



Section 17. 10-Bit A/D Converter

HIGHLIGHTS

This section of the manual contains the following major topics:

17.1	Introduction	17-2
17.2	A/D Terminology and Conversion Sequence	17-4
17.3	Registers	17-5
17.4	A/D Module Configuration	17-11
17.5	Initialization	17-14
17.6	Controlling the Sampling Process	17-15
17.7	Controlling the Conversion Process	17-15
17.8	A/D Results Buffer	17-21
17.9	Conversion Sequence Examples	17-23
17.10	A/D Sampling Requirements	17-31
17.11	Transfer Function	17-32
17.12	A/D Accuracy/Error	17-33
17.13	Operation During Sleep and Idle Modes	17-33
17.14	Effects of a Reset	17-34
17.15	Register Maps	17-35
17.16	Electrical Specifications	17-36
17.17	Design Tips	17-37
17.18	Related Application Notes	17-38
17.19	Revision History	17-39

17.1 INTRODUCTION

The PIC24F 10-bit A/D converter has the following key features:

- Successive Approximation Register (SAR) conversion
- Conversion speeds of up to 500 ksps
- Up to 16 analog input channels
- External voltage reference input pins
- Unipolar differential Sample-and-Hold (S/H) amplifier
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 16-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

A block diagram of the 10-bit A/D is shown in Figure 17-1. The converter can have up to 16 analog input channels, designated AN0 through AN15, on as many pins. The actual number of analog input pins and external voltage reference input pins will depend on the specific PIC24F device. For device-specific information, refer to the appropriate PIC24F data sheet.

There are also two analog input pins, VREF+ and VREF-, for external voltage reference connections. These voltage reference inputs may be shared with other analog input pins. The A/D reference voltages may be selected under software control from either the device supply voltage (AVDD/AVSS) or the voltage level on the VREF+/VREF- pins.

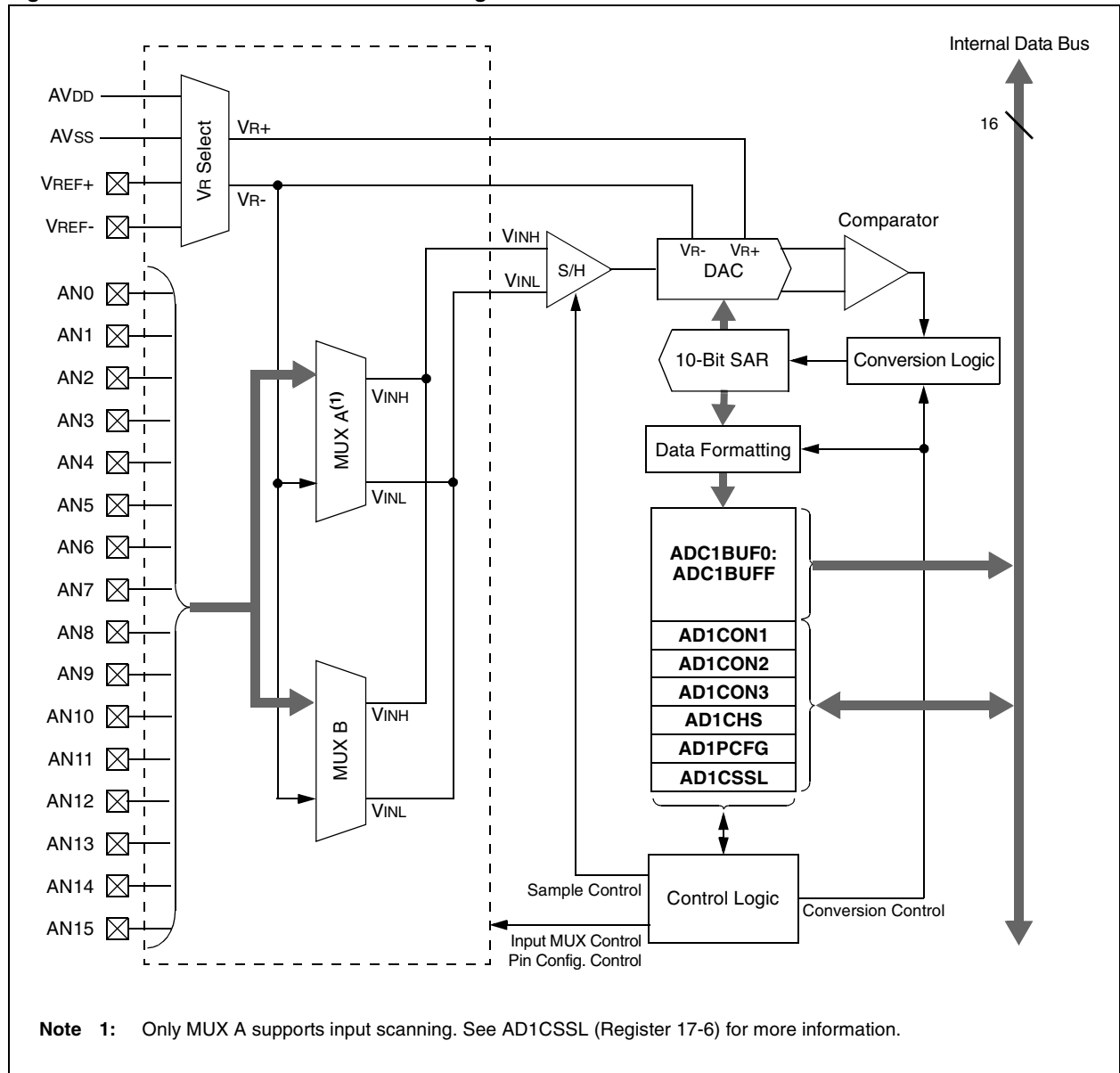
The analog inputs are connected via two independent multiplexers (MUX A and MUX B) to the S/H amplifier, also designated as the S/H channel or CH0. This gives the converter the ability to switch between two sets of analog inputs during conversions. Unipolar differential conversions are possible using certain input pins. An optional Analog Input Scan mode allows the converter to sequentially scan a selected range of channels automatically.

The 10-bit A/D includes a number of methods for controlling the sample and conversion process. Sample and conversion trigger sources can be taken from a variety of hardware sources, or can be controlled manually in software. The Auto-Sample mode and auto-conversion trigger can be used together to provide endless automatic conversions without software intervention.

A controller level interrupt may be generated at the end of each sample/convert sequence, or after multiple sequences. The number of sequences per interrupt event can vary between one and sixteen.

The converter stores its results in an internal 16-word data buffer, which is mapped into the device data space. Each of the 10-bit results can be stored in one of four 16-bit output formats.

Figure 17-1: 10-Bit A/D Converter Block Diagram



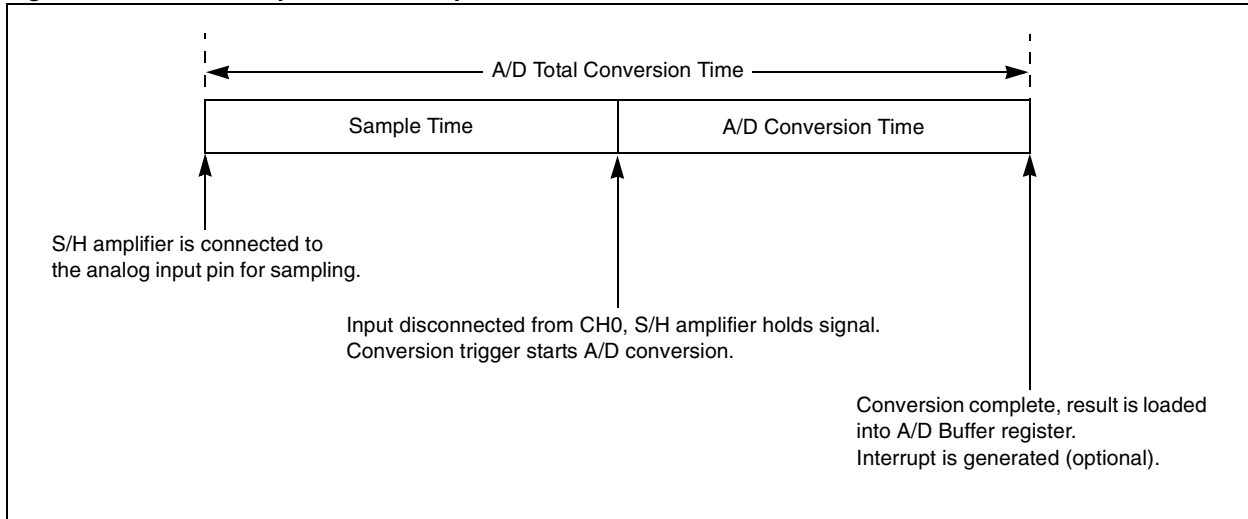
17.2 A/D TERMINOLOGY AND CONVERSION SEQUENCE

Sample time is the time that the A/D module's S/H amplifier is connected to the analog input pin. The sample time may be started and ended manually or automatically by the A/D converter hardware. There is a minimum sample time to ensure that the S/H amplifier will give the desired accuracy for the A/D conversion.

Conversion time is the time required for the A/D converter to convert the voltage held by the S/H amplifier. The conversion trigger ends the sampling time and begins an A/D conversion or a sample/convert sequence. The conversion trigger sources can be taken from a variety of hardware sources, or can be controlled manually in software. The A/D converter requires one A/D clock cycle (TAD) to convert each bit of the result, plus two additional clock cycles, or a total of 12 TAD cycles for a 10-bit conversion. When the conversion time is complete, the result is loaded into one of 16 A/D result buffers. The S/H can be reconnected to the input pin and a CPU interrupt may be generated. The sum of the sample time and the A/D conversion time provides the total conversion time. Figure 17-2 shows the basic conversion sequence and the relationship between intervals.

The conversion trigger sources can be taken from a variety of hardware sources, or can be controlled manually in software. One of the conversion trigger options is an auto-conversion, which uses a counter and the A/D clock to set the time between auto-conversions. The Auto-Sample mode and auto-conversion trigger can be used together to provide endless automatic conversions without software intervention.

Figure 17-2: A/D Sample/Convert Sequence



17.3 REGISTERS

The 10-bit A/D converter module uses a total of 22 registers for its operation. All registers are mapped in the data memory space.

17.3.1 Control Registers

The module has six control and status registers:

- AD1CON1: A/D Control Register 1
- AD1CON2: A/D Control Register 2
- AD1CON3: A/D Control Register 3
- AD1CHS: A/D Input Channel Select Register
- AD1PCFG: A/D Port Configuration Register
- AD1CSSL: A/D Input Scan Select Register

The AD1CON1, AD1CON2 and AD1CON3 registers (Register 17-1, Register 17-2 and Register 17-3) control the overall operation of the A/D module. This includes enabling the module, configuring the conversion clock and voltage reference sources, selecting the sampling and conversion triggers, and manually controlling the sample/convert sequences.

The AD1CHS register (Register 17-4) selects the input channels to be connected to the S/H amplifier. It also allows the choice of input multiplexers and the selection of a reference source for differential sampling.

The AD1PCFG register (Register 17-5) configures I/O pins as analog inputs or digital I/Os.

The AD1CSSL register (Register 17-6) selects the channels to be included for sequential scanning.

17.3.2 A/D Result Buffers

The module incorporates a 16-word, dual port RAM, called ADC1BUF, to store the A/D results. Each of the locations are mapped into the data memory space and are separately addressable. The 16 buffer locations are referred to as ADC1BUF0 through ADC1BUFF. The A/D result buffers are read-only.

PIC24F Family Reference Manual

Register 17-1: AD1CON1: A/D Control Register 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HCS	R/C-0, HCS
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7						bit 0	

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HCS = Cleared/Set in Hardware	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 15 **ADON:** A/D Operating Mode bit
1 = A/D converter module is operating
0 = A/D converter is disabled
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **ADSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode
- bit 12-10 **Unimplemented:** Read as '0'
- bit 9-8 **FORM1:FORM0:** Data Output Format bits
11 = Signed fractional (sddd dddd dd00 0000)
10 = Fractional (dddd dddd dd00 0000)
01 = Signed integer (ssss sssd dddd dddd)
00 = Integer (0000 00dd dddd dddd)
- bit 7-5 **SSRC2:SSRC0:** Conversion Trigger Source Select bits
111 = Internal counter ends sampling and starts conversion (auto convert)
110 = Reserved
10x = Reserved
100 = Reserved
011 = Reserved
010 = Timer3 compare match ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing SAMP bit ends sampling and starts conversion
- bit 4-3 **Unimplemented:** Read as '0'
- bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after last conversion completes; SAMP bit is automatically set.
0 = Sampling begins when SAMP bit is set
- bit 1 **SAMP:** A/D Sample Enable bit
1 = At least one A/D sample/hold amplifier is sampling
0 = A/D sample/hold amplifiers are holding
When ASAM = 0, writing '1' to this bit starts sampling. When SSRC<2:0> = 000, writing '0' to this bit will end sampling and start conversion.
- bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion is done
0 = A/D conversion is not done or has not started
Clearing this bit will not affect any operation in progress. Cleared by software or start of a new conversion.

