9 Analog-to-Digital Converter

CHAPTER 9
ANALOG-TO-DIGITAL CONVERTER

This chapter provides an overview of the analog-to-digital (A/D) conversion process and describes how to program the A/D converter, read the conversion results, and interface with external circuitry. It also describes the transfer function and error sources.

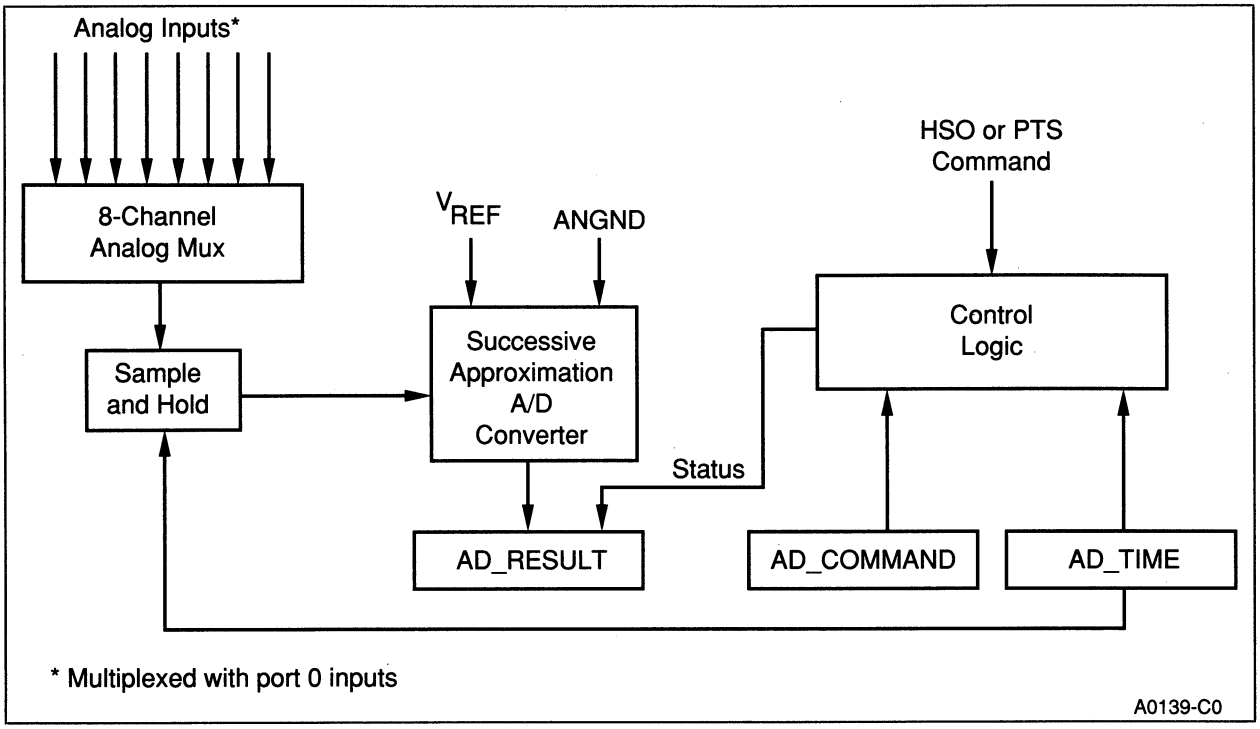

Figure 9-1. A/D Converter Block Diagram

9.1. A/D FUNCTIONAL OVERVIEW

The A/D converter module (shown in Figure 9-1) converts an analog input into an 8- or 10-bit digital representation. The main parts of the A/D converter module are:

- The eight analog inputs (ACH0–ACH7).
- An 8-channel multiplexer to select one of the eight input channels.
- The sample-and-hold circuit.
- The 10-bit successive approximation A/D converter.
- The AD_COMMAND register, which controls the converter operation via the control logic.
- The two-byte AD_RESULT register, which contains the conversion results and status information.
- The AD_TIME register, which contains the sample and conversion times.

Once the A/D converter receives the command to start a conversion, one state time elapses before sampling begins. During this sample delay, the successive approximation register is reset and the designated multiplexer channel is selected.

After the sample delay, the multiplexer output is connected to the sample capacitor and remains connected for the specified sample time. After this *sample window* closes, the input to the sample capacitor is disconnected from the multiplexer so that changes on the input pin will not alter the stored charge while the conversion is in progress. The comparator is then auto-zeroed and the conversion begins.

The A/D converter uses a successive approximation algorithm to perform the analog-to-digital conversion. The converter hardware consists of a 256-resistor ladder, a comparator, coupling capacitors, and a 10-bit successive approximation register (SAR) with logic that guides the process. The resistive ladder provides 20 mV steps (V_{REF} = 5.12 Volts), while capacitive coupling creates 5 mV steps within the 20 mV ladder voltages. Therefore, 1024 internal reference voltages are available for comparison against the analog input to generate a 10-bit conversion result.

The successive approximation conversion is performed by comparing a sequence of reference voltages to the analog input, in a binary search for the reference voltage that most closely matches the input. The 1/2 full scale reference voltage is the first tested. This corresponds to a 10-bit result where the most-significant bit is zero and all other bits are ones (0111.1111.11B). If the analog input was less than the test voltage, bit 10 of the SAR is left a zero, and a new test voltage of 1/4 full scale (0011.1111.11B) is tried. If the analog input was greater than the test voltage, bit 9 of the SAR is set. Bit 8 is then cleared for the next test (0101.1111.11B). This binary search continues until 10 (or 8) tests have occurred, at which time the valid conversion result resides in the SAR where it can be read by software. The result is equal to the ratio of the input voltage divided by the analog supply voltage. If the ratio is 1.00, then the result will be all ones.

9.2. PROGRAMMING THE A/D CONVERTER

The following A/D converter parameters are programmable:

- Either eight-bit or ten-bit conversions
- Input channel selection
- Sample and Convert times
- Interrupt on completion or no interrupt
- 80196KB-compatible mode
- Immediate conversions or timed High-Speed Output (HSO) module or Peripheral Transaction Server (PTS) starts.

Table 9-1 lists the programmable registers that affect that performance and function of the A/D converter.

Table 9-1. A/D Control and Status Registers

Register Mnemonic	Register Name	Description
AD_COMMAND	A/D Command Register	This register selects the A/D channel number to be converted, determines whether the A/D conversion starts immediately or with an HSO command, and selects either the 8-bit or 10-bit conversion mode.
AD_RESULT	A/D Result Register	This register consists of two bytes. The high byte contains the eight most-significant bits from the conversion. The low byte indicates the A/D channel number that was used for the conversion, indicates whether a conversion is currently in progress, and contains the two least-significant bits from a ten-bit conversion.
AD_TIME	A/D Conversion Time	This register defines the sample and convert times, if enabled by the IOC2 register.
IOC2	Input/Output Control Register 2	This register selects whether the A/D conversion times are controlled by the AD_TIME register or by emulating the fast and normal conversion modes of the 8XC196KB.
INT_MASK	Interrupt Mask	This register enables/disables the A/D Conversion Complete interrupt (INT01, 2002H). Setting bit 1 enables this interrupt; clearing bit 1 disables (masks) the interrupt.
INT_PEND	Interrupt Pending	This register indicates that an A/D Conversion Complete interrupt is pending when bit 1 is set. Bit 1 is cleared when the interrupt vectors to 2002H.

9.2.1. Selecting the A/D Sample and Convert Times

Two parameters, sample time and convert time, control the time required for an A/D conversion. The sample time is the length of time that the analog input voltage is actually connected to the sample capacitor. If this time is too short, the sample capacitor will not charge completely. If the sample time is too long, the input voltage may change and cause conversion errors. The convert time is the length of time required to convert the analog input voltage stored on the sample capacitor to a digital value. The convert time must be long enough for the comparator and circuitry to settle and resolve the voltage. Excessively long conversion times allow the sample capacitor to discharge, degrading accuracy.

The sample and convert times either default to a predefined 80C196KB-compatible mode or are specified in the AD_TIME register. Clearing the AD_TIME_ENA bit (IOC2.3) enables an 80C196KB-compatible conversion. In the 80C196KB-compatible mode, the AD_FAST bit (IOC2.4) controls the sample and convert times. The 80C196KB slow mode uses 15 state times for the sample time and a total of 158 state times for the entire conversion. The 80C196KB fast mode uses eight state times for the sample time and a total of 91 state times for the conversion. When using either the fast or slow mode, convert these state times into microseconds to ensure that the data sheet T_{SAM} and T_{CONV} values are met at the operating clock frequency.

Setting the AD_TIME_ENA bit (IOC2.3) enables the AD_TIME register shown in Figure 9-2. The AD_TIME register programs the A/D conversion speed. The resolution and clock frequency determine the conversion speed. Use the T_{SAM} and T_{CONV} specifications on the data sheet to determine appropriate values for sample (SAM) and convert (CONV) times; otherwise, erroneous conversion results may occur.

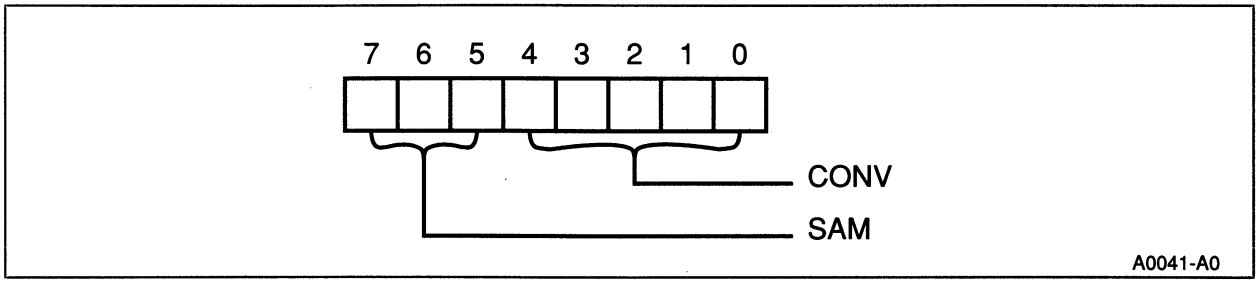

Figure 9-2. AD_TIME Register

Use the following formulas to determine the optimal SAM and CONV values:

$$SAM = \frac{(T_{SAM} \times F_{OSC} - 2)}{2} / 4 \qquad CONV = \left[\frac{(T_{CONV} \times F_{OSC} - 3)}{2} / B\right] - 1$$

where:

SAM = 1 to 7

CONV = 2 to 31

T_{SAM} is the sample time, in µsec, from the data sheet

T_{CONV} is the conversion time, in µsec, from the data sheet

F_{OSC} is the XTAL1 frequency, in MHz

B is the number of bits to be converted (8 or 10)

When the SAM and CONV values are known, write them to the AD_TIME register. Do not write to this register while a conversion is in progress; the results are unpredictable.

9.2.2. Programming the IOC2 Register

The IOC2 register (see Figure 9-3) selects whether the A/D conversion times are controlled by the AD_TIME register or by the AD_FAST bit (IOC2.4). Setting the AD_TIME_ENA bit enables the AD_TIME register and the converter uses the sample and convert times specified in AD_TIME (see Figure 9-2).

Clearing the AD_TIME_ENA bit selects the 80C196KB-compatible mode. In this mode, the AD_FAST bit enables or disables the A/D clock prescaler for complete compatibility with the 80C196KB's slow and fast conversion modes. Clearing AD_FAST enables the 80C196KB slow mode, which uses 15 state times for the sample time and a total of 158

state times for the entire conversion. Setting AD_FAST enables the 80C196KB fast mode, which uses 8 state times for the sample time and a total of 91 state times for the conversion.

NOTE

Changing modes while a conversion is in progress will cause unpredictable results.

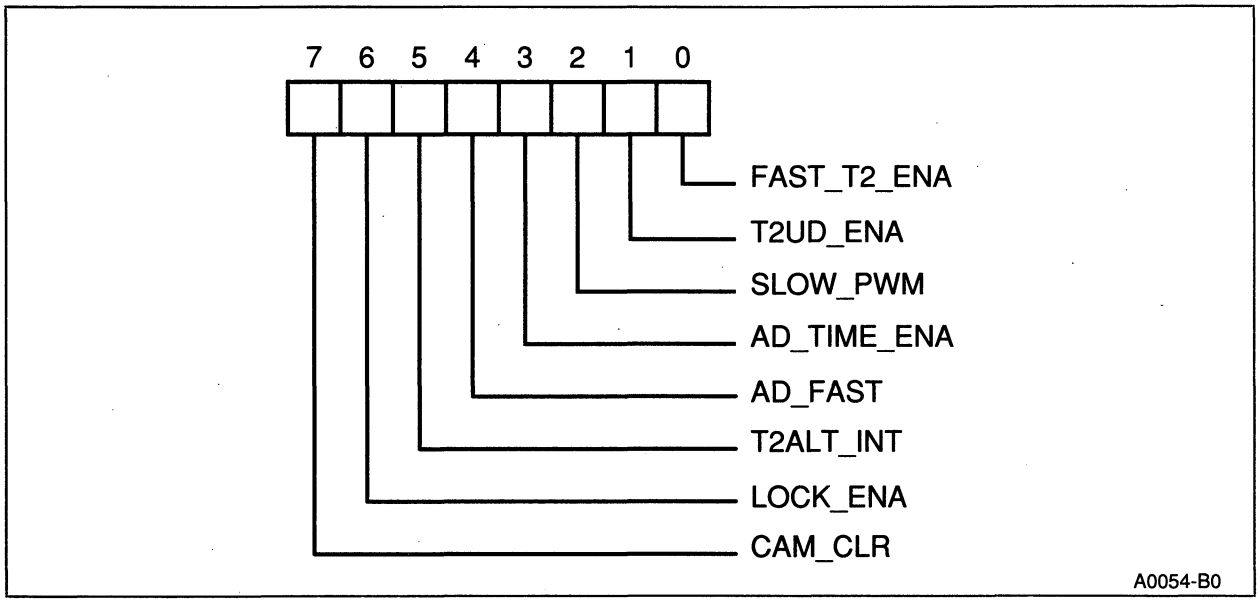

Figure 9-3. The IOC2 Register

9.2.3. Programming the AD_COMMAND Register

The AD_COMMAND register, shown in Figure 9-4, selects the channel to convert, selects an 8-bit or 10-bit conversion, and either starts the conversion or designates the High-Speed Input (HSO) module as the trigger.

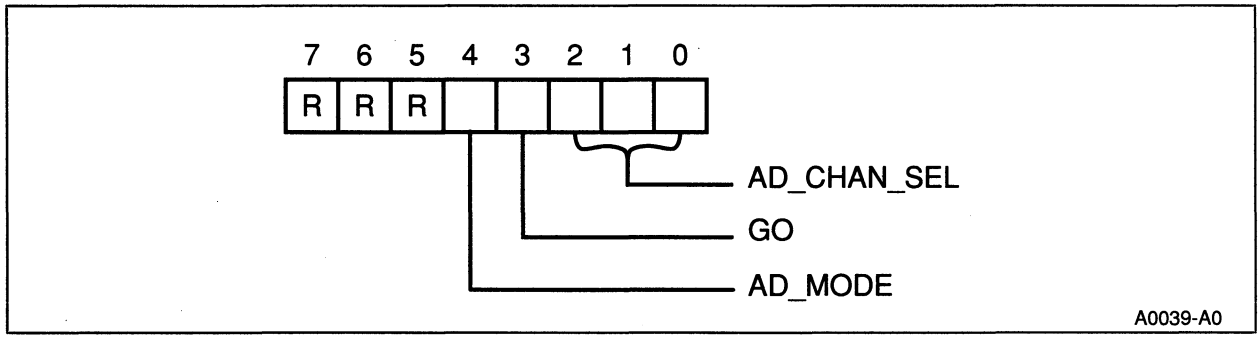

Figure 9-4. The AD_COMMAND Register

Write the channel number to the AD_CHAN_SEL bits (bits 0-2). To select 10-bit conversions, clear the AD_MODE bit (bit 4). To select 8-bit conversions, set the AD_MODE bit. Clear reserved bits 5–7.

Set the GO bit (bit 3) to start the conversion immediately. Conversions initiated by setting the GO bit will start within three state times after the instruction executes.

Clear the GO bit to have the HSO module initiate the conversion at a future time. The HSO module triggers the conversion by executing the command (CMD_TAG = 0FH). The conversion process begins when Timer 1 increments to the programmed value. This aids applications attempting to approach spectrally pure sampling because successive samples spaced by equal Timer 1 delays will occur with a variance of about ± 50 ns (assuming a stable clock on XTAL1).

The following code sample shows how to use the HSO module to trigger the conversion.

```
LDB AD_COMMAND, XXXX1XXXB            ;sets the GO bit
LDB HSO_COMMAND, #START_AD           ;loads the start AD conversion command
                                     ;into the HSO_COMMAND register.
ADD HSO_TIME, TIMER1, #TIME_DELAY    ;adds a 16-bit constant to the current
                                     ;contents of TIMER1, and puts the result
                                     ;in HSO_TIME
```

The HSO register allows multiple START_AD commands to be loaded into the CAM. However, be careful when writing the HSO commands. If Timer 1 increments and the new conversion is started while the previous one is still in progress, the conversion in progress is aborted and the new one is started.

The PTS can begin a new conversion in response to an A/D Conversion Complete interrupt (INT01). Chapter 5, "Interrupts," describes how to use the PTS in the A/D Mode.

