1
0	1	2
1	0	4
1	1	8

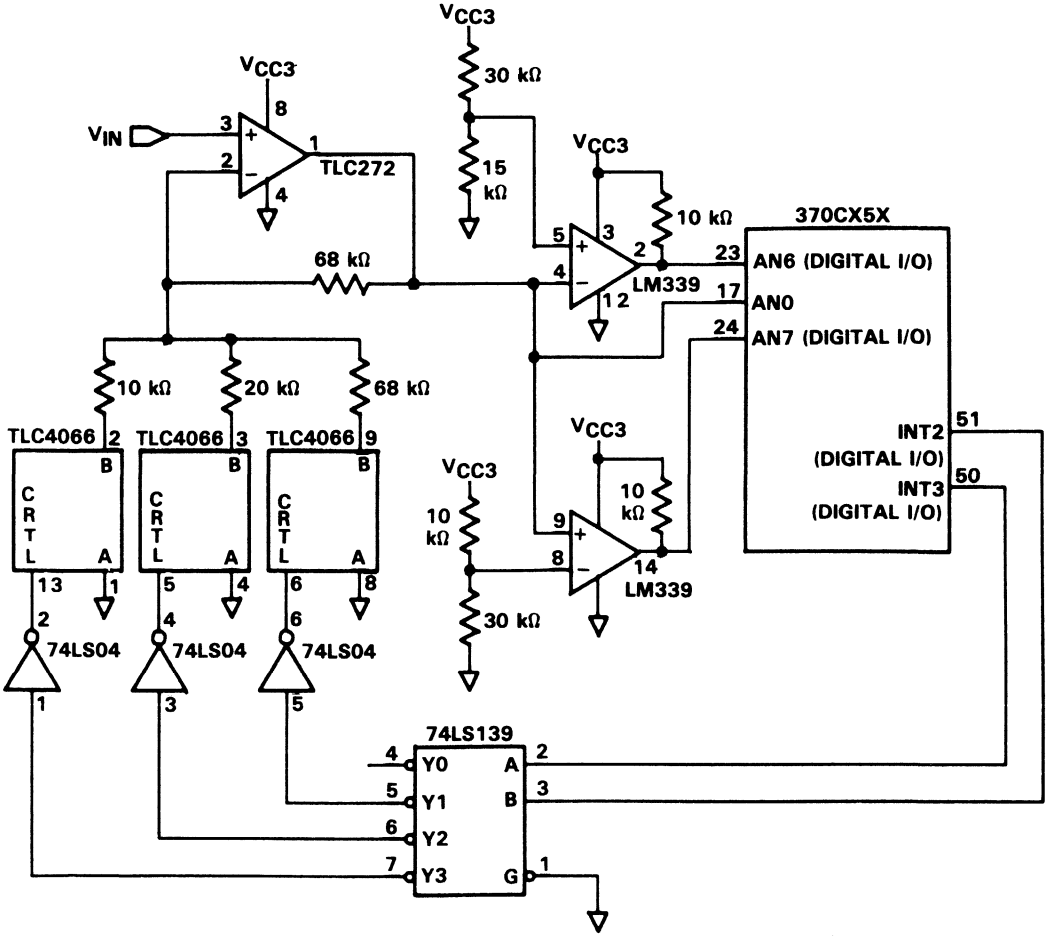


Figure 5-2. Auto-Ranging Circuit Diagram

The following example shows the software required to control the interface.

```
;
;       ANALOG INPUT CHANNEL            REF CHANNEL
;              AN0                          VCC3
;
;    AN6   -   GENERAL PURPOSE INPUT PIN  (DETERMINE GAIN RANGE)
;    AN7   -   GENERAL PURPOSE INPUT PIN  (DETERMINE GAIN RANGE)
;    INT2  -   GENERAL PURPOSE OUTPUT PIN (SELECT GAIN RANGE)
;    INT3  -   GENERAL PURPOSE OUTPUT PIN (SELECT GAIN RANGE)
;
INT2    .EQU     P018              ;INT2 PIN CONTROL REGISTER
G1      .DBIT    3,INT2            ;GAIN FACTOR CONTROL BIT 1
INT3    .EQU     P019              ;INT3 PIN CONTROL REGISTER
G0      .DBIT    3,INT3            ;GAIN FACTOR CONTROL BIT 0
;
;
;
;
        .REG     RESERVE,10
;
;    RESULT-1 : INDICATE THE INPUT SIGNAL RANGE (GAIN FACTOR)
;    RESULT : CONVERSION RESULT
;
        .REGPAIR RESULT            ;16 BITS REGISTER FOR CONVERSION RESULT
        .REGPAIR GAIN              ;TEMP REG
        .TEXT    7000H
;
;
;
INIT    MOV      #0FEH,ADENA       ;ENABLE AN0 AS ANALOG CHANNELS
                                   ;AN1 - AN7 AS GENERAL PURPOSE
                                   ;INPUT PINS
        MOV      #10H,INT3         ;SET INT3 PIN AS GENERAL PURPOSE
                                   ;OUTPUT PIN
        MOV      #10H,INT2         ;SET INT2 PIN AS GENERAL PURPOSE
                                   ;OUTPUT PIN
;
        MOV      #20H,A            ;OPTIONAL - NOT NECESSARY IF
                                   ;ENOUGH TIME BETWEEN THE LAST INSTR
                                   ;AND THE FIRST SAMPLE
INIT0   DJNZ     A,INIT0           ;WAIT UNTIL OP AMP IS STABLE
;
;
        MOV      #0A0H,B
        LDSP                       ;INITIALIZE STACK POINTER
        MOVW     #0,RESULT         ;INITIALIZE THE REGISTER
                                   ;INITIAL GAIN FACTOR EQUAL TO 1
        EINT                       ;ENABLE INTERRUPT
;
;         .
;         .
;       MAIN PROGRAM
;         .
;         .
;         .
AGAIN2  CALL     SAMPLE            ;SAMPLE ANOTHER SET OF DATA
WAIT2   BTJZ     #04H,ADSTAT,WAIT2 ;CHECK THE 'AD READY' BIT
;         .
;         .
```