Register 17-2: AD1CON2: A/D Control Register 2

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—
bit 15							bit 8
R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS ⁽¹⁾	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-13 **VCFG2:VCFG0:** Voltage Reference Configuration bits

VCFG2:VCFG0	VR+	VR-
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **Reserved:** Maintain as '0'

bit 11 **Unimplemented:** Read as '0'

bit 10 **CSCNA:** Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit
 1 = Scan inputs
 0 = Do not scan inputs

bit 9-8 **Unimplemented:** Read as '0'

bit 7 **BUFS:** Buffer Fill Status bit⁽¹⁾
 1 = A/D is currently filling ADC1BUF8-ADC1BUFF, user should access data in ADC1BUF0-ADC1BUF7
 0 = A/D is currently filling ADC1BUF0-ADC1BUF7, user should access data in ADC1BUF8-ADC1BUFF

bit 6 **Unimplemented:** Read as '0'

bit 5-2 **SMPI3:SMPI0:** Sample/Convert Sequences Per Interrupt Selection bits
 1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence
 1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence

 0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence
 0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **BUFM:** Buffer Mode Select bit
 1 = Buffer configured as two 8-word buffers (ADC1BUF0 to ADC1BUF7 and ADC1BUF8 to ADC1BUFF)
 0 = Buffer configured as one 16-word buffer (ADC1BUF0 to ADC1BUFF)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
 1 = Uses MUX A input multiplexer settings for first sample, then alternates between MUX B and MUX A input multiplexer settings for all subsequent samples
 0 = Always uses MUX A input multiplexer settings

Note 1: Only valid when ADC1BUF is functioning as two buffers (BUFM = 1).

PIC24F Family Reference Manual

Register 17-3: AD1CON3: A/D Control Register 3

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

bit 15 **ADRC:** A/D Conversion Clock Source bit

1 = A/D internal RC clock
0 = Clock derived from system clock

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **SAMC4:SAMC0:** Auto-Sample Time bits

11111 = 31 TAD
.....
00001 = 1 TAD
00000 = 0 TAD (not recommended)

bit 7-0 **ADCS7:ADCS0:** A/D Conversion Clock Select bits

11111111 = 128 TCY
11111110 = 127 TCY
.....
00000001 = TCY
00000000 = TCY/2

Section 17. 10-Bit A/D Converter

Register 17-4: AD1CHS: A/D Input Channel Select Register

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15				bit 8			

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	—	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7				bit 0			

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

- bit 15 **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit
 1 = Channel 0 negative input is AN1
 0 = Channel 0 negative input is VR-
- bit 14-12 **Unimplemented:** Read as '0'
- bit 11-8 **CH0SB3:CH0SB0:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bits
 1111 = Channel 0 positive input is AN15
 1110 = Channel 0 positive input is AN14
 1101 = Channel 0 positive input is AN13

 0001 = Channel 0 positive input is AN1
 0000 = Channel 0 positive input is AN0
- bit 7 **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit
 1 = Channel 0 negative input is AN1
 0 = Channel 0 negative input is VR-
- bit 6-4 **Unimplemented:** Read as '0'
- bit 3-0 **CH0SA3:CH0SA0:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bits
 1111 = Channel 0 positive input is AN15
 1110 = Channel 0 positive input is AN14
 1101 = Channel 0 positive input is AN13

 0001 = Channel 0 positive input is AN1
 0000 = Channel 0 positive input is AN0

PIC24F Family Reference Manual

Register 17-5: AD1PCFG: A/D Port Configuration Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-0 **PCFG15:PCFG0:** Analog Input Pin Configuration Control bits
 1 = Pin for corresponding analog channel is in Digital mode; port read input enabled, A/D input multiplexer input connected to AVss
 0 = Pin for corresponding analog channel is in Analog mode; port read input disabled, A/D module samples pin voltage

Register 17-6: AD1CSSL: A/D Input Scan Select Register for MUX A⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-0 **CSSL15:CSSL0:** A/D Input Channel Scan Selection bits
 1 = Corresponding analog channel, ANxx, is selected for sequential scanning on MUX A
 0 = Corresponding analog channel is ignored in sequential scanning

Note 1: Only MUX A scanning is supported.

17.4 A/D MODULE CONFIGURATION

The following steps should be followed for performing an A/D conversion.

1. Configure the A/D module:
 - Select voltage reference source to match expected range on analog inputs
 - Select the analog conversion clock to match desired data rate with processor clock
 - Determine how sampling will occur
 - Determine how inputs will be allocated to the S/H channel
 - Select the desired sample/conversion sequence
 - Select how conversion results are presented in the buffer
 - Select interrupt rate
 - Turn on A/D module
2. Configure A/D interrupt (if required):
 - Clear AD1IF bit
 - Select A/D interrupt priority

The options for each configuration step are described in the subsequent sections.

Note: The SSRC, ASAM, BUFS SMPI, BUFM and ALTS bits, as well as the AD1CON3 and AD1CSSL registers, should not be written to while ADON = 1. Indeterminate conversion data may result.

17.4.1 Selecting the Voltage Reference Source

The voltage references for A/D conversions are selected using the VCFG2:VCFG0 control bits (AD1CON2<15:13>). The upper voltage reference (VR+) and the lower voltage reference (VR-) may be the internal AVDD and AVSS voltage rails, or the VREF+ and VREF- input pins.

The external voltage reference pins may be shared with the AN0 and AN1 inputs on low pin count devices. The A/D converter can still perform conversions on these pins when they are shared with the VREF+ and VREF- input pins.

The voltages applied to the external reference pins must meet certain specifications. Refer to **Section 17.16 “Electrical Specifications”** for further details.

17.4.2 Selecting the A/D Conversion Clock

The A/D converter has a maximum rate at which conversions may be completed. An analog module clock, TAD, controls the conversion timing. The A/D conversion requires 12 clock periods (12 TAD). The A/D clock is derived from the device instruction clock.

The period of the A/D conversion clock is software selected using an 8-bit counter. There are 64 possible options for TAD, specified by the ADCS7:ADCS0 bits (AD1CON3<7:0>). Equation 17-1 gives the TAD value as a function of the ADCS control bits and the device instruction cycle clock period, T_{CY}. For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 75 ns.

Equation 17-1: A/D Conversion Clock Period

$$T_{AD} = \frac{T_{CY} (ADCS + 1)}{2}$$

$$ADCS = \frac{2 T_{AD}}{T_{CY}} - 1$$

Note: Based on T_{CY} = FOSC/2, Doze mode and PLL are disabled.

The A/D converter also has its own dedicated RC clock source that can be used to perform conversions. The A/D RC clock source should be used when conversions are performed while the device is in Sleep mode. The RC oscillator is selected by setting the ADRC bit (AD1CON3<15>). When the ADRC bit is set, the ADCS bits have no effect on A/D operation.

17.4.3 Configuring Analog Port Pins

The AD1PCFG register specifies the input condition of device pins used as analog inputs. A pin is configured as an analog input when the corresponding PCFGn bit (AD1PCFG<n>) for its analog channel is cleared. The AD1PCFG register is cleared on device Resets, causing the A/D input pins to be configured for analog inputs by default. When configured for analog inputs, the associated port I/O digital input buffer is disabled so it does not consume current.

Both the AD1PCFG register and the corresponding TRIS register bits control the operation of the A/D port pins. The port pins that will function as analog inputs must also have their corresponding TRIS bits set, specifying the pins as inputs. After a device Reset, all TRIS bits are set.

If the I/O pin associated with an A/D channel is configured as a digital output (TRIS bit is cleared), while the pin is configured for Analog mode (AD1PCFG<n> = 0), the port digital output level (VOH or VOL) will be converted.

A pin is configured as digital I/O when the corresponding PCFGn bit is set. In this configuration, the input to the analog multiplexer is connected to AVSS.

- | |
|---|
| <p>Note 1: When reading a PORT register, any pin configured as an analog input reads as '0'.</p> <p>2: Analog levels on any pin that is defined as a digital input (including the AN15:AN0 pins) may cause the input buffer to consume current that is out of the device's specification.</p> |
|---|

17.4.4 CH0 Input Selection

The A/D converter incorporates two independent sets of input multiplexers that allow users to choose which analog channels are to be sampled. Collectively, they are known as Multiplexer A (MUX A) and Multiplexer B (MUX B). The inputs specified by CH0SA3:CH0SA0 and CH0NA are collectively called the MUX A inputs. The inputs specified by CH0SB3:CH0SB0 and CH0NB are collectively called the MUX B inputs.

Functionally, MUX A and MUX B are very similar to each other. Both multiplexers allow any of the analog input channels to be selected for individual sampling, and allow selection between several options for a negative reference source in differential sampling. In addition, MUX A can be configured for sequential analog channel scanning, while MUX B allows for a wider selection of reference sources. This is discussed in more detail in **Section 17.4.4.1 "Configuring MUX A and MUX B Inputs"** and **Section 17.4.4.3 "Scanning Through Several Inputs"**.

- | |
|---|
| <p>Note: Different PIC24F devices will have different numbers of analog inputs. Verify the analog input availability against the particular device's data sheet.</p> |
|---|

17.4.4.1 CONFIGURING MUX A AND MUX B INPUTS

The user may select any one of the 16 analog inputs to connect to the positive input of CH0. For MUX A, the CH0SA3:CH0SA0 bits (AD1CHS<3:0>) normally select the analog channel for the positive input. For MUX B, the positive channel is selected by the CH0SB3:CH0SB0 bits (AD1CHS<11:8>).

For the negative (inverting) input of CH0, the user has two options, selected by the CH0NA and CH0NB bits (AD1CHS<7,15>, respectively). Setting either bit selects AN1 as the multiplexer's negative input; clearing the bit selects the current VR- source.

17.4.4.2 ALTERNATING MUX A AND MUX B INPUT SELECTIONS

By default, the A/D converter only samples and converts the inputs selected by MUX A. The ALTS bit (AD1CON2<0>) enables the module to alternate between two sets of inputs selected by MUX A and MUX B during successive samples.

If the ALTS bit is '0', only the inputs specified by the CH0SA and CH0NA bits are selected for sampling. When the ALTS bit is '1', the module will alternate between the MUX A inputs on one sample and the MUX B inputs on the subsequent sample.

If the ALTS bit is '1' on the first sample/convert sequence for Channel 0, the inputs specified by CH0SA<3:0> and CH0NA are selected for sampling. On the next sample/convert sequence, the inputs specified by CH0SB<3:0> and CH0NB are selected for sampling. This pattern repeats for subsequent sample conversion sequences.

17.4.4.3 SCANNING THROUGH SEVERAL INPUTS

When using MUX A to select analog inputs, the A/D module has the ability to scan multiple analog channels. The CSCNA bit (AD1CON2<10>) enables the CH0 channel inputs to be scanned across a selected number of analog inputs. When CSCNA is set, the CH0SA bits are ignored and the channels specified by the AD1CSSL register are sequentially sampled.

Each bit in the AD1CSSL register corresponds to one of the analog channels. If a bit in the AD1CSSL register is set, the corresponding analog channel is included in the scan sequence. Inputs are always scanned from lower to higher numbered inputs, starting at the first selected channel after each interrupt occurs.

Note: If the number of scanned inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs will not be sampled.

The AD1CSSL bits only specify the input of the positive input of the channel. The CH0NA bit still selects the negative input of the channel during scanning.

Scanning is only available on the MUX A input selection. The MUX B input selection, as specified by the CH0SB<3:0> bits, will still select the alternating input. When alternated sampling between MUX A and MUX B is selected (ALTS = 1), the input will alternate between a set of scanning inputs specified by the AD1CSSL register and a fixed input specified by the CH0SB bits.

17.4.5 Enabling the Module

When the ADON bit (AD1CON1<15>) is set, the module is fully powered and functional. When ADON is '0', the module is disabled. The digital and analog portions of the circuit are turned off for maximum current savings.

When enabling the module by setting the ADON bit, the user must wait for the analog stages to stabilize. For the stabilization time, refer to **Section 17.16 “Electrical Specifications”**.

17.5 INITIALIZATION

Example 17-1 shows a simple initialization code example for the A/D module. In this particular configuration, all 16 analog input pins are set up as analog inputs. Operation in Idle mode is disabled, output data is in unsigned fractional format and AVDD and AVSS are used for VR+ and VR-. The start of sampling, as well as the start of conversion (conversion trigger), are performed manually in software. Scanning of inputs is disabled and an interrupt occurs after every sample/convert sequence (1 conversion result), with only one channel (AN0) being converted. The A/D conversion clock is $T_{CY}/2$.

This example shows one method of controlling a sample/convert sequence by manually setting and clearing the SAMP bit (AD1CON1<1>). This method, among others, is more fully discussed in **Section 17.6 “Controlling the Sampling Process”** and **Section 17.7 “Controlling the Conversion Process”**.