9.3. READING THE CONVERSION RESULTS

The AD_RESULT register consists of two bytes (see Figure 9-5). The high byte (03H) contains the eight most-significant bits from the A/D conversion. The low byte (02H) indicates the A/D channel number that was used for the conversion, indicates whether a conversion is currently in progress, and contains the two least-significant bits from a ten-bit A/D conversion. The AD_RESULT register is cleared when a new conversion is started; therefore, to prevent losing data, both bytes must be read before a new conversion starts. See Appendix C for a detailed description of the AD_RESULT register.

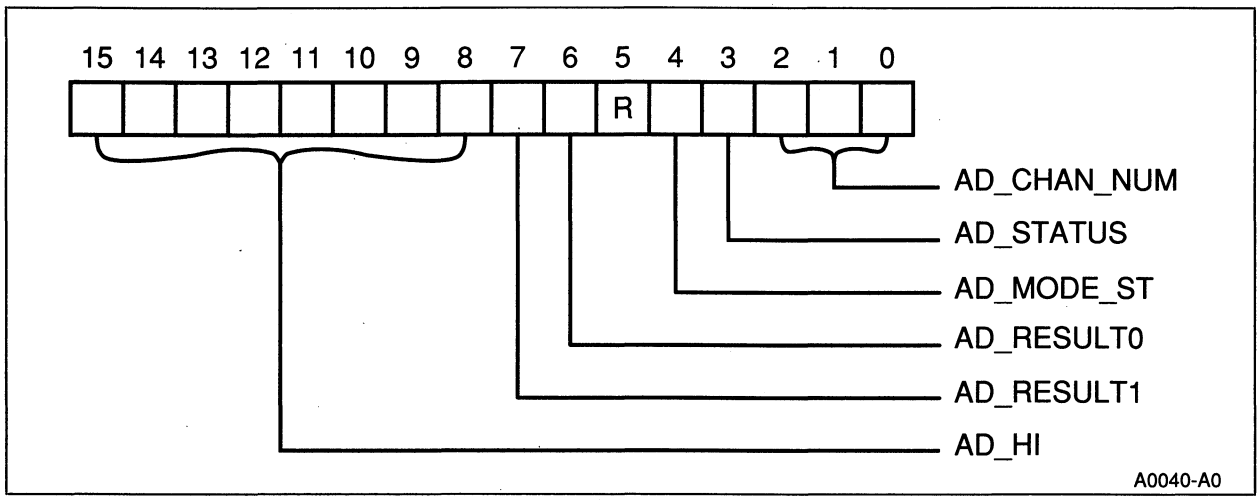

Figure 9-5. The AD_RESULT Register

The A/D converter generates the A/D Conversion Complete interrupt (INT01) when it completes a conversion. This interrupt vectors through location 2002H and is enabled by setting the AD_MASK bit (INT_MASK.1). The AD_PEND bit (INT_PEND.1) is set when a conversion completes, even if the interrupt is masked. This bit stays set until the interrupt vectors to 2002H or the register is cleared. See Chapter 5 for a detailed explanation of interrupts.

Another way to determine the conversion status is to poll the AD_RESULT register. The AD_STATUS bit (AD_RESULT.3) is set while a conversion is in progress. Since it can take up to eight state times to set this bit, do not begin polling until at least eight state times after starting the conversion.

9.4. INTERFACING WITH THE A/D CONVERTER

The external interface circuitry to an analog input is highly dependent upon the application, and can affect the converter characteristics. In the external circuit's design, factors such as input pin leakage, sample capacitor size, and multiplexer series resistance from the input pin to the sample capacitor must be considered.

These external factors are idealized in Figure 9-6. The external input circuit must be able to charge a sample capacitor (C_S) through a series input resistance (R_1) to an accurate voltage given a DC input leakage (I_{LI1}). Typically, C_S is about 3.0 pF, R_1 is about 1 KΩ and I_{LI1} is specified as 3 µA maximum.

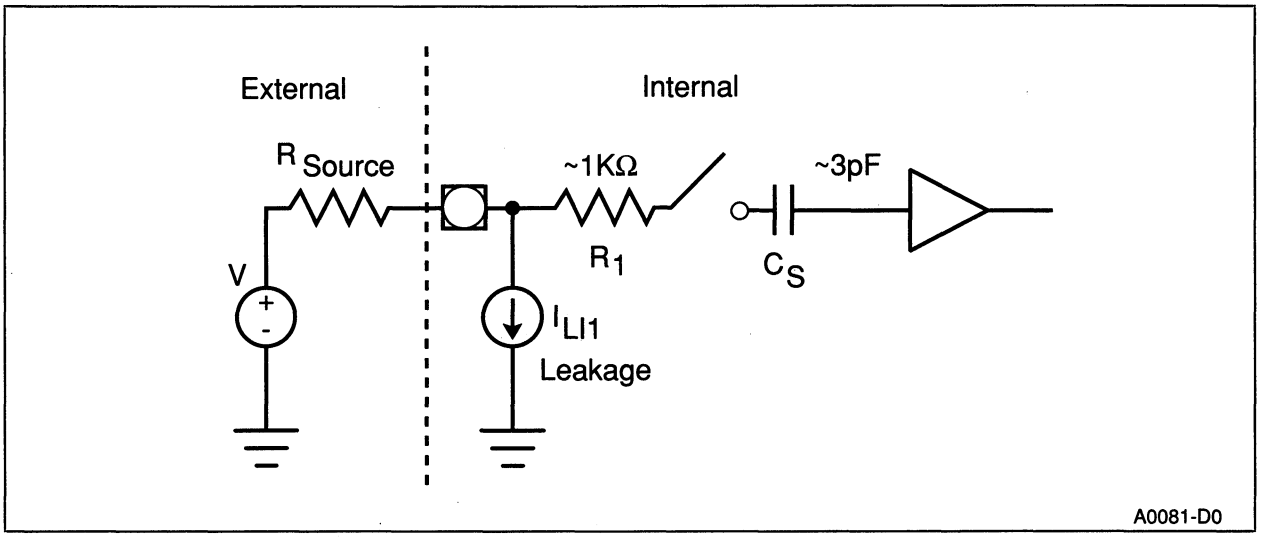

Figure 9-6. Idealized A/D Sampling Circuitry

External circuits with source impedances (R_{SOURCE}) of 1 KΩ or less will be able to maintain an input voltage within a tolerance of about ± 0.6 LSB (1.0 KΩ × 3.0 µA = 3.0 mV) given a DC input leakage (I_{LI1}). Source impedances above 2 KΩ can result in an external error of at least one LSB, due to the voltage drop caused by the 3 µA leakage.

