Section 49. 10-Bit ADC with 4 Simultaneous Conversions

HIGHLIGHTS

This section of the manual contains the following major topics:

> **Note:** This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all PIC24F devices.
>
> Please consult the note at the beginning of the **"10-Bit ADC with 4 Simultaneous Conversions"** chapter in the current device data sheet to check whether this document supports the device you are using.
>
> Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com

49.1 INTRODUCTION

This document describes the features and associated operational modes of the Successive Approximation (SAR) Analog-to-Digital Converter (ADC) available on the PIC24F families of devices.

Figure 49-1 illustrates a block diagram of the ADC module.

The PIC24F ADC module has the following key features:

- SAR conversion
- Up to 1.1 Msps conversion speed
- Up to 6 analog input pins
- External voltage reference input pins
- Four unipolar differential Sample and Hold (S&H) amplifiers
- Simultaneous sampling of up to four analog input pins
- Automatic Channel Scan mode
- Selectable conversion trigger source
- Up to 16-word conversion result buffer
- Selectable Buffer Fill modes
- Operation during CPU Sleep and Idle modes

Depending on the device variant, the ADC module may have up to 6 analog input pins, designated AN0-AN5. These analog inputs are connected by multiplexers to four S&H amplifiers, designated CH0-CH3. The analog input multiplexers have two sets of control bits, designated as MUXA (CHySA/CHyNA) and MUXB (CHySB/CHyNB). These control bits select a particular analog input for conversion. The MUXA and MUXB control bits can alternatively select the analog input for conversion. Unipolar differential conversions are possible on all channels using certain input pins (see Figure 49-1).

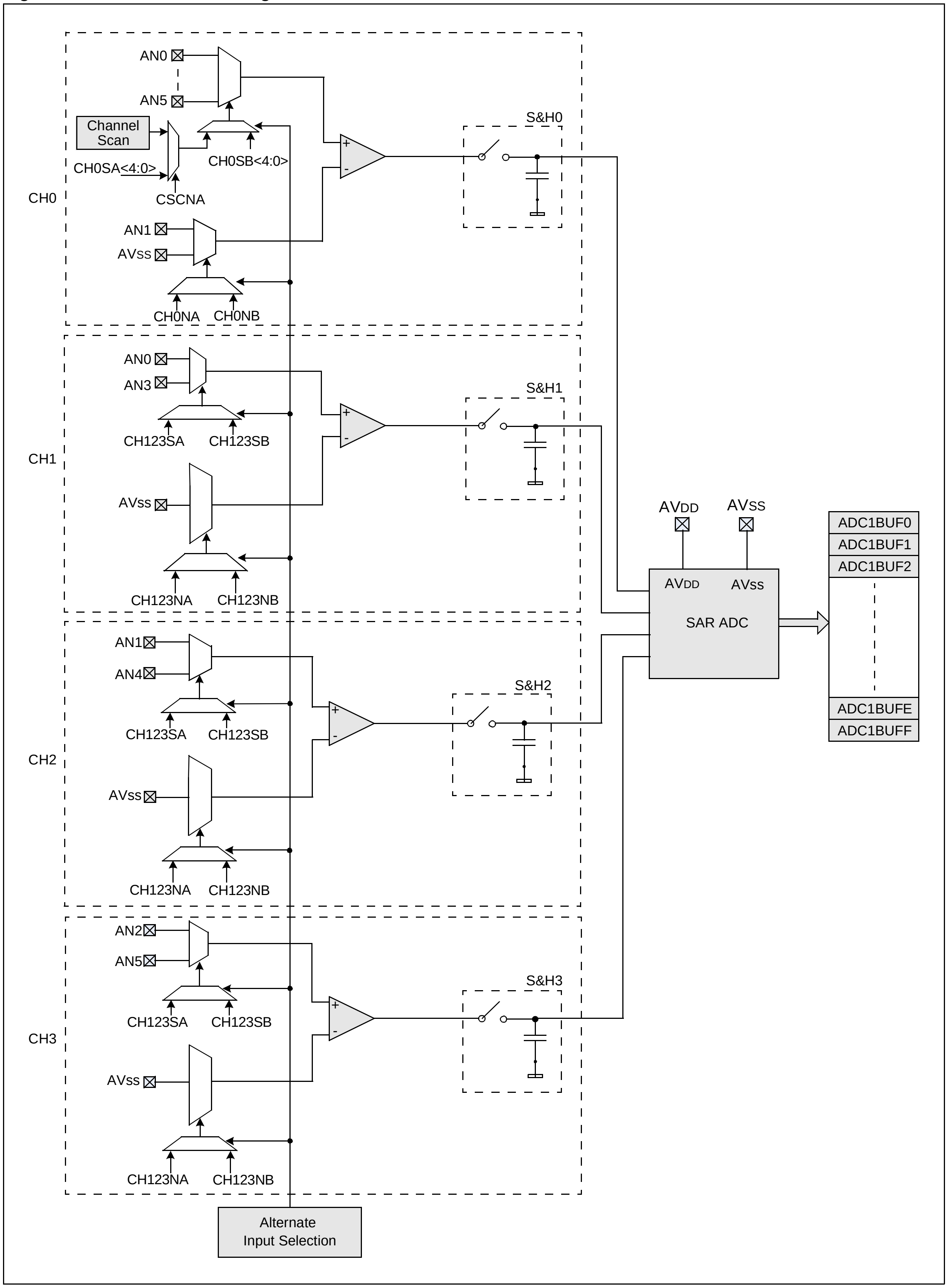
Channel Scan mode can be enabled for the CH0 S&H amplifier. Any subset of the analog inputs (AN0 to AN5 based on availability) can be selected by the user application. The selected inputs are converted in ascending order using CH0.

The ADC module supports simultaneous sampling using multiple S&H channels to sample the inputs at the same time, and then performs the conversion for each channel sequentially. By default, the multiple channels are sampled and converted sequentially.

The ADC module is connected to a 16-word result buffer. The ADC result is available in four different numerical formats (see Figure 49-13).

> **Note 1:** A 'y' is used with MUXA and MUXB control bits to specify the S&H channel numbers (y = 0 or 123).
>
> **2:** Depending on a particular device pinout, the ADC can have up to 6 analog input pins, designated AN0 through AN5. The actual number of analog input pins depends on the specific device. For further details, refer to the specific device data sheet.

Figure 49-1: ADC Block Diagram



49.2 CONTROL REGISTERS

The ADC module has seven Control and Status registers. These registers are:

- **ADxCON1: ADCx Control Register 1**
- **ADxCON2: ADCx Control Register 2**
- **ADxCON3: ADCx Control Register 3**
- **ADxCHS123: ADCx Input Channel 1, 2, 3 Select Register**
- **ADxCHS0: ADCx Input Channel 0 Select Register**
- **ADxCSSL: ADCx Input Scan Select Register Low**
- **ADxPCFGL: ADCx Port Configuration Register Low(1)**

The ADxCON1, ADxCON2 and ADxCON3 registers control the operation of the ADC module. The ADxCHS123 and ADxCHS0 registers select the input pins to be connected to the S&H amplifiers. The ADxCSSH/L registers select inputs to be sequentially scanned. The ADxPCFGH/L registers configure the analog input pins as analog inputs or as digital I/O.

49.2.1 ADC Result Buffer

The ADC module contains a 16-word dual port RAM, to buffer the results. The 16 buffer locations are referred to as ADC1BUF0, ADC1BUF1, ADC1BUF2, ..., ADC1BUFE and ADC1BUFF.

Note:	After a device Reset, the ADC buffer register(s) will contain unknown data.

Register 49-1: ADxCON1: ADCx Control Register 1

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

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

Legend:	HC = Hardware Clearable bit	HS = Hardware Settable bit	C = Clearable bit
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 **ADON:** ADC Operating Mode bit
1 = ADC module is operating
0 = ADC 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 12-10 **Unimplemented:** Read as ‘0’

bit 9-8 **FORM<1:0>:** Data Output Format bits
11 = Signed fractional (DOUT = `sddd dddd dd00 0000`, where s = sign, d = data)
10 = Fractional (DOUT = `dddd dddd dd00 0000`)
01 = Signed integer (DOUT = `ssss sssd dddd dddd`, where s = sign, d = data)
00 = Integer (DOUT = `0000 00dd dddd dddd`)

bit 7-5 **SSRC<2:0>:** Sample Clock Source Select bits
111 = Internal counter ends sampling and starts conversion (auto-convert)
110 = Reserved
101 = Reserved
100 = Reserved
011 = Motor control PWM1 interval ends sampling and starts conversion
010 = GP Timer3 compare ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing sample bit ends sampling and starts conversion

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **SIMSAM:** Simultaneous Sample Select bit (only applicable when CHPS<1:0> = 01 or 1x)
1 = Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS<1:0> = 1x) or samples CH0 and CH1 simultaneously (when CHPS<1:0> = 01)
0 = Samples multiple channels individually in sequence

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

bit 1 **SAMP:** ADC Sample Enable bit
1 = ADC S&H amplifiers are sampling
0 = ADC S&H amplifiers are holding
If ASAM = 0, software can write ‘1’ to begin sampling. Automatically set by hardware if ASAM = 1.
If SSRC = 000, software can write ‘0’ to end sampling and start conversion. If SSRC ≠ 000, automatically cleared by hardware to end sampling and start conversion.

bit 0 **DONE:** ADC Conversion Status bit
1 = ADC conversion cycle is completed
0 = ADC conversion not started or in progress
Automatically set by hardware when the A/D conversion is complete. Software can write ‘0’ to clear the DONE status bit (software not allowed to write ‘1’). Clearing this bit does NOT affect any operation in progress. Automatically cleared by hardware at the start of a new conversion.

Register 49-2: ADxCON2: ADCx Control Register 2

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
VCFG2	VCFG1	VCFG0	—	—	CSCNA	CHPS1	CHPS0
bit 15							bit 8

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

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

bit 15-13 **VCFG<2:0>:** Converter Voltage Reference Configuration bits

VREFH = AVDD
VREFL = AVSS

bit 12-11 **Unimplemented:** Read as ‘0’

bit 10 **CSCNA:** Input Scan Select bit

1 = Scan inputs for CH0+ during Sample A bit
0 = Do not scan inputs

bit 9-8 **CHPS<1:0>:** Channel Select bits

When AD12B = 1, CHPS<1:0> is: U-0, Unimplemented, Read as ‘0’
1x =Converts CH0, CH1, CH2 and CH3
01 =Converts CH0 and CH1
00 =Converts CH0

bit 7 **BUFS:** Buffer Fill Status bit (only valid when BUFM = 1)

1 = ADC is currently filling the second half of the buffer. The user application should access data in the first half of the buffer.
0 = ADC is currently filling the first half of the buffer. The user application should access data in the second half of the buffer.

bit 6 **Unimplemented:** Read as ‘0’

bit 5-2 **SMPI<3:0>:** Sample and Conversion Operation bits

1111 =ADC interrupt is generated at the completion of every 16th sample/conversion operation
1110 =ADC interrupt is generated at the completion of every 15th sample/conversion operation
•
•
•
0001 =ADC interrupt is generated at the completion of every 2nd sample/conversion operation
0000 =ADC interrupt is generated at the completion of every sample/conversion operation

bit 1 **BUFM:** Buffer Fill Mode Select bit

1 = Starts filling the first half of the buffer on the first interrupt and the second half of the buffer on the next interrupt
0 = Always starts filling the buffer from the start address

bit 0 **ALTS:** Alternate Input Sample Mode Select bit

1 = Uses channel input selects for Sample A on first sample and Sample B on next sample
0 = Always uses channel input selects for Sample A

Register 49-3: ADxCON3: ADCx Control Register 3

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC4(1,2)	SAMC3(1,2)	SAMC2(1,2)	SAMC1(1,2)	SAMC0(1,2)
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:** ADC Conversion Clock Source bit
1 = ADC internal RC clock
0 = Clock derived from system clock

bit 14-13 **Unimplemented:** Read as ‘0’

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits(1,2)
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** ADC Conversion Clock Select bits
11111111 = Reserved
•
•
•
01000000 = Reserved
00111111 = TCY • (ADCS<7:0> + 1) = 64 • TCY = TAD
•
•
•
00000010 = TCY • (ADCS<7:0> + 1) = 3 • TCY = TAD
00000001 = TCY • (ADCS<7:0> + 1) = 2 • TCY = TAD
00000000 = TCY • (ADCS<7:0> + 1) = 1 • TCY = TAD

Note 1: This bit is only used when the SSRC<2:0> bits (ADxCON1<7:5>) = 111.

2: If SSRC<2:0> = 111, the SAMC bit should be set to at least ‘1’ when using one S&H channel or using simultaneous sampling. When using multiple S&H channels with sequential sampling, the SAMC bit should be set to ‘0’ for the fastest possible conversion rate.

Register 49-4: ADxCHS123: ADCx Input Channel 1, 2, 3 Select Register

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NB1	CH123NB0	CH123SB
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NA1	CH123NA0	CH123SA
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 15-11	**Unimplemented:** Read as ‘0’
bit 10-9	**CH123NB<1:0>:** Channel 1, 2, 3 Negative Input Select for Sample B bits 1x = Reserved 0x = CH1, CH2, CH3 negative input is AVss
bit 8	**CH123SB:** Channel 1, 2, 3 Positive Input Select for Sample B bit 1 = CH1 positive input is AN3; CH2 positive input is AN4; CH3 positive input is AN5 0 = CH1 positive input is AN0; CH2 positive input is AN1; CH3 positive input is AN2
bit 7-3	**Unimplemented:** Read as ‘0’
bit 2-1	**CH123NA<1:0>:** Channel 1, 2, 3 Negative Input Select for Sample A bits 1x = Reserved 0x = CH1, CH2, CH3 negative input is AVss
bit 0	**CH123SA:** Channel 1, 2, 3 Positive Input Select for Sample A bit 1 = CH1 positive input is AN3; CH2 positive input is AN4; CH3 positive input is AN5 0 = CH1 positive input is AN0; CH2 positive input is AN1; CH3 positive input is AN2

Register 49-5: ADxCHS0: ADCx Input Channel 0 Select Register

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	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	**CH0NB:** Channel 0 Negative Input Select for Sample B bit Same definition as bit 7.
bit 14-13	**Unimplemented:** Read as ‘0’
bit 12-8	**CH0SB<4:0>:** Channel 0 Positive Input Select for Sample B bits Same definition as bit<4:0>.
bit 7	**CH0NA:** Channel 0 Negative Input Select for Sample A bit 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is AVss
bit 6-5	**Unimplemented:** Read as ‘0’
bit 4-0	**CH0SA<4:0>:** Channel 0 Positive Input Select for Sample A bits(1) 00101 = Channel 0 positive input is AN5 00100 = Channel 0 positive input is AN4 00011 = Channel 0 positive input is AN3 00010 = Channel 0 positive input is AN2 00001 = Channel 0 positive input is AN1 00000 = Channel 0 positive input is AN0

Note 1: These bits have no effect when the CSCNA bit (ADxCON2<10>) = 1.

Register 49-6: ADxCSSL: ADCx Input Scan Select Register Low

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	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<5:0>: ADC Input Scan Selection bits(1)**
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

Note 1: Inputs selected for scan without a corresponding input on the device convert AVss.

Register 49-7: ADxPCFGL: ADCx Port Configuration Register Low(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PCFG5(1)	PCFG4(1)	PCFG3(1)	PCFG2(1)	PCFG1(1)	PCFG0(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 **PCFG<5:0>:** ADC Port Configuration Control bits(1)
`1` = Port pin in Digital mode; port read input enabled; ADC input multiplexer connected to AVss
`0` = Port pin in Analog mode, port read input disabled; ADC samples pin voltage

Note 1: PCFG bits are ignored on ports without a corresponding input on the device.

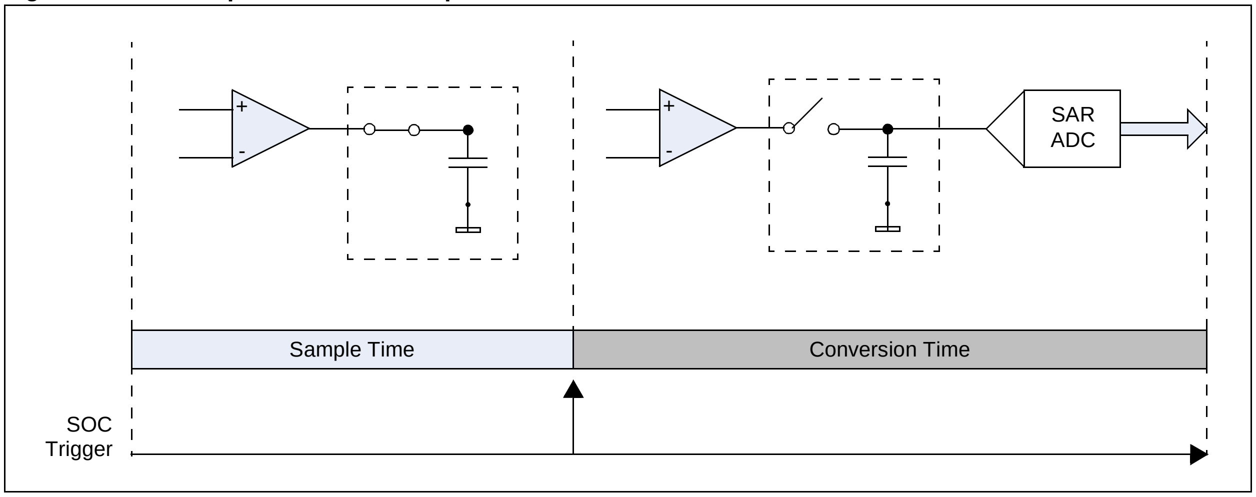
49.3 OVERVIEW OF SAMPLE AND CONVERSION SEQUENCE

Figure 49-2 illustrates that the A/D conversion is a three-step process:

1. The input voltage signal is connected to the sample capacitor.
2. The sample capacitor is disconnected from the input.
3. The stored voltage is converted to equivalent digital bits.

The two distinct phases, sample and conversion, are independently controlled.

