MICROCHIP

Section 18. 12-bit A/D Converter

HIGHLIGHTS

This section of the manual contains the following major topics:

18.1 Introduction 18-2
18.2 Control Registers 18-4
18.3 A/D Result Buffer 18-4
18.4 A/D Terminology and Conversion Sequence 18-10
18.5 A/D Module Configuration 18-11
18.6 Selecting the Voltage Reference Source 18-11
18.7 Selecting the A/D Conversion Clock 18-12
18.8 Selecting Analog Inputs for Sampling 18-12
18.9 Enabling the Module 18-14
18.10 How to Start Sampling 18-14
18.11 How to Stop Sampling and Start Conversions 18-14
18.12 Controlling Sample/Conversion Operation 18-19
18.13 Specifying How Conversion Results are Written into the Buffer 18-20
18.14 Turning the A/D Module Off 18-20
18.15 Conversion Sequence Examples 18-21
18.16 A/D Sampling Requirements 18-26
18.17 Reading the A/D Result Buffer 18-27
18.18 Transfer Function 18-28
18.19 A/D Accuracy/Error 18-28
18.20 Connection Considerations 18-28
18.21 Initialization 18-29
18.22 A/D Conversion Speeds 18-30
18.23 Operation During Sleep and Idle Modes 18-33
18.24 Effects of a Reset 18-33
18.25 Special Function Registers Associated with the 12-bit A/D Converter 18-34
18.26 Design Tips 18-35
18.27 Related Application Notes 18-36
18.28 Revision History 18-37

18.1 Introduction

The dsPIC30F 12-bit A/D converter has the following key features:

- Successive Approximation Register (SAR) conversion
- Up to 200 ksps conversion speed
- Up to 16 analog input pins
- External voltage reference input pins
- Unipolar differential S/H amplifier
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 16-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

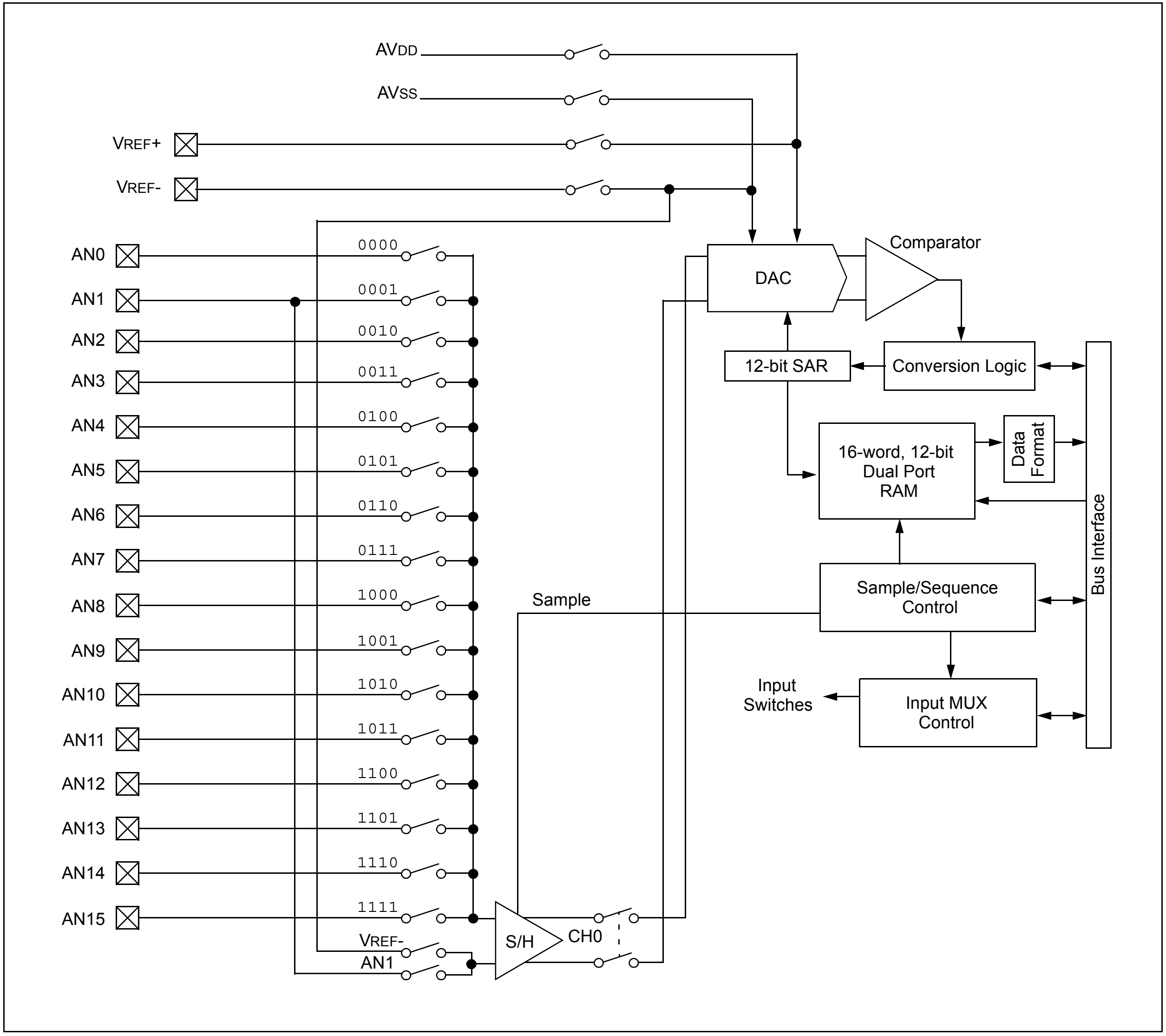
A block diagram of the 12-bit A/D is shown in Figure 18-1. The 12-bit A/D converter can have up to 16 analog input pins, designated AN0-AN15. In addition, there are two analog input pins for external voltage reference connections. These voltage reference inputs may be shared with other analog input pins. The actual number of analog input pins and external voltage reference input configuration will depend on the specific dsPIC30F device. Refer to the dsPIC30F device data sheets (DS70082 and DS70083) for further details.

The analog inputs are connected via multiplexers to the S/H amplifier, designated CH0. The analog input multiplexer can be switched between two sets of analog inputs during conversions. Unipolar differential conversions are possible using certain input pins (see Figure 18-1).

An Analog Input Scan mode may be enabled for the CH0 S/H amplifier. A Control register specifies which analog input channels will be included in the scanning sequence.

The 12-bit A/D is connected to a 16-word result buffer. Each 12-bit result is converted to one of four 16-bit output formats when it is read from the buffer.

Figure 18-1: 12-bit High Speed A/D Block Diagram



18.2 Control Registers

The A/D module has six Control and Status registers. These registers are:

- ADCON1: A/D Control Register 1
- ADCON2: A/D Control Register 2
- ADCON3: A/D Control Register 3
- ADCHS: A/D Input Channel Select Register
- ADPCFG: A/D Port Configuration Register
- ADCSSL: A/D Input Scan Selection Register

The ADCON1, ADCON2 and ADCON3 registers control the operation of the A/D module. The ADCHS register selects the input pins to be connected to the S/H amplifiers. The ADPCFG register configures the analog input pins as analog inputs or as digital I/O. The ADCSSL register selects inputs to be sequentially scanned.

18.3 A/D Result Buffer

The module contains a 16-word dual port RAM, called ADCBUF, to buffer the A/D results. The 16 buffer locations are referred to as ADCBUF0, ADCBUF1, ADCBUF2, ..., ADCBUFE, ADCBUFF.

> **Note:** The A/D result buffer is a read only buffer.

Register 18-1: ADCON1: A/D Control Register 1

Upper Byte:

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	—	FORM<1:0>	
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0 HC, HS	R/C-0 HC, HS
SSRC<2:0>			—	—	ASAM	SAMP	DONE
bit 7							bit 0

bit 15 **ADON:** A/D Operating Mode bit
1 = A/D converter module is operating
0 = A/D converter is off

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ADSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-10 **Unimplemented:** Read as ‘0’

bit 9-8 **FORM<1:0>:** Data Output Format bits
11 = Signed fractional (DOUT = sddd dddd dddd 0000)
10 = Fractional (DOUT = dddd dddd dddd 0000)
01 = Signed integer (DOUT = ssss sddd dddd dddd)
00 = Integer (DOUT = 0000 dddd dddd dddd)

bit 7-5 **SSRC<2:0>:** Conversion Trigger Source Select bits
111 = Internal counter ends sampling and starts conversion (auto convert)
110 = Reserved
101 = Reserved
100 = Reserved
011 = Motor Control PWM interval ends sampling and starts conversion
010 = General purpose Timer3 compare ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing SAMP bit ends sampling and starts conversion

bit 4-3 **Unimplemented:** Read as ‘0’

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

bit 1 **SAMP:** A/D Sample Enable bit
1 = At least one A/D sample/hold amplifier is sampling
0 = A/D sample/hold amplifiers are holding
When ASAM = 0, writing ‘1’ to this bit will start sampling.
When SSRC = 000, writing ‘0’ to this bit will end sampling and start conversion.

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion is done
0 = A/D conversion is not done
Clearing this bit will not effect any operation in progress.
Cleared by software or start of a new conversion.

Legend:

R = Readable bit	W = Writable bit	C = Clearable by software	
HC = Hardware clear	HS = Hardware set	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

Register 18-2: ADCON2: A/D Control Register 2

Upper Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
VCFG<2:0>			—	—	CSCNA	—	—
bit 15							bit 8

Lower Byte:							
R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI<3:0>				BUFM	ALTS
bit 7							bit 0

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

	A/D VREFH	A/D VREFL
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **Reserved:** User should write ‘0’ to this location

bit 11 **Unimplemented:** Read as ‘0’

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

bit 9-8 **Unimplemented:** Read as ‘0’

bit 7 **BUFS:** Buffer Fill Status bit
Only valid when BUFM = 1 (ADRES split into 2 x 8-word buffers)
1 = A/D is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7
0 = A/D is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF

bit 6 **Unimplemented:** Read as ‘0’

bit 5-2 **SMPI<3:0>:** Sample/Convert Sequences Per Interrupt Selection bits
1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence
1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence
. . . . .
0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence
0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **BUFM:** Buffer Mode Select bit
1 = Buffer configured as two 8-word buffers ADCBUF(15...8), ADCBUF(7...0)
0 = Buffer configured as one 16-word buffer ADCBUF(15...0)

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

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

Register 18-3: ADCON3: A/D Control Register 3

Upper Byte:							
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SAMC<4:0>				
bit 15							bit 8

Lower Byte:							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	ADCS<5:0>					
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **SAMC<4:0>:** Auto Sample Time bits
11111 = 31 T_{AD}
· · · · ·
00001 = 1 T_{AD}
00000 = 0 T_{AD}

bit 7 **ADRC:** A/D Conversion Clock Source bit
1 = A/D internal RC clock
0 = Clock derived from system clock

bit 6 **Unimplemented:** Read as ‘0’

bit 5-0 **ADCS<5:0>:** A/D Conversion Clock Select bits
111111 = T_{CY}/2 • (ADCS<5:0> + 1) = 32 • T_{CY}
· · · · · ·
000001 = T_{CY}/2 • (ADCS<5:0> + 1) = T_{CY}
000000 = T_{CY}/2 • (ADCS<5:0> + 1) = T_{CY}/2

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

Register 18-4: ADCHS: A/D Input Select Register

Upper Byte:

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CH0NB	CH0SB<3:0>			
bit 15							bit 8

Lower Byte:

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CH0NA	CH0SA<3:0>			
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit
Same definition as bit <4> (see **Note**).

bit 11-8 **CH0SB<3:0>:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bit
Same definition as bits <3:0> (see **Note**).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VREF-

bit 3-0 **CH0SA<3:0>:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bit
1111 = Channel 0 positive input is AN15
1110 = Channel 0 positive input is AN14
1101 = Channel 0 positive input is AN13
. . . . .
0001 = Channel 0 positive input is AN1
0000 = Channel 0 positive input is AN0

Note: The analog input multiplexer supports two input setting configurations, denoted MUX A and MUX B. ADCHS<15:8> determines the settings for MUX B, and ADCHS<7:0> determines the settings for MUX A. Both sets of control bits function identically.

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

Register 18-5: ADPCFG: A/D Port Configuration Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 15-0 **PCFG<15:0>:** Analog Input Pin Configuration Control bits
`1` = Analog input pin in Digital mode, port read input enabled, A/D input multiplexer input connected to AVSS
`0` = Analog input pin in Analog mode, port read input disabled, A/D samples pin voltage

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

Register 18-6: ADCSSL: A/D Input Scan Select Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

bit 15-0 **CSSL<15:0>:** A/D Input Pin Scan Selection bits
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

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

18.4 A/D Terminology and Conversion Sequence

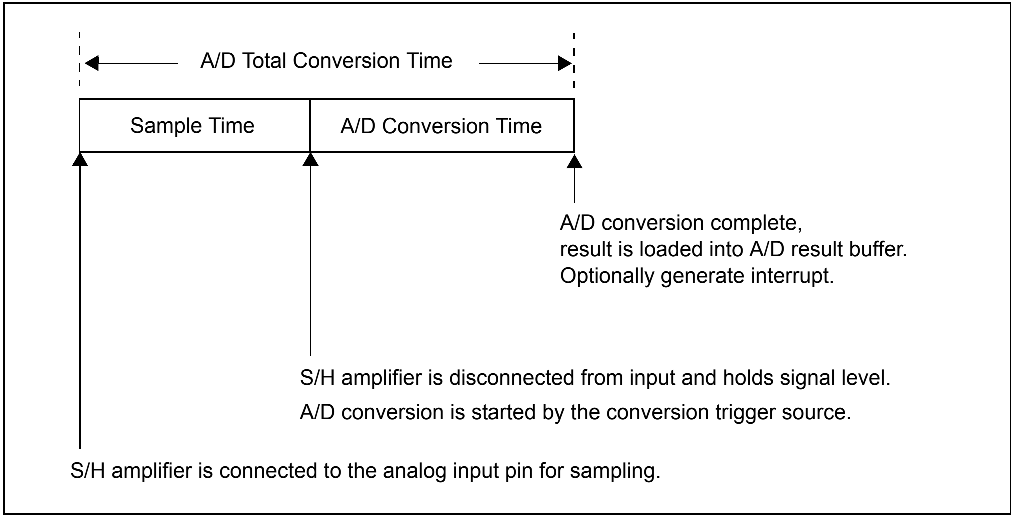
Figure 18-2 shows a basic conversion sequence and the terms that are used. A sampling of the analog input pin voltage is performed by sample and hold S/H amplifiers. The S/H amplifiers are also called S/H channels. The 12-bit A/D converter has one S/H channel, designated CH0. The S/H channel is connected to the analog input pins via the analog input multiplexer. The analog input multiplexer is controlled by the ADCHS register. There are two sets of multiplexer control bits in the ADCHS register that function identically. These two sets of control bits allow two different analog input multiplexer configurations to be programmed, which are called MUX A and MUX B. The A/D converter can optionally switch between the MUX A and MUX B configurations between conversions. The A/D converter can also optionally scan through a series of analog inputs.

Sample time is the time that the A/D module's S/H amplifier is connected to the analog input pin. The sample time may be started manually by setting the SAMP bit (ADCON1<1>) or started automatically by the A/D converter hardware. The sample time is ended manually by clearing the SAMP control bit in the user software or automatically by a conversion trigger source.

Conversion time is the time required for the A/D converter to convert the voltage held by the S/H amplifier. The A/D is disconnected from the analog input pin at the end of the sample time. The A/D converter requires one A/D clock cycle (TAD) to convert each bit of the result plus one additional clock cycle. A total of 14 TAD cycles are required to perform the complete conversion. When the conversion time is complete, the result is loaded into one of 16 A/D result registers (ADCBUF0...ADCBUFF), the S/H can be reconnected to the input pin, and a CPU interrupt may be generated.

The sum of the sample time and the A/D conversion time provides the total conversion time. There is a minimum sample time to ensure that the S/H amplifier will give the desired accuracy for the A/D conversion (see **Section 18.16 "A/D Sampling Requirements"**). Furthermore, there are multiple input clock options for the A/D converter. The user must select an input clock option that does not violate the minimum TAD specification.

Figure 18-2: A/D Sample/Conversion Sequence


A/D Total Conversion Time
Sample Time
A/D Conversion Time
A/D conversion complete,
result is loaded into A/D result buffer.
Optionally generate interrupt.
S/H amplifier is disconnected from input and holds signal level.
A/D conversion is started by the conversion trigger source.
S/H amplifier is connected to the analog input pin for sampling.

The start time for sampling can be controlled in software by setting the SAMP control bit. The start of the sampling time can also be controlled automatically by the hardware. When the A/D converter operates in the Auto Sample mode, the S/H amplifier(s) is reconnected to the analog input pin at the end of the conversion in the sample/convert sequence. The auto sample function is controlled by the ASAM control bit.

The conversion trigger source ends the sampling time and begins an A/D conversion or a sample/convert sequence. The conversion trigger source is selected by the SSRC control bits. The conversion trigger can be taken from a variety of hardware sources or can be controlled manually in software by clearing the SAMP control bit. One of the conversion trigger sources is an auto conversion. The time between auto conversions is set by a counter and the A/D clock. The Auto Sample mode and auto conversion trigger can be used together to provide endless automatic conversions without software intervention.

An interrupt may be generated at the end of each sample/convert sequence or multiple sample/convert sequences, as determined by the value of the SMPI control bits. The number of sample/convert sequences between interrupts can vary between 1 and 16.

18.5 A/D Module Configuration

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

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

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

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.

18.6 Selecting the Voltage Reference Source

The voltage references for A/D conversions are selected using the VCFG<2:0> control bits (ADCON2<15:13>). The upper voltage reference (VREFH) and the lower voltage reference (VREFL) may be the internal AVDD and AVSS voltage rails or the VREF+ and VREF- input pins.

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

The voltages applied to the external reference pins must meet certain specifications. Refer to the “Electrical Specifications” section of the device data sheet for further details.

Note: External VREF+ and VREF- pins must be selected for the conversion rates above 100 ksps. See **Section 18.22 “A/D Conversion Speeds”** for further details.

18.7 Selecting the A/D Conversion Clock

The A/D converter has a maximum rate at which conversions may be completed. An analog module clock, TAD, controls the conversion timing. The A/D conversion requires 14 clock periods (14 TAD). The A/D clock is derived from the device instruction clock.

The period of the A/D conversion clock is software selected using a six-bit counter. There are 64 possible options for TAD, specified by the ADCS<5:0> bits (ADCON3<5:0>). Equation 18-1 gives the TAD value as a function of the ADCS control bits and the device instruction cycle clock period, TCY.

Equation 18-1: A/D Conversion Clock Period

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

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

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 333.33 nsec (see **Section 18.22 "A/D Conversion Speeds"** for further details).

The A/D converter has a dedicated internal RC clock source that can be used to perform conversions. The internal RC clock source should be used when A/D conversions are performed while the dsPIC30F is in Sleep mode. The internal RC oscillator is selected by setting the ADRC bit (ADCON3<7>). When the ADRC bit is set, the ADCS<5:0> bits have no effect on the A/D operation.

18.8 Selecting Analog Inputs for Sampling

The Sample-and-Hold Amplifier has analog multiplexers (see Figure 18-1) on both its non-inverting and inverting inputs, to select which analog input(s) are sampled. Once the sample/convert sequence is specified, the ADCHS bits determine which analog inputs are selected for each sample.

Additionally, the selected inputs may vary on an alternating sample basis, or may vary on a repeated sequence of samples.

> **Note:** Different devices will have different numbers of analog inputs. Verify the analog input availability against the device data sheet.

18.8.1 Configuring Analog Port Pins

The ADPCFG register specifies the input condition of device pins used as analog inputs.

A pin is configured as analog input when the corresponding PCFGn bit (ADPCFG<n>) is clear. The ADPCFG register is clear at Reset, causing the A/D 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 it does not consume current.

The ADPCFG register and the TRISB register control the operation of the A/D port pins.

The port pins that are desired as analog inputs must have their corresponding TRIS bit set, specifying port input. If the I/O pin associated with an A/D input is configured as an output, TRIS bit is cleared, the pin is in Analog mode (ADPCFG<n> = 0) and the port digital output level (VOH or VOL) will be converted. After a device Reset, all TRIS bits are set.

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

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

18.8.2 Channel 0 Input Selection

The user may select any one of the up to 16 analog inputs to connect to the positive input of the channel. The CH0SA<3:0> bits (ADCHS<3:0>) normally select the analog input for the positive input of channel 0.

The user may select either VREF- or AN1 as the negative input of the channel. The CH0NA bit (ADCHS<4>) normally selects the analog input for the negative input of channel 0.

18.8.2.1 Specifying Alternating Channel 0 Input Selections

The ALTS bit (ADCON2<0>) causes the module to alternate between two sets of inputs that are selected during successive samples.

The inputs specified by CH0SA<3:0>, CH0NA, CHXSA and CHXNA<1:0> are collectively called the MUX A inputs. The inputs specified by CH0SB<3:0>, CH0NB, CHXSB and CHXNB<1:0> are collectively called the MUX B inputs. When the ALTS bit is ‘1’, the module will alternate between the MUX A inputs on one sample and the MUX B inputs on the subsequent sample.

For channel 0, if the ALTS bit is ‘0’, only the inputs specified by CH0SA<3:0> and CH0NA are selected for sampling.

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

18.8.2.2 Scanning Through Several Inputs

Channel 0 has the ability to scan through a selected vector of inputs. The CSCNA bit (ADCON2<10>) enables the CH0 channel inputs to be scanned across a selected number of analog inputs. When CSCNA is set, the CH0SA<3:0> bits are ignored.

The ADCSSL register specifies the inputs to be scanned. Each bit in the ADCSSL register corresponds to an analog input. Bit 0 corresponds to AN0, bit 1 corresponds to AN1 and so on. If a particular bit in the ADCSSL register is ‘1’, the corresponding input is part of the scan sequence. The inputs are always scanned from lower to higher numbered inputs, starting at the first selected channel after each interrupt occurs.

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

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

If the ALTS bit is ‘1’, the scanning only applies to the MUX A input selection. The MUX B input selection, as specified by the CH0SB<3:0>, will still select the alternating input. When the input selections are programmed in this manner, the input will alternate between a set of scanning inputs specified by the ADCSSL register and a fixed input specified by the CH0SB bits.

18.9 Enabling the Module

When the ADON bit (ADCON1<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.

In order to return to the Active mode from the Off mode, the user must wait for the analog stages to stabilize. For the stabilization time, refer to the "Electrical Characteristics" section of the device data sheet.

18.10 How to Start Sampling

18.10.1 Manual

Setting the SAMP bit (ADCON1<1>) causes the A/D to begin sampling. One of several options can be used to end sampling and complete the conversions. Sampling will not resume until the SAMP bit is once again set. For an example, see Figure 18-3**.**

18.10.2 Automatic

Setting the ASAM bit (ADCON1<2>) causes the A/D to automatically begin sampling a channel whenever a conversion is not active on that channel. One of several options can be used to end sampling and complete the conversions. Sampling on a channel resumes after the conversion of that channel completes. For an example, see Figure 18-4.

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 T_{PDU} parameter in the device datasheet), therefore, if automatic sampling is enabled, there is no guarantee than 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.

18.11 How to Stop Sampling and Start Conversions

The conversion trigger source will terminate sampling and start a selected sequence of conversions. The SSRC<2:0> bits (ADCON1<7:5>) select the source of the conversion trigger.

> **Note:** The available conversion trigger sources may vary depending on the dsPIC30F device variant. Please refer to the specific device data sheet for the available conversion trigger sources.

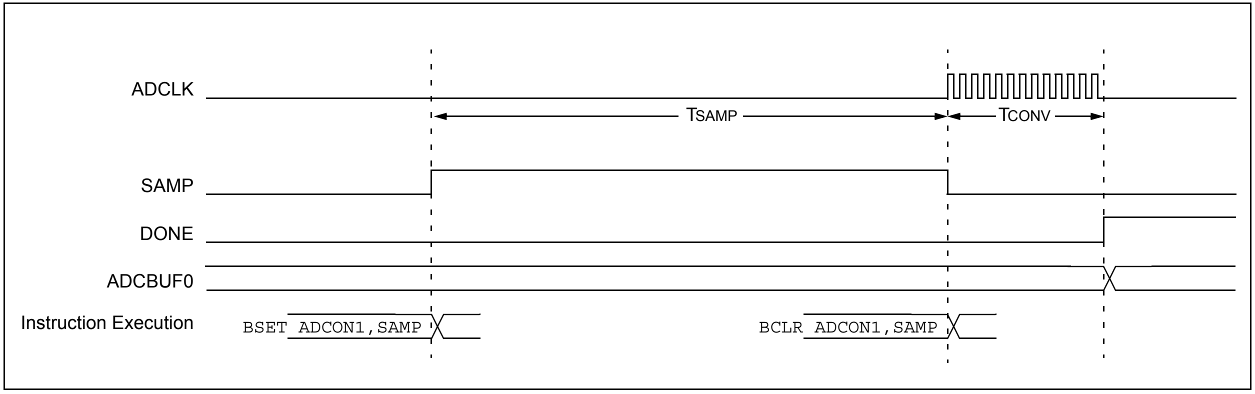
> **Note:** The SSRC selection bits should not be changed when the A/D module is enabled. If the user wishes to change the conversion trigger source, the A/D module should be disabled first by clearing the ADON bit (ADCON1<15>).

18.11.1 Manual

When SSRC<2:0> = 000, the conversion trigger is under software control. Clearing the SAMP bit (ADCON1<1>) starts the conversion sequence.

Figure 18-3 is an example where setting the SAMP bit initiates sampling and clearing the SAMP bit, terminates sampling and starts conversion. The user software must time the setting and clearing of the SAMP bit to ensure adequate sampling time of the input signal.

Figure 18-3: Converting 1 Channel, Manual Sample Start, Manual Conversion Start

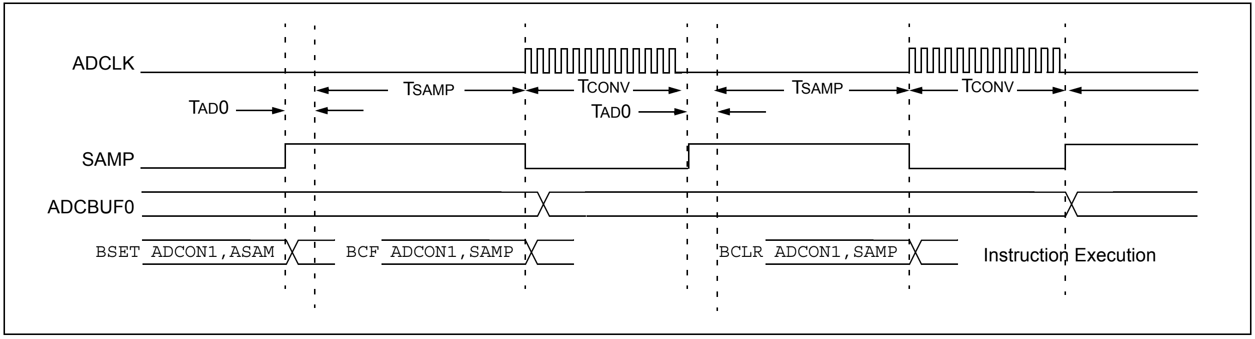


Example 18-1: Converting 1 Channel, Manual Sample Start, Manual Conversion Start Code Example

```
ADPCFG = 0xFFFB;               // all PORTB = Digital; RB2 = analog
ADCON1 = 0x0000;               // SAMP bit = 0 ends sampling ...
                               // and starts converting
ADCHS  = 0x0002;               // Connect RB2/AN2 as CH0 input  ..
                               // in this example RB2/AN2 is the input
ADCSSL = 0;
ADCON3 = 0x0002;               // Manual Sample, Tad = internal 2 Tcy
ADCON2 = 0;

ADCON1bits.ADON = 1;           // turn ADC ON
while (1)                      // repeat continuously
   {
   ADCON1bits.SAMP = 1;        // start sampling ...
   DelayNmSec(100);            // for 100 mS
   ADCON1bits.SAMP = 0;        // start Converting
   while (!ADCON1bits.DONE);   // conversion done?
   ADCValue = ADCBUF0;         // yes then get ADC value
   }                           // repeat
```

Figure 18-4 is an example where setting the ASAM bit initiates automatic sampling and clearing the SAMP bit, terminates sampling and starts conversion. After the conversion completes, the module will automatically return to a sampling state. The SAMP bit is automatically set at the start of the sample interval. The user software must time the clearing of the SAMP bit to ensure adequate sampling time of the input signal, understanding that the time between clearing of the SAMP bit includes the conversion time, as well as the sampling time.

Figure 18-4: Converting 1 Channel, Automatic Sample Start, Manual Conversion Start



18.11.2 Clocked Conversion Trigger

When SSRC<2:0> = 111, the conversion trigger is under A/D clock control. The SAMC bits (ADCON3<12:8>) select the number of TAD clock cycles between the start of sampling and the start of conversion. After the start of sampling, the module will count a number of TAD clocks specified by the SAMC bits.

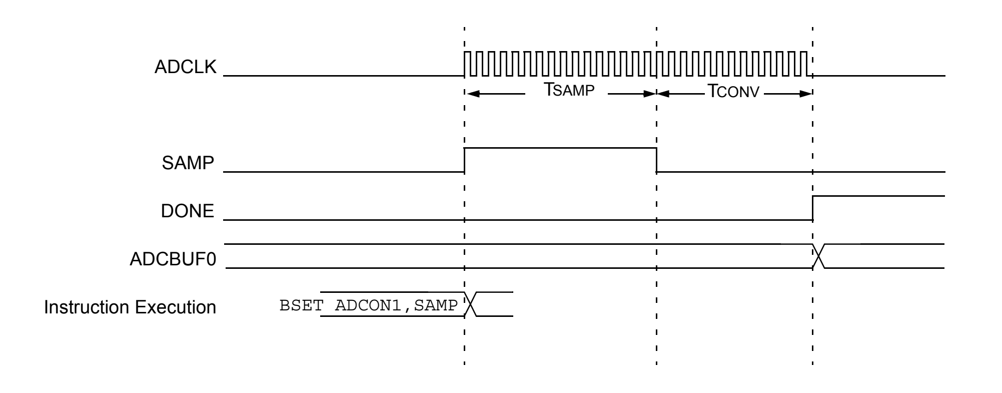
Equation 18-2: Clocked Conversion Trigger Time

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

SAMC must always be programmed for at least 1 clock cycle to ensure sampling requirements are met.

Figure 18-5 shows how to use the clocked conversion trigger with the sampling started by the user software.

Figure 18-5: Converting 1 Channel, Manual Sample Start, TAD Based Conversion Start



Example 18-2: Converting 1 Channel, Manual Sample Start, TAD Based Conversion Start Code Example

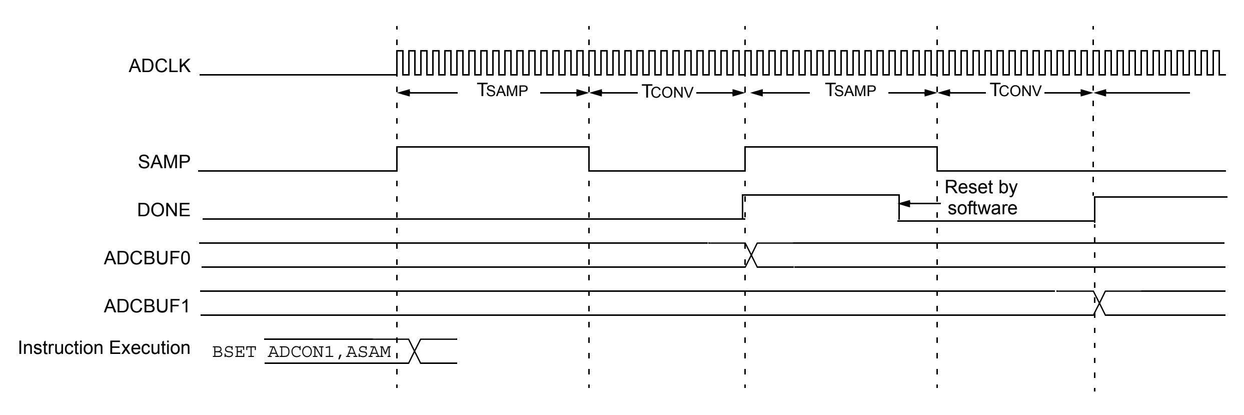
```
ADPCFG = 0xEFFF;            // all PORTB = Digital; RB12 = analog
ADCON1 = 0x00E0;            // SSRC bit = 111 implies internal
                            // counter ends sampling and starts
                            // converting.
 ADCHS = 0x000C;            // Connect RB12/AN12 as CH0 input  ..
                            // in this example RB12/AN12 is the input
ADCSSL = 0;
ADCON3 = 0x1F02;            // Sample time = 31Tad, Tad = internal 2
Tcy
ADCON2 = 0;

ADCON1bits.ADON = 1;        // turn ADC ON
while (1)                   // repeat continuously
  {
  ADCON1bits.SAMP = 1;      // start sampling then ...
                            // after 31Tad go to conversion
  while (!ADCON1bits.DONE);// conversion done?
  ADCValue = ADCBUF0;       // yes then get ADC value
  }                         // repeat// repeat
```

18.11.2.1 Free Running Sample Conversion Sequence

As shown in Figure 18-6, using the Auto-Convert Conversion Trigger mode (SSRC = `111`) in combination with the Auto-Sample Start mode (ASAM = `1`) allows the A/D module to schedule sample/conversion sequences with no intervention by the user or other device resources. This “Clocked” mode allows continuous data collection after module initialization.

> **Note:** This A/D configuration must be enabled for the conversion rate of 200 ksps (see **Section 18.22 “A/D Conversion Speeds”** for details).

Figure 18-6: Converting 1 Channel, Auto-Sample Start, TAD Based Conversion Start



18.11.2.2 Sample Time Considerations Using Clocked Conversion Trigger and Automatic Sampling

The user must ensure the sampling time exceeds the sampling requirements as outlined in **Section 18.16 “A/D Sampling Requirements”**.

Assuming that the module is set for automatic sampling and using a clocked conversion trigger, the sampling interval is specified by the SAMC bits.

18.11.3 Event Trigger Conversion Start

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

18.11.3.1 External INT Pin Trigger

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

18.11.3.2 General Purpose Timer Compare Trigger

The A/D is configured in this Trigger mode by setting SSRC<2:0> = `010`. When a match occurs between the 32-bit timer TMR3/TMR2 and the 32-bit Combined Period register PR3/PR2, a special ADC trigger event signal is generated by Timer3. This feature does not exist for the TMR5/TMR4 timer pair. Refer to **Section 12. “Timers”** for more details.

18.11.3.3 Motor Control PWM 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 A/D conversion results are acquired and the time when the duty cycle value is updated. Refer to **Section 15. “Motor Control PWM”** for more details.

18.11.3.4 Synchronizing A/D Operations to Internal or External Events

The modes where an external event trigger pulse ends sampling and starts conversion (SSRC = `001`, `010`, `011`) may be used in combination with auto sampling (ASAM = `1`) to cause the A/D to synchronize the sample conversion events to the trigger pulse source. For example, in Figure 18-8 where SSRC = `010` and ASAM = `1`, the A/D will always end sampling and start conversions synchronously with the timer compare trigger event. The A/D will have a sample conversion rate that corresponds to the timer comparison event rate.

Figure 18-7: Manual Sample Start, Conversion Trigger Based Conversion Start

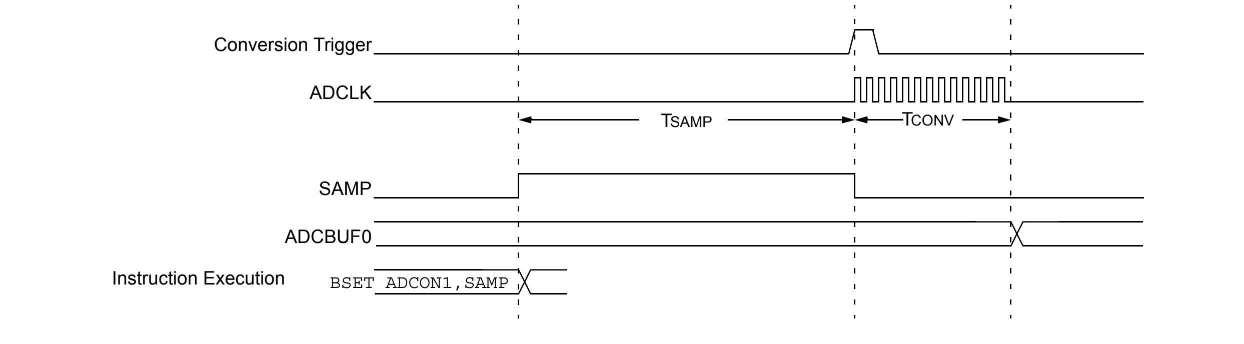
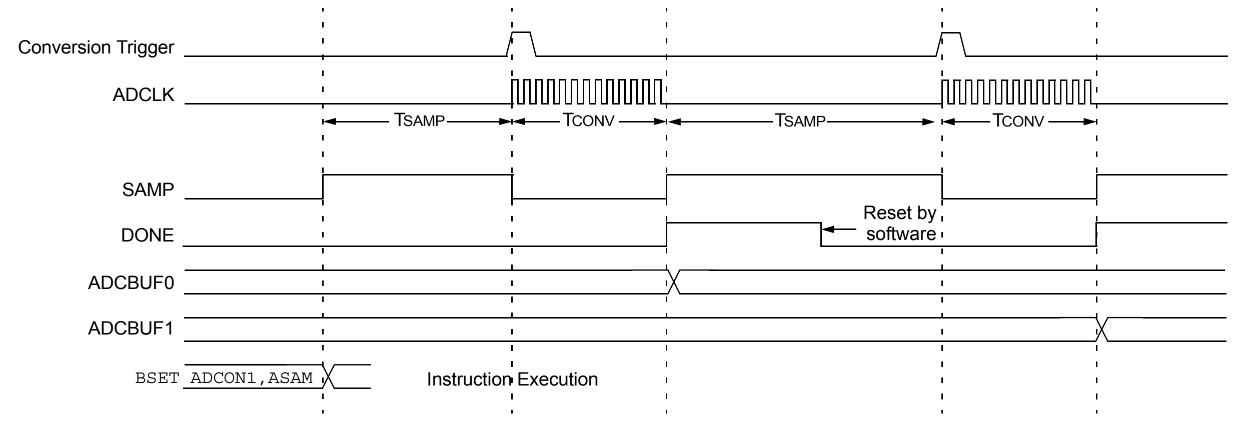


Figure 18-8: Auto-Sample Start, Conversion Trigger Based Conversion Start



18.11.3.5 Sample Time Considerations for Automatic Sampling/Conversion Sequences

Different sample/conversion sequences provide different available sampling times for the S/H channel to acquire the analog signal. The user must ensure the sampling time exceeds the sampling requirements, as outlined in **Section 18.16 "A/D Sampling Requirements"**.

Assuming that the module is set for automatic sampling and an external trigger pulse is used as the conversion trigger, the sampling interval is a portion of the trigger pulse interval.

The sampling time is the trigger pulse period, less the time required to complete the conversion.

Equation 18-3: Available Sampling Time, Sequential Sampling

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

$$T_{SMP} = T_{SEQ} - T_{CONV}$$

Note: TSEQ is the trigger pulse interval time.

18.12 Controlling Sample/Conversion Operation

The application software may poll the SAMP and CONV bits to keep track of the A/D operations, or the module can interrupt the CPU when conversions are complete. The application software may also abort A/D operations if necessary.

18.12.1 Monitoring Sample/Conversion Status

The SAMP (ADCON1<1>) and CONV (ADCON1<0>) bits indicate the sampling state and the conversion state of the A/D, respectively. Generally, when the SAMP bit clears indicating end of sampling, the CONV bit is automatically set indicating start of conversion. If both SAMP and CONV are ‘0’, the A/D is in an inactive state. In some operational modes, the SAMP bit may also invoke and terminate sampling and the CONV bit may terminate conversion.

18.12.2 Generating an A/D Interrupt

The SMPI<3:0> bits control the generation of interrupts. The interrupt will occur some number of sample/conversion sequences after starting sampling and re-occur on each equivalent number of samples.

The value specified by the SMPI bits will correspond to the number of data samples in the buffer, up to the maximum of 16.

Disabling the A/D interrupt is not done with the SMPI bits. To disable the interrupt, clear the ADIE analog module interrupt enable bit.

18.12.3 Aborting Sampling

Clearing the SAMP bit while in Manual Sampling mode will terminate sampling, but may also start a conversion if SSRC = 000.

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

18.12.4 Aborting a Conversion

Clearing the ADON bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the corresponding ADCBUF buffer location will continue to contain the value of the last completed conversion (or the last value written to the buffer).

18.13 Specifying How Conversion Results are Written into the Buffer

As conversions are completed, the module writes the results of the conversions into the A/D result buffer. This buffer is a RAM array of sixteen 12-bit words. The buffer is accessed through 16 address locations within the SFR space, named ADCBUF0...ADCBUFF.

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

18.13.1 Number of Conversions per Interrupt

The SMPI<3:0> bits (ADCON2<5:2>) will select how many A/D conversions will take place before the CPU is interrupted. This can vary from 1 sample per interrupt to 16 samples per interrupt. The A/D converter module always starts writing its conversion results at the beginning of the buffer, after each interrupt. For example, if SMPI<3:0> = 0000, the conversion results will always be written to ADCBUF0. In this example, no other buffer locations would be used.

18.13.2 Restrictions Due to Buffer Size

The user cannot program the SMPI bits to a value that specifies more than 8 conversions per interrupt when the BUFM bit (ADCON2<1>) is '1'. The BUFM bit function is described below.

18.13.3 Buffer Fill Mode

When the BUFM bit (ADCON2<1>) is '1', the 16-word results buffer (ADRES) will be split into two 8-word groups. The 8-word buffers will alternately receive the conversion results after each interrupt event. The initial 8-word buffer used after BUFM is set will be located at the lower addresses of ADCBUF. When BUFM is '0', the complete 16-word buffer is used for all conversion sequences.

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

If the processor cannot unload the buffer within the sample and conversion time, the BUFM bit should be '1'. For example, if SMPI<3:0> = 0111, then eight conversions will be loaded into 1/2 of the buffer, following which an interrupt will occur. The next eight conversions will be loaded into the other 1/2 of the buffer. The processor will, therefore, have the entire time between interrupts to move the eight conversions out of the buffer.

18.13.4 Buffer Fill Status

When the conversion result buffer is split using the BUFM control bit, the BUFS status bit (ADCON2<7>) indicates the half of the buffer that the A/D converter is currently filling. If BUFS = 0, then the A/D converter is filling ADCBUF0-ADCBUF7 and the user software should read conversion values from ADCBUF8-ADCBUFF. If BUFS = 1, the situation is reversed, and the user software should read conversion values from ADCBUF0-ADCBUF7.

18.14 Turning the A/D Module Off

The following sequence is recommended for turning the A/D module off:

1. Clear the ADON bit (ADCON1<15>).
2. Set the ADCMD bit (PMD1<0>).
3. Wait 2 instructions.
4. Clear the ADCMD bit (PMD1<0>).
5. Reinitialize the ADC module (optional).

The ADCMD bit is the peripheral module disable bit. Setting this bit prevents the A/D module from receiving a clock. Setting and then clearing this bit resets the A/D module.

18.15 Conversion Sequence Examples

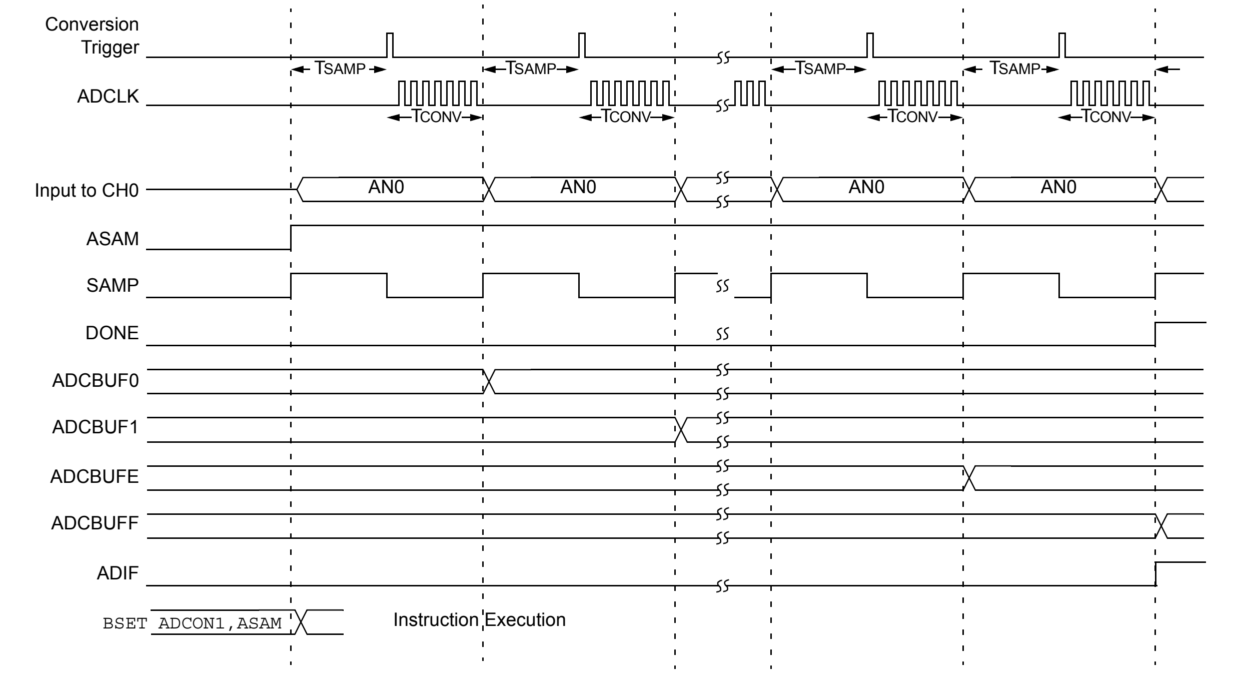
The following configuration examples show the A/D operation in different sampling and buffering configurations. In each example, setting the ASAM bit starts automatic sampling. A conversion trigger ends sampling and starts conversion.

18.15.1 Example: Sampling and Converting a Single Channel Multiple Times

Figure 18-9 and Table 18-1 illustrate a basic configuration of the A/D. In this case, one A/D input, AN0, will be sampled and converted. The results are stored in the ADCBUF buffer. This process repeats 16 times until the buffer is full and then the module generates an interrupt. The entire process will then repeat.

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

Figure 18-9: Converting One Channel 16 Times/Interrupt



Example 18-3: Sampling and Converting a Single Channel Multiple Times Code Example

```
ADPCFG = 0xFFFB;               // all PORTB = Digital; RB2 = analog
ADCON1 = 0x00E0;               // SSRC bit = 111 implies internal
                               // counter ends sampling and starts
                               // converting.
ADCHS = 0x0002;                // Connect RB2/AN2 as CH0 input  ..
                               // in this example RB2/AN2 is the input
ADCSSL = 0;
ADCON3 = 0x0F00;               // Sample time = 15Tad, Tad = internal Tcy/2
ADCON2 = 0x003C;               // Interrupt after every 16 samples

ADCON1bits.ADON = 1;           // turn ADC ON
while (1)                      // repeat continuously
   {
   ADCValue = 0;               // clear value
   ADC16Ptr = &ADCBUF0;        // initialize ADCBUF pointer
   IFS0bits.ADIF = 0;          // clear ADC interrupt flag
   ADCON1bits.ASAM = 1;        // auto start sampling
                               // for 31Tad then go to conversion
   while (!IFS0bits.ADIF);     // conversion done?
   ADCON1bits.ASAM = 0;        // yes then stop sample/convert
   for (count = 0; count < 16; count++) // average the 16 ADC value
                               ADCValue = ADCValue + *ADC16Ptr++;
   ADCValue = ADCValue >> 4;
   }                           // repeat
```

Table 18-1: Converting One Channel 16 Times/Interrupt

CONTROL BITS

Sequence Select	
SMPI<2:0> = 1111	Interrupt on 16th sample
BUFM = 0	Single 16-word result buffer
ALTS = 0	Always use MUX A input select
MUX A Input Select	
CH0SA<3:0> = 0000	Select AN0 for CH0+ input
CH0NA = 0	Select VREF- for CH0- input
CSCNA = 0	No input scan
CSSL<15:0> = n/a	Scan input select unused
MUX B Input Select	
CH0SB<3:0> = n/a	Channel CH0+ input unused
CH0NB = n/a	Channel CH0- input unused

OPERATION SEQUENCE
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x0
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x1
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x2
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x3
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x4
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x5
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x6
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x7
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x8
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x9
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xA
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xB
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xC
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xD
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xE
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0xF
Interrupt
Repeat

Buffer Address	**Buffer @ 1st Interrupt**	**Buffer @ 2nd Interrupt**	
ADCBUF0	AN0 sample 1	AN0 sample 17	
ADCBUF1	AN0 sample 2	AN0 sample 18	
ADCBUF2	AN0 sample 3	AN0 sample 19	
ADCBUF3	AN0 sample 4	AN0 sample 20	
ADCBUF4	AN0 sample 5	AN0 sample 21	
ADCBUF5	AN0 sample 6	AN0 sample 22	
ADCBUF6	AN0 sample 7	AN0 sample 23	
ADCBUF7	AN0 sample 8	AN0 sample 24	• • •
ADCBUF8	AN0 sample 9	AN0 sample 25	
ADCBUF9	AN0 sample 10	AN0 sample 26	
ADCBUFA	AN0 sample 11	AN0 sample 27	
ADCBUFB	AN0 sample 12	AN0 sample 28	
ADCBUFC	AN0 sample 13	AN0 sample 29	
ADCBUFD	AN0 sample 14	AN0 sample 30	
ADCBUFE	AN0 sample 15	AN0 sample 31	
ADCBUFF	AN0 sample 16	AN0 sample 32	

18.15.2 Example: A/D Conversions While Scanning Through All Analog Inputs

Figure 18-10 and Table 18-2 illustrate a typical setup, where all available analog input channels are sampled and converted. The set CSCNA bit specifies scanning of the A/D inputs to the CH0 positive input. Other conditions are similar to Subsection 18.15.1.

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

Figure 18-10: Scanning Through 16 Inputs/Interrupt

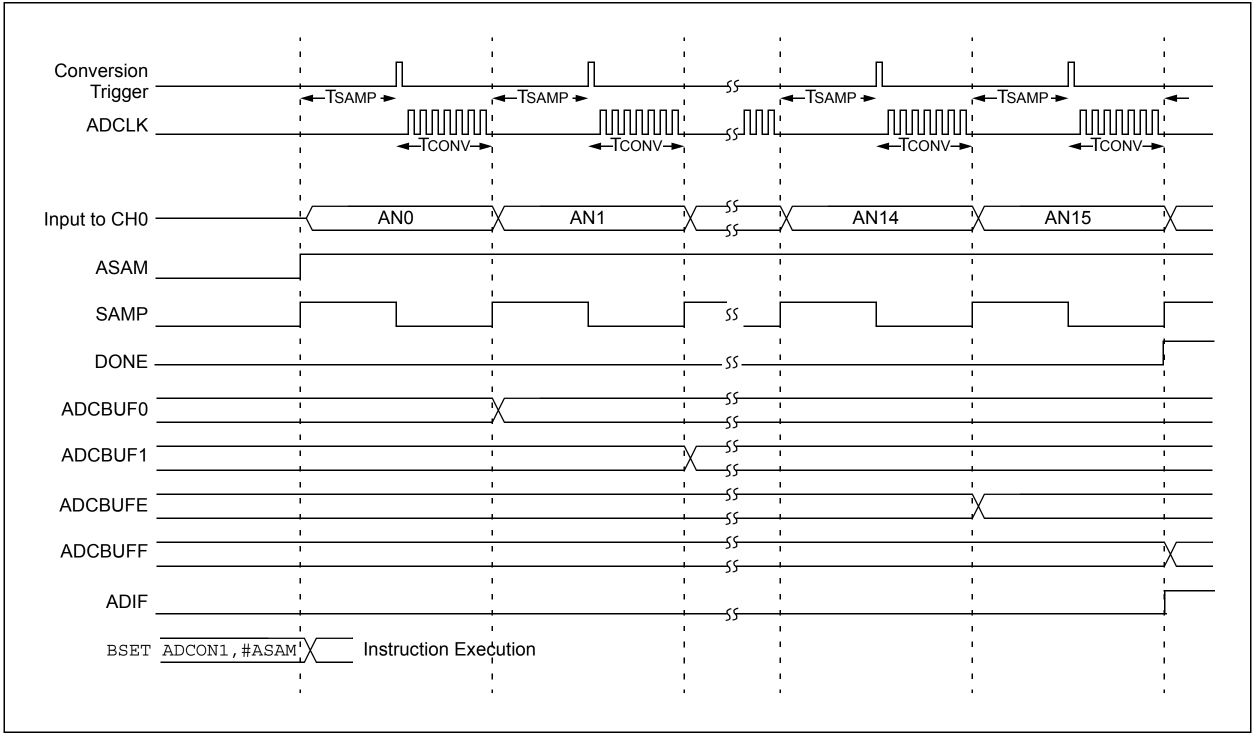


Table 18-2: Scanning Through 16 Inputs/Interrupt

CONTROL BITS	
Sequence Select	
SMPI<2:0> = 1111	Interrupt on 16th sample
BUFM = 0	Single 16-word result buffer
ALTS = 0	Always use MUX A input select
MUX A Input Select	
CH0SA<3:0> = n/a	Override by CSCNA
CH0NA = 0	Select VREF- for CH0- input
CSCNA = 1	Scan CH0+ Inputs
CSSL<15:0> = 1111 1111 1111 1111	Scan all inputs
MUX B Input Select	
CH0SB<3:0> = n/a	Channel CH0+ input unused
CH0NB = n/a	Channel CH0- input unused

OPERATION SEQUENCE
Sample MUX A Inputs: AN0 -> CH0
Convert CH0, Write Buffer 0x0
Sample MUX A Inputs: AN1 -> CH0
Convert CH0, Write Buffer 0x1
Sample MUX A Inputs: AN2 -> CH0
Convert CH0, Write Buffer 0x2
Sample MUX A Inputs: AN3 -> CH0
Convert CH0, Write Buffer 0x3
Sample MUX A Inputs: AN4 -> CH0
Convert CH0, Write Buffer 0x4
Sample MUX A Inputs: AN5 -> CH0
Convert CH0, Write Buffer 0x5
Sample MUX A Inputs: AN6 -> CH0
Convert CH0, Write Buffer 0x6
Sample MUX A Inputs: AN7 -> CH0
Convert CH0, Write Buffer 0x7
Sample MUX A Inputs: AN8 -> CH0
Convert CH0, Write Buffer 0x8
Sample MUX A Inputs: AN9 -> CH0
Convert CH0, Write Buffer 0x9
Sample MUX A Inputs: AN10 -> CH0
Convert CH0, Write Buffer 0xA
Sample MUX A Inputs: AN11 -> CH0
Convert CH0, Write Buffer 0xB
Sample MUX A Inputs: AN12 -> CH0
Convert CH0, Write Buffer 0xC
Sample MUX A Inputs: AN13 -> CH0
Convert CH0, Write Buffer 0xD
Sample MUX A Inputs: AN14 -> CH0
Convert CH0, Write Buffer 0xE
Sample MUX A Inputs: AN15 -> CH0
Convert CH0, Write Buffer 0xF
Interrupt
Repeat

Buffer Address	Buffer @ 1st Interrupt	Buffer @ 2nd Interrupt
ADCBUF0	AN0 sample 1	AN0 sample 17
ADCBUF1	AN1 sample 2	AN1 sample 18
ADCBUF2	AN2 sample 3	AN2 sample 19
ADCBUF3	AN3 sample 4	AN3 sample 20
ADCBUF4	AN4 sample 5	AN4 sample 21
ADCBUF5	AN5 sample 6	AN5 sample 22
ADCBUF6	AN6 sample 7	AN6 sample 23
ADCBUF7	AN7 sample 8	AN7 sample 24
ADCBUF8	AN8 sample 9	AN8 sample 25
ADCBUF9	AN9 sample 10	AN9 sample 26
ADCBUFA	AN10 sample 11	AN10 sample 27
ADCBUFB	AN11 sample 12	AN11 sample 28
ADCBUFC	AN12 sample 13	AN12 sample 29
ADCBUFD	AN13 sample 14	AN13 sample 30
ADCBUFE	AN14 sample 15	AN14 sample 31
ADCBUFF	AN15 sample 16	AN15 sample 32

• • •

18.15.3 Example: Using Dual 8-Word Buffers

Refer to Subsection 17.15.4 in **Section 17. “10-bit A/D Converter”** for an example that uses dual buffers.

18.15.4 Example: Using Alternating MUX A, MUX B Input Selections

See Subsection 17.15.5 in **Section 17. “10-bit A/D Converter”** for an example that uses the MUX A and MUX B input selections.

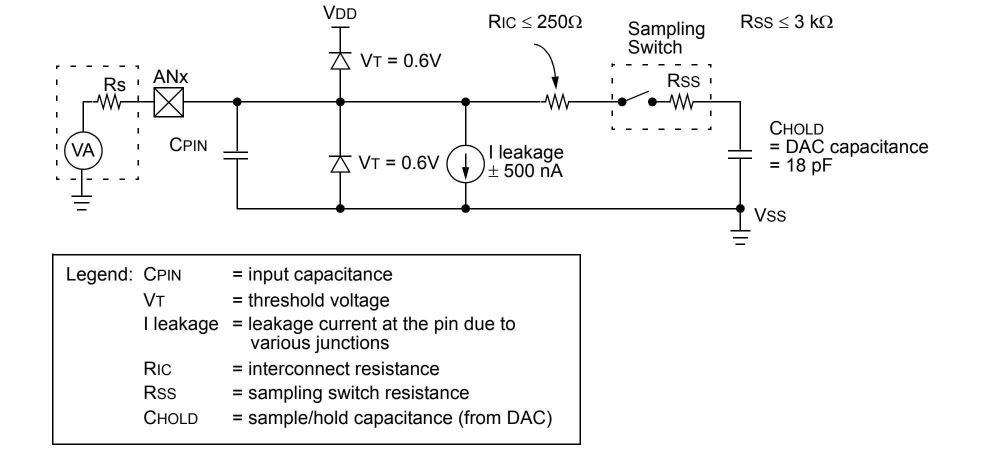
18.16 A/D Sampling Requirements

The analog input model of the 12-bit A/D converter is shown in Figure 18-11. The total sampling time for the A/D is a function of the internal amplifier settling time and the holding capacitor charge time.

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

At least 1 TAD time period should be allowed between conversions for the sample time. For more details, see the device electrical specifications.

Figure 18-11: 12-bit A/D Converter Analog Input Model


VDD
VT = 0.6V
RIC ≤ 250Ω
Sampling Switch
RSS ≤ 3 kΩ
Rs
ANx
RSS
VA
CPIN
VT = 0.6V
I leakage
± 500 nA
CHOLD
= DAC capacitance
= 18 pF
VSS
Legend: CPIN = input capacitance
VT = threshold voltage
I leakage = leakage current at the pin due to various junctions
RIC = interconnect resistance
RSS = sampling switch resistance
CHOLD = sample/hold capacitance (from DAC)

Note: CPIN value depends on device package and is not tested. Effect of CPIN negligible if Rs ≤ 2.5 kΩ.

18.17 Reading the A/D Result Buffer

The RAM is 12 bits wide, but the data is automatically formatted to one of four selectable formats when a read from the buffer is performed. 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 18-12 shows the data output formats that can be selected using the FORM<1:0> control bits.

Figure 18-12: A/D Output Data Formats

RAM Contents:					d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Integer	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Signed Integer	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Fractional	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
Signed Fractional (1.15)	$\overline{d11}$	d10	d09	d08	d07	d04	d03	d02	d01	d00	d01	d00	0	0	0	0

Table 18-3: Numerical Equivalents of Various Result Codes

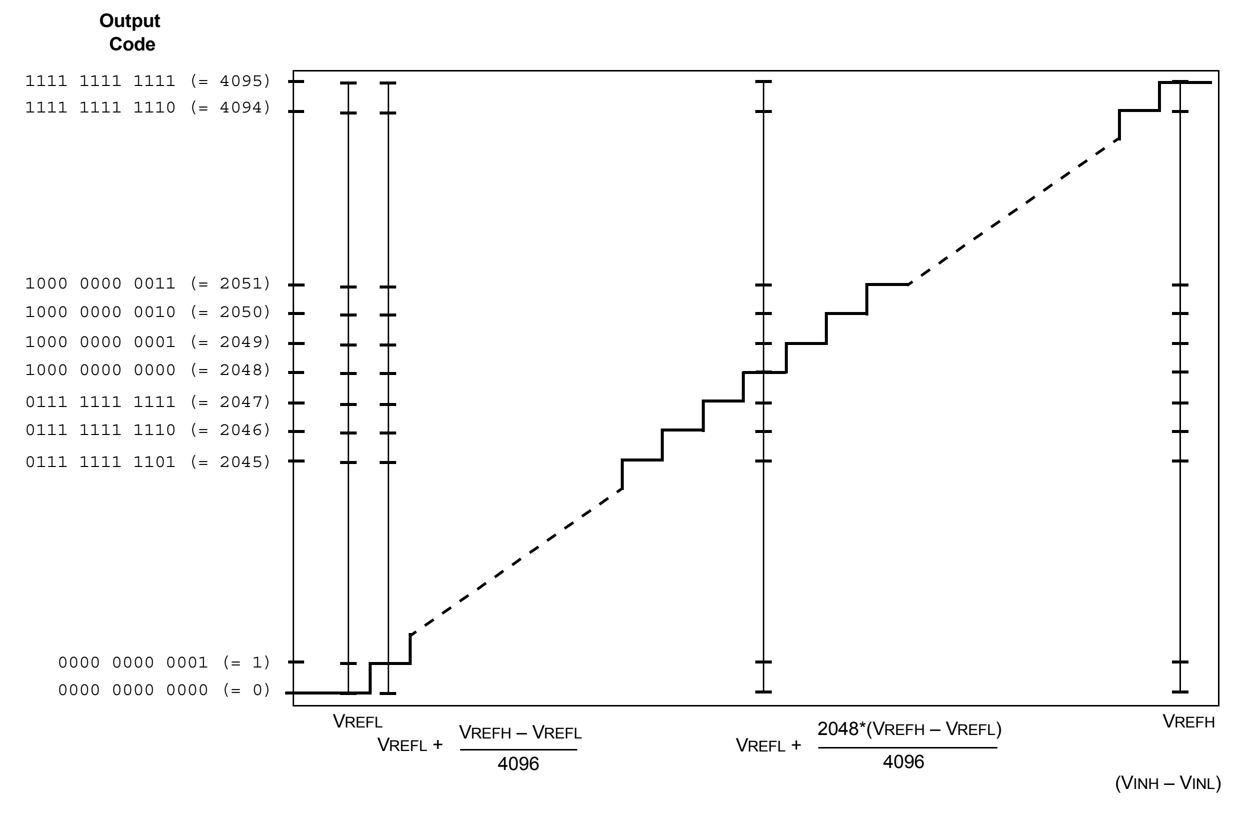
VIN/VREF	12-bit Output Code	16-bit Unsigned Integer Format	16-bit Signed Integer Format	16-bit Unsigned Fractional Format	16-bit Signed Fractional Format
4095/4096	1111 1111 1111	0000 1111 1111 1111 = 4095	0000 0111 1111 1111 = 2047	1111 1111 1111 0000 = 0.9998	0111 1111 1111 0000 = 0.9995
4094/4096	1111 1111 1110	0000 1111 1111 1110 = 4094	0000 0111 1111 1110 = 2046	1111 1111 1110 0000 = 0.9995	0111 1111 1110 0000 = 0.9990
•••					
2049/4096	1000 0000 0001	0000 1000 0000 0001 = 2049	0000 0000 0000 0001 = 1	1000 0000 0001 0000 = 0.5002	0000 0000 0001 0000 = 0.0005
2048/4096	1000 0000 0000	0000 1000 0000 0000 = 2048	0000 0000 0000 0000 = 0	1000 0000 0000 0000 = 0.500	0000 0000 0000 0000 = 0.000
2047/4096	0111 1111 1111	0000 0111 1111 1111 = 2047	1111 1111 1111 1111 = -1	0111 1111 1111 0000 = 0.4998	1111 1111 1111 0000 = -0.0005
•••					
1/4096	0000 0000 0001	0000 0000 0000 0001 = 1	1111 1000 0000 0001 = -2047	0000 0000 0001 0000 = 0.0002	1000 0000 0001 0000 = -0.9995
0/4096	0000 0000 0000	0000 0000 0000 0000 = 0	1111 1000 0000 0000 = -2048	0000 0000 0000 0000 = 0.000	1000 0000 0000 0000 = -1.000

18.18 Transfer Function

The ideal transfer function of the A/D converter is shown in Figure 18-13. The difference of the input voltages (VINH – VINL) is compared to the reference (VREFH – VREFL).

- The first code transition occurs when the input voltage is (VREFH – VREFL/8192) or 0.5 LSb.
- The `00 0000 0001` code is centered at (VREFH – VREFL/4096) or 1.0 LSb.
- The `10 0000 0000` code is centered at (2048*(VREFH – VREFL)/4096).
- An input voltage less than (1*(VREFH – VREFL)/8192) converts as `00 0000 0000`.
- An input greater than (8192*(VREFH – VREFL)/8192) converts as `11 1111 1111`.

Figure 18-13: A/D Transfer Function



18.19 A/D Accuracy/Error

Refer to **Section 18.27 "Related Application Notes"** for a list of documents that discuss A/D accuracy.

18.20 Connection Considerations

Since the analog inputs employ ESD protection, they have diodes to VDD and VSS. This requires that the analog input must be between VDD and VSS. If the input voltage exceeds this range by greater than 0.3V (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.

18.21 Initialization

Example 18-4 shows a simple initialization code example for the A/D module.

In this particular configuration, all 16 analog input pins, AN0-AN15, are set up as analog inputs. Operation in Idle mode is disabled, output data is in unsigned fractional format, and AVDD and AVSS are used for VREFH and VREFL. The start of sampling, as well as the start of conversion (conversion trigger), are performed manually in software. Scanning of inputs is disabled and an interrupt occurs after every sample/convert sequence (1 conversion result). The A/D conversion clock is TCY/2; AN0 is converted.

Since sampling is started manually by setting the SAMP bit (ADCON1<1>) after each conversion is complete, the auto-sample time bits, SAMC<4:0> (ADCON3<12:8>), are ignored. Moreover, since the start of conversion (i.e., end of sampling) is also triggered manually, the SAMP bit needs to be cleared each time a new sample needs to be converted.

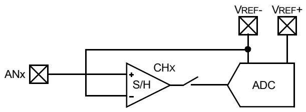
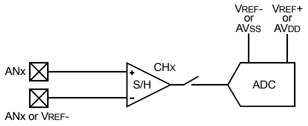
Example 18-4: A/D Initialization Code Example

```
CLR     ADPCFG            ; Configure A/D port,
                          ; all input pins are analog
MOV     #0x2200,W0
MOV     W0,ADCON1         ; Configure sample clock source
                          ; and conversion trigger mode.
                          ;    Unsigned Fractional format,
                          ;    Manual conversion trigger,
                          ;    Manual start of sampling,
                          ;    No operation in IDLE mode.
CLR     ADCON2            ; Configure A/D voltage reference
                          ; and buffer fill modes.
                          ;    VREF from AVDD and AVSS,
                          ;    Inputs are not scanned,
                          ;    Interrupt every sample

CLR     ADCON3            ; Configure A/D conversion clock

CLR     ADCHS             ; Configure input channels,
                          ;    CH0+ input is AN0.
                          ;    CHO- input is VREFL  (AVss)
CLR     ADCSSL            ; No inputs are scanned.

BCLR    IFS0,#ADIF        ; Clear A/D conversion interrupt flag

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

BSET    IEC0,#ADIE        ; Enable A/D conversion interrupt

BSET    ADCON1,#ADON      ; Turn on A/D
BSET    ADCON1,#SAMP      ; Start sampling the input
CALL    DELAY             ; Ensure the correct sampling time has
                          ; elapsed before starting conversion.
BCLR    ADCON1,#SAMP      ; End A/D Sampling and start Conversion
  :                       ; The DONE bit is set by hardware when
                            conversion sequence is complete.
  :                       ; The ADIF bit will be set.
```

18.22 A/D Conversion Speeds

The dsPIC30F 12-bit A/D converter specifications permit a maximum of 200 ksps sampling rate. The table below summarizes the conversion speeds for the dsPIC30F 12-bit A/D converter and the required operating conditions.

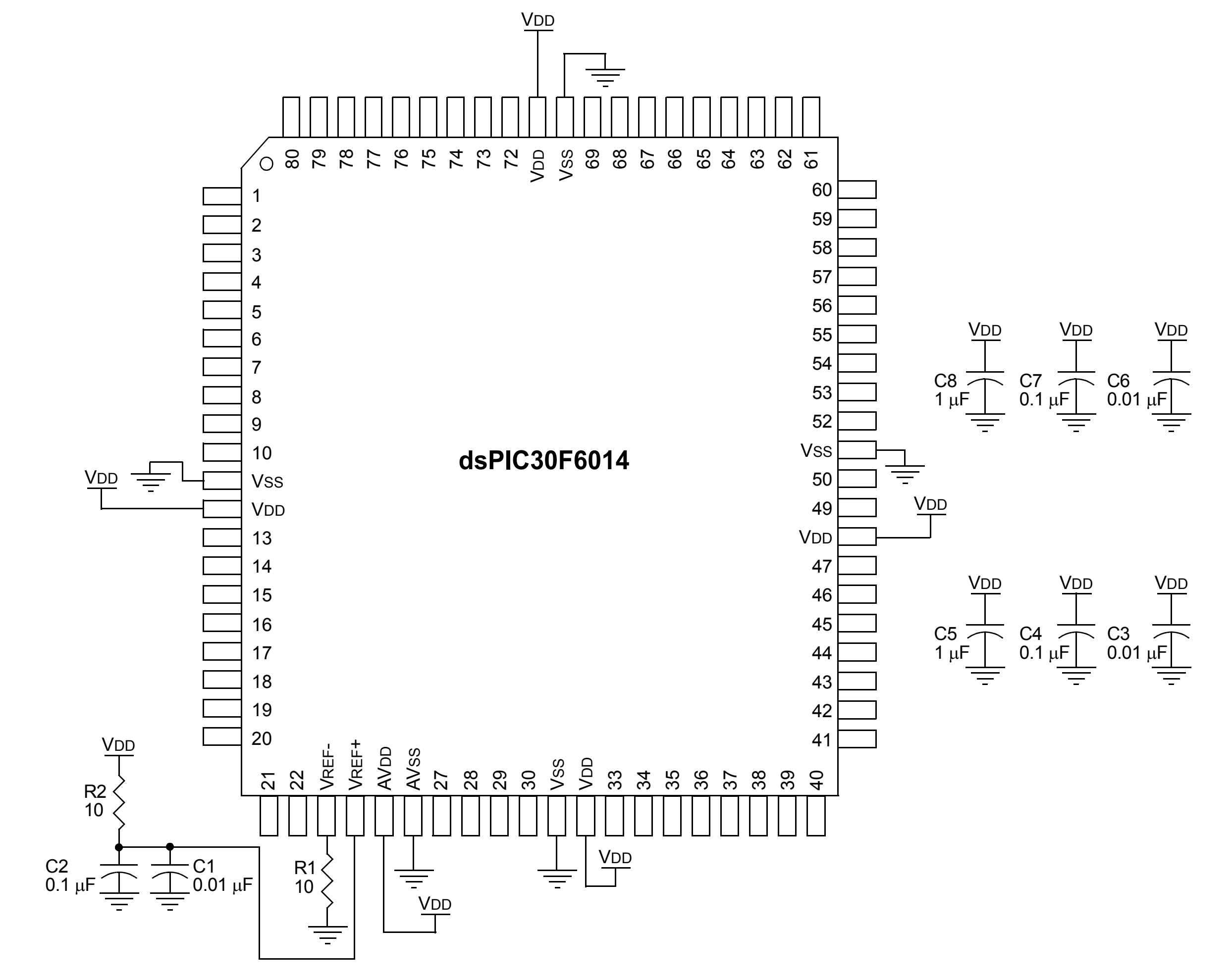
TABLE 18-4: 12-BIT A/D CONVERTER EXTENDED CONVERSION RATES

dsPIC30F 10-bit A/D Converter Conversion Rates						
A/D Speed	**TAD Minimum**	**Sampling Time Min**	**R_S Max**	**VDD**	**Temperature**	**A/D Channels Configuration**
Up to 200 ksps[1]	333.33 ns	1 TAD	2.5 kΩ	4.5V to 5.5V	-40°C to +85°C	
Up to 100 ksps	666.67 ns	1 TAD	2.5 kΩ	3.0V to 5.5V	-40°C to +125°C	 VREF- or AVSS VREF+ or AVDD ANx CHx + S/H - ADC ANx or VREF-

Note 1: External VREF- and VREF+ pins must be used for correct operation. See Figure 18-14 for recommended circuit.

The following figure depicts the recommended circuit for the conversion rates above 100 ksps. The dsPIC30F6014 is shown as an example.

Figure 18-14: A/D Converter Voltage Reference Schematic



The configuration procedures below give the required setup values for the conversion speeds above 100 ksps.

18.22.1 200 ksps Configuration Guideline

The following configuration items are required to achieve a 200 ksps conversion rate.

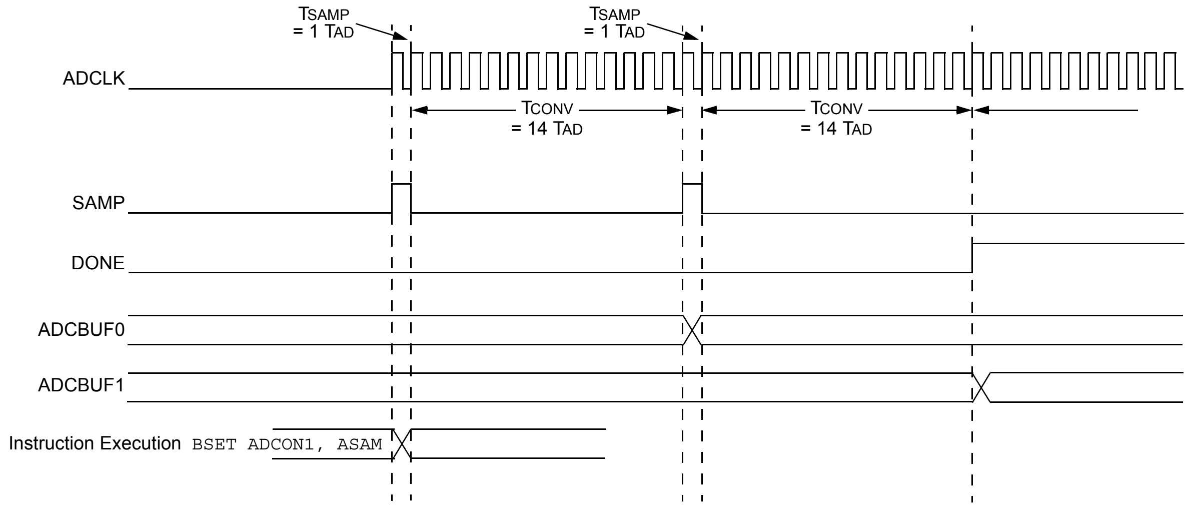
- Comply with conditions provided in Table 18-4.
- Connect external VREF+ and VREF- pins following the recommended circuit shown in Figure 18-14.
- Set SSRC<2:0> = `111` in the ADCON1 register to enable the auto-convert option.
- Enable automatic sampling by setting the ASAM control bit in the ADCON1 register.
- Write the SMPI<3:0> control bits in the ADCON2 register for the desired number of conversions between interrupts.
- Configure the A/D clock period to be:

$$\frac{1}{(14 + 1) \times 200{,}000} = 333.33 \text{ ns}$$

  by writing to the ADCS<5:0> control bits in the ADCON3 register.
- Configure the sampling time to be 1 TAD by writing: SAMC<4:0> = `00001`.

The following figure shows the timing diagram of the A/D running at 200 ksps. The TAD selection in conjunction with the guidelines described above allows a conversion speed of 200 ksps. See Example 18-1 for code example.

Figure 18-15: Converting 1 Channel at 200 ksps, Auto-Sample Start, 1 TAD Sampling Time



Example 18-1: Converting at 200 ksps, Auto-Sample Start, 1 TAD Sampling Time Code Example

```
ADPCFG  =   0xFFFB;             //  all PORTB = Digital; RB2 = analog
ADCON1  =   0x00E0;             //  SSRC bit = 111 implies internal
                                //  counter ends sampling and starts
                                //  converting.
ADCHS   =   0x0002;             //  Connect RB2/AN2 as CH0 input
                                //  in this example RB2/AN2 is the input
ADCSSL  =   0;
ADCON3  =   0x0113;             //  Sample time = 1Tad, Tad = 333.33 ns @ 30 MIPS
                                //  which will give 1 / (15 * 333.33 ns) = 200 ksps
ADCON2  =   0x6004;             //  Select external VREF+ and VREF- pins
                                //  Interrupt after every 2 samples
ADCON1bits.ADON = 1;            //  turn ADC ON
while (1)                       //  repeat continuously
{
    ADCValue = 0;                       //  clear value
    ADC16Ptr = &ADCBUF0;                //  initialize ADCBUF pointer
    IFS0bits.ADIF = 0;                  //  clear ADC interrupt flag
    ADCON1bits.ASAM = 1;                //  auto start sampling
                                        //  for 31Tad then go to conversion
    while (!IFS0bits.ADIF);             //  conversion done?
    ADCON1bits.ASAM = 0;                //  yes then stop sample/convert
    for (count = 0; count <2; count++)//  average the 2 ADC value
    ADCValue = ADCValue + *ADC16Ptr++;
    ADCValue = ADCValue >> 1;
    }                                   //  repeat
```

18.23 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.

18.23.1 CPU Sleep Mode Without RC A/D Clock

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

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

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

18.23.2 CPU Sleep Mode With RC A/D Clock

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

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

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

To minimize the effects of digital noise on the A/D module operation, the user should select a conversion trigger source that ensures the A/D conversion will take place in Sleep mode. The automatic conversion trigger option can be used for sampling and conversion in Sleep (SSRC<2:0> = 111). To use the automatic conversion option, the ADON bit should be set in the instruction prior to the PWRSAV instruction.

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

18.23.3 A/D Operation During CPU Idle Mode

For the A/D, the ADSIDL bit (ADCON1<13>) selects if the module will stop on Idle or continue on Idle. If ADSIDL = 0, the module will continue normal operation when the device enters Idle mode. If the A/D interrupt is enabled (ADIE = 1), the device will wake-up from Idle mode when the A/D interrupt occurs. Program execution will resume at the A/D Interrupt Service Routine if the A/D interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the PWRSAV instruction that placed the device in Idle mode.

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

18.24 Effects of a Reset

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

The values in the ADCBUF registers are not initialized during a Power-on Reset. ADCBUF0...ADCBUFF will contain unknown data.

18.25 Special Function Registers Associated with the 12-bit A/D Converter

The following table lists dsPIC30F 12-bit A/D Converter Special Function Registers, including their addresses and formats. All unimplemented registers and/or bits within a register read as zeros.

Table 18-4: ADC Register Map

File Name	ADR	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	Reset States
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	—	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
ADCBUF0	0280	ADC Data Buffer 0																uuuu uuuu uuuu uuuu
ADCBUF1	0282	ADC Data Buffer 1																uuuu uuuu uuuu uuuu
ADCBUF2	0284	ADC Data Buffer 2																uuuu uuuu uuuu uuuu
ADCBUF3	0286	ADC Data Buffer 3																uuuu uuuu uuuu uuuu
ADCBUF4	0288	ADC Data Buffer 4																uuuu uuuu uuuu uuuu
ADCBUF5	028A	ADC Data Buffer 5																uuuu uuuu uuuu uuuu
ADCBUF6	028C	ADC Data Buffer 6																uuuu uuuu uuuu uuuu
ADCBUF7	028E	ADC Data Buffer 7																uuuu uuuu uuuu uuuu
ADCBUF8	0290	ADC Data Buffer 8																uuuu uuuu uuuu uuuu
ADCBUF9	0292	ADC Data Buffer 9																uuuu uuuu uuuu uuuu
ADCBUFA	0294	ADC Data Buffer 10																uuuu uuuu uuuu uuuu
ADCBUFB	0296	ADC Data Buffer 11																uuuu uuuu uuuu uuuu
ADCBUFC	0298	ADC Data Buffer 12																uuuu uuuu uuuu uuuu
ADCBUFD	029A	ADC Data Buffer 13																uuuu uuuu uuuu uuuu
ADCBUFE	029C	ADC Data Buffer 14																uuuu uuuu uuuu uuuu
ADCBUFF	029E	ADC Data Buffer 15																uuuu uuuu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	—	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>					ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	—	—	—	CH0NB	CH0SB<3:0>				—	—	—	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	CSSL<15:0>																0000 0000 0000 0000

Legend: u = unknown

Note: All interrupt sources and their associated control bits may not be available on a particular device. Refer to the device data sheet for details.

18.26 Design Tips

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

Answer:

1. Make sure you are meeting all of the timing specifications. If you are turning the module off and on, there is a minimum delay you must wait before taking a sample. If you are changing input channels, there is a minimum delay you must wait for this as well, and finally, there is TAD, which is the time selected for each bit conversion. This is selected in ADCON3 and should be within a certain range, as specified in the Electrical Characteristics. If TAD is too short, the result may not be fully converted before the conversion is terminated, and if TAD is made too long, the voltage on the sampling capacitor can decay before the conversion is complete. These timing specifications are provided in the “Electrical Specifications” section of the device data sheets.
2. Often, the source impedance of the analog signal is high (greater than 10 kΩ), so the current drawn from the source by leakage, and to charge the sample capacitor, can affect accuracy. If the input signal does not change too quickly, try putting a 0.1 μF capacitor on the analog input. This capacitor will charge to the analog voltage being sampled and supply the instantaneous current needed to charge the 18 pF internal holding capacitor.
3. Put the device into Sleep mode before the start of the A/D conversion. The RC clock source selection is required for conversions in Sleep mode. This technique increases accuracy, because digital noise from the CPU and other peripherals is minimized.

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

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.

18.27 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 dsPIC30F Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the 12-bit A/D Converter module are:

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

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC30F Family of devices.

18.28 Revision History

Revision A

This is the initial released revision of this document.

Revision B

To reflect editorial and technical content revisions for the dsPIC30F 12-bit A/D Converter module.

Revision C

This revision incorporates all known errata at the time of this document update.

Revision D

This revision includes the extended conversion rate guidelines.

Revision E

- Added a paragraph to **18.10.2 “Automatic”**, which references automatic sampling and the TPDU parameter.
- Added **18.14 “Turning the A/D Module Off”**.
- Incorporated minor changes to the document text.

NOTES: