TB3194

PIC16/PIC18 ADC² Technical Brief

Introduction

Author: Christopher Best, Microchip Technology Inc.

An Analog-to-Digital Converter (ADC) converts an analog input signal into a digital number representing the magnitude of the input voltage. Microchip's 12-bit ADC with Computation (ADC^2) outputs a 12-bit binary representation of the original signal and adds special hardware features to provide post-processing functions that can be performed on the conversion result.

This technical brief provides an overview of the basic ADC features and functions, and describes the additional computation features that are not found in other ADC modules. This document will not cover the Capacitive Voltage Divider (CVD) feature, which is covered in TB3198, but may refer to CVD in general terms. This technical brief will cover the ADC^2 module found in Microchip's PIC16 and PIC18 8-bit architecture.

Table of Contents

1. ADC Module Overview

The ADC^2 module consists of two main blocks: the acquisition/conversion block (ADC) and the computation block. The ADC block reads an analog signal and converts it into a digital number, while the computation block takes the converted digital number and applies post-processing functions.

The ADC block provides the following features:

- 13-bit Acquisition Timer
- Automatic Repeat and Sequencing:
  – Two result registers
  – Auto-conversion trigger
  – Automated double sample conversion for CVD
- Hardware Capacitive Voltage Divider (CVD) Support:
  – 13-bit Precharge Timer
  – Adjustable sampling capacitor array
  – Guard ring digital output drive

The Computation block provides the following features:

- Averaging and Low-Pass Filter Functions
- Reference Comparison
- 2-level Threshold Comparison
- Selectable Interrupts

1.1 ADC Block

The ADC block contains all the circuitry needed to convert an analog input signal into a digital representation of that signal. Analog input channels are multiplexed into the converter's single sampling circuit. The converter's sampling circuit generates a 12-bit binary result via successive approximation and stores the conversion result into the ADC result registers.

1.1.1 ADC Inputs

The ADC can be used to convert both analog and digital signals. The signals may come from external sources via a port pin, or from internal sources. The ADC Positive Channel Selection (ADPCH) register controls the input multiplexer and determines which analog input channel is connected to the sampling circuit.

There may be several external input channels depending on the pin count of the selected device, but only one channel, either internal or external, can be selected and converted at a time. To determine the number and location of ADC channels in any PIC® device, refer to the specific device data sheet. Many data sheets list the location of each channel in the 'Pin Diagrams' charts. Each external channel is labeled 'ANxy', where 'x' refers to the I/O port, and 'y' refers to the pin within the port. For example, an external input signal, ANA1, points to the analog channel of PORTA, pin one.

Older data sheets, or data sheets for low pin count devices may label each external channel simply as 'ANx', where 'x' is the input channel number and may or may not refer to the actual port pin. For example, a 16-lead device may use the label AN1, but AN1 may point to PORTA's pin number two.

When converting an analog input signal, the I/O pin must be configured as an input by setting the port pin's TRIS bit corresponding to the selected input channel. Additionally, the pin must be configured as an analog input by setting the port pin's ANSEL bit. It is important to note that analog voltages on any pin that is not defined as an analog input may cause the input buffer to conduct excess current.

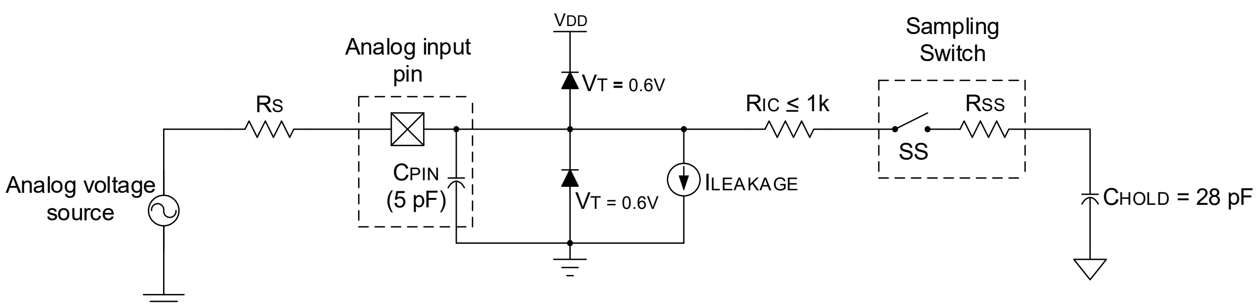
When converting a digital signal, care must be taken when considering the ANSEL selection. If the ADC is to convert a digital input signal, the port pin's ANSEL bit should be cleared. The ADC can also read and convert a digital output signal. In this case, the ANSEL bit can be cleared.

When switching between input channels, an acquisition delay is required before starting the next conversion. The acquisition delay is required to meet the ADC's specified accuracy. The delay allows time for the sampling capacitor to fully charge to the input voltage level.

External source impedance (R_S), internal interconnect impedance (R_{IC}), and sampling switch impedance (R_{SS}) have a direct effect on the acquisition time (see figure below). Microchip typically recommends a maximum source impedance of no greater than 10 kΩ. The device-specific recommended maximum source impedance can be found in the device data sheet's electrical specifications. Higher impedance values require a longer acquisition time, and, as impedance is decreased, acquisition time may be decreased.

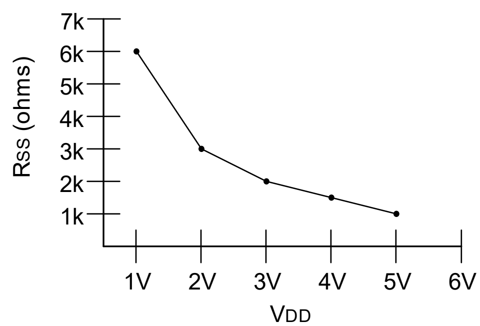
Refer to Equation 1-1 to calculate the minimum acquisition time. This equation assumes that the sample and hold capacitor is charged to within one-half of a Least Significant bit (LSb), which is the maximum error allowed for the ADC to meet its specified resolution.

Figure 1-1. Analog Input Model



LEGEND:

CHOLD = Sample/Hold capacitance
CPIN = Input pin capacitance
ILEAKAGE = Pin leakage current
RS = External input impedance
RIC = Internal interconnect impedance
RSS = Sampling switch impedance
SS = Sampling switch
VT = Protection diode voltage threshold



Equation 1-1. Acquisition Time Example

$$T_{ACQ}(MIN) = T_{AD} + T_C + T_{COFF}$$

Where:

T_{AD} = ADC example clock period (2 us)

T_{COFF} = Temperature Coefficient = (Temperature - 25°C) * $\left(\frac{0.002\text{ us}}{°\text{C}}\right)$

T_C = Hold Capacitor Charging Time = $T_C = -C_{HOLD}\left(R_{IC} + R_{SS} + R_S\right) * \ln\left(\frac{1}{\left(2^{N+1}\right) - 1}\right)$

Where:

C_{HOLD} = sample and hold capacitor value (typical 28 pF)

R_{IC} = internal interconnect impedance (1 kΩ typical)

R_{SS} = internal sampling switch impedance (varies depending on V_{DD}, see Figure 1-1)

R_S = external analog source impedance

Example:

Find T_{ACQ}(MIN) when the temperature is 30°C, V_{DD} = 5V, ADC resolution = 12 bits, and input impedance is 1 kΩ:

First, T_C is unknown, so it must be calculated:

$$T_C = -28\,pF\left(1\,k\Omega + 1\,k\Omega + 1\,k\Omega\right) * \ln\left(\frac{1}{\left(2^{12+1}\right) - 1}\right) = -28e - 12\left(3\,k\Omega\right) * \ln\left(\frac{1}{8191}\right) = 756\,ns$$

Next, calculate T_{ACQ}:

$$T_{ACQ}\left(MIN\right) = 2\,us + 756\,ns + \left(\left(30°C - 25°C\right) * \left(\frac{0.002\,us}{°C}\right)\right) = 2.77\,us$$

Since the example ADC clock period is 2.0 us and the T_{ACQ} is 2.77 us, it will take two ADC clock periods to complete the conversion. The ADC Acquisition Time Control Register pair (ADACQH:ADACQL) can now be loaded with the register value corresponding to two ADC clock periods.

