Section 51. 12-Bit A/D Converter with Threshold Detect

HIGHLIGHTS

This section of the manual contains the following major topics:

51.1 INTRODUCTION

The PIC24F 12-bit A/D Converter has the following key features:

- Successive Approximation Register (SAR) Conversion
- Conversion Speeds of up to 200 ksps
- Up to 32 Analog Input Channels (Internal and External)
- Multiple Internal Reference Input Channels
- External Voltage Reference Input Pins
- Unipolar Differential Sample-and-Hold (S/H) Amplifier
- Automated Threshold Scan and Compare Operation to Pre-Evaluate Conversion Results
- Selectable Conversion Trigger Source
- Fixed-Length (one word per channel), Configurable Conversion Result Buffer
- Four Options for Results Alignment
- Configurable Interrupt Generation
- Operation During CPU Sleep and Idle modes

The 12-bit A/D Converter module is an enhanced version of the 10-bit module offered in some PIC24 devices. Both modules are Successive Approximation Register (SAR) Converters at their cores, surrounded by a range of hardware features for flexible configuration. This version of the module extends functionality by providing 12-bit resolution, a wider range of automatic sampling options, tighter integration with other analog modules, such as the CTMU, and a configurable results buffer. This module also includes a unique Threshold Detect feature that allows the module itself to make simple decisions based on the conversion results.

As before, an internal Sample-and-Hold (S/H) amplifier acquires a sample of an input signal, then holds that value constant during the conversion process. A combination of input multiplexers selects the signal to be converted from up to 32 analog inputs, both external (analog input pins) and internal (e.g., on-chip voltage references and other analog modules). The whole multiplexer path includes provision for differential analog input, although, with a limited number of negative input pins. The sampled voltage is held and converted to a digital value, which strictly speaking, represents the ratio of that input voltage to a reference voltage. Configuration choices allow connection of an external reference or use of the device power and ground (AV_{DD} and AV_{SS}). Reference and input signal pins are assigned differently depending on the particular device.

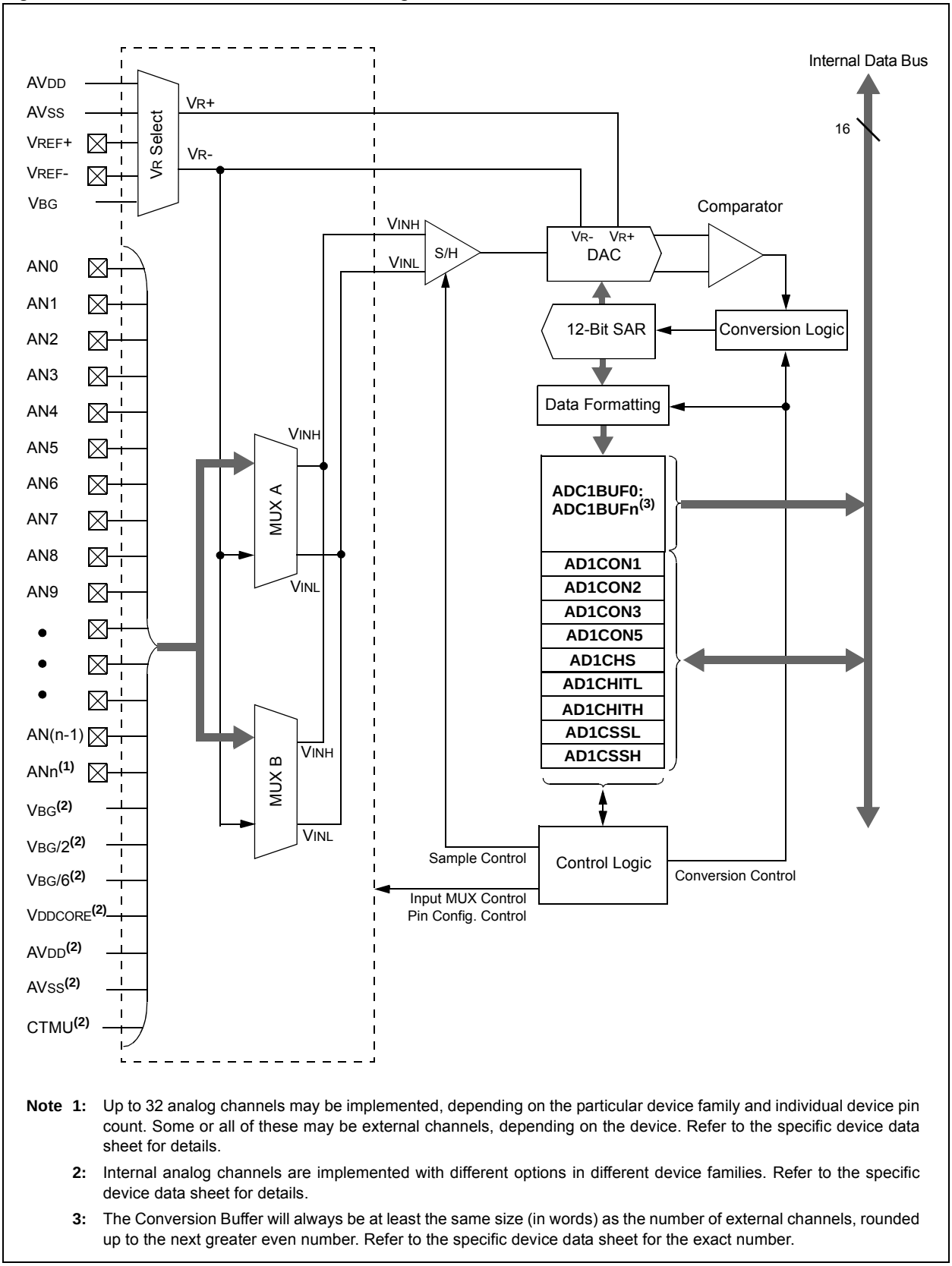
An array of timing and control selections allow the user to create flexible scanning sequences. Conversions can be started individually by program control, continuously free running, or triggered by selected hardware events. A single channel may be repeatedly converted; alternate conversions may be performed on two channels, or any or all of the channels may be sequentially scanned and converted according to a user-defined bit map. The resulting conversion output is a 12-bit digital number which can be signed or unsigned, left or right justified. (In some devices, a user-selectable resolution of 10 bits is available; in other devices, 10-bit resolution is the only option available).

Conversions are automatically stored in a dedicated buffer, allowing for multiple successive readings to be taken before software service is needed. The buffer can be configured to function as a FIFO buffer, or as a channel indexed buffer. In FIFO mode, the buffer can be split into two equal sections for simultaneous conversion and read operations. In Indexed mode, the buffer can use the Threshold Scan feature to determine if a conversion meets specific user-defined criteria, storing or discarding the converted value as appropriate, and then set semaphore flags to indicate the event. This allows conversions to occur in low-power modes when the CPU is inactive, waking the device only when specific conditions have occurred.

The module sets its interrupt flag after a selectable number of conversions, when the buffer can be read, or after a successful Threshold Detect comparison. After the interrupt, the sequence restarts at the beginning of the buffer. When the interrupt flag is set, according to the earlier selection, scan selections and the Output Buffer Pointer return to their starting positions.

A simplified block diagram for the module is shown in Figure 51-1.

Figure 51-1: 12-Bit A/D Converter Block Diagram



Note 1: Up to 32 analog channels may be implemented, depending on the particular device family and individual device pin count. Some or all of these may be external channels, depending on the device. Refer to the specific device data sheet for details.

2: Internal analog channels are implemented with different options in different device families. Refer to the specific device data sheet for details.

3: The Conversion Buffer will always be at least the same size (in words) as the number of external channels, rounded up to the next greater even number. Refer to the specific device data sheet for the exact number.

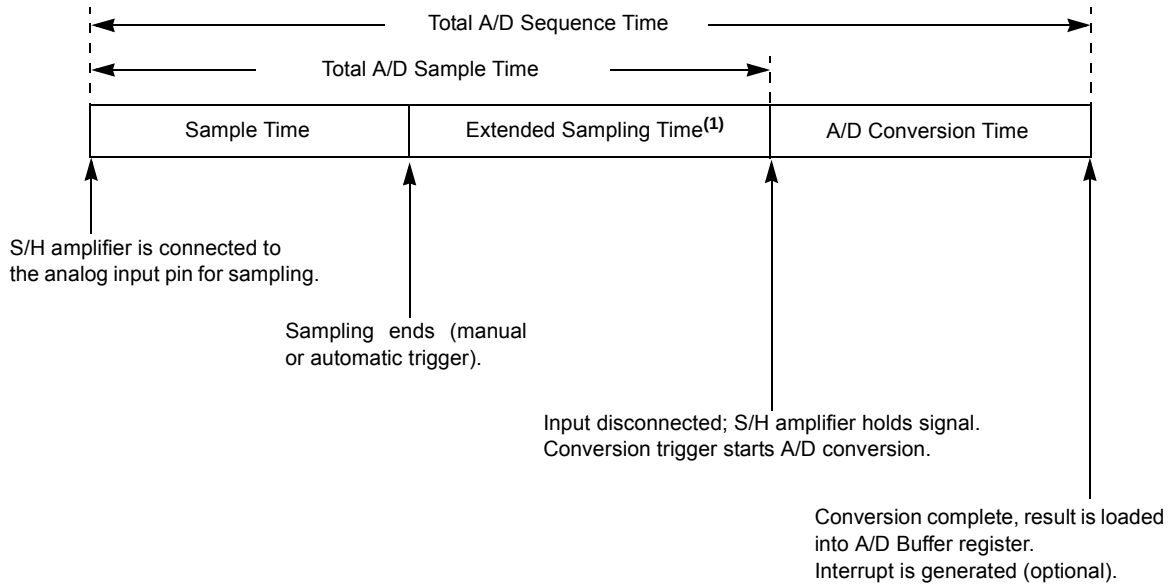
51.2 A/D TERMINOLOGY AND CONVERSION SEQUENCE

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 automatically by the A/D Converter's hardware or under direct program control. There is a minimum sample time to ensure that the S/H amplifier will give sufficient accuracy for the A/D conversion.

The conversion trigger ends the sampling time and begins an A/D conversion or a repeating sequence. The conversion trigger sources can be taken from a variety of hardware sources or can be controlled directly 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 continuous, automatic conversions without software intervention. When automatic sampling is used, an extended sampling interval is extended between the time the sampling ends and the conversion starts.

Conversion time is the time required for the A/D Converter to convert the voltage held by the S/H amplifier. An A/D conversion requires one A/D clock cycle (TAD) to convert each bit of the result, plus two additional clock cycles, or a total of 14 TAD cycles for a 12-bit conversion. When the conversion is complete, the result is loaded into one of the 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(s) and the A/D conversion time provides the total A/D sequence time. Figure 51-2 shows the basic conversion sequence and the relationship between intervals.

Figure 51-2: A/D Sample/Convert Sequence



Note 1: In Automatic Sampling modes, Extended Sampling Time is added to the sequence when the value for the Auto-Sampling Time is greater than 0. Otherwise, sampling ends and conversion starts whenever the SAMP bit is cleared.

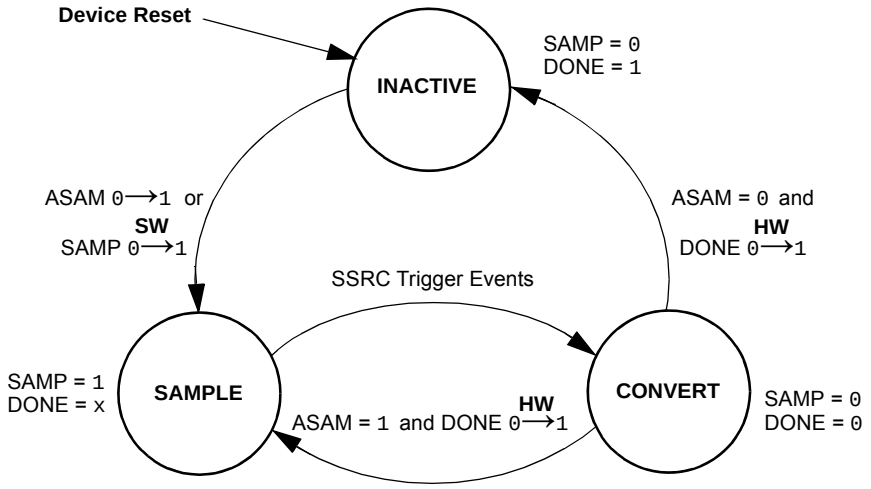
51.2.1 Operation as a State Machine

The A/D conversion process can be thought of in terms of a finite state machine (Figure 51-3). The sample state represents the time that the input channel is connected to the S/H amplifier and the signal is passed to the converter input. The convert state is transitory; the module enters this state as soon as it exits the sample state and transitions to a different state when that is done. The inactive state is the default state prior to module initialization and following a software-controlled conversion; it can be avoided in operation by using Auto-Sample mode. Machine states are identified by the state of several control and status bits in AD1CON1 (Register 51-1).

If the module is configured for Auto-Sample mode, the operation “ping-pongs” continuously between the sample and convert states. The module automatically selects the input channels to be sampled (if channel scanning is enabled), while the selected conversion trigger source paces the entire operation. Any time that Auto-Sample mode is not used for conversion, it is available for the sample state. The user needs to make certain that acquisition time is sufficient, in addition to accounting for the normal concerns about system throughput.

Whenever the issue of sampling time is important, the significant event is the transition from sample to convert state. This is the point where the Sample-and-Hold aperture closes, and it is essentially the signal value at this instant, which is applied to the A/D for conversion to digital.

Figure 51-3: A/D Module State Machine Model



Legend: **HW** = Automatic Hardware event; **SW** = Software Controlled event.

Note: See Register 51-1 for definitions of the ASAM, SAMP, DONE and SSRC<3:0> bits.

51.3 REGISTERS

The 12-bit A/D Converter module uses up to 43 registers for its operation. All registers are mapped in the data memory space.

51.3.1 Control Registers

Depending on the specific device, the module has up to eleven control and status registers:

- AD1CON1: A/D Control Register 1
- AD1CON2: A/D Control Register 2
- AD1CON3: A/D Control Register 3
- AD1CON5: A/D Control Register 5
- AD1CHS: A/D Input Channel Select Register
- AD1CHITH and AD1CHITL: A/D Scan Compare Hit Registers
- AD1CSSL and AD1CSSH: A/D Input Scan Select Registers
- AD1CTMENH and AD1CTMENL: CTMU Enable Registers

The AD1CON1, AD1CON2 and AD1CON3 registers (Register 51-1, Register 51-2 and Register 51-3) 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 AD1CON5 register (Register 51-4) specifically controls features of Threshold Detect operation, including its functioning in power-saving modes.