Figure 49-2: Sample Conversion Sequence



49.3.1 Sample Time

Sample time is when the selected analog input is connected to the sample capacitor. There is a minimum sample time to ensure that the S&H amplifier provides a desired accuracy for the A/D conversion (see **Section 49.10 “A/D Sampling Requirements”**).

Note:	The ADC module requires a finite number of A/D clock cycles to start conversion after receiving a conversion trigger or stopping the sampling process. Refer to the TPCS parameter in the **“Electrical Characteristics”** chapter of the specific device data sheet for further details.

The sampling phase can be set up to start automatically upon conversion or by manually setting the Sample bit (SAMP) in the ADC Control Register 1 (ADxCON1<1>). The sampling phase is controlled by the Auto-Sample bit (ASAM) in the ADC Control Register 1 (ADxCON1<2>). Table 49-1 lists the options selected by the specific bit configuration.

Table 49-1: Start of Sampling Selection

ASAM	Start of Sampling Selection
0	Manual sampling
1	Automatic sampling

If automatic sampling is enabled, the Sampling Time (TSMP) taken by the ADC module is equal to the number of TAD cycles defined by the SAMC<4:0> bits (ADxCON3<12:8>), as shown by Equation 49-1.

Equation 49-1: Sampling Time Calculation

$$T_{SMP} = SAMC\langle 4:0\rangle \bullet T_{AD}$$

If manual sampling is desired, the user software must provide sufficient time to ensure adequate sampling time.

49.3.2 Conversion Time

The Start-of-Conversion (SOC) trigger ends the sampling time and begins an A/D conversion. During the conversion period, the sample capacitor is disconnected from the multiplexer and the stored voltage is converted to equivalent digital bits. The 10-bit conversion time is shown in Equation 49-2. The sum of the sample time and the A/D conversion time provides the total conversion time.

For correct A/D conversion, the A/D conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time. Refer to the **"Electrical Characteristics"** chapter of the specific device data sheet for the minimum T_{AD} specifications for 10-bit mode.

Equation 49-2: 10-Bit ADC Conversion Time

$$T_{CONV} = 12 \cdot T_{AD}$$

Where:

T_{CONV} = Conversion Time
T_{AD} = ADC Clock Period

The SOC can be triggered by a variety of hardware sources or controlled manually in user software. The trigger source to initiate conversion is selected by the SOC Trigger Source Select bits (SSRC<2:0>) in the ADC Control Register 1 (ADxCON1<7:5>). Table 49-2 lists the conversion trigger source selection for different bit settings.

Table 49-2: SOC Trigger Selection

SSRC<2:0>(1)	SOC Trigger Source
`000`	Manual Trigger
`001`	External Interrupt Trigger (INT0)
`010`	Timer Interrupt Trigger
`011`	Motor Control PWM Special Event Trigger
`100`	Timer Interrupt Trigger
`111`	Automatic Trigger

Note 1: The SSRC<2:0> selection bits should not be changed when the ADC module is enabled.

Table 49-3 lists the sample conversion sequence with different sample and conversion phase selections.

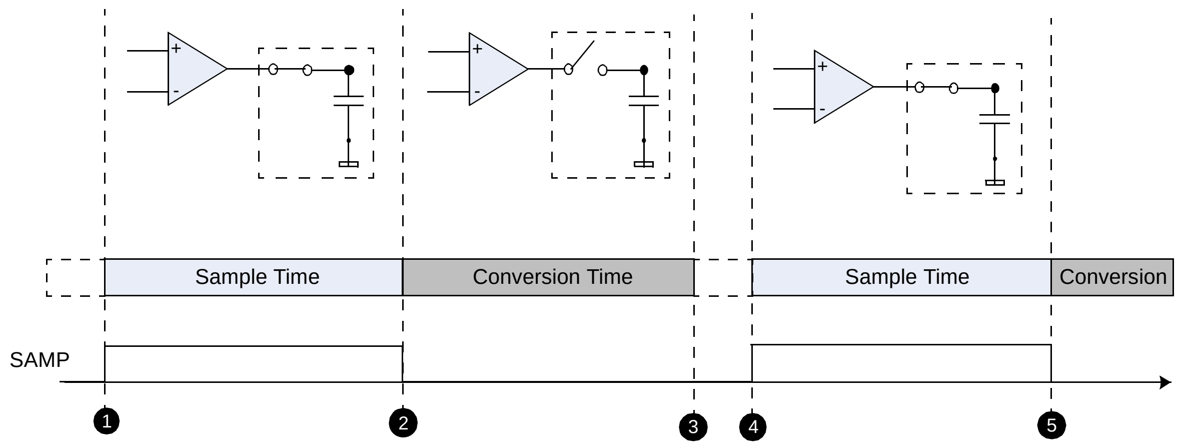
Table 49-3: Sample Conversion Sequence Selection

ASAM	SSRC<2:0>	Description
0	000	Manual Sample and Manual Conversion Sequence
0	111	Manual Sample and Automatic Conversion Sequence
0	001 010 011 100	Manual Sample and Triggered Conversion Sequence
1	000	Automatic Sample and Manual Conversion Sequence
1	111	Automatic Sample and Automatic Conversion Sequence
1	001 010 011 100	Automatic Sample and Triggered Conversion Sequence

49.3.3 Manual Sample and Manual Conversion Sequence

In the Manual Sample and Manual Conversion Sequence, setting the Sample bit (SAMP) in the ADC Control Register 1 (ADxCON1<1>) initiates sampling, and clearing the SAMP bit terminates sampling and starts conversion (see Figure 49-3). The user application must time the setting and clearing of the SAMP bit to ensure adequate sampling time for the input signal. Example 49-1 illustrates a code sequence for Manual Sample and Manual Conversion.

Figure 49-3: Manual Sample and Manual Conversion Sequence



Note 1: Sampling is started by setting the SAMP bit in software.
2: Conversion is started by clearing the SAMP bit in software.
3: Conversion is complete.
4: Sampling is started by setting the SAMP bit in software.
5: Conversion is started by clearing the SAMP bit in software.

Example 49-1: Code Sequence for Manual Sample and Manual Conversion

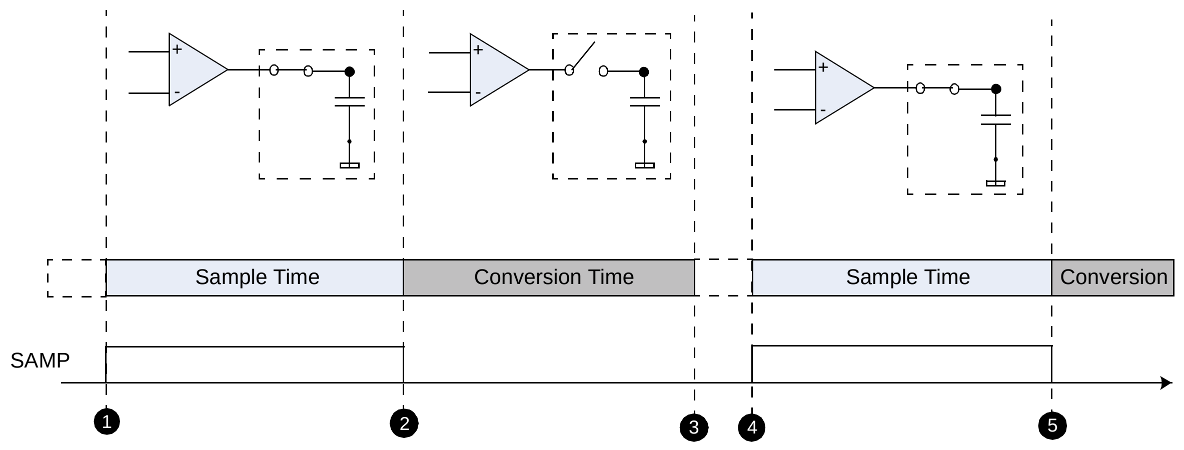
```
AD1CON1bits.SAMP = 1;       // Start sampling
DelayUs(10);                // Wait for sampling time (10us)
AD1CON1bits.SAMP = 0;       // Start the conversion
while (!AD1CON1bits.DONE); // Wait for the conversion to complete
ADCValue = ADC1BUF0;        // Read the conversion result
```

Note: Due to the internal delay within the ADC module, the SAMP bit will read as ‘0’ to the user software, after a small interval of time, after the conversion has already begun. In general, the time interval will be 2 TCY.

49.3.4 Automatic Sample and Manual Conversion Sequence

In the Automatic Sample and Manual Conversion Sequence, sampling starts automatically after conversion of the previous sample. The user application must allocate sufficient time for sampling before clearing the SAMP bit. Clearing the SAMP bit initiates the conversion (see Figure 49-4).

Figure 49-4: Automatic Sample and Manual Conversion Sequence



Note 1: Sampling is started automatically after conversion completion of the previous sample.
2: Conversion is started by clearing the SAMP bit in software.
3: Conversion is complete.
4: Sampling is started automatically after conversion completion of the previous sample.
5: Conversion is started by clearing the SAMP bit in software.

Example 49-2: Code Sequence for Automatic Sample and Manual Conversion

```
    while (1)                          // Repeat continuously
    {
        DelayNmSec(100);               // Sample for 100 ms
        AD1CON1bits.SAMP = 0;          // Start converting
        while (!AD1CON1bits.DONE);     // Conversion done?
        AD1CON1bits.DONE = 0;          // Clear conversion done status bit
        ADCValue = ADC1BUF0;           // If yes, then get the ADC value
    }                                  // Repeat
```

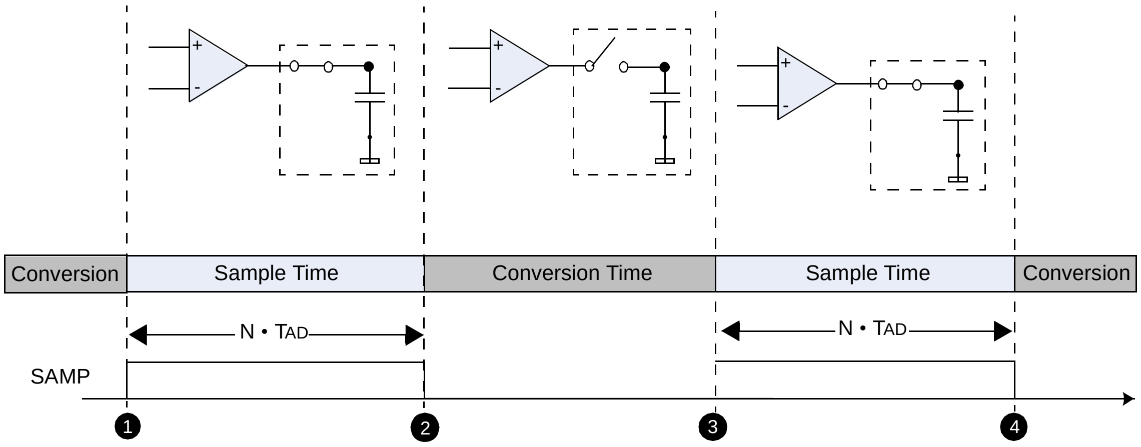
49.3.5 Automatic Sample and Automatic Conversion Sequence

49.3.5.1 CLOCKED CONVERSION TRIGGER

The auto-conversion method provides a more automated process to sample and convert the analog inputs, as shown in Figure 49-5. The sampling period is self-timed and the conversion starts automatically upon termination of a self-timed sampling period. The Auto-Sample Time bits (SAMC<4:0>) in the ADxCON3 register (ADxCON3<12:8>) select 0 to 31 ADC clock cycles (T_{AD}) for sampling period. Refer to the **"Electrical Characteristics"** chapter of the specific device data sheet for a minimum recommended sampling time (SAMC value).

The SSRC<2:0> bits are set to '111' to choose the internal counter as the sample clock source, which ends sampling and starts conversion.

Figure 49-5: Automatic Sample and Automatic Conversion Sequence



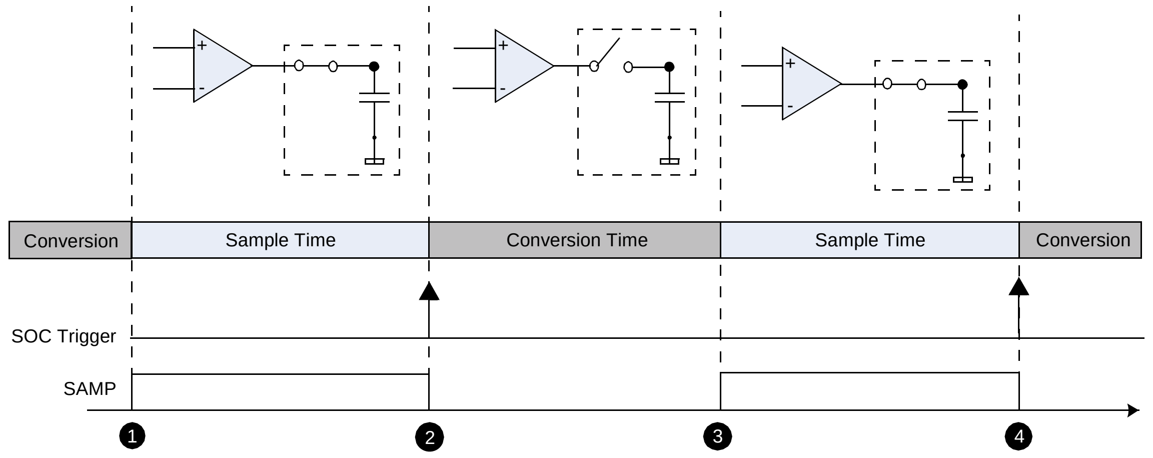
Note 1: Sampling starts automatically after conversion.
2: Conversion starts automatically upon termination of self-timed sampling period.
3: Sampling starts automatically after conversion.
4: Conversion starts automatically upon termination of self-timed sampling period.

49.3.5.2 EXTERNAL CONVERSION TRIGGER

In an automatic sample and triggered conversion sequence, the sampling starts automatically after conversion and the conversion is started upon trigger event from the selected peripheral, as shown in Figure 49-6. This allows ADC conversion to be synchronized with the internal or external events. The external conversion trigger is selected by configuring the SSRC<2:0> bits to '`001`', '`010`' or '`011`'. See **Section 49.4.5 "Conversion Trigger Sources"** for various external conversion trigger sources.

The ASAM bit should not be modified while the A/D Converter is turned on. If automatic sampling is desired, the ASAM bit must be set before turning the module on. The A/D module does take some amount of time to stabilize (see the TPDU parameter in the specific device data sheet); therefore, if automatic sampling is enabled, there is no assurance that the first ADC result will be correct until the ADC module stabilizes. It may be necessary to discard the first ADC result depending on the A/D clock speed.

Figure 49-6: Automatic Sample and Triggered Conversion Sequence



Note 1: Sampling starts automatically after conversion.
2: Conversion starts upon trigger event.
3: Sampling starts automatically after conversion.
4: Conversion starts upon trigger event.

49.3.6 Multi-Channel Sample Conversion Sequence

Multi-channel A/D Converters typically convert each input channel sequentially using an input multiplexer. Simultaneously sampling multiple signals ensures that the snapshot of the analog inputs occurs at precisely the same time for all inputs, as shown in Figure 49-7.

Certain applications require simultaneous sampling, especially when phase information exists between different channels. Sequential sampling takes a snapshot of each analog input, just before conversion starts on that input, as shown in Figure 49-7. The sampling of multiple inputs is not correlated. For example, motor control and power monitoring require voltage and current measurements and the phase angle between them.

Figure 49-7: Simultaneous and Sequential Sampling

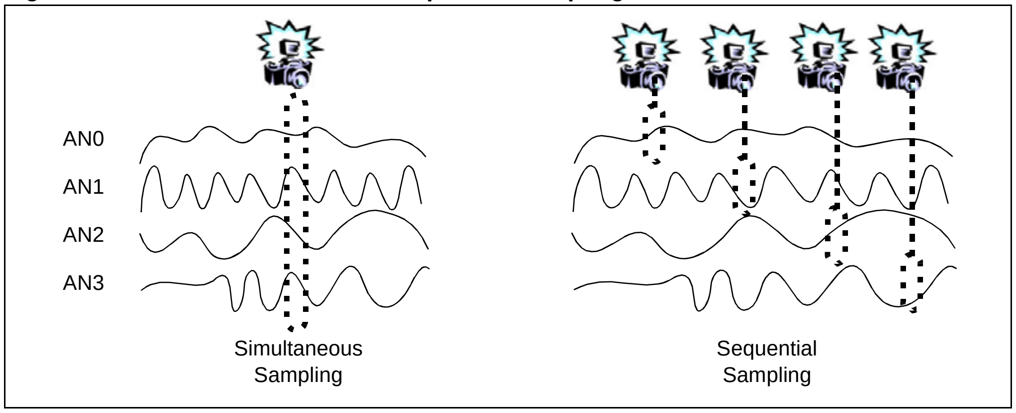


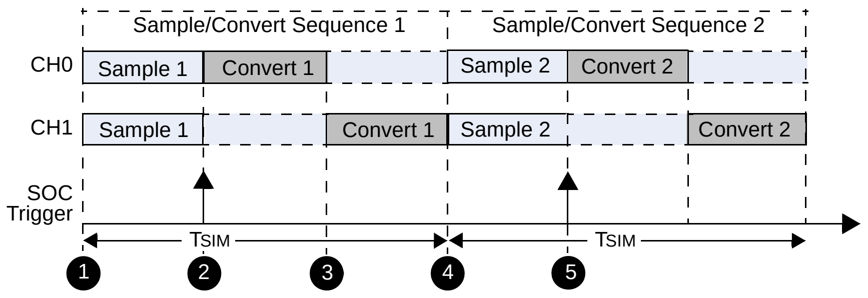
Figure 49-8 and Figure 49-9 illustrate that the ADC module supports simultaneous sampling, using two S&H or four S&H channels to sample the inputs at the same instant, and then performs the conversion for each channel sequentially.

The Simultaneous Sampling mode is selected by setting Simultaneous Sampling bit (SIMSAM) in the ADC Control Register 1 (ADxCON1<3>). By default, the channels are sampled and converted sequentially. Table 49-4 lists the options selected by a specific bit configuration. The CHPS<1:0> bits determine the channels to be sampled, either sequentially or simultaneously.

