Section 17. 10-Bit A/D Converter

HIGHLIGHTS

This section of the manual contains the following topics:

17.1 INTRODUCTION

The PIC32MX 10-bit Analog-to-Digital (A/D) converter (or ADC) includes the following features:

- Successive Approximation Register (SAR) conversion
- Up to 16 analog input pins
- External voltage reference input pins
- One unipolar differential Sample-and-Hold Amplifier (SHA)
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 16 word conversion result buffer
- Selectable Buffer Fill modes
- Eight conversion result format options
- Operation during CPU SLEEP and IDLE modes

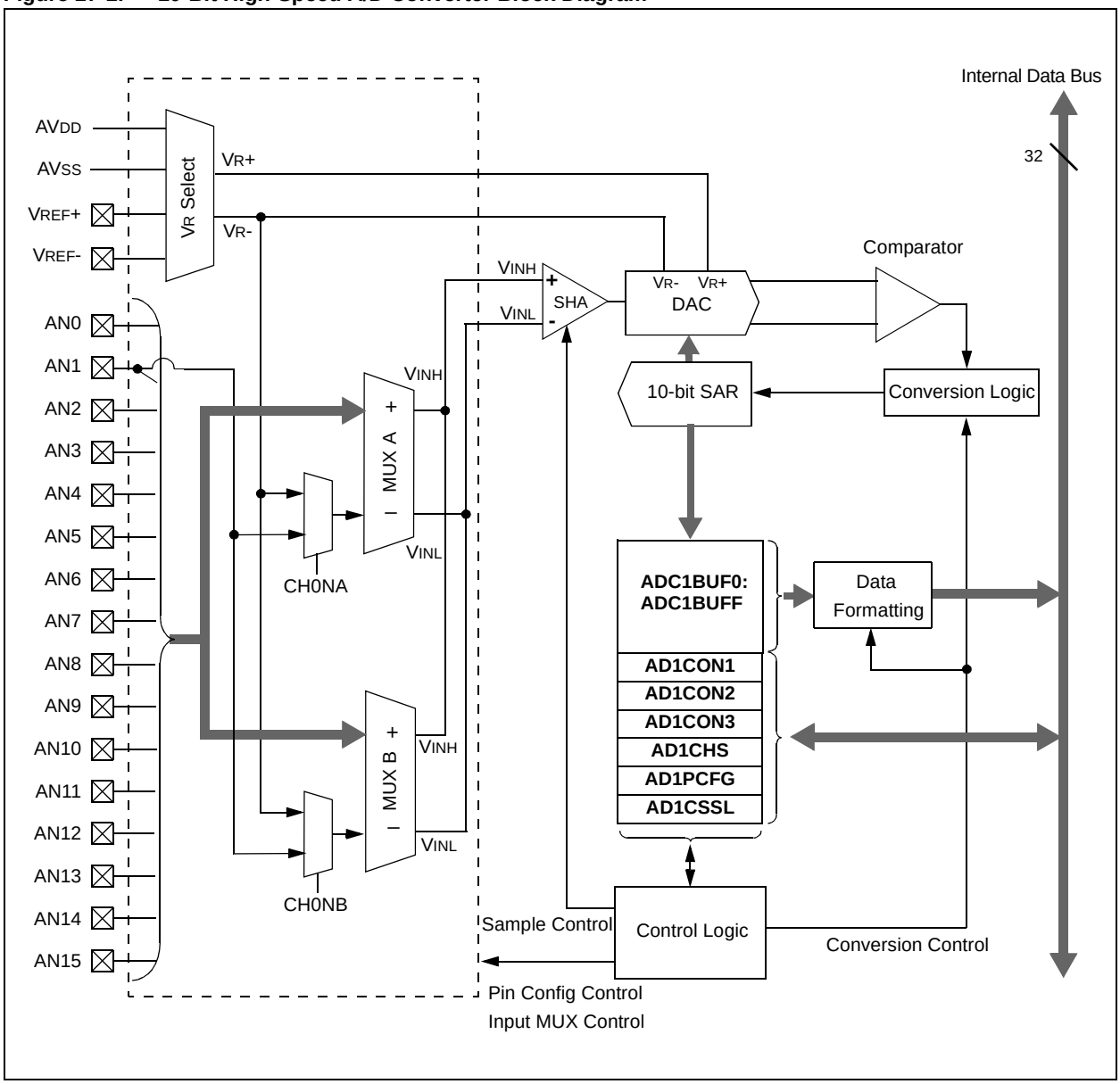
A block diagram of the 10-bit ADC is shown in Figure 17-1. The 10-bit ADC can have up to 16 analog input pins, designated AN0-AN15. In addition, there are two analog input pins for external voltage reference connections. These voltage reference inputs may be shared with other analog input pins and may be common to other analog module references. The actual number of analog input pins and external voltage reference input configuration will depend on the specific PIC32MX device. Refer to the device data sheet for further details.

The analog inputs are connected through two multiplexers (MUXs) to one SHA. The analog input MUXs can be switched between two sets of analog inputs between conversions. Unipolar differential conversions are possible on all channels, other than the pin used as the reference, using a reference input pin (see Figure 17-1).

The Analog Input Scan mode sequentially converts user-specified channels. A Control register specifies which analog input channels will be included in the scanning sequence.

The 10-bit ADC is connected to a 16-word result buffer. Each 10-bit result is converted to one of eight 32-bit output formats when it is read from the result buffer.

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



17.2 CONTROL REGISTERS

The ADC module includes the following Special Function Registers (SFRs):

The AD1CON1, AD1CON2 and AD1CON3 registers control the operation of the ADC module.

- AD1CON1: ADC Control Register 1
  AD1CON1CLR, AD1CON1SET, AD1CON1INV: Atomic Bit Manipulation, Write-only Registers for AD1CON1.
- AD1CON2: ADC Control Register 2
  AD1CON2CLR, AD1CON2SET, AD1CON2INV: Atomic Bit Manipulation, Write-only Registers for AD1CON2.
- AD1CON3: ADC Control Register 3
  AD1CON3CLR, AD1CON3SET, AD1CON3INV: Atomic Bit Manipulation, Write-only Registers for AD1CON3.

The AD1CHS register selects the input pins to be connected to the SHA.

- AD1CHS: ADC Input Channel Select Register
  AD1CHSCLR, AD1CHSSET, AD1CHSINV: Atomic Bit Manipulation, Write-only Registers for AD1CHS.

The AD1PCFG register configures the analog input pins as analog inputs or as digital I/O.

- AD1PCFG: ADC Port Configuration Register
  AD1PCFGCLR, AD1PCFGSET, AD1PCFGINV: Atomic Bit Manipulation, Write-only Registers for AD1PCFG.

The AD1CSSL register selects inputs to be sequentially scanned.

- AD1CSSL: ADC Input Scan Selection Register
  AD1CSSLCLR, AD1CSSLSET, AD1CSSLINV: Atomic Bit Manipulation, Write-only Registers for AD1CSSL.

The ADC module also has the following associated bits for interrupt control:

- Interrupt Request Flag Status bit (AD1IF) in IFS1: Interrupt Flag Status Register 1
- Interrupt Enable Control bit (AD1IE) in IEC1: Interrupt Enable Control Register 1
- Interrupt Priority Control bits (AD1IP<2:0>) and (AD1IS<1:0>) in IPC6: Interrupt Priority Control Register 6

17.2.1 Special Function Registers Associated with the 10-Bit ADC

The following table provides a summary of all ADC-related registers, including their addresses and formats. Corresponding registers appear after the summary, followed by a detailed description of each register. All unimplemented registers and/or bits within a register read as zeros.

Table 17-1: ADC SFR Summary

Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
AD1CON1	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ON	FRZ	SIDL	—	—	FORM2	FORM1	FORM0
	7:0	SSRC2	SSRC1	SSRC0	CLRASAM	—	ASAM	SAMP	DONE
AD1CON1CLR	31:0	Write clears selected bits in AD1CON1, read yields undefined value							
AD1CON1SET	31:0	Write sets selected bits in AD1CON1, read yields undefined value							
AD1CON1INV	31:0	Write inverts selected bits in AD1CON1, read yields undefined value							
AD1CON2	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	VCFG2	VCFG1	VCFG0	OFFCAL	—	CSCNA	—	—
	7:0	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
AD1CON2CLR	31:0	Write clears selected bits in AD1CON2, read yields undefined value							
AD1CON2SET	31:0	Write sets selected bits in AD1CON2, read yields undefined value							
AD1CON2INV	31:0	Write inverts selected bits in AD1CON2, read yields undefined value							
AD1CON3	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ADRC	—	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
	7:0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
AD1CON3CLR	31:0	Write clears selected bits in AD1CON3, read yields undefined value							
AD1CON3SET	31:0	Write sets selected bits in AD1CON3, read yields undefined value							
AD1CON3INV	31:0	Write inverts selected bits in AD1CON3, read yields undefined value							
AD1CHS	31:24	CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0
	23:16	CH0NA	—	—	—	CH0SA3	CH0SA2	CH0SA1	CH0SA0
	15:8	—	—	—	—	—	—	—	—
	7:0	—	—	—	—	—	—	—	—
AD1CHSCLR	31:0	Write clears selected bits in AD1CHS, read yields undefined value							
AD1CHSSET	31:0	Write sets selected bits in AD1CHS, read yields undefined value							
AD1CHS1INV	31:0	Write inverts selected bits in AD1CHS, read yields undefined value							
AD1PCFG	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
	7:0	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
AD1PCFGCLR	31:0	Write clears selected bits in AD1PCFG, read yields undefined value							
AD1PCFGSET	31:0	Write sets selected bits in AD1PCFG, read yields undefined value							
AD1PCFGINV	31:0	Write inverts selected bits in AD1PCFG, read yields undefined value							
AD1CSSL	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
	7:0	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
AD1CSSLCLR	31:0	Write clears selected bits in AD1CSSL, read yields undefined value							

Table 17-1: ADC SFR Summary (Continued)

Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
AD1CSSLSET	31:0	Write sets selected bits in AD1CSSL, read yields undefined value							
AD1CSSLINV	31:0	Write inverts selected bits in AD1CSSL, read yields undefined value							
ADC1BUF0	31:0	ADC Result Word 0 (ADC1BUF0<31:0>)							
ADC1BUF1	31:0	ADC Result Word 1 (ADC1BUF1<31:0>)							
ADC1BUF2	31:0	ADC Result Word 2 (ADC1BUF2<31:0>)							
ADC1BUF3	31:0	ADC Result Word 3 (ADC1BUF3<31:0>)							
ADC1BUF4	31:0	ADC Result Word 4 (ADC1BUF4<31:0>)							
ADC1BUF5	31:0	ADC Result Word 5 (ADC1BUF5<31:0>)							
ADC1BUF6	31:0	ADC Result Word 6 (ADC1BUF6<31:0>)							
ADC1BUF7	31:0	ADC Result Word 7 (ADC1BUF7<31:0>)							
ADC1BUF8	31:0	ADC Result Word 8 (ADC1BUF8<31:0>)							
ADC1BUF9	31:0	ADC Result Word 9 (ADC1BUF9<31:0>)							
ADC1BUFA	31:0	ADC Result Word A (ADC1BUFA<31:0>)							
ADC1BUFB	31:0	ADC Result Word B (ADC1BUFB<31:0>)							
ADC1BUFC	31:0	ADC Result Word C (ADC1BUFC<31:0>)							
ADC1BUFD	31:0	ADC Result Word D (ADC1BUFD<31:0>)							
ADC1BUFE	31:0	ADC Result Word E (ADC1BUFE<31:0>)							
ADC1BUFF	31:0	ADC Result Word F (ADC1BUFF<31:0>)							
IFS1	31:24	—	—	—	—	—	—	USBIF	FCEIF
	23:16	—	—	—	—	DMA3IF	DMA2IF	DMA1IF	DMA0IF
	15:8	RTCCIF	FSCMIF	I2C2MIF	I2C2SIF	I2C2BIF	U2TXIF	U2RXIF	U2EIF
	7:0	SPI2RXIF	SPI2TXIF	SPI2EIF	CMP2IF	CMP1IF	PMPIF	AD1IF	CNIF
IFS1CLR	31:0	Write clears the selected bits in IFS1, read yields undefined value							
IFS1SET	31:0	Write sets the selected bits in IFS1, read yields undefined value							
IFS1INV	31:0	Write inverts the selected bits in IFS1, read yields undefined value							
IEC1	31:24	—	—	—	—	—	—	USBIE	FCEIE
	23:16	—	—	—	—	DMA3IE	DMA2IE	DMA1IE	DMA0IE
	15:8	RTCCIE	FSCMIE	I2C2MIE	I2C2SIE	I2C2BIE	U2TXIE	U2RXIE	U2EIE
	7:0	SPI2RXIE	SPI2TXIE	SPI2EIE	CMP2IE	CMP1IE	PMPIE	AD1IE	CNIE
IPC6	31:24	—	—	—	AD1IP<2:0>			AD1IS<1:0>	
	23:16	—	—	—	CNIP<2:0>			CNIS<1:0>	
	15:8	—	—	—	I2C1IP<2:0>			I2C1IS<1:0>	
	7:0	—	—	—	U1IP<2:0>			U1IS<1:0>	
IPC6CLR	31:0	Write clears the selected bits in IPC6, read yields undefined value							
IPC6SET	31:0	Write sets the selected bits in IPC6, read yields undefined value							
IPC6INV	31:0	Write inverts the selected bits in IPC6, read yields undefined value							

Register 17-1: AD1CON1: ADC Control Register 1

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	r-x	r-x	R/W-0	R/W-0	R/W-0
ON	FRZ	SIDL	—	—	FORM<2:0>		
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	r-x	R/W-0	R/W-0	R/C-0
SSRC<2:0>			CLRASAM	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **ON:** ADC Operating Mode bit
1 = A/D converter module is operating
0 = A/D converter is off
Note: When using 1:1 PBCLK divisor, the user’s software should not read/write the peripheral’s SFRs in the SYSCLK cycle immediately following the instruction that clears the module’s ON bit.

bit 14 **FRZ:** Freeze in Debug Exception Mode bit
1 = Freeze operation when CPU enters Debug Exception mode
0 = Continue operation when CPU enters Debug Exception mode
Note: FRZ is writable in Debug Exception mode only. It reads ‘0’ in Normal mode.