Example 17-1: A/D Initialization Code Example

```
AD1PCFG      = 0;           // Configure A/D port
                        // all input pins are analog
AD1CON1      = 0x2208;      // Configure sample clock source
                        // and conversion trigger mode.
                        // Unsigned Fraction format (FORM<1:0>=10),
                        // Manual conversion trigger (SSRC<3:0>=000),
                        // Manual start of sampling (ASAM=0),
                        // No operation in Idle mode (ADSIDL=1).
AD1CON2      = 0;           // Configure A/D voltage reference
                        // and buffer fill modes.
                        // Vr+ and Vr- from AVdd and AVss (VCFG<2:0>=000),
                        // Inputs are not scanned,
                        // Interrupt after every sample
AD1CON3      = 0;           // Configure A/D conversion clock as Tcy/2
AD1CHS       = 0;           // Configure input channels,
                        // CH0+ input is AN0,
                        // CH0- input is Vr- (AVss).
AD1CSSL      = 0;           // No inputs are scanned.
IFS0bits.AD1IF = 0;        // Clear A/D conversion interrupt.

// Configure A/D interrupt priority bits (AD1IP<2:0>) here, if
// required. Default priority level is 4.

IEC0bits.AD1IE = 1;        // Enable A/D conversion interrupt
AD1CON1bits.ADON = 1;      // Turn on A/D
AD1CON1bits.SAMP = 1;      // Start sampling the input
Delay();                  // Ensure the correct sampling time has elapsed
                        // before starting conversion.
AD1CON1bits.SAMP = 0;      // End A/D sampling and start conversion

// Example code for A/D ISR:
void __attribute__((__interrupt__)) _ADC1Interrupt(void)
{
    IFS0bits.AD1IF = 0;
}
```

17.6 CONTROLLING THE SAMPLING PROCESS

17.6.1 Manual Sampling

Setting the SAMP bit (AD1CON1<1>) while the ASAM bit (AD1CON1<2>) is clear causes the A/D to begin sampling. One of several options can be used to end sampling and complete the conversions. Sampling will not resume until the SAMP bit is once again set. For an example, see Figure 17-3.

17.6.2 Automatic Sampling

Setting the ASAM bit causes the A/D to automatically begin sampling after a conversion has been completed. One of several options can be used to end sampling and complete the conversions. Sampling on a channel resumes after the conversion of that channel completes. For an example, see Figure 17-4.

17.6.3 Monitoring Sample Status

The SAMP bit indicates the sampling state of the A/D. Generally, when the SAMP bit clears, indicating the end of sampling, the DONE bit is automatically cleared to indicate the start of conversion. If SAMP is '0' while DONE is '1', the A/D is in an inactive state.

17.6.4 Aborting a Sample

While in Manual Sampling mode, clearing the SAMP bit will terminate sampling. If SSRC2:SSRC0 = 000, it may also start a conversion automatically.

Clearing the ASAM bit while in Automatic Sampling mode will not terminate an ongoing sample/convert sequence, however, sampling will not automatically resume after a subsequent conversion.

17.7 CONTROLLING THE CONVERSION PROCESS

The conversion trigger source will terminate sampling and start a selected sequence of conversions. The SSRC2:SSRC0 bits (AD1CON1<7:5>) select the source of the conversion trigger.

Note 1: The available conversion trigger sources may vary depending on the PIC24F device variant. Please refer to the specific device data sheet for the available conversion trigger sources.

2: The SSRC selection bits should not be changed when the A/D module is enabled. If the user wishes to change the conversion trigger source, the A/D module should be disabled first by clearing the ADON bit (AD1CON1<15>).

17.7.1 Manual Control

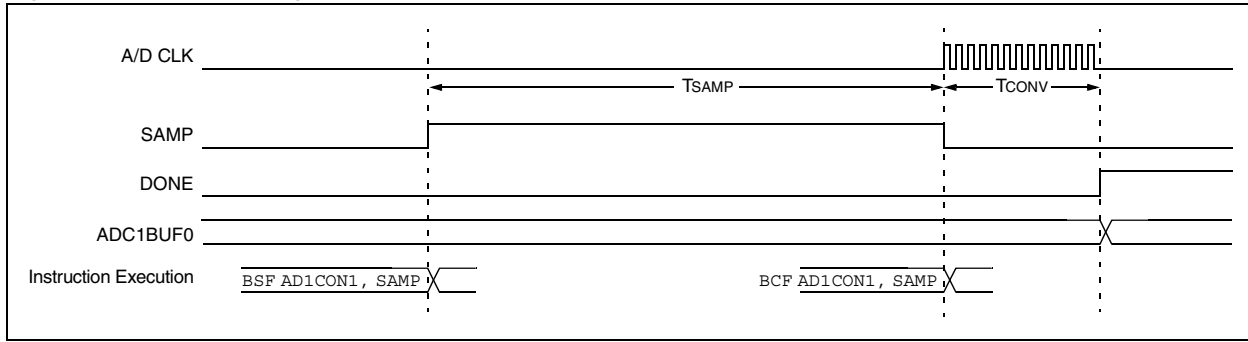
When SSRC2:SSRC0 = 000, the conversion trigger is under software control. Clearing the SAMP bit (AD1CON1<1>) starts the conversion sequence.

Figure 17-3 is an example where setting the SAMP bit initiates sampling, and clearing the SAMP bit terminates sampling and starts conversion. The user software must time the setting and clearing of the SAMP bit to ensure adequate sampling time of the input signal.

Figure 17-4 is an example where setting the ASAM bit initiates automatic sampling, and clearing the SAMP bit terminates sampling and starts conversion. After the conversion completes, the module will automatically return to a sampling state. The SAMP bit is automatically set at the start of the sample interval. The user software must time the clearing of the SAMP bit to ensure adequate sampling time of the input signal, understanding that the time between clearing of the SAMP bit includes the conversion time, as well as the sampling time.

PIC24F Family Reference Manual

Figure 17-3: Converting One Channel, Manual Sample Start, Manual Conversion Start



Example 17-2: Converting One Channel, Manual Sample Start, Manual Conversion Start Code Example

```

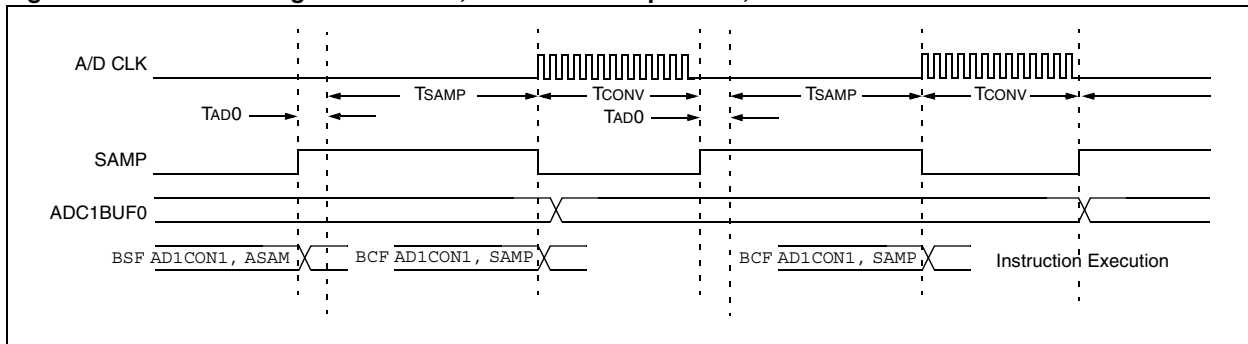
int          ADCValue;

AD1PCFG     = 0xFFFB;           // AN2 as analog, all other pins are digital
AD1CON1     = 0x0000;           // SAMP bit = 0 ends sampling
                                           // and starts converting
AD1CHS      = 0x0002;           // Connect AN2 as CH0 input
                                           // in this example AN2 is the input

AD1CSSL     = 0;
AD1CON3     = 0x0002;           // Manual Sample, Tad = 2 Tcy
AD1CON2     = 0;
AD1CON1bits.ADON = 1;           // turn ADC ON

while (1) // repeat continuously
{
    AD1CON1bits.SAMP = 1;       // start sampling...
    Delay();                    // Ensure the correct sampling time has elapsed
                                // before starting conversion.
    AD1CON1bits.SAMP = 0;       // start Converting
    while (!AD1CON1bits.DONE);  // conversion done?
    ADCValue = ADC1BUF0;        // yes then get ADC value
}
    
```

Figure 17-4: Converting One Channel, Automatic Sample Start, Manual Conversion Start



17.7.2 Clocked Conversion Trigger

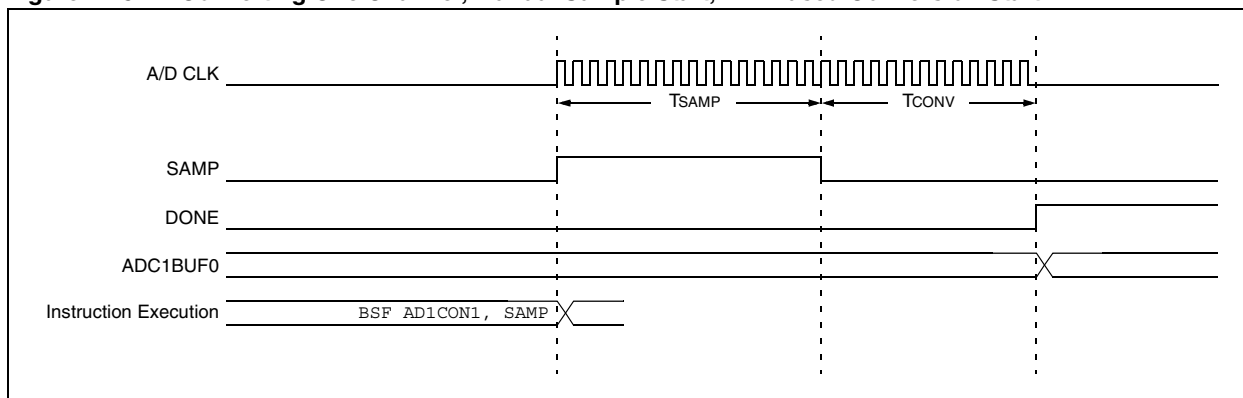
When SSRC2:SSRC0 = 111, the conversion trigger is under A/D clock control. The SAMC bits (AD1CON3<12:8>) select the number of T_{AD} clock cycles between the start of sampling and the start of conversion. After the start of sampling, the module will count a number of T_{AD} clocks specified by the SAMC bits. The SAMC bits must always be programmed for at least 1 clock cycle to ensure sampling requirements are met.

Equation 17-2: Clocked Conversion Trigger Time

$$T_{SMP} = SAMC<4:0> * T_{AD}$$

Figure 17-5 shows how to use the clocked conversion trigger with the sampling started by the user software.

Figure 17-5: Converting One Channel, Manual Sample Start, T_{AD} Based Conversion Start



Example 17-3: Converting One Channel, Manual Sample Start, T_{AD} Based Conversion Start Code Example

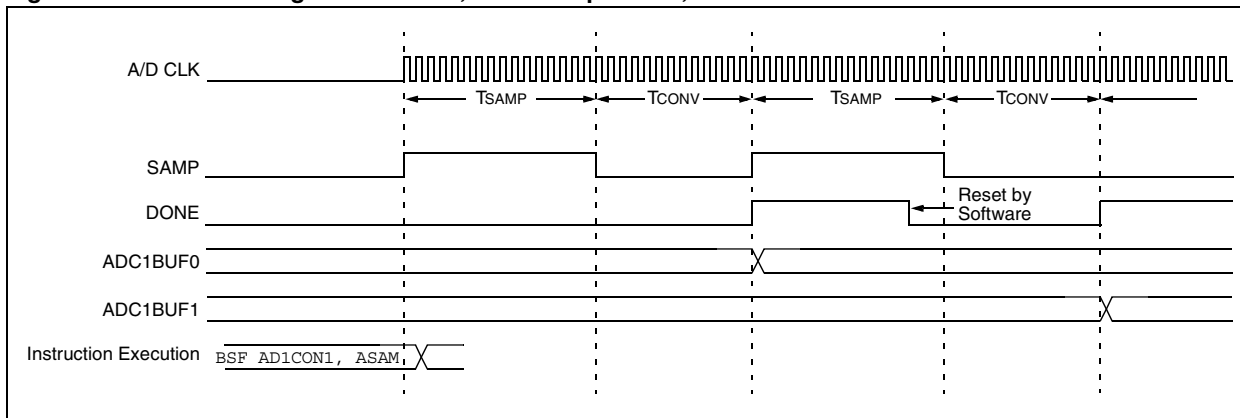
```
int          ADCValue;

AD1PCFG     = 0xEFFF;           // all PORTB = Digital; RB12 = analog
AD1CON1     = 0x00E0;           // SSRC<3:0> = 111 implies internal
                                           // counter ends sampling and starts
                                           // converting.
AD1CHS     = 0x000C;           // Connect AN12 as CH0 input.
                                           // in this example AN12 is the input
AD1CSSL     = 0;
AD1CON3     = 0x1F02;           // Sample time = 31Tad,
                                           // Tad = 2 Tcy
AD1CON2     = 0;
AD1CON1bits.ADON = 1;           // turn ADC ON
while (1) // repeat continuously
{
    AD1CON1bits.SAMP = 1;       // start sampling then...
                                // after 31Tad go to conversion
while (!AD1CON1bits.DONE);     // conversion done?
    ADCValue = ADC1BUF0;       // yes then get ADC value
}                               // repeat
```

17.7.2.1 FREE-RUNNING SAMPLE CONVERSION SEQUENCE

Using the Auto-Convert Conversion Trigger mode (SSRC2:SSRC0 = 111), in combination with the Auto-Sample Start mode (ASAM = 1), allows the A/D module to schedule sample/conversion sequences with no intervention by the user or other device resources. This “Clocked” mode, shown in Figure 17-6, allows continuous data collection after module initialization.