Typically, I_{LI1} leakage is much lower than the maximum specification. Given typical leakage, source impedance may be increased substantially before a one-LSB error is apparent. However, a high source impedance may prevent the internal sample capacitor from fully charging during the sample window. This error can be calculated using the following formula.

$$\text{error (LSBs)} = \left(e^{\frac{-T_{SAM}}{RC}}\right) \times 1024$$

where:

T_{SAM} = sample time, in µseconds
$R = R_{SOURCE} + R_1$, in ohms
$C = C_S$, in µfarads

The effects of this error can be minimized by connecting an external capacitor (C_{EXT}) from the input pin to ANGND. The external signal will charge C_{EXT} to the source voltage. When the channel is sampled, a small portion of the charge stored in C_{EXT} will be transferred to the internal sample capacitor. The ratio of C_S to C_{EXT} causes the loss in accuracy. If C_{EXT} is .005 µF or greater, the maximum error will be –0.6 LSB.

Placing an external capacitor on each analog input will also reduce the sensitivity to noise, as the capacitor combines with series resistance in the external circuit to form a low-pass filter (see Figure 9-7). In practice, one should include a small series resistor prior to the external capacitor on the analog input pin and choose the largest capacitor value practical, given the frequency of the signal being converted. This provides a low-pass filter on the input, while the resistor will also limit input current during over-voltage conditions.

The suggested A/D input circuit shown in Figure 9-7 provides limited protection against over-voltage conditions on the analog input. Should the input voltage inappropriately drop significantly below ground, diode D2 will forward bias at about 0.8 DCV. The pin has an absolute maximum low voltage rating of –0.5 DCV; this leaves about 0.3 DCV across the 270Ω resistor, or about 1 mA of current.

NOTE

Driving any analog input more than 0.5 V beyond ANGND or V_{REF} activates the input protection devices. This drives current into the internal reference circuitry and substantially degrades the accuracy of A/D conversions on all channels.

Thoroughly analyze the applicability of the circuit shown in Figure 9-7 before using it in an actual application.

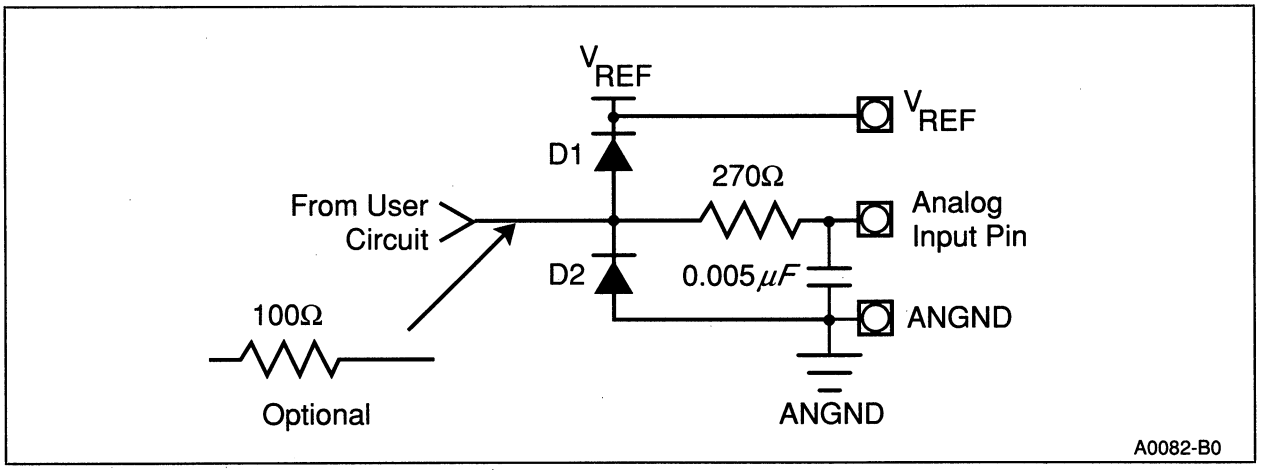

Figure 9-7. Suggested A/D Input Circuit

9.4.1. Analog Ground and Reference

Reference supply levels strongly influence the absolute accuracy of the conversion. For this reason, we recommend that you tie the ANGND pin to the V_{SS} pin as close to the device as possible, using a minimum trace length. In a noisy environment, we highly recommend the use of a separate analog ground plane that connects to V_{SS} at a single point as close to the device as possible. Bypass capacitors should also be used between V_{REF} and ANGND. ANGND should be within about a tenth of a volt of V_{SS}. V_{REF} should be well regulated and used only for the A/D converter. The V_{REF} supply can be between 4.0 and 5.5 V and needs to be able to source approximately 5 mA (see the data sheet for actual specifications). Large negative current spikes on the ANGND pin relative to V_{SS} may cause the analog circuitry to latch-up. This is an additional reason to follow careful grounding practice.

The analog voltage reference (V_{REF}) is the positive supply to which all A/D conversions are compared. It is also the supply to Port 0 if the A/D converter is not being used. If high accuracy is not required, V_{REF} can be tied to V_{CC}. If accuracy is important, V_{REF} must be very stable. One way to accomplish this is through the use of a precision power supply or a separate voltage regulator (usually an IC). These devices must be referenced to ANGND, **not** to V_{SS}, to ensure that V_{REF} tracks ANGND and not V_{SS}.