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

bit 12-11 **Reserved:** Write ‘0’; ignore read

bit 10-8 **FORM<2:0>:** Data Output Format bits
011 = Signed Fractional 16-bit (DOUT = 0000 0000 0000 0000 sddd dddd dd00 0000)
010 = Fractional 16-bit (DOUT = 0000 0000 0000 0000 dddd dddd dd00 0000)
001 = Signed Integer 16-bit (DOUT = 0000 0000 0000 0000 ssss sssd dddd dddd)
000 = Integer 16-bit (DOUT = 0000 0000 0000 0000 0000 00dd dddd dddd)
111 = Signed Fractional 32-bit (DOUT = sddd dddd dd00 0000 0000 0000 0000)
110 = Fractional 32-bit (DOUT = dddd dddd dd00 0000 0000 0000 0000 0000)
101 = Signed Integer 32-bit (DOUT = ssss ssss ssss ssss ssss sssd dddd dddd)
100 = Integer 32-bit (DOUT = 0000 0000 0000 0000 0000 00dd dddd dddd)

Register 17-1: AD1CON1: ADC Control Register 1 (Continued)

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 = Reserved
010 = Timer 3 period match ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing SAMP bit ends sampling and starts conversion

bit 4 **CLRASAM:** Stop Conversion Sequence bit (when the first A/D converter interrupt is generated)

1 = Stop conversions when the first ADC interrupt is generated. Hardware clears the ASAM bit when the ADC interrupt is generated.
0 = Normal operation, buffer contents will be overwritten by the next conversion sequence

bit 3 **Reserved:** Write '0'; ignore read

bit 2 **ASAM:** ADC Sample Auto-Start bit

1 = Sampling begins immediately after last conversion completes; SAMP bit is automatically set.
0 = Sampling begins when SAMP bit is set

bit 1 **SAMP:** ADC Sample Enable bit

1 = The ADC SHA is sampling
0 = The ADC sample/hold amplifier is holding
When ASAM = 0, writing '1' to this bit starts 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 or has not started
Clearing this bit will not affect any operation in progress.

Note: The DONE bit isn't persistent in automatic modes. It is cleared by hardware at the beginning of the next sample.

Register 17-2: AD1CON1CLR: ADC Port Configuration Register

Write clears selected bits in AD1CON1, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1CON1**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1CON1 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON1CLR = 0x00008002` will clear bits 15 and 1 in AD1CON1 register.

Register 17-3: AD1CON1SET:ADC Port Configuration Register

Write sets selected bits in AD1CON1, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1CON1**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1CON1 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON1SET = 0x00008002` will set bits 15 and 1in AD1CON1 register.

Register 17-4: AD1CON1INV:ADC Port Configuration Register

Write inverts selected bits in AD1CON1, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1CON1**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1CON1 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON1INV = 0x00008002` will invert bits 15 and 1 in AD1CON1 register.

Register 17-5: AD1CON2: ADC Control Register 2

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	r-x	R/W-0	r-x	r-x
VCFG<2:0>			OFFCAL	—	CSCNA	—	—
bit 15							bit 8

R-0	r-x	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

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

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

	ADC VR+	**ADC VR-**
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **OFFCAL:** Input Offset Calibration Mode Select bit

1 = Enable Offset Calibration mode
VINH and VINL of the SHA are connected to **VR-**
0 = Disable Offset Calibration mode
The inputs to the SHA are controlled by AD1CHS or AD1CSSL

bit 11 **Reserved:** Write ‘0’; ignore read

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

bit 9-8 **Reserved:** Write ‘0’; ignore read’

bit 7 **BUFS:** Buffer Fill Status bit

Only valid when BUFM = 1 (ADRES split into 2 x 8-word buffers).
1 = ADC is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7
0 = ADC is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF

bit 6 **Reserved:** Write ‘0’; ignore read

Register 17-5: AD1CON2: ADC Control Register 2 (Continued)

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:** ADC Result Buffer Mode Select bit

1 = Buffer configured as two 8-word buffers, ADC1BUF(7...0), ADC1BUF(15...8)
0 = Buffer configured as one 16-word buffer ADC1BUF(15...0.)

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

1 = Uses MUX A input multiplexer settings for first sample, then alternates between MUX B and MUX A input multiplexer settings for all subsequent samples
0 = Always use MUX A input multiplexer settings

Register 17-6: AD1CON2CLR: ADC Port Configuration Register

Write clears selected bits in AD1CON2, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1CON2**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1CON2 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON2CLR = 0x00008001` will clear bits 15 and 0 in AD1CON2 register.

Register 17-7: AD1CON2SET:ADC Port Configuration Register

Write sets selected bits in AD1CON2, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1CON2**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1CON2 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON2SET = 0x00008001` will set bits 15 and 0 in AD1CON2 register.

Register 17-8: AD1CON2INV:ADC Port Configuration Register

Write inverts selected bits in AD1CON2, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1CON2**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1CON2 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON2INV = 0x00008001` will invert bits 15 and 0 in AD1CON2 register.

Register 17-9: AD1CON3: ADC Control Register 3

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC<4:0>				
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W	R/W-0
ADCS<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **ADRC:** ADC Conversion Clock Source bit
1 = ADC internal RC clock
0 = Clock derived from Peripheral Bus Clock (PBclock)

bit 14-13 **Reserved:** Write ‘0’; ignore read

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits
11111 = 31 TAD
.....
00001 = 1 TAD
00000 = 0 TAD (Not allowed)

bit 7-0 **ADCS<7:0>:** ADC Conversion Clock Select bits
11111111 = TPB • 2 • (ADCS<7:0> + 1) = 512 • TPB = TAD
.....
00000001 = TPB • 2 • (ADCS<7:0> + 1) = 4 • TPB = TAD
00000000 = TPB • 2 • (ADCS<7:0> + 1) = 2 • TPB = TAD

Register 17-10: AD1CON3CLR: ADC Port Configuration Register

Write clears selected bits in AD1CON3, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1CON3**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1CON3 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON3CLR = 0x00008001` will clear bits 15 and 0 in AD1CON3 register.

Register 17-11: AD1CON3SET:ADC Port Configuration Register

Write sets selected bits in AD1CON3, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1CON3**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1CON3 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON3SET = 0x00008001` will set bits 15 and 0 in AD1CON3 register.

Register 17-12: AD1CON3INV:ADC Port Configuration Register

Write inverts selected bits in AD1CON3, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1CON3**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1CON3 register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CON3INV = 0x00008001` will invert bits 15 and 0 in AD1CON3 register.

Register 17-13: AD1CHS:ADC Input Select Register

R/W-0	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	—	CH0SB<3:0>			
bit 31							bit 24

R/W-0	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	—	CH0SA<3:0>			
bit 23							bit 16

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 15							bit 8

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31 **CH0NB:** Negative Input Select bit for MUX B
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VR-

bit 30-28 **Reserved:** Write ‘0’; ignore read

bit 27-24 **CH0SB<3:0>:** Positive Input Select bits for MUX B
1111 = Channel 0 positive input is AN15
1110 = Channel 0 positive input is AN14
1101 = Channel 0 positive input is AN13
...
...
...
0001 = Channel 0 positive input is AN1
0000 = Channel 0 positive input is AN0

bit 23 **CH0NA:** Negative Input Select bit for MUX A Multiplexer Setting(2)
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VR-

bit 22-20 **Reserved:** Write ‘0’; ignore read

bit 19-16 **CH0SA<3:0>:** Positive Input Select bits for MUX A Multiplexer Setting
1111 = Channel 0 positive input is AN15
1110 = Channel 0 positive input is AN14
1101 = Channel 0 positive input is AN13
...
...
...
0001 = Channel 0 positive input is AN1
0000 = Channel 0 positive input is AN0

bit 15-0 **Reserved:** Write ‘0’; ignore read

Register 17-14: AD1CHSCLR: ADC Port Configuration Register

Write clears selected bits in AD1CHS, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1CHS**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1CHS register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CHSCLR = 0x80010000` will clear bits 15 and 0 in AD1CHS register.

Register 17-15: AD1CHSSET:ADC Port Configuration Register

Write sets selected bits in AD1CHS, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1CHS**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1CHS register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CHSSET = 0x80010000` will set bits 15 and 0 in AD1CHS register.

Register 17-16: AD1CHSINV:ADC Port Configuration Register

Write inverts selected bits in AD1CHS, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1CHS**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1CHS register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CHSINV = 0x80010000` will invert bits 15 and 0 in AD1CHS register.

Register 17-17: AD1PCFG:ADC Port Configuration Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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

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

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15-0 **PCFG<15:0>:** Analog Input Pin Configuration Control bits

1 = Analog input pin in Digital mode, port read input enabled, ADC input multiplexer input for this analog input connected to AVss

0 = Analog input pin in Analog mode, digital port read will return as a ‘1’ without regard to the voltage on the pin, ADC samples pin voltage

Note: The AD1PCFG register functionality will vary depending on the number of ADC inputs available on the selected device. Please refer to the specific device data sheet for additional details on this register.

Register 17-18: AD1PCFGCLR: ADC Port Configuration Register

Write clears selected bits in AD1PCFG, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1PCFG**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1PCFG register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1PCFGCLR = 0x00008001` will clear bits 15 and 0 in AD1PCFG register.

Register 17-19: AD1PCFGSET:ADC Port Configuration Register

Write sets selected bits in AD1PCFG, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1PCFG**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1PCFG register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1PCFGSET = 0x00008001` will set bits 15 and 0 in AD1PCFG register.

Register 17-20: AD1PCFGINV:ADC Port Configuration Register

Write inverts selected bits in AD1PCFG, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1PCFG**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1PCFG register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1PCFGINV = 0x00008001` will invert bits 15 and 0 in AD1PCFG register.

Register 17-21: AD1CSSL: ADC Input Scan Select Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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

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

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15-0 **CSSL<15:0>:** ADC Input Pin Scan Selection bits

1 = Select ANx for input scan

0 = Skip ANx for input scan

Note: The AD1CSSL register functionality will vary depending on the number of ADC inputs available on the selected device. Please refer to the specific device data sheet for additional details on this register.

Register 17-22: AD1CSSLCLR: ADC Port Configuration Register

Write clears selected bits in AD1CSSL, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in AD1CSSL**

A write of '1' in one or more bit positions clears the corresponding bit(s) in AD1CSSL register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CSSLCLR = 0x00008001` will clear bits 15 and 0 in AD1CSSL register.

Register 17-23: AD1CSSLSET:ADC Port Configuration Register

Write sets selected bits in AD1CSSL, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in AD1CSSL**

A write of '1' in one or more bit positions sets the corresponding bit(s) in AD1CSSL register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CSSLSET = 0x00008001` will set bits 15 and 0 in AD1CSSL register.

Register 17-24: AD1CSSLINV:ADC Port Configuration Register

Write inverts selected bits in AD1CSSL, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in AD1CSSL**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in AD1CSSL register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `AD1CSSLINV = 0x00008001` will invert bits 15 and 0 in AD1CSSL register.

Register 17-25: IFS1: Interrupt Flag Status Register 1(1)

r-x	r-x	r-x	r-x	r-x	r-x	R/W-0	R/W-0
—	—	—	—	—	—	USBIF	FCEIF
bit 31							bit 24

r-x	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3IF	DMA2IF	DMA1IF	DMA0IF
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RTCCIF	FSCMIF	I2C2MIF	I2C2SIF	I2C2BIF	U2TXIF	U2RXIF	U2EIF
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
SPI2RXIF	SPI2TXIF	SPI2EIF	CMP2IF	CMP1IF	PMPIF	AD1IF	CNIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 1 **AD1IF:** Analog-to-Digital Converter 1 Interrupt Request Flag bit

1 = Interrupt request has occurred
0 = No interrupt request has a occurred

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the ADC.

Register 17-26: IEC1: Interrupt Enable Control Register 1(1)

r-x	r-x	r-x	r-x	r-x	r-x	R/W-0	R/W-0
—	—	—	—	—	—	USBIE	FCEIE
bit 31							bit 24

r-x	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3IE	DMA2IE	DMA1IE	DMA0IE
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RTCCIE	FSCMIE	I2C2MIE	I2C2SIE	I2C2BIE	U2TXIE	U2RXIE	U2EIE
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
SPI2RXIE	SPI2TXIE	SPI2EIE	CMP2IE	CMP1IE	PMPIE	AD1IE	I2C1MIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 1 **AD1IE:** Analog-to-Digital Converter 1 Interrupt Enable bit

1 = Interrupt is enabled.
0 = Interrupt is disabled.

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the ADC.

Register 17-27: IPC6:Interrupt Priority Control Register 6(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	AD1IP<2:0>			AD1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CNIP<2:0>			CNIS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	I2C1IP<2:0>			I2C1IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W	R/W-0
—	—	—	U1IP<2:0>			U1IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 28 - 26 **AD1IP<2:0>:** Analog-to-Digital Converter 1 Interrupt Priority bits
- 111 = Interrupt Priority is 7
- 110 = Interrupt Priority is 6
- 101 = Interrupt Priority is 5
- 100 = Interrupt Priority is 4
- 011 = Interrupt Priority is 3
- 010 = Interrupt Priority is 2
- 001 = Interrupt Priority is 1
- 000 = Interrupt is disabled

bit 25-24 **AD1IS<1:0>:** Analog-to-Digital 1 Subpriority bits
- 11 = Interrupt Subpriority is 3
- 10 = Interrupt Subpriority is 2
- 01 = Interrupt Subpriority is 1
- 00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the ADC.

17.3 ADC OPERATION, TERMINOLOGY AND CONVERSION SEQUENCE

This section will describe the operation the A/D converter, the steps required to configure the converter, describe special feature of the module, and provide examples of ADC configuration with timing diagrams and charts showing the expected output of the converter.

17.3.1 Overview of Operation

Analog sampling consists of two steps: acquisition and conversion (see Figure 17-2). During acquisition the analog input pin is connected to the Sample and Hold Amplifier (SHA). After the pin has been sampled for a sufficient period, the sample voltage is equivalent to the input, the pin is disconnected from the SHA to provide a stable input voltage for the conversion process. The conversion process then converts the analog sample voltage to a binary representation.

An overview of the ADC is presented in Figure 17-1. The 10-bit A/D converter has a single SHA. The SHA is connected to the analog input pins via the analog input MUXs, MUX A and MUX B. The analog input MUXs are controlled by the AD1CHS register. There are two sets of MUX control bits in the AD1CHS register. These two sets of control bits allow the two different analog input to be independently controlled. The A/D converter can optionally switch between MUX A and MUX B configurations between conversions. The A/D converter can also optionally scan through a series of analog inputs using a single MUX.