The AD1CHS register (Register 51-5) 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 AD1CHITH and AD1CHITL registers (Register 51-6 and Register 51-7) are semaphore registers used with Threshold Detect operations. The status of individual bits, or bit pairs in some cases, indicate if a match condition has occurred. Their use is described in more detail in **Section 51.9 "Threshold Detect Operation"**. AD1CHITL is always implemented, whereas AD1CHITH may not be implemented in devices with 16 or fewer channels.

The AD1CSSH/L registers (Register 51-8 and Register 51-9) select the channels to be included for sequential scanning.

The AD1CTMENH/L registers (Register 51-10 and Register 51-11) select the channel(s) to be used by the CTMU during conversions. Selecting a particular channel allows the A/D Converter to control the CTMU (particularly, its current source) and read its data through that channel. AD1CTMENL is always implemented, whereas AD1CTMENH may not be implemented in devices with 16 or fewer channels.

51.3.2 A/D Result Buffers

The module incorporates a multi-word, dual port RAM, called ADC1BUF. The buffer is composed of at least the same number of word locations as there are external analog channels for a particular device, with a maximum number of 32. The number of buffer addresses is always even. Each of the locations is mapped into the data memory space and is separately addressable.The buffer locations are referred to as ADC1BUF0 through ADC1BUFn (up to 31).

The A/D result buffers are both readable and writable. When the module is active (AD1CON<15> = 1), the buffers are read-only, and store the results of A/D conversions. When the module is inactive (AD1CON<15> = 0), the buffers are both readable and writable. In this state, writing to a buffer location programs a conversion threshold for Threshold Detect operations, as described in **Section 51.9.2 "Setting Comparison Thresholds"**.

Buffer contents are not cleared when the module is deactivated with the ADON bit (AD1CON1<15>). Conversion results and any programmed threshold values are maintained when ADON is set or cleared.

Register 51-1: AD1CON1: A/D Control Register 1

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0/U	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	MODE12(1)	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0 HSC	R/C-0 HSC
SSRC3(2)	SSRC2(2)	SSRC1(2)	SSRC0(2)	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-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 off

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 12-11 **Unimplemented:** Read as '0'

bit 10 **MODE12:** 12-Bit Operation Mode bit(1)
1 = 12-bit A/D operation
0 = 10-bit A/D operation

bit 9-8 **FORM<1:0>:** Data Output Format bits (see formats following)
11 = Fractional result, signed, left-justified
10 = Absolute fractional result, unsigned, left-justified
01 = Decimal result, signed, right-justified
00 = Absolute decimal result, unsigned, right-justified

bit 7-4 **SSRC<3:0>:** Sample Clock Source Select bits(2)
1111 = The SAMP bit is cleared by a rising edge of Trigger Input 15
1110 = The SAMP bit is cleared by a rising edge of Trigger Input 14
• • •
1001 = The SAMP bit is cleared by a rising edge of Trigger Input 9
1000 = The SAMP bit is cleared by a rising edge of Trigger Input 8
0111 = The SAMP bit is cleared after SAMC<4:0> number of TAD clocks following the SAMP bit being set (Auto-Convert mode). No Extended Sample Time is present.
0110 = The SAMP bit is cleared by a rising edge of Trigger Input 6
• • •
0001 = The SAMP bit is cleared by a rising edge of Trigger Input 1
0000 = The SAMP bit must be cleared by software

bit 3 **Unimplemented:** Read as '0'

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after last conversion; SAMP bit is auto-set
0 = Sampling begins when SAMP bit is manually set

Note 1: This bit is implemented only in modules with user-selectable resolution. For A/D modules with fixed 10-bit or 12-bit resolution, this bit is unimplemented and should be maintained as '1' or '0'. Refer to the device data sheet for the specific setting of this bit.

2: Sample clock trigger sources are defined differently for each device family; not all trigger sources may be defined. Refer to the specific data sheet for details.

Register 51-1: AD1CON1: A/D Control Register 1 (Continued)

bit 1 **SAMP:** A/D Sample Enable bit
1 = A/D Sample-and-Hold amplifiers are sampling
0 = A/D Sample-and-Hold amplifiers amplifiers are holding

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion cycle is completed
0 = A/D conversion not started, or in progress

Note 1: This bit is implemented only in modules with user-selectable resolution. For A/D modules with fixed 10-bit or 12-bit resolution, this bit is unimplemented and should be maintained as '1' or '0'. Refer to the device data sheet for the specific setting of this bit.

2: Sample clock trigger sources are defined differently for each device family; not all trigger sources may be defined. Refer to the specific data sheet for details.

Register 51-2: AD1CON2: A/D Control Register 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—
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
BUFS(1)	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM(1)	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 15-14 **PVCFG<1:0>:** Converter Positive Voltage Reference Configuration bits
11 = Internal VRH2
10 = Internal VRH1
01 = External VREF+
00 = AVDD

bit 13 **NVCFG0:** Converter Negative Voltage Reference Configuration bits
1 = External VREF-
0 = AVSS

bit 12 **OFFCAL:** Offset Calibration Mode Select bit
1 = Inverting and non-inverting inputs of channel Sample-and-Hold are connected to AVSS
0 = Inverting and non-inverting inputs of channel Sample-and-Hold are connected to normal inputs

bit 11 **BUFREGEN:** A/D Buffer Register Enable bit
1 = Conversion result is loaded into buffer location determined by the converted channel
0 = A/D result buffer is treated as a FIFO

bit 10 **CSCNA:** MUX A Channel Scan Enable bit
1 = Scan inputs selected by the AD1CSSH/AD1CSSL registers as the MUX A input(s)
0 = Do not scan inputs

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

bit 7 **BUFS:** Buffer Fill Status bit(1)
1 = A/D is filling the upper half of the buffer; user should access data in the lower half
0 = A/D is filling the lower half of the buffer; user should access data in the upper half

bit 6-2 **SMPI<4:0>:** Interrupt Sample Rate Select bits
11111 = Interrupts at the completion of conversion for each 32nd sample
11110 = Interrupts at the completion of conversion for each 31st sample
• • •
00001 = Interrupts at the completion of conversion for every other sample
00000 = Interrupts at the completion of conversion for each sample

bit 1 **BUFM:** Buffer Fill Mode Select bit(1)
1 = Starts buffer filling at AD1BUF0 on first interrupt and AD1BUF(n/2) on next interrupt (Split Buffer mode)
0 = Starts filling buffer at address, ADCBUF0, and each sequential address on successive interrupts (FIFO mode)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
1 = Uses channel input selects for SAMPLE A on first sample and SAMPLE B on the next sample
0 = Always uses channel input selects for SAMPLE A

Note 1: Applicable only when the buffer is used in FIFO mode (BUFREGEN = 0). In addition, BUFS is only used when BUFM = 1.

Register 51-3: AD1CON3: A/D Control Register 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN(1)	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 = RC clock
0 = Clock derived from system clock

bit 14 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 13 **PUMPEN:** Charge Pump Enable bit(1)
1 = Charge pump for switches is enabled
0 = Charge pump for switches is disabled

bit 12-8 **SAMC<4:0>:** Auto-Sample Time Select bits
11111 = 31 TAD
• • •
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111
• • • = Reserved
01000000
00111111 = 64 · TCY = TAD
• • •
00000001 = 2 · TCY = TAD
00000000 = TCY = TAD

Note 1: Available on select devices only.

Register 51-4: AD1CON5: A/D Control Register 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ASEN	LPEN	CTMREQ	BGREQ(1)	VRSREQ(1)	—	ASINT1	ASINT0
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	CM1	CM0
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 **ASEN:** Auto-Scan Enable bit
1 = Auto-Scan is enabled
0 = Auto-Scan is disabled

bit 14 **LPEN:** Low-Power Enable bit
1 = Low power enabled after scan
0 = Full power enabled after scan

bit 13 **CTMREQ:** CTMU Request bit
1 = CTMU is enabled when the ADC is enabled and active
0 = CTMU is not enabled by the ADC

bit 12 **BGREQ:** Band Gap Request bit(1)
1 = Band gap is enabled when the ADC is enabled and active
0 = Band gap is not enabled by the ADC

bit 11 **VRSREQ:** VREG Scan Request bit(1)
1 = On-chip regulator is enabled when the ADC is enabled and active
0 = On-chip regulator is not enabled by the ADC

bit 10 **Unimplemented:** Read as '0'

bit 9-8 **ASINT<1:0>:** Auto-Scan (Threshold Detect) Interrupt Mode bits
11 = Interrupt after Threshold Detect sequence is completed and valid compare has occurred
10 = Interrupt after valid compare has occurred
01 = Interrupt after Threshold Detect sequence is completed
00 = No interrupt

bit 7-4 **Unimplemented:** Read as '0'

bit 3-2 **WM<1:0>:** Write Mode bits
11 = Reserved
10 = Auto-compare only (conversion results not saved, but interrupts are generated when a valid match as defined by CM and ASINT bits occurs)
01 = Convert and save (conversion results saved to locations as determined by register bits when a match as defined by CM bits occurs)
00 = Legacy operation (conversion data saved to location determined by buffer register bits)

bit 1-0 **CM<1:0>:** Compare Mode bits
11 = Outside Window mode (valid match occurs if the conversion result is outside of the window defined by the corresponding buffer pair)
10 = Inside Window mode (valid match occurs if the conversion result is inside the window defined by the corresponding buffer pair)
01 = Greater Than mode (valid match occurs if the result is greater than value in the corresponding buffer register)
00 = Less Than mode (valid match occurs if the result is less than value in the corresponding buffer register)

Note 1: These bits are not implemented in all devices.

Register 51-5: AD1CHS: A/D Sample Select 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
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
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
CH0NA2	CH0NA1	CH0NA0	CH0SA4	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 15-13 **CH0NB<2:0>:** Sample B Channel 0 Negative Input Select bits

The number of implemented inputs and the bit combinations assigned to them vary between device families. In general, device analog ground and/or the negative external voltage reference (V_{REF-}) is assigned to '000'. Inputs from other external analog channels or internal references may be assigned to the other combinations. Refer to the specific device data sheet for a complete listing of implemented inputs for a particular device.

Any bit combinations not explicitly listed in the device data sheet are unimplemented. Using an unimplemented channel for a conversion will produce unpredictable results.

bit 12-8 **CH0SB<4:0>:** S/H Amplifier Positive Input Select for MUX B Multiplexer Setting bits

The number of implemented inputs and the bit combinations assigned to them vary significantly between device families. In general, external analog inputs are sequentially assigned from '00000', starting with AN0, potentially up to AN31 ('11111').
If 32 external inputs are not implemented, inputs for internal band gap references, external voltage references, and other analog modules, such as the CTMU, are implemented following the external analog channels. Refer to the specific device data sheet for a complete listing of implemented inputs for a particular device.

If a sequential input is unimplemented, its corresponding bit value is also unimplemented. Also, any bit combinations not explicitly listed in the device data sheet are unimplemented. Using an unimplemented channel for a conversion will produce unpredictable results.

bit 7-5 **CH0NA<2:0>:** Sample A Channel 0 Negative Input Select bits

Same definitions as for CHONB<2:0>.

bit 4-0 **CH0SA<4:0>:** Sample A Channel 0 Positive Input Select bits

Same definitions as for CHOSB<4:0>.

Register 51-6: AD1CHITH: A/D Scan Compare Hit Register (High Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH31	CHH30	CHH29	CHH28	CHH27	CHH26	CHH25	CHH24
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
CHH23	CHH22	CHH21	CHH20	CHH19	CHH18	CHH17	CHH16
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 **CHH<31:16>:** A/D Compare Hit bits(1)

If CM<1:0> = 11:

1 = A/D Result Buffer n has been written with data or a match has occurred

0 = A/D Result Buffer n has not been written with data

For all other values of CM<1:0>:

1 = A match has occurred on A/D Result Channel n

0 = No match has occurred on A/D Result Channel n

Note 1: The actual number of channels available depends on which channels are implemented on a specific device. Refer to the device data sheet for details. Unimplemented channels are read as '0'.

Register 51-7: AD1CHITL: A/D Scan Compare Hit Register (Low Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH15(1)	CHH14(1)	CHH13(1)	CHH12(1)	CHH11(1)	CHH10(1)	CHH9(1)	CHH8(1)
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
CHH7(1)	CHH6(1)	CHH5(1)	CHH4(1)	CHH3(1)	CHH2(1)	CHH1(1)	CHH0(1)
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 **CHH<15:0>:** A/D Compare Hit bits(1)

If CM<1:0> = 11:

1 = A/D Result Buffer n has been written with data or a match has occurred

0 = A/D Result Buffer n has not been written with data

For all other values of CM<1:0>:

1 = A match has occurred on A/D Result Channel n

0 = No match has occurred on A/D Result Channel n

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as '0'.

Register 51-8: AD1CSSH: A/D Input Scan Select Register (High Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS31(1)	CSS30(1)	CSS29(1)	CSS28(1)	CSS27(1)	CSS26(1)	CSS25(1)	CSS24(1)
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
CSS23(1)	CSS22(1)	CSS21(1)	CSS20(1)	CSS19(1)	CSS18(1)	CSS17(1)	CSS16(1)
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 **CSS<31:16>:** A/D Input Scan Selection bits(1)
1 = Include corresponding channel for input scan
0 = Skip channel for input scan

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as '0'. Do not select unimplemented channels for sampling, as indeterminate results may be produced.

Register 51-9: AD1CSSL: A/D Input Scan Select Register (Low Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS15(1)	CSS14(1)	CSS13(1)	CSS12(1)	CSS11(1)	CSS10(1)	CSS9(1)	CSS8(1)
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
CSS7(1)	CSS6(1)	CSS5(1)	CSS4(1)	CSS3(1)	CSS2(1)	CSS1(1)	CSS0(1)
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 **CSS<15:0>:** A/D Input Scan Selection bits(1)
1 = Include corresponding channel for input scan
0 = Skip channel for input scan

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as '0'. Do not select unimplemented channels for sampling, as indeterminate results may be produced.

Register 51-10: AD1CTMENH: CTMU Enable Register (High Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN31(1)	CTMEN30(1)	CTMEN29(1)	CTMEN28(1)	CTMEN27(1)	CTMEN26(1)	CTMEN25(1)	CTMEN24(1)
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
CTMEN23(1)	CTMEN22(1)	CTMEN21(1)	CTMEN20(1)	CTMEN19(1)	CTMEN18(1)	CTMEN17(1)	CTMEN16(1)
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 **CTMUEN<31:16>:** CTMU Enabled During Conversion bits(1)

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as '0'.

Register 51-11: AD1CTMENL: CTMU Enable Register (Low Word)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN15(1)	CTMEN14(1)	CTMEN13(1)	CTMEN12(1)	CTMUEN11(1)	CTMEN10(1)	CTMEN9(1)	CTMEN8(1)
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
CTMEN7(1)	CTMEN6(1)	CTMEN5(1)	CTMEN4(1)	CTMEN3(1)	CTMEN2(1)	CTMEN1(1)	CTMEN0(1)
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 **CTMUEN<15:0>:** CTMU Enabled During Conversion bits(1)

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as '0'.