Table 49-4: Start of Sampling Selection

SIMSAM	Sampling Mode
0	Sequential Sampling
1	Simultaneous Sampling

Figure 49-8: 2-Channel Simultaneous Sampling (ASAM = 1)



Note 1: The CH0-CH1 input multiplexer selects the analog input for sampling. The selected analog input is connected to the sample capacitor.

2: On the SOC Trigger, CH0-CH1 samples capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0 is converted to equivalent digital bits.

3: The analog voltage captured in CH1 is converted to equivalent digital bits.

4: The CH0-CH1 input multiplexer selects the next analog input for sampling. The selected analog input is connected to the sample capacitor.

5: On SOC Trigger, CH0-CH1 samples capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0 is converted to equivalent digital bits.

For simultaneous sampling, the total time taken to sample and convert the channels is shown by Equation 49-3.

Equation 49-3: Channel Sample and Conversion Total Time, Simultaneous Sampling Selected

$$T_{SIM} = T_{SMP} + (M \bullet T_{CONV})$$

Where:

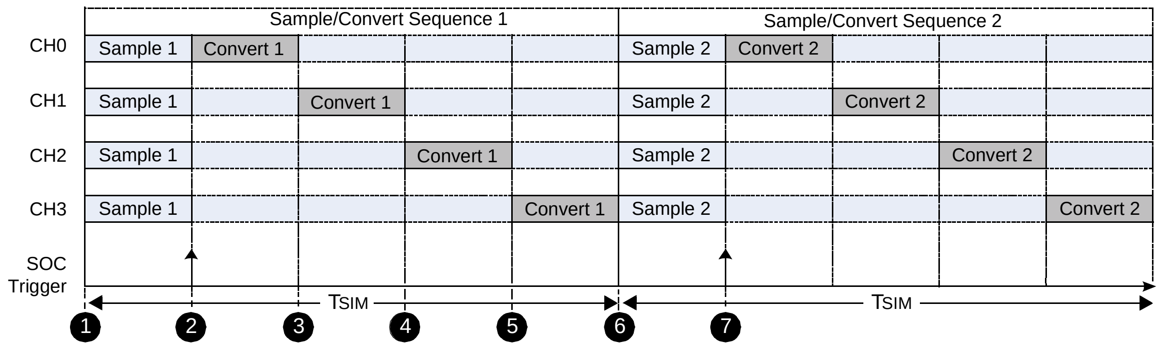
T_{SIM} = Total time to sample and convert multiple channels with simultaneous sampling.

T_{SMP} = Sampling time (see Equation 49-1)

T_{CONV} = Conversion time (see Equation 49-2)

M = Number of channels selected by the CHPS<1:0> bits.

Figure 49-9: 4-Channel Simultaneous Sampling



Note 1: The CH0-CH3 input multiplexer selects an analog input for sampling. The selected analog input is connected to the sample capacitor.

2: On SOC Trigger, CH0-CH3 sample capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0 is converted to equivalent digital bits.

3: The analog voltage captured in CH1 is converted to equivalent digital bits.

4: The analog voltage captured in CH2 is converted to equivalent digital bits.

5: The analog voltage captured in CH3 is converted to equivalent digital bits.

6: The CH0-CH3 input multiplexer selects the next analog input for sampling. The selected analog input is connected to the sample capacitor.

7: On SOC Trigger, CH0-CH3 sample capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0 is converted to equivalent digital bits.

Figure 49-10 and Figure 49-11 illustrate that by default, the multiple channels are sampled and converted sequentially.

For sequential sampling, the total time taken to sample and convert the channels is shown in Equation 49-4.

Equation 49-4: Channel Sample and Conversion Total Time, Sequential Sampling Selected

When $T_{SMP} < T_{CONV}$:

$$T_{SEQ} = M \bullet T_{CONV} \quad \text{(if } M > 1\text{)}$$
$$T_{SEQ} = T_{SMP} + T_{CONV} \quad \text{(if } M = 1\text{)}$$

Where:

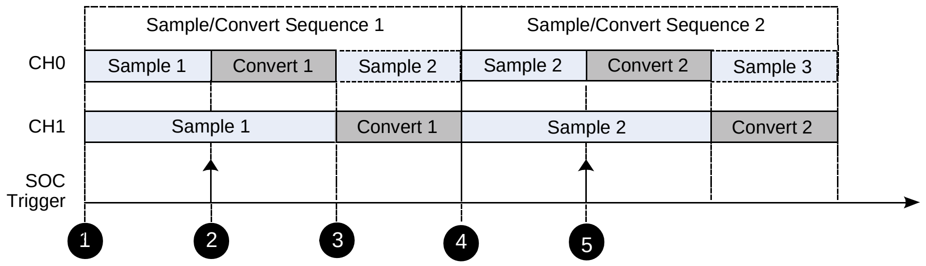
T_{SEQ} = Total time to sample and convert multiple channels with sequential sampling.

T_{CONV} = Conversion time (see Equation 49-2).

T_{SMP} = Sampling time (see Equation 49-1).

M = Number of channels selected by the CHPS<1:0> bits.

Figure 49-10: 2-Channel Sequential Sampling (ASAM = 1)



Note 1: The CH0-CH1 input multiplexer selects an analog input for sampling. The selected analog input is connected to the sample capacitor.

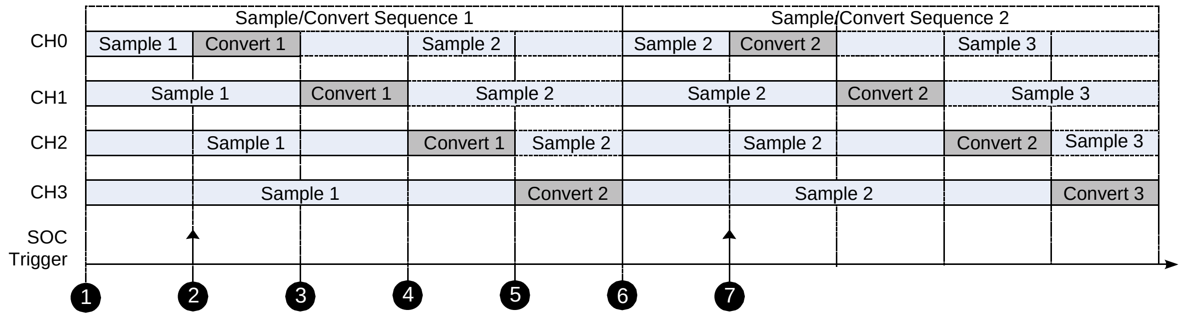
2: On SOC Trigger, CH0 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH0 is converted to equivalent digital bits.

3: The CH0 multiplexer output is connected to the sample capacitor after conversion. CH1 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH1 is converted to equivalent digital bits.

4: The CH1 multiplexer output is connected to sample the capacitor after conversion. The CH0-CH1 input multiplexer selects the next analog input for sampling.

5: On SOC Trigger, CH0 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH0 is converted to equivalent digital bits.

Figure 49-11: 4-Channel Sequential Sampling



Note 1: The CH0-CH3 input multiplexer selects an analog input for sampling. The selected analog input is connected to the sample capacitor.

2: On SOC Trigger, CH0 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH0 is converted to equivalent digital bits.

3: The CH0 multiplexer output is connected to the sample capacitor after conversion. CH1 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH1 is converted to equivalent digital bits.

4: The CH1 multiplexer output is connected to the sample capacitor after conversion. CH2 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH2 is converted to equivalent digital bits.

5: The CH2 multiplexer output is connected to the sample capacitor after conversion. CH3 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH3 is converted to equivalent digital bits.

6: The CH3 multiplexer output is connected to the sample capacitor after conversion. CH0-CH3 input multiplexer selects the next analog input for sampling.

7: On SOC Trigger, CH0 sample capacitor is disconnected from the multiplexer to hold the input voltage constant during conversion. The analog value captured in CH0 is converted to equivalent digital bits.

49.4 ADC CONFIGURATION

49.4.1 ADC Channel Selection

The user application can select 1-Channel (CH0), 2-Channel (CH0, CH1) or 4-Channel mode (CH0-CH3) using the Channel Select bits (CHPS<1:0>) in the ADC Control Register 2 (ADxCON2<9:8>).

Table 49-5: 10-Bit ADC Channel Selection

CHPS<1:0>	Channel Selection
`00`	CH0
`01`	Dual Channel (CH0, CH1)
`1x`	Multi-Channel (CH0-CH3)

49.4.2 ADC Clock Selection

The ADC module can be clocked from the instruction cycle clock (TCY) or by using the dedicated internal RC clock (see Figure 49-12). When using the instruction cycle clock, a clock divider drives the instruction cycle clock and allows a lower frequency to be chosen. The clock divider is controlled by the ADC Conversion Clock Select bits (ADCS<7:0>) in the ADC Control Register 3 (ADxCON3<7:0>), which allows 64 settings, from 1:1 to 1:64, to be chosen.

For correct A/D conversion, the ADC clock period (TAD) must be a minimum of 75 ns.

Equation 49-5 shows the ADC clock period (TAD) as a function of the ADCS control bits and the device instruction cycle clock period, TCY.

Equation 49-5: ADC Clock Period

If ADRC = 0

$$\text{ADC Clock Period } (T_{AD}) = T_{CY} \cdot (\text{ADCS} + 1)$$

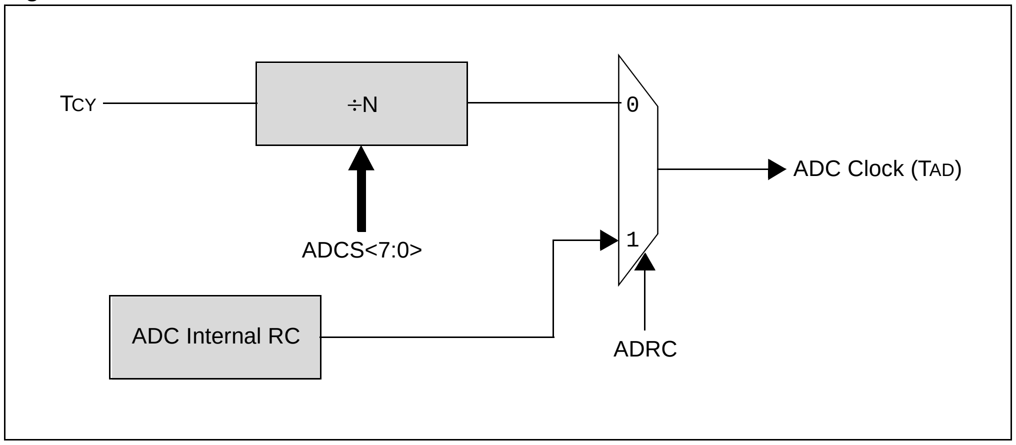
If ADRC = 1

$$\text{ADC Clock Period } (T_{AD}) = T_{ADRC}$$

The ADC module has a dedicated internal RC clock source that can be used to perform conversions. The internal RC clock source is used when A/D conversions are performed while the device is in Sleep mode. The internal RC oscillator is selected by setting the ADC Conversion Clock Source bit (ADRC) in the ADC Control Register 3 (ADxCON3<15>). When the ADRC bit is set, the ADCS<7:0> bits have no effect on the ADC operation.

Note:	Refer to the specific device data sheet for ADRC frequency specifications.

Figure 49-12: ADC Clock Generation



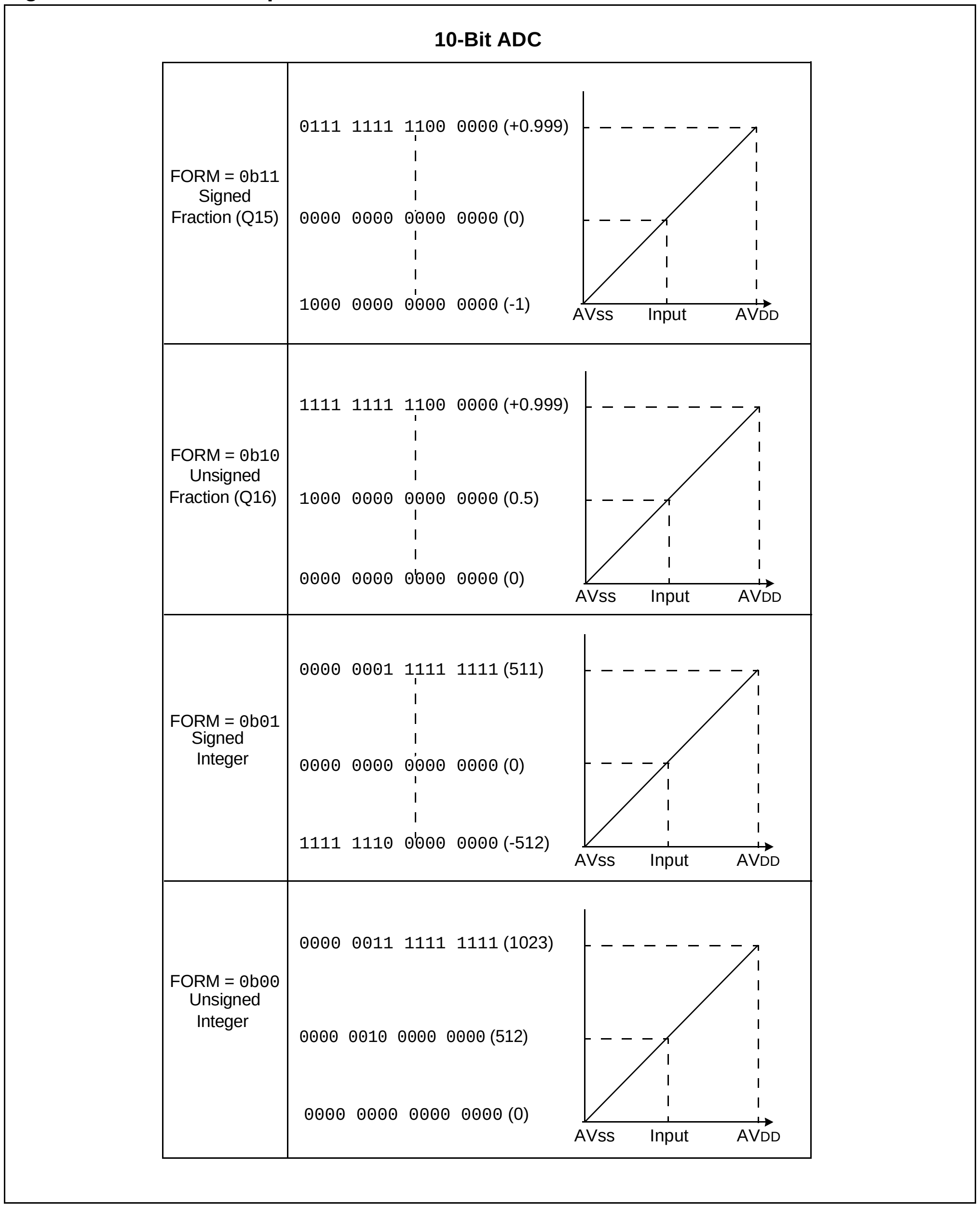
49.4.3 Output Data Format Selection

Figure 49-13 illustrates that the ADC result is available in four different numerical formats. The Data Output Format bits (FORM<1:0>) in the ADC Control Register 1 (ADxCON1<9:8>) select the output data format. Table 49-6 lists the ADC output format for different bit settings.

Table 49-6: Voltage Reference Selection

FORM<1:0>	Data Information Selection
`11`	Signed Fractional Format
`10`	Unsigned Fractional format
`01`	Signed Integer format
`00`	Unsigned Integer format

Figure 49-13: ADC Output Format



49.4.4 Sample and Conversion Operation (SMPI) Bits

The SMPI<3:0> bits are referred to as the Number of Samples Per Interrupt Select bits.

An interrupt can be generated at the end of each sample/convert sequence, or after multiple sample/convert sequences, as determined by the value of the SMPI<3:0> bits. The number of sample/convert sequences between interrupts can vary between 1 and 16. The total number of conversion results between interrupts is the product of the number of channels per sample, created by the CHPS<1:0> bits, and the value of the SMPI<3:0> bits. See **Section 49.5 "ADC Interrupt Generation"** for the SMPI values for various sampling modes.

49.4.5 Conversion Trigger Sources

It is often desirable to synchronize the end of sampling and the start of conversion with some other timed event. The ADC module can use one of the following sources as a conversion trigger:

- External Interrupt Trigger (INT0 only)
- Timer Interrupt Trigger
- Motor Control PWM Special Event Trigger (PIC24F Motor Control Devices Only)

49.4.5.1 EXTERNAL INTERRUPT TRIGGER (INT0 ONLY)

When SSRC<2:0> = `001`, the A/D conversion is triggered by an active transition on the INT0 pin. The INT0 pin can be programmed for either a rising edge input or a falling edge input.

49.4.5.2 TIMER INTERRUPT TRIGGER

This ADC Module Trigger mode is configured by setting SSRC<2:0> = `010`. TMR3 can be used to trigger the start of the A/D conversion when a match occurs between the 16-bit Timer Count register (TMR3) and the 16-bit Timer Period register (PR3).

49.4.5.3 MOTOR CONTROL PWM SPECIAL EVENT TRIGGER

The PWM module has an event trigger that allows A/D conversions to be synchronized to the PWM time base. When SSRC<2:0> = `011`, the A/D sampling and conversion times occur at any user programmable point within the PWM period. The Special Event Trigger allows the user to minimize the delay between the time when the A/D conversion results are acquired and the time when the duty cycle value is updated.

The application should set the ASAM bit in order to ensure that the ADC module has sampled the input sufficiently before the next conversion trigger arrives.

49.4.6 Configuring Analog Port Pins

The Analog/Digital Pin Configuration register (ADxPCFGL) specifies the input condition of the device pins used as analog inputs. Along with the Data Direction register (TRISx) in the Parallel I/O Port module, these registers control the operation of the ADC pins.

A pin is configured as an analog input when the corresponding PCFGn bit (ADxPCFGL<n>) is clear. The ADxPCFGL register is cleared at Reset, causing the ADC input pins to be configured for analog input by default at Reset.