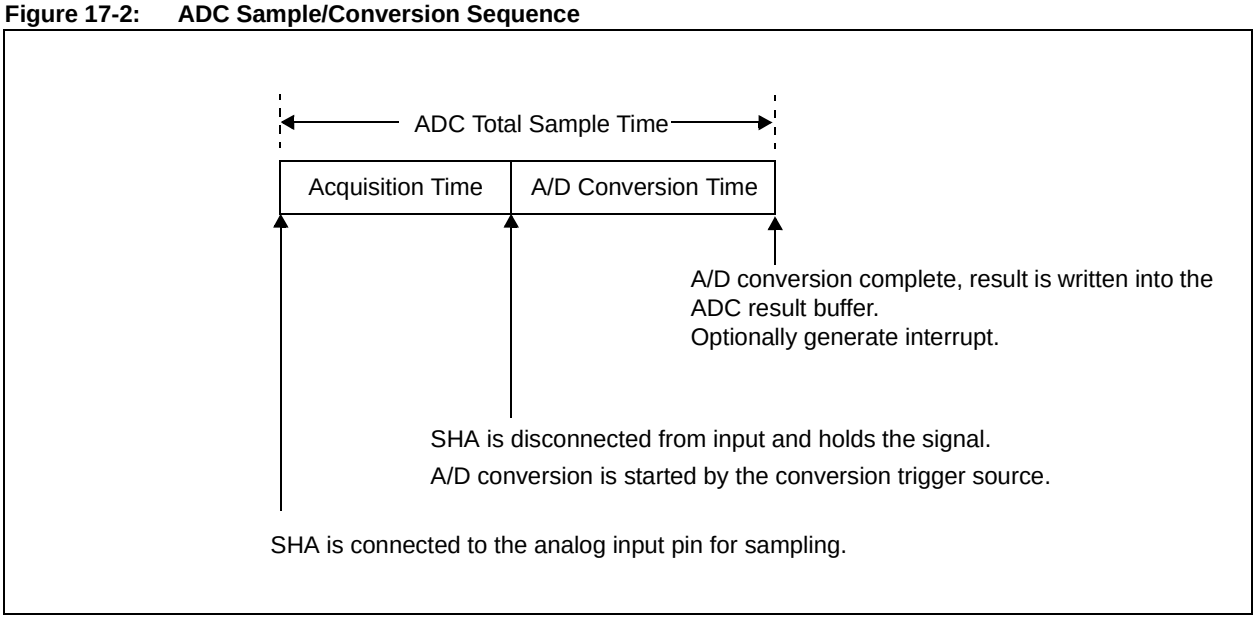
Acquisition time can be controlled manually or automatically. The acquisition time may be started manually by setting the SAMP bit (AD1CON1<1>), and ended manually by clearing the SAMP in the user software. The acquisition time may be started automatically by the A/D converter hardware and ended automatically by a conversion trigger source. The acquisition time is set by the SAMC bits (AD1CON3<12:8>). The SHA has a minimum acquisition period. Refer to the device data sheet for acquisition time specifications

Conversion time is the time required for the A/D converter to convert the voltage held by the SHA. The A/D converter requires one ADC clock cycle (TAD) to convert each bit of the result, plus two additional clock cycles. Therefore a total of 12 TAD cycles are required to perform the complete conversion. When the conversion time is complete, the result is written into one of the 16 ADC result registers (ADC1BUF0...ADC1BUFF).

The sum of the acquisition time and the A/D conversion time provides the total sample time (refer to Figure 17-2). There are multiple input clock options for the A/D converter that are used to create the TAD clock. The user must select an input clock option that does not violate the minimum TAD specification.

The sampling process can be performed once, periodically, or based on a trigger as defined by the module configuration.

Figure 17-2: ADC Sample/Conversion Sequence



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 SHA 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 (AD1CON1<2>).

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 control bits SSRC<2:0> (AD1CON1<7:5>). 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 ADC 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 sequence or multiple sample sequences as determined by the value of the SMPI<3:0> (AD1CON2<5:2>). The number of sample sequences between interrupts can vary between 1 and 16. The user should note that the A/D conversion buffer holds the results of a single conversion sequence. The next sequence starts filling the buffer from the top even if the number of samples in the previous sequence was less than 16. The total number of conversion results between interrupts is the SMPI value. The total number of conversions between interrupts cannot exceed the physical buffer length.

17.4 ADC MODULE CONFIGURATION

Operation of the ADC module is directed through bit settings in the appropriate registers. The following instructions summarize the actions and the settings. Options and details for each configuration step are provided in subsequent sections.

1. To configure the ADC module, perform the following steps:

   A-1. Configure analog port pins in AD1PCFG<15:0>, as described in **Section 17.4.1 "Configuring Analog Port Pins"**.

   B-1. Select the analog inputs to the ADC MUXs in AD1CHS<32:0>, as described in **Section 17.4.2 "Selecting the Analog Inputs to the ADC MUXs"**.

   C-1. Select the format of the ADC result using FORM<2:0> (AD1CON1<10:8>), as described in **Section 17.4.3 "Selecting the Format of the ADC Result"**.
   C-2. Select the sample clock source using SSRC<2:0> (AD1CON1<7:5>), as described in **Section 17.4.4 "Selecting the Sample Clock Source"**.

   D-1. Select the voltage reference source using VCFG<2:0> (AD1CON2<15:13>), as described in **Section 17.4.7 "Selecting the Voltage Reference Source"**.
   D-2. Select the Scan mode using CSCNA (AD1CON2<10>), as described in **Section 17.4.8 "Selecting the Scan Mode"**.
   D-3. Set the number of conversions per interrupt SMP<3:0> (AD1CON2<5:2>), if interrupts are to be used, as described in **Section 17.4.9 "Setting the Number of Conversions per Interrupt"**.
   D-4. Set Buffer Fill mode using BUFM (AD1CON2<1>), as described in **Section 17.4.10 "Buffer Fill Mode"**.
   D-5. Select the MUX to be connected to the ADC in ALTS AD1CON2<0>, as described in **Section 17.4.11 "Selecting the MUX to be Connected to the ADC (Alternating Sample Mode)"**.

   E-1. Select the ADC clock source using ADRC (AD1CON3<15>), as described in **Section 17.4.12 "Selecting the ADC Conversion Clock Source and Prescaler"**.
   E-2. Select the sample time using SAMC<4:0> (AD1CON3<12:8>), if auto-convert is to be used, as described in **Section 17.4.13 "Acquisition Time Considerations"**.
   E-3. Select the ADC clock prescaler using ADCS<7:0> (AD1CON3<7:0>), as described in **Section 17.4.12 "Selecting the ADC Conversion Clock Source and Prescaler"**.

   F. Turn on ADC module using AD1CON1<15>, as described in **Section 17.4.14 "Turning the ADC On"**.

Note:	Steps A through E, above, can be performed in any order, but Step F must be the final step in every case.

2. To configure ADC interrupt (if required).

   A-1. Clear AD1IF bit (IFS1<1>), as described in **Section 17.7 "Interrupts"**.
   A-2. Select ADC interrupt priority AD1IP<2:0> (IPC<28:26>) and sub priority AD1IS<1:0> (IPC<24:24>), as described in **Section 17.7 "Interrupts"**, if interrupts are to be used.

3. Start the conversion sequence by initiating sampling, as described in **Section 17.4.15 "Initiating Sampling"**.

17.4.1 Configuring Analog Port Pins

The AD1PCFG register and the TRISB register control the operation of the ADC port pins.

AD1PCFG specifies the configuration of device pins to be used as analog inputs. A pin is configured as analog input when the corresponding PCFGn bit (AD1PCFG<n>) = 0. When the bit = 1, the pin is set to digital control. When configured for analog input, the associated port I/O digital input buffer is disabled so it does not consume current. The AD1PCFG register is cleared at Reset, causing the ADC input pins to be configured for analog input by default at Reset.

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

Notes: When reading a PORT register that shares pins with the ADC, any pin configured as an analog input reads as a '0' when the PORT latch is read.

Analog levels on any pin that is defined as a digital input (including the AN15:AN0 pins), but is not configured as an analog input, may cause the input buffer to consume current that is out of the device's specification.

17.4.2 Selecting the Analog Inputs to the ADC MUXs

The AD1CHS register is used to select which analog input pin is connected to MUX A and MUX B. Each MUX has two inputs referred to as the positive and the negative input. The positive input to MUX A is controlled by CH0SA<4:0> and the negative input is controlled by CH0NA. The positive input for MUX B is controlled by CH0SB<4:0> and the negative input is controlled by CH0NB.

The positive input can be selected from any one of the available analog input pins. The negative input can be selected as the ADC negative reference or AN1. The use of AN1 as the negative input allows the ADC to be used in a Unipolar Differential mode. Refer to the device data sheet for AN1 input voltage restrictions when used as a negative reference.

Note: When using Scan mode CH0SA<4:0> may be overridden. Refer to **Section 17.4.8 "Selecting the Scan Mode"** for more information.

17.4.3 Selecting the Format of the ADC Result

The data in the ADC result register can be read as one of eight formats. The format is controlled by FORM<2:0> (AD1CON1<10:8>). The user can select from Integer, Signed Integer, Fractional, or Signed Fractional as a 16-bit or 32-bit result. Figure 17-3 shows how a result is formatted. Table 17-2 and Table 17-3 show examples of results for select results in each of the four formats with 32-bit and 16-bit results.

Note: There is no numeric difference between 32-bit and 16-bit modes. In 32-bit mode, the sign extension is applied to all 32-bits; whereas in 16-bit mode, sign extension is only applied to the lower 16-bits of the result.