51.4 A/D MODULE CONFIGURATION

All of the registers described in the previous section must be configured for module operation to be fully defined. An effective approach is first, to describe the signals and sequences for the particular application. Typically, it is an iterative process to assign signals to port pins, to establish timing methods and to organize a scanning scheme, as well as to integrate the whole process with the software design.

The various configuration and control functions of the module are distributed throughout the module's six control registers. Control functions can be broadly sorted into four groups: input, timing, conversion and output. Table 51-1 shows the register location of control or status bits by register.

Table 51-1: A/D Module Functions by Registers and Bits

A/D Function	Register(s)	Specific Bits
Input	AD1CON2	PVCFG<1:0>, NVCFG, OFFCAL, CSCNA, ALTS
	AD1CON5	CTMREQ, BGREQ, VRSREQ
	AD1CHS	CH0NB<7:5>, CH0SB<4:0>, CH0NA<7:5>, CH0SA<4:0>
	AD1CSSH/L	CSSL<31:16>, CSSL<15:0>(1)
	AD1CTMENH/L	CTMEN<31:16>, CTMEN<15:0>(1)
Conversion	AD1CON1	ADON, ADSIDL, SSRC<4:0>, ASAM, SAMP, DONE
	AD1CON2	SMPI<3:0>
	AD1CON3	EXTSAM
	AD1CON5	ASEN, LPEN, ASINT<1:0>
Timing	AD1CON3	ADRC, SAMC<4:0>, ADCS<7:0>
Output	AD1CON1	FORM<1:0>
	AD1CON2	BUFS, BUFM, BUFREGEN
	AD1CON5	WM<1:0>, CM<1:0>

Note 1: The number and location of implemented bits is device-specific. Refer to the device data sheet for more information.

Note: Do not write to the SSRC, ASAM, BUFS SMPI, BUFM and ALTS bits, or the AD1CON3 and AD1CSSL registers, while ADON = 1; otherwise, indeterminate conversion data may result.

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

1. Configure the A/D module:
   - Select the output resolution (if configurable)
   - Select the voltage reference source to match the expected range on analog inputs
   - Select the analog conversion clock to match the desired data rate with a processor clock
   - Determine how sampling will occur
   - Set the multiplexer input assignments
   - Select the desired sample/conversion sequence
   - Select the output data format
   - Select the number of readings per interrupt
2. Configure A/D interrupt (if required):
   - Clear AD1IF bit
   - Select A/D interrupt priority
3. Turn on A/D module.

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

51.4.1 Selecting the Resolution

Most versions of the A/D module will have a fixed conversion resolution of either 12 bits (the majority) or 10 bits. Where configuration is permitted, the MODE12 bit (AD1CON1<10>) controls output resolution. Setting this bit selects 12-bit resolution.

51.4.2 Selecting the Voltage Reference Source

The voltage references for A/D conversions are selected using the PVCFG<1:0> and NVCFG control bits (AD1CON2<15:13>). The upper voltage reference (VR+) may be AVDD, the external VREF+ or an internal band gap reference voltage. The lower voltage reference (VR-) may be AVSS or the VREF- input pin. The available options vary between device families.

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 51.17 "Electrical Specifications"** for further details.

51.4.3 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 14 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 a 6-bit counter. There are 64 possible options for TAD, specified by the ADCS bits in the AD1CON3 register. Equation 51-1 gives the TAD value as a function of the ADCS control bits and the device instruction cycle clock period, TCY. For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 75 ns.

Equation 51-1: A/D Conversion Clock Period

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

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

Note: Based on TCY = 2/FOSC; 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.

51.4.4 Configuring Analog Port Pins

The A/D module does not have an internal provision to configure port pins for analog operation. Instead, input pins are configured as analog inputs through the Analog Select registers (ANSn, where 'n' is the port name). A pin is configured as an analog input when the corresponding ANSn bit is set. By default, pins with multiplexed analog and digital functions are configured as analog pins on device Reset.

For external analog inputs, both the ANSn 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, the port digital output level (VOH or VOL) will be converted.

Note 1: When reading a PORT register, any pin configured as an analog input reads as '0'.

2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume current that is out of the device's specification.

51.4.5 Input Channel Selection

The A/D Converter incorporates two independent sets of input multiplexers (MUX A and MUX B) that allow users to choose which analog channels are to be sampled. The inputs specified by the CH0SA bits and CH0NA are collectively called the MUX A inputs. The inputs specified by the CH0SB bits 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 of a negative reference source for differential signals. In addition, MUX A can be configured for sequential analog channel scanning. This is discussed in more detail in **Section 51.4.5.1 "Configuring MUX A and MUX B Inputs"** and **Section 51.4.5.3 "Scanning Through Several Inputs"**.

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

51.4.5.1 CONFIGURING MUX A AND MUX B INPUTS

The user may select any one of up to 32 inputs available to the A/D Converter as the positive input of the S/H amplifier. For MUX A, the CH0SA<4:0> bits (AD1CHS<4:0>) normally select the analog channel for the positive input. For MUX B, the positive channel is selected by the CH0SB<4:0> bits (AD1CHS<12:8>).

All of the external analog channels are available as positive inputs. In addition to the external inputs, these may also include device supply voltage (AV_{DD}), the logic core supply voltage (V_{DDCORE}), the internal band gap voltage (V_{BG}) and/or multiples or fractions of V_{BG}. One or more additional input channels are used for the CTMU. These selections leave the A/D disconnected from all other inputs. The options vary by device family; refer to the specific device data sheet for details.

The CTMU input is selected by the AD1CTMENH/L registers. Setting a particular bit in one of these registers effectively assigns the analog output from the CTMU to the corresponding A/D input channel, automatically enabling the CTMU. Many devices will already have a CHOSA bit combination designated for use of the CTMU. This setting disconnects the converter from any other load. This channel should be the one selected by the appropriate AD1CTMEN bit. If another channel is selected, verify that any other analog sources are disconnected from that channel; otherwise, erroneous readings may result.

For the negative (inverting) input of the amplifier, the user has up to eight options, selected by the CH0NA<2:0> and CH0NB<2:0> bits (AD1CHS<7:5> and AD1CHS<15:13>, respectively). Options typically include the device ground (AVss), the current V_{R-} source designated by the NVCFG bit (AD1CON2<13>), and one or more of the external analog input channels. As with the non-inverting inputs, the options vary by device family.

51.4.5.2 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, the inputs specified by the CH0SA bits and CH0NA are selected for sampling. On the next sample/convert sequence, the inputs specified by the CH0SB bits and CH0NB bits are selected for sampling. This pattern repeats for subsequent sample conversion sequences.

51.4.5.3 SCANNING THROUGH SEVERAL INPUTS

When using MUX A to select analog inputs, the A/D module has the ability to scan multiple analog channels. When the CSCNA bit (AD1CON2<10>) is set, the CH0SA bits are ignored and the channels specified by the AD1CSSL register are sequentially sampled.

Each bit in the AD1CSSL register and AD1CSSH register (when implemented) corresponds to one of the analog channels. If a bit in the AD1CSSL or AD1CSSH 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 1:	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.
2:	If the CTMU channel is to be included in a scan operation, verify that the proper analog input channel is selected and that the AD1CTMEN register(s) are correctly configured. See **Section 51.4.5.1 "Configuring MUX A and MUX B Inputs"** for more information.

The AD1CSSH/L registers bits specify the positive input of the channel. The CH0NA bits still select 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 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 AD1CSSH/L registers, and a fixed input specified by the CH0SB bits.

Automatic scanning can be used in conjunction with the Threshold Detect feature to determine if one or more analog channels meet a predetermined set of conditions while the CPU is inactive. This is described in detail in **Section 51.9 "Threshold Detect Operation"**.

51.4.5.4 INTERNAL CHANNELS IN LOW-POWER MODES

While the A/D module can scan and convert analog inputs in low-power modes, some internal analog inputs may be unavailable in Sleep mode. The main examples are the CTMU module, the internal band gap voltage source and the on-chip voltage regulator (for those devices that include one). The A/D module provides a method to make these resources available automatically through the CTMREQ, BGREQ and VRSREQ bits (AD1CON5<13:11>, respectively). Setting one or more of these bits causes the corresponding internal analog source(s) to become active during a channel scan.

The CTMREQ bit is available on all devices. The BGREQ bit is available on those devices that make the internal band gap reference available to the A/D module as an input channel and/or as V_{REF+}.The VRSREQ bit is only available in device families that use an on-chip voltage regulator.

51.4.6 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. Although the digital and analog portions of the circuit are turned off for maximum current savings, the contents of all registers are maintained.

Conversion data stored in the ADC1BUF registers will also be maintained, including any threshold values stored by the user. It may be necessary to re-initialize these registers to their proper values before re-enabling the module.

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 51.17 "Electrical Specifications"**.

51.5 INITIALIZATION

Example 51-1 shows a simple initialization code example for the A/D module. 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 directly 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 TCY/2.

In this particular configuration, all 16 analog input pins are set up as analog inputs. It is important to note that with this A/D module, I/O pins are configured for analog or digital operation at the I/O port with the ANSn Analog Select registers. The use of these registers is described in detail in the I/O Port chapter of the specific device data sheet.

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 51.6 "Controlling the Sampling Process"** and **Section 51.7 "Controlling the Conversion Process"**.

Example 51-1: A/D Initialization Code Example

```
AD1CON1 = 0x2200;      // Configure sample clock source
                       // and conversion trigger mode.
                       // Unsigned Fraction format (FORM<1:0>=10),
                       // Manual conversion trigger (SSRC<3:0>=0000),
                       // Manual start of sampling (ASAM=0),
                       // No operation in Idle mode (ADSIDL=1),
                       // S/H in Sample (SAMP = 1)
AD1CON2 = 0;           // Configure A/D voltage reference
                       // and buffer fill modes.
                       // Vr+ and Vr- from AVdd and AVss(PVCFG<1:0>=00, NVCFG=0),
                       // Inputs are not scanned,
                       // Interrupt after every sample
AD1CON3 = 0;           // Configure sample time = 1Tad,
                       // A/D conversion clock as Tcy
AD1CHS  = 0;           // Configure input channels,
                       // S/H+ input is AN0,
                       // S/H- 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;
}
```

51.6 CONTROLLING THE SAMPLING PROCESS

51.6.1 Manual Sampling

Setting the SAMP bit (AD1CON1<1>) while the ASAM bit (AD1CON1<2>) is clear causes the A/D to begin sampling. Clearing the SAMP bit ends sampling and automatically begins the conversion; however, there must be a sufficient delay between setting and clearing SAMP for the sampling process to start (tPSS, Parameter AD61). Sampling will not resume until the SAMP bit is once again set. For an example, see Figure 51-4**.**

51.6.2 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 as an event to end sampling and complete the conversions. Sampling will continue on the next selected channel after the conversion in progress has completed. For an example, see Figure 51-5**.**

51.6.3 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.

51.6.4 Aborting a Sample

While in Manual Sampling mode, clearing the SAMP bit will terminate sampling. If SSRC<3:0> = 0000, 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.

51.7 CONTROLLING THE CONVERSION PROCESS

The conversion trigger source will terminate sampling and start a selected sequence of conversions. The SSRC<3:0> bits (AD1CON1<7:4>) 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, disable the A/D module first by clearing the ADON bit (AD1CON1<15>).

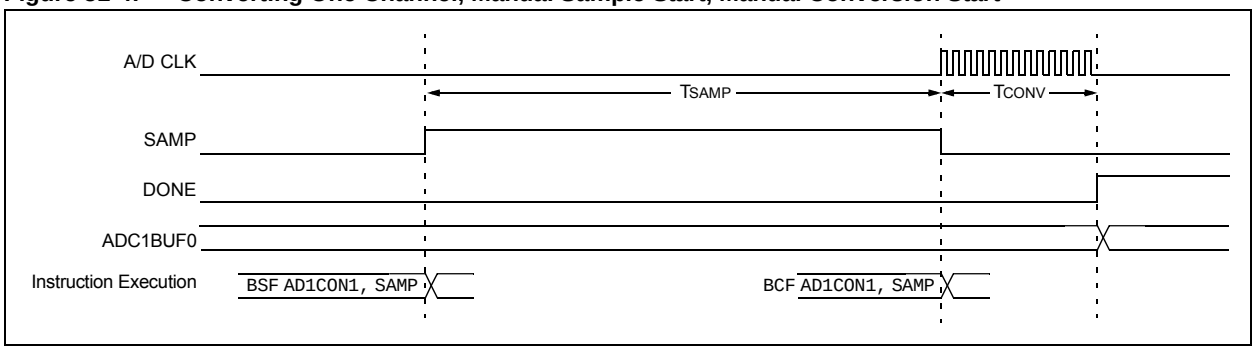
51.7.1 Manual Control

When SSRC<3:0> = 0000, the conversion trigger is under software control. Clearing the SAMP bit (AD1CON1<1>) starts the conversion sequence.

Figure 51-4 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 51-5 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 sets the SAMP bit and returns to the sample state. The user software must time the clearing of the SAMP bit to ensure adequate sampling time of the input signal, understanding that the time since previously clearing the SAMP bit includes the conversion time, which immediately follows, as well as the next sampling time.