The following section is the subroutine to initiate the A/D conversion. The subroutine first read in the output of the comparators (via AN6 and AN7) to determine the input voltage range. If the input signal is within the desired range, then an A/D conversion will be initiated. Otherwise, the subroutine will adjust the gain factor and repeat the process one more time.

```
;
;   SUBROUTINE SECTION
SAMPLE PUSH     A
UPPER  MOV      ADIN,A
       BTJO     #80H,A,LOWER        ;IS THE INPUT SIGNAL EXCEEDS THE
                                    ;UPPER LIMIT
       CMP      #0,RESULT-1         ;IS THE GAIN FACTOR ALREADY SET TO
                                    ;MIN GAIN
       JEQ      CONVRT
       DEC      RESULT-1            ;SET TO LOWER GAIN FACTOR
       SBIT0    G0
       BTJZ     #1,RESULT-1,WAIT
       SBIT0    G1
       SBIT1    G0
       JMP      WAIT
LOWER  BTJO     #40H,A,CONVRT       ;IS THE INPUT SIGNAL BELOW THE
                                    ;LOWER LIMIT
       CMP      #3,RESULT-1         ;IS THE GAIN FACTOR ALREADY SET TO
                                    ;MAX GAIN
       JEQ      CONVRT
       INC      RESULT-1            ;SET TO HIGHER GAIN FACTOR
       SBIT1    G0
       BTJO     #1,RESULT-1,WAIT
       SBIT0    G0
       SBIT1    G1
WAIT   MOV      #10,A               ;SET COUNT
LOOP   DJNZ     A,LOOP              ;WAIT FOR 20 uS UNTIL THE OP-AMP
                                    ;IS STABLE
       JMP      UPPER
CONVRT MOV      #01H,ADSTAT         ;ENABLE THE INTERRUPT AND CLEAR
                                    ;ANY FLAGS
       MOV      #040H,ADCTL         ;START SAMPLING (APPROX. 2uS DELAY
                                    ;FOR CLOCKIN = 20 MHZ)
       MOV      #0C0H,ADCTL         ;START CONVERSION
       POP      A
       RTS
```

The following section is the A/D interrupt routine. It saves the conversion result in the register RESULT.

```
;
;   INTERRUPT ROUTINE FOR A/D CONVERTER
ATOD   MOV      #01,ADSTAT          ;CLEAR THE INTERRUPT FLAG
       MOV      ADDATA,RESULT       ;SAVE THE CONVERSION RESULTS
       RTI                          ;
;
;   INIT INTERRUPT VECTORS
       .SECT    "vect",7FECH
       .WORD    ATOD,0,0,0,0,0,0,0,0,INIT
                                    ;
                                    ;
```

5.4 Interfacing a Serial A/D Converter with TMS370 Family Microcontrollers

Sections 5.4.1 and 5.4.2 demonstrate the interface between a 10-bit serial A/D converter (TLC1540/1) and TMS370. This will be useful for those who want to use TMS370X10 but still need A/D, or those systems that require high accuracy (down to 5 mV resolution) and better isolation of the analog system from the relatively noisy digital controller.

The TLC1540 and TLC1541 are both 10-bit, 11 channels serial A/D converter with sample-and-hold circuitry. TLC1540 has +/- 0.5 LSB error whereas TLC1541 has +/- 1 LSB error. The serial A/D converter has four control inputs: Chip Select (CS), Address input, I/O clock and System Clock. The first example uses the on-chip SPI to interface with the Serial A/D, whereas the second example uses software routines to interface with the Serial A/D.

5.4.1 Using On-Chip Serial Peripheral Interface SPI

Figure 5-3 shows the circuit diagram of the interface between TLC1540/1 and TMS370. This section describes the interface of a 10-bit serial A/D converter through the Serial Peripheral Interface (SPI). The System Clock of the TLC1540/1 is provided by the CLKOUT pin of the TMS370 (Note the maximum TLC1540/1 system clock frequency is only 2.1 MHz, additional frequency divider/counter is required or use a 8.4 MHz crystal for TMS370).

The serial A/D receives the I/O clock 500 ns after (delay by the dual D flip flops as shift register) the SPICLK is active, this ensure enough set up time for the channel address. The conversion cycle takes 44 TLC1540/1 System Clock cycles and is initiated on the tenth falling edge of the I/O clock.

The following example program converts data from all 11 channels consecutively. It assumes that a TMS370 using an 8.4 MHz crystal, i.e., 2.1 MHz for CLKOUT. If the application program requires different system clock rates or I/O transmission clock rate, you must insure that the time between executing the instruction at label TRAN8 (initiate the conversion) and TRAN2 (transmitting the next channel address) is greater than the time transmitting 8-bit data plus 44 TLC1540/1 System clock cycles.