Figure 17-3: ADC Output Data Formats, 32-Bit Mode

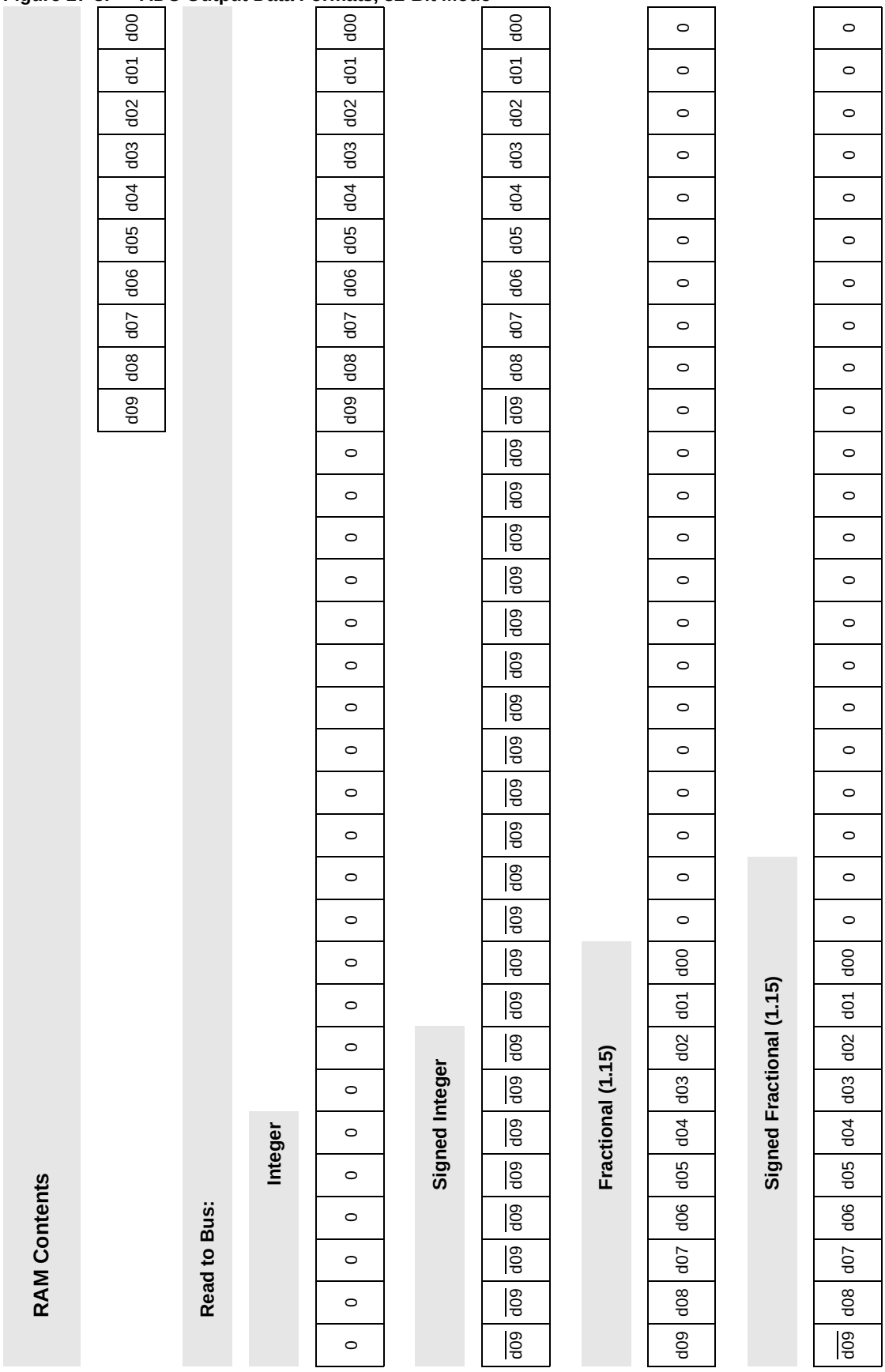


Figure 17-4: ADC Output Data Formats, 16-Bit Mode

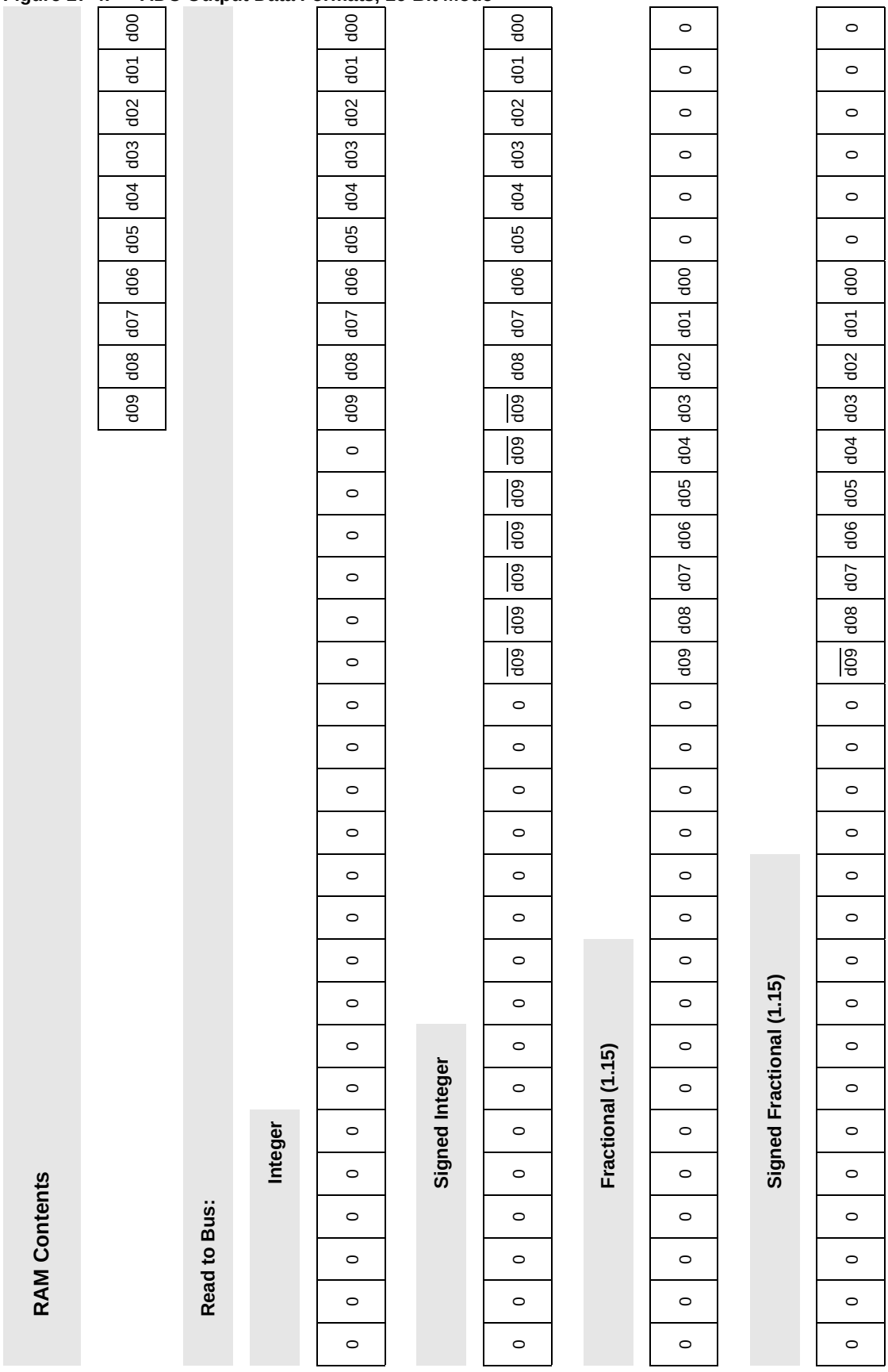


Table 17-2: Numerical Equivalents of Select Result Codes for FORM<2> (AD1CON1 <10>) = 1, 32-Bit Result

VIN/ VR	10-Bit Output Code	32-Bit Integer Format	32-Bit Signed Integer Format	32-Bit Fractional Format	32-Bit Signed Fractional Format
1023/1024	11 1111 1111	0000 0000 0000 0000 0000 0011 1111 1111 = 1023	0000 0000 0000 0000 0000 0001 1111 1111 = 511	1111 1111 1100 0000 0000 0000 0000 0000 = 0.999	0111 1111 1100 0000 0000 0000 0000 0000 = 0.499
1022/1024	11 1111 1110	0000 0000 0000 0000 0000 0011 1111 1110 = 1022	0000 0000 0000 0000 0000 0001 1111 1110 = 510	1111 1111 1000 0000 0000 0000 0000 0000 = 0.998	0111 1111 1000 0000 0000 0000 0000 0000 = 0.498
•••					
513/1024	10 0000 0001	0000 0000 0000 0000 0000 0010 0000 0001 = 513	0000 0000 0000 0000 0000 0000 0000 0001 = 1	1000 0000 0100 0000 0000 0000 0000 0000 = 0.501	0 000 0000 0100 0000 0000 0000 0000 0000 = 0.001
512/1024	10 0000 0000	0000 0000 0000 0000 0000 0010 0000 0000 = 512	0000 0000 0000 0000 0000 0000 0000 0000 = 0	1000 0000 0000 0000 0000 0000 0000 0000 = 0.500	0000 0000 0000 0000 0000 0000 0000 0000 = 0.000
511/1024	01 1111 1111	0000 0000 0000 0000 0000 0001 1111 1111 = 511	1111 1111 1111 1111 1111 1111 1111 1111 = -1	0111 1111 1100 0000 0000 0000 0000 0000 = .499	1111 1111 1100 0000 0000 0000 0000 0000 = -0.001
•••					
1/1024	00 0000 0001	0000 0000 0000 0000 0000 0000 0000 0001 = 1	1111 1111 1111 1111 1111 1110 0000 0001 = -511	0000 0000 0100 0000 0000 0000 0000 0000 = 0.001	1000 0000 0100 0000 0000 0000 0000 0000 = -0.499
0/1024	00 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 = 0	1111 1111 1111 1111 1111 1110 0000 0000 = -512	0000 0000 0000 0000 0000 0000 0000 0000 = 0.000	1000 0000 0000 0000 0000 0000 0000 0000 = -0.500

Table 17-3: Numerical Equivalents of Select Result Codes for FORM<2> (AD1CON1 <10>) = 0, 16-Bit Result

VIN/ VR	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 = 0.499
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 = 0.498
•••					
513/1024	10 0000 0001	0000 0010 0000 0001 = 513	0000 0000 0000 0001 = 1	1000 0000 0100 0000 = 0.501	0 000 0000 0100 0000 = 0.001
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 = 0.000
511/1024	01 1111 1111	0000 0001 1111 1111 = 511	1111 1111 1111 1111 = -1	0111 1111 1100 0000 = .499	1111 1111 1100 0000 = -0.001
•••					
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 = -0.499
0/1024	00 0000 0000	0000 0000 0000 0000 = 0	1111 1110 0000 0000 = -512	0000 0000 0000 0000 = 0.000	1000 0000 0000 0000 = -0.500

17.4.4 Selecting the Sample Clock Source

It is often desirable to synchronize the end of sampling and the start of conversion with some other time event. The ADC module may use one of four sources as a conversion trigger. The selection of the conversion trigger source is controlled by the SSRC<2:0> (AD1CON1<7:5>) bits.

17.4.4.1 Manual Conversion

To configure the ADC to end sampling and start a conversion when SAMP is cleared (= 0), SSRC is set to '000'.

17.4.4.2 Timer Compare Trigger

The ADC is configured for this Trigger mode by setting SSRC<2:0> = 010. When a period match occurs for the 32-bit timer, TMR3/TMR2, or the 16-bit Timer3 a special A/D converter trigger event signal is generated by Timer3. This feature does not exist for the TMR5/TMR4 timer pair or for 16-bit timers other than Timer3. Refer to **Section 14. "Timers"** for more details.

17.4.4.2.1 External INT0 Pin Trigger

To configure the ADC to begin a conversion on an active transition on the INT0 pin, SSRC<2:0> is set to '001'. The INT0 pin may be programmed for either a rising edge input or a falling edge input to trigger the conversion process.

17.4.4.2.2 Auto-Convert

The ADC can be configured to automatically perform conversions at the rate selected by the Auto Sample Time bits SAMC<4:0>. The ADC is configured for this Trigger mode by setting SSRC<2:0> = 111. In this mode, the ADC will perform continuous conversions on the selected channels.

17.4.5 Synchronizing ADC Operations to Internal or External Events

The modes where an external event trigger pulse ends sampling and starts conversion (SSRC2:SSRC0 = 001, 010 or 011) may be used in combination with auto-sampling (ASAM = 1) to cause the ADC to synchronize the sample conversion events to the trigger pulse source. For example, in Figure 17-13 where SSRC = 010 and ASAM = 1, the ADC will always end sampling and start conversions synchronously with the timer compare trigger event. The ADC will have a sample conversion rate that corresponds to the timer comparison event rate. See Example 17-5 for a code example.

17.4.6 Selecting Automatic or Manual Sampling

Sampling can be started manually or automatically when the previous conversion is complete.

17.4.6.1 Manual

Clearing the ASAM (AD1CON1<2>) bit disables the Auto-Sample mode. Acquisition will begin when the SAMP (AD1CON1<1>) bit is set by software. Acquisition will not resume until the SAMP bit is once again set. For an example, see Figure 17-8.

17.4.6.2 Automatic

Setting the ASAM (AD1CON1<2>) bit enables the Auto-Sample mode. In this mode, the sampling will start automatically after the pervious sample has been converted. For an example, see Figure 17-9.

17.4.7 Selecting the Voltage Reference Source

The user can select the voltage reference for the ADC module. The reference can be internal or external.

The VCFG<2:0> control bits (AD1CON2<15:13>) select the voltage reference for A/D conversions. The upper voltage reference (V_R+) and the lower voltage reference (V_R-) may be the internal AV_{DD} and AV_{SS} voltage rails, or the $V_{REF}+$ and $V_{REF}-$ input pins. The external ADC voltage reference may be used to reduce noise in the converter.

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 $V_{REF}+$ and $V_{REF}-$ 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 the electrical specifications.

Notes: External references $V_{REF}+$ and $V_{REF}-$ must be selected for high conversion. See the data sheet for further details.

The external $V_{REF}+$ and $V_{REF}-$ pins may be shared with other analog peripherals. Refer the device data sheet for further details.

17.4.8 Selecting the Scan Mode

The ADC module has the ability to scan through a selected vector of inputs. The CSCNA bit (AD1CON2<10>) enables the MUX A input to be scanned across a selected number of analog inputs.

17.4.8.1 Scan Mode Enable

Scan mode is enabled by setting CSCNA (AD1CON2<10>). When Scan mode is enabled the positive input of MUX A is controlled by the contents of the AD1CSSL register. Each bit in the AD1CSSL 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 AD1CSSL 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. When Scan mode is enabled the CH0SA<3:0> bits are ignored.

Notes:	If the number of scanned inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs will not be sampled. The AD1CSSL bits only specify the input of the positive input of the channel. The CH0NA bit selects the input of the negative input of the channel during scanning.

17.4.8.2 Scan Mode Disable

When CSCNA = 0, Scan mode is disabled and the positive input to MUX A is controlled by CH0SA<3:0>.

17.4.8.3 Using Scan and Alternate Modes Together

The Scan and Alternate modes may be combined to allow a vector of inputs to be scanned and a single input to be converted every other sample.

This mode is enabled by setting the CSCNA bit = 1, and setting the ALTS (AD1CON2<0>) bit = 1.

The CSCNA bit enables the scan for MUX A, and the CH0SB<3:0> (AD1CHS<27:24>) and CH0NB (AD1CHS<31>) are used to configure the inputs to MUX B. Scanning only applies to the MUX A input selection. The MUX B input selection, as specified by CH0SB<3:0>, will still select a single input.

The following sequence is an example of 3 scanned channels (MUX A) and a single fixed channel (MUX B):

1. The first input in the scan list is sampled.
2. The input selected by CH0SB<3:0> and CH0NB is sampled.
3. The second input in the scan list is sampled.
4. The input selected by CH0SB<3:0> and CH0NB is sampled.
5. The third input in the scan list is sampled.
6. The input selected by CH0SB<3:0> and CH0NB is sampled.

The process is repeated.

17.4.9 Setting the Number of Conversions per Interrupt

The SMPI<3:0> bits (AD1CON2<5:2>) select how many A/D conversions will take place before a CPU interrupt is generated. This also defines the number of locations that will be written in the result buffer stating with ADC1BUF0 (ADC1BUF0 or ADC1BUF8 for Dual Buffer mode). This can vary from 1 sample to 16 samples (1 to 8 samples for Dual Buffer mode). After the interrupt is generated, the sampling sequence restarts; with the result of the first sample being written to the first buffer location.

For example, if SMPI<3:0> = 0000, the conversion results will always be written to ADC1BUF0. In this example, no other buffer locations would be used.

For example, if SMPI<3:0> = 1110, 15 samples would be converted and stored in buffer locations ADC1BUF0 through ADC1BUFE. An interrupt would be generated after ADC1BUFE was written. The next sample would be written to ADC1BUF0. In this example ADC1BUFF would not be used.

The data in the result registers will be overwritten by the next sampling sequence. The data in the result buffer must be read before the completion of the first sample after the interrupt is generated. The Buffer Fill mode can be used to increase the time between interrupt generation and the overwriting of data. Refer to the Buffer Fill Mode section.

The user cannot program a combination of samples and SMPI bits that results in more than 16 conversions per interrupt when the BUFM bit (AD1CON2<1>) is '1', or more than 8 conversions per interrupt when the BUFM bit (AD1CON2<1>) is '0'. Attempting to create a conversion list with the number of samples greater than 16 will result in the sampling sequence being truncated to 16 samples.

17.4.10 Buffer Fill Mode

The Buffer Fill mode allows the output buffer to be used as a single 16-word buffer or two 8-word buffers.

When BUFM is '0', the complete 16-word buffer is used for all conversion sequences. Conversion results will be written sequentially in the buffer starting at ADC1BUF0 until the number of samples as defined by SMPI<3:0> (AD1CON2<5:2>) is reached. The next conversion result will be written to ADC1BUF0 and the process repeats. If the ADC interrupt is enabled an interrupt will be generated when the number of samples in the buffer equals SMPI<3:0>.

When the BUFM bit (AD1CON2<1>) is '1', the 16-word results buffer (ADRES) will be split into two 8-word groups. Conversion results will be written sequentially into the first buffer starting at ADC1BUF0, BUFS (AD1CON2<7>) will be cleared, until the number of samples as defined by SMPI<3:0> (AD1CON2<5:2>) is reached. The ADC interrupt flag will then be set.