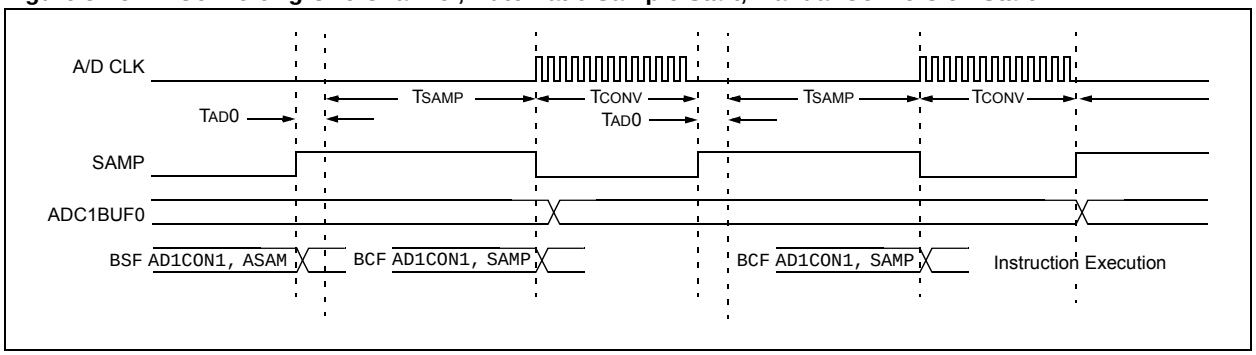
Figure 51-4: Converting One Channel, Manual Sample Start, Manual Conversion Start



Example 51-2: Converting One Channel, Manual Sample Start, Manual Conversion Start Code

```
int ADCValue;

ANSB    = 0x0001;                // AN2 as analog, all other pins are digital
AD1CON1 = 0x0000;                // SAMP bit = 0 ends sampling and starts converting
AD1CHS  = 0x0002;                // Connect AN2 as S/H+ input
                                 // in this example AN2 is the input
AD1CSSL = 0;
AD1CON3 = 0x0002;                // Manual Sample, Tad = 3Tcy
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 51-5: Converting One Channel, Automatic Sample Start, Manual Conversion Start



51.7.2 Clocked Conversion Trigger

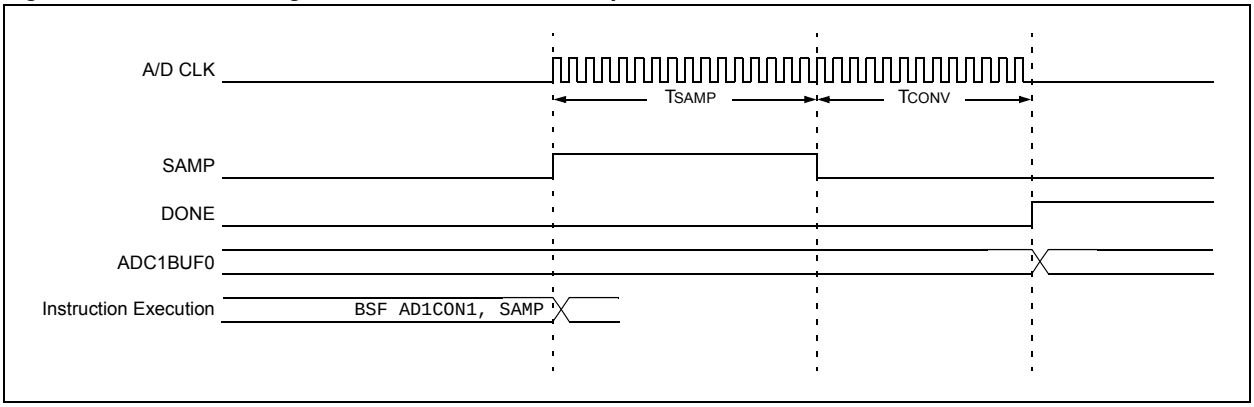
When ADRC = 1, the conversion trigger is under A/D clock control. The SAMC bits (AD1CON3<12:8>) select the number of TAD clock cycles between the start of sampling and the start of conversion. After the start of sampling, the module will count a number of TAD clocks specified by the SAMC bits. The SAMC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

Equation 51-2: Clocked Conversion Trigger Time

$$TSMP = SAMC<4:0> * TAD$$

Figure 51-6 shows how to use the clocked conversion trigger with the sampling started by the user software.

Figure 51-6: Converting One Channel, Manual Sample Start, TAD-Based Conversion Start



Example 51-3: Converting One Channel, Manual Sample Start, TAD-Based Conversion Start Code

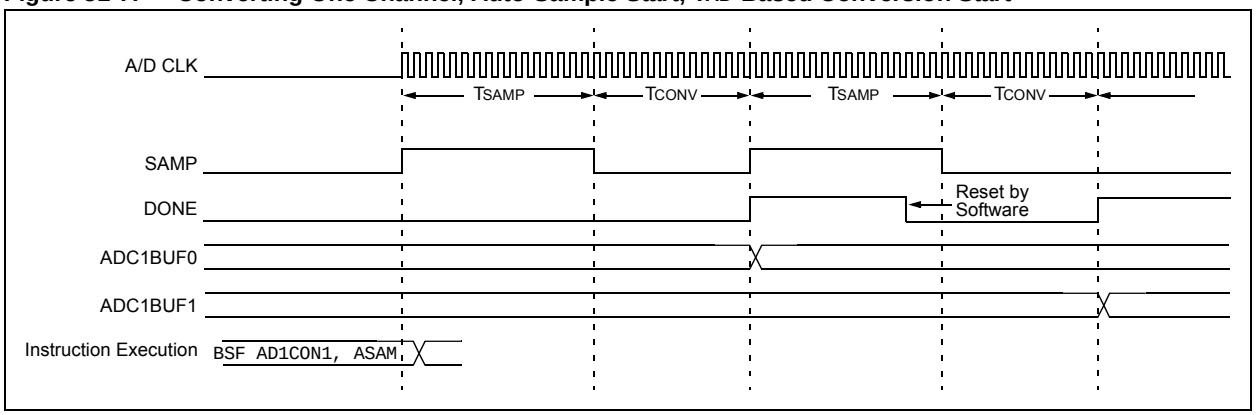
```
int ADCValue;

ANSB    = 0x1000;                    // all PORTB = Digital; RB12 = analog
AD1CON1 = 0x00E0;                    // SSRC<2:0> = 111 implies internal counter ends sampling
                                     // and starts converting.
AD1CHS  = 0x000C;                    // Connect AN12 as S/H input.
                                     // in this example AN12 is the input
AD1CSSL = 0;
AD1CON3 = 0x1F02;                    // Sample time = 31Tad, Tad = 3Tcy
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
```

51.7.2.1 FREE-RUNNING SAMPLE CONVERSION SEQUENCE

Using the Auto-Convert Conversion Trigger mode (SSRC<3:0> = 0111), 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 51-7, allows continuous data collection after module initialization.

Note that all timing in this mode scales with TAD, either from the A/D internal RC clock or from TCY (as prescaled by the ADCS<7:0> bits). In both cases, the SAMC<4:0> bits set the number of TAD clocks in TSAMP. TCONV is fixed at 12 TAD.

Figure 51-7: Converting One Channel, Auto-Sample Start, TAD-Based Conversion Start



Example 51-4: Converting One Channel, Auto-Sample Start, TAD-Based Conversion Start Code

```
int ADCValue, count;
int *ADC16Ptr;

ANSB    = 0x0001;                          // 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
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
```

51.7.2.2 SAMPLE TIME CONSIDERATIONS USING CLOCKED CONVERSION TRIGGER AND AUTOMATIC SAMPLING

The user must ensure the sampling time satisfies the sampling requirements as outlined in **Section 51.11 "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.

51.7.3 Event Trigger Conversion Start

It is often desirable to synchronize the end of sampling and the start of conversion with some other time event. Depending on the device family, the A/D module has up to 16 sources available to use as a conversion trigger event. The event trigger is selected by the SSRC<3:0> bits (AD1CON1<7:4>).

As noted, the available event triggers vary between device families. Refer to the specific device data sheet for specific information. The examples that follow are represent trigger sources that are implemented in most devices. Note that the SSRC bit assignments may vary in some devices.

51.7.3.1 EXTERNAL INT0 PIN TRIGGER

When SSRC<3:0> = 0001, the A/D conversion is triggered by an active transition on the INT0 pin. The pin may be programmed for either a rising edge input or a falling edge input.

51.7.3.2 GENERAL PURPOSE TIMER COMPARE TRIGGER

When SSRC<3:0> = 0010, the A/D is triggered by a match between the 32-bit timer pair, TMR3/TMR2, and the 32-bit combined Period register, PR3/PR2. The match causes the timer to generate a special ADC trigger event signal.

In some devices, this feature is also implemented for the TMR5/TMR4 timer pair. Refer to the specific device data sheet for details.

51.7.3.3 SYNCHRONIZING A/D OPERATIONS TO INTERNAL OR EXTERNAL EVENTS

The modes where an external event trigger pulse ends sampling and starts conversion 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 51-9 where SSRC<3:0> = 0010 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.

51.7.3.4 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 satisfies the sampling requirements, as outlined in **Section 51.11 "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 51-3: Calculating Available Sampling Time for Sequential Sampling

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

Figure 51-8: Manual Sample Start, Conversion Trigger-Based Conversion Start

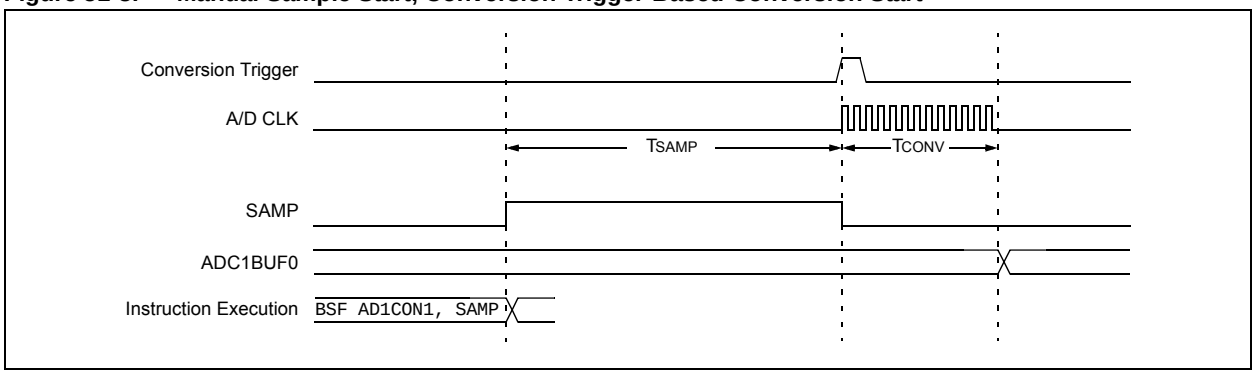
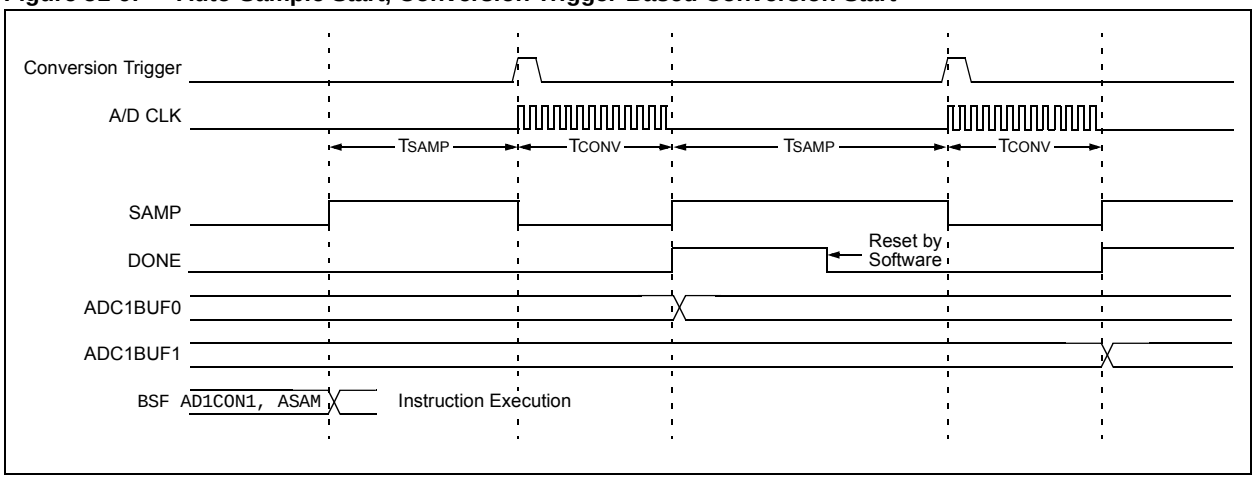


Figure 51-9: Auto-Sample Start, Conversion Trigger-Based Conversion Start



Example 51-5: Converting One Channel, Auto-Sample Start, Conversion Trigger-Based Conversion Start Code

```
int ADCValue;