1.1.2 ADC Voltage Reference

The ADC[2] module's positive and negative voltage references are controlled by the ADC Reference Selection (ADREF) register. The difference between the positive and negative reference voltages is referred to as the reference voltage V_{REF}. Positive reference voltage sources are selected via the PREF<1:0> bits of the ADREF register. Typical positive voltage sources include:

- Internal connection to the Fixed Voltage Reference (FVR) module
- Internal connection to device V_{DD}
- External connection to the V_{REF+} pin

The FVR module produces a fixed 1.024V, 2.048V, or 4.096V level, which can be internally routed as either the positive voltage reference for the ADC or as an input channel to the ADC. It is important to note that the FVR output cannot exceed device V_{DD}, and V_{DD} must be high enough to support the requested FVR voltage. In other words, the 4.096V FVR level is only available on 'F' rated devices operating at or above 4.5V, while the 2.048V FVR level is available on both 'LF' and 'F' rated devices operating at or above 2.5V. If selected as the voltage reference source, the FVR module must also be configured and enabled. See the specific device data sheet for more information on the FVR.

The V_{REF+} pin allows an external voltage to be used as the positive voltage reference. Positive voltages can range from 0V to device V_{DD}.

Negative reference voltage sources are selected via the NREF bit of the ADREF register. Typical negative voltage sources include:

- External connection to the V_{REF-} pin

- Internal connection to device V_{SS}

The V_{REF-} pin allows an external voltage to be used as the negative voltage reference. Negative voltages can range from 0V to 1V.

It is important to note that the positive reference voltage should always be greater than the negative voltage reference by at least 1.8V. Since all ADC conversions are performed with respect to V_{REF}, it is critical to ensure that the V_{REF} sources are stable to avoid incorrect conversion results. Additionally, the V_{REF} source impedance must be extremely low.

1.1.3 ADC Charge Pump

The ADC module includes a dedicated charge pump. The purpose of the charge pump is to ensure that the ADC internal logic remains at a constant voltage level, which helps ensure consistent ADC operation, especially at lower device operating voltages.

The ADC charge pump is enabled/disabled via the Charge Pump On (CPON) bit of the ADC Charge Pump Control (ADCP) register. Once the charge pump is enabled (CPON = `1`), the pump must undergo a stabilization period, which may take up to 35 us. Once the pump's output has stabilized and is ready for use, the Charge Pump Ready Status (CPRDY) bit is set.

It should be noted that the charge pump consumes additional current when enabled. When device V_{DD} is above 3.5V, internal voltage levels are sufficient for consistent ADC operation; therefore, the use of the charge pump will not be effective in improving ADC performance, but will continue to draw current.

1.1.4 ADC Conversion Clock

The ADC conversion clock is used to generate conversion timing. The conversion clock source comes from either the system clock source (F_{OSC}) or the dedicated ADCRC clock source. The ADC Clock Selection (CS) bit of the ADC Control Register 0 (ADCON0) determines which clock source is used by the module.

When the F_{OSC} is selected as the conversion clock source, the conversion clock frequency is determined by the ADC Conversion Clock Select (CS) bits of the ADC Clock Selection (ADCLK) register. The ADCLK register acts as a prescaler for the F_{OSC}, dividing the clock to a frequency that meets the ADC clock period (T_{AD}) specification. A T_{AD} cycle is defined as the time to complete a single bit conversion. Table 1-1 illustrates the possible T_{AD} periods based on the ADCLK configurations and system clock frequencies.

Table 1-1. ADC Clock Period (T_{AD}) When F_{OSC} Is The Clock Source

ADCLK CS<5:0>	Device Frequency (F_{OSC})						
	64 MHz	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}
`000000` (F_{OSC}/2)	31.25 ns	62.5 ns	100 ns	125 ns	250 ns	500 ns	2 us
`000001` (F_{OSC}/4)	62.5 ns	125 ns	200 ns	250 ns	500 ns	1 us	4 us
`000010` (F_{OSC}/6)	93.75 ns	187.5 ns	300 ns	375 ns	750 ns	1.5 us	6 us
`000011` (F_{OSC}/8)	125 ns	250 ns	400 ns	500 ns	1 us	2 us	8 us
…	…	…	…	…	…	…	…
`000111` (F_{OSC}/16)	250 ns	500 ns	800 ns	1 us	2 us	4 us	16 us

ADCLK CS<5:0>	Device Frequency (F_{OSC})						
	64 MHz	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}	T_{AD}
...	...	...	...	...	...	...	...
`001111` (F_{OSC}/32)	500 ns	1 us	1.6 us	2 us	4 us	8 us	32 us
...	...	...	...	...	...	...	...
`011111` (F_{OSC}/64)	1 us	2 us	3.2 us	4 us	8 us	16 us	64 us
...	...	...	...	...	...	...	...
`111111` (F_{OSC}/128)	2 us	4 us	6.4 us	8 us	16 us	32 us	128 us

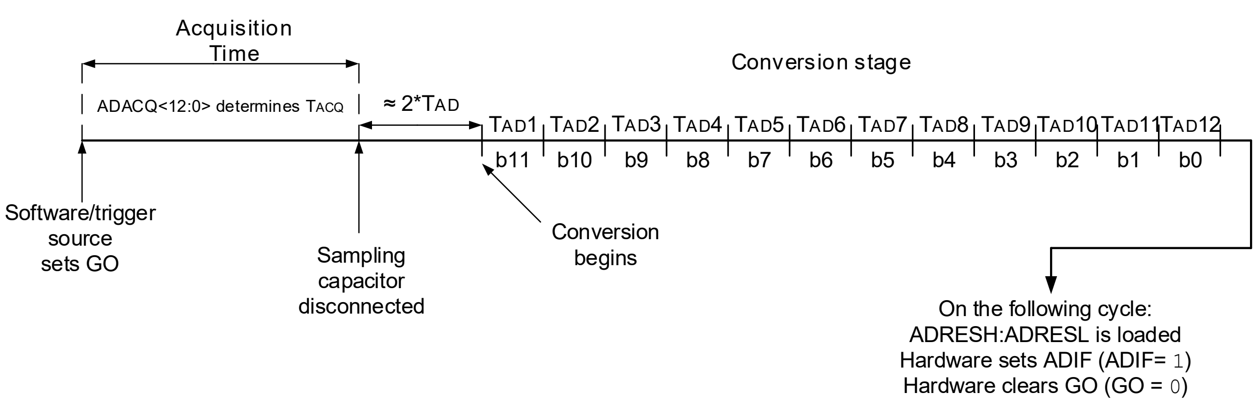
Note: Shaded cells violate T_{AD} requirements.

When the dedicated ADCRC clock is selected as the ADC conversion clock source, the conversion clock operates at a nominal 600 kHz clock frequency. The ADCRC can be used in applications that do not require high speed conversions. The ADCRC allows the ADC to operate in Sleep mode, which is great for low-power applications. The ADCRC produces a range of T_{AD} times which vary from 1.0 to 3.0 us.

To ensure correct conversion results, the appropriate T_{AD} requirements must be met. Typically, one T_{AD} is required for each bit conversion, with an additional two T_{AD} cycles required to cover the time elapsed from the disconnection of the sampling capacitor to when the conversion actually begins (see figure below).

It is important to note that when using F_{OSC} as the clock source, any changes in the F_{OSC} frequency will also change the ADC clock frequency, which may cause erroneous conversion results. The F_{OSC} allows for faster T_{AD} cycles, which result in faster conversion times, but cannot operate in Sleep mode.

Figure 1-2. Analog-to-Digital Conversion Cycles (12-Bit ADC)



1.1.5 Continuous Sampling Mode

The ADC Continuous Operation Enable (CONT) bit of ADCON0 enables/disables Continuous Sampling mode. In this mode, hardware automatically retriggers a new conversion cycle after computation and threshold tests have completed. If the result of the threshold test causes the ADTIF bit to become set, and the ADC Stop-on-Interrupt (SOI) bit is set, module hardware will clear GO bit and the conversion will stop. Conversions may also be halted by clearing GO bit in software.

1.1.6 Double Sample Conversion Mode

Double sampling is enabled by setting the Double Sample Enable (DSEN) bit of the ADCON1 register. When double sampling is enabled, two consecutive conversions are acquired, and the resulting final conversion value is the difference between the second sample and the first (S2 – S1).

When Continuous Sampling mode is enabled (CONT = `1`), both conversions are completed automatically, requiring only one trigger event to capture both conversions. The GO bit is maintained by hardware between conversion cycles, and is cleared by hardware after both conversions are complete. When CONT = `0`, two conversion trigger events are required to capture both conversions. The GO bit is cleared between each conversion.

The first completed conversion (S1) is written into ADRES. Once the second conversion has completed, the new (second) conversion (S2) is stored in ADRES, and if the ADC Previous Sample Input Select (PSIS) bit of ADCON2 is clear, the first completed conversion is transferred to the ADPREV register pair. If the PSIS bit is set, the ADFLTR value is loaded into ADPREV instead of the first conversion. At this point, module hardware calculates the difference between the two conversions (S2 - S1), adds the difference to the accumulator, and performs a threshold test on the updated accumulator value (except in Basic mode).

1.1.7 Auto-Conversion Trigger

The auto-conversion trigger allows periodic ADC measurements without software intervention. Auto-conversion trigger sources may be internal, such as the overflow of a Timer, or from an external source connected to the ADACTPPS input pin. Auto-trigger sources are selected using the ADC Auto-Conversion Trigger Control (ADACT) register.

The Direct Memory Access (DMA) module may be used as an auto-conversion trigger. Since the DMA has the ability to read/write any SFR, it can be used to trigger an auto-conversion by writing to the ADC Positive Channel Selection (ADPCH), or by reading the ADC Result High (ADRESH) or ADC Error High (ADERRH) registers. For more information on the DMA, please refer to the selected device data sheet.

When a trigger event occurs, module hardware sets the GO bit and the conversion process begins. Once the conversion is complete, hardware clears the GO bit, loads the ADRESH:ADRESL registers with the conversion result, and sets the ADC Interrupt Flag (ADIF).

Auto-conversion trigger sources may or may not be synchronized to the ADC clock; therefore, it is important to assure that all ADC timing requirements are met. If a trigger is received during an active conversion cycle (GO = `1`), the trigger is ignored and has no effect on the current conversion cycle.

1.1.8 Conversion Result Formatting

The ADC result can be calculated using equation below.

Equation 1-2. ADC Result Calculation

$$\text{ADRESH:ADRESL} = \frac{2^N - 1}{V_{REF}} * \left(V_{IN} - V_{REF-}\right)$$

Where:

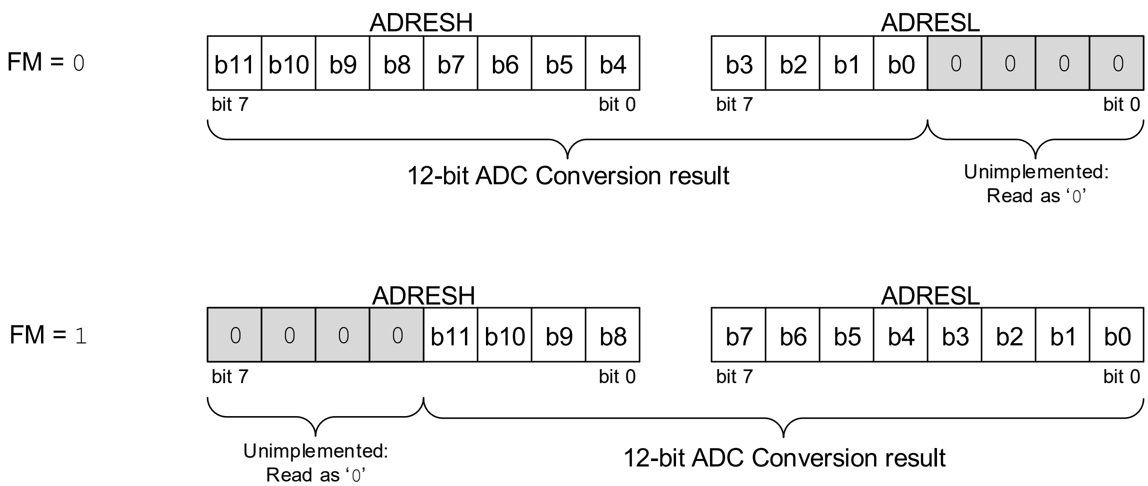
N = number of ADC bits

$V_{REF} = V_{REF+} - V_{REF-}$

V_{IN} = Analog input voltage

When an ADC conversion is complete, the results are stored in the ADRESH:ADRESL register pair. The results are supplied to the register pair in one of two formats, either left or right justified (see figure below). Result formatting is controlled by the ADC Results Format/Alignment Selection (FM) bit of the ADCON0 register. The selected format for the ADRESH:ADRESL register pair also applies to the ADC Previous Result (ADCPREVH:ADCPREVL) register pair.