When configured for analog input, the associated port I/O digital input buffer is disabled so that it does not consume current.

The port pins that are desired as analog inputs must have their corresponding TRIS bit set, specifying the port input. If the I/O pin associated with an A/D input is configured as an output, the TRIS bit is cleared and the digital output level (V_{OH} or V_{OL}) of the port is converted. After a device Reset, all TRIS bits are set.

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

Note 1: When the ADC Port register is read, any pin configured as an analog input reads as a '`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 specification.

49.4.7 Enabling the ADC Module

When the ADON bit (ADxCON1<15>) is '1', the module is in active mode and is fully powered and functional.

When ADON is '0', the module is disabled. The digital and analog portions of the circuit are turned off for maximum current savings.

To return to the active mode from the off mode, the user application must wait for the analog stages to stabilize. For the stabilization time, refer to the **"Electrical Characteristics"** chapter of the specific device data sheet.

Note:	The SSRC<2:0>, SIMSAM, ASAM, CHPS<1:0>, SMPI<3:0>, BUFM and ALTS bits, as well as the ADCON3 and ADCSSL registers, should not be written to while ADON = 1. This would lead to indeterminate results.

49.4.8 Turning the ADC Module Off

Clearing the ADON bit disables the ADC module (stops any scanning, sampling and conversion processes). In this state, the ADC module still consumes some current. Setting the ADxMD bit in the PMD register will disable the ADC module and will stop the ADC clock source, which reduces device current consumption. Note that setting the ADxMD bit, and then clearing the bit, will reset the ADC module registers to their default state. Additionally, any digital pins that share their function with an ADC input pin revert to the analog function. While the ADxMD bit is set, these pins will be set to digital function. In this case, the ADxPCFG bits will not have any effect.

Note:	Clearing the ADON bit during a conversion will abort the current A/D conversion. The ADC buffer will not be updated with the partially completed conversion sample.

49.5 ADC INTERRUPT GENERATION

As conversions are completed, the ADC module writes the results of the conversions into the Analog-to-Digital result buffer. The ADC result buffer is an array of sixteen words, accessed through the SFR space. The user application may attempt to read each Analog-to-Digital conversion result as it is generated. However, this might consume too much CPU time. Generally, to simplify the code, the module fills the buffer with results and generates an interrupt when the buffer is filled. The ADC module supports 16 result buffers. Therefore, the maximum number of conversions per interrupt must not exceed 16.

The number of conversions per ADC interrupt depends on the following parameters, which can vary from one to 16 conversions per interrupt.

- Number of S&H Channels Selected
- Sequential or Simultaneous Sampling
- Samples Convert Sequences Per Interrupt bit (SMPI<3:0>) Settings

Table 49-7 lists the number of conversions per ADC interrupt for different configuration modes.

Table 49-7: Samples Per Interrupt in Alternate Sampling Mode

CHPS<1:0>	SIMSAM	SMPI<3:0>	Conversions/ Interrupt	Description
`00`	`x`	N-1	N	1-Channel mode
`01`	`0`	N-1	N	2-Channel Sequential Sampling mode
`1x`	`0`	N-1	N	4-Channel Sequential Sampling mode
`01`	`1`	N-1	2 • N	2-Channel Simultaneous Sampling mode
`1x`	`1`	N-1	4 • N	4-Channel Simultaneous Sampling mode

Note 1: In 2-Channel Simultaneous Sampling mode, SMPI<3:0> bit settings must be less than eight.

2: In 4-Channel Simultaneous Sampling mode, SMPI<3:0> bit settings must be less than four.

The DONE bit (ADxCON1<0>) is set when an ADC interrupt is generated to indicate completion of a required sample/conversion sequence. This bit is automatically cleared by the hardware at the beginning of the next sample/conversion sequence.

Interrupt generation is based on the SMPI<3:0> and CHPS bits, so the DONE bit is not set after every conversion, but is set when the ADC Interrupt Flag (ADxIF) is set.

49.5.1 Buffer Fill Mode

When the Buffer Fill Mode bit (BUFM) in the ADC Control Register 2 (ADxCON2<1>) is ‘`1`’, the 16-word results buffer is split into two 8-word groups: a lower group (ADC1BUF0 through ADC1BUF7) and an upper group (ADC1BUF8 through ADC1BUFF). The 8-word buffers alternately receive the conversion results after each ADC interrupt event. When the BUFM bit is set, each buffer size is equal to eight. Therefore, the maximum number of conversions per interrupt must not exceed eight.

When the BUFM bit is ‘`0`’, the complete 16-word buffer is used for all conversion sequences. The decision to use the split buffer feature depends on the time 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 taken to sample and convert one channel, the BUFM bit can be ‘`0`’, and up to 16 conversions may be done per interrupt. The application has 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 an ADC interrupt is generated every eight conversions, the processor has the entire time between interrupts to move the eight conversions out of the buffer.

49.5.2 Buffer Fill Status

When the conversion result buffer is split using the BUFM control bit, the BUFS status bit (ADxCON2<7>) indicates half of the buffer that the ADC module is currently writing. If BUFS = `0`, the ADC module 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.

49.6 ANALOG INPUT SELECTION FOR CONVERSION

The ADC module provides a flexible mechanism to select analog inputs for conversion:

- Fixed input selection
- Alternate input selection
- Channel scanning (CH0 only)

49.6.1 Fixed Input Selection

The 10-bit ADC configuration can use up to four S&H channels, designated CH0-CH3. The S&H channels are connected to the analog input pins through the analog multiplexer.

When ALTS = `0`, the CH0SA<4:0>, CH0NA, CH123SA and CH123NA<1:0> bits select the analog inputs.

Table 49-8: Analog Input Selection

		MUXA	
		Control bits	**Analog Inputs**
CH0	+ve	CH0SA<4:0>	AN0 to AN5
	-ve	CH0NA	AVss, AN1
CH1	+ve	CH123SA	AN0, AN3
	-ve	CH123NA<1:0>	AVss
CH2	+ve	CH123SA	AN1, AN4
	-ve	CH123NA<1:0>	AVss
CH3	+ve	CH123SA	AN2, AN5
	-ve	CH123NA<1:0>	AVss

Note: Not all inputs are present on all devices.

All four channels can be enabled in Simultaneous or Sequential Sampling modes by configuring the CHPS bit and the SIMSAM bit.

Example 49-3 shows the code sequence to set up ADC inputs for a 4-channel ADC configuration.

Example 49-3: Code Sequence to Set Up ADC Inputs

```
// Initialize MUXA Input Selection
AD1CHS0bits.CH0SA = 3;     // Select AN3 for CH0 +ve input
AD1CHS0bits.CH0NA = 0;     // Select AVss for CH0 -ve input

AD1CHS123bits.CH123SA=0;   // Select AN0 for CH1 +ve input
                           // Select AN1 for CH2+ve input
                           // Select AN2 for CH3 +ve input
AD1CHS123bits.CH123NA=0;   // Select AVss for CH1/CH2/CH3 -ve inputs
```

49.6.2 Alternate Input Selection Mode

In an Alternate Input Selection mode, the MUXA and MUXB control bits select the channel for conversion. The ADC completes one sweep using the MUXA selection, and then another sweep using the MUXB selection, and then another sweep using the MUXA selection, and so on. The Alternate Input Selection mode is enabled by setting the Alternate Sample bit (ALTS) in the ADC Control Register 2 (ADxCON2<0>).

The analog input multiplexer is controlled by the AD1CHS123 and AD1CHS0 registers. There are two sets of control bits, designated as MUXA (CHySA/CHyNA) and MUXB (CHySB/CHyNB), to select a particular input source for conversion. The MUXB control bits are used in Alternate Input Selection mode.

Table 49-9: Analog Input Selection

		MUXA		MUXB	
		Control bits	**Analog Inputs**	**Control bits**	**Analog Inputs**
CH0	+ve	CH0SA<4:0>	AN0 to AN5	CH0SB<4:0>	AN0 to AN5
	-ve	CH0NA	AVss, AN1	CH0NB	AVss, AN1
CH1	+ve	CH123SA	AN0, AN3	CH123SB	AN0, AN3
	-ve	CH123NA<1:0>	AVss	CH123NB<1:0>	AVss
CH2	+ve	CH123SA	AN1, AN4	CH123SB	AN1, AN4
	-ve	CH123NA<1:0>	AVss	CH123NB<1:0>	AVss
CH3	+ve	CH123SA	AN2, AN5	CH123SB	AN2, AN5
	-ve	CH123NA<1:0>	AVss	CH123NB<1:0>	AVss

Note: Not all inputs are present on all devices.

For Alternate Input Selection mode, an ADC interrupt must be generated after an even number of sample/conversion sequences by programming the Samples Convert Sequences Per Interrupt bits (SMPI<3:0>). Table 49-10 lists the valid SMPI values for Alternate Input Selection mode in different ADC configurations.

Table 49-10: Valid SMPI Values for Alternate Input Selection Mode

CHPS<1:0>	SIMSAM	SMPI<3:0> (Decimal)	Conversions/ Interrupts	Description
00	x	1,3,5,7,9,11,13,15	2,4,6,8,10,12,14,16	1-Channel mode
01	0	3,7,11,15	4,8,12,16	2-Channel Sequential Sampling mode
1x	0	7,15	8,16	4-Channel Sequential Sampling mode
01	1	1,3,5,7	4,8,12,16	2-Channel Simultaneous Sampling mode
1x	1	1,3	8,16	4-Channel Simultaneous Sampling mode

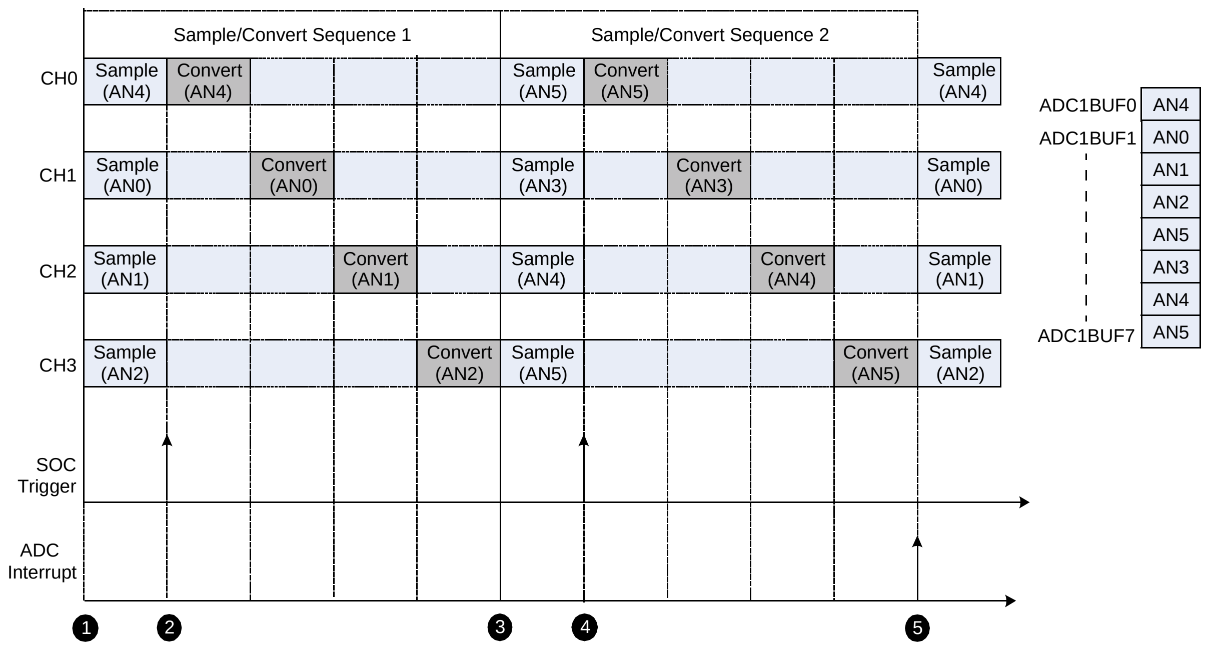
Example 49-4 shows the code sequence to set up the ADC module for Alternate Input Selection mode in the 4-channel simultaneous sampling configuration. Figure 49-14 illustrates the ADC module operation sequence.

Note: On ADC interrupt, the ADC internal logic is initialized to restart the conversion sequence from the beginning.

Example 49-4: Code Sequence to Set Up ADC for Alternate Input Selection Mode for 4-Channel Simultaneous Sampling

```
    AD1CON2bits.CHPS = 3;       // Select 4-channel mode
    AD1CON1bits.SIMSAM = 1;     // Enable Simultaneous Sampling
    AD1CON2bits.ALTS = 1;       // Enable Alternate Input Selection
    AD1CON2bits.SMPI = 1;       // Select 8 conversion between interrupt
    AD1CON1bits.ASAM = 1;       // Enable Automatic Sampling
    AD1CON1bits.SSRC = 2;       // Timer3 generates SOC trigger

    // Initialize MUXA Input Selection
    AD1CHS0bits.CH0SA = 4;      // Select AN4 for CH0 +ve input
    AD1CHS0bits.CH0NA = 0;      // Select AVss for CH0 -ve input
    AD1CHS123bits.CH123SA = 0;  // Select CH1 +ve = AN0, CH2 +ve = AN1, CH3 +ve = AN2
    AD1CHS123bits.CH123NA = 0;  // Select AVss for CH1/CH2/CH3 -ve inputs

    // Initialize MUXB Input Selection
    AD1CHS0bits.CH0SB = 5;      // Select AN5 for CH0 +ve input
    AD1CHS0bits.CH0NB = 0;      // Select AVss for CH0 -ve input

    AD1CHS123bits.CH123SB = 1;  // Select CH1 +ve = AN3, CH2 +ve = AN4, CH3 +ve = AN5
    AD1CHS123bits.CH123NB = 0;  // Select AVss for CH1/CH2/CH3 -ve inputs
```

Figure 49-14: Alternate Input Selection in 4-Channel Simultaneous Sampling Configuration



Note 1: CH0-CH3 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

2: On SOC Trigger, CH0-CH3 sample capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0/CH1/CH2/CH3 is converted sequentially to equivalent digital counts.

3: CH0-CH3 input multiplexer selects analog input for sampling using MUXB control bits (CHySB/CHyNB). The selected analog input is connected to the sample capacitor.

4: On SOC Trigger, CH0-CH3 sample capacitor is disconnected from the multiplexer to simultaneously sample the analog inputs. The analog value captured in CH0/CH1/CH2/CH3 is converted sequentially to equivalent digital counts.

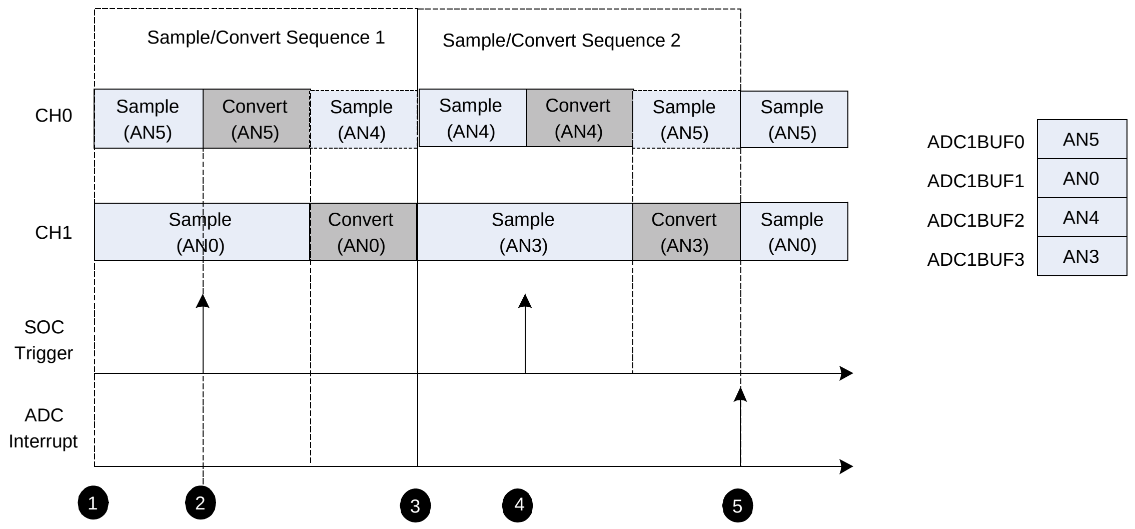
5: ADC interrupt is generated after converting 8 samples. CH0-CH3 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

Example 49-5 shows the code sequence to set up the ADC module for Alternate Input Selection mode in a 2-channel sequential sampling configuration.

Example 49-5: Code Sequence to Set Up ADC for Alternate Input Selection for 2-Channel Sequential Sampling

```
AD1CON2bits.CHPS=1;         // Select 2-channel mode
AD1CON2bits.SMPI = 3;       // Select 4 conversion between interrupt
AD1CON1bits.ASAM = 1;       // Enable Automatic Sampling
AD1CON2bits.ALTS = 1;       // Enable Alternate Input Selection
AD1CON1bits.SIMSAM = 0;     // Enable Sequential Sampling
AD1CON1bits.SSRC = 2;       // Timer3 generates SOC trigger

// Initialize MUXA Input Selection
AD1CHS0bits.CH0SA = 5;      // Select AN5 for CH0 +ve input
AD1CHS0bits.CH0NA = 0;      // Select AVss for CH0 -ve input

AD1CHS123bits.CH123SA=0;    // Select AN0 for CH1 +ve input
AD1CHS123bits.CH123NA=0;    // Select AVss for CH1 -ve inputs

// Initialize MUXB Input Selection
AD1CHS0bits.CH0SB = 4;      // Select AN4 for CH0 +ve input
AD1CHS0bits.CH0NB = 0;      // Select AVss for CH0 -ve input

AD1CHS123bits.CH123SB=1;    // Select AN3 for CH1 +ve input
AD1CHS123bits.CH123NB=0;    // Select AVss for CH1-ve inputs
```

Figure 49-15: Alternate Input Selection in 2-Channel Sequential Sampling Configuration



Note 1: CH0-CH1 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

2: On SOC Trigger, CH0/CH1 inputs are sequentially sampled and converted to equivalent digital counts.