ANSB    = 0x0001;                // AN2 as analog, all other pins are 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
AD1CON2 = 0x0004;                // Set AD1IF after 2 conversions
TMR3 = 0x0000;                   // set TMR3 to time out every 125 ms
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
}
```

51.7.4 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.

51.7.5 Generating A/D Interrupts

The SMPI<4:0> bits (AD1CON2<6:2>) 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 SMPI bits, after the start of sampling, and continues to recur after that number of samples. The value specified by the SMPI 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, AD1IE.

51.7.6 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).

51.7.7 Offset Calibration

The module provides a simple calibration method to offset the effects of internal device noise. While not always necessary, this may be helpful in situations where weak analog signals are being converted. Calibration is performed by using the OFFCAL bit (AD1CON2<12>). This disconnects the S/H amplifier entirely from any inputs. With the OFFCAL bit set, a single reference conversion is performed. The results of this conversion are value added by internal device noise. This result can be stored by the application, then used as an offset value for future conversions.

51.8 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 fixed word size, accessed through the SFR space. The size of the buffer is determined by the number of external analog input channels on the device, allowing one word for each channel. Depending on the device, additional buffer space may be provided for one or more internal analog channels (e.g., band gap sources). The number of buffer addresses is always even, and always at least equal to the number of external channels.

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 minimize software overhead, the module will fill the buffer with results and then generate an interrupt when the buffer is filled.

Note:	This section describes buffer operation in Legacy mode (AD1CON5<3:2> = 00). Buffer operation is different when the Compare Only or Compare and Save modes are used with the Threshold Detect feature. See **Section 51.9 "Threshold Detect Operation"** for more information.

51.8.1 Number of Conversions per Interrupt

The SMPI<3:0> bits 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 SMPI<3:0> = 0000, the conversion results will always be written to the ADC1BUF0. In this example, no other buffer locations would be used, since only one sequence per interrupt is specified.

51.8.2 Buffer Fill Modes

The results buffer can be configured to operate in either of two modes: a standard FIFO mode, compatible with the earlier 10-bit A/D module (default), or a Channel Indexed mode. The Fill mode is selected by the BUFREGEN bit (AD1CON2<11>).

51.8.2.1 FIFO MODES

When BUFREGEN = 0, the results buffer operates in FIFO mode. The first conversion results, after initiating conversions, is written to the first available buffer address. Subsequent conversions are written to the next sequential buffer location, continuing until the process is interrupted. If allowed to continue without interrupts, the module would fill each location and then wrap around to the first address, continuing the process.

The BUFM bit (AD1CON2<1>) controls how the buffer is filled. When BUFM is '1', the buffer is split into two equal halves: a lower half (ADC1BUF0 through ADC1BUF[(n/2)-1]) and an upper half (ADC1BUF[n/2] through ADC1BUFn), where n is the number of available analog channels (both internal and external). The buffers will alternately receive the conversion results after each interrupt event. The initial buffer used after BUFM is set is the lower group.

When BUFM is '0', the entire buffer is used for all conversion sequences.

Note:	When the BUFM bit is set, the user should not program the SMPI bits to a value that specifies more than (n/2) 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 SMPI<3:0> = 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.

51.8.2.1.1 Buffer Fill Status

When the conversion result buffer is split (BUFM = 1), the BUFS status bit (AD1CON2<7>) indicates which 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 application should read conversion values from the upper group. If BUFS = 1, the situation is reversed, and the user application should read conversion values from the lower group.

51.8.2.2 CHANNEL INDEXED MODE

When BUFREGEN = 1, FIFO operation is disabled. In this Fill mode, the conversion result for each channel is written only to the buffer location that corresponds to that channel. For example, any conversions performed on AN0 are stored only in ADC1BUF0. The same holds true for AN1 and ADC1BUF1, and so on. Subsequent conversions on a particular channel that occur prior to an interrupt will result in any previous data in that location being overwritten.

Channel Indexed mode is particularly useful when used with the Threshold Detect feature, as this allows the user to easily test for a particular condition on a specific analog channel without creating an excess of CPU overhead. This is covered in more detail in **Section 51.9 "Threshold Detect Operation"**.

51.8.3 Buffer Data Formats

The results of each A/D conversion are 12 bits wide (optionally, 10 bits wide in some devices). To maintain data format compatibility, the result of each conversion is automatically converted to one of four selectable, 16-bit formats. The FORM<1:0> bits (AD1CON1<9:8>) select the format. Figure 51-10 and Figure 51-11 show the data output formats that can be selected. Tables 51-2 through 51-5 show the numerical equivalents for the various conversion result codes.

Figure 51-10: A/D Output Data Formats (12-Bit)

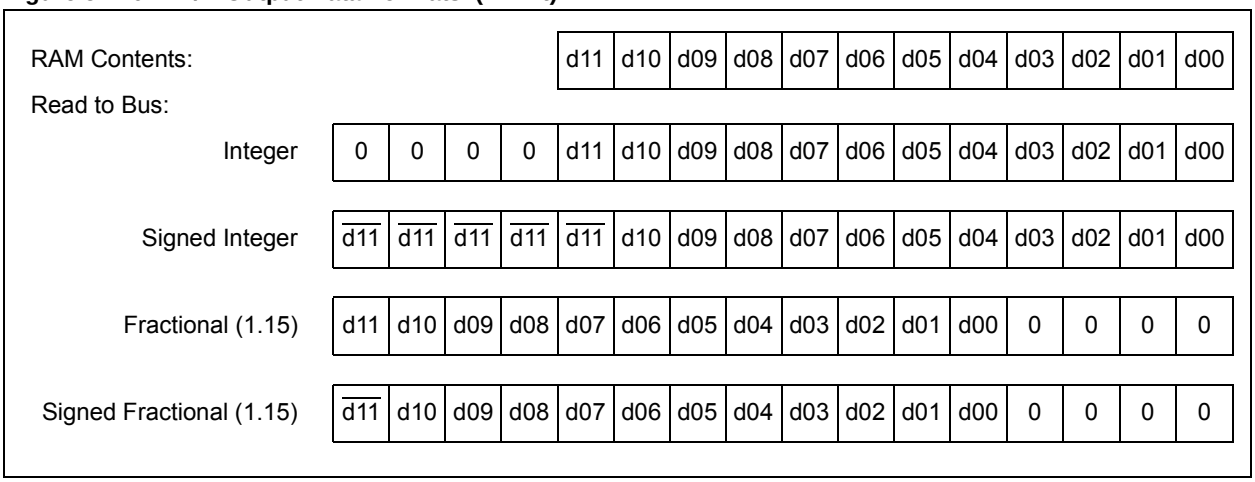


Table 51-2: Numerical Equivalents of Various Result Codes: 12-Bit Integer Formats

VIN/VREF	12-Bit Output Code	16-Bit Integer Format/ Equivalent Decimal Value		16-Bit Signed Integer Format/ Equivalent Decimal Value	
4095/4096	1111 1111 1111	0000 1111 1111 1111	4095	0000 0111 1111 1111	2047
4094/4096	1111 1111 1110	0000 1111 1111 1110	4094	0000 0111 1111 1110	2046
•••					
2049/4096	1000 0000 0001	0000 1000 0000 0001	2049	0000 0000 0000 0001	1
2048/4096	1000 0000 0000	0000 1000 0000 0000	2048	0000 0000 0000 0000	0
2047/4096	0111 1111 1111	0000 0111 1111 1111	2047	1111 1111 1111 1111	-1
•••					
1/4096	0000 0000 0001	0000 0000 0000 0001	1	1111 1000 0000 0001	-2047
0/4096	0000 0000 0000	0000 0000 0000 0000	0	1111 1000 0000 0000	-2048

Table 51-3: Numerical Equivalents of Various Result Codes: 12-Bit Fractional Formats

VIN/VREF	12-Bit Output Code	16-Bit Fractional Format/ Equivalent Decimal Value		16-Bit Signed Fractional Format/ Equivalent Decimal Value	
4095/4096	1111 1111 1111	1111 1111 1111 0000	0.999	0111 1111 1111 0000	0.499
4094/4096	1111 1111 1110	1111 1111 1110 0000	0.998	0111 1111 1110 0000	0.498
•••					
2049/4096	1000 0000 0001	1000 0000 0001 0000	0.501	0000 0000 0001 0000	0.001
2048/4096	1000 0000 0000	1000 0000 0000 0000	0.500	0000 0000 0000 0000	0.000
2047/4096	0111 1111 1111	0111 1111 1111 0000	0.499	1111 1111 1111 0000	-0.001
•••					
1/4096	0000 0000 0001	0000 0000 0001 0000	0.001	1000 0000 0001 0000	-0.499
0/4096	0000 0000 0000	0000 0000 0000 0000	0.000	1000 0000 0000 0000	-0.500

Figure 51-11: A/D Output Data Formats (10-Bit)

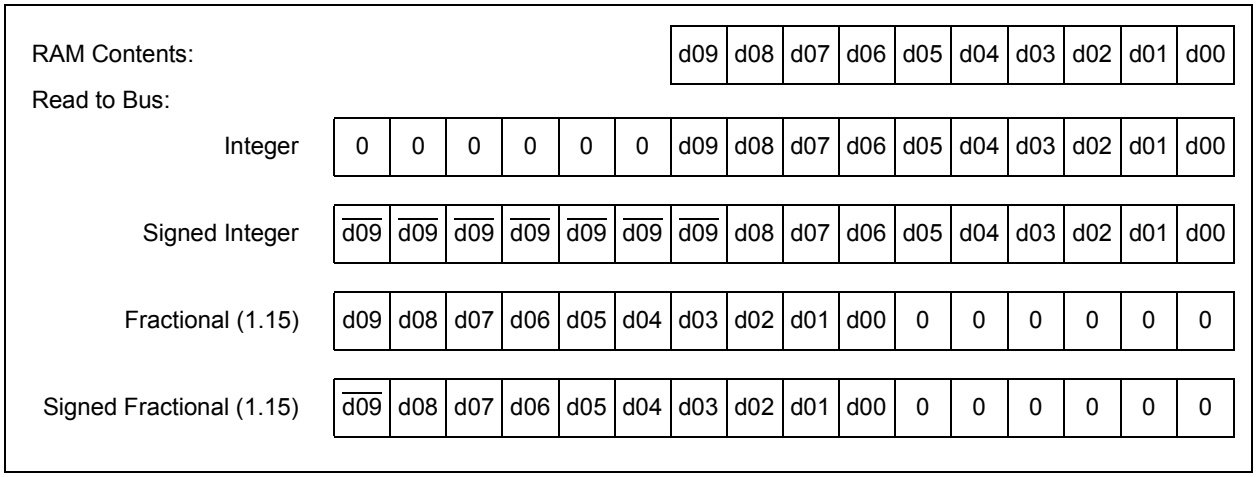


Table 51-4: Numerical Equivalents of Various Result Codes: 10-Bit Integer Formats

VIN/VREF	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	0000 0001 1111 1111	511
1022/1024	11 1111 1110	0000 0011 1111 1110	1022	0000 0001 1111 1110	510
		•••			
513/1024	10 0000 0001	0000 0010 0000 0001	513	0000 0000 0000 0001	1
512/1024	10 0000 0000	0000 0010 0000 0000	512	0000 0000 0000 0000	0
511/1024	01 1111 1111	0000 0001 1111 1111	511	1111 1111 1111 1111	-1
		•••			
1/1024	00 0000 0001	0000 0000 0000 0001	1	1111 1110 0000 0001	-511
0/1024	00 0000 0000	0000 0000 0000 0000	0	1111 1110 0000 0000	-512

Table 51-5: Numerical Equivalents of Various Result Codes: 10-Bit Fractional Formats

VIN/VREF	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	0111 1111 1100 0000	0.499
1022/1024	11 1111 1110	1111 1111 1000 0000	0.998	0111 1111 1000 0000	0.498
		•••			
513/1024	10 0000 0001	1000 0000 0100 0000	0.501	0000 0000 0100 0000	0.001
512/1024	10 0000 0000	1000 0000 0000 0000	0.500	0000 0000 0000 0000	0.000
511/1024	01 1111 1111	0111 1111 1100 0000	0.499	1111 1111 1100 0000	-0.001
		•••			
1/1024	00 0000 0001	0000 0000 0100 0000	0.001	1000 0000 0100 0000	-0.499
0/1024	00 0000 0000	0000 0000 0000 0000	0.000	1000 0000 0000 0000	-0.500

51.9 THRESHOLD DETECT OPERATION

Threshold Detect is a significant extension of the Auto-Scan feature offered in previous 10-bit A/D modules. In addition to being able to repeatedly sample a predefined sequence of analog channels, Threshold Detect allows the user to define match conditions based on the conversion results, and generate an interrupt based on these conditions. During normal operation, this can potentially reduce the amount of CPU time spent on processing A/D interrupts. For low-power applications, this can allow the CPU to remain inactive for longer periods, waking only when specific analog conditions are met.

When selected by the user, Threshold Detect changes the operation of the A/D results buffer by making it a read/write array for both conversion results and comparison (threshold) values. It also brings into play the AD1CHIT registers, which are used to indicate match conditions. Independently selectable comparison and buffer storage settings make a wide range of operating combinations possible.

51.9.1 Operating Modes

The operation of Threshold Detect is mostly controlled by the AD1CON5 register (Figure 51-4). The ASEN bit (AD1CON5<15>) controls overall operation of Threshold Detect; setting this bit enables the functionality.

As with Legacy Auto-Scan operation, the channels to be included are selected using the AD1CSSH/L registers. Setting a particular bit in either register includes the corresponding channel in an automatic sequential scan. One or more channels may be selected. After the channels have been selected, setting both the CSCNA and ASEN bits to enable a single scan of the designated channels. The scan itself is triggered by the trigger source programmed by the SSRC<3:0> bits.

Note:	Legacy Auto-Scan (i.e., sequential scanning of analog channels on MUX A, without any comparison) is controlled by the CSCNA bit (AD1CON2<10>) and does not depend on the ASEN bit to function.

The LPEN bit (AD1CON5<14>) allows Threshold Detect to function with a low-power feature. By design, Threshold Detect can perform comparison operations when the device is in Sleep or Idle modes, waking the CPU when it generates an interrupt. Setting LPEN configures the device to return to low-power operation after the interrupt has been serviced.

The Compare Mode bits, CM<1:0> (AD1CON5<1:0>), select the type of comparison to be performed. Four types are available:

- the result of the current conversion is greater than a reference threshold;
- the result of the current conversion is less than a reference threshold;
- the result of the current conversion is between two predefined thresholds (“Inside Window”); and
- the result of the current conversion is outside of the predefined thresholds (“Outside Window”)

The Write Mode bits WM<1:0> (AD1CON5<3:2>) determine the disposition of the conversion. Three options are available:

- discard the conversion after the comparison has been performed;
- store the conversion after the comparison has been performed; and
- store the conversion without comparison (Legacy mode).

51.9.1.1 BUFFER OPERATION AND COMPARISONS

For buffer write modes that involve storing conversions (WM<1:0> = 0x), the BUFM and BUFREGEN bits control how the buffer functions (as a channel indexed, single FIFO or split FIFO buffer). However, when the compare and store option is selected (WM<1:0> = 01), using a FIFO mode may overwrite the buffers of other channels and cause unpredictable comparison results. For that reason, always use Channel Indexed Buffer mode (BUFREGEN = 1) when using the compare and store option.

51.9.1.2 BUFFER OPERATION IN WINDOWED COMPARISONS (CHANNEL MIRRORING)

The use of windowed comparisons changes the available options for the results buffer. To accommodate the storage of two threshold values, the buffer is automatically split into halves, similar to Split FIFO mode. Buffer addresses in each half are paired, with the lowest address in one buffer matched to the buffer address in the upper half. (For example, in a 16-word buffer, ADC1BUF0 is paired with ADC1BUF9, ADC1BUF1 is paired with ADC1BUF10, and so on.) This pairing is referred to as “channel mirroring”.

Mirroring can obviously be applied only to the lower A/D channels; for most devices, this corresponds to the lower half of the external analog channels. This does not mean that those buffer locations cannot be used for other purposes. However, storing any other data in a particular buffer location where channel mirroring is being used may result in misleading comparison evaluations.

51.9.2 Setting Comparison Thresholds

The comparison thresholds for Threshold Detect are set by writing the desired values to an appropriate location in the A/D results buffer. This can only be done when the module is deactivated (AD1CON1<15> = 0).

The location of the threshold is determined by the comparison type. For simple greater than and less than comparisons, the value is written to the buffer location corresponding to the input channel to be monitored. For example, if AN0 is to be monitored for a voltage over a certain level, the ceiling threshold is stored in ADC1BUF0.

The location of the thresholds for windowed comparisons are written to two addresses. The lower value is written to the address corresponding to the monitored channel. The upper value is stored in the corresponding mirrored address in the upper half of the buffer. To expand on the previous example, if the conversion on AN0 is to be a windowed comparison, the floor threshold is stored in ADC1BUF0, while the ceiling threshold is stored in ADC1BUF9.

51.9.3 Compare Hit Registers

To determine if a particular event has occurred, the A/D module uses two registers to record match events. These registers are referred to as the Compare Hit registers, and are designated: AD1CHITL and AD1CHITH. The registers map their individual bits sequentially to each of the (up to) 32 analog channels. If a particular channel in a device is not implemented, the corresponding Compare Hit bit is not implemented.

Each bit serves as an event semaphore for its corresponding channel. When the programmed event occurs on that channel, the bit becomes set and stays set until it is cleared by the application. It is the user’s responsibility to clear the bits after the application has evaluated them.

Depending on the event, more than one Compare Hit bit may be set. The significance of a set bit must be interpreted by the application in the context of the Compare mode selected. Particular examples are covered in **Section 51.9.5 “Comparison Mode Examples”**.

51.9.4 Threshold Detect Interrupts

The A/D module can generate an interrupt and set the AD1IF flag based on Threshold Detect operation. This is based on completion of a Threshold Detect sequence and/or the occurrence of a valid comparison. Threshold Detect interrupts are in addition to the interrupt configured by the SMPI bits (AD1CON2<6:2>), described in **Section 51.8.1 “Number of Conversions per Interrupt”**.

The Threshold Detect interrupt is configured by the ASINT<1:0> bits (AD1CON5<9:8>). Options include interrupt after a scan sequence, interrupt after a scan sequence with a valid match, interrupt after a valid match (without waiting for the sequence to end) or no interrupt.

Since the SMPI and ASINT bits configure the same interrupt flag, it is possible that an interrupt may mean the routine completion of a number of samples or by the completion of a Threshold Detect sequence. If both types of interrupt are being used, it is the user’s responsibility to determine the type of interrupt that has occurred in the interrupt service routine. If a Threshold Detect interrupt is enabled (ASINT<1:0> $\neq$ 00), the application must examine the Compare Hit registers to determine on which channel the event occurred, then clear the CHHn bits.

51.9.5 Comparison Mode Examples

The following examples show the effect of valid comparisons on the results buffer and the Compare Hit registers. In each figure, changes within the registers are indicated in bold.

For the sake of simplicity, the examples assume a device with only 16 analog inputs. Devices with a greater number of channels, and thus, larger results buffers and two Compare Hit registers, will function in a similar fashion.

Note:	When using any comparison mode, always use channel indexed buffer storage (BUFREGEN = 1). Otherwise, the threshold values for other channels may be overwritten, resulting in unpredictable comparisons.

51.9.5.1 SIMPLE COMPARISONS (GREATER AND LESS THAN RESULTS)

When the Compare Mode bits, CM<1:0> (AD1CON5<1:0>), are programmed as '0x', the converter compares the sampled value to see if it is greater than (CM<1:0> = 01) or less than (CM<1:0> = 00) the threshold value in the buffer location. If the condition is met, both of the following occur:

- the Compare Hit bit (CHHn) for the corresponding channel is set
- if the Write Mode bits, WM<1:0> (AD1CON5<3:2>), are programmed to '01', the converted value is written to the buffer, replacing the threshold value. If WM<1:0> = '10', the converted value is discarded.

The changes to the result buffer and the Compare Hit register are shown in Figure 51-12. Note that they are the same for both types of simple comparison.

Figure 51-12: Simple Comparison Operations (Greater Than and Less Than)

Before Conversion and Comparison

ADC1BUF15	—
ADC1BUF14	—
ADC1BUF13	—
ADC1BUF12	—
ADC1BUF11	—
ADC1BUF10	—
ADC1BUF9	—
ADC1BUF8	—
ADC1BUF7	—
ADC1BUF6	—
ADC1BUF5	—
ADC1BUF4	—
ADC1BUF3	—
ADC1BUF2	Threshold Value
ADC1BUF1	—
ADC1BUF0	—

After Conversion and Comparison

	Compare Only ('10')	**Compare and Store ('01')**
ADC1BUF15	—	—
ADC1BUF14	—	—
ADC1BUF13	—	—
ADC1BUF12	—	—
ADC1BUF11	—	—
ADC1BUF10	—	—
ADC1BUF9	—	—
ADC1BUF8	—	—
ADC1BUF7	—	—
ADC1BUF6	—	—
ADC1BUF5	—	—
ADC1BUF4	—	—
ADC1BUF3	—	—
ADC1BUF2	Threshold Value	**Conversion Value**
ADC1BUF1	—	—
ADC1BUF0	—	—

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	**2**	1	0
0	0	0	0	0	**1**	0	0

51.9.5.2 INSIDE WINDOW COMPARISON

When the Compare Mode bits, CM<1:0>, are programmed as '10', the converter compares the sampled value to see if it falls between the threshold values in the buffer and mirrored channel location. Since the value in the mirrored channel location is always the greater value of the two thresholds, the condition is met when:

Threshold 2 > Converted Value > Threshold 1

In this case, both of the following occur:

- the Compare Hit bit (CHHn) for the corresponding channel is set; the Compare Hit bit for the mirrored channel remains cleared
- if the Write Mode bits, WM<1:0> (AD1CON5<3:2>), are programmed to '01', the converted value is written to the buffer, replacing the lower threshold value. If WM<1:0> = '10', the converted value is discarded.

The changes to the result buffer and the Compare Hit register are shown in Figure 51-13.

Figure 51-13: Inside Window Comparison Operation

Before Conversion and Comparison

ADC1BUF15	—
ADC1BUF14	—
ADC1BUF13	—
ADC1BUF12	—
ADC1BUF11	—
ADC1BUF10	Threshold 2
ADC1BUF9	—
ADC1BUF8	—
ADC1BUF7	—
ADC1BUF6	—
ADC1BUF5	—
ADC1BUF4	—
ADC1BUF3	—
ADC1BUF2	Threshold 1
ADC1BUF1	—
ADC1BUF0	—

After Conversion and Comparison

	Compare Only ('10')	Compare and Store ('01')
ADC1BUF15	—	—
ADC1BUF14	—	—
ADC1BUF13	—	—
ADC1BUF12	—	—
ADC1BUF11	—	—
ADC1BUF10	Threshold 2	Threshold 2
ADC1BUF9	—	—
ADC1BUF8	—	—
ADC1BUF7	—	—
ADC1BUF6	—	—
ADC1BUF5	—	—
ADC1BUF4	—	—
ADC1BUF3	—	—
ADC1BUF2	Threshold 1	**Conversion Value**
ADC1BUF1	—	—
ADC1BUF0	—	—

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	**2**	1	0
0	0	0	0	0	**1**	0	0

51.9.5.3 OUTSIDE WINDOW COMPARISON

When the Compare Mode bits CM<1:0> are programmed as '11', the converter compares the sampled value to see if it falls outside of the threshold values in the buffer and mirrored channel location. Again, since the value in the mirrored channel location is always the greater value of the two thresholds, the condition is met when either:

Converted value >Threshold 2,

or

Threshold 1 > Converted Value

In these cases, the following occurs:

- The Compare Hit bit (CHHn) for the corresponding channel is set.
- If the converted value is greater than Threshold 2, the CHHn bit for the mirrored channel is also set. If it is less than Threshold 1, the mirrored channel bit remains '0'.
- if the Write Mode bits, WM<1:0> (AD1CON5<3:2>), are programmed to '01':
  - if the converted value is above Threshold 2, the converted value is written to the mirrored channel address, replacing the upper threshold value.
  - if the converted value is below Threshold 1, the converted value is written to the channel address, replacing the lower threshold value.
- If WM<1:0> = 10, the converted value is discarded.

The changes to the result buffer and the Compare Hit register are shown in Figure 51-14 (over the upper threshold) and Figure 51-15 (under the lower threshold).

Note that when a Windowed Comparison mode is selected and channel mirroring is enabled, nothing prevents a conversion from another operation from being stored in the mirrored channel location. In the previous examples of windowed operation, if AN10 is included in a Threshold Detect operation, a conversion on AN10 might be tested against the upper threshold for AN2, stored in that location. This could result in the threshold value being overwritten, and/or the CHH10 bit being set.

For this reason, users must always carefully consider the allocation and use of the upper analog channels (both external and internal) when using Windowed Compare modes. Wherever possible, exclude the upper analog channels for Threshold Detect operations, and convert and test those channels in a separate routine.

Figure 51-14: Outside Window Comparison Operation (Over Threshold 2)

Before Conversion and Comparison

ADC1BUF15	—
ADC1BUF14	—
ADC1BUF13	—
ADC1BUF12	—
ADC1BUF11	—
ADC1BUF10	Threshold 2
ADC1BUF9	—
ADC1BUF8	—
ADC1BUF7	—
ADC1BUF6	—
ADC1BUF5	—
ADC1BUF4	—
ADC1BUF3	—
ADC1BUF2	Threshold 1
ADC1BUF1	—
ADC1BUF0	—

After Conversion and Comparison

	Compare Only (‘10’)	Compare and Store (‘01’)
ADC1BUF15	—	—
ADC1BUF14	—	—
ADC1BUF13	—	—
ADC1BUF12	—	—
ADC1BUF11	—	—
ADC1BUF10	Threshold 2	**Conversion Value**
ADC1BUF9	—	—
ADC1BUF8	—	—
ADC1BUF7	—	—
ADC1BUF6	—	—
ADC1BUF5	—	—
ADC1BUF4	—	—
ADC1BUF3	—	—
ADC1BUF2	Threshold 1	Threshold 1
ADC1BUF1	—	—
ADC1BUF0	—	—

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

AD1CHITL							
15	14	13	12	11	**10**	9	8
0	0	0	0	0	**1**	0	0
7	6	5	4	3	**2**	1	0
0	0	0	0	0	**1**	0	0

Figure 51-15: Outside Window Comparison Operation (Under Threshold 1)

Before Conversion and Comparison

ADC1BUF15	—
ADC1BUF14	—
ADC1BUF13	—
ADC1BUF12	—
ADC1BUF11	—
ADC1BUF10	Threshold 2
ADC1BUF9	—
ADC1BUF8	—
ADC1BUF7	—
ADC1BUF6	—
ADC1BUF5	—
ADC1BUF4	—
ADC1BUF3	—
ADC1BUF2	Threshold 1
ADC1BUF1	—
ADC1BUF0	—

After Conversion and Comparison

	Compare Only (‘10’)	Compare and Store (‘01’)
ADC1BUF15	—	—
ADC1BUF14	—	—
ADC1BUF13	—	—
ADC1BUF12	—	—
ADC1BUF11	—	—
ADC1BUF10	Threshold 2	Threshold 2
ADC1BUF9	—	—
ADC1BUF8	—	—
ADC1BUF7	—	—
ADC1BUF6	—	—
ADC1BUF5	—	—
ADC1BUF4	—	—
ADC1BUF3	—	—
ADC1BUF2	Threshold 1	**Conversion Value**
ADC1BUF1	—	—
ADC1BUF0	—	—

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

AD1CHITL							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	**2**	1	0
0	0	0	0	0	**1**	0	0

51.10 CONVERSION SEQUENCE EXAMPLES

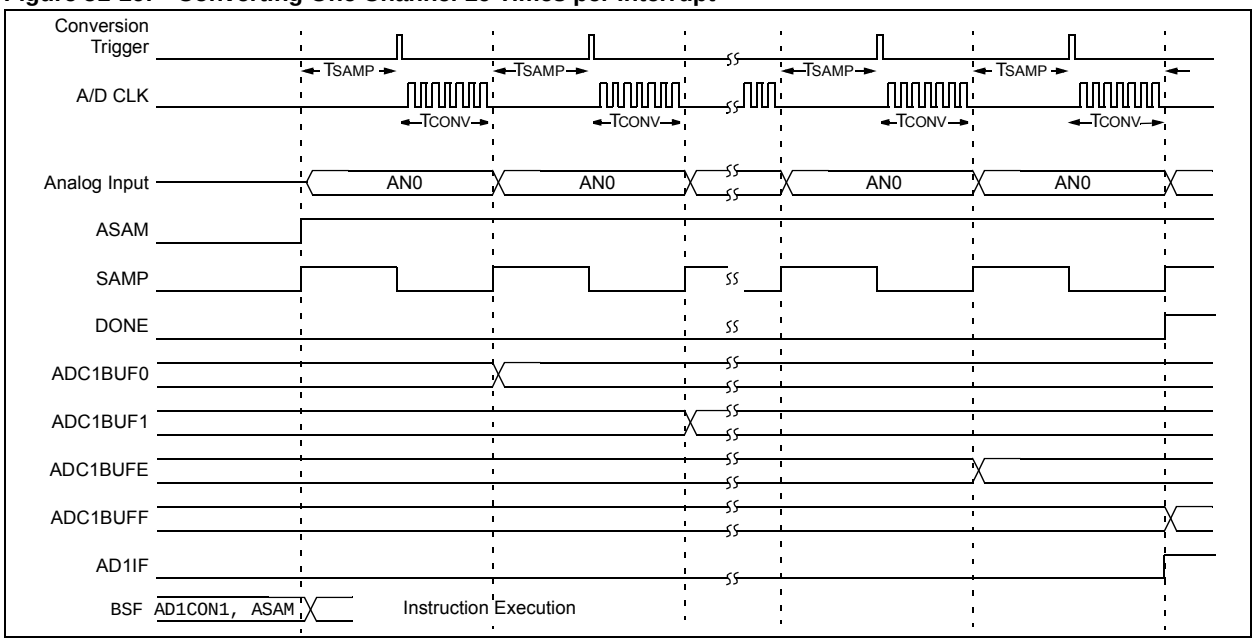
The following configuration examples show the A/D operation in different sampling and buffering configurations. In each example, setting the ASAM bit starts automatic sampling. A conversion trigger ends sampling and starts conversion.

51.10.1 Sampling and Converting a Single Channel Multiple Times

Figure 51-16 and Example 51-6 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-and-Hold channel. All other input selection bits are unused.

Figure 51-16: Converting One Channel 16 Times per Interrupt



Example 51-6: Sampling and Converting a Single Channel Multiple Times

```
int ADCValue, count;
int *ADC16Ptr;
ANSELB  = 0x0001;                       // Only AN0 as analog input
AD1CON1 = 0x00E0;                       // Internal counter triggers conversion
AD1CHS  = 0x0000;                       // Connect AN0 as positive input
AD1CSSL = 0;
AD1CON3 = 0x0F00;                       // Sample time = 15Tad, Tad = Tcy
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 51-7: Converting a Single Channel 16 Times per Interrupt

A/D Configuration:

- Select AN0 for S/H+ Input (CH0SA<3:0> = 0000)
- Select VR- for S/H- Input (CH0NA = 0)
- Configure for No Input Scan (CSCNA = 0)
- Use Only MUX A for Sampling (ALTS = 0)
- Set AD1IF on Every 16th Sample (SMPI<3:0> = 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):

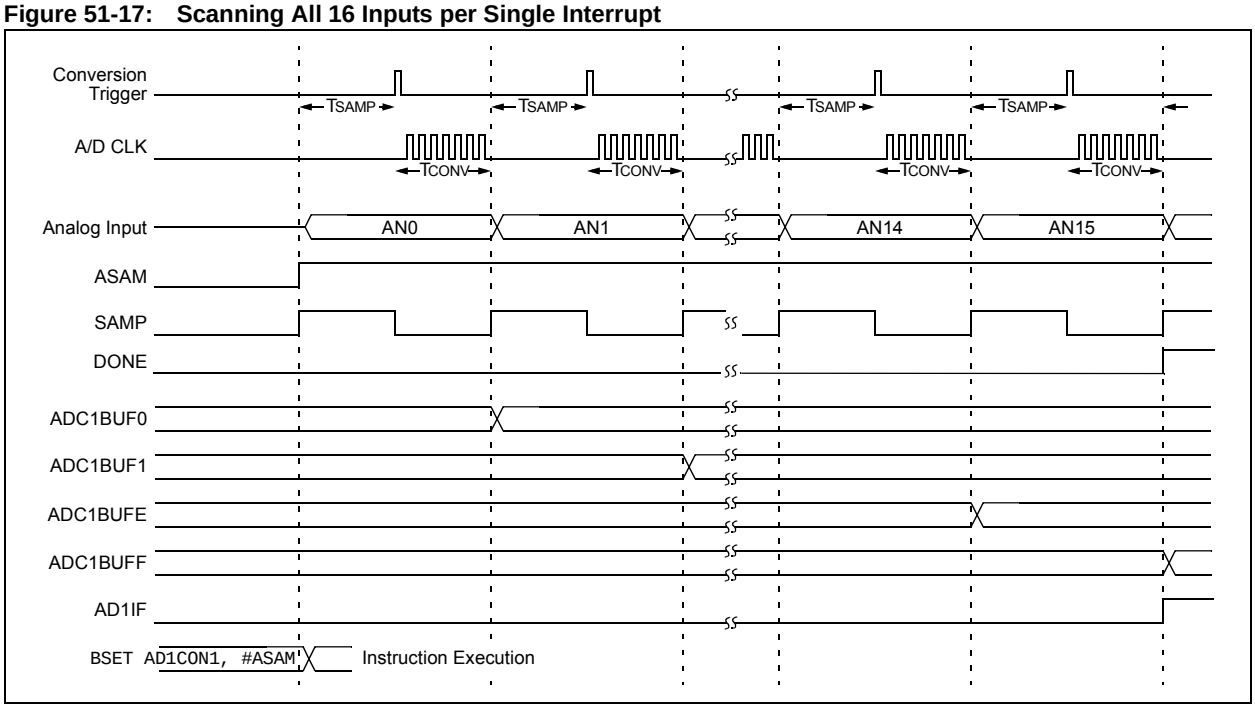
Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	AN0, Sample 1	AN0, Sample 17
ADC1BUF1	AN0, Sample 2	AN0, Sample 18
ADC1BUF2	AN0, Sample 3	AN0, Sample 19
ADC1BUF3	AN0, Sample 4	AN0, Sample 20
ADC1BUF4	AN0, Sample 5	AN0, Sample 21
ADC1BUF5	AN0, Sample 6	AN0, 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

51.10.2 A/D Conversions While Scanning Through All Analog Inputs

Figure 51-17 and Example 51-9 illustrate a typical setup, where all available analog input channels are sampled and converted. In this instance, 16 analog inputs are assumed. The set CSCNA bit specifies scanning of the A/D inputs to the S/H positive input. Other conditions are similar to **Section 51.10.1 "Sampling and Converting a Single Channel Multiple Times"**.

Initially, the AN0 input is sampled and converted. The result is stored in the ADC1BUF 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 51-17: Scanning All 16 Inputs per Single Interrupt