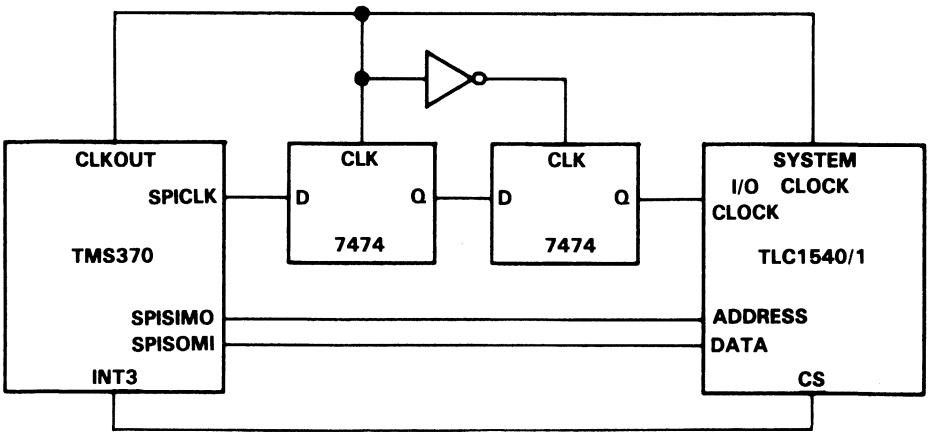


Figure 5-3. Interfacing Circuit Using SPI

This example program converts data from all 11 channels and stores the digital results in a table beginning at ATABLE. The table contains 11 16-bit registers. The least significant byte is located at the lower address. The routine stops interrupting the main program after it finishes all 11 channels. If the main program wants more recent data it needs only to execute the code at RESTART and the SPI routine will again transmit the channel address to the serial A/D (TLC1540/1) and receive data from the A/D. The flag CNVCMPL is set by the SPI routine as a signal to the main program that all 11 channels have been processed. The address label SPIINT must be placed in the interrupt vector table located at 7FF6h and 7FF7h.

```
;    SPISIMO - SPI FUNCTIONAL PIN, (CONNECT TO TLC1540/1 ADDRESS INPUT)
;    SPISOMI - SPI FUNCTIONAL PIN, (CONNECT TO TLC1540/1 DATA OUTPUT)
;    SPICLK  - SPI FUNCTIONAL PIN, (CONNECT TO TLC1540/1 I/O CLOCK)
;    CLKOUT  - SYSTEM CLKOUT, (CONNECT TO TLC1540/1 SYSTEM CLOCK)
;    INT3    - GENERAL PURPOSE OUTPUT PIN (CONNECT TO TLC1540/1 CHIP
;              SELECT)
;
SPICCR   .EQU   P030              ;SPI CONFIGURATION CONTROL REG
SPICTL   .EQU   P031              ;SPI CONTROL REGISTER
SPIBUF   .EQU   P037              ;RECEIVE DATA BUFFER REGISTER
SPIDAT   .EQU   P039              ;SERIAL DATA REGISTER
SPIPC1   .EQU   P03D              ;SPI PIN CONTROL 1
SPIPC2   .EQU   P03E              ;SPI PIN CONTROL 2
SPIPRI   .EQU   P03F              ;SPI PRIORITY CONTROL
DPORT2   .EQU   P02C              ;DPORT 2, CLKOUT CONFIGURATION REG
INT3     .EQU   P019              ;INT3 PIN CONTROL REGISTER
         .REG   ATABLE,22         ;16 BIT REGISTERS FOR CONVERSION RESULT
         .REG   FLAGS             ;REG FLAG
TRANSL   .DBIT  0,FLAGS           ;INDICATE MSB OR LSB TRANSMISSION
CNVCMPL  .DBIT  1,FLAGS           ;CONVERSIONS COMPLETE
         .REG   ADCHANL
         .TEXT  7000H
```

The following section sets up the Serial Peripheral Interface (SPI) for communication. The SPI is configured as the MASTER processor to control the communication. For more detailed information about the on-chip SPI, the user is referred to the TMS370 Data Manual or Using the TMS370 SPI and SCI Modules Application Report.

```
;
;    SET UP SPI CONFIGURATION
;
INIT    MOV     #087H,SPICCR        ;INITIALIZES SPI CIRCUITRY
                                    ;SELECT CLOCK POLARITY INACTIVE LOW
                                    ;SELECT BIT RATE = CLKIN/8
                                    ;SELECT CHARACTER LENGTH = 8
        MOV     #07H,SPICTL         ;CONFIGURE AS MASTER
                                    ;TRANSMISSION ENABLE, TALK = 1
                                    ;INTERRUPT ENABLE
        MOV     #02H,SPIPC1         ;SET SPICLK AS FUNCTION PIN
        MOV     #22H,SPIPC2         ;SET SPISOMI AND SPISIMO AS
                                    ;FUNCTION PIN
        MOV     #20H,SPIPRI         ;SET EMULATOR SUSPEND BIT
;
        MOV     #18H,INT3           ;SET INT3 AS OUTPUT PIN
        MOV     #08H,DPORT2         ;SET CLKOUT AS FUNCTIONAL PIN
;
        MOV     #0A0H,B
        LDSP                        ;INITIALIZE STACK POINTER TO 0A0H
;
        CLR     A
        MOV     #22,B
AGAIN   MOV     A,ATABLE-1(B)       ;INITIALIZE THE TABLE
        DJNZ    B,AGAIN
        EINT                        ;ENABLE INTERRUPT
LOOP    CALL    RESTART             ;START CONVERSIONS
;       .
;       .
;       .
;
;    CHECK CNVCMPL BIT IF ALL 11 CONVERSIONS DONE
;
WAIT    BTJZ    #02H,FLAGS,WAIT
;
;    ALL CONVERSIONS DONE, DATA ARE READY
;       .
;       .
;       .
;
;       MAIN PROGRAM GOES HERE
;
;       .
;       .
;       .
;    NEED MORE RECENT DATA
        CALL    RESTART             ;START TAKING MORE DATA
;
;       MORE MAIN PROGRAM
```

The following section is the subroutine to initiate the transmission. When the transmission is completed, an interrupt request will be generated. Subsequent transmissions will be driven by the interrupt routine.

```
;
;    SUBROUTINE SECTION
RESTART         CLR     ADCHANL         ;INITIALIZE CHANNEL ADDRESS
                CLR     FLAGS           ;CLEAR ALL FLAGS
                MOV     #01H,SPICCR     ;SET CHARACTER LENGTH TO 2
                MOV     #10H,INT3       ;ACTIVATE TLC1540/1 CHIP SELECT
                MOV     #00H,SPIDAT     ;TRANSMIT THE CHANNEL ADDR
                RTS
```

The following section is the SPI interrupt routine. It saves the previous conversion result in ATABLE and initiate another transmission until all 11 channels have been processed.

```
;
;   INTERRUPT ROUTINE FOR SPI
SPIINT PUSH    A                ;SAVE REGISTERS
       PUSH    B
       MOV     SPIBUF,A         ;GET THE CONVERSION RESULT AND CLEAR
                                ;INTERRUPT FLAG
       MOV     ADCHANL,B        ;GET CHANNEL NUMBER
       JZ      NOST0            ;DO NOT DECREMENT IF THIS IS CHANNEL 0
       DEC     B                ;GET CHANNEL NUMBER FOR RECEIVING DATA
       RL      B                ;MULTIPLY BY 2
NOST0  BTJO    #01H,FLAGS,CMPLT ;CHECK IF ALL 10 BITS DATA RECEIVED
;
;   SAVE THE MSB 2 BITS RESULT AND
;   INITIATE THE TRANSMISSION OF THE LAST 8 BITS RESULT
;
       MOV     #07H,SPICCR      ;SET THE CHARACTER LENGTH TO 8
;
; THE MOST SIGNIFICANT 2 BITS ARE LEFT OVER FROM FROM PREVIOUS TRANSMISSION
; THEY ARE THE LEAST 2 SIGNIFICANT BITS OF THE CHANNEL ADDRESS
;
TRAN8  MOV     A,SPIDAT         ;INITIATE TRANSMISSION
       AND     #03H,A           ;GET THE LAST 2 BITS ONLY
       MOV     A,ATABLE+1(B)    ;STORE THE MOST SIGNIFICANT 2 BITS
NOST   INC     FLAGS            ;SET THE FLAG INDICATE THE
                                ;LSB RESULT ALREADY RECEIVED
       JMP     EXITSP
CMPLT  MOV     A,ATABLE(B)      ;STORE THE LEAST SIGNIFICANT 8 BITS
NOST1  CMP     #0BH,ADCHANL     ;CHECK IF ALL CONVERSIONS DONE
       JNZ     GOCONVT
       MOV     #18H,INT3        ;DESELECT TLC1540/1 CHIP SELECT
       SBIT1   CNVCMPL          ;INDICATE ALL CONVERSIONS COMPLETED
       JMP     EXITSP
;
;   INITIATE MORE CONVERSION
GOCONVT
       INC     ADCHANL          ;POINT TO NEXT CHANNEL
       MOV     ADCHANL,B
       SWAP    B                ;LEFT JUSTIFY THE CHANNEL ADDR
       MOV     #01H,SPICCR      ;SET CHARACTER LENGTH TO 2
TRAN2  MOV     B,SPIDAT         ;INITIATE ANOTHER TRANSMISSION
       CLR     FLAGS            ;CLEAR THE FLAG INDICATE THE
                                ;CHANNEL ADDRESS ALREADY TRANSMITTED
EXITSP POP     B                ;RESTORE THE REGISTERS
       POP     A
EXIT   RTI
;
;
;   INIT INTERRUPT VECTORS
           .SECT  "vect",7FECH
           .WORD  0,0,0,0,0,SPIINT,0,0,0,INIT
;
;
```

5.4.2 Using Software Routine Interfacing with Serial A/D Converter

This section demonstrates the interface of TLC1540 through software routines. This will be useful for cost sensitive applications that need to minimize external hardware.