After the ADC interrupt flag is set the following result will be written sequentially to the second buffer starting at ADC1BUF8 The next conversion result will be written to the second buffer starting at ADC1BUF8, BUFS (AD1CON2<7>) will be set, until the number of samples as defined by SMPI<3:0> (AD1CON2<5:2>) is reached. The ADC interrupt flag will then be set.

The process then restarts with BUFS = 0 and results being written to the first buffer.

The decision of which Buffer Fill mode to use will depend upon how much time is available to move the buffer contents after the A/D interrupt and the interrupt latency, as determined by the application. If the processor can 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 acquisition-and-conversion period before the first buffer location is overwritten.

If the processor cannot unload the buffer within the sample-and-conversion time the Dual Buffer mode BUFM bit = 1, should be used to prevent overwriting result data. For example, if SMPI<3:0> = 0111, then eight conversions will be written loaded into the first buffer, following which an interrupt will occur. The next eight conversions will be written to the second buffer. Therefore the processor will have the entire time between interrupts to read the eight conversions out of the buffer.

17.4.11 Selecting the MUX to be Connected to the ADC (Alternating Sample Mode)

The ADC has two input MUXs that connect to the SHA. These MUXs are used to select which analog input is to be sampled. Each of the MUXs have a positive and a negative input (see Figure 17-5 and Figure 17-6).

Note:	The number of analog inputs will vary among different devices. Verify the analog input availability with the appropriate device data sheet.

17.4.11.1 Single Input Selection

The user may select one of up to 16 analog inputs, as determined by the number of analog channels on the device, as the positive input of the SHA. The CH0SA<3:0> bits (AD1CHS<19:16>) select the positive analog input.

The user may select either V_{R-} or AN1 as the negative input. The CH0NA bit (AD1CHS<23>) selects the analog input for the negative input of channel 0. Using AN1 as the negative input allows unipolar differential measurements.

The ALTS bit (AD1CON2<0>) must be clear for this mode of operation.

17.4.11.2 Alternating Input Selections

The ALTS bit causes the module to alternate between the two input MUXs.

The inputs specified by CH0SA<3:0> and CH0NA are called the MUX A inputs. The inputs specified by CH0SB<3:0> and CH0NB are called the MUX B inputs, see Figure .

When ALTS is '1', the module will alternate between the MUX A inputs on one sample and the MUX B inputs on the subsequent sample. When ALTS is '0', only the inputs specified by CH0SA<3:0> and CH0NA are selected for sampling.

For example, if ALTS is '1' on the first sample/convert sequence, the inputs specified by CH0SA<3:0> and CH0NA are selected for sampling. On the next sample, the inputs specified by CH0SB<3:0> and CH0NB are selected for sampling. Then the pattern repeats.

17.4.12 Selecting the ADC Conversion Clock Source and Prescaler

The ADC module can use the internal RC oscillator or the PBCLK as the conversion clock source.

When the internal RC oscillator is used as the clock source, ADRC (AD1CON3<15>) = 1, the TAD is the period of the oscillator, no prescaler are used. When using the internal oscillator the ADC can continue to function in SLEEP and in IDLE.

Note:	The ADRC is intended for ADC operation in Sleep it is not calibrated. Applications requiring precise timing of ADC acquisitions should use a stable calibrated clock source for the ADC.

When the PBCLK is used as the conversion clock source, ADRC = 0, the TAD is the period of the PBCLK after the prescaler ADCS<7:0> (AD1CON3<7:0>) is applied.

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

The period of the ADC conversion clock is software selected using a 8-bit counter. There are 256 possible options for TAD, specified by the ADCS<7:0> bits (AD1CON3<7:0>).

Equation 17-1 gives the TAD value as a function of the ADCS control bits and the device instruction cycle clock period, TCY.

Equation 17-1: ADC Conversion Clock Period

$$T_{AD} = 2 \bullet (T_{PB}(ADCS + 1)$$

$$ADCS = (T_{AD}/(2 \bullet T_{PB})) - 1$$

For correct A/D conversions, the ADC conversion clock (TAD) must be selected to ensure a minimum TAD time of 83.33 nsec (see Section 17.11.1).

Equation 17-2: 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.

17.4.13 Acquisition Time Considerations

Different acquisition/conversion sequences provide different times for the sample-and-hold channel to acquire the analog signal. The user must ensure the acquisition time meets the sampling requirements, as outlined in **Section 17.11.3 "ADC Sampling Requirements"**.

When SSRC<2:0> (AD1CON1<7:5>) = 111, the conversion trigger is under ADC clock control. The SAMC<4:0> bits (AD1CON3<12:8>) select the number of TAD clock cycles between the start of acquisition and the start of conversion. This trigger option provides the fastest conversion rates on multiple channels. After the start of acquisition, the module will count a number of TAD clocks specified by the SAMC bits.

17.4.14 Turning the ADC On

When the ON bit (AD1CON1<15>) is '1', the module is in Active mode and is fully powered and functional.

When ON 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.

Note:	Writing to ADC control bits other than ON (AD1CON1<15>), SAMP (AD1CON1<1>), and DONE (AD1CON1<0>) is not recommended while the A/D converter is running.

17.4.15 Initiating Sampling

17.4.15.1 Manual Mode

In manual sampling, a acquisition is started by writing a '1' to the SAMP (AD1CON1<1>) bit. Software must manually manage the start and end of the acquisition period by setting SAMP and then clearing SAMP after the desired acquisition period has elapsed.

17.4.15.2 Auto-Sample Mode

In Auto-Sample mode, the sampling process is started by writing a '1' to the ASAM (AD1CON1<2>) bit. In Auto-Sample mode, the acquisition period is defined by ADCS<7:0> (AD1CON3<7:0>). Acquisition is automatically started after a conversion is completed. Auto-Sample mode can be used with any trigger source other than manual.

17.5 MISCELLANEOUS ADC FUNCTIONS

The following section describes bits not covered in the previous section.

17.5.1 Aborting Sampling

Clearing the SAMP (AD1CON1<1>) bit while in Manual Sample mode will terminate sampling, but may also start a conversion if SSRC (AD1CON1<7:5>) = 000.

Clearing the ASAM (AD1CON1<2>) bit while in Auto-Sample mode will not terminate an ongoing acquire/convert sequence, however, sampling will not automatically resume after the current sample is converted.

17.5.2 Aborting a Conversion

Clearing the ON (AD1CON1<15>) bit during a conversion will abort the current conversion. The ADC Result register will NOT be updated with the partially completed A/D conversion sample. That is, the corresponding result buffer location will continue to contain the value of the last completed conversion (or the last value written to the buffer).

17.5.3 Buffer Fill Status

When the conversion result buffer is split using the BUFM control bit, the BUFS Status bit (AD1CON2<7>) indicates which half of the buffer the A/D converter is currently filling. If BUFS = 0, then the A/D converter is filling ADC1BUF0-ADC1BUF7 and the user software should read conversion values from ADC1BUF8-ADC1BUFF. If BUFS = 1, the situation is reversed and the user software should read conversion values from ADC1BUF0-ADC1BUF7.

17.5.4 Offset Calibration

The ADC module provides a method of measuring the internal offset error. After this offset error is measured, it can be subtracted, in software, from the result of a A/D conversion. Use the following steps to perform an offset measurement:

1. Configure the A/D converter in the same manner as it will be used in the application.
2. Set the OFFCAL bit (AD1CON2<12>). This overrides the input selections and connects the sample-and-hold inputs to AVss.
3. If auto-sample is used set the CLRASAM bit (AD1CON1<4>) to stop conversions when the number of samples stated by SMPI is reached.
4. Enable the A/D converter and perform a conversion. The result that is written to the ADC result buffer is the internal offset error.
5. Clear the OFFCAL (AD2CON<12>) bit to return the A/D converter to normal operation.

Note:	Only positive ADC offsets can be measured with this method.

17.5.5 Terminate Conversion Sequence after an Interrupt

The CLRASAM bit provides a method to terminate auto-sample after the first sequence is completed. Setting the CLRASAM and starting an auto-sample sequence will cause the A/D converter to complete one auto-sample sequence (the number of samples as defined by SMPI<3:0> (AD1CON2<5:2>)). Hardware will the clear ASAM (AD1CON1<2>) and set the interrupt flag. This will stop the sampling process to allow inspection of the result buffer without results being overwritten by the next automatic conversion sequence. The CLRASAM must be cleared by software to disable this mode.

Note:	Disabling Interrupts or masking the ADC interrupt has no effect on the operation of the CLRASAM bit.

17.5.6 DONE Bit Operation

The DONE (AD1CON1<0>) bit is set when a conversion sequence is complete.

In Manual mode the DONE bit is persistent. It remains set until it is cleared by software. The DONE bit can be polled to determine when the conversion has completed.

In all automatic sample modes (ASAM = 1) the DONE bit is not persistent. It is set at the end of a conversion sequence and cleared by hardware when the next acquisition is started. Polling the DONE bit is not recommended when operating the ADC in automatic modes. The AD1IF (IFS1<1>) flag is latched after a conversion sequence is completed and can therefore be polled.

Figure 17-5: Simplified 10-Bit High-Speed A/D Converter Block Diagram for Alternate Sample Mode

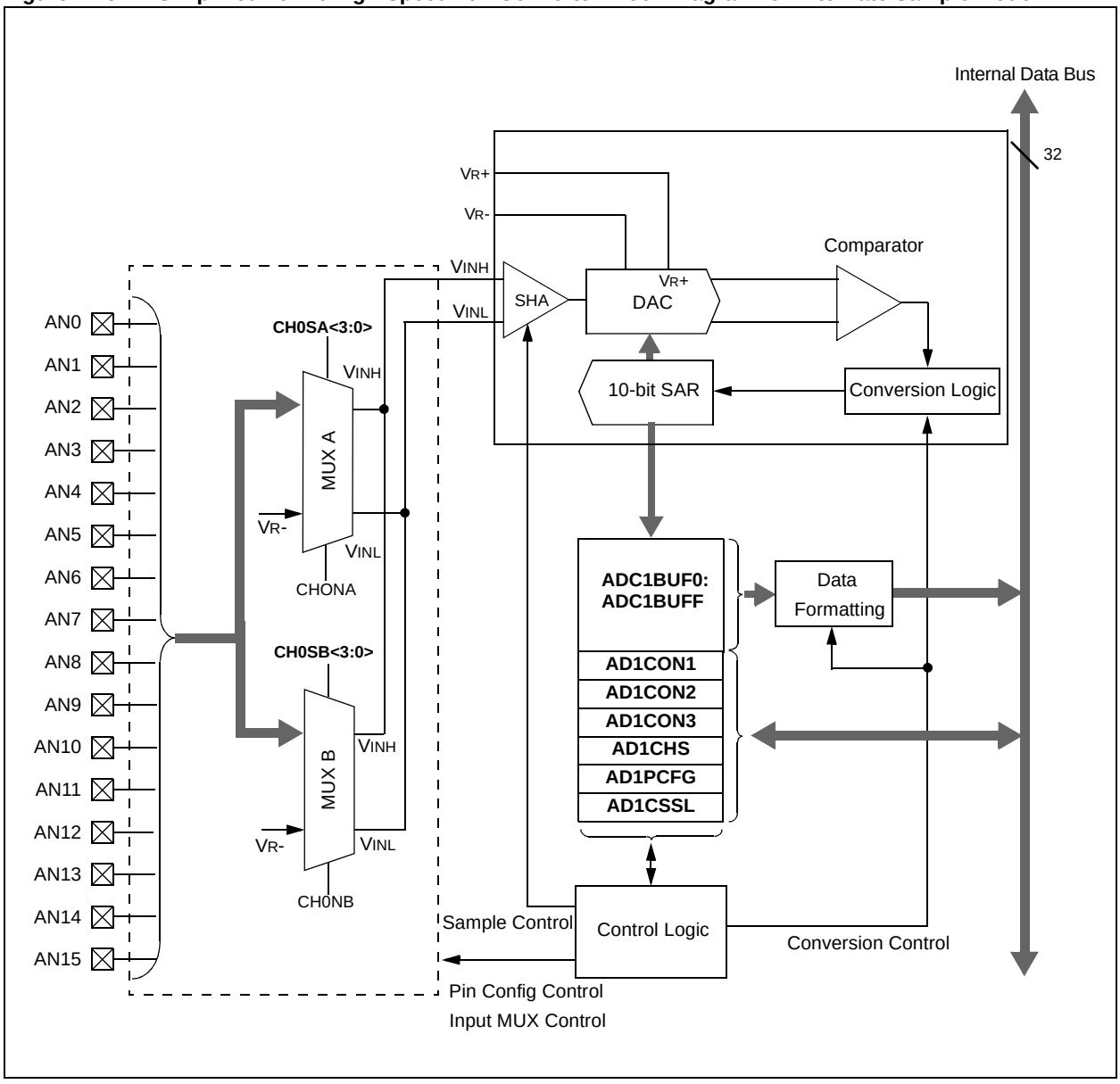


Figure 17-6: Simplified 10-Bit High-Speed A/D Converter Block Diagram for Scan Mode

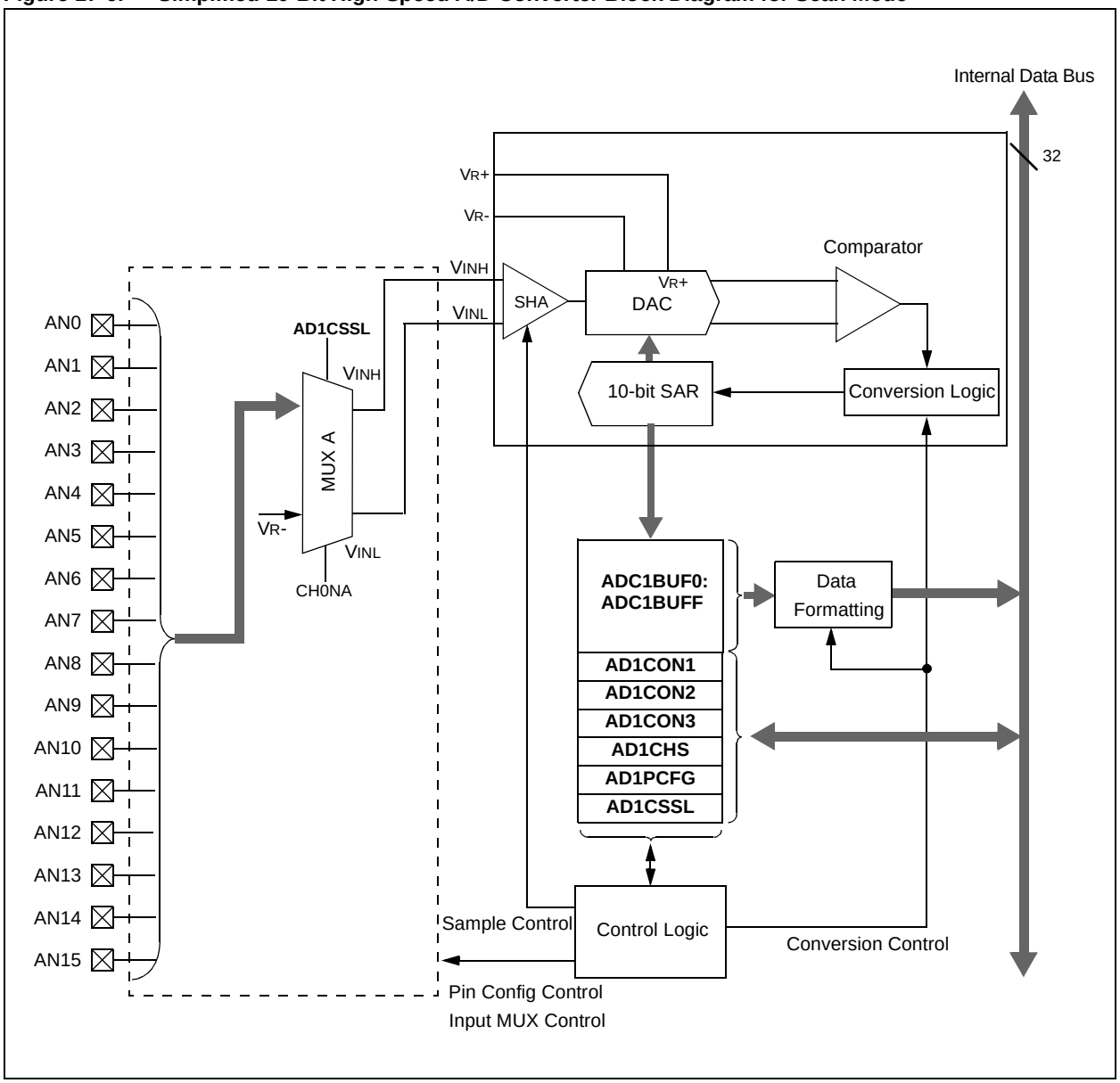
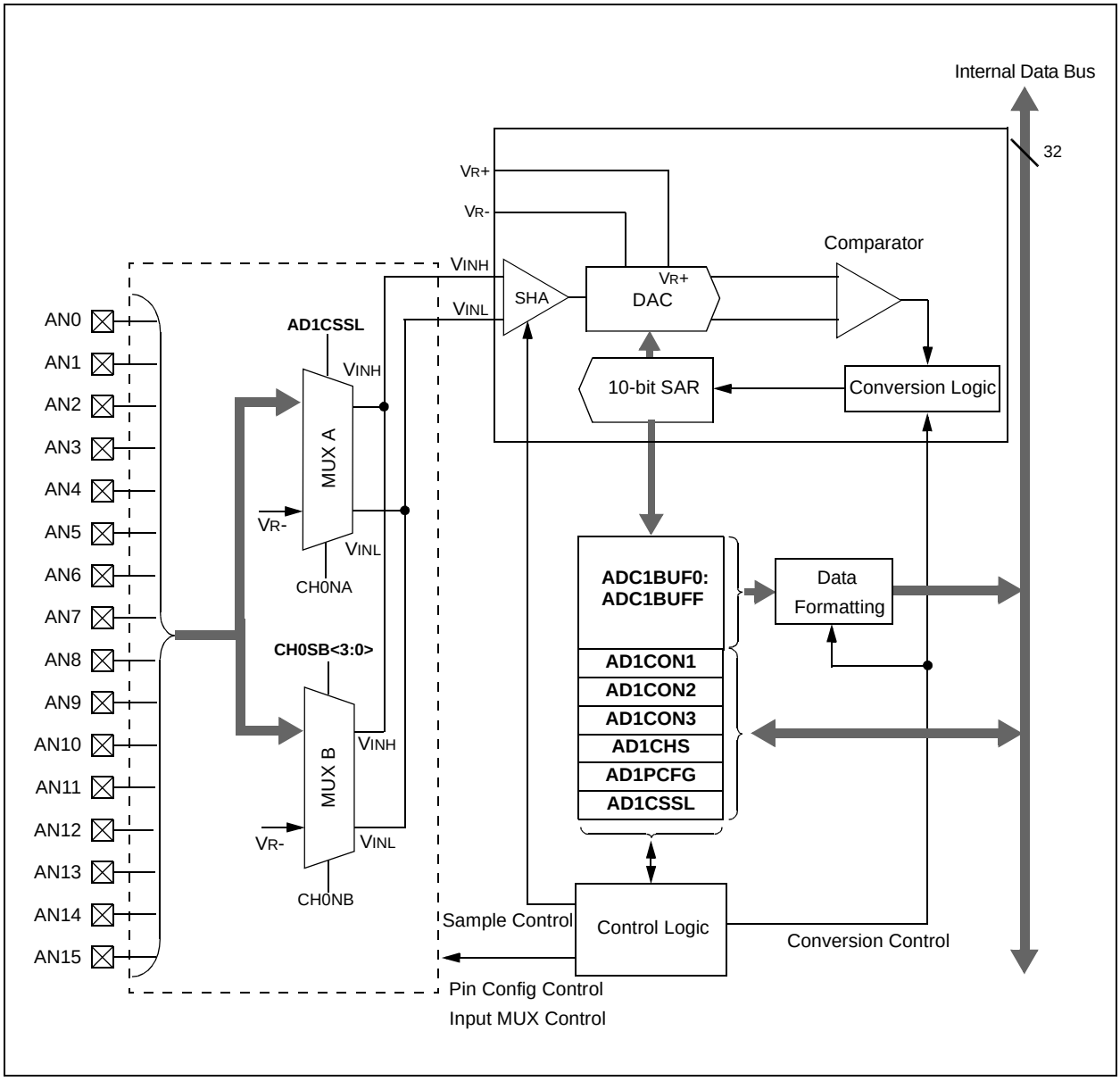


Figure 17-7: Simplified 10-Bit High-Speed A/D Converter Block Diagram for Alternate Sample and Scan Mode



17.5.7 Conversion Sequence Examples

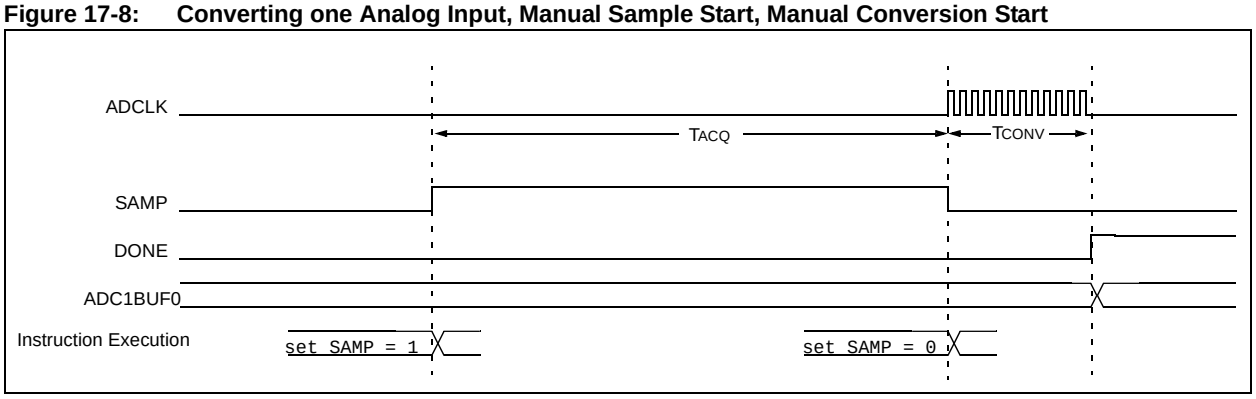
The following configuration examples show the ADC operation in different sampling and buffering configurations. In each example, setting the ASAM bit starts automatic sampling. A conversion trigger ends sampling and starts conversion.

17.5.8 Manual Conversion Control

When SSRC<2:0> = 000, the conversion trigger is under software control. Clearing the SAMP bit (AD1CON1<1>) starts the conversion sequence.

Figure 17-8 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 acquisition time of the input signal. See Example 17-1 for a code example.

Figure 17-8: Converting one Analog Input, Manual Sample Start, Manual Conversion Start



Example 17-1: Converting 1 Channel, Manual Sample Start, Manual Conversion Start Code

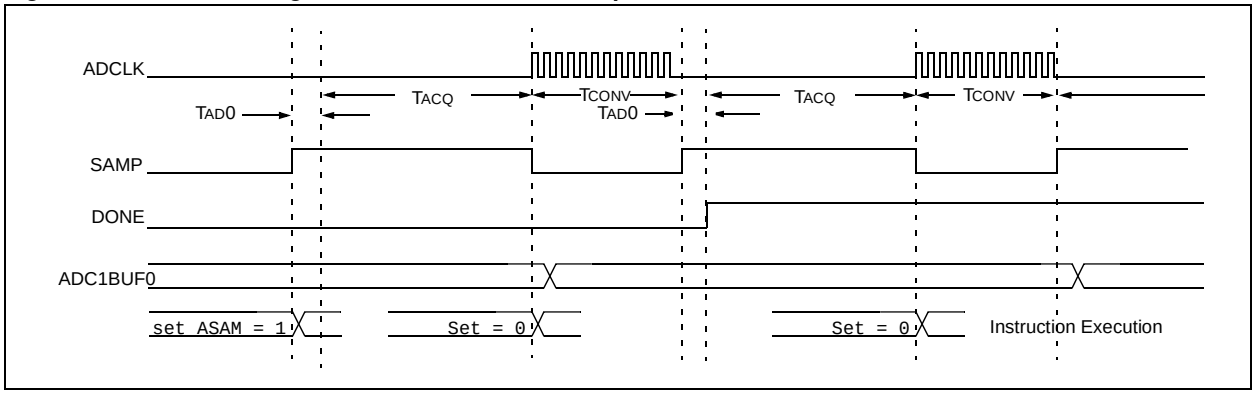
```
  AD1PCFG = 0xFFFB;             // PORTB = Digital; RB2 = analog
  AD1CON1 = 0x0000;             // SAMP bit = 0 ends sampling ...
                                // and starts converting
  AD1CHS = 0x00020000;          // Connect RB2/AN2 as CH0 input  ..
                                // in this example RB2/AN2 is the input
  AD1CSSL = 0;
  AD1CON3 = 0x0002;             // Manual Sample, Tad = internal 6 TPB
  AD1CON2 = 0;


  AD1CON1SET = 0x8000;          // turn ADC ON
  while (1)                     // repeat continuously
  {
     AD1CON1SET = 0x0002;       // start sampling ...
     DelayNmSec(100);           //  for 100 mS
     AD1CON1CLR = 0x0002;       // start Converting
     while (!(AD1CON1 & 0x0001));// conversion done?
     ADCValue = ADC1BUF0;       // yes then get ADC value
  }                             // repeat
```


17.5.9 Automatic Acquisition

Figure 17-9 is an example in which setting the ASAM (AD1CON1<2>) bit initiates automatic acquisition, and clearing the SAMP (AD1CON1<1>) bit terminates sampling and starts conversion. After the conversion completes, the module will automatically return to a acquisition state. The SAMP bit is automatically set at the start of the acquisition interval. The user software must time the clearing of the SAMP bit to ensure adequate acquisition time of the input signal, understanding that the time between clearing of the SAMP bit includes the conversion time as well as the acquisition time. See Example 17-2 for a code example.

Figure 17-9: Converting 1 Channel, Automatic Sample Start, Manual Conversion Start



Example 17-2: Converting 1 Channel, Automatic Sample Start, Manual Conversion Start Code

```
AD1PCFG = 0xFF7F;            // all PORTB = Digital but RB7 = analog
AD1CON1 = 0x0004;            // ASAM bit = 1 implies acquisition ..
                             // starts immediately after last
                             // conversion is done
AD1CHS = 0x00070000;         // Connect RB7/AN7 as CH0 input  ..
                             // in this example RB7/AN7 is the input
AD1CSSL = 0;
AD1CON3 = 0x0002;            // Sample time manual, Tad = internal 6 TPB
AD1CON2 = 0;


AD1CON1SET = 0x8000;         // turn ADC ON
while (1)                    // repeat continuously
{
  DelayNmSec(100);           // sample for 100 mS
  AD1CON1SET = 0x0002;       // start Converting
  while (!(AD1CON1 & 0x0001));// conversion done?
  ADCValue = ADC1BUF0;       // yes then get ADC value
}                            // repeat
```

17.5.10 Clocked Conversion Trigger

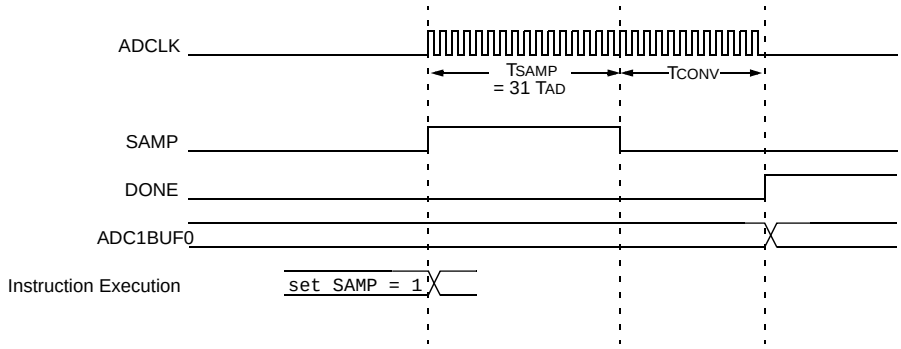
When SSRC<2:0> = 111, the conversion trigger is under ADC clock control. The SAMC bits (AD1CON3<4:0>) select the number of TAD clock cycles between the start of acquisition and the start of conversion. This trigger option provides the fastest conversion rates on multiple channels. After the start of acquisition, the module will count a number of TAD clocks specified by the SAMC bits.

Equation 17-3: Clocked Conversion Trigger Time

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

SAMC must always be programmed for at least one clock cycle. See Example 17-3 for a code example.

Figure 17-10: Converting 1 Channel, Manual Sample Start, TAD Based Conversion Start



Example 17-3: Converting 1 Channel, Manual Sample Start, TAD Based Conversion Start Code

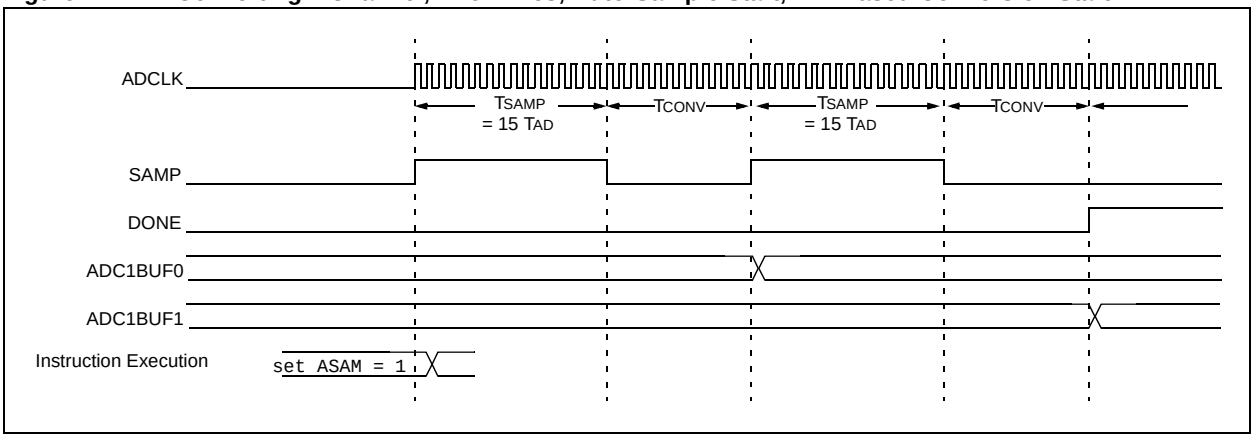
```
AD1PCFG = 0xEFFF;                   // all PORTB = Digital; RB12 = analog
AD1CON1 = 0x00E0;                   // SSRC bit = 111 implies internal
                                    // counter ends sampling and starts
                                    // converting.
AD1CHS = 0x000C0000;                // Connect RB12/AN12 as CH0 input  ..
                                    // in this example RB12/AN12 is the input
AD1CSSL = 0;
AD1CON3 = 0x1F02;                   // Sample time = 31Tad
AD1CON2 = 0;


AD1CON1SET = 0x8000;                // turn ADC ON
while (1)                           // repeat continuously
{
   AD1CON1CLR = 0x0002;             // start sampling then ...
                                    // after 31Tad go to conversion
   while (!(AD1CON1 & 0x0001));     // conversion done?
   ADCValue = ADC1BUF0;             // yes then get ADC value
}                                   // repeat
```


17.5.11 Free Running Sample Conversion Sequence

As shown in Figure 17-11, using the Auto-Convert Conversion Trigger mode (SSRC = 111) in combination with the Automatic Sampling Start mode (ASAM = 1), allows the ADC module to schedule acquisition/conversion sequences with no intervention by the user or other device resources. This “Clocked” mode allows continuous data collection after module initialization. See Example 17-4 for a code example.

Figure 17-11: Converting 1 Channel, Two Times, Auto-Sample Start, TAD Based Conversion Start



Example 17-4: Converting 1 Channel, Auto-Sample Start, TAD Based Conversion Start Code

```
AD1PCFG = 0xFFFB;                   // all PORTB = Digital; RB2 = analog
AD1CON1 = 0x00E0;                   // SSRC bit = 111 implies internal
                                    // counter ends sampling and starts
                                    // converting.
AD1CHS  = 0x00020000;               // Connect RB2/AN2 as CH0 input  ..
                                    // in this example RB2/AN2 is the input
AD1CSSL = 0;
AD1CON3 = 0x0F00;                   // Sample time = 15Tad
AD1CON2 = 0x0004;                   // Interrupt after every 2 samples



AD1CON1SET = 0x8000;                // turn ADC ON
while (1)                           // repeat continuously
{
   ADCValue = 0;                    // clear value
   ADC16Ptr = &ADC1BUF0;            // initialize ADC1BUF0 pointer
   IFS1CLR = 0x0002;                // clear ADC interrupt flag
   AD1CON1SET = 0x0004;             // auto start sampling
                                    // for 31Tad then go to conversion
   while (!IFS1 & 0x0002);          // conversion done?
   AD1CON1CLR = 0x0004;             // yes then stop sample/convert
   for (count = 0; count < 2; count++)// average the 2 ADC values
   {
      ADCValue = ADCValue + *(ADC16Ptr++);
      ADCValue = ADCValue >> 1;
   }                                // repeat
}
```

17.5.12 Acquisition Time Considerations Using Clocked Conversion Trigger and Automatic Sampling

Different acquisition/conversion sequences provide different available acquisition times for the sample-and-hold channel to acquire the analog signal. The user must ensure the acquisition time exceeds the acquisition requirements, as outlined in **Section 17.11.3 "ADC Sampling Requirements"**.

Assuming that the module is set for automatic sampling and using a clocked conversion trigger, the acquisition interval is determined by the SAMC (AD1CON3<12:8>) bits.

Equation 17-4: Available Sampling Time

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

Figure 17-12: Converting 1 Channel, Manual Sample Start, Conversion Trigger Based Conversion Start

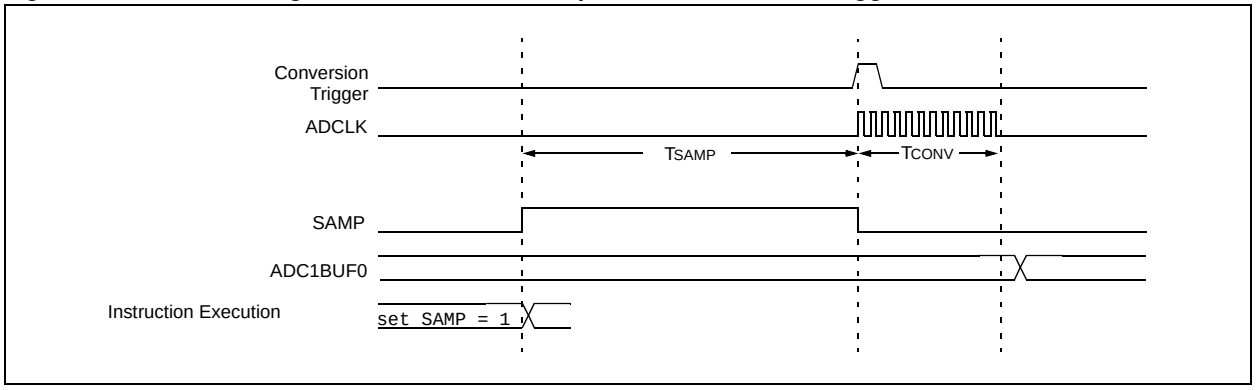
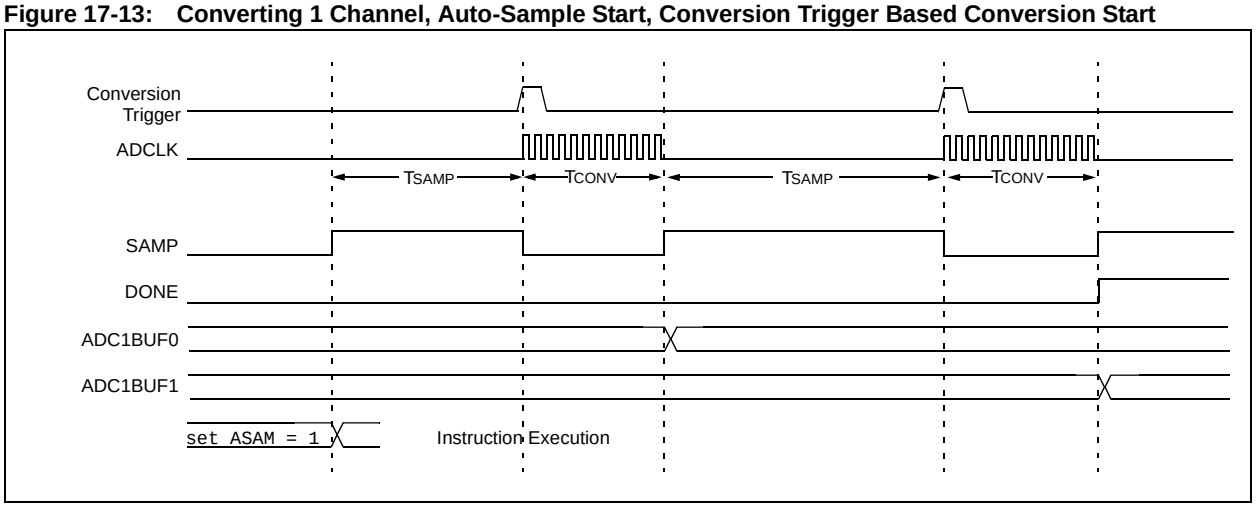


Figure 17-13: Converting 1 Channel, Auto-Sample Start, Conversion Trigger Based Conversion Start



Example 17-5: Converting 1 Channel, Auto-Sample Start, Conversion Trigger Based Conversion Start Code

```
AD1PCFG = 0xFFFB;                   // all PORTB = Digital; RB2 analog
AD1CON1 = 0x0040;                   // SSRC bit = 010 implies GP TMR3
                                    // compare ends sampling and starts
                                    // converting.
AD1CHS = 0x00020000;                // Connect RB2/AN2 as CH0 input  ..
                                    // in this example RB2/AN2 is the input
AD1CSSL = 0;
AD1CON3 = 0x0000;                   // Sample time is TMR3, Tad = internal TPB*2
AD1CON2 = 0x0004;                   // Interrupt after 2 conversions

                                    // set TMR3 to time out every 125 mSecs
TMR3= 0x0000;
PR3= 0x3FFF;
T3CON = 0x8010;

AD1CON1SET = 0x8000;                // turn ADC ON
AD1CON1SET = 0x0004;                // start auto sampling every 125 mSecs
while (1)                           // repeat continuously
{
  while (!IFS1 & 0x0002){};         // conversion done?
  ADCValue = ADC1BUF0;              // yes then get first ADC value
  IFS1CLR = 0x0002;                 // clear ADIF
}                                   // repeat
```

17.5.13 Sampling a Single Channel Multiple Times

Figure 17-14 and Table 17-4 illustrate a basic configuration of the A/D converter. In this case, one ADC input, AN0, will be acquired and converted. The results are stored in the ADC1BUF buffer. This process repeats 15 times until the buffer is full, and then the module generates an interrupt. Then entire process repeats.

With ALTS (AD1CON2<0>) clear, only the MUX A inputs are active. The CH0SA (AD1CHS<19:16>) bits and CH0NA (AD1CHS<23>) bit are specified (AN0-VREF-) as the input to the sample/hold channel. Other input selection bits are not used.

Figure 17-14: Converting One Channel 15 Times 15 Samples Per Interrupt

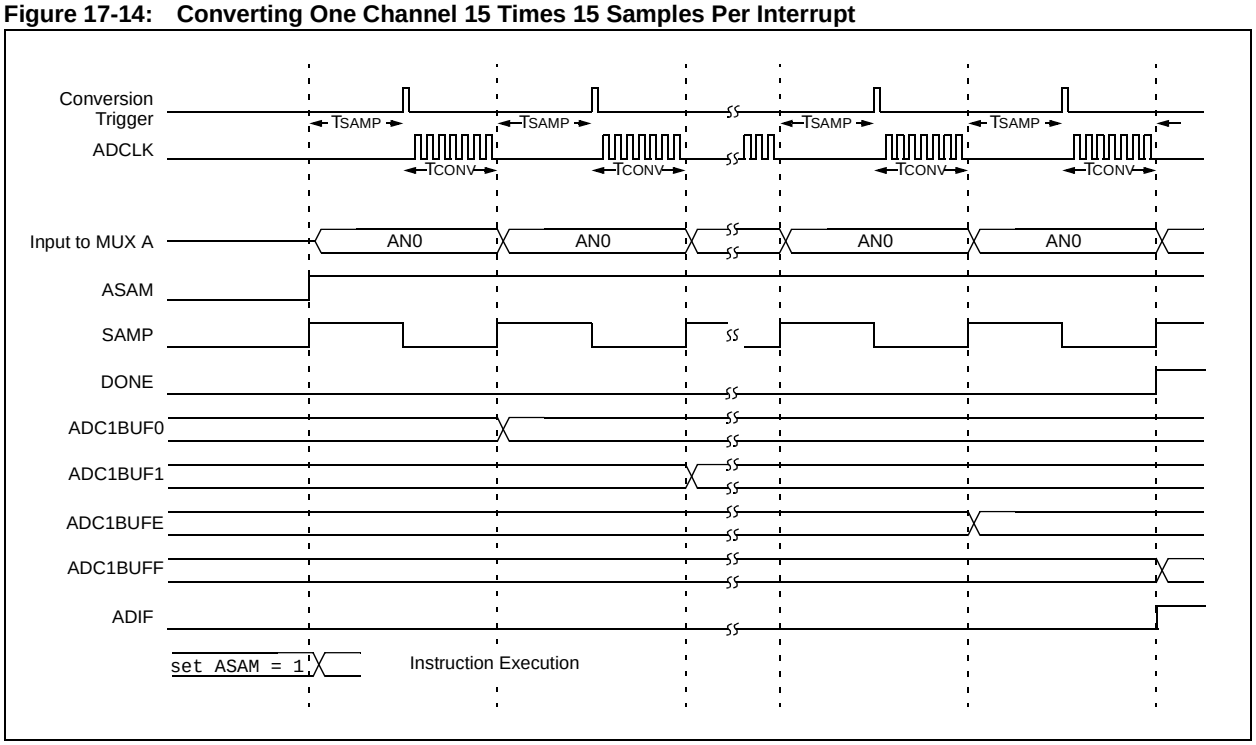


Table 17-4: Converting One Channel 15 Times/Interrupt

CONTROL BITS	OPERATION SEQUENCE
Sequence Select	Sample MUX A Inputs: AN0
SMPI<2:0> = 1111	Convert, Write Buffer 0x0
Interrupt on 15th sample	Sample MUX A Inputs: AN0
—	Convert, Write Buffer 0x1
—	Sample MUX A Inputs: AN0
BUFM = 0	Convert, Write Buffer 0x2
Single 16-word result buffer	Sample MUX A Inputs: AN0
ALTS = 0	Convert, Write Buffer 0x3
Always use MUX A input select	Sample MUX A Inputs: AN0
MUX A Input Select	Convert, Write Buffer 0x4
CH0SA<3:0> = 0000	Sample MUX A Inputs: AN0
Select AN0 for CH0+ input	Convert, Write Buffer 0x5
CH0NA = 0	Sample MUX A Inputs: AN0
Select VR- for CH0- input	Convert, Write Buffer 0x6
CSCNA = 0	Sample MUX A Inputs: AN0
No input scan	Convert, Write Buffer 0x7
CSSL<15:0> = n/a	Sample MUX A Inputs: AN0
Scan input select unused	Convert, Write Buffer 0x8
—	Sample MUX A Inputs: AN0
—	Convert, Write Buffer 0x9
MUX B Input Select	Sample MUX A Inputs: AN0
CH0SB<3:0> = n/a	Convert, Write Buffer 0xA
Mux B positive input unused	Sample MUX A Inputs: AN0
CH0NB = n/a	Convert, Write Buffer 0xB
Mux B negative input unused	Sample MUX A Inputs: AN0
—	Convert, Write Buffer 0xC
—	Sample MUX A Inputs: AN0
	Convert, Write Buffer 0xD
	Sample MUX A Inputs: AN0
	Convert, Write Buffer 0xE
	Interrupt
	Repeat

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

• • •

17.5.14 Example: A/D Conversions While Scanning Through Analog Inputs

Figure 17-15 and Table 17.5.14.1 illustrate a typical setup where all available analog input channels are sampled and converted. The set CSCNA (AD1CON2<10>) bit specifies scanning of the ADC inputs. Other conditions are similar to the previous example, (see **Section 17.5.13 "Sampling a Single Channel Multiple Times"**).

Initially, the AN0 input is acquired and converted. The result is stored in the ADC1BUF buffer. Then the AN1 input is acquired and converted. This process of scanning the inputs repeats 16 times until the buffer is full and then the module generates an interrupt. Then the entire process repeats.

Figure 17-15: Scanning Through 16 Inputs 16 Samples Per Interrupt

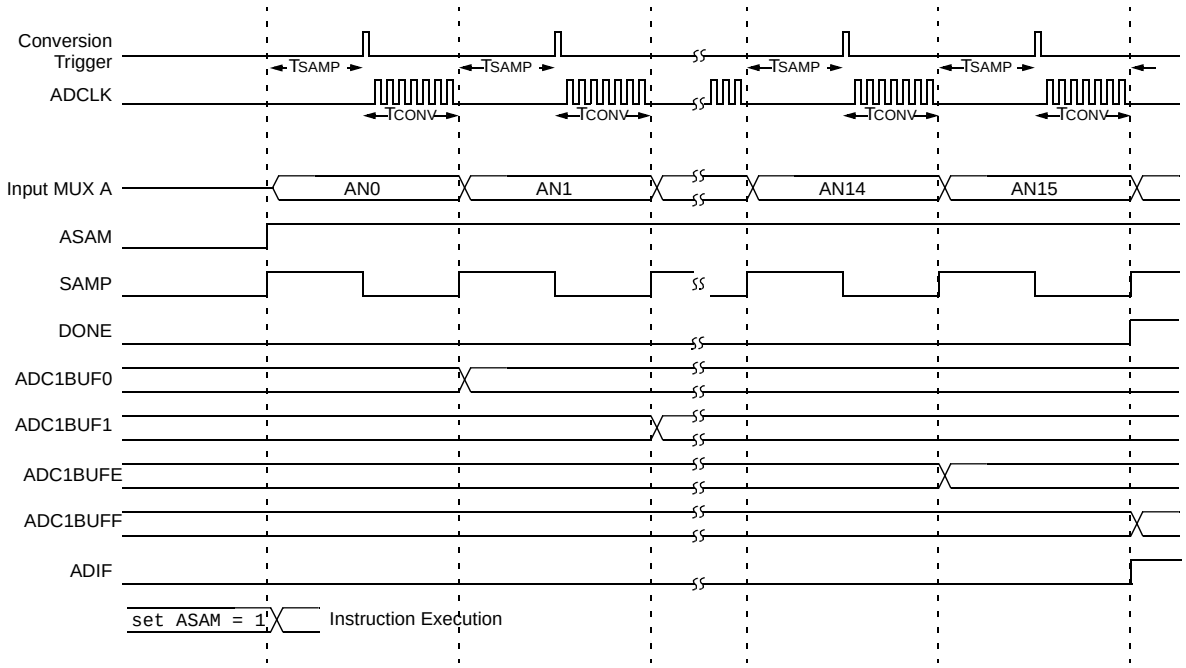


Table 17-5: Scanning Through 16 Inputs/Interrupt

CONTROL BITS	OPERATION SEQUENCE
Sequence Select	Sample MUX A Inputs: AN0
SMPI<2:0> = 1111	Convert, Write Buffer 0x0
Interrupt on 16th sample	Sample MUX A Inputs: AN1
—	Convert, Write Buffer 0x1
—	Sample MUX A Inputs: AN2
BUFM = 0	Convert, Write Buffer 0x2
Single 16-word result buffer	Sample MUX A Inputs: AN3
ALTS = 0	Convert, Write Buffer 0x3
Always use MUX A input select	Sample MUX A Inputs: AN4
MUX A Input Select	Convert, Write Buffer 0x4
CH0SA<3:0> = n/a	Sample MUX A Inputs: AN5
Overridden by CSCNA	Convert, Write Buffer 0x5
CH0NA = 0	Sample MUX A Inputs: AN6
Select VR- for MUX A negative input	Convert, Write Buffer 0x6
CSCNA = 1	Sample MUX A Inputs: AN7
Scan inputs	Convert, Write Buffer 0x7
CSSL<15:0> = 1111 1111 1111 1111	Sample MUX A Inputs: AN8
Scan input select	Convert, Write Buffer 0x8
—	Sample MUX A Inputs: AN9
—	Convert, Write Buffer 0x9
MUX B Input Select	Sample MUX A Inputs: AN10
SB<3:0> = n/a	Convert, Write Buffer 0xA
MUX B positive input unused	Sample MUX A Inputs: AN11
CH0NB = n/a	Convert, Write Buffer 0xB
MUX B negative input unused	Sample MUX A Inputs: AN12
—	Convert, Write Buffer 0xC
—	Sample MUX A Inputs: AN13
	Convert, Write Buffer 0xD
	Sample MUX A Inputs: AN14
	Convert, Write Buffer 0xE
	Sample MUX A Inputs: AN15
	Convert, Write Buffer 0xF
	Interrupt
	Repeat

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

17.5.14.1 Example: Using Dual 8-Word Buffers

Figure 17-16 and Table 17.5.14.2 demonstrate using dual 8-word buffers and alternating the buffer fill. Setting the BUFM (AD1CON2<1>) bit enables dual 8-word buffers. The BUFM setting does not affect other operational parameters. First, the conversion sequence starts filling the buffer at ADC1BUF0 (buffer location 0 x 0). After the first interrupt occurs, the buffer begins to fill at ADC1BUF8 (buffer location 0 x 8). The BUFS (AD1CON2<7>) Status bit is alternately set and cleared after each interrupt to show which buffer is being filled. In this example, three analog inputs are sampled and an interrupt occurs after every third sample.

Figure 17-16: Converting Three Inputs, Three Samples Per Interrupt Using Dual 8-Word Buffers

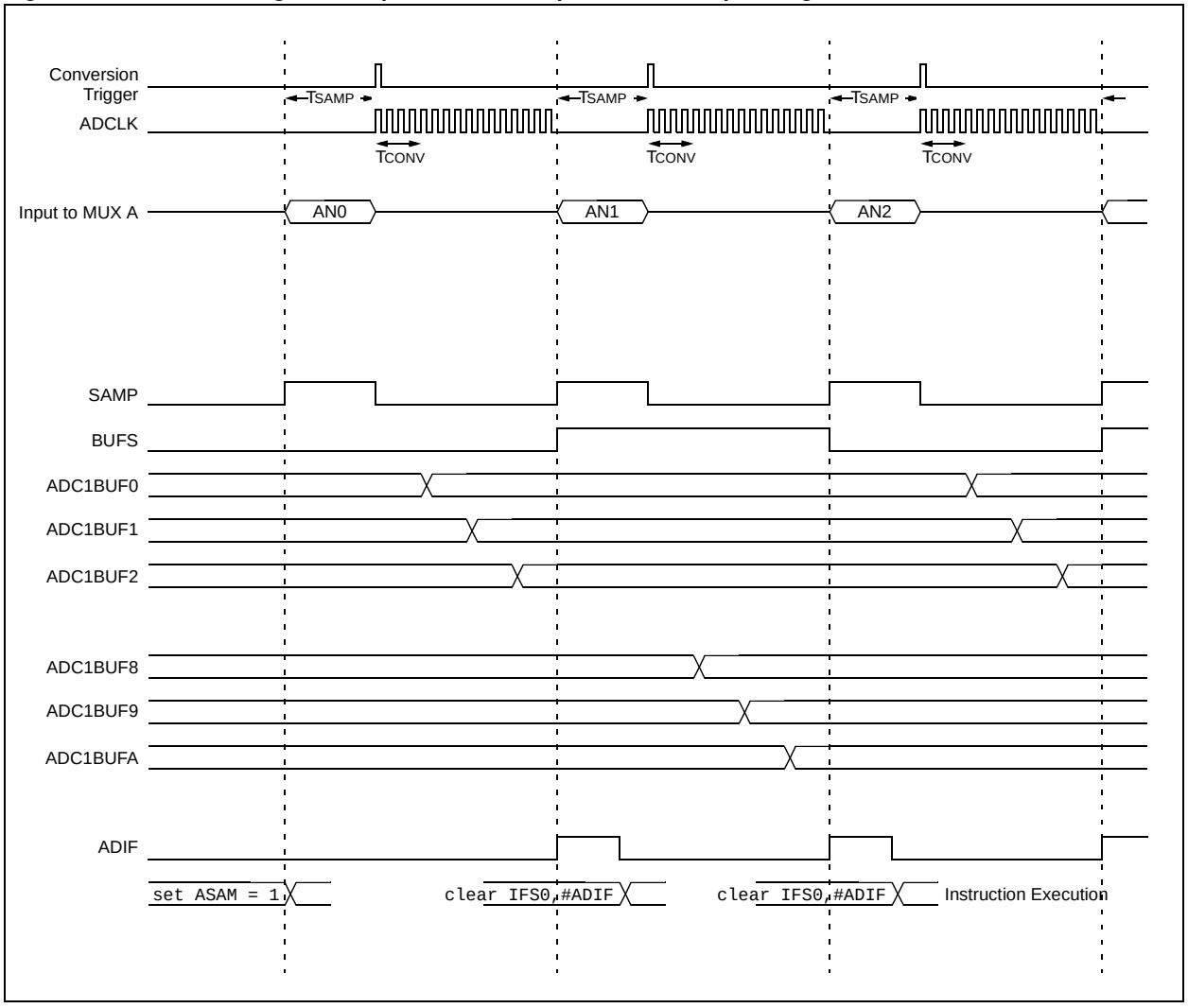


Table 17-6: Converting Three Inputs, Three Samples/Interrupt Using Dual 8-Word Buffers

CONTROL BITS Sequence Select
SMPI<2:0> = 0010 Interrupt after every third sample
—
—
BUFM = 1 Dual 8-word result buffers
ALTS = 0 Always use MUX A
MUX A Input Select
CH0SA<3:0> = n/a MUX A positive input select is not used
CH0NA = 0 Select V_{R-} for MUX A negative input
CSCNA = 1 Enable input scan
CSSL<15:0> = 0x0007 Scan input select scan list consisting of AN0, AN1, and AN2
AD1PCFG = 0X0007 Select Analog Input mode for AN0, AN1, and AN2
—
MUX B Input Select
CH0SB<3:0> = n/a MUX B positive input unused
CH0NB = n/a MUX B negative input unused
—
—

OPERATION SEQUENCE
Sample MUX A Inputs: AN0
Convert AN0, Write Buffer 0x0
Sample MUX A Inputs: AN1
Convert AN1, Write Buffer 0x1
Sample MUX A Inputs: AN2
Convert AN2, Write Buffer 0x2
Interrupt; Change Buffer
Sample MUX A Inputs: AN0
Convert AN0, Write Buffer 0x8
Sample MUX A Inputs: AN1
Convert AN1, Write Buffer 0x9
Sample MUX A Inputs: AN2
Convert AN2, Write Buffer 0xA
Interrupt; Change Buffer
Repeat

Buffer Address	Buffer @ 1st Interrupt	Buffer @ 2nd Interrupt
ADC1BUF0	AN0 sample 1	
ADC1BUF1	AN1 sample 1	
ADC1BUF2	AN2 sample 1	
ADC1BUF3		
ADC1BUF4		
ADC1BUF5		
ADC1BUF6		
ADC1BUF7		
ADC1BUF8		AN0 sample 2
ADC1BUF9		AN1 sample 2
ADC1BUFA		AN2 sample 2
ADC1BUFB		
ADC1BUFC		
ADC1BUFD		
ADC1BUFE		
ADC1BUFF		

• • •

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

Figure 17-17 and Table 17.5.14.3 demonstrate alternating sampling of the inputs assigned to MUX A and MUX B. Setting the ALTS (AD1CON2<0>) bit enables alternating input selections. The first sample uses the MUX A inputs specified by the CH0SA (AD1CHS<19:16>) and CH0NA (AD1CHS<23>) bits. The next sample uses the MUX B inputs specified by the CH0SB (AD1CHS<27:24>) and CH0NB (AD1CHS<31>) bits.

In the following example, one of the MUX B input specifications uses 2 analog inputs as a differential source to the sample/hold.

This example also demonstrates use of the dual 8-word buffers. An interrupt occurs after every 4th sample, which results in filling 4-words into the buffer on each interrupt.

Figure 17-17: Converting Two Analog Inputs by Alternating with Four Samples Per Interrupt