Example 51-8: Sampling and Converting All Channels

```
int ADCValue, count;
int *ADC16Ptr;
ANSELA  = 0xFFFF;                       // Configure all pins as analog inputs
ANSELB  = 0xFFFF;                       // (small-pincount device w/16 channels shown here)
ANSELC  = 0xFFFF;                       //
AD1CSSL = 0xFFFF;                       // Include all channels in scan
AD1CON1 = 0x00E0;                       // Internal counter triggers conversion
AD1CON3 = 0x0F00;                       // Sample time = 15Tad, Tad = Tcy
AD1CON2 = 0x043C;                       // 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 51-9: Scanning and Converting All 16 Channels per Single Interrupt

A/D Configuration:

- Select Any Channel for S/H+ Input (CH0SA<3:0> = xxxx)
- Select VR- for S/H- 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 (SMPI<3:0> = 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

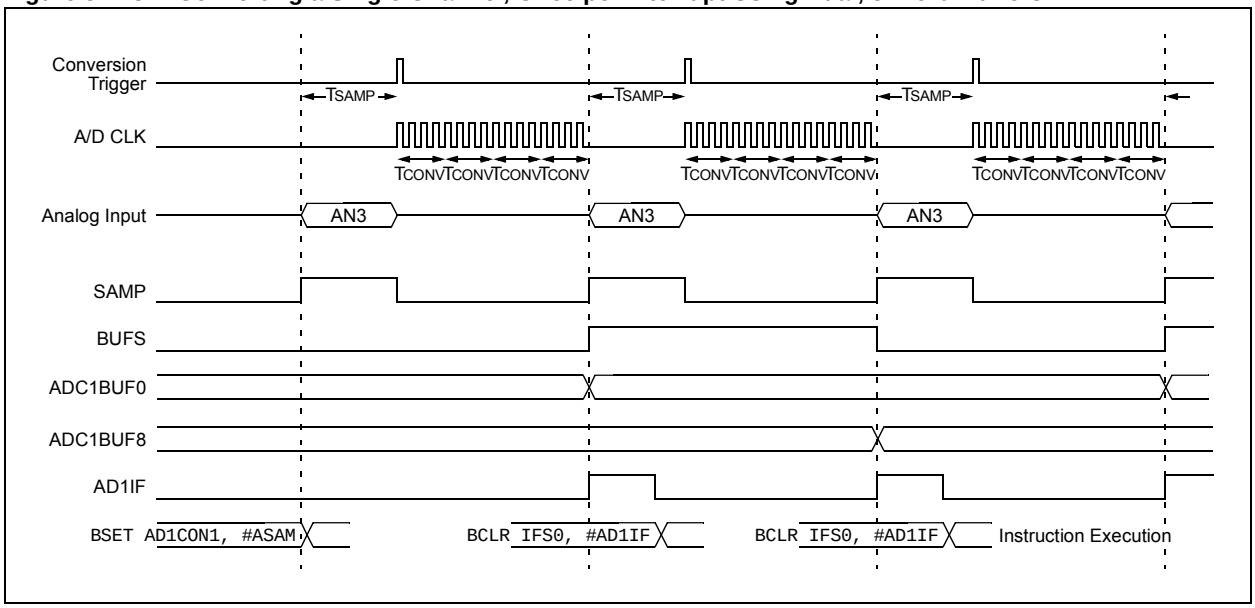
Results Stored in Buffer (after 2 cycles):

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)

51.10.3 Using Dual Buffers

Figure 51-18 and Example 51-10 demonstrate using dual buffers and alternating the buffer fill. Setting the BUFM bit enables dual 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 toggled after each interrupt.

Figure 51-18: Converting a Single Channel, Once per Interrupt Using Dual, 8-Word Buffers



Example 51-10: Converting a Single Channel Once per Interrupt, Dual Buffer Mode

A/D Configuration:

- Select AN3 for S/H+ Input (CH0SA<3:0> = 0011)
- Select VR- for S/H- Input (CH0NA = 0)
- Configure for No Input Scan (CSCNA = 0)
- Use Only MUX A for Sampling (ALTS = 0)
- Set AD1IF on Every Sample (SMPI<3:0> = 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):

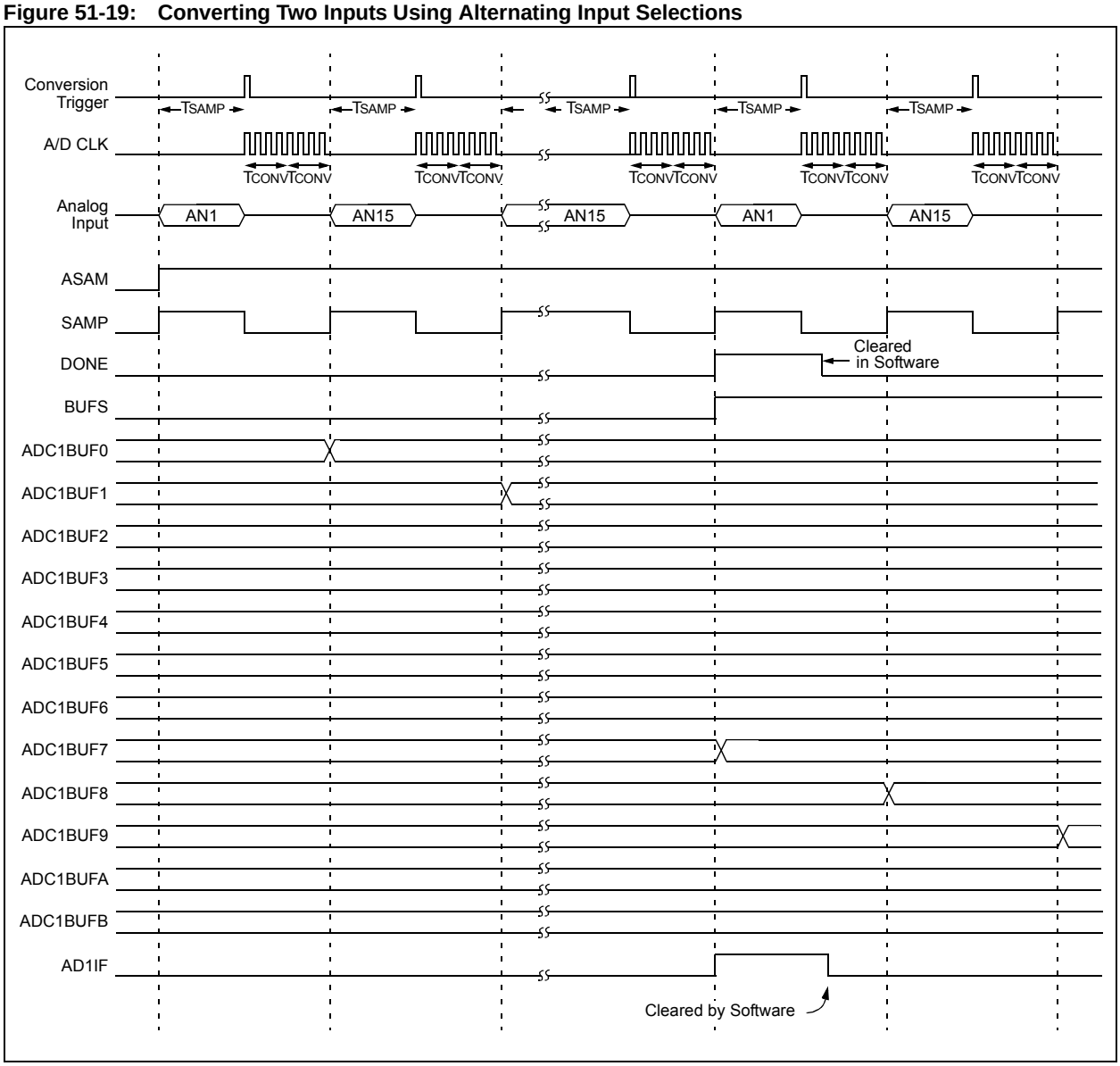
Buffer Address	Buffer Contents at 1st AD1IF Event	Buffer Contents at 2nd AD1IF Event
ADC1BUF0	Sample 1 (AN3, Sample 1)	(undefined)
ADC1BUF1	(undefined)	(undefined)
ADC1BUF2	(undefined)	(undefined)
ADC1BUF3	(undefined)	(undefined)
ADC1BUF4	(undefined)	(undefined)
ADC1BUF5	(undefined)	(undefined)
ADC1BUF6	(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)

51.10.4 Using Alternating MUX A and MUX B Input Selections

Figure 51-19 and Example 51-11 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 51-19: Converting Two Inputs Using Alternating Input Selections



Example 51-11: Converting Two Inputs by Alternating MUX A and MUX B

A/D Configuration:

- Select AN1 for MUX A S/H+ Input (CH0SA<3:0> = 0001)
- Select VR- for MUX A S/H- Input (CH0NA = 0)
- Configure for No Input Scan (CSCNA = 0)
- Select AN15 for MUX B S/H+ Input (CH0SB<3:0> = 1111)
- Select VR- for MUX B S/H- Input (CH0NB = 0)
- Alternate MUX A and MUX B for Sampling (ALTS = 1)
- Set AD1IF on Every 8th Sample (SMPI<3:0> = 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)

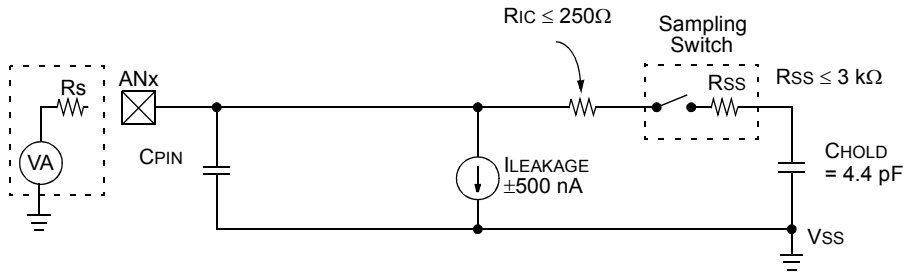
51.11 A/D SAMPLING REQUIREMENTS

The Analog Input model of the 12-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 51.17 "Electrical Specifications"**.

Figure 51-20: 10-Bit A/D Converter Analog Input Model



Legend:	CPIN	= Input Capacitance
	VT	= Threshold Voltage
	ILEAKAGE	= Leakage Current at the pin due to various junctions
	RIC	= Interconnect Resistance
	RSS	= Sampling Switch Resistance
	CHOLD	= Sample/Hold Capacitance (from DAC)

Note: CPIN value depends on device package and is not tested. Effect of CPIN negligible if Rs ≤ 5 kΩ.

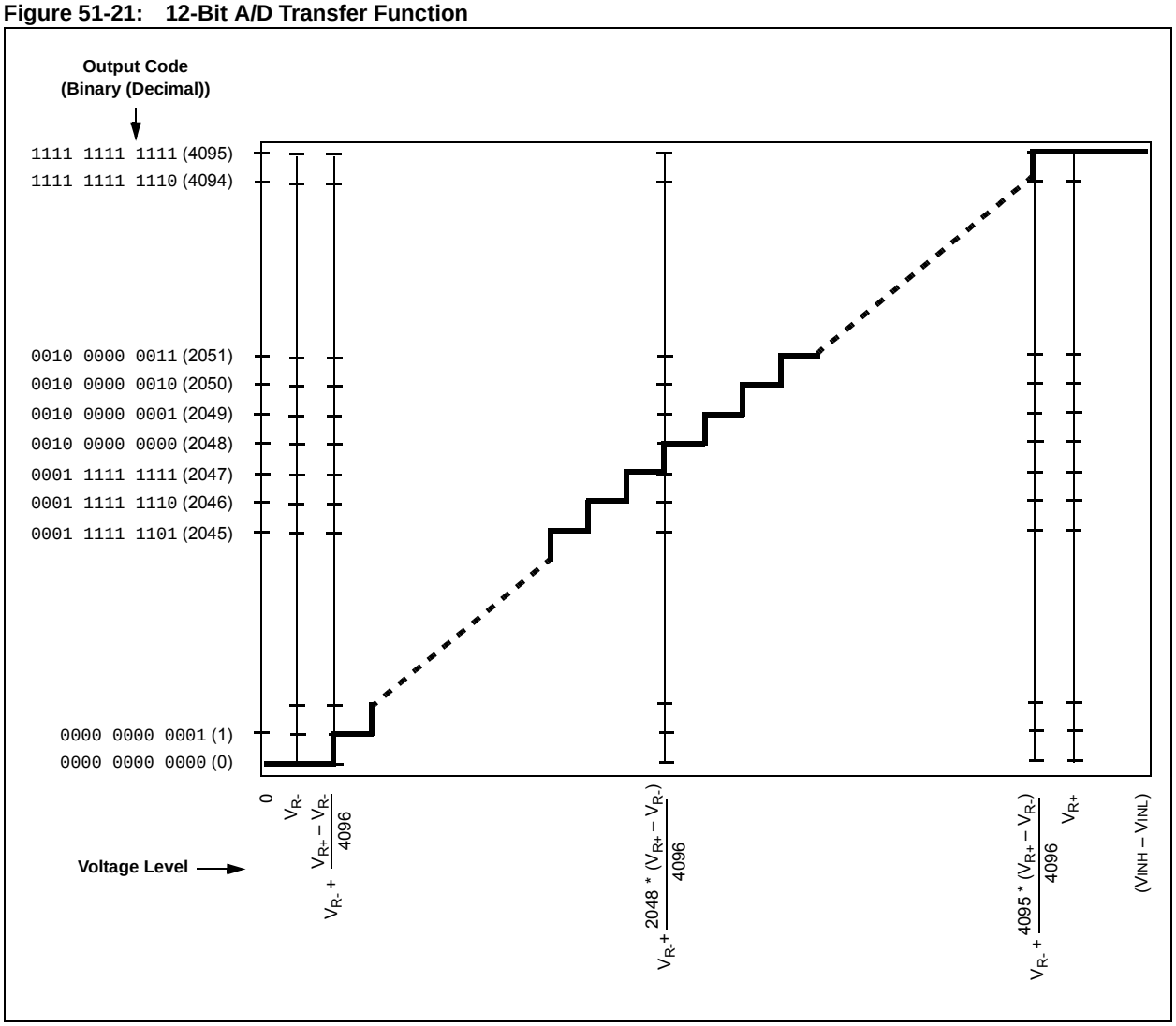
51.12 TRANSFER FUNCTIONS

The transfer functions of the A/D Converter, in 12-bit and 10-bit resolution, are shown in Figure 51-21 and Figure 51-22, respectively. In both cases, the difference of the input voltages, (VINH – VINL), is compared to the reference, ((VR+) – (VR-)).

For the 12-bit transfer function:

- The first code transition occurs when the input voltage is ((VR+) – (VR-))/4096 or 1.0 LSb.
- The 0000 0000 0001 code is centered at VR- + (1.5 * ((VR+) – (VR-))/4096).
- The 0010 0000 0000 code is centered at VREFL + (2048.5 * ((VR+) – (VR-))/4096).
- An input voltage less than VR- + (((VR-) – (VR-))/4096) converts as 0000 0000 0000.
- An input voltage greater than (VR-) + (1023 ((VR+) – (VR-))/4096) converts as 1111 1111 1111.

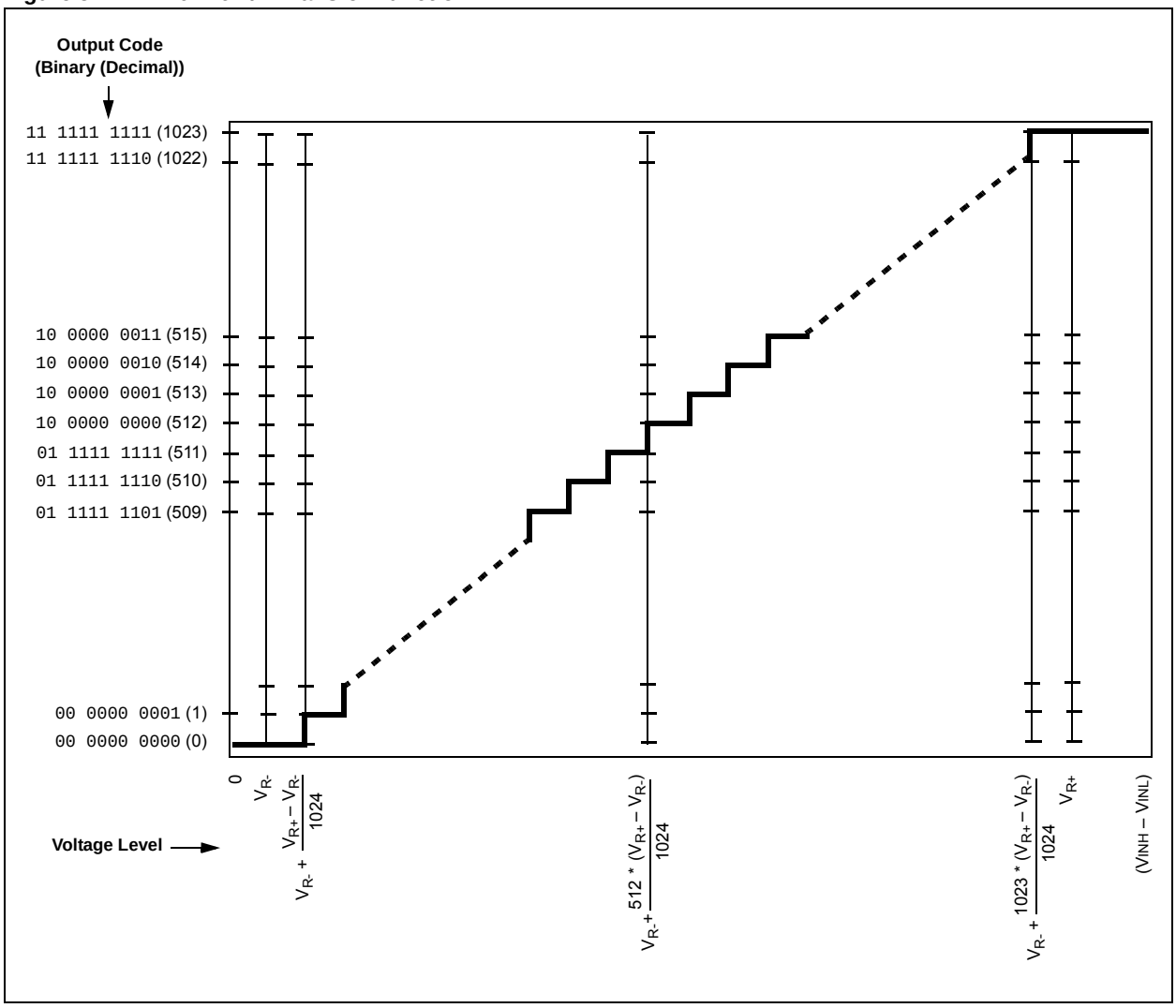
Figure 51-21: 12-Bit A/D Transfer Function



For the 10-bit transfer function (when 10-bit resolution is available):

- The first code transition occurs when the input voltage is ((VR+) – (VR-))/1024 or 1.0 LSb.
- The `00 0000 0001` code is centered at VR- + (1.5 * ((VR+) – (VR-))/1024).
- The `10 0000 0000` code is centered at VREFL + (512.5 * ((VR+) – (VR-))/1024).
- An input voltage less than VR- + (((VR-) – (VR-))/1024) converts as `00 0000 0000`.
- An input voltage greater than (VR-) + (1023((VR+) – (VR-))/1024) converts as `11 1111 1111`.

Figure 51-22: 10-Bit A/D Transfer Function



51.13 A/D ACCURACY/ERROR

Refer to **Section 51.19 "Related Application Notes"** for a list of documents that discuss A/D accuracy.

51.14 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.

51.14.1 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.

51.14.2 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 PWRSAV 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 (SSRC<3:0> = 0111). To use the automatic conversion option, the ADON bit should be set in the instruction prior to the PWRSAV instruction.

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

51.14.3 A/D Operation During CPU Idle Mode

The ADSIDL bit (AD1CON1<13>) determines whether the module stops or continues operation 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 PWRSAV 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.

51.14.4 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. The ADC module is enabled only when the ADC1MD bit in the PMDx register is cleared.

51.15 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 to be 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.

51.16 REGISTER MAPS

A summary of the registers associated with the PIC24F 10-Bit A/D Converter is provided in Table 51-6.

Table 51-6: 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	ADC Data Buffer 0/Threshold Detect Value for AN0																xxxx
ADC1BUF1	ADC Data Buffer 1/Threshold Detect Value for AN1																xxxx
ADC1BUF2	ADC Data Buffer 2/Threshold Detect Value for AN2																xxxx
ADC1BUF3	ADC Data Buffer 3/Threshold Detect Value for AN3																xxxx
ADC1BUF4	ADC Data Buffer 4/Threshold Detect Value for AN4																xxxx
ADC1BUF5	ADC Data Buffer/Threshold Detect Value for AN5																xxxx
ADC1BUF6	ADC Data Buffer/Threshold Detect Value for AN6																xxxx
ADC1BUF7	ADC Data Buffer 7/Threshold Detect Value for AN7																xxxx
ADC1BUF8	ADC Data Buffer 8/Threshold Detect Value for AN8																xxxx
ADC1BUF9	ADC Data Buffer 9/Threshold Detect Value for AN9																xxxx
. . .	. . .																. . .
ADC1BUFn(1)	ADC Data Buffer n/Threshold Detect Value for ANn																xxxx
AD1CON1	ADON	—	ADSIDL	—	—	MODE12(2)	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE	0000
AD1CON2	PVCFG1	PVCFG0	NVCFG	OFFCAL	BUFREGEN	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	ADRC	EXTSAM	PUMPEN(2)	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH(3)	CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16	xxxx
AD1CSSL	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON5	ASEN	LPEN	CTMREQ	BGREQ(2)	VRSREQ(2)	—	ASINT1	ASINT0	—	—	—	—	WM1	WM0	CM1	CM0	0000
AD1CHITH(3)	CHH31	CHH30	CHH29	CHH28	CHH27	CHH26	CHH25	CHH24	CHH23	CHH22	CHH21	CHH20	CHH19	CHH18	CHH17	CHH16	0000
ADCCHITL	CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8	CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0	0000
AD1CTMENH(3)	CTMEN31	CTMEN30	CTMEN29	CTMEN28	CTMEN27	CTMEN26	CTMEN25	CTMEN24	CTMEN23	CTMEN22	CTMEN21	CTMEN20	CTMEN19	CTMEN18	CTMEN17	CTMEN16	0000
AD1CTMENL	CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMEN11	CTMEN10	CTMEN9	CTMEN8	CTMEN7	CTMEN6	CTMEN5	CTMEN4	CTMEN3	CTMEN2	CTMEN1	CTMEN0	0000

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

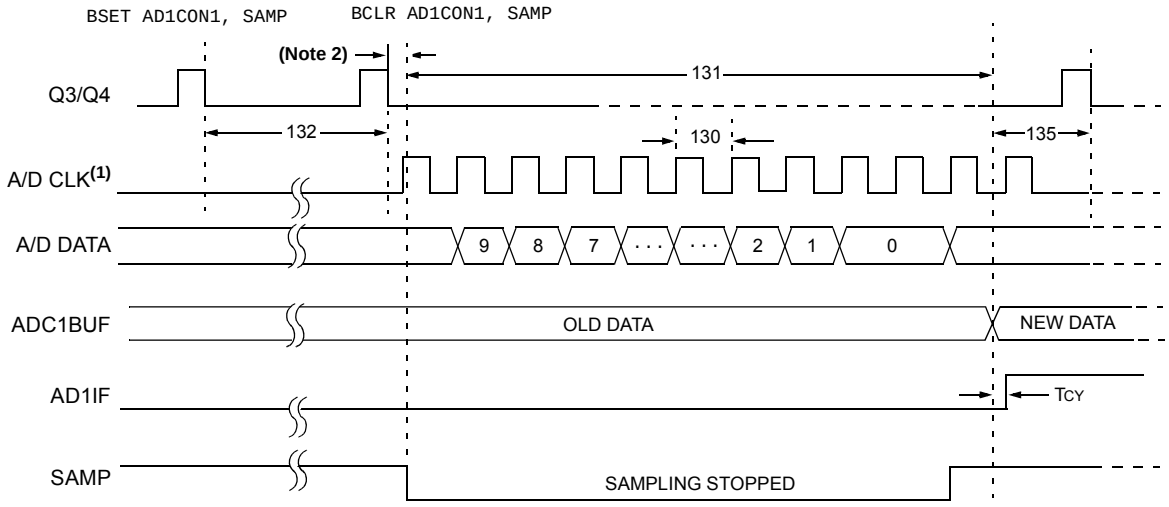
Note 1: The number of AD1BUF registers is equal to at least the number of external analog channels, rounded up the nearest even number. Refer to the specific device data sheet for the exact number.

2: These bits are not implemented in all devices. Refer to the specific device data sheet.

3: These registers and bits are implemented in devices with more than 16 analog channels only. Not all bits may be implemented in 'H' registers. Refer to the specific device data sheet.

51.17 ELECTRICAL SPECIFICATIONS

Figure 51-23: A/D Conversion Timing



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

2: This is a minimal RC delay (typically 100 ns) which also disconnects the holding capacitor from the analog input.

Table 51-7: A/D Conversion Requirements

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
AD61	tPSS	Sample Start Delay from Setting SAMP	2	—	3	TAD	
AD130	TAD	A/D Clock Period	75	—	—	ns	TOSC-based
			—	250	—	ns	A/D RC mode
AD131	TCNV	Conversion Time (not including acquisition time)	11	—	12	TAD	**(Note 1)**
AD132	TACQ	Acquisition Time	—	—	750	ns	**(Note 2)**
AD135	TSWC	Switching Time from Convert to Sample	—	—	**(Note 3)**		
AD137	TDIS	Discharge Time	0.5	—	—	TAD	
		A/D Stabilization Time (from setting ADON to setting SAMP)	—	300	—	ns	

Note 1: The ADC1BUF register may be read on the following TCY cycle.

2: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (AVDD to AVSS or AVSS to AVDD).

3: On the following cycle of the device clock.

51.18 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 51.17 "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 kΩ), 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 μF 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.

51.19 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 12-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.

51.20 REVISION HISTORY

Revision A (March 2011)

This is the initial released revision of this document.

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.

Trademarks

Printed on recycled paper.

ISBN: 978-1-60932-987-7

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 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

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

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/18/11