Figure 17-6: Converting One Channel, Auto-Sample Start, TAD Based Conversion Start



Example 17-4: Converting One Channel, Auto-Sample Start, TAD Based Conversion Start Code

```

int      ADCValue, count;
int      *ADC16Ptr;

AD1PCFG  = 0xFFFB;           // AN2 as analog,
                             // all other pins are digital
AD1CON1  = 0x00E0;           // SSRC bit = 111 implies internal
                             // counter ends sampling
                             // and starts converting.
AD1CHS   = 0x0002;           // Connect RB2/AN2 as CH0 input..
                             // in this example RB2/AN2 is
                             // the input

AD1CSSL  = 0;
AD1CON3  = 0x0F00;           // Sample time = 15Tad,
                             // Tad = Tcy/2
AD1CON2  = 0x0004;           // Set AD1IF after every 2 samples
AD1CON1bits.ADON = 1;        // turn ADC ON
while (1)                     // repeat continuously
{
    ADCValue = 0;             // clear variable
    ADC16Ptr = &ADC1BUF0;    // initialize ADC1BUF pointer
    IFS0bits.AD1IF = 0;      // clear ADC interrupt flag
    AD1CON1bits.ASAM = 1;    // auto start sampling
                             // for 31Tad then go to conversion
    while (!IFS0bits.AD1IF); // conversion done?
    AD1CON1bits.ASAM = 0;    // yes then stop sample/convert
    for (count = 0; count < 2; count++) // average the 2 ADC value
        ADCValue = ADCValue + *ADC16Ptr++;
    ADCValue = ADCValue >> 1;
}                               // repeat}
    
```

17.7.2.2 SAMPLE TIME CONSIDERATIONS USING CLOCKED CONVERSION TRIGGER AND AUTOMATIC SAMPLING

The user must ensure the sampling time exceeds the sampling requirements as outlined in **Section 17.10 “A/D Sampling Requirements”**. Assuming that the module is set for automatic sampling and using a clocked conversion trigger, the sampling interval is specified by the SAMC bits.

17.7.3 Event Trigger Conversion Start

It is often desirable to synchronize the end of sampling and the start of conversion with some other time event. The A/D module may use one of three sources as a conversion trigger event.

17.7.3.1 EXTERNAL INTO PIN TRIGGER

When $SSRC2:SSRC0 = 001$, the A/D conversion is triggered by an active transition on the INTO pin. The pin may be programmed for either a rising edge input or a falling edge input.

17.7.3.2 GENERAL PURPOSE TIMER COMPARE TRIGGER

The A/D is configured in this Trigger mode by setting $SSRC<2:0> = 010$. When a match occurs between the 32-bit timer, TMR3/TMR2, and the 32-bit combined period register, PR3/PR2, a special ADC trigger event signal is generated by Timer3. This feature does not exist for the TMR5/TMR4 timer pair. Refer to **Section 14. "Timers"** for more details.

17.7.3.3 SYNCHRONIZING A/D OPERATIONS TO INTERNAL OR EXTERNAL EVENTS

The modes where an external event trigger pulse ends sampling and starts conversion ($SSRC2:SSRC0 = 001, 010$ or 011) may be used in combination with auto-sampling ($ASAM = 1$) to cause the A/D to synchronize the sample conversion events to the trigger pulse source. For example, in Figure 17-8 where $SSRC2:SSRC0 = 010$ and $ASAM = 1$, the A/D will always end sampling and start conversions synchronously with the timer compare trigger event. The A/D will have a sample conversion rate that corresponds to the timer comparison event rate.

Figure 17-7: Manual Sample Start, Conversion Trigger Based Conversion Start

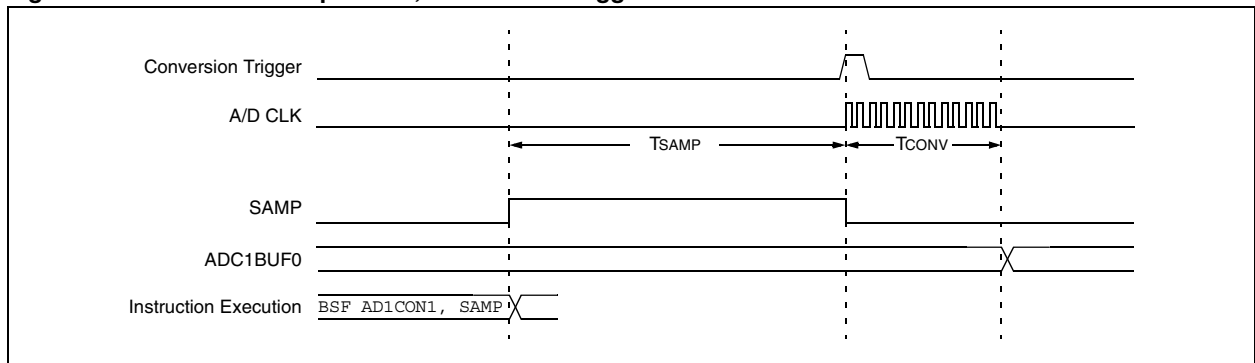
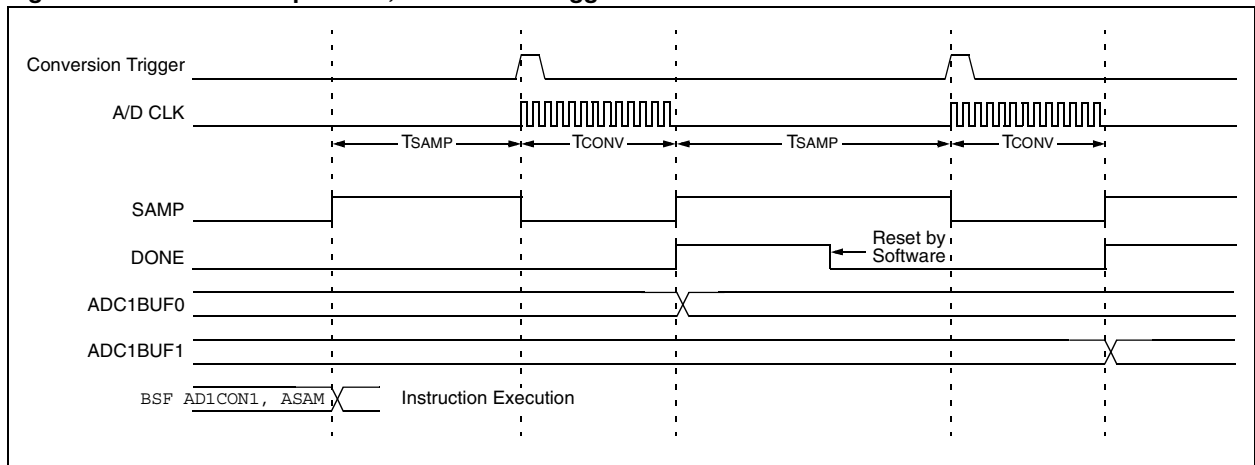


Figure 17-8: Auto-Sample Start, Conversion Trigger Based Conversion Start



Example 17-5: Converting One Channel, Auto-Sample Start, Conversion Trigger Based Conversion Start Code

```

int          ADCValue;

AD1PCFG     = 0xFFFF;           // AN2 analog, all other pins digital
AD1CON1     = 0x0040;           // SSRC bit = 010 implies GP TMR3
                                           // compare ends sampling and starts
                                           // converting.
AD1CHS      = 0x0002;           // Connect AN2 as CH0 input...
                                           // in this example AN2 is the input

AD1CSSL     = 0;
AD1CON3     = 0x0000;           // Sample time is TMR3, Tad = Tcy/2
AD1CON2     = 0x0004;           // Set AD1IF after 2 conversions
                                           // set TMR3 to time out every 125 ms

TMR3        = 0x0000;
PR3         = 0x3FFF;
T3CON       = 0x8010;

AD1CON1bits.ADON = 1;           // turn ADC ON
AD1CON1bits.ASAM = 1;           // start auto sampling every 125 ms
while (1)           // repeat continuously
{
while (!IFS0bits.AD1IF);       // conversion done?
ADCValue = ADC1BUF0;           // yes then get first ADC value
IFS0bits.AD1IF = 0;           // clear AD1IF
}
    
```

17.7.3.4 SAMPLE TIME CONSIDERATIONS FOR AUTOMATIC SAMPLING/CONVERSION SEQUENCES

Different sample/conversion sequences provide different available sampling times for the S/H channel to acquire the analog signal. The user must ensure the sampling time exceeds the sampling requirements, as outlined in **Section 17.10 “A/D Sampling Requirements”**.

Assuming that the module is set for automatic sampling and an external trigger pulse is used as the conversion trigger, the sampling interval is a portion of the trigger pulse interval. The sampling time is the trigger pulse period, less the time required to complete the conversion.

Equation 17-3: Calculating Available Sampling Time for Sequential Sampling

$$T_{SMP} = \text{Trigger Pulse Interval (TSEQ)} - \text{Conversion Time (TCONV)} = T_{SEQ} - T_{CONV}$$

17.7.4 Monitoring Sample/Conversion Status

The DONE bit (AD1CON1<0>) indicates the conversion state of the A/D. Generally, when the SAMP bit clears, indicating the end of sampling, the DONE bit is automatically cleared to indicate the start of conversion. If SAMP is ‘0’ while DONE is ‘1’, the A/D is in an inactive state.

In some operational modes, the SAMP bit may also invoke and terminate sampling. In these modes, the DONE bit cannot be used to terminate conversions in progress.

17.7.5 Generating A/D Interrupts

The SMP13:SMP10 bits (AD1CON2<5:2>) control the generation of the AD1IF interrupt flag. The A/D interrupt flag is set after the number of sample/conversion sequences is specified by the SMP1 bits after the start of sampling, and continues to recur after that number of samples. The value specified by the SMP1 bits also corresponds to the number of data samples in the buffer, up to the maximum of 16. To enable the interrupt, it is necessary to set the A/D Interrupt Enable bit, AD11E.

17.7.6 Aborting a Conversion

Clearing the ADON bit during a conversion will abort the current conversion. The A/D results buffer will not be updated with the partially completed A/D conversion sample; that is, the corresponding ADC1BUF buffer location will continue to contain the value of the last completed conversion (or the last value written to the buffer).

17.8 A/D RESULTS BUFFER

As conversions are completed, the module writes the results of the conversions into the A/D result buffer. This buffer is a RAM array of sixteen words, accessed through the SFR space.

User software may attempt to read each A/D conversion result as it is generated, however, this might consume too much CPU time. Generally, to simplify the code, the module will fill the buffer with results and then generate an interrupt when the buffer is filled.

17.8.1 Number of Conversions per Interrupt

The SMPI3:SMPI0 bits will select how many A/D conversions will take place before the CPU is interrupted. This can vary from one to 16 samples per interrupt. The A/D converter module always starts writing its conversion results at the beginning of the buffer, after each interrupt. For example, if SMPI3:SMPI0 = 0000, the conversion results will always be written to the ADC1BUF0. In this example, no other buffer locations would be used.

17.8.2 Buffer Fill Mode

When the BUFM bit (AD1CON2<1>) is '1', the 16-word results buffer is split into two 8-word groups: a lower group (ADC1BUF0 through ADC1BUF7) and an upper group (ADC1BUF8 through ADC1BUFF). The 8-word buffers will alternately receive the conversion results after each interrupt event. The initial 8-word buffer used after BUFM is set is the lower group. When BUFM is '0', the complete 16-word buffer is used for all conversion sequences.

Note: When the BUFM bit (AD1CON2<1>) is set, the user should not program the SMPI bits to a value that specifies more than 8 conversions per interrupt

The decision to use the split buffer feature will depend upon how much time is available to move the buffer contents, after the interrupt, as determined by the application. If the application can quickly unload a full buffer within the time it takes to sample and convert one channel, the BUFM bit can be '0', and up to 16 conversions may be done per interrupt. The application will have one sample/convert time before the first buffer location is overwritten.

If the processor cannot unload the buffer within the sample and conversion time, the BUFM bit should be '1'. For example, if SMPI3:SMPI0 = 0111, then eight conversions will be loaded into the lower half of the buffer, following which, an interrupt may occur. The next eight conversions will be loaded into the upper half of the buffer. The processor will, therefore, have the entire time between interrupts to move the eight conversions out of the buffer.

17.8.3 Buffer Fill Status

When the conversion result buffer is split, using the BUFM control bit, the BUFS status bit (AD1CON2<7>) indicates the half of the buffer that the A/D converter is currently writing. If BUFS = 0, the A/D converter is filling the lower group, and the user software should read conversion values from the upper group. If BUFS = 1, the situation is reversed, and the user software should read conversion values from the lower group.

17.8.4 Buffer Data Formats

The results of each A/D conversion are 10 bits wide. To maintain data format compatibility, the result of each conversion is automatically converted to one of four selectable, 16-bit formats. The FORM1:FORM0 bits (AD1CON1<9:8>) select the format. Figure 17-9 shows the data output formats that can be selected.

PIC24F Family Reference Manual

Figure 17-9: A/D Output Data Formats

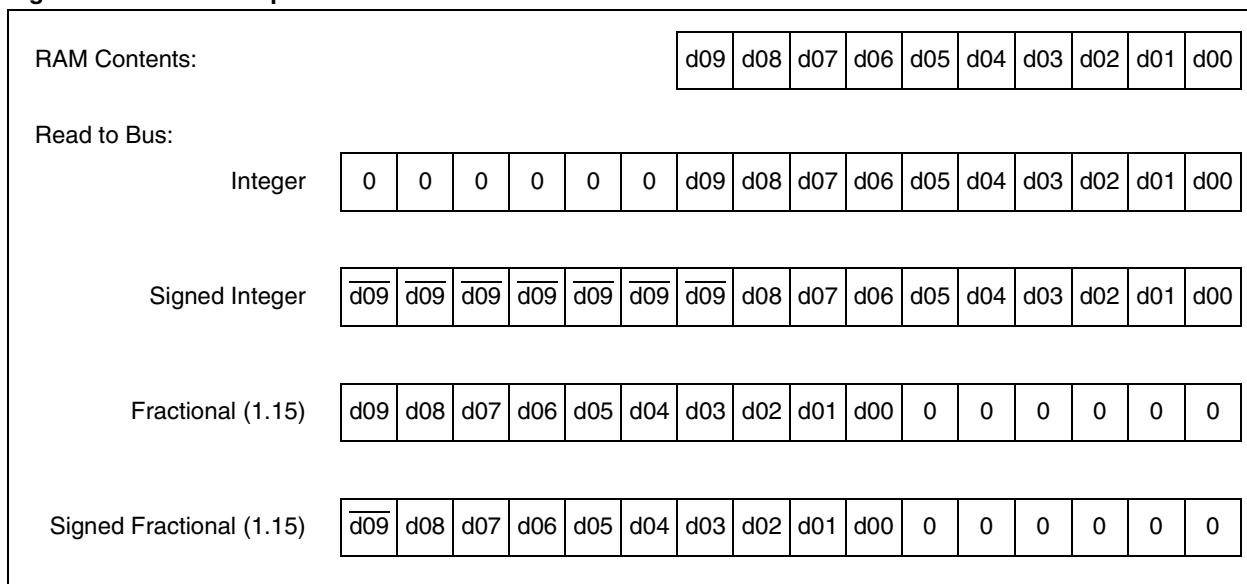


Table 17-1: Numerical Equivalents of Various Result Codes: Integer Formats

V _{IN} /V _{REF}	10-Bit Output Code	16-Bit Integer Format/ Equivalent Decimal Value	16-Bit Signed Integer Format/ Equivalent Decimal Value
1023/1024	11 1111 1111	0000 0011 1111 1111	1023
1022/1024	11 1111 1110	0000 0011 1111 1110	1022
...			
513/1024	10 0000 0001	0000 0010 0000 0001	513
512/1024	10 0000 0000	0000 0010 0000 0000	512
511/1024	01 1111 1111	0000 0001 1111 1111	511
...			
1/1024	00 0000 0001	0000 0000 0000 0001	1
0/1024	00 0000 0000	0000 0000 0000 0000	0

Table 17-2: Numerical Equivalents of Various Result Codes: Fractional Formats

V _{IN} /V _{REF}	10-Bit Output Code	16-Bit Fractional Format/ Equivalent Decimal Value	16-Bit Signed Fractional Format/ Equivalent Decimal Value
1023/1024	11 1111 1111	1111 1111 1100 0000	0.999
1022/1024	11 1111 1110	1111 1111 1000 0000	0.998
...			
513/1024	10 0000 0001	1000 0000 0100 0000	0.501
512/1024	10 0000 0000	1000 0000 0000 0000	0.500
511/1024	01 1111 1111	0111 1111 1100 0000	0.499
...			
1/1024	00 0000 0001	0000 0000 0100 0000	0.001
0/1024	00 0000 0000	0000 0000 0000 0000	0.000

17.9 CONVERSION SEQUENCE EXAMPLES

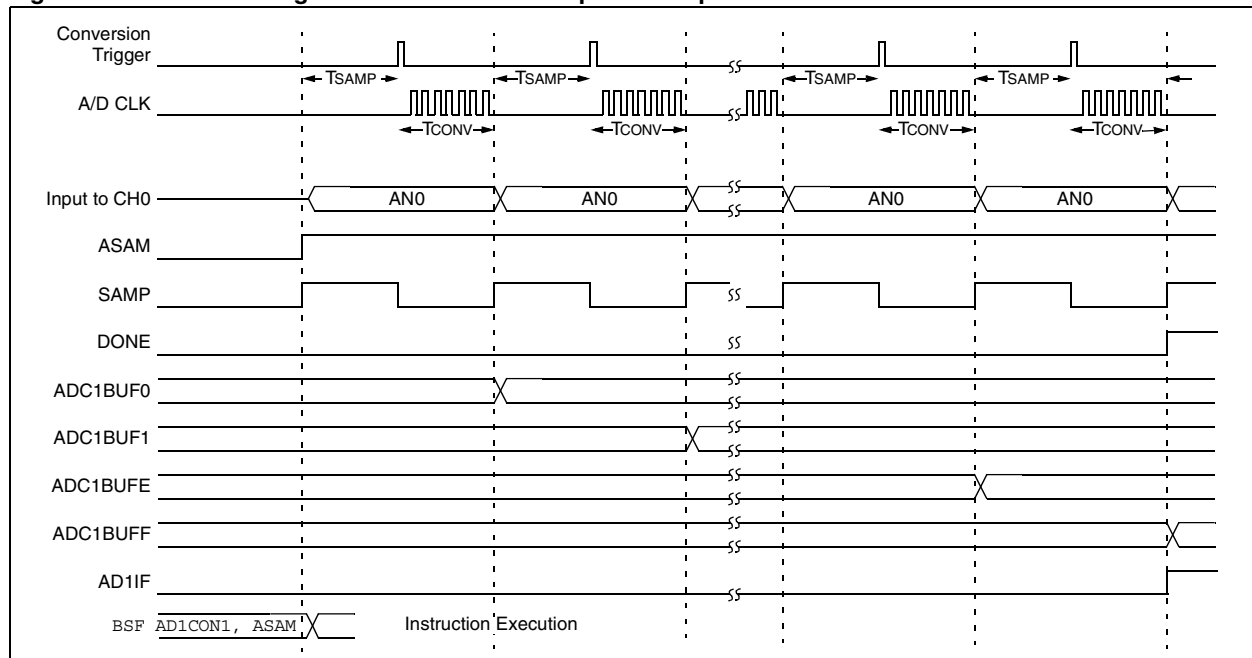
The following configuration examples show the A/D operation in different sampling and buffering configurations. In each example, setting the ASAM bit starts automatic sampling. A conversion trigger ends sampling and starts conversion.

17.9.1 Sampling and Converting a Single Channel Multiple Times

Figure 17-10 and Example 17-6 illustrate a basic configuration of the A/D. In this case, one A/D input, AN0, will be sampled and converted. The results are stored in the ADC1BUF buffer. This process repeats 16 times until the buffer is full and then the module generates an interrupt. The entire process will then repeat.

With ALTS clear, only the MUX A inputs are active. The CH0SA bits and CH0NA bit are specified (AN0-VR-) as the inputs to the sample/hold channel. All other input selection bits are not used.

Figure 17-10: Converting One Channel 16 Times per Interrupt



Example 17-6: Sampling and Converting a Single Channel Multiple Times

```

int      ADCValue, count;
int      *ADC16Ptr;

AD1PCFG  = 0xFFFB;           // Only AN2 as analog input
AD1CON1  = 0x00E0;           // Internal counter triggers conversion
AD1CHS   = 0x0002;           // Connect AN2 as CH0 positive input
AD1CSSL  = 0;
AD1CON3  = 0x0F00;           // Sample time = 15Tad, Tad = Tcy/2
AD1CON2  = 0x003C;           // Set AD1IF after every 16 samples
AD1CON1bits.ADON = 1;        // turn ADC ON
while    (1)                  // repeat continuously
{
  ADCValue = 0;               // clear value
  ADC16Ptr = &ADC1BUF0;      // initialize ADC1BUF pointer
  IFS0bits.AD1IF = 0;         // clear ADC interrupt flag
  AD1CON1bits.ASAM = 1;       // auto start sampling for 31Tad
                                // then go to conversion
  while    (!IFS0bits.AD1IF);  // conversion done?
  AD1CON1bits.ASAM = 0;       // yes then stop sample/convert
  for      (count = 0; count < 16; count++) // average the 16 ADC value
  {
    ADCValue = ADCValue + *ADC16Ptr++;
    ADCValue = ADCValue >> 4;
  }
}
// repeat

```

Example 17-7: Converting a Single Channel 16 Times per Interrupt

A/D Configuration:

- Select AN0 for CH0+ input (CH0SA3:CH0SA0 = 0000)
- Select VR- for CH0- input (CH0NA = 0)
- Configure for no input scan (CSCNA = 0)
- Use only MUX A for sampling (ALTS = 0)
- Set AD1IF on every 16th sample (SMPI3:SMPI0 = 1111)
- Configure buffers for single, 16-word results (BUFM = 0)

Operational Sequence:

1. Sample MUX A Input AN0; Convert and Write to Buffer 0h
2. Sample MUX A Input AN0; Convert and Write to Buffer 1h
3. Sample MUX A Input AN0; Convert and Write to Buffer 2h
4. Sample MUX A Input AN0; Convert and Write to Buffer 3h
5. Sample MUX A Input AN0; Convert and Write to Buffer 4h
6. Sample MUX A Input AN0; Convert and Write to Buffer 5h
7. Sample MUX A Input AN0; Convert and Write to Buffer 6h
8. Sample MUX A Input AN0; Convert and Write to Buffer 7h
9. Sample MUX A Input AN0; Convert and Write to Buffer 8h
10. Sample MUX A Input AN0; Convert and Write to Buffer 9h
11. Sample MUX A Input AN0; Convert and Write to Buffer Ah
12. Sample MUX A Input AN0; Convert and Write to Buffer Bh
13. Sample MUX A Input AN0; Convert and Write to Buffer Ch
14. Sample MUX A Input AN0; Convert and Write to Buffer Dh
15. Sample MUX A Input AN0; Convert and Write to Buffer Eh
16. Sample MUX A Input AN0; Convert and Write to Buffer Fh
17. Set AD1IF flag (and generate interrupt, if enabled)
18. Repeat (1-16) after Return from Interrupt

Results Stored in Buffer (after 2 cycles):