Figure 1-3. 12-Bit ADC Conversion Result Format



1.1.9 ADC Conversion Interrupt

The ADC module can generate an interrupt upon the completion of an ADC conversion. The ADC Interrupt Flag (ADIF) bit becomes set every time a conversion cycle has completed, regardless of the state of the ADC Interrupt Enable (ADIE) bit. The ADIF bit must be cleared by software.

The ADC Interrupt can be generated while the device is operating or while in Sleep mode. If the device is in Sleep mode, the interrupt will wake the device. It is important to note that the ADC can only operate in Sleep mode when the FRC is selected as the ADC clock source.

1.1.10 Sleep Mode

The ADC has the ability to operate in Sleep mode, but requires the ADC to use the dedicated ADCRC as its clock source. When the ADCRC is selected as the clock source, ADC hardware waits one additional instruction cycle (T_{CY}) before starting the conversion. This allows the `SLEEP` instruction to be executed, which may reduce system noise during the conversion process.

If the ADC interrupt is enabled (ADIE = `1`), the device will wake up from Sleep immediately following the completed conversion. If the ADC interrupt is disabled (ADIE = `0`), the module is shut off after the conversion completes, although the ON bit remains set.

If an auto-conversion trigger is invoked during Sleep and the ADCRC is the clock source, the ADC will perform the conversion, set ADIF, and may wake the device from Sleep.

If an auto-conversion trigger is invoked during Sleep and the F_{OSC} is the clock source, the trigger will be recorded, but the conversion will not begin until the device exits Sleep via an interrupt. It is important to note that some trigger sources may have interrupt features built in. If the trigger source's interrupt is enabled and the trigger source is invoked while in Sleep, internal functions, such as oscillator start-up, may result in a slight time delay and additional system noise, which can directly affect the ADC result. Disabling the trigger source's interrupt or choosing a different trigger source can prevent the ADC from waking up during a conversion.

The ADC module is not affected by either Idle or Doze modes which are available for use with both F_{OSC} and ADCRC clock sources. Idle or Doze modes may be used instead of Sleep mode to reduce the effects of system noise.

1.2 Computation

The ADC^2 module features post-conversion computation. After an ADC conversion has completed, the result can be passed through one of the computation functions. The computation mode can be selected by the ADC Operating Mode Selection (MD<2:0>) bits of the ADCON2 control register.

The computation modes include:

- Basic
- Accumulate
- Average
- Burst Average
- Low-Pass Filter

1.2.1 Basic Mode

Basic mode disables all additional computation features. This mode closely resembles a typical ADC module (without computation), and is considered the Legacy mode. Threshold comparison tests are still performed, which may or may not set the ADTIF flag, and the Double Sampling mode, Continuous mode, and all CVD features are still available. No accumulation occurs, and no features involving the digital filter or average features are used. The auto-conversion trigger feature is still available in Basic mode.

1.2.2 Accumulate Mode

In Accumulate mode, each new conversion is added to the ADC Accumulator Register trio (ADACCU:ADACCH:ADACCL) in addition to the ADRES register pair. A threshold comparison is performed on each new sample, and may set the ADTIF interrupt flag.

After each conversion, the result is added to the Accumulator, the threshold test is performed, and the ADC Count Register (ADCNT) is increased by one. The ADCNT register holds the number of conversion results that have been added to the Accumulator, up to a count of 255. It is important to note that ADCNT does not roll over, and any additional accumulation past 255 counts will not be tracked, although the Accumulator will still add new conversion results until the Accumulator overflows.

If an overflow occurs (in any computation mode except Basic mode), the ADC Accumulator Overflow (AOV) bit of the ADSTAT register will be set. The ADC Accumulator Clear Command (ACLR) of the ADCON2 register is used to clear the Accumulator, ADCNT, and the AOV bit. When ACLR = `1`, the Accumulator, ADCNT, and the AOV bit are all cleared, but when using the ADCRC, this process may take a few instruction cycles to complete. When complete, hardware automatically clears ACLR.

The accumulated value can be right-shifted (divided) via the ADC Accumulated Calculation Right Shift Selection (CRS<2:0>) bits of the ADCON2 register. These bits allow the accumulated results to be right-shifted by the value of CRS (see Equation 1-3). The right-shifted result is stored in the ADC Filter Register pair (ADFLTRH:ADFLTRL), and is updated every conversion (see Table 1-2). It is important to note that if the right-shifted accumulator value exceeds the capacity of ADFLTR, or if the threshold computation overflows, the AOV bit will be set.

Equation 1-3. ADFLTR Register Pair Calculation

$$ADFLTRH:ADFLTRL = \frac{(ADACCU:ADACCH:ADACCL)}{2^{CRS}}$$

Table 1-2. Accumulate Mode Example

ADCNT	CRS<2:0>	V_{IN}	12-BIT ADRES	18-BIT ADACC	16-BIT ADFLTR	Threshold Test Performed?
1	4	2.5V	2047	2047	127	Yes
2	4	2.5V	2047	4094	256	Yes
3	4	2.5V	2047	6141	383	Yes
4	4	2.5V	2047	8188	511	Yes
5	4	2.5V	2047	10, 235	639	Yes
6	4	2.5V	2047	12, 282	767	Yes
7	4	2.5V	2047	14, 329	895	Yes
8	4	2.5V	2047	16, 376	1023	Yes

1.2.3 Average Mode

Average mode is very similar to Accumulate mode in the sense that the ADACC register trio accumulates with each new conversion, increments ADCNT, and updates the ADFLTR register pair with the right shifted value of ADACC. In Average mode, the number of accumulated conversion results depends on the value of the ADC Repeat Setting Register (ADRPT). This register holds the number of samples that are to be accumulated.

The value of ADRPT should be set based on the number of right shifts that will be performed to get the average of the accumulator data (see Equation 1-4). Once ADCNT is equal to ADRPT, a threshold test is performed on the ADFLTR value, and ADTIF may be set depending on the threshold settings. The next trigger event clears ADCNT and ADACC, and the conversion result is recorded as sample number one.

Table 1-3 gives an Average mode example. In this case, the CRS<2:0> bits of ADCRS are set to a value of 2. Based on Equation 1-4, the RPT<7:0> bits of ADRPT should be loaded with the value of 4. This means that four samples will be taken, and the accumulated value after the four samples are taken will be right shifted by two places, or a divide by 4.

Equation 1-4. Number of Samples Calculation

$$ADRPT = 2^{ADCRS}$$

Table 1-3. Average Mode Example

ADCRS	ADRPT	ADCNT	ADRES	ADACC	ADFLTR	Threshold Test Performed?
2	4	1	500	500	125	No
2	4	2	500	1000	250	No
2	4	3	500	1500	375	No
2	4	4	500	2000	500	Yes
2	4	1	500	500	125	No

1.2.4 Burst Average Mode

Burst Average mode is essentially the same as the Average mode, with one difference. In Burst Average mode, once the GO bit is set by software or an auto-trigger source, hardware continuously retriggers until

ADCNT is equal to ADRPT. At that point, the ADFLTR holds the average value of the samples acquired during the burst, a threshold test is performed on the ADFLTR value, and ADTIF may be set depending on the threshold settings.

The table below shows a Burst Average mode example. In this case, ADRPT is loaded with a value of 8 (based on Equation 1-4). That means that for each trigger event, the number of samples taken in each burst is equal to ADRPT, or 8. After all eight samples are accumulated, the ADACC register is right-shifted by the value of ADCRS, and the result transferred into ADFLTR.