3: CH0-CH1 input multiplexer selects the analog input for sampling using MUXB control bits (CHySB/CHyNB). The selected analog input is connected to the sample capacitor.

4: On SOC Trigger, CH0/CH1 inputs are sequentially sampled and converted to equivalent digital counts.

5: ADC interrupt is generated after converting 4 samples. CH0-CH1 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

49.6.3 Channel Scanning

The ADC module supports the Channel Scan mode using CH0 (S&H Channel 0). The number of inputs scanned is software-selectable. Any subset of the analog inputs from AN0 to AN5 can be selected for conversion. The selected inputs are converted in ascending order. For example, if the input selection includes AN4, AN1 and AN3, the conversion sequence is AN1, AN3 and AN4. The conversion sequence selection is made by programming the Channel Select register (AD1CSSL). A logic '1' in the Channel Select register marks the associated analog input channel for inclusion in the conversion sequence. The Channel Scanning mode is enabled by setting the Channel Scan bit (CSCNA) in the ADC Control Register 2 (ADxCON2<10>). In Channel Scan mode, MUXA software control is ignored and the ADC module sequences through the enabled channels.

For every sample/convert sequence, one analog input is scanned. The ADC interrupt must be generated after all selected channels are scanned. If "N" inputs are enabled for channel scan, an interrupt must be generated after an "N" sample/convert sequence. Table 49-11 lists the SMPI values to scan "N" analog inputs using CH0 in different ADC configurations.

Table 49-11: Conversions Per Interrupt in Channel Scan Mode

CHPS<1:0>	SIMSAM	SMPI<3:0> (Decimal)	Conversions/ Interrupt	Description
00	x	N-1	N	1-Channel mode
01	0	2N-1	2N	2-Channel Sequential Sampling mode
1x	0	4N-1	4N	4-Channel Sequential Sampling mode
01	1	N-1	2N	2-Channel Simultaneous Sampling mode
1x	1	N-1	4N	4-Channel Simultaneous Sampling mode

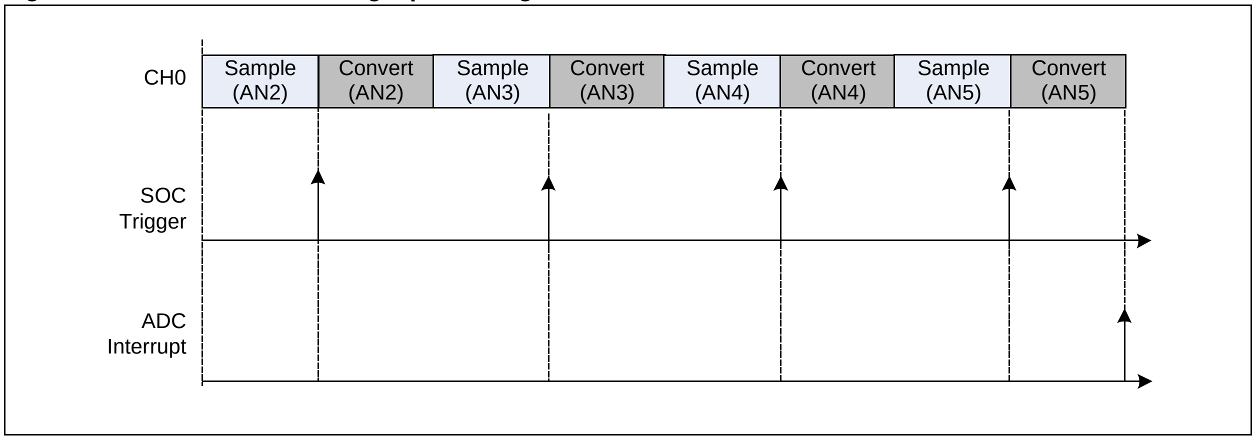
Example 49-6 shows the code sequence to scan four analog inputs using CH0. Figure 49-16 illustrates the ADC operation sequence.

Note: On ADC Interrupt, the ADC internal logic is initialized to restart the conversion sequence from the beginning.

Example 49-6: Code Sequence to Scan Four Analog Inputs Using CH0

```
AD1CON2bits.SMPI = 3;      // Select 4 conversions between interrupt
AD1CHS0bits.ASAM = 1;      // Enable Automatic Sampling
AD1CON2bits.CSCNA = 1;     // Enable Channel Scanning

// Initialize Channel Scan Selection
AD1CSSLbits.CSS2=1;        // Enable AN2 for scan
AD1CSSLbits.CSS3=1;        // Enable AN3 for scan
AD1CSSLbits.CSS4=1;        // Enable AN4 for scan
AD1CSSLbits.CSS5=1;        // Enable AN5 for scan
```

Figure 49-16: Scan Four Analog Inputs Using CH0

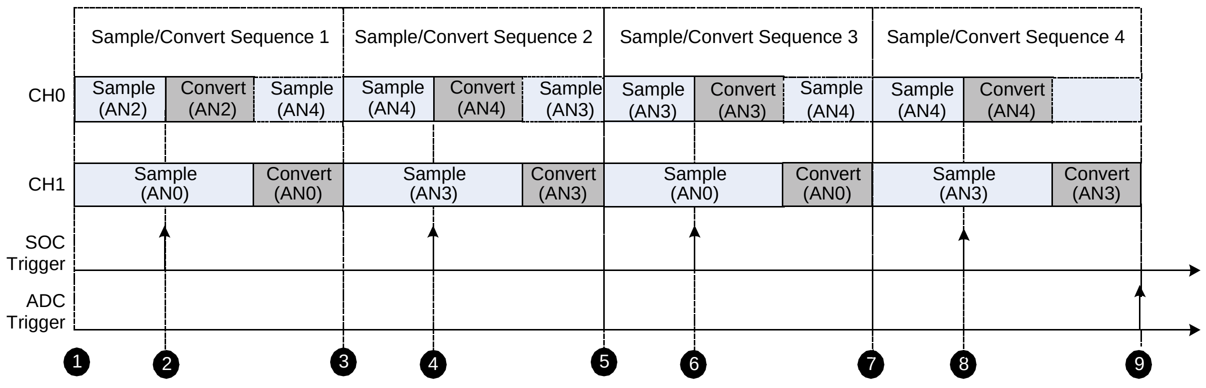


Example 49-7 shows the code sequence to scan two analog inputs using CH0 in a 2-channel alternate input selection configuration. Figure 49-17 illustrates the ADC operation sequence.

Example 49-7: Code Sequence for Channel Scan with Alternate Input Selection (Devices without DMA)

```
AD1CON2bits.CHPS = 1;        // Select 2-channel mode
AD1CON1bits.SIMSAM = 0;      // Enable Sequential Sampling
AD1CON2bits.ALTS = 1;        // Enable Alternate Input Selection
AD1CON2bits.CSCNA = 1;       // Enable Channel Scanning
AD1CON2bits.SMPI = 7;        // Select 8 conversion between interrupt
AD1CON1bits.ASAM = 1;        // Enable Automatic Sampling

// Initialize Channel Scan Selection
AD1CSSLbits.CSS2 = 1;        // Enable AN2 for scan
AD1CSSLbits.CSS3 = 1;        // Enable AN3 for scan

// Initialize MUXA Input Selection
AD1CHS123bits.CH123SA = 0;   // Select AN0 for CH1 +ve input
AD1CHS123bits.CH123NA = 0;   // Select AVss for CH1 -ve inputs

// Initialize MUXB Input Selection
AD1CHS0bits.CH0SB = 4;       // Select AN4 for CH0 +ve input
AD1CHS0bits.CH0NB = 0;       // Select AVss for CH0 -ve inputs

AD1CHS123bits.CH123SB = 1;   // Select AN3 for CH1 +ve input
AD1CHS123bits.CH123NB = 0;   // Select AVss for CH1 -ve inputs
```

Figure 49-17: Channel Scan with Alternate Input Selection (Devices without DMA)



Note 1: CH0 input multiplexer selects analog input for sampling using internally generated control bits (from channel scan logic) instead of MUXA control bits. CH1 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

2: On SOC trigger, CH0-CH1 inputs are sequentially sampled and converted to equivalent digital counts.

3: CH0-CH1 input multiplexer selects the analog input for sampling using MUXB control bits (CHySB/CHyNB). The selected analog input is connected to the sample capacitor.

4: On SOC trigger, CH0-CH1 inputs are sequentially sampled and converted to equivalent digital counts.

5: CH0 input multiplexer selects the analog input for sampling using internally generated control bits (from channel scan logic) instead of MUXA control bits. CH1 input multiplexer selects the analog input for sampling using MUXA control bits (CHySA/CHyNA). The selected analog input is connected to the sample capacitor.

6: On SOC trigger, CH0-CH1 inputs are sequentially sampled and converted to equivalent digital counts.

7: CH0-CH1 input multiplexer selects the analog input for sampling using MUXB control bits (CHySB/CHyNB). The selected analog input is connected to the sample capacitor.

8: On SOC trigger, CH0-CH1 inputs are sequentially sampled and converted to equivalent digital counts.

9: ADC interrupt is generated after converting eight samples.

49.7 ADC CONFIGURATION EXAMPLE

The following steps should be used for performing an A/D conversion:

1. Select the analog conversion clock to match the desired data rate with the processor clock (ADxCON3<7:0>).
2. Select the port pins as analog inputs (ADxPCFGL<5:0>).
3. Determine how inputs will be allocated to Sample and Hold channels (ADxCHS0<15:0> and ADxCHS123<15:0>).
4. Determine how many Sample and Hold channels will be used (ADxCON2<9:8> and ADxPCFGL<5:0>).
5. Determine how sampling will occur (ADxCON1<3> and ADxCSSL<5:0>).
6. Select manual or auto-sampling.
7. Select the conversion trigger and sampling time.
8. Select how the conversion results are stored in the buffer (ADxCON1<9:8>).
9. Select the data format.
10. Configure the ADC interrupt (if required):
    - Clear the ADxIF bit
    - Select interrupt priority (ADxIP<2:0>)
    - Set the ADxIE bit
11. Turn on the ADC module (ADxCON1<15>).

The options for these configuration steps are described in subsequent sections.

49.8 ADC CONFIGURATION FOR 1.1 Msps

When the device is running at 13.3 MIPS, the ADC module can be configured to sample at a 1.1 Msps throughput rate.

The ASAM bit (ADxCON1<2>) is set to '1' to begin sampling automatically after the conversion completes. The internal counter, which ends sampling and starts conversion, is set as the sample clock source by setting the SSRC<2:0> bits = 111 (ADxCON1<7:5>). The system clock is selected to be the ADC conversion clock by setting the ADRC bit to '0' (ADxCON3<15>). The automatic sample time bit is set to less than 12 TAD. The ADC conversion time is configured to 75 ns by setting the ADCS<7:0> bits to '00000000' (ADxCON3<7:0>), as calculated in Equation 49-7.

Equation 49-6: ADC Conversion Time When Running at 13.3 MIPS

$$T_{AD} = T_{CY} * (ADCS<7:0> + 1) = (1/13.3M) * 1 = 75 \text{ ns } (13.3 \text{ MHz})$$

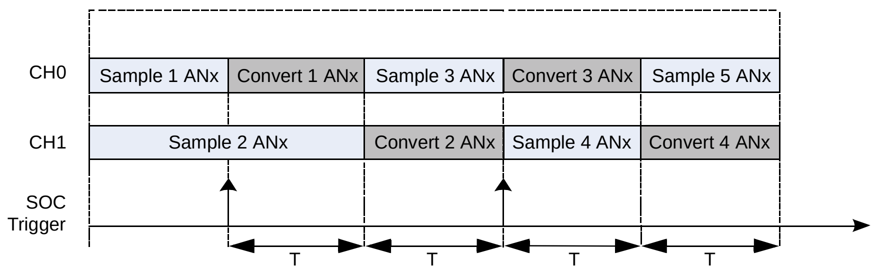
The ADC conversion time will be 12 TAD, as calculated in Equation 49-7.

Equation 49-7: ADC Conversion Time

$$T_{CONV} = 12 * T_{AD} = 900 \text{ ns } (1.1 \text{ MHz})$$

The ADC channels, CH0 and CH1 (CHPS<1:0> = 01), are set up to convert analog input AN0 or AN3 (only one at any time) in Sequential mode (SIMSAM = 0). Figure 49-18 illustrates the sampling sequence.

Figure 49-18: Sampling Sequence for 1.1 Msps



Note: The 'x' in ANx is either 0 or 3. T is 900 ns and the frequency is 1.1 Msps.

The samples are transferred to ADC1BUF0-ADC1BUFF at a rate of 1.1 Msps. The data can be processed by accessing half of the buffers at a time by setting the BUFS bit.

Example 49-8: ADC Configuration Code for 1.1 Msps

```
void initAdc1(void)
{
    AD1CON1bits.FORM = 3;   // Data Output Format: Signed Fraction (Q15 format)
    AD1CON1bits.SSRC = 7;   // Internal Counter (SAMC) ends sampling and starts conversion
    AD1CON1bits.ASAM = 1;   // ADC Sample Control: Sampling begins immediately after
                            // conversion
    AD1CON2bits.SIMSAM = 0; // Sequential sampling of channels

    AD1CON2bits.CHPS = 1;   // Converts channels CH0/CH1

    AD1CON3bits.ADRC = 0;   // ADC Clock is derived from Systems Clock
    AD1CON3bits.SAMC = 0;   // Auto Sample Time = 0 * TAD
    AD1CON3bits.ADCS = 0;   // ADC Conversion Clock TAD = TCY * (ADCS + 1) = (1/13.3M) * 1 =
                            // 75 ns (13.3 MHz)
                            // ADC Conversion Time for Tconv = 12 * TAD = 900 ns (1.1 MHz)

    AD1CON2bits.SMPI = 0;         // SMPI must be 0

    //AD1CHS0/AD1CHS123: A/D Input Select Register
    AD1CHS0bits.CH0SA = 0;        // MUXA +ve input selection (AIN0) for CH0
    AD1CHS0bits.CH0NA = 0;        // MUXA -ve input selection (AVSS) for CH0

    AD1CHS123bits.CH123SA = 0;    // MUXA +ve input selection (AIN0) for CH1
    AD1CHS123bits.CH123NA = 0;    // MUXA -ve input selection (AVSS) for CH1

    //AD1PCFGL: Port Configuration Register
    AD1PCFGL = 0xFFFF;
    AD1PCFGLbits.PCFG0 = 0;    // AN0 as Analog Input
    IFS0bits.AD1IF = 0;        // Clear the A/D interrupt flag bit
    IEC0bits.AD1IE = 0;        // Do Not Enable A/D interrupt
    AD1CON1bits.ADON = 1;      // Turn on the A/D converter
}
```

49.9 SAMPLE AND 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.

49.9.1 Sampling and Converting a Single Channel Multiple Times

Figure 49-19 and Table 49-12 illustrate a basic configuration of the ADC. In this case, one ADC input, AN0, is sampled by one S&H channel, CH0, and converted. The results are stored in the ADC buffer (ADC1BUF0-ADC1BUFF). This process repeats 16 times until the buffer is full and then the ADC module generates an interrupt. The entire process then repeats.

The CHPS bits specify that only S&H CH0 is active. With ALTS clear, only the MUXA inputs are active. The CH0SA bits and CH0NA bit are specified (AN0-AVss) as the input to the S&H channel. All other input selection bits are not used.

Figure 49-19: Converting One Channel 16 Times/Interrupt

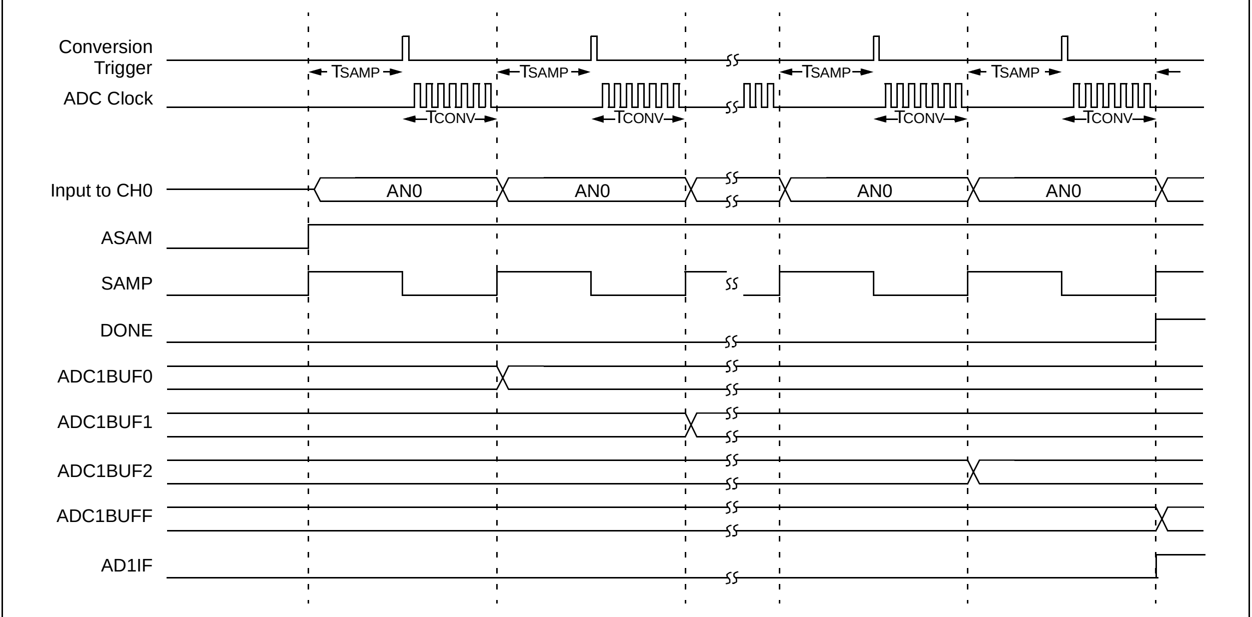


Table 49-12: Converting One Channel 16 Times Per ADC Interrupt

CONTROL BITS

Sequence Select

SMPI<3:0> = 1111	Interrupt on 16th Sample
CHPS<1:0> = 00	Sample Channel CH0
SIMSAM = N/A	Not Applicable for Single Channel Sample
BUFM = 0	Single 16-Word Result Buffer
ALTS = 0	Always use MUXA Input Select

MUXA Input Select

CH0SA<3:0> = 0000	Select AN0 for CH0+ Input
CH0NA = 0	Select AVSS for CH0- Input
CSCNA = 0	No Input Scan
CSSL<15:0> = N/A	Scan Input Select Unused
CH123SA = N/A	Channel CH1, CH2, CH3 + Input Unused
CH123NA<1:0> = N/A	Channel CH1, CH2, CH3 – Input Unused

MUXB Input Select

CH0SB<3:0> = N/A	Channel CH0+ Input Unused
CH0NB = N/A	Channel CH0- Input Unused
CH123SB = N/A	Channel CH1, CH2, CH3 + Input Unused
CH123NB<1:0> = N/A	Channel CH1, CH2, CH3 – Input Unused

OPERATION SEQUENCE
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF0
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF1
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF2
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF3
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF4
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF5
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF6
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF7
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF8
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUF9
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFA
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFB
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFC
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFD
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFE
Sample MUXA Inputs: AN0 → CH0
Convert CH0, Write ADC1BUFF
ADC Interrupt
Repeat

ADC Buffer @ First ADC Interrupt

ADC1BUF0	AN0 Sample 1
ADC1BUF1	AN0 Sample 2
ADC1BUF2	AN0 Sample 3
ADC1BUF3	AN0 Sample 4
ADC1BUF4	AN0 Sample 5
ADC1BUF5	AN0 Sample 6
ADC1BUF6	AN0 Sample 7
ADC1BUF7	AN0 Sample 8
ADC1BUF8	AN0 Sample 9
ADC1BUF9	AN0 Sample 10
ADC1BUFA	AN0 Sample 11
ADC1BUFB	AN0 Sample 12
ADC1BUFC	AN0 Sample 13
ADC1BUFD	AN0 Sample 14
ADC1BUFE	AN0 Sample 15
ADC1BUFF	AN0 Sample 16

ADC Buffer @ Second ADC Interrupt

AN0 Sample 17
AN0 Sample 18
AN0 Sample 19
AN0 Sample 20
AN0 Sample 21
AN0 Sample 22
AN0 Sample 23
AN0 Sample 24
AN0 Sample 25
AN0 Sample 26
AN0 Sample 27
AN0 Sample 28
AN0 Sample 29
AN0 Sample 30
AN0 Sample 31
AN0 Sample 32

49.9.2 A/D Conversions While Scanning Through All Analog Inputs

Figure 49-20 and Table 49-13 illustrate a typical setup where all available analog input channels are sampled by one S&H channel, CH0, and converted. The Set Scan Input Selection bit (CSCNA) in the ADC Control Register 2 (ADxCON2<10>) specifies scanning of the ADC inputs to the CH0 positive input. Other conditions are similar to those described in **Section 49.9.1 "Sampling and Converting a Single Channel Multiple Times"**.

Initially, the AN0 input is sampled by CH0 and converted, and then the AN1 input is sampled and converted. This process of scanning the inputs repeats 6 times until the buffer is full. The result is stored in the ADC buffer (ADC1BUF0-ADC1BUF5), and then the ADC module generates an interrupt. The entire process then repeats.

Figure 49-20: Scanning Through 16 Inputs/Interrupt

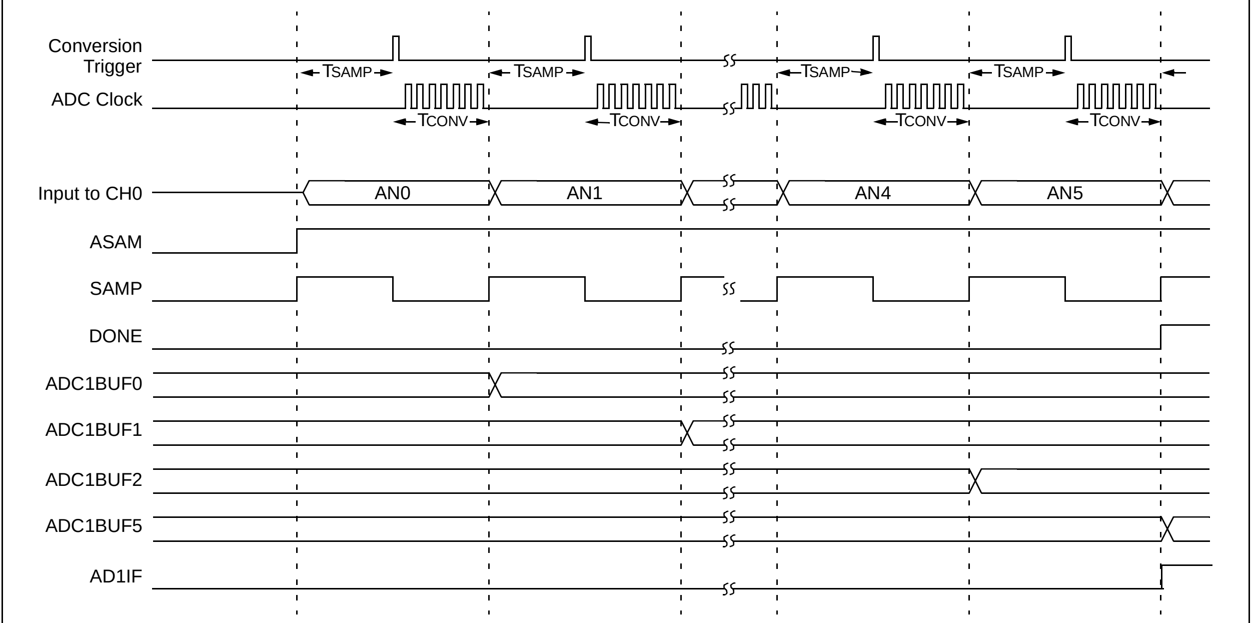


Table 49-13: Scanning Through 6 Inputs per ADC Interrupt

CONTROL BITS

Sequence Select

SMPI<3:0> = `0110`	Interrupt on 6th Sample
CHPS<1:0> = `00`	Sample Channel CH0
SIMSAM = N/A	Not Applicable for Single Channel Sample
BUFM = `0`	Single 16-Word Result Buffer
ALTS = `0`	Always use MUXA Input Select

MUXA Input Select

CH0SA<3:0> = N/A	Override by CSCNA
CH0NA = `0`	Select AVss for CH0- Input
CSCNA = `1`	Scan CH0+ Inputs
CSSL<15:0> = `0000 0000 0011 1111`	Scan Input Select Unused
CH123SA = N/A	Channel CH1, CH2, CH3 + Input Unused
CH123NA<1:0> = N/A	Channel CH1, CH2, CH3 – Input Unused

MUXB Input Select

CH0SB<3:0> = N/A	Channel CH0+ Input Unused
CH0NB = N/A	Channel CH0- Input Unused
CH123SB = N/A	Channel CH1, CH2, CH3 + Input Unused
CH123NB<1:0> = N/A	Channel CH1, CH2, CH3 – Input Unused

OPERATION SEQUENCE
Sample MUXA Inputs: AN0 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF0
Sample MUXA Inputs: AN1 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF1
Sample MUXA Inputs: AN2 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF2
Sample MUXA Inputs: AN3 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF3
Sample MUXA Inputs: AN4 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF4
Sample MUXA Inputs: AN5 $\rightarrow$ CH0
Convert CH0, Write ADC1BUF5
ADC Interrupt
Repeat

	ADC Buffer @ First ADC Interrupt
ADC1BUF0	AN0 Sample 1
ADC1BUF1	AN1 Sample 2
ADC1BUF2	AN2 Sample 3
ADC1BUF3	AN3 Sample 4
ADC1BUF4	AN4 Sample 5
ADC1BUF5	AN5 Sample 6

ADC Buffer @ Second ADC Interrupt
AN0 Sample 7
AN1 Sample 8
AN2 Sample 9
AN3 Sample 10
AN4 Sample 11
AN5 Sample 12

49.9.3 Sampling Three Inputs Frequently While Scanning Three Other Inputs

Figure 49-21 and Table 49-14 illustrate how the ADC module could be configured to sample three inputs frequently using S&H channels, CH1, CH2 and CH3, while four other inputs are sampled less frequently by scanning them using S&H channel, CH0. In this case, only MUXA inputs are used and all four channels are sampled simultaneously. Three different inputs (AN3, AN4 and AN5) are scanned in CH0, whereas AN0, AN1 and AN2 are the fixed inputs for CH1, CH2 and CH3, respectively. Thus, in every set of 12 samples, AN0, AN1 and AN2 are sampled three times, while AN3, AN4 and AN5 are sampled only once each.

Figure 49-21: Converting Three Inputs, Three Times and Three Inputs, One Time/Interrupt

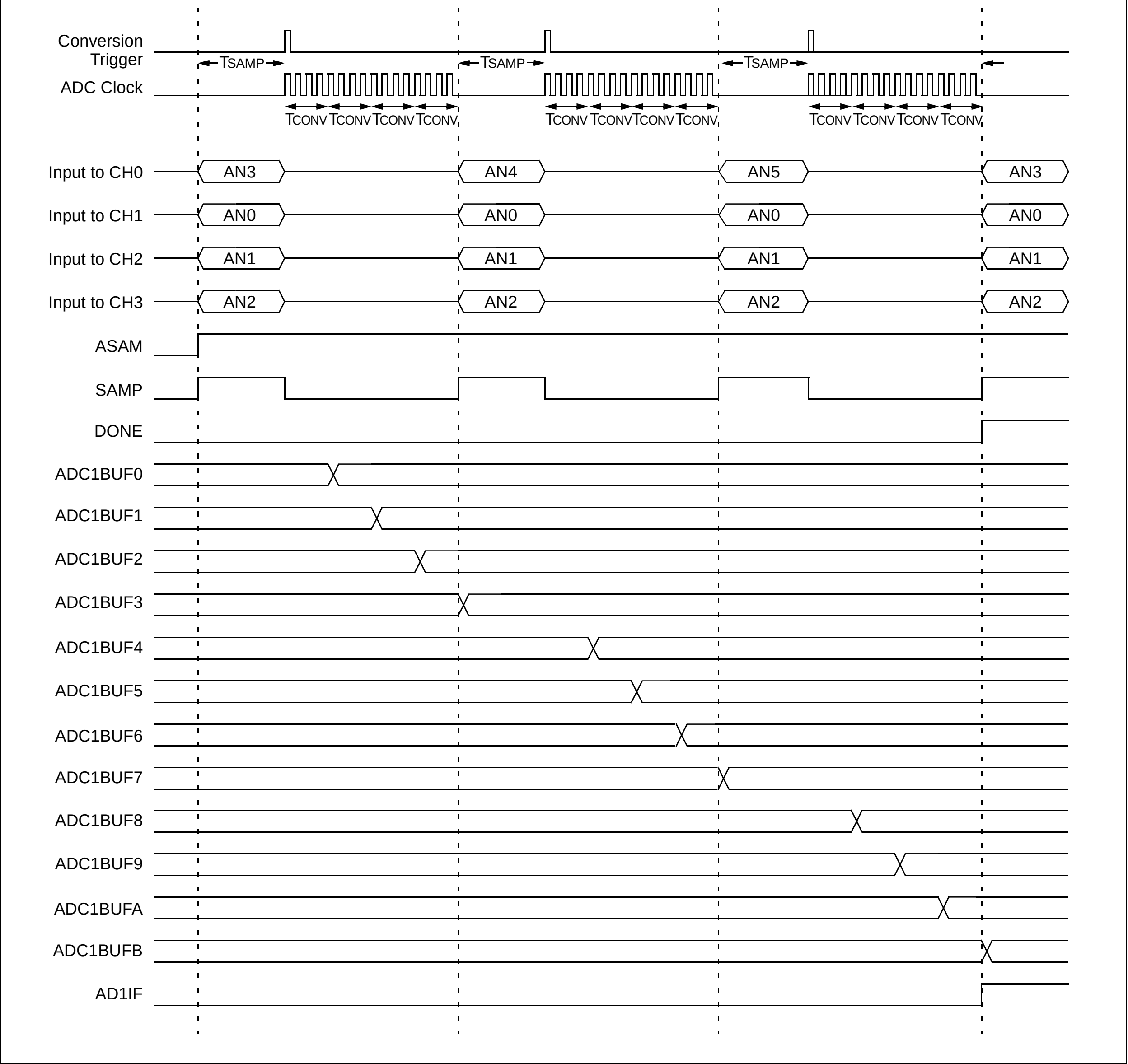


Table 49-14: Converting Three Inputs, Four Times and Four Inputs, One Time per ADC Interrupt

CONTROL BITS	
Sequence Select	
SMPI<3:0> = `0010`	Interrupt on 3rd Sample
CHPS<1:0> = `1x`	Sample Channels CH0, CH1, CH2, CH3
SIMSAM = `1`	Sample All Channels Simultaneously
BUFM = `0`	Single 16-Word Result Buffer
ALTS = `0`	Always Use MUXA Input Select
MUXA Input Select	
CH0SA<3:0> = N/A	Override by CSCNA
CH0NA = `0`	Select AVss for CH0- Input
CSCNA = `1`	Scan CH0+ Inputs
CSSL<15:0> = `0000 0000 1111 0000`	Scan AN4, AN5, AN6, AN7
CH123SA = `0`	CH1+ = AN0, CH2+ = AN1, CH3+ = AN2
CH123NA<1:0> = `0x`	CH1-,CH2-,CH3- = AVss
MUXB Input Select	
CH0SB<3:0> = N/A	Channel CH0+ Input Unused
CH0NB = N/A	Channel CH0- Input Unused
CH123SB = N/A	Channel CH1, CH2, CH3 + Input Unused
CH123NB<1:0> = N/A	Channel CH1, CH2, CH3 – Input Unused

OPERATION SEQUENCE
Sample MUXA Inputs: AN3 → CH0, AN0 → CH1, AN1 → CH2, AN2 → CH3
Convert CH0, Write ADC1BUF0
Convert CH1, Write ADC1BUF1
Convert CH2, Write ADC1BUF2
Convert CH3, Write ADC1BUF3
Sample MUXA Inputs: AN4 → CH0, AN0 → CH1, AN1 → CH2, AN2 → CH3
Convert CH0, Write ADC1BUF4
Convert CH1, Write ADC1BUF5
Convert CH2, Write ADC1BUF6
Convert CH3, Write ADC1BUF7
Sample MUXA Inputs: AN5 → CH0, AN0 → CH1, AN1 → CH2, AN2 → CH3
Convert CH0, Write ADC1BUF8
Convert CH1, Write ADC1BUF9
Convert CH2, Write ADC1BUFA
Convert CH3, Write ADC1BUFB
ADC Interrupt
Repeat

	ADC Buffer @ First ADC Interrupt	ADC Buffer @ Second ADC Interrupt
ADC1BUF0	AN3 Sample 1	AN3 Sample 2
ADC1BUF1	AN0 Sample 1	AN0 Sample 4
ADC1BUF2	AN1 Sample 1	AN1 Sample 4
ADC1BUF3	AN2 Sample 1	AN2 Sample 4
ADC1BUF4	AN4 Sample 1	AN4 Sample 2
ADC1BUF5	AN0 Sample 2	AN0 Sample 5
ADC1BUF6	AN1 Sample 2	AN1 Sample 5
ADC1BUF7	AN2 Sample 2	AN2 Sample 5
ADC1BUF8	AN5 Sample 1	AN5 Sample 2
ADC1BUF9	AN0 Sample 3	AN0 Sample 6
ADC1BUFA	AN1 Sample 3	AN1 Sample 6
ADC1BUFB	AN2 Sample 3	AN2 Sample 6

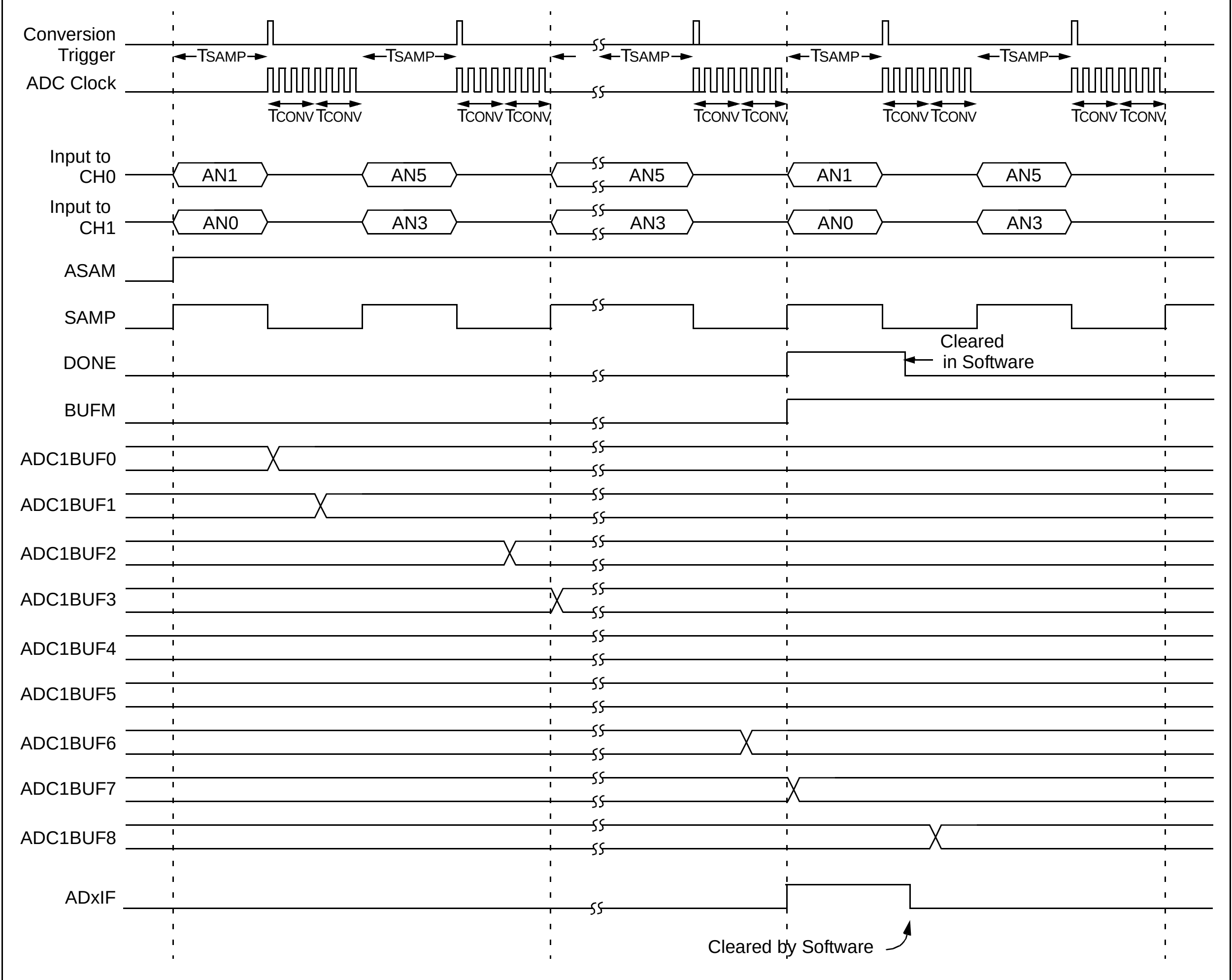
Note: In this instance of simultaneous sampling, one sample and four conversions are treated as one sample and a convert sequence. Therefore, when SMPI<3:0> = `0010`, an ADC interrupt is generated after 12 samples are converted and buffered in ADC1BUF0-ADC1BUFB.

49.9.4 Using Alternating MUXA, MUXB Input Selections

Figure 49-22 and Table 49-15 demonstrate alternate sampling of the inputs assigned to MUXA and MUXB. In this example, two channels are enabled to sample simultaneously. Setting the ALTS bit (ADCxCON2<0>) enables alternating input selections. The first sample uses the MUXA inputs specified by the CH0SA, CH0NA, CH123SA and CH123NA bits. The next sample uses the MUXB inputs specified by the CH0SB, CH0NB, CH123SB and CH123NB bits.

Note that using four S&H channels, without alternating input selections, results in the same number of conversions as this example, using two channels with alternating input selections. However, because the CH1, CH2 and CH3 channels are more limited in the selectivity of the analog inputs, this example method provides more flexibility of input selection than using four channels.

Figure 49-22: Converting Two Sets of Two Inputs Using Alternating Input Selections



Preliminary © 2010 Microchip Technology Inc.

Table 49-15: Converting Two Sets of Two Inputs Using Alternating Input Selections

CONTROL BITS

Sequence Select

SMPI<3:0> = 0011	Interrupt on 4th Sample
CHPS<1:0> = 01	Sample Channels CH0, CH1
SIMSAM = 1	Sample All Channels Simultaneously
BUFM = 1	Dual 8-Word Result Buffers
ALTS = 1	Alternate MUXA/MUXB Input Select

MUXA Input Select

CH0SA<3:0> = 0001	Select AN1 for CH0+ Input
CH0NA = 0	Select AVss for CH0- Input
CSCNA = 0	No Input Scan
CSSL<15:0> = N/A	Scan Input Select Unused
CH123SA = 0	CH1+ = AN0, CH2+ = AN1,CH3+ = AN2
CH123NA<1:0> = 0x	CH1-, CH2-, CH3- = AVss

MUXB Input Select

CH0SB<3:0> = 0101	Select AN5 for CH0+ Input
CH0NB = 0	Select AVss for CH0- Input
CH123SB = 1	CH1+ = AN3, CH2+ = AN4, CH3+ = AN5
CH123NB<1:0> = 0x	CH1-, CH2-, CH3- = AVss

OPERATION SEQUENCE

Sample MUXA Inputs: AN1 → CH0, AN0 → CH1
Convert CH0, Write ADC1BUF0
Convert CH1, Write ADC1BUF1
Sample MUXB Inputs: AN5 → CH0, AN3 → CH1
Convert CH0, Write ADC1BUF2
Convert CH1, Write ADC1BUF3
Sample MUXA Inputs: AN1 → CH0, AN0 → CH1
Convert CH0, Write ADC1BUF4
Convert CH1, Write ADC1BUF5
Sample MUXB Inputs: AN5 → CH0, AN3 → CH1
Convert CH0, Write ADC1BUF6
Convert CH1, Write ADC1BUF7
Interrupt; Change Buffer
Sample MUXA Inputs: AN1 → CH0, AN0 → CH1
Convert CH0, Write ADC1BUF8
Convert CH1, Write ADC1BUF9
Sample MUXB Inputs: AN5 → CH0, AN3 → CH1
Convert CH0, Write ADC1BUFA
Convert CH1, Write ADC1BUFB
Sample MUXA Inputs: AN1 → CH0, AN0 → CH1
Convert CH0, Write ADC1BUFC
Convert CH1, Write ADC1BUFD
Sample MUXB Inputs: AN5 → CH0, AN3 → CH1
Convert CH0, Write ADC1BUFE
Convert CH1, Write ADC1BUFF
ADC Interrupt; Change Buffer
Repeat

	ADC Buffer @ First ADC Interrupt	ADC Buffer @ Second ADC Interrupt
ADC1BUF0	AN1 Sample 1	
ADC1BUF1	AN0 Sample 1	
ADC1BUF2	AN5 Sample 2	
ADC1BUF3	AN3 Sample 2	
ADC1BUF4	AN1 Sample 3	
ADC1BUF5	AN0 Sample 3	
ADC1BUF6	AN5 Sample 4	
ADC1BUF7	AN3 Sample 4	
ADC1BUF8		AN1 Sample 5
ADC1BUF9		AN0 Sample 5
ADC1BUFA		AN5 Sample 6
ADC1BUFB		AN3 Sample 6
ADC1BUFC		AN1 Sample 7
ADC1BUFD		AN0 Sample 7
ADC1BUFE		AN5 Sample 8
ADC1BUFF		AN3 Sample 8

49.9.5 Sampling Eight Inputs Using Simultaneous Sampling

This section and the next example demonstrate identical setups with the exception that this example uses simultaneous sampling (SIMSAM = `1`), and the following example uses sequential sampling (SIMSAM = `0`). Both examples use alternating inputs and specify differential inputs to the S&H.

Figure 49-23 and Table 49-16 demonstrate simultaneous sampling. When converting more than one channel and selecting simultaneous sampling, the ADC module samples all channels, then performs the required conversions in sequence. In this example, with ASAM set, sampling begins after the conversions complete.

Figure 49-23: Sampling Eight Inputs Using Simultaneous Sampling

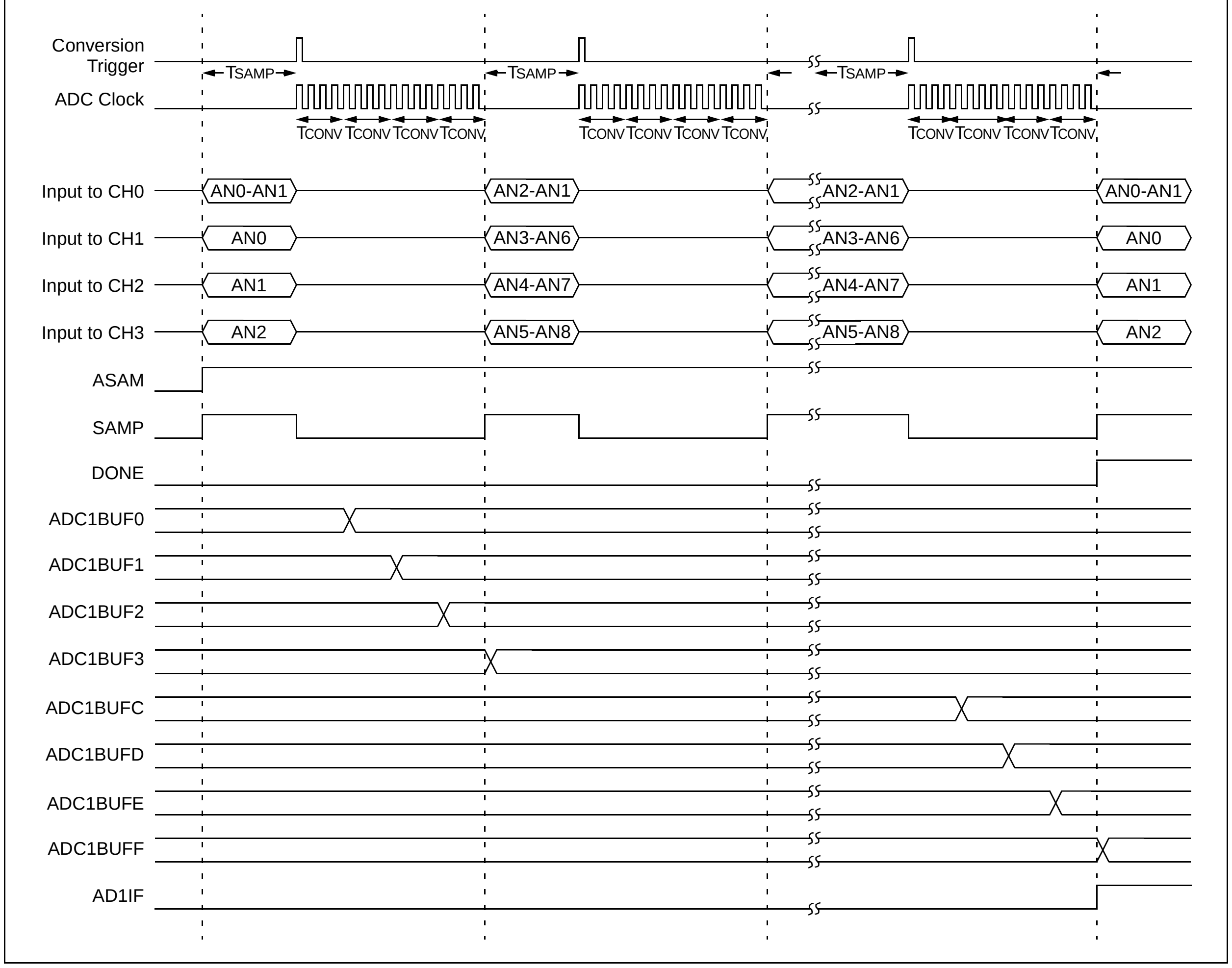


Table 49-16: Sampling Eight Inputs Using Simultaneous Sampling

CONTROL BITS

Sequence Select

SMPI<3:0> = `0011`	Interrupt on 4th Sample
CHPS<1:0> = `1x`	Sample Channels CH0, CH1, CH2, CH3
SIMSAM = `1`	Sample All Channels Simultaneously
BUFM = `0`	Single 16-Word Result Buffer
ALTS = `1`	Alternate MUXA/MUXB Input Select

MUXA Input Select

CH0SA<3:0> = `0000`	Select AN0 for CH0+ Input
CH0NA = `1`	Select AN1 for CH0- Input
CSCNA = `0`	No Input Scan
CSSL<15:0> = N/A	Scan Input Select Unused
CH123SA = `0`	CH1+ = AN0, CH2+ = AN1, CH3+ = AN2
CH123NA<1:0> = `0x`	CH1-, CH2-, CH3- = AVss

MUXB Input Select

CH0SB<3:0> = `0010`	Select AN2 for CH0+ Input
CH0NB = `1`	Select AN1 for CH0- Input
CH123SB = `1`	CH1+ = AN3, CH2+ = AN4, CH3+ = AN5
CH123NB<1:0> = `0x`	CH1-, CH2-, CH3- = AVss

OPERATION SEQUENCE

Sample MUXA Inputs: (AN0-AN1) → CH0, AN0 → CH1, AN1 → CH2, AN2 → CH3
Convert CH0, Write ADC1BUF0
Convert CH1, Write ADC1BUF1
Convert CH2, Write ADC1BUF2
Convert CH3, Write ADC1BUF3
Sample MUXB Inputs: (AN2-AN1) → CH0,
AN3 → CH1, AN4 → CH2, AN5 → CH3
Convert CH0, Write ADC1BUF4
Convert CH1, Write ADC1BUF5
Convert CH2, Write ADC1BUF6
Convert CH3, Write ADC1BUF7
Sample MUXA Inputs: (AN0-AN1) → CH0, AN0 → CH1, AN1 → CH2, AN2 → CH3
Convert CH0, Write ADC1BUF8
Convert CH1, Write ADC1BUF9
Convert CH2, Write ADC1BUFA
Convert CH3, Write ADC1BUFB
Sample MUXB Inputs: (AN2-AN1) → CH0,
AN3 → CH1, AN4 → CH2, AN5 → CH3
Convert CH0, Write ADC1BUFC
Convert CH1, Write ADC1BUFD
Convert CH2, Write ADC1BUFE
Convert CH3, Write ADC1BUFF
ADC Interrupt
Repeat

	ADC Buffer @ First ADC Interrupt	ADC Buffer @ Second ADC Interrupt
ADC1BUF0	(AN0-AN1) Sample 1	(AN0-AN1) Sample 3
ADC1BUF1	AN0 Sample 1	AN0 Sample 3
ADC1BUF2	AN1 Sample 1	AN1 Sample 3
ADC1BUF3	AN2 Sample 1	AN2 Sample 3
ADC1BUF4	AN2-AN1 Sample 1	AN2-AN1 Sample 3
ADC1BUF5	AN3 Sample 1	AN3 Sample 3
ADC1BUF6	AN4 Sample 1	AN4 Sample 3
ADC1BUF7	AN5 Sample 1	AN5 Sample 3
ADC1BUF8	(AN0-AN1) Sample 1	(AN0-AN1) Sample 4
ADC1BUF9	AN0 Sample 2	AN0 Sample 4
ADC1BUFA	AN1 Sample 2	AN1 Sample 4
ADC1BUFB	AN2 Sample 2	AN2 Sample 4
ADC1BUFC	(AN2-AN1 Sample 2	(AN2-AN1) Sample 4
ADC1BUFD	AN3 Sample 2	AN3 Sample 4
ADC1BUFE	AN4 Sample 2	AN4 Sample 4
ADC1BUFF	AN5 Sample 2	AN5 Sample 4

49.9.6 Sampling Eight Inputs Using Sequential Sampling

Figure 49-24 and Table 49-17 demonstrate sequential sampling. When converting more than one channel and selecting sequential sampling, the ADC module starts sampling a channel at the earliest opportunity, then performs the required conversions in sequence. In this example, with ASAM set, sampling of a channel begins after the conversion of that channel completes.

When ASAM is clear, sampling does not resume after conversion completion, but occurs when the SAMP bit is set.

When utilizing more than one channel, sequential sampling provides more sampling time, since a channel can be sampled while conversion occurs on another.

Figure 49-24: Sampling Eight Inputs Using Sequential Sampling

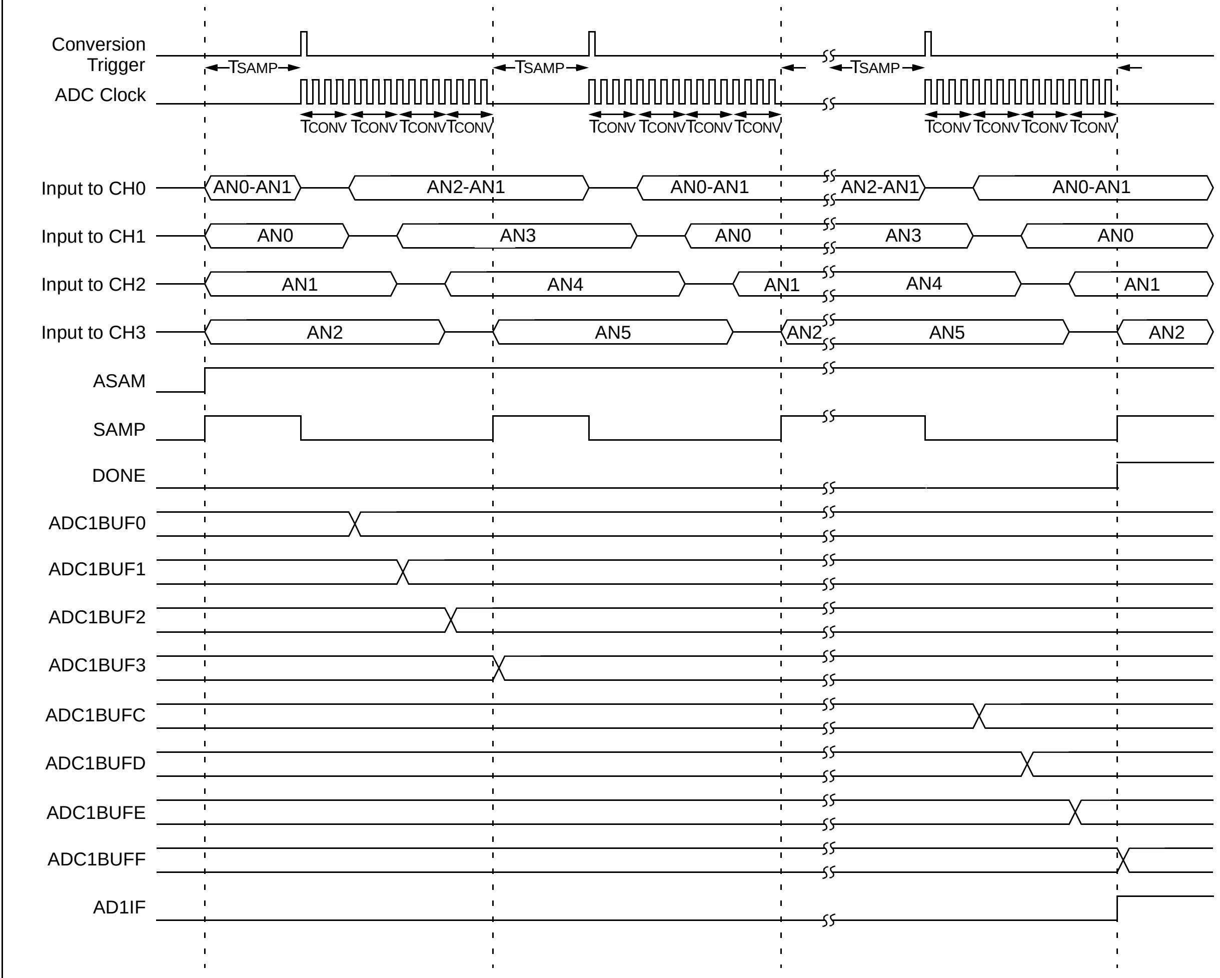


Table 49-17: Sampling Eight Inputs Using Sequential Sampling

CONTROL BITS

Sequence Select

SMPI<3:0> = `1111`	Interrupt on 16th Sample
CHPS<1:0> = `1x`	Sample Channels CH0, CH1, CH2, CH3
SIMSAM = `0`	Sample All Channels Sequentially
BUFM = `0`	Single 16-Word Result Buffer
ALTS = `1`	Alternate MUXA/MUXB Input Select

MUXA Input Select

CH0SA<3:0> = `0000`	Select AN0 for CH0+ Input
CH0NA = `1`	Select AN1 for CH0- Input
CSCNA = `0`	No Input Scan
CSSL<15:0> = N/A	Scan Input Select Unused
CH123SA = `0`	CH1+ = AN0, CH2+ = AN1, CH3+ = AN2
CH123NA<1:0> = `0X`	CH1-, CH2-, CH3- = AVss

MUXB Input Select

CH0SB<3:0> = `0010`	Select AN2 for CH0+ Input
CH0NB = `1`	Select AN1 for CH0- Input
CH123SB = `1`	CH1+ = AN3, CH2+ = AN4, CH3+ = AN5
CH123NB<1:0> = `0x`	CH1-, CH2-, CH3- = AVss

OPERATION SEQUENCE
Sample: (AN0-AN1) → CH0
Convert CH0, Write ADC1BUF0
Sample: AN0 → CH1
Convert CH1, Write ADC1BUF1
Sample: AN1 → CH2
Convert CH2, Write ADC1BUF2
Sample: AN2 → CH3
Convert CH3, Write ADC1BUF3
Sample: (AN2-AN1) → CH0
Convert CH0, Write ADC1BUF4
Sample: AN3 → CH1
Convert CH1, Write ADC1BUF5
Sample: AN4 → CH2
Convert CH2, Write ADC1BUF6
Sample: AN5 → CH3
Convert CH3, Write ADC1BUF7
Sample: (AN0-AN1) → CH0
Convert CH0, Write ADC1BUF8
Sample: AN0 → CH1
Convert CH1, Write ADC1BUF9
Sample: AN1 → CH2
Convert CH2, Write ADC1BUFA
Sample: AN2 → CH3
Convert CH3, Write ADC1BUFB
Sample: (AN2-AN1) → CH0
Convert CH0, Write ADC1BUFC
Sample: AN3 → CH1
Convert CH1, Write ADC1BUFD
Sample: AN4 → CH2
Convert CH2, Write ADC1BUFE
Sample: AN5 → CH3
Convert CH3, Write ADC1BUFF
ADC Interrupt
Repeat

	ADC Buffer @ First ADC Interrupt	ADC Buffer @ Second ADC Interrupt
ADC1BUF0	(AN0-AN1) Sample 1	(AN0-AN1) Sample 3
ADC1BUF1	AN0 Sample 1	AN0 Sample 3
ADC1BUF2	AN1 Sample 1	AN1 Sample 3
ADC1BUF3	AN2 Sample 1	AN2 Sample 3
ADC1BUF4	(AN2-AN1) Sample 1	(AN2-AN1) Sample 3
ADC1BUF5	AN3 Sample 1	AN3 Sample 3
ADC1BUF6	AN4 Sample 1	AN4 Sample 3
ADC1BUF7	AN5 Sample 1	AN5 Sample 3
ADC1BUF8	(AN0-AN1) Sample 2	(AN0-AN1) Sample 4
ADC1BUF9	AN0 Sample 2	AN0 Sample 4
ADC1BUFA	AN1 Sample 2	AN1 Sample 4
ADC1BUFB	AN2 Sample 2	AN2 Sample 28
ADC1BUFC	(AN2-AN1) Sample 2	(AN2-AN1) Sample 4
ADC1BUFD	AN3 Sample 2	AN3 Sample 4
ADC1BUFE	AN4 Sample 2	AN4 Sample 4
ADC1BUFF	AN5 Sample 2	AN5 Sample 4

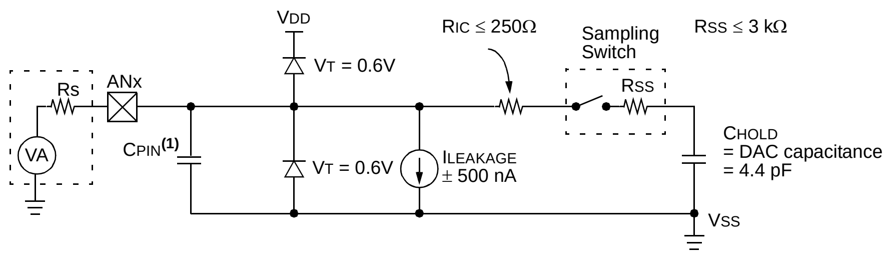
49.10 A/D SAMPLING REQUIREMENTS

The analog input model of the 10-bit ADC model is shown in Figure 49-25. The total sampling time for the A/D conversion is a function of the internal amplifier settling time and the holding capacitor charge time.

For the ADC module to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the voltage level on the analog input pin. The analog output source impedance (R_S), the interconnect impedance (R_{IC}) and the internal sampling switch (R_{SS}) impedance combine to directly affect the time required to charge the capacitor C_{HOLD}. The combined impedance 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 ADC module, the maximum recommended source impedance, R_S, which is 200Ω. After the analog input channel is selected, 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.

A minimum time period should be allowed between conversions for the sample time. For more details about the minimum sampling time for a device, refer to the **"Electrical Characteristics"** chapter of the specific device data sheet.

Figure 49-25: 10-Bit Analog Input Model



Legend:	C_{PIN}	= Input Capacitance
	V_T	= Threshold Voltage
	$I_{LEAKAGE}$	= Leakage Current at the pin due to various junctions
	R_{IC}	= Interconnect Resistance
	R_{SS}	= Sampling Switch Resistance
	C_{HOLD}	= S&H Capacitance (from DAC)

Note 1: C_{PIN} value depends on device package and is not tested. The effect of C_{PIN} is negligible if $R_S \leq 500\Omega$.

49.11 READING THE ADC RESULT BUFFER

The RAM is 10 bits wide, but the data is automatically formatted to one of four selectable formats when the buffer is read. The FORM<1:0> bits (ADCON1<9:8>) select the format. The formatting hardware provides a 16-bit result on the data bus for all of the data formats. Figure 49-26 illustrates the data output formats that can be selected using the FORM<1:0> control bits.

Figure 49-26: 10-Bit A/D Output Data Formats

RAM Contents:							d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Unsigned Integer	0	0	0	0	0	0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Signed Integer	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	d08	d07	d06	d05	d04	d03	d02	d01	d00
Unsigned Fractional (1.15)	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
Signed Fractional (1.15)	$\overline{d09}$	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0

Table 49-18 lists the numerical equivalents of various result codes for 10-bit.

Table 49-18: Numerical Equivalents of Various Result Codes

VIN/AVSS	10-Bit Output Code	16-Bit Integer Format	16-Bit Signed Integer Format	16-Bit Fractional Format	16-Bit Signed Fractional Format
1023/1024	`11 1111 1111`	`0000 0011 1111 1111 = 1023`	`0000 0001 1111 1111 = 511`	`1111 1111 1100 0000 = 0.999`	`0111 1111 1100 0000 = 0.99804`
1022/1024	`11 1111 1110`	`0000 0011 1111 1110 = 1022`	`0000 0001 1111 1110 = 510`	`1111 1111 1000 0000 = 0.998`	`0111 1111 1000 0000 = 0.499609`
			⋮		
513/1024	`10 0000 0001`	`0000 0010 0000 0001 = 513`	`0000 0000 0000 0001 = 1`	`1000 0000 0100 0000 = 0.501`	`0000 0000 0100 0000 = 0.00195`
512/1024	`10 0000 0000`	`0000 0010 0000 0000 = 512`	`0000 0000 0000 0000 = 0`	`1000 0000 0000 0000 = 0.500`	`0000 0000 0000 0000 = 0`
511/1024	`01 1111 1111`	`0000 0001 1111 1111 = 511`	`1111 1111 1111 1111 = -1`	`0111 1111 1100 0000 = .499`	`1111 1111 1100 0000 = -0.00195`
			⋮		
1/1024	`00 0000 0001`	`0000 0000 0000 0001 = 1`	`1111 1110 0000 0001 = -511`	`0000 0000 0100 0000 = 0.001`	`1000 0000 0100 0000 = -0.99804`
0/1024	`00 0000 0000`	`0000 0000 0000 0000 = 0`	`1111 1110 0000 0000 = -512`	`0000 0000 0000 0000 = 0`	`1000 0000 0000 0000 = -1`

Preliminary

© 2010 Microchip Technology Inc.

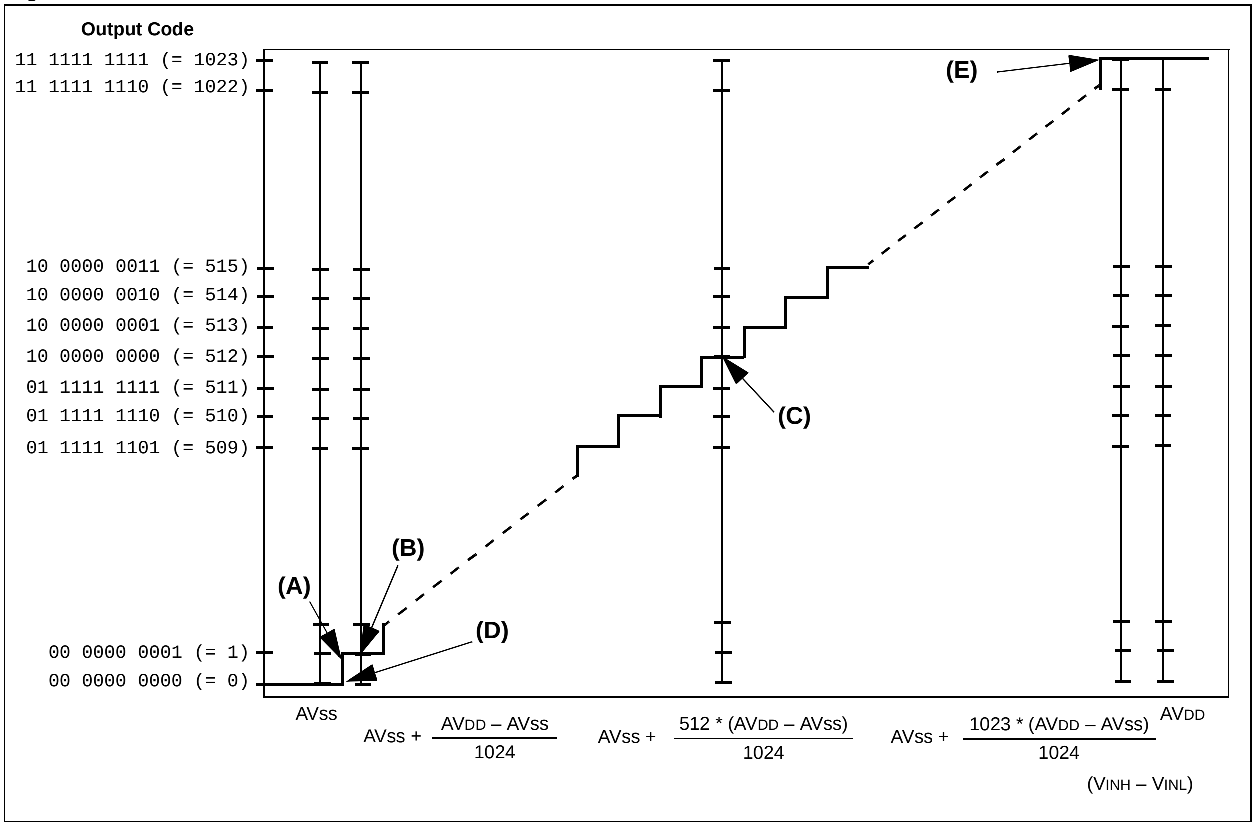
49.12 TRANSFER FUNCTIONS

49.12.1 10-Bit

The ideal transfer function of the ADC module is shown in Figure 49-27. The difference of the input voltages (VINH – VINL) is compared to the reference (AVDD – AVSS).

- The first code transition (**A**) occurs when the input voltage is AVDD – AVSS/2048 or 0.5 LSb.
- The `00 0000 0001` code is centered at AVDD – AVSS/1024 or 1.0 LSb (**B**).
- The `10 0000 0000` code is centered at 512 * (AVDD – AVSS)/1024 (**C**).
- An input voltage less than 1 * (AVDD – AVSS)/2048 converts as `00 0000 0000` (**D**).
- An input greater than 2045 * (AVDD – AVSS)/2048 converts as `11 1111 1111` (**E**).

Figure 49-27: 10-bit ADC Module Transfer Function



49.13 ADC ACCURACY/ERROR

Refer to the **"Electrical Characteristics"** chapter of the specific device data sheet for information on the INL, DNL, gain and offset errors. In addition, see **Section 49.19 "Related Application Notes"** for a list of documents that discuss ADC accuracy.

49.14 CONNECTION CONSIDERATIONS

Since the analog inputs employ ESD protection, they have diodes to VDD and VSS. As a result, the analog input must be between VDD and VSS. If the input voltage exceeds this range by greater than 0.3 V (either direction), one of the diodes becomes forward biased, and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the sampling time requirements are satisfied. Any external components connected (via high-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

49.15 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.

49.15.1 CPU Sleep Mode without RC A/D Clock

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

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

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

49.15.2 CPU Sleep Mode with RC A/D Clock

The ADC 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 is set and the result is loaded into the ADC Result Buffer, ADCxBUF0.

If the ADC interrupt is enabled (ADxIE = 1), the device wakes up from Sleep when the ADC interrupt occurs. Program execution resumes at the ADC Interrupt Service Routine (ISR) if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction, after the PWRSAV instruction, that placed the device in Sleep mode.

If the ADC interrupt is not enabled, the ADC module is turned off, although the ADON bit remains set.

To minimize the effects of digital noise on the ADC 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<2:0> = 111). To use the automatic conversion option, the ADON bit should be set in the instruction before the PWRSAV instruction.

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

49.15.3 ADC Operation During CPU Idle Mode

For the A/D conversion, the ADSIDL bit (ADxCON1<13>) selects if the ADC module stops or continues on Idle. If ADSIDL = 0, the ADC module continues normal operation when the device enters Idle mode. If the ADC interrupt is enabled (ADxIE = 1), the device wakes up from Idle mode when the ADC interrupt occurs. Program execution resumes at the ADC Interrupt Service Routine if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction, after the PWRSAV instruction, that placed the device in Idle mode.

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

49.16 EFFECTS OF A RESET

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

The value in the ADCxBUF0-ADCxBUFF registers is not initialized during a Power-on Reset (POR) and contains unknown data.

49.17 SPECIAL FUNCTION REGISTERS

A summary of the registers associated with the PIC24F 10-Bit ADC with 4 Simultaneous Conversions module is provided in Table 49-19.

Table 49-19: 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
ADCxBUF0	ADCx Data Buffer 0																uuuu
ADCxBUF1	ADCx Data Buffer 1																uuuu
ADCxBUF2	ADCx Data Buffer 2																uuuu
ADCxBUF3	ADCx Data Buffer 3																uuuu
ADCxBUF4	ADCx Data Buffer 4																uuuu
ADCxBUF5	ADCx Data Buffer 5																uuuu
ADCxBUF6	ADCx Data Buffer 6																uuuu
ADCxBUF7	ADCx Data Buffer 7																uuuu
ADCxBUF8	ADCx Data Buffer 8																uuuu
ADCxBUF9	ADCx Data Buffer 9																uuuu
ADCxBUFA	ADCx Data Buffer 10																uuuu
ADCxBUFB	ADCx Data Buffer 11																uuuu
ADCxBUFC	ADCx Data Buffer 12																uuuu
ADCxBUFD	ADCx Data Buffer 13																uuuu
ADCxBUFE	ADCx Data Buffer 14																uuuu
ADCxBUFF	ADCx Data Buffer 15																uuuu
ADxCON1	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000
ADxCON2	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000
ADxCON3	ADRC	—	—	SAMC<4:0>					ADCS<7:0>								0000
ADxCHS123	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
ADxCHS0	CH0NB	—	—	CH0SB<4:0>					CH0NA	—	—	CH0SA<4:0>					0000
ADxPCFGL	—	—	—	—	—	—	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
ADxCSSL	—	—	—	—	—	—	—	—	—	—	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000

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

49.18 DESIGN TIPS

Question 1: ***How can I optimize the system performance of the ADC module?***

Answer: Here are three suggestions for optimizing performance:

1. Make sure you are meeting all of the timing specifications. If you are turning the ADC 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. Finally, there is T_{AD}, which is the time selected for each bit conversion. T_{AD} is selected in ADxCON3 and should be within a range, as specified in the **"Electrical Characteristics"** chapter of the specific device data sheet. If T_{AD} is too short, the result may not be fully converted before the conversion is terminated. If T_{AD} is too long, the voltage on the sampling capacitor can decay before the conversion is complete. These timing specifications are provided in the **"Electrical Characteristics"** chapter of the specific device data sheet.
2. Often the source impedance of the analog signal is high (greater than 10 kΩ), so the current drawn from the source 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 charges to the analog voltage being sampled and supplies 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 ADCs?***

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/sample 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.

49.19 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 I/O Ports with 10-Bit ADC with 4 Simultaneous Conversions 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.

49.20 REVISION HISTORY

Revision A (August 2010)

This is the initial released version 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.

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

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

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 978-1-60932-479-7

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.

MICROCHIP

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://support.microchip.com
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

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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

08/04/10