9.4.2. Using Mixed Analog and Digital Inputs

Port 0 may be used for both analog and digital input signals at the same time. However, when Port 0 is read (by reading location 0EH), some noise may be injected into the analog circuitry. For this reason, make certain that an analog conversion is not in progress when the port is read. Refer to Chapter 6, "I/O Ports" for information about using Port 0 as a digital input.

9.5. TRANSFER FUNCTION AND A/D ERROR SOURCES

The conversion result is an 8- or 10-bit representation of the ratio of the input voltage to the reference voltage. Use the following formula to calculate the 10-bit conversion result:

$$\text{RESULT (10-bit)} = 1023 \times \frac{V_{IN} - \text{ANGND}}{V_{REF} - \text{ANGND}}$$

This produces a stair-stepped *transfer function* when the output code is plotted versus input voltage. The resulting digital codes can be taken as simple ratiometric information, or they provide information about absolute voltages or relative voltage changes on the inputs.

The more demanding the application, the more important it is to fully understand the converter's operation. For simple applications, knowing the *absolute error* of the converter is sufficient. However, closing a servo-loop with analog inputs requires a detailed understanding of an A/D converter's operation and errors.

In many applications, it is less critical to record the absolute accuracy of an input than it is to detect that a change has occurred. This approach is acceptable as long as the converter is *monotonic* and has *no missing codes*. That is, increasing input voltages produce adjacent, unique output codes that are also increasing. Decreasing input voltages produce adjacent, unique output codes that are also decreasing. In other words, there exists a unique input voltage range for each 10-bit output code that produces that code only, with a repeatability of typically ± .25 LSBs (1.5 mV).

The inherent errors in an analog-to-digital conversion process are quantizing error, zero offset error, full-scale error, differential non-linearity, and non-linearity. All of these are *transfer function* errors related to the A/D converter. In addition, temperature coefficients, V_{CC} rejection, sample-hold feedthrough, multiplexer off-isolation, channel-to-channel matching, and random noise should be considered. Fortunately, one absolute error specification describes the sum total of all deviations between the actual conversion process and an ideal converter. However, the various sub-components of error are important in many applications.

An unavoidable error results from the conversion of a continuous voltage to an integer digital representation. This error is called *quantizing error* and is always ± 0.5 LSB. Quantizing error is the only error seen in a perfect A/D converter, and is obviously present in actual converters. The transfer function for an ideal 3-bit A/D converter is shown as the Ideal Characteristic (see Figure 9-8).

Note that the Ideal Characteristic possesses unique qualities:

- its first code transition occurs when the input voltage is 0.5 LSB;
- its full-scale code transition occurs when the input voltage equals the full-scale reference minus 1.5 LSB; and
- its code widths are all exactly one LSB.

These qualities result in a digitization without zero offset, full-scale, or linearity errors. In other words, a perfect conversion.