Table 1-4. Burst Average Mode Example

Trigger	Samples	ADCNT	ADCRS	ADRPT	ADACC	ADFLTR
1	1-8	8	3	8	2000	250
2	9-16	8	3	8	2160	270
3	17-24	8	3	8	2080	260
4	25-32	8	3	8	2120	265
5	33-40	8	3	8	2136	267
6	41-48	8	3	8	2160	270
7	49-56	8	3	8	2136	267

1.2.5 Low-Pass Filter Mode

Low-Pass Filter (LPF) mode works in a similar fashion to Average mode, except that after an initial accumulation of samples, the module continues to acquire and accumulate samples indefinitely. LPF can be considered as having two main processes that work in succession - an initial average process followed by a continuous filtering operation.

The initial averaging process begins by accumulating samples until ADCNT is equal to ADRPT. During the initial process, each sample is added to the accumulator. The new accumulator value is right-shifted by the ADCRS value, and the result is loaded into ADFLTR. When ADCNT = ADRPT, a threshold test is performed on the ADFLTR value. This initial averaging process prevents threshold tests from being performed on each sample until after an average has been taken, which helps reduce ‘false alarm’ threshold violations due to random variations of a single sample. For the initial averaging process, ADRPT acts as a time constant, allowing the computed average to reach a steady state before threshold comparisons begins.

Once the initial averaging process completes, the module moves into continuous filtering operation. The module will then add the next conversion result to the accumulator to get a new accumulator value. Then, the previous accumulator value is right shifted by the ADCRS value, and then subtracted from the new accumulator value (see Equation 1-5).

Once these calculations have been performed, the shifted value is stored in ADFLTR as the filtered value, and a threshold test is performed. This process repeats for each new conversion. It is important to note that the accumulator is not cleared after the initial averaging process, or after any subsequent conversion, but instead continues to accumulate samples until software disables the module. During the continuous filtering operation, ADRPT is ignored, ADCNT continues to count (until ADCNT = 0xFF), and ADCRS continues to act as the accumulator divider.

The ADCRS value also influences the filter performance. When ADCRS is a low value, the ADFLTR output reaches a steady state very quickly, but any deviations from the averaged value make a noticeable difference on the filtered output. As the ADCRS value increases, the time it takes for the ADFLTR output to achieve a steady state increases, but the effects of any deviations from the overall average have less of an impact on the filtered output (see figure below).

Table 1-5 shows the effects of ADCRS on the ADFLTR output. In this comparison, the ADRES values are centered around a value of 200. At random sample points (shaded), the ADRES values are changed to simulate an unwanted noise component that the ADCC acquired. When the ADCRS bits are set to '6', the 'noise' does not have much of an effect on the filtered output. Conversely, when ADCRS is set to '1', the 'noise' has much more of an impact on the filtered output.

Essentially, when ADCRS values are higher, the effects of noise on the output are reduced, but sudden changes in the input may take longer to influence the output. When the ADCRS values are lower, the effects of noise have a larger impact on the filtered output, but sudden changes would be detected quickly.

Equation 1-5. ADFLTR Calculation in Low-Pass Filter Mode

$$ADFLTR = \frac{ACC_{NEW}}{2^{ADCRS}}$$

Where:

$$ACC_{NEW} = (ACC_{PREV} + ADRES) - \frac{ACC_{PREV}}{2^{ADCRS}}$$

ACC_{PREV} = Previous accumulator result

ADRES = Current conversion result

Figure 1-4. ADCRS Effects on ADFLTR Output

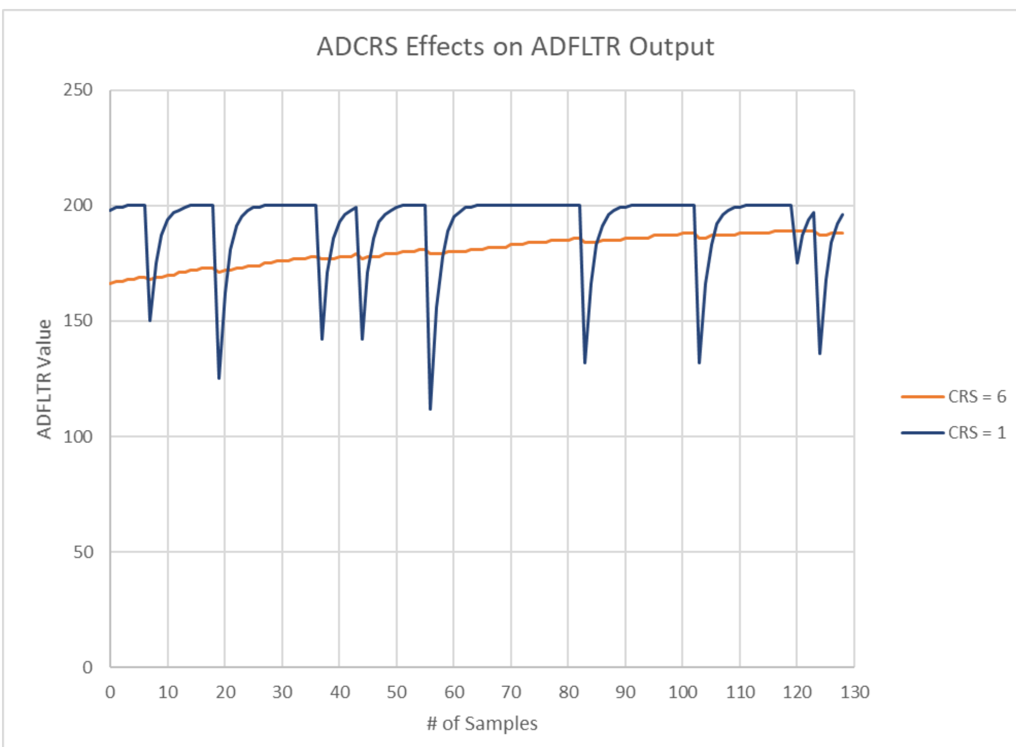


Table 1-5. Effects of ADCRS Values on ADFLTR

Sample#	ADRES	ADFLTR Output	
		ADCRS = 6	ADCRS = 1
0	200	166	198
1	200	167	199
2	200	167	199
3	200	168	200
7	100	168	150
8	200	169	175
17	200	173	200
18	200	173	200
19	50	171	125
20	200	172	162
36	200	178	200
37	85	177	142
38	200	177	171
43	200	179	199
44	85	177	142
45	200	178	171
55	200	181	200
56	25	179	112
82	200	186	200
83	64	184	132
128	200	188	196

1.2.5.1 ADCRS Effects on -3dB Roll-Off Frequency

In Low-Pass Filter mode, the ADCRS value also determines the -3dB roll-off frequency of the single-pole filter. The table below shows the radian values at the -3dB roll-off frequency based on ADCRS values.

Table 1-6. Radian Values at -3dB Roll-Off

CRS	RPT	Radians @ -3dB Roll-Off
1	2	0.72
2	4	0.284
3	8	0.134
4	16	0.065

CRS	RPT	Radians @ -3dB Roll-Off
5	32	0.032
6	64	0.016

The radian values listed in the table above are defined by the ADCC's hardware. These values are used to calculate the -3dB roll-off point in terms of frequency. The following equation can be used to determine the -3dB point; however, there is one fundamental part of this equation that can cause confusion.

Equation 1-6.

Frequency @ -3dB roll-off point = $\frac{Radians@-3dB}{2\Pi T}$

Where:

Radians @ -3dB = the value from the table above based on the CRS value

T = total sampling time

The 'T' term indicates the total sampling time. The total sampling time is the measured time between samples. The total sampling time is critical since it is the actual time it takes to acquire a single filtered conversion result.

The ADC's sampling rate is only part of the total sampling time necessary to properly calculate the roll-off frequency. We know that the ADC's sampling frequency influences the ADC result. What may not be known is that the number of instructions contained in the ADC routine also influences the total sampling time. Once the ADC's conversion result has been acquired, the result must still pass through the filter. The conversion result may need to be sent to the DAC to output the filtered waveform, or sent to a logging file via a serial port. For example, if the ADC routine transmits the filtered result to the UART using 'printf' commands, the total sampling time will be longer than if the filtered value was 'manually' written to the UART transmit buffer. The total sampling time includes the ADC acquisition time, the conversion time, interrupt time, and any output transmission time.

The table below shows the difference in roll-off frequencies based on the sampling time 'T'. In this example, the ADCC acquires each sample in the same manner; however, the methods used to transmit data over the UART are different. One method uses 'printf' statements, which are easy to use, but at the expense of additional instruction time. The other method loads the UART TX buffer with the filtered results through software instructions, which is slightly more cumbersome, but require fewer instructions than 'printf' statements.

When shorter total sampling times are desired, consider the following:

- System clock (F_{OSC}) - when used as the ADC clock, the system clock determines the T_{AD} period
- Number of instructions - every instruction in the ADC routine takes time to execute, which adds to the total sampling time
- Number of instructions in the ISR - interrupt routines should typically be as short as possible
- Type of instructions - as previously mentioned, using the 'printf' library function may be very easy to use, but at the expense of additional instruction cycles
- ADC acquisition time - faster acquisition times reduce total sampling time

Table 1-7. Effects of Sampling Time on Roll-Off Frequency

CRS	Radians @ -3dB Cut-Off	Method Using 'printf'		Method Using Direct UART Writes	
		Measured Sampling Time (us)	Calculated Frequency @ -3dB Point (Hz)	Measured Sampling Time (us)	Calculated Frequency @ -3dB Point (Hz)
1	0.72	520.0	220.37	435.0	263.43
2	0.284	520.0	86.92	435.0	103.91
3	0.134	520.0	41.01	435.0	49.03
4	0.065	520.0	19.89	435.0	23.78
5	0.032	520.0	9.79	435.0	11.71
6	0.016	520.0	4.90	435.0	5.85

One way to measure the total sampling time would be to use the Stopwatch function built in to the MPLAB® X debugger. This is accomplished by placing a breakpoint at the beginning and at the end of the ADC routine. The debugger will calculate the amount of time it takes to execute the ADC function in its entirety, including any interrupts. Of course, there are other ways to calculate the routine's time, such as toggling a pin at the beginning and end of the routine and measuring the time in between pin states, or using a timer that is enabled at the beginning of the routine and stops at the end of the routine.

LPF EXAMPLE

This example illustrates the expected output of the ADCC using the Low-Pass Filter function with a CRS value of '`1`'. For this example, the 'Method Using Direct UART Writes' (table above) is used since it has the fastest total sampling time, and gives a Nyquest limit of approximately 1.15 kHz.

A function generator is configured such that its output is 50 Hz sinewave, with a peak-to-peak value of 2 volts. The sinewave is offset by 1500 mV so that the voltage ranges from 500 mV to 2.5V because the ADC cannot read voltages below the negative reference voltage. The output of the function generator is connected to an analog input of the PIC18F26K42 microcontroller.

ADC Threshold interrupts are set to always interrupt after the completion of each sample.

The filtered result is copied to the UART, which sends the results to the Data Visualizer plug-in feature of the Atmel Studio 7 IDE. The Data Visualizer accepts serial data and, amongst other features, converts the data back into an analog equivalent that is shown on its built-in oscilloscope.

Figure 1-5 shows a 50 Hz sinewave reconstructed by the Data Visualizer. With the CRS value at 1, and the sample time equal to 435 µs, the 50 Hz signal is well below the expected 263 Hz roll-off point. As the sinewave's frequency is increased, once it reaches approximately 270 Hz, a reduction in peak-to-peak voltage takes place as the filter actively reduces the magnitude of the signal, as observed in Figure 1-6. As the frequency continues to increase, the peak-to-peak range will shrink, as observed in Figure 1-7.

Figure 1-5. Reconstructed Sinewave at 50 Hz

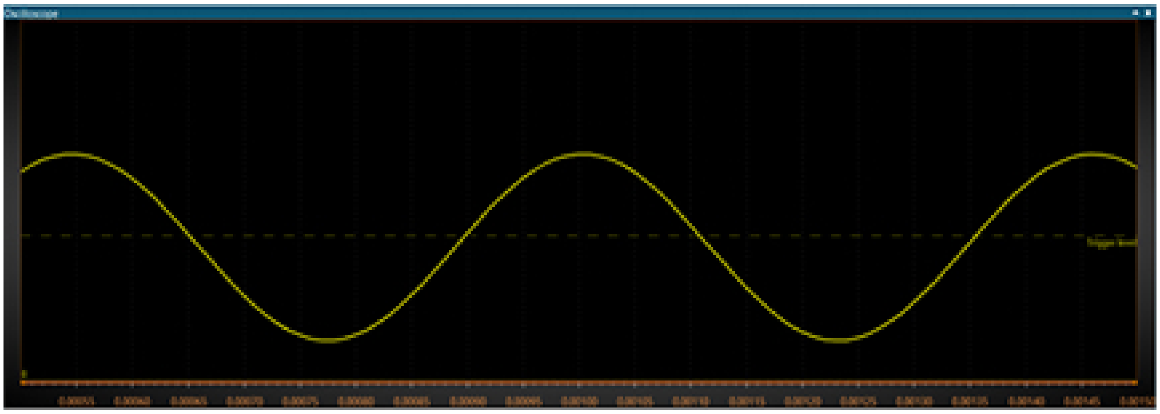


Figure 1-6. Reconstructed Sinewave at 270 Hz

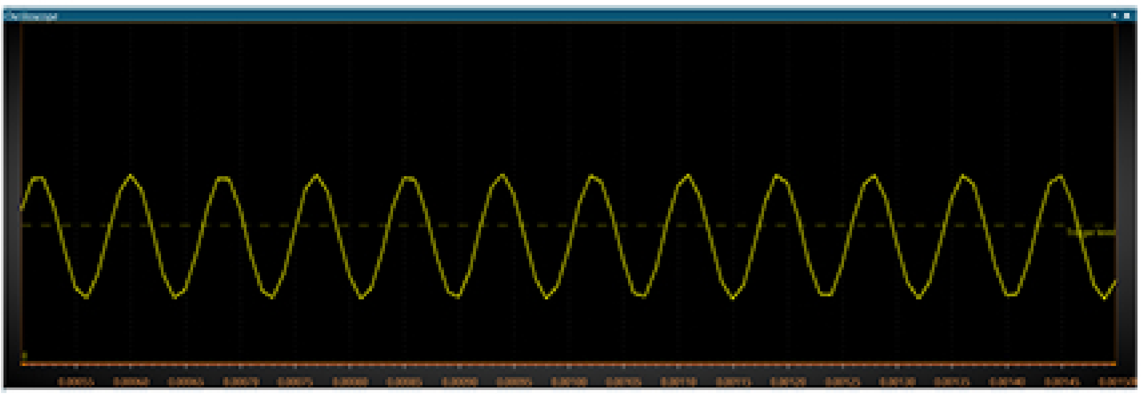
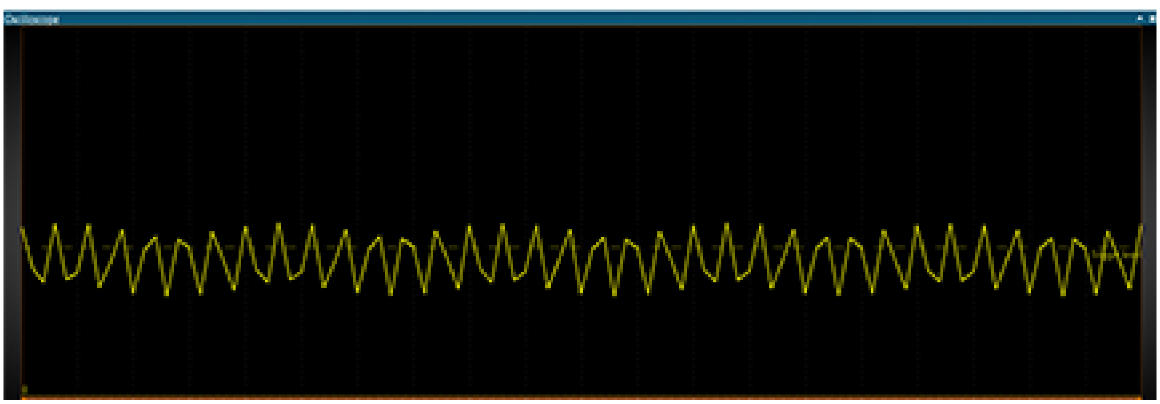


Figure 1-7. Reconstructed Sinewave at 800 Hz



1.3 Threshold Comparison

After the ADC completes a conversion, the result is stored in the ADRES register pair. If there is a result currently in the ADRES register pair, it is transferred into the ADPREV register pair, and the new conversion result is stored in the ADRES register pair. After each sample in Basic or Accumulate modes, or once ADCNT is equal to ADRPT in Average, Burst-Average, or LPF mode, an error calculation is performed based on the configuration of the ADC Error Calculation Mode Select (CALC<2:0>) bits of the ADC Control Register 3 (ADCON3).

The Error Calculation modes include:

- The difference of consecutive measurements
- The actual result compared to a setpoint
- The actual result compared to an averaged/filtered value
- The difference of filtered values
- The averaged/filtered value compared to a setpoint

Depending on the Error mode selection, error calculations may involve the following registers:

- ADC Result Register pair (ADRESH:ADRESL)
- ADC Previous Result Register pair (ADPREVH:ADPREVL)
- ADC Threshold Setpoint Register pair (ADSTPTH:ADSTPTL)
- ADC Filter Register pair (ADFLTRH:ADFLTRL)

The DSEN bit determines the number of conversions needed before the module begins the calculations and threshold comparison.

When DSEN is set, the module is in Double Sampling mode. Two conversion results are required before the module begins its error calculations and threshold comparison tests. If the ADC Continuous Operation Enable (CONT) bit of ADC Control Register 0 (ADCON0) is set, the ADC Conversion Status (GO) bit is held by hardware until two consecutive conversions are recorded. If the CONT bit is clear, the GO bit is cleared after each conversion, meaning that software or an external trigger must set the GO bit to trigger the second conversion.

The first conversion is loaded into the ADRES register pair, the ADC Module Computation Status (MATH) bit of the ADC Status Register (ADSTAT) is set, and the ADC Accumulator Register trio (ADACCU:ADACCH:ADACCL) is updated, but the module will not calculate the error or set the ADC Threshold Interrupt (ADTIF) bit. After the second conversion is complete, the first conversion result is transferred to the ADPREV register pair and the second result is loaded into the ADRES register pair. At this point, the error is calculated and the threshold comparison test is performed.

When DSEN is clear, a single conversion takes place each time the GO bit is set. In this case, only a single conversion is required before hardware begins the error calculation and threshold comparison. Once the new conversion is complete, the new result is stored in the ADRES register pair, and the previous conversion result is transferred to ADPREV to allow difference calculations to be performed. If CONT is set, the module will automatically reset the GO bit after each conversion. If CONT is clear, software must set GO before another conversion begins.

Once the error calculation is complete, hardware transfers the result into the ADC Error Register pair (ADERRH:ADERRL). The error is then compared to the ADC Upper Threshold Register (ADUTHH:ADUTHL) pair and the ADC Lower Threshold Register (ADLTHH:ADLTHL) pair. The threshold registers hold the user-defined threshold values that are used for error comparison. If the error value is greater than the upper threshold value, the ADC Module Greater-than Upper Threshold Flag (UTHR) bit of the ADC Status Register (ADSTAT) is set. If the error is less than the lower threshold value, the ADC Module Less-than Lower Threshold Flag (LTHR) bit of ADSTAT is set.

When a threshold comparison is made, an interrupt may be generated. The Threshold Interrupt Mode Select (TMD<2:0>) bits of ADCON3 select which interrupt condition to test for. The Interrupt modes include:

- Interrupt regardless of the threshold test results
- Interrupt if the error is greater than the upper threshold (also sets UTHR)

- Interrupt if the error is less than or equal to the upper threshold
- Interrupt if the error is less than the lower threshold OR greater than the upper threshold
- Interrupt if the error is greater than the lower threshold AND less than the upper threshold
- Interrupt if the error is greater than or equal to the lower threshold
- Interrupt if the error is less than the lower threshold (also sets LTHR)
- Never interrupt

If the selected interrupt condition is met, the ADC Threshold Interrupt Flag (ADTIF) bit is set, and if the ADC Threshold Interrupt Enable (ADTIE) bit is set, an interrupt will be generated. Software must clear ADTIF.

2. Conclusion

This technical brief describes basic ADC operation as well as the computation block features found in Microchip's Analog-to-Digital Converter with Computation (ADC^2) module. Code examples can be found at http://www.microchip.com/mplab/mplab-xpress.

3. Appendix A: Basic Analog Terminology

To better understand the specifications of the ADC[2], it is important to understand some basic terminology that may be used to describe the operation of the ADC or the electrical parameters that govern the module.

Full-Scale Range: The operating voltage range between V_{REF-} and V_{REF+}.

Successive Approximation Register (SAR): Microchip's PIC16 and PIC18 devices employ the Successive Approximation Register (SAR) type of ADC. This type of ADC converts a continuous analog input into an approximate digital representation using a binary search algorithm. The entire SAR conversion process is performed in hardware, so no additional conversion software is needed.

The SAR ADC uses a sampling capacitor to compare the input voltage to the reference voltage. The sampling capacitor voltage is compared to the output of an internal DAC via an internal comparator, which is connected to a successive approximation register. The successive approximation register begins the binary search by setting its Most Significant bit (MSb) to a '1', which forces the DAC output to be $V_{REF}/2$. The DAC output is compared to the analog input.

If the analog input is greater than the $V_{REF}/2$ DAC output, the comparator outputs a logic '1'; if the analog input is less than $V_{REF}/2$, the comparator outputs a logic '0'. The comparator output is then compared to the MSb of the successive approximation register. If the comparator output is a '1', the MSb of the SAR remains '1'; if the comparator output is '0', the SAR clears the MSb.

This process repeats for each bit until the LSb has been processed. Once the LSb has been processed, the conversion is complete, and the conversion result is transferred to the ADRES register pair.

Voltage Resolution: The minimum change in voltage required to ensure a change in the output code level. The voltage resolution of an ADC is equal to the full-scale voltage range of the ADC divided by the number of possible intervals. The number of possible intervals is determined by 2^N, where N is the number of ADC bits.

Digital Resolution: Digital resolution is defined in bits, and determines how many distinct output codes the converter can produce over a range of analog input voltages. Digital resolution is illustrated as 2^N, where N is the number of ADC bits. For example, a 12-bit ADC would produce 2^{12}, or 4096, possible output codes.

Acquisition Time: The time required for the ADC to capture the input voltage during sampling, also referred to as sampling time. Acquisition time for a Successive Approximation Register (SAR) ADC is the time required to charge the sampling capacitor (C_{HOLD}). Insufficient acquisition times may result in inaccurate conversion results.

Code Width: The distance between two transition points, expressed in LSb or voltage.

Monotonic: Any increase in the analog input voltage produces a greater digital output code value, while a decrease in analog input voltage produces a decreased code value.

Transition Point: The analog input voltage at which the digital output switches from one code to the next.

Offset Error: The difference between the measured first transition point and the ideal first transition point of the ADC transfer function expressed in LSb. Offset error can be corrected by subtracting the offset error from the conversion result. Offset error is calculated using the following equation:

Equation 3-1. Offset Error Calculation

$$ERR_{OFFSET} = \frac{V_{TRANS} - 0.5\,LSb}{LSb}$$

V_{TRANS} - The *measured* voltage at the first transition point

LSb - The *ideal* voltage of the first transition point calculated as $\frac{V_{REF}}{2^N}$

Where:

N = number of ADC bits

$V_{REF} = V_{REF+} - V_{REF-}$

Gain Error: The difference between the ideal full-scale range and the measured full-scale range expressed in percentage of the full-scale range. In other words, gain error is the difference between the slope of the ideal transfer function and the measured transfer function. Gain error can be corrected by multiplying each conversion result by the inverse of the gain error.

After correcting for gain and offset errors, the transfer function is considered normalized, and the corrected conversion results can be used to measure Integral Nonlinearity (INL) and Differential Nonlinearity (DNL) errors. Gain error is calculated using the following equation:

Equation 3-2. Gain Error Calculation

$$ERR_{GAIN} = \frac{\left(V_{REF} - 2\,LSb\right) - V\left[\left(2^N - 2\right) : \left(2^N - 1\right)\right] - V_{TRANS}}{LSb}$$

V_{TRANS} - The *measured* voltage at the first transition point

$V\left[\left(2^N - 2\right) : \left(2^N - 1\right)\right]$ - The *measured* voltage at the final transition point

LSb = $\frac{V_{REF}}{2^N}$

Where:

N = number of ADC bits

$V_{REF} = V_{REF+} - V_{REF-}$

Differential Nonlinearity (DNL) Error: The difference between a measured code width and the ideal value of one LSb. In an ideal ADC, when the DNL error is zero, each analog step equals one LSb, where one LSb is equal to the ratio of the reference voltage to the ADC resolution (see equation below). In this case, each transition is equally spaced one LSb apart.

DNL errors are calculated for each transition point, and the largest error is reported as the ADC's DNL. DNL errors are measured after the transfer function has been normalized. The possible range for DNL error values is ± 1 LSb. If the error is ≤ -1 LSb, there will be missing codes in the transfer function. If the error is zero, each LSb is considered ideal and no missing codes are reported in the transfer function. If the error is greater than zero but less than or equal to +1 LSb, a monotonic transfer function is guaranteed and there are no missing codes.

Equation 3-3. Differential Nonlinearity (DNL) Error Calculation

$$ERR_{DNL} = \frac{V_{OC+1} - V_{OC}}{V_{LSb-IDEAL}} - 1, \text{ where } 0 < OC < 2^N - 2$$

V_{OC+1} - Measured voltage value of the adjacent output code

V_{OC} - Measured voltage value of the current output code

$V_{LSb\text{-}IDEAL}$ - Ideal voltage value of one LSb

OC – ADC’s digital output code

LSb = $\frac{V_{REF}}{2^N}$ (IDEAL LSb value)

Where:

N = number of ADC bits

$V_{REF} = V_{REF+} - V_{REF-}$

Integral Nonlinearity (INL) Error: The difference between a measured transition point and the corresponding transition point on the ideal transfer curve with the offset and gain errors already corrected. Offset and gain errors *must* be normalized before measuring INL errors. INL errors are calculated for each transition point, and the largest error is reported as the ADC’s INL. INL errors are expressed in LSb, and are calculated using the following equation:

Equation 3-4. Integral Nonlinearity (INL) Error Calculation

$ERR_{INL} = \frac{V_{OC} - V_{ZERO}}{V_{LSb-IDEAL}} - OC$, where $0 < OC < 2^N - 1$

V_{OC} - Measured voltage value of the current output code

V_{ZERO} - Minimum analog input voltage which corresponds to an all-zero output code

$V_{LSb\text{-}IDEAL}$ - Ideal voltage value of one LSb

OC – ADC’s digital output code

LSb = $\frac{V_{REF}}{2^N}$ (Ideal LSb value)

Where:

N = number of ADC bits

$V_{REF} = V_{REF+} - V_{REF-}$

Absolute Error: The maximum deviation between any measured transition point and the corresponding ideal transfer function transition point. The absolute error includes the offset, gain, and INL errors, and defines the overall accuracy of the ADC. Offset and gain errors are not normalized when calculating the absolute error. Absolute error is calculated using the following equation:

Equation 3-5. Absolute Error Calculation

$$ERR_{ABS} = \frac{V_{OC-MEASURED} - V_{OC-IDEAL}}{V_{LSb-IDEAL}}$$

$V_{OC\text{-}MEASURED}$ - Measured voltage value of the current output code

$V_{OC\text{-}IDEAL}$ - Voltage value of the ideal corresponding output code

$V_{LSb\text{-}IDEAL}$ - Ideal voltage value of one LSb

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-3069-8

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820