Four general purpose I/O pins are used to interface with the TLC1540. The following software example performs the same function as in Section 5.4.1. without any additional hardware. It converts data from all 11 channels and stores the digital results into a table beginning at ATABLE. The table contains 11 16-bit registers. The least significant byte is located at the lower address. The routine stops interrupting the main program after it finishes all 11 channels. If the main program wants more recent data it needs only to execute the code at CONVRT. Figure 5-4 shows the interconnection between TMS370 and TLC1540.

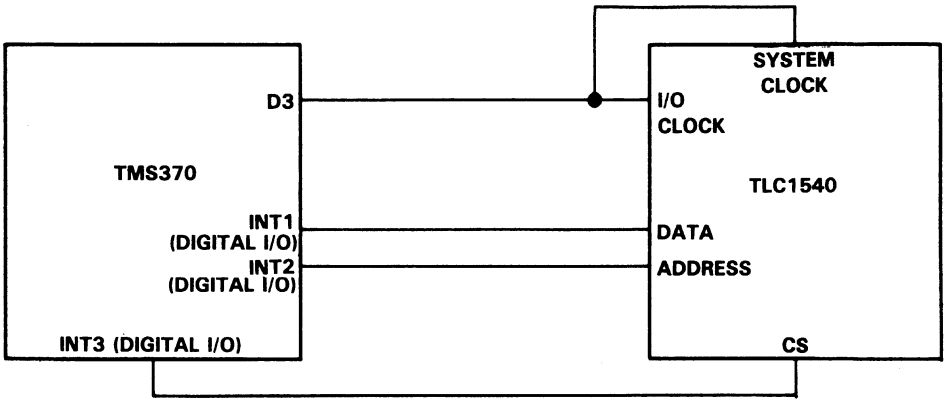


Figure 5-4. Interfacing Circuit Using Software Routine

```
;
;
;  D3/CLKOUT - GENERAL PURPOSE OUTPUT PIN, (CONNECT TO TLC1540/1
;              I/O CLOCK AND TLC1540/1 SYSTEM CLOCK)
;  INT1      - GENERAL PURPOSE INPUT PIN (CONNECT TO TLC1540/1
;              DATA OUTPUT)
;  INT2      - GENERAL PURPOSE OUTPUT PIN (CONNECT TO TLC1540/1
;              ADDRESS INPUT)
;  INT3      - GENERAL PURPOSE OUTPUT PIN (CONNECT TO TLC1540/1
;              CHIP SELECT)
;
DPORT1 .EQU     P02C              ;DPORT 1, CLKOUT CONFIGURATION REG
DPORT2 .EQU     P02D              ;DPORT 2, CLKOUT CONFIGURATION REG
DDATA  .EQU     P02E              ;DPORT DATA REG
DDIR   .EQU     P02F              ;DPORT DATA DIR REG
INT1   .EQU     P017              ;INT1 PIN CONTROL REGISTER
INT2   .EQU     P018              ;INT2 PIN CONTROL REGISTER
INT3   .EQU     P019              ;INT3 PIN CONTROL REGISTER
       .REG     RESERVE,10
       .REG     ATABLE,22         ;16 BIT REGISTERS FOR CONVERSION RESULT
       .REGPAIR RESULT,2          ;TEMPORARY RESULT REGISTER
       .REG     FLAG              ;REG FLAG
       .REG     ADCHANL
       .REG     BITCNT
       .REG     CHNLCNT
IOCLK  .DBIT    3,DDATA           ;TLC1540 SYSTEM CLOCK
                                  ;AND I/O CLOCK FOR TRANSMISSION
CS     .DBIT    3,INT3            ;TLC1540 CHIP SELECT
ADADDR .DBIT    3,INT2            ;TLC1540 ADDRESS INPUT
DATAOUT.DBIT    6,INT1            ;TLC1540 DATA OUTPUT
;
;
       .TEXT    7000H
;
;
BEGIN  MOV      #18H,INT3         ;SET INT3 AS OUTPUT PIN
       MOV      #18H,INT2         ;SET INT2 AS OUTPUT PIN
       MOV      #00H,DPORT1       ;SET CLKOUT AS GENERAL PURPOSE I/O
       MOV      #00H,DPORT2
       MOV      #08H,DDIR
;
       MOV      #0A0H,B
       LDSP                       ;INITIALIZE STACK POINTER TO 0A0H
;
       CLR      A
       MOV      #22,B
AGAIN  MOV      A,ATABLE-1(B)     ;INITIALIZE THE TABLE
       DJNZ     B,AGAIN
       EINT                       ;ENABLE INTERRUPT
LOOP   CALL     CONVRT            ;START CONVERSIONS
;      .
;      .
;      .
;
;
;      MAIN PROGRAM GOES HERE
;
;      .
;      .
;      .
;   NEED MORE RECENT DATA
       NOP
       CALL     CONVRT            ;START TAKING MORE DATA
       NOP
;
;      MORE MAIN PROGRAM
;
```

The following section is the subroutine CONVRT that initiate the A/D conversion. It sets up the channel address and invokes subroutine ADTRAN for serial transmission. When the transmission finishes, it saves the previous conversion result in ATABLE and generates 44 I/O clocks for current A/D conversion.

```
;
;    SUBROUTINE SECTION
;
;
;    SUBROUNTINE CONVRT
;
;    ENTER : NO PARAMETERS
;    EXIT  : ATABLE - FILL 22 ENTRIES STARTING FROM ATABLE
;
CONVRT PUSH     A
       PUSH     B
       CLR      ADCHANL          ;INITIALIZE CHANNEL ADDRESS
                                 ;THE UPPER 4 BITS INDICATE THE CHANNEL ADDRESS
       CLR      FLAG             ;CLEAR ALL FLAGS
       MOV      #12,CHNLCNT      ;SET COUNT TO NUMBER OF CHANNELS +  1
                                 ;ONE MORE TRANSMISSION TO READ BACK
                                 ;THE CONVERSION RESULT
NEXT   MOV      ADCHANL,B        ;
       SWAP     B                ;PASS THE CHANNEL ADDRESS TO
                                 ;SUBROUNTINE THROUGH REGISTER B,
                                 ;THE UPPER 4 BITS IS THE CHANNEL ADDRESS
       CLR      RESULT           ;CLEAR THE TEMPORARY REGISTER
       CLR      RESULT-1
       CALL     ADTRAN           ;TRANSMIT ADDRESS AND RECEIVE DATA
       MOV      ADCHANL,B        ;IS THE CHANNEL ADDRESS IS 0?
       JZ       SKSAVE           ;SKIP THE FIRST ONE
       RLC      B                ;MULTIPLY BY TWO
       MOV      RESULT-1,A       ;SAVE THE RESULT
       MOV      A,ATABLE-2(B)
       MOV      RESULT,A
       MOV      A,ATABLE-1(B)
SKSAVE INC      ADCHANL          ;NEXT CHANNEL
;
       MOV      #44,B
REPEAT SBIT1    IOCLK            ;44 SYSTEM CLOCKS FOR CONVERSION
       SBIT0    IOCLK
       DJNZ     B,REPEAT
;
       DJNZ     CHNLCNT,NEXT
       POP      B
       POP      A
       RTS
```

The following section is subroutine ADTRAN that handles the communication between TMS370 and TLC1540/1.

```
;
;    SUBROUTINE ADTRAN
;
;    BIT BANGING ROUTINE
;    TRANSMITTING AND RECEIVING DATA TO/FROM TLC1540
;
;    ENTER : B = AD CHANNEL ADDRESS (UPPER 4 BITS)
;    EXIT : RESULT - 10 BITS RESULT
;
ADTRAN SBIT0    CS                  ;CHIP SELECT ACTIVE
       SBIT1    IOCLK               ;SEND TWO CLOCK PULSES TO TLC1540
       SBIT0    IOCLK
       SBIT1    IOCLK
       SBIT0    IOCLK
       MOV      #8,BITCNT           ;SET UP COUNTER
ADRTRA SBIT1    ADADDR              ;TRANSMIT THE ADDRESS
       RL       B
       JC       BIT1                ;IS ADDRESS EQUAL TO 1
       SBIT0    ADADDR              ;NO, SET IT BACK TO 0
BIT1   SBIT1    IOCLK
       RLC      RESULT              ;GET THE CONVERTED RESULT
       RLC      RESULT-1            ;THE BIT IS EQUAL TO 1
       JBIT0    DATAOUT,BIT0        ;IS THE DATA BIT EQUAL TO 0
       OR       #1,RESULT           ;NO, SET IT BACK TO 1
BIT0   SBIT0    IOCLK
       DJNZ     BITCNT,ADRTRA
;
;
       INV      FLAG                ;UPDATE THE FLAG
       BTJZ     #1,FLAG,DONE
       MOV      #2,BITCNT           ;SET COUNTER FOR THE LAST 2 BIT
       SBIT1    CS                  ;CS GO INACTIVE AFTER THE EIGHTH
                                    ;I/O CLOCK, CS MUST BE DEACTIVATED
                                    ;TWO I/O CLOCK BEFORE THE END OF
                                    ;TRANSMISSION
       JMP      BIT1
DONE   RTS
;
;
;   INIT INTERRUPT VECTORS
           .SECT  "vect",7FFEH
           .WORD  BEGIN
;
;
```

The above examples demonstrate the basic principle of interfacing a serial A/D with the TMS370 family microcontrollers. For applications that use TMS370X10, but only need one channel A/D, you may consider TLC548/9 (no system clock and channel address, much simpler compared to TLC1540/1) which is a single-channel 8-bit A/D converter. (More information about external A/D converters is provided in Appendix C.)

Section 6

Conclusions

This application report provides A/D conversion information using the TMS370 family microcontrollers. Examples have been given to demonstrate the operation of the A/D module, typical methods of interfacing to the external circuits and interaction with other modules to provide cost-effective system solution. The TMS370 on-chip timer can provide a handy method to control the sampling frequency of conversions. Calibration data of analog components can be stored in the Data EEPROM module. Calibration data can be used to adjust the conversion result to achieve high system accuracy with less expensive analog components.

Appendix A

A/D Control Registers

The A/D is controlled and accessed through registers in the peripheral file. These registers are listed in Figure A-1 and described in the TMS370 Data Manual Section 11.4. The bits shown in shaded boxes in Figure A-1 are Privilege mode bits, they can only be written to in the Privilege mode.

PERIPHERAL FILE FRAME 7: A-TO-D CONVERTER CONTROL REGISTERS

ADDR	PF	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
1070h	070	CONVERT START	SAMPLE START	REF VOLT SELECT 2	REF VOLT SELECT 1	REF VOLT SELECT 0	AD INPUT SELECT 2	AD INPUT SELECT 1	AD INPUT SELECT 0	ADCTL
1071h	071	---	---	---	---	---	AD READY	AD INT FLAG	AD INT ENA	ADSTAT
1072h	072	A-TO-D CONVERSION DATA REGISTER								ADDATA
1073h TO 107Ch	073 TO 07C	RESERVED								
107Dh	07D	PORT E DATA INPUT REGISTER								ADIN
107Eh	07E	PORT E INPUT ENABLE REGISTER								ADENA
107Fh	07F	AD STEST	AD PRIORITY	AD ESPEN	---	---	---	---	---	ADPRI

Figure A-1. Peripheral File Frame 7: A/D Converter Control Registers

Appendix B

A/D Errors

Figure B-1 shows the transfer characteristics of the A/D conversion and the related errors. Note the TMS370 specification limits are included in the parentheses below.

Absolute Accuracy (The absolute error denoted by "a" is ± 1 LSB)

This is an indication of the discrepancy between the A/D converted value of a given input and the theoretical value. It is measured by the difference (positive or negative) between the theoretical midpoint of a given digital output code and any analog input that will produce that code. Absolute error comprises offset error, gain error, linearity error and is generally expressed in terms of LSB.

Differential Linearity Error (The absolute error denoted by "b" is ± 1/2 LSB)

The difference between the actual step width and the ideal value. If the differential linearity error is greater than 1 LSB, this can lead to missing codes in the A/D conversion (non-monotonicity).

Gain Error (refers to absolute accuracy)

The difference between the actual midstep value and the nominal midstep value in the transfer curve at the specified gain point after the offset error has been adjusted to zero.

Offset Error (refer to absolute accuracy)

The difference between the actual midstep value and the nominal midstep value at the offset point.

Quantization Error (The quantization error denoted by "c" ± 1/2 LSB)

Quantization error is an inherent error in any A/D converter. It is the maximum possible deviation of the actual analog input value from the nominal midstep value. For a linear A/D converter, its value equals +/- 1/2 LSB.

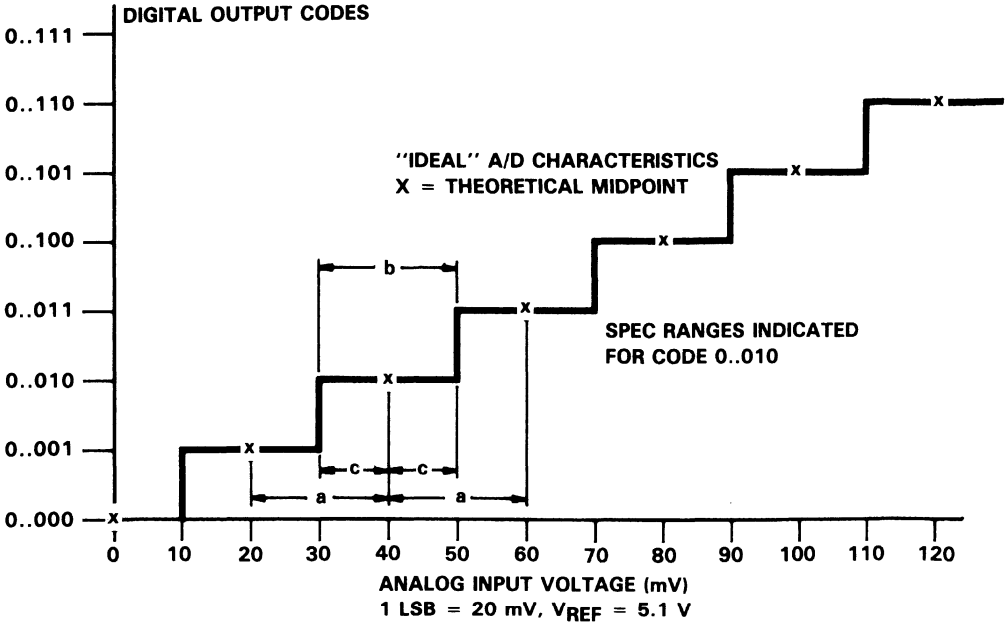


Figure B-1. A/D Transfer Characteristics

Appendix C

External A/D Converters

The following section provides some hints for using external components to perform A/D conversion. This will be useful for low end applications using TMS370X10 but still need A/D, or those applications that need more resolution than the on-chip A/D can provide.

For applications requiring high accuracy but slow conversion rate (in terms of ms). One can use a dual slope A/D converter like TL505C. The on-chip timer can be used to generate precise timing control signals and measure the output timing (input capture function) to determine the input voltage.

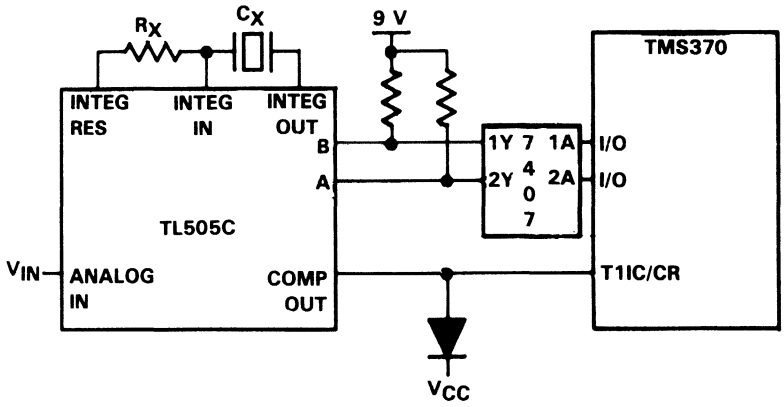
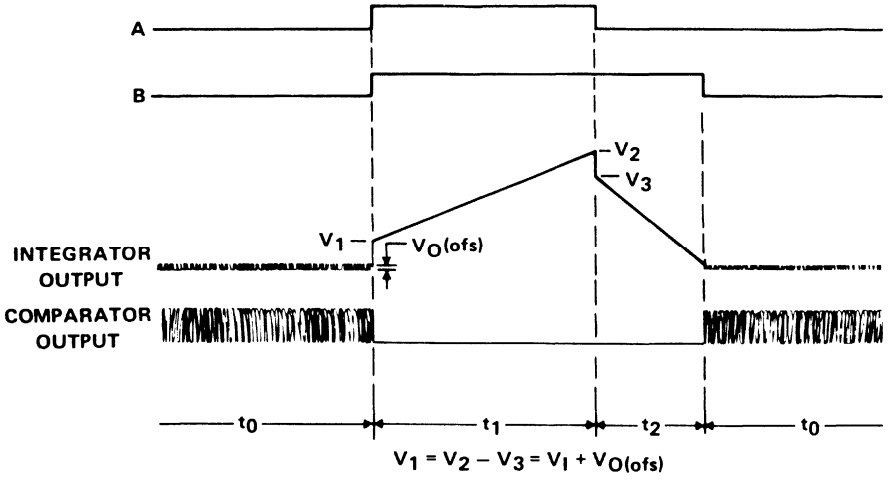


Figure C-1. Functional Block Diagram of TL505C Interface with TMS370



FUNCTION TABLE

CONTROLS A	CONTROLS B	ANALOG SWITCHES CLOSED
L	L	S1, S2
H	H	S3
L	H	S1, S4

H = V_{IH}, L = V_{IL}

$$V_{IN} = - V_{REF} \frac{t2}{t1}$$

Figure C-2. Conversion Process Timing Diagram

Instead of using commercial A/D converters, you can also build your own A/D. One of the simplest implementations is to use a 10-bit D/A converter with a voltage comparator to determine the input voltage. The TMS370 performs a binary search to determine the digital value of the input voltage (10 conversions for 10-bit D/A converter).

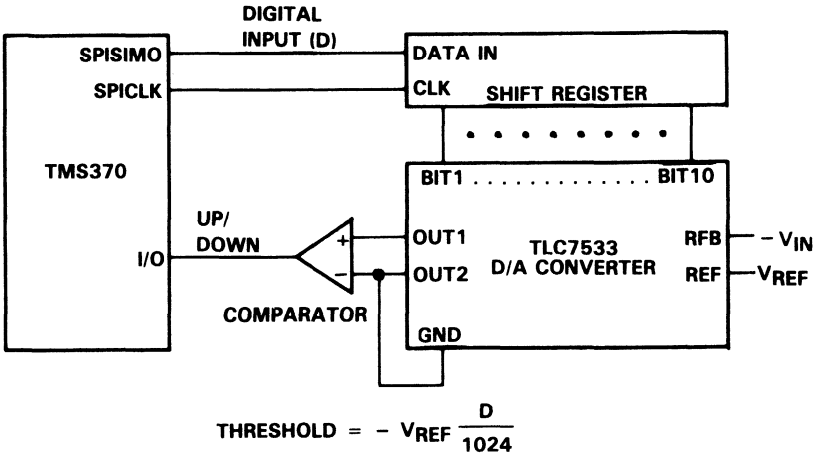


Figure C-3. Functional Block Diagram Using D/A Converter as A/D

Another way to implement an A/D is using a V/F converter. The frequency output can be measured by the on-chip timer using the input capture function. Voltage-Frequency converter can generate frequency outputs up to 500 kHz. The on-chip timer can provide precise timing measurements for the frequency output signal. For a clock frequency of 5 MHz, the timer clock period is 200 ns, the accuracy of the A/D conversion will mainly depend on the V/F converter.

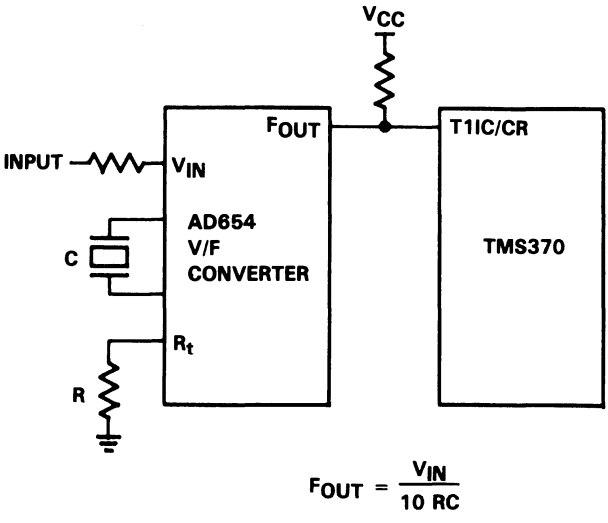


Figure C-4. Functional Block Diagram Using V/F Converter as A/D

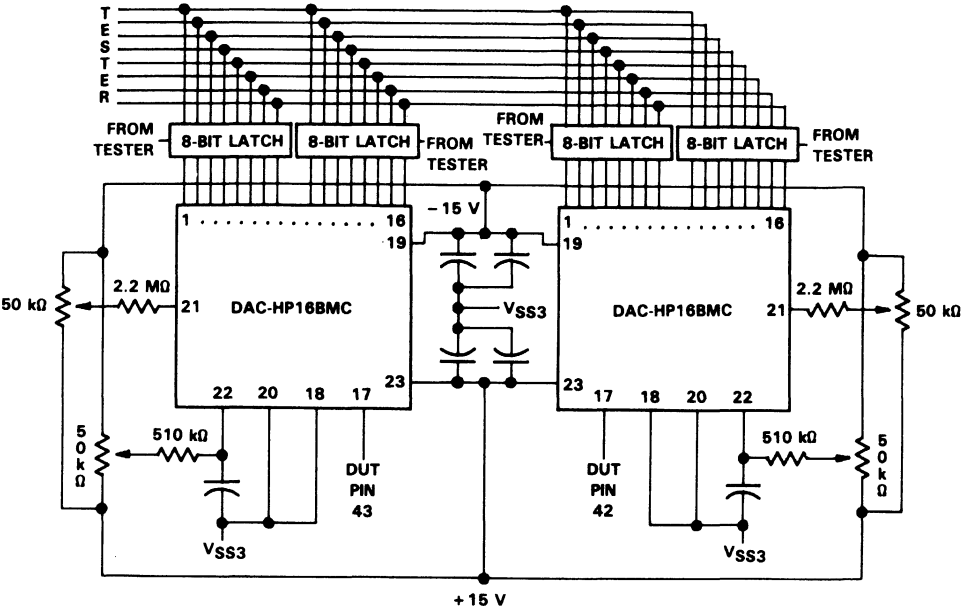
Appendix D

A/D Testing

The following section provides information about testing procedures of two A/D converter's parameters (absolute accuracy and differential linearity error).

Test Conditions

CLKIN	2 MHz and 20 MHz
V_{CC3}	5.5 V
V_{REF}	5.1 V
Sampling time	2 µs (CLKIN = 20 MHz) 20 µs (CLKIN = 2 MHz)



Note: Pin 24 of DACs left open, latches connected to digital +5 V and GND.

Figure D-1. Block Diagram of Test Set-Up

Two 16-bit D/A converters are used to provide accurate reference voltage and analog input signal.

At the theoretical midpoint of each code 256 conversions are performed. If all 256 digital codes are generated by these conversions, this will guarantee that the A/D conversions are within one LSB absolute accuracy.

The differential linearity error is measured by the code width (voltage range) of each individual code. With V_{REF} at 5.1V, one-half LSB corresponds to 10 mV. For +/- 1/2 LSB differential linearity error, the code width of any individual code will need to be within 10 mV to 30 mV range. Figure D-2 illustrates the code width measurement :

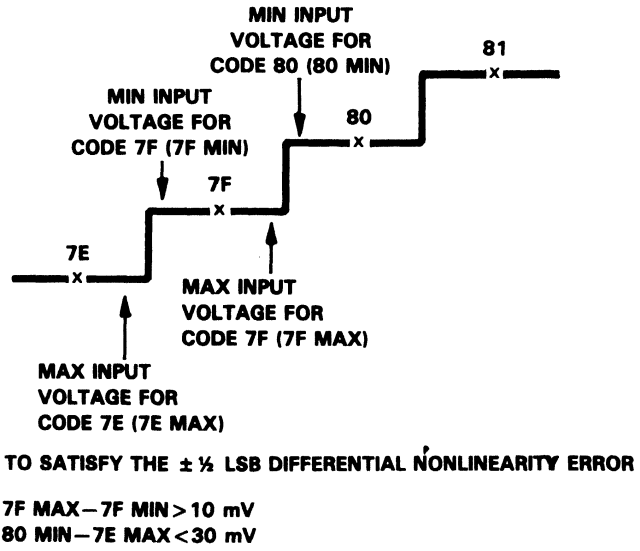


Figure D-2. Code Width Measurement

Conversions are performed with input incremented by steps of 2 mV starting from the midpoint of 7E. The analog voltage 7Emax is the maximum possible value before any conversion that generates 7F.

Another set of conversions is performed with input decremented at a step of 2 mV starting from the midpoint of 80. The analog voltage 80min is the minimum possible value before any conversion that generates 7F.

In order to minimize the test time for the A/D modules, only 14 codes are tested for the differential linearity error (see Figure D-3). These 14 codes have the largest differential linearity errors. In Section 2, we explained that the conversion is achieved by switching the capacitors one at a time. The transition of these codes correspond to switching the capacitor array to the next significant (weighted capacitance) stage. Figure D-4 shows a typical A/D differential linearity characterization result.

0000 0000	0000 0011
0000 0001	0000 0100
0000 0111	0000 1111
0000 1000	0001 0000
0001 1111	0000 1111
0010 0000	0100 0000
0111 1111	
1000 0000	

Figure D-3. Codes Having Maximum Differential Linearity Error

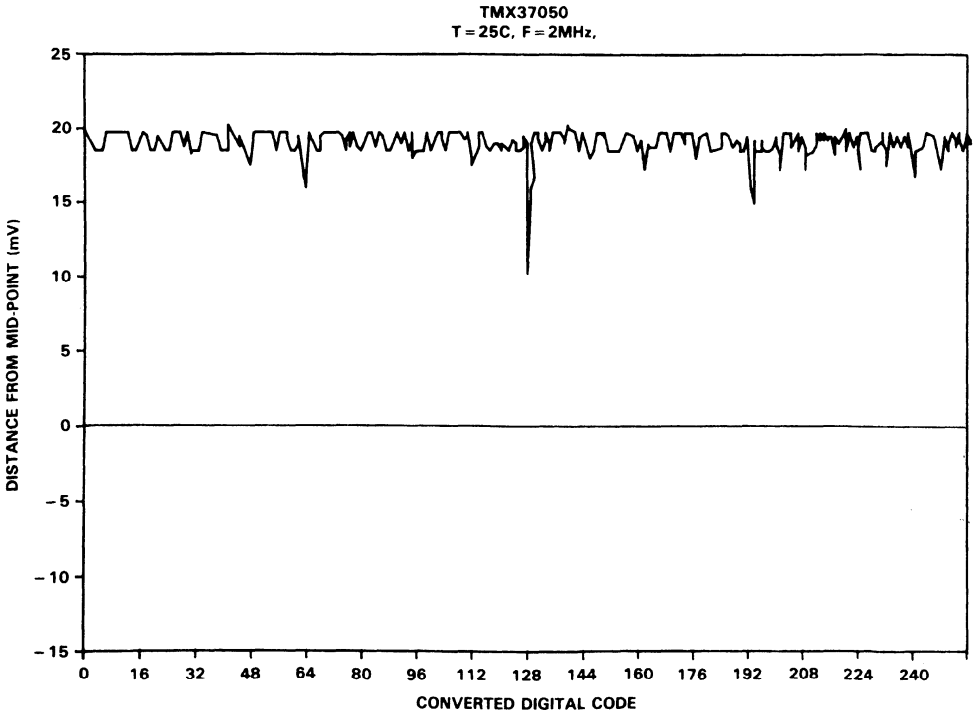


Figure D-4. Differential Linearity Error

Appendix E

Glossary

Aliasing Signal: An aliasing signal is the false lower frequency signal reconstructed from an analog input because of insufficient sampling rate (see Nyquist Criterion).

Conversion speed: Conversion speed provides an indication of system sampling rate, and is usually expressed in conversions per second.

Code width or Step width: The voltage corresponding to the difference between two adjacent code transitions.

Input Leakage: Leakage current of an analog input pin.

Monotonicity: There is at least one analog input voltage for every possible digital output code (that is, no missing code) occurring in ascending or descending order.

Nyquist Criterion: The Nyquist criterion states that the original signal can be recovered without distortion if the sampling frequency is greater than twice that of the highest frequency to be sampled.

Ratiometric Conversion: The output of an A/D conversion is a digital number proportional to the ratio of the input to a reference (fixed or variable). In some applications, where the measurement is affected by the changes of the reference voltage (slow varying comparable to the conversion time), it is advantageous to use that same reference as the reference for the conversion to eliminate the effect of variation.

Resolution: This is the ability of the converter to distinguish between adjacent analog input levels. A 8-bit converter would be capable of distinguish between input level that differ by 1/256 of the full- scale range.

Sample-and-Hold circuit: A circuit which accurately acquires and stores an analog voltage on a capacitor for a certain period of time.

Transducer: Transducer is a device that converts a physical parameter into an electrical signal.

Appendix F

References

Linear Applications Group. Linear Circuits Applications, LinCMOS Products: Op-Amps, Comparators and Timers. SLNA001. Dallas, Texas. Texas Instruments Technical Publishing. 1987.

Linear Applications Handbook. Milpitas, California. Linear Technology Corporation. 1987.

Linear Products. TLC1540 LinCMOS 10-bit Analog-to-Digital Converter Data Sheet. SLNS010B. Dallas, Texas. Texas Instruments Technical Publishing. 1986.

McCreary, James L. All-MOS Charge Redistribution Analog-to-Digital Conversion Technique. IEEE Journal of Solid-State Circuits. 1975.

Pippenger, D.E. and Tobaben, E.J. Linear and Interface Circuits Applications. Dallas, Texas. Texas Instruments Technical Publishing. 1986.

Sheingold, Daniel H. Transducer Interface Handbook. Massachusettes. Analog Devices, Inc. 1981.

The Handbook of Linear IC Applications. Tucson, Arizona. Burr-Brown Corporation. 1987.

T.I. Microcontroller Applications Group. TMS370 Family Data Manual. Dallas, Texas. SPNS014. Texas Instruments Technical Publishing. 1988.

T.I. Microcontroller Applications Group. Using the TMS370 EEPROM Application Report. SPNA007. Dallas, Texas. Texas Instruments Technical Publishing. 1989.

T.I. Microcontroller Applications Group. Using the TMS370 Timer Application Report. SPNA008. Dallas, Texas. Texas Instruments Technical Publishing. 1989.

T.I. Microcontroller Applications Group. Using the TMS370 SPI and SCI Modules Application Report. SPNA006. Dallas, Texas. Texas Instruments Technical Publishing. 1989.

T.I. Microcontroller Group. TMS370 Family Assembly Language Tools User's Guide. Dallas, Texas. SPNU010. Texas Instruments Technical Publishing. 1987.

T.I. Microcontroller Group. TMS370Cx5x 8-Bit Microcontrollers Data Sheet. Dallas, Texas. SPNS010A. Texas Instruments Technical Publishing. 1988.