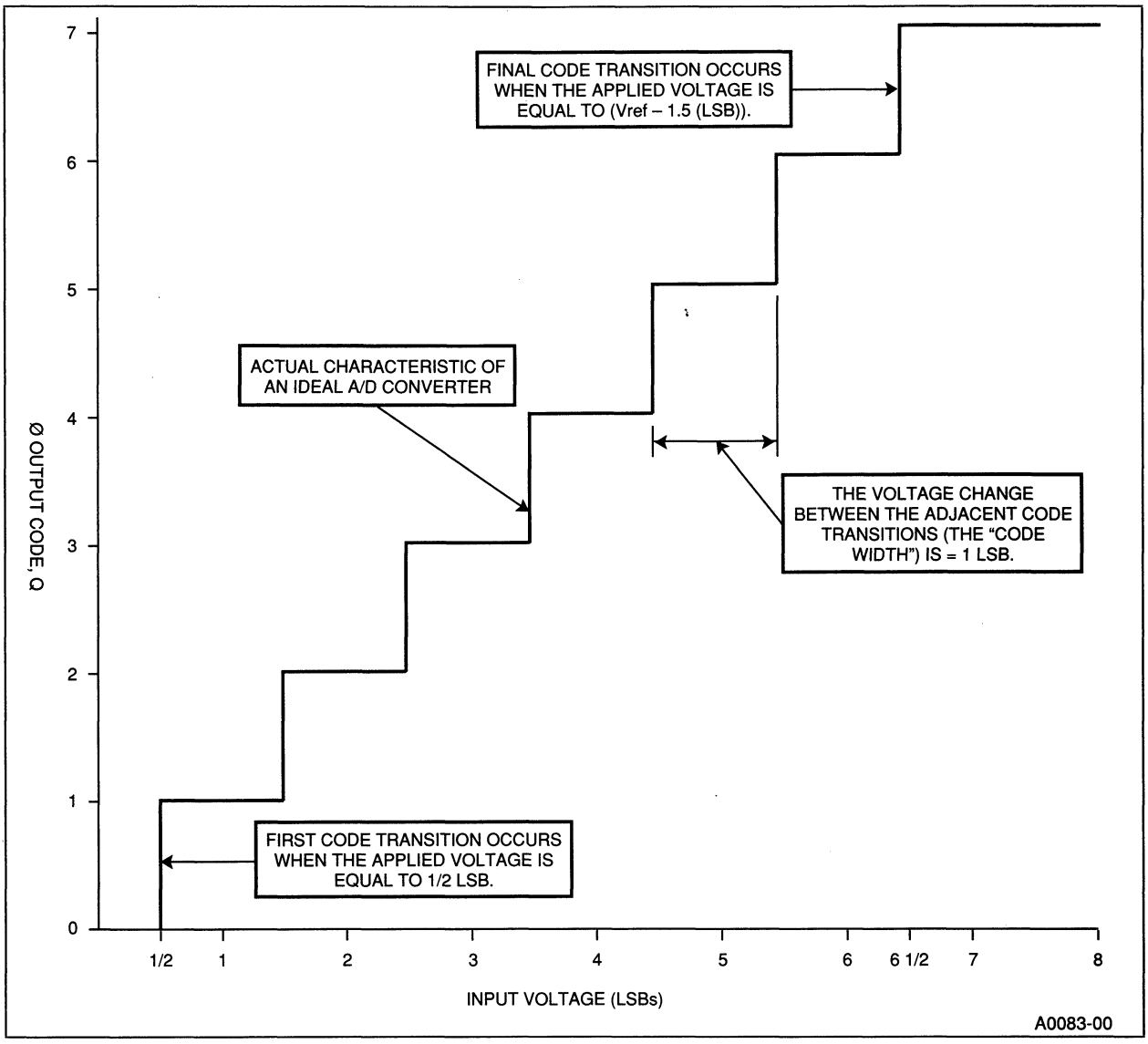

Figure 9-8. Ideal A/D Conversion Characteristics

The Actual Characteristic of a hypothetical 3-bit converter is not perfect. When the Ideal Characteristic is overlaid with the actual characteristic, the actual converter is seen to exhibit errors in the locations of the first and final code transitions and in code widths, as shown in Figure 9-9. The deviation of the first code transition from ideal is called *zero offset* error, and the deviation of the final code transition from ideal is *full-scale* error. The deviation of a code width from ideal causes two types of errors: Differential Non-Linearity and Non-Linearity. *Differential Non-Linearity* is a measure of local code-width error, whereas *Non-Linearity* is a measure of overall code-transition error.

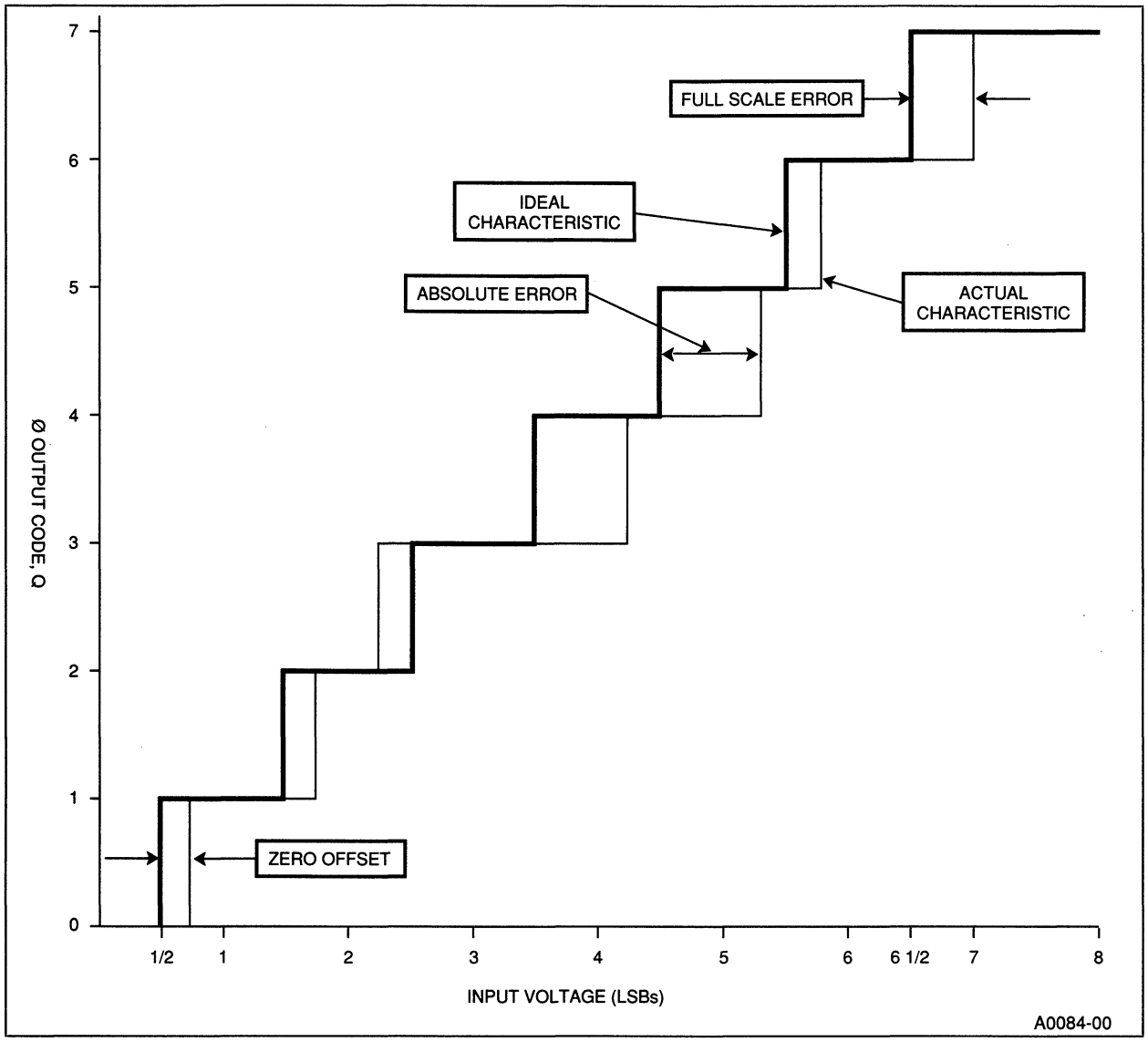

Figure 9-9. Actual and Ideal A/D Conversion Characteristics

Differential Non-Linearity is the degree to which actual *code widths* differ from the ideal one-LSB width. It gives the user a measure of how much the input voltage may have changed in order to produce a one-count change in the conversion result. In the 10-bit converter, the code widths are ideally 5 mV (5.12 V_{REF} / 1024). If such a converter is specified to have a maximum Differential Non-Linearity of 2 LSBs (10 mV), then the maximum code width will be no greater than 10 mV larger than ideal, or 15 mV.

Because the 8XC196KC/KD converter has *no missing codes*, the minimum code width will always be greater than –1 (negative one). The Differential Non-Linearity error on a particular code width is compensated for by other code widths in the transfer function, such that 1024 unique steps occur. The actual code widths in this converter typically vary from 2.5 mV to 7.5 mV.

Non-Linearity is the worst case deviation of *code transitions* from the corresponding code transitions of the Ideal Characteristic. Non-Linearity describes how much Differential Non-Linearities could add up to produce an overall maximum departure from a linear characteristic. If the Differential Non-Linearity errors are too large, it is possible for an A/D converter to miss codes or to exhibit non-monotonic behavior. Neither behavior is desirable in a closed-loop system. A converter has *no missing codes* if there exists for each output code a unique input voltage range that produces that code only. A converter is *monotonic* if every subsequent code change represents an input voltage change in the same direction.

Differential Non-Linearity and Non-Linearity are quantified by measuring the Terminal-Based Linearity errors. A Terminal-Based Characteristic results when an Actual Characteristic is translated and scaled to eliminate zero offset and full-scale error, as shown in Figure 9-10. The Terminal-Based Characteristic is similar to the Actual Characteristic that would be seen if zero offset and full-scale error were externally trimmed away. In practice, this is done by using input circuits that include gain and offset trimming. In addition, V_{REF} on the 80C196KC/KD could also be closely regulated and trimmed within the specified range to affect full-scale error.

Other factors that affect a real A/D converter system include temperature drift, failure to completely reject unwanted signals, multiplexer channel dissimilarities, and random noise. Fortunately, these effects are small.

Temperature drift is the rate at which typical specifications change with a change in temperature. These changes are reflected in the *temperature coefficients*.

Unwanted signals come from three main sources: noise on V_{CC}, input signal changes on the channel being converted (after the sample window has closed), and signals applied to channels not selected by the multiplexer.

Finally, multiplexer on-channel resistances differ slightly from one channel to the next, which causes *channel-to-channel matching* errors and *repeatability* errors; differences in DC leakage current from one channel to another and random noise in general contribute to repeatability errors.

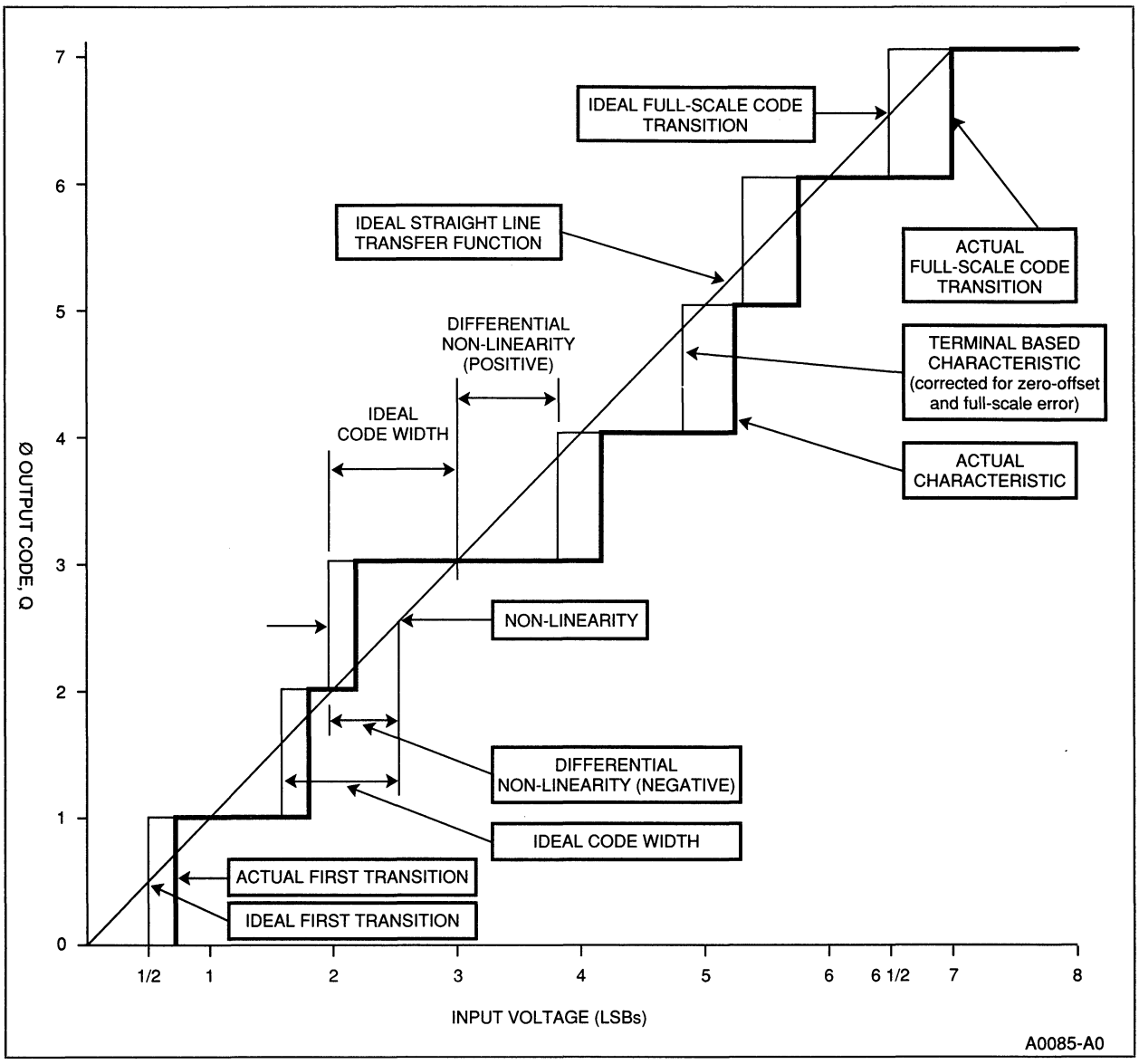

Figure 9-10. Terminal-Based A/D Conversion Characteristics