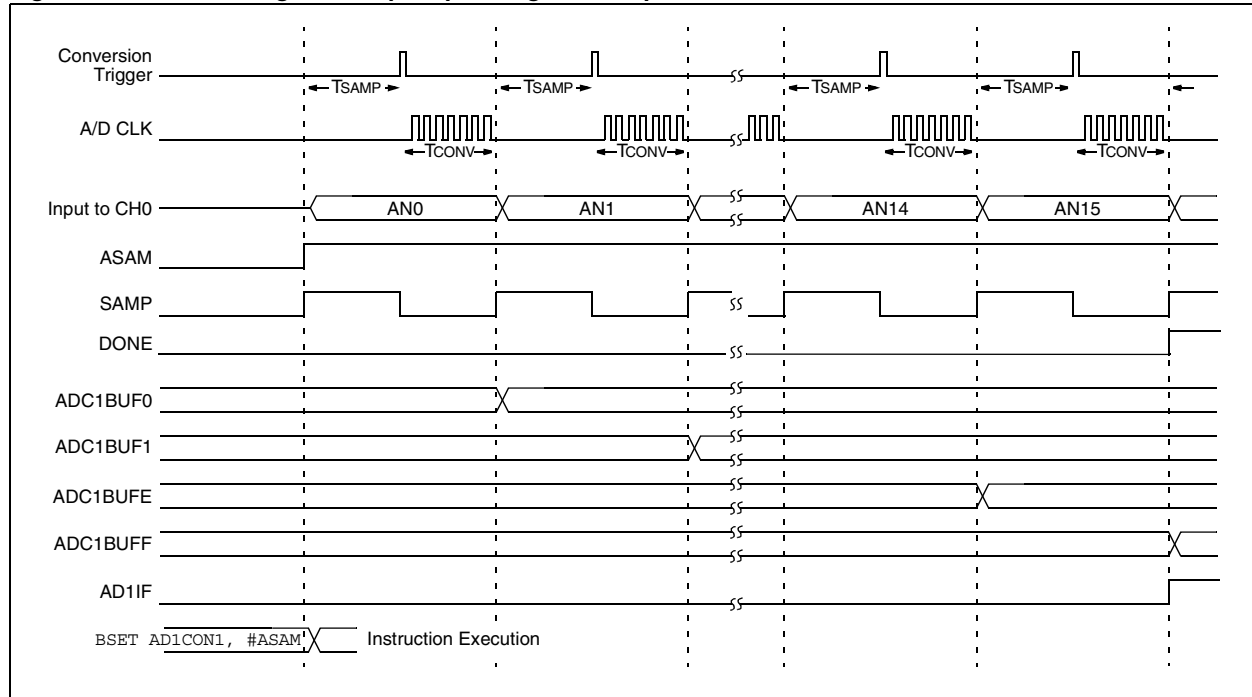
Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	AN0, Sample 1	AN0, Sample 17
ADC1BUF1	AN0, Sample 2	AN0, Sample 18
ADC1BUF2	AN0, Sample 3	AN0, Sample 19
ADC1BUF3	AN0, Sample 4	AN0, Sample 20
ADC1BUF4	AN0, Sample 5	AN0, Sample 21
ADC1BUF5	AN0, Sample 6	AN0, Sample 22
ADC1BUF6	AN0, Sample 7	AN0, Sample 23
ADC1BUF7	AN0, Sample 8	AN0, Sample 24
ADC1BUF8	AN0, Sample 9	AN0, Sample 25
ADC1BUF9	AN0, Sample 10	AN0, Sample 26
ADC1BUFA	AN0, Sample 11	AN0, Sample 27
ADC1BUFB	AN0, Sample 12	AN0, Sample 28
ADC1BUFC	AN0, Sample 13	AN0, Sample 29
ADC1BUFD	AN0, Sample 14	AN0, Sample 30
ADC1BUFE	AN0, Sample 15	AN0, Sample 31
ADC1BUFF	AN0, Sample 16	AN0, Sample 32

17.9.2 A/D Conversions While Scanning Through All Analog Inputs

Figure 17-11 and Example 17-9 illustrate a typical setup, where all available analog input channels are sampled and converted. The set CSCNA bit specifies scanning of the A/D inputs to the CH0 positive input. Other conditions are similar to **Section 17.9.1 “Sampling and Converting a Single Channel Multiple Times”**.

Initially, the AN0 input is sampled by CH0 and converted. The result is stored in the ADC1BUF0 buffer. Then, the AN1 input is sampled and converted. This process of scanning the inputs repeats 16 times until the buffer is full and then the module generates an interrupt. The entire process will then repeat.

Figure 17-11: Scanning All 16 Inputs per Single Interrupt



Example 17-8: Sampling and Converting All Channels

```

int      ADCValue, count;
int      *ADC16Ptr;

AD1PCFG    = 0x0000;           // Configure all pins as analog inputs
AD1CSSL    = 0xFFFF;          // Include all channels in scan
AD1CON1    = 0x00E0;           // Internal counter triggers conversion
AD1CON3    = 0x0F00;           // Sample time = 15Tad, Tad = Tcy/2
AD1CON2    = 0x023C;           // Set AD1IF after every 16 samples,
                                // enable scanning
AD1CON1bits.ADON = 1;         // turn ADC ON
while      (1)                 // repeat continuously
{
    ADCValue      = 0;           // clear value
    ADC16Ptr      = &ADC1BUF0;  // initialize ADC1BUF pointer
    IFS0bits.AD1IF = 0;         // clear ADC interrupt flag
    AD1CON1bits.ASAM = 1;       // auto start sampling for 31Tad
                                // then go to conversion
    while        (!IFS0bits.AD1IF); // conversion done?
    AD1CON1bits.ASAM = 0;       // yes then stop sample/convert
    for          (count = 0; count < 16; count++) // average the 16 ADC value
    ADCValue     = ADCValue + *ADC16Ptr++;
    ADCValue     = ADCValue >> 4;
}
// repeat
    
```

Example 17-9: Scanning and Converting All 16 Channels per Single Interrupt

A/D Configuration:

- Select any channel for CH0+ input (CH0SA3:CH0SA0 = xxxx)
- Select VR- for CH0- input (CH0NA = 0)
- Use only MUX A for sampling (ALTS = 0)
- Configure MUX A for input scan (CSCNA = 1)
- Include all analog channels in scanning (AD1CSSL = 1111 1111 1111 1111)
- Set AD1IF on every 16th sample (SMPI3:SMPI0 = 1111)
- Configure buffers for single, 16-word results (BUFM = 0)

Operational Sequence:

1. Sample MUX A Input AN0; Convert and Write to Buffer 0h
2. Sample MUX A Input AN1; Convert and Write to Buffer 1h
3. Sample MUX A Input AN2; Convert and Write to Buffer 2h
4. Sample MUX A Input AN3; Convert and Write to Buffer 3h
5. Sample MUX A Input AN4; Convert and Write to Buffer 4h
6. Sample MUX A Input AN5; Convert and Write to Buffer 5h
7. Sample MUX A Input AN6; Convert and Write to Buffer 6h
8. Sample MUX A Input AN7; Convert and Write to Buffer 7h
9. Sample MUX A Input AN8; Convert and Write to Buffer 8h
10. Sample MUX A Input AN9; Convert and Write to Buffer 9h
11. Sample MUX A Input AN10; Convert and Write to Buffer Ah
12. Sample MUX A Input AN11; Convert and Write to Buffer Bh
13. Sample MUX A Input AN12; Convert and Write to Buffer Ch
14. Sample MUX A Input AN13; Convert and Write to Buffer Dh
15. Sample MUX A Input AN14; Convert and Write to Buffer Eh
16. Sample MUX A Input AN15; Convert and Write to Buffer Fh
17. Set AD1IF flag (and generate interrupt, if enabled)
18. Repeat (1-16) after Return from Interrupt

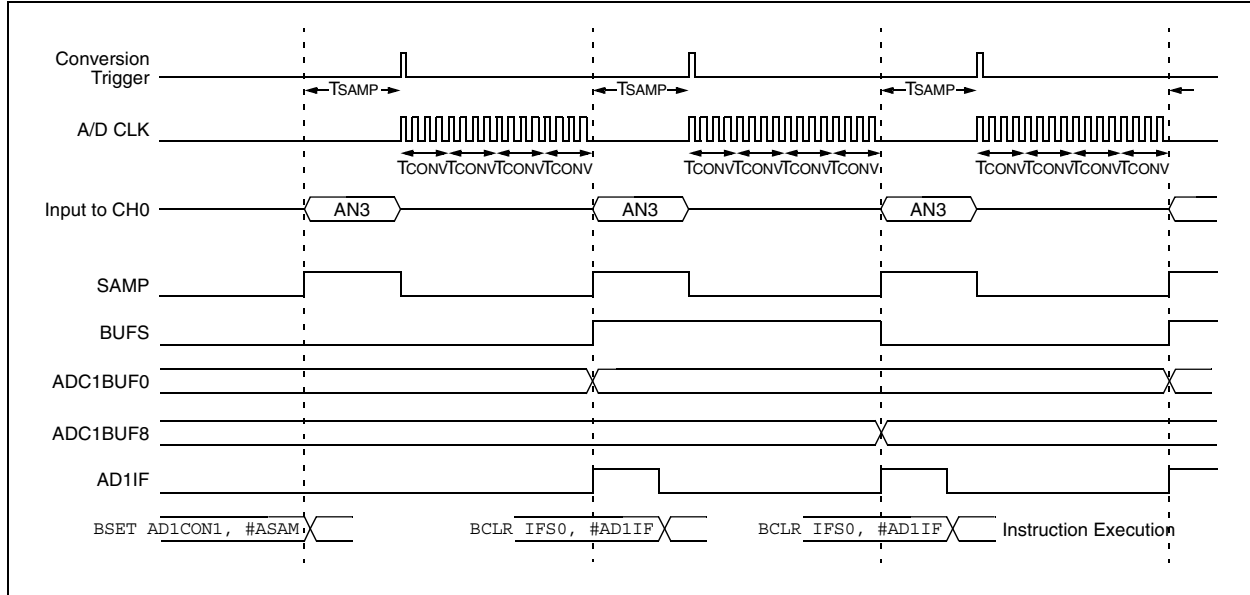
Results Stored in Buffer (after 2 cycles):

Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	Sample 1 (AN0, Sample 1)	Sample 17 (AN0, Sample 2)
ADC1BUF1	Sample 2 (AN1, Sample 1)	Sample 18 (AN1, Sample 2)
ADC1BUF2	Sample 3 (AN2, Sample 1)	Sample 19 (AN2, Sample 2)
ADC1BUF3	Sample 4 (AN3, Sample 1)	Sample 20 (AN3, Sample 2)
ADC1BUF4	Sample 5 (AN4, Sample 1)	Sample 21 (AN4, Sample 2)
ADC1BUF5	Sample 6 (AN5, Sample 1)	Sample 22 (AN5, Sample 2)
ADC1BUF6	Sample 7 (AN6, Sample 1)	Sample 23 (AN6, Sample 2)
ADC1BUF7	Sample 8 (AN7, Sample 1)	Sample 24 (AN7, Sample 2)
ADC1BUF8	Sample 9 (AN8, Sample 1)	Sample 25 (AN8, Sample 2)
ADC1BUF9	Sample 10 (AN9, Sample 1)	Sample 26 (AN9, Sample 2)
ADC1BUFA	Sample 11 (AN10, Sample 1)	Sample 27 (AN10, Sample 2)
ADC1BUFB	Sample 12 (AN11, Sample 1)	Sample 28 (AN11, Sample 2)
ADC1BUFC	Sample 13 (AN12, Sample 1)	Sample 29 (AN12, Sample 2)
ADC1BUFD	Sample 14 (AN13, Sample 1)	Sample 30 (AN13, Sample 2)
ADC1BUFE	Sample 15 (AN14, Sample 1)	Sample 31 (AN14, Sample 2)
ADC1BUFF	Sample 16 (AN15, Sample 1)	Sample 32 (AN15, Sample 2)

17.9.3 Using Dual, 8-Word Buffers

Figure 17-12 and Example 17-10 demonstrate using dual, 8-word buffers and alternating the buffer fill. Setting the BUFM bit enables dual, 8-word buffers. In this example, an interrupt is generated after each sample. The BUFM setting does not affect other operational parameters. First, the conversion sequence starts filling the buffer at ADC1BUF0. After the first interrupt occurs, the buffer begins to fill at ADC1BUF8. The BUFS status bit is set and cleared alternately after each interrupt.

Figure 17-12: Converting a Single Channel, Once per Interrupt Using Dual, 8-Word Buffers



Example 17-10: Converting a Single Channel Once per Interrupt, Dual Buffer Mode

A/D Configuration:

- Select AN3 for CH0+ input (CH0SA3:CH0SA0 = 0011)
- Select VR- for CH0- input (CH0NA = 0)
- Configure for no input scan (CSCNA = 0)
- Use only MUX A for sampling (ALTS = 0)
- Set AD1IF on every sample (SMPI3:SMPI0 = 0000)
- Configure buffer as dual, 8-word segments (BUFM = 1)

Operational Sequence:

1. Sample MUX A Input AN3; Convert and Write to Buffer 0h
2. Set AD1IF flag (and generate interrupt, if enabled); write access automatically switches to alternate buffer
3. Sample MUX A Input AN3; Convert and Write to Buffer 8h
4. Set AD1IF flag (and generate interrupt, if enabled); write access automatically switches to alternate buffer
5. Repeat (1-4)

Results Stored in Buffer (after 2 cycles):

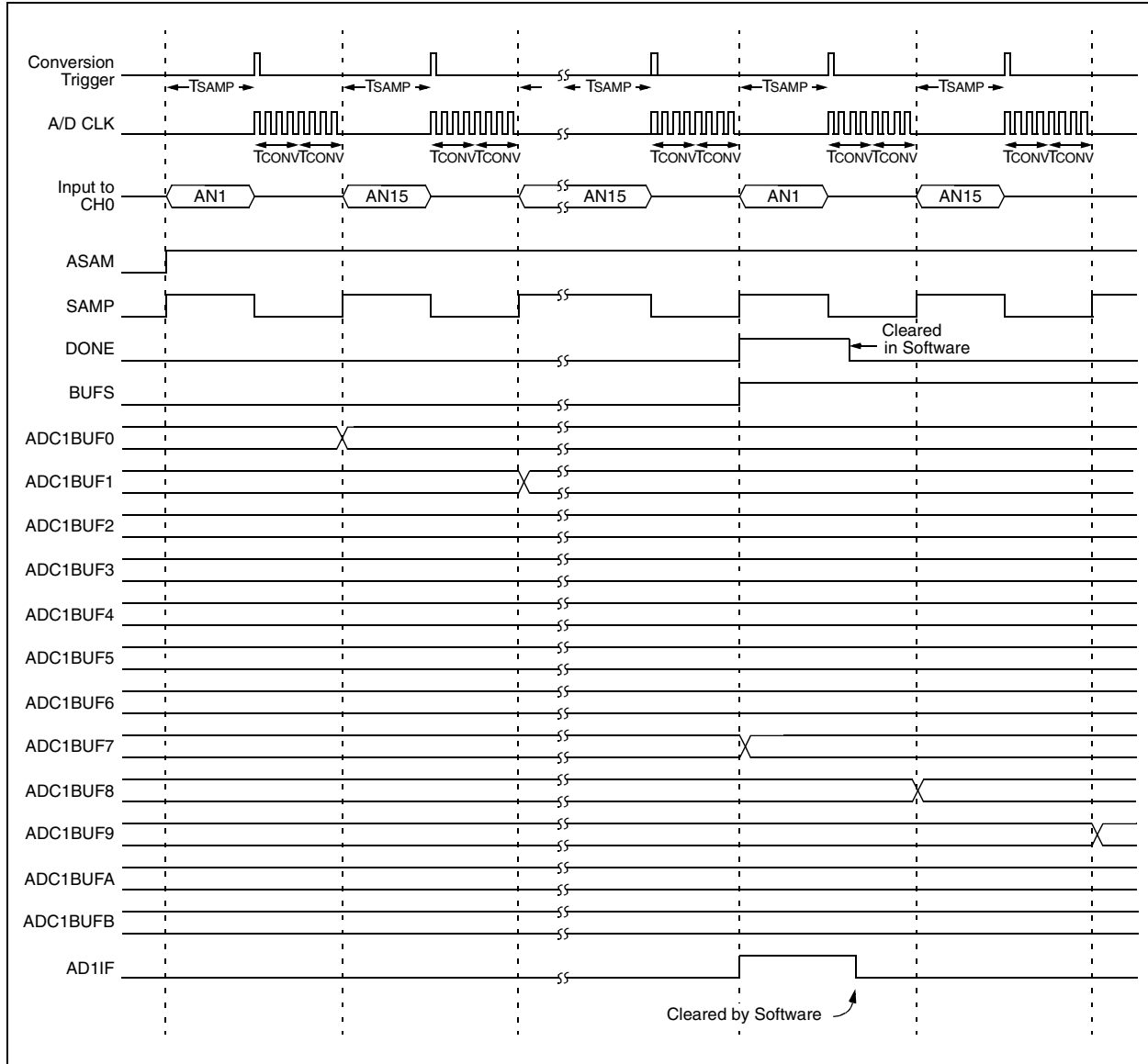
Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	Sample 1 (AN3, Sample 1)	(undefined)
ADC1BUF1	(undefined)	(undefined)
ADC1BUF2	(undefined)	(undefined)
ADC1BUF3	(undefined)	(undefined)
ADC1BUF4	(undefined)	(undefined)
ADC1BUF5	(undefined)	(undefined)
ADC1BUF6	(undefined)	(undefined)
ADC1BUF7	(undefined)	(undefined)
ADC1BUF8	(undefined)	Sample 2 (AN3, Sample 2)
ADC1BUF9	(undefined)	(undefined)
ADC1BUFA	(undefined)	(undefined)
ADC1BUFB	(undefined)	(undefined)
ADC1BUFC	(undefined)	(undefined)
ADC1BUFD	(undefined)	(undefined)
ADC1BUFE	(undefined)	(undefined)
ADC1BUFF	(undefined)	(undefined)

17.9.4 Using Alternating MUX A and MUX B Input Selections

Figure 17-13 and Example 17-11 demonstrate alternate sampling of the inputs assigned to MUX A and MUX B. Setting the ALTS bit enables alternating input selections. The first sample uses the MUX A inputs specified by the CH0SA and CH0NA bits. The next sample uses the MUX B inputs specified by the CH0SB and CH0NB bits.

This example also demonstrates use of the dual, 8-word buffers. An interrupt occurs after every 8th sample, resulting in filling 8 words into the buffer on each interrupt.

Figure 17-13: Converting Two Inputs Using Alternating Input Selections



Example 17-11: Converting Two Inputs by Alternating MUX A and MUX B

A/D Configuration:

- Select AN1 for MUX A CH0+ input (CH0SA3:CH0SA0 = 0001)
- Select VR- for MUX A CH0- input (CH0NA = 0)
- Configure for no input scan (CSCNA = 0)
- Select AN15 for MUX B CH0+ input (CH0SB3:CH0SB0 = 1111)
- Select VR- for MUX B CH0- input (CH0NB = 0)
- Alternate MUX A and MUX B for sampling (ALTS = 1)
- Set AD1IF on every 8th sample (SMPI3:SMPI0 = 0111)
- Configure buffer as two, 8-word segments (BUFM = 1)

Operational Sequence:

1. Sample MUX A Input AN1; Convert and Write to Buffer 0h
2. Sample MUX B Input AN15; Convert and Write to Buffer 1h
3. Sample MUX A Input AN1; Convert and Write to Buffer 2h
4. Sample MUX B Input AN15; Convert and Write to Buffer 3h
5. Sample MUX A Input AN1; Convert and Write to Buffer 4h
6. Sample MUX B Input AN15; Convert and Write to Buffer 5h
7. Sample MUX A Input AN1; Convert and Write to Buffer 6h
8. Sample MUX B Input AN15; Convert and Write to Buffer 7h
9. Set AD1IF flag (and generate interrupt, if enabled); write access automatically switches to alternate buffer
10. Repeat (1-9); resume writing to buffer with Buffer 8h (first address of alternate buffer)

Results Stored in Buffer (after 2 cycles):

Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	Sample 1 (AN1, Sample 1)	(undefined)
ADC1BUF1	Sample 2 (AN15, Sample 1)	(undefined)
ADC1BUF2	Sample 3 (AN1, Sample 2)	(undefined)
ADC1BUF3	Sample 4 (AN15, Sample 2)	(undefined)
ADC1BUF4	Sample 5 (AN1, Sample 3)	(undefined)
ADC1BUF5	Sample 6 (AN15, Sample 3)	(undefined)
ADC1BUF6	Sample 7 (AN1, Sample 4)	(undefined)
ADC1BUF7	Sample 8 (AN15, Sample 4)	(undefined)
ADC1BUF8	(undefined)	Sample 9 (AN1, Sample 5)
ADC1BUF9	(undefined)	Sample 10 (AN15, Sample 5)
ADC1BUFA	(undefined)	Sample 11 (AN1, Sample 6)
ADC1BUFB	(undefined)	Sample 12 (AN15, Sample 6)
ADC1BUFC	(undefined)	Sample 13 (AN1, Sample 7)
ADC1BUFD	(undefined)	Sample 14 (AN15, Sample 7)
ADC1BUFE	(undefined)	Sample 15 (AN1, Sample 8)
ADC1BUFF	(undefined)	Sample 16 (AN15, Sample 8)

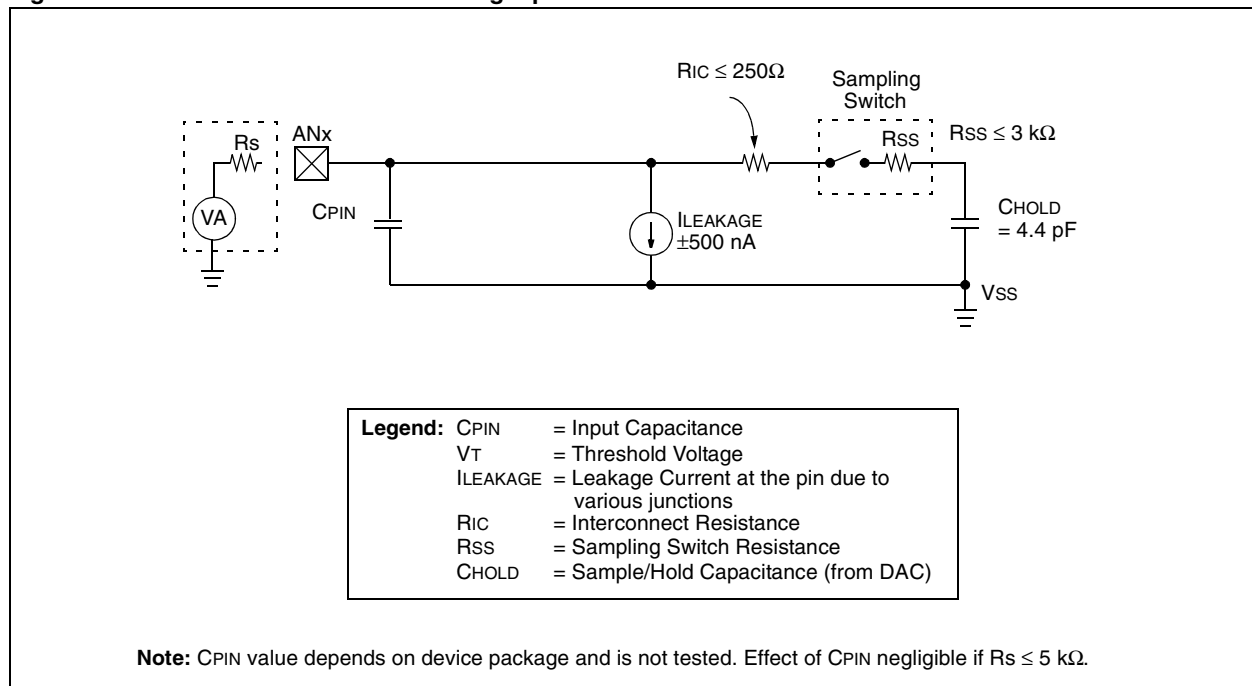
17.10 A/D SAMPLING REQUIREMENTS

The analog input model of the 10-bit A/D converter is shown in Figure 18-11. The total sampling time for the A/D is a function of the holding capacitor charge time.

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the voltage level on the analog input pin. The source impedance (Rs), the interconnect impedance (Ric) and the internal sampling switch (Rss) impedance combine to directly affect the time required to charge CHOLD. The combined impedance of the analog sources must, therefore, be small enough to fully charge the holding capacitor within the chosen sample time. To minimize the effects of pin leakage currents on the accuracy of the A/D converter, the maximum recommended source impedance, Rs, is 2.5 kΩ. After the analog input channel is selected (changed), this sampling function must be completed prior to starting the conversion. The internal holding capacitor will be in a discharged state prior to each sample operation.

At least 1 TAD time period should be allowed between conversions for the sample time. For more details, see **Section 17.16 “Electrical Specifications”**.

Figure 17-14: 10-Bit A/D Converter Analog Input Model

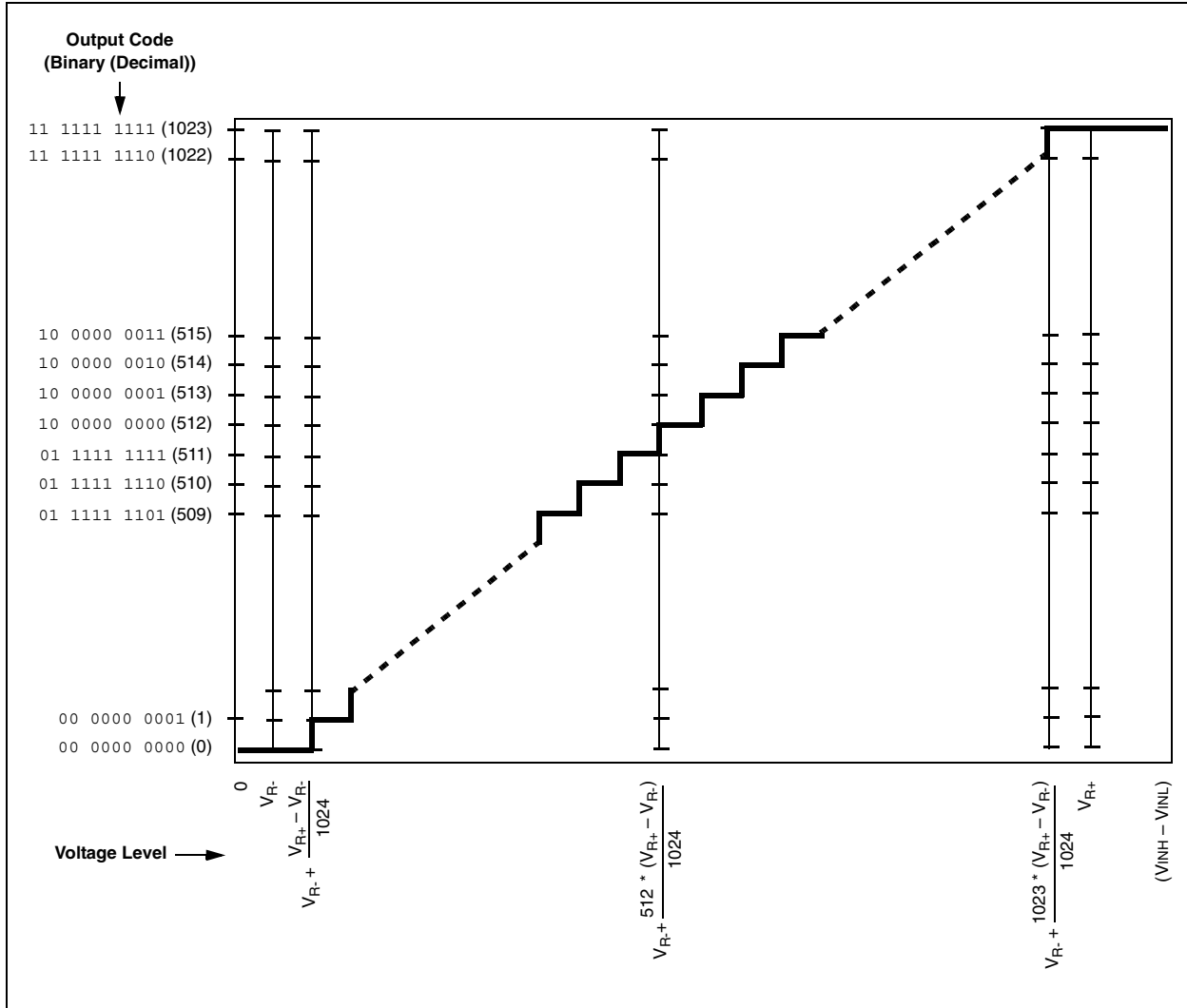


17.11 TRANSFER FUNCTION

The transfer function of the A/D converter is shown in Figure 17-15. The difference of the input voltages, $(V_{INH} - V_{INL})$, is compared to the reference, $((V_{R+}) - (V_{R-}))$.

- The first code transition occurs when the input voltage is $((V_{R+}) - (V_{R-}))/1024$ or 1.0 LSB.
- The 00 0000 0001 code is centered at $V_{R-} + (1.5 * ((V_{R+}) - (V_{R-}))/1024)$.
- The 10 0000 0000 code is centered at $V_{REFL} + (512.5 * ((V_{R+}) - (V_{R-}))/1024)$.
- An input voltage less than $V_{R-} + (((V_{R-}) - (V_{R-}))/1024)$ converts as 00 0000 0000.
- An input voltage greater than $(V_{R-}) + (1023 * ((V_{R+}) - (V_{R-}))/1024)$ converts as 11 1111 1111.

Figure 17-15: A/D Transfer Function



17.12 A/D ACCURACY/ERROR

Refer to **Section 17.18 “Related Application Notes”** for a list of documents that discuss A/D accuracy.

17.13 OPERATION DURING SLEEP AND IDLE MODES

Sleep and Idle modes are useful for minimizing conversion noise because the digital activity of the CPU, buses and other peripherals is minimized.

17.13.1 CPU Sleep Mode Without RC A/D Clock

When the device enters Sleep mode, all clock sources to the module are shut down and stay at logic '0'.

If Sleep occurs in the middle of a conversion, the conversion is aborted unless the A/D is clocked from its internal RC clock generator. The converter will not resume a partially completed conversion on exiting from Sleep mode.

Register contents are not affected by the device entering or leaving Sleep mode.

17.13.2 CPU Sleep Mode With RC A/D Clock

The A/D module can operate during Sleep mode if the A/D clock source is set to the internal A/D RC oscillator ($ADRC = 1$). This eliminates digital switching noise from the conversion. When the conversion is completed, the DONE bit will be set and the result loaded into the A/D Result Buffer, ADC1BUF.

If the A/D interrupt is enabled ($AD1IE = 1$), the device will wake-up from Sleep when the A/D interrupt occurs. Program execution will resume at the A/D Interrupt Service Routine if the A/D interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the `PWRSVAV` instruction that placed the device in Sleep mode.

If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

To minimize the effects of digital noise on the A/D module operation, the user should select a conversion trigger source that ensures the A/D conversion will take place in Sleep mode. The automatic conversion trigger option can be used for sampling and conversion in Sleep ($SSRC2:SSRC0 = 111$). To use the automatic conversion option, the ADON bit should be set in the instruction prior to the `PWRSVAV` instruction.

Note: For the A/D module to operate in Sleep, the A/D clock source must be set to RC ($ADRC = 1$).

17.13.3 A/D Operation During CPU Idle Mode

For the A/D, the ADSIDL bit ($AD1CON1<13>$) selects if the module will stop on Idle or continue on Idle. If $ADSIDL = 0$, the module will continue normal operation when the device enters Idle mode. If the A/D interrupt is enabled ($AD1IE = 1$), the device will wake-up from Idle mode when the A/D interrupt occurs. Program execution will resume at the A/D Interrupt Service Routine if the A/D interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the `PWRSVAV` instruction that placed the device in Idle mode.

If $ADSIDL = 1$, the module will stop in Idle. If the device enters Idle mode in the middle of a conversion, the conversion is aborted. The converter will not resume a partially completed conversion on exiting from Idle mode.

17.13.4 Peripheral Module Disable (PMD) Register

The Peripheral Module Disable (PMD) registers provide a method to disable the A/D module by stopping all clock sources supplied to that module. When a peripheral is disabled via the appropriate PMD control bit, the peripheral is in a minimum power consumption state. The control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect and read values will be invalid. A peripheral module will only be enabled if the ADC1MD bit in the the PMDx register is cleared.

17.14 EFFECTS OF A RESET

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. All pins that are multiplexed with analog inputs will be configured as analog inputs. The corresponding TRIS bits will be set.

The values in the ADC1BUF registers are not initialized during a Power-on Reset; they will contain unknown data.

17.15 REGISTER MAPS

A summary of the registers associated with the PIC24F 10-Bit A/D Converter is provided in Table 17-3.

Table 17-3: ADC Register Map

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0																	xxxxx
ADC1BUF1																	xxxxx
ADC1BUF2																	xxxxx
ADC1BUF3																	xxxxx
ADC1BUF4																	xxxxx
ADC1BUF5																	xxxxx
ADC1BUF6																	xxxxx
ADC1BUF7																	xxxxx
ADC1BUF8																	xxxxx
ADC1BUF9																	xxxxx
ADC1BUFA																	xxxxx
ADC1BUFB																	xxxxx
ADC1BUFC																	xxxxx
ADC1BUFD																	xxxxx
ADC1BUFE																	xxxxx
ADC1BUFF																	xxxxx
AD1CON1	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	VCFG2	VCFG1	VCFG0	OFFCAL	—	CSCNA	—	—	BUFS	—	SMP13	SMP12	SMP11	SMP10	BUFM	ALTS	0000
AD1CON3	ADRC	—	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1TCHS	CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	—	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFG	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

PIC24F Family Reference Manual

17.16 ELECTRICAL SPECIFICATIONS

Figure 17-16: A/D Conversion Timing

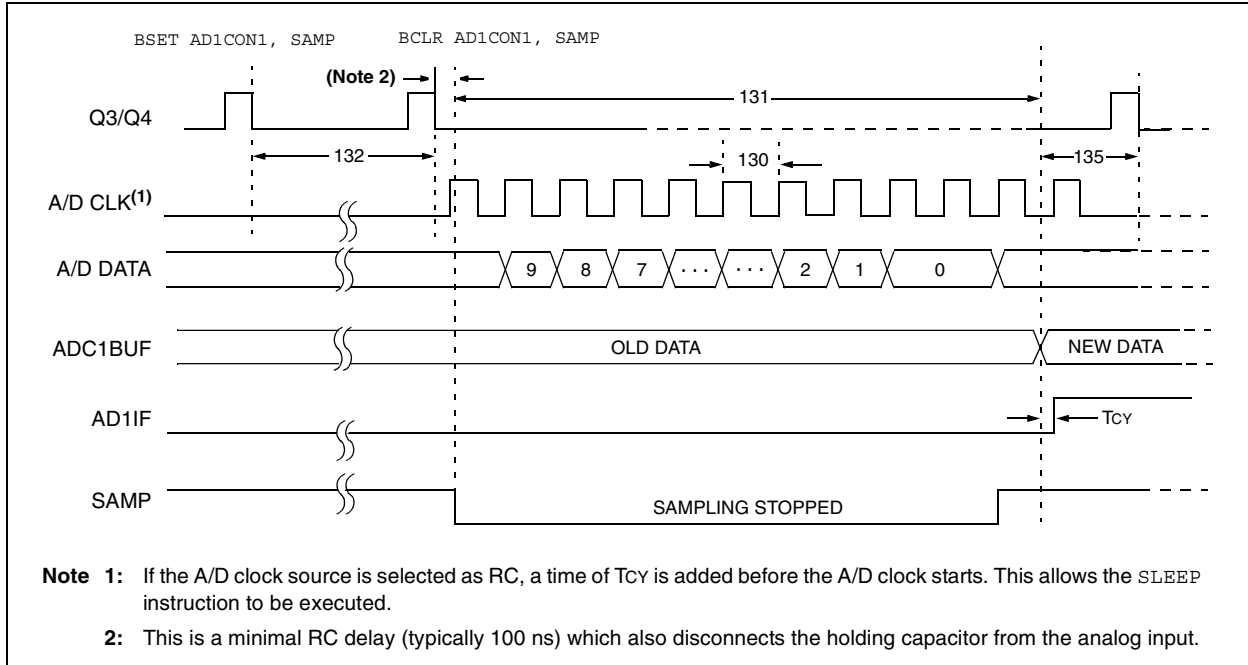


Table 17-4: A/D Conversion Requirements

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
AD130	TAD	A/D Clock Period	75	—	—	ns	TOSC based
			—	250	—	ns	A/D RC mode
AD131	T _{cnv}	Conversion Time (not including acquisition time)	11	—	12	TAD	(Note 1)
AD132	T _{acq}	Acquisition Time	750	—	—	ns	(Note 2)
AD135	T _{swc}	Switching Time from Convert to Sample	—	—	(Note 3)		
AD137	T _{dis}	Discharge Time	0.5	—	—	TAD	

Note 1: The ADC1BUF register may be read on the following T_{cy} cycle.

Note 2: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (V_{DD} to V_{SS} or V_{SS} to V_{DD}).

Note 3: On the following cycle of the device clock.

17.17 DESIGN TIPS

Question 1: *How can I optimize the system performance of the A/D converter?*

Answer: There are three main things to consider in optimizing A/D performance:

1. Make sure you are meeting all of the timing specifications. If you are turning the module off and on, there is a minimum delay you must wait before taking a sample. If you are changing input channels, there is a minimum delay you must wait for this as well, and finally, there is TAD, which is the time selected for each bit conversion. This is selected in AD1CON3 and should be within a certain range, as specified in **Section 17.16 “Electrical Specifications”**. If TAD is too short, the result may not be fully converted before the conversion is terminated, and if TAD is made too long, the voltage on the sampling capacitor can decay before the conversion is complete. These timing specifications are provided in the **“Electrical Characteristics”** section of the device data sheets.
2. Often, the source impedance of the analog signal is high (greater than 2.5 k Ω), so the current drawn from the source by leakage, and to charge the sample capacitor, can affect accuracy. If the input signal does not change too quickly, try putting a 0.1 μ F capacitor on the analog input. This capacitor will charge to the analog voltage being sampled and supply the instantaneous current needed to charge the 4.4 pF internal holding capacitor.
3. Put the device into Sleep mode before the start of the A/D conversion. The RC clock source selection is required for conversions in Sleep mode. This technique increases accuracy, because digital noise from the CPU and other peripherals is minimized.

Question 2: *Do you know of a good reference on A/D converters?*

Answer: A good reference for understanding A/D conversions is the *“Analog-Digital Conversion Handbook”* third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

Question 3: *My combination of channels/samples and samples/interrupt is greater than the size of the buffer. What will happen to the buffer?*

Answer: This configuration is not recommended. The buffer will contain unknown results.

17.18 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the 10-Bit A/D Converter module are:

Title	Application Note #
Using the Analog-to-Digital (A/D) Converter	AN546
Four-Channel Digital Voltmeter with Display and Keyboard	AN557
Understanding A/D Converter Performance Specifications	AN693

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

17.19 REVISION HISTORY

Revision A (April 2006)

This is the initial released revision of this document.

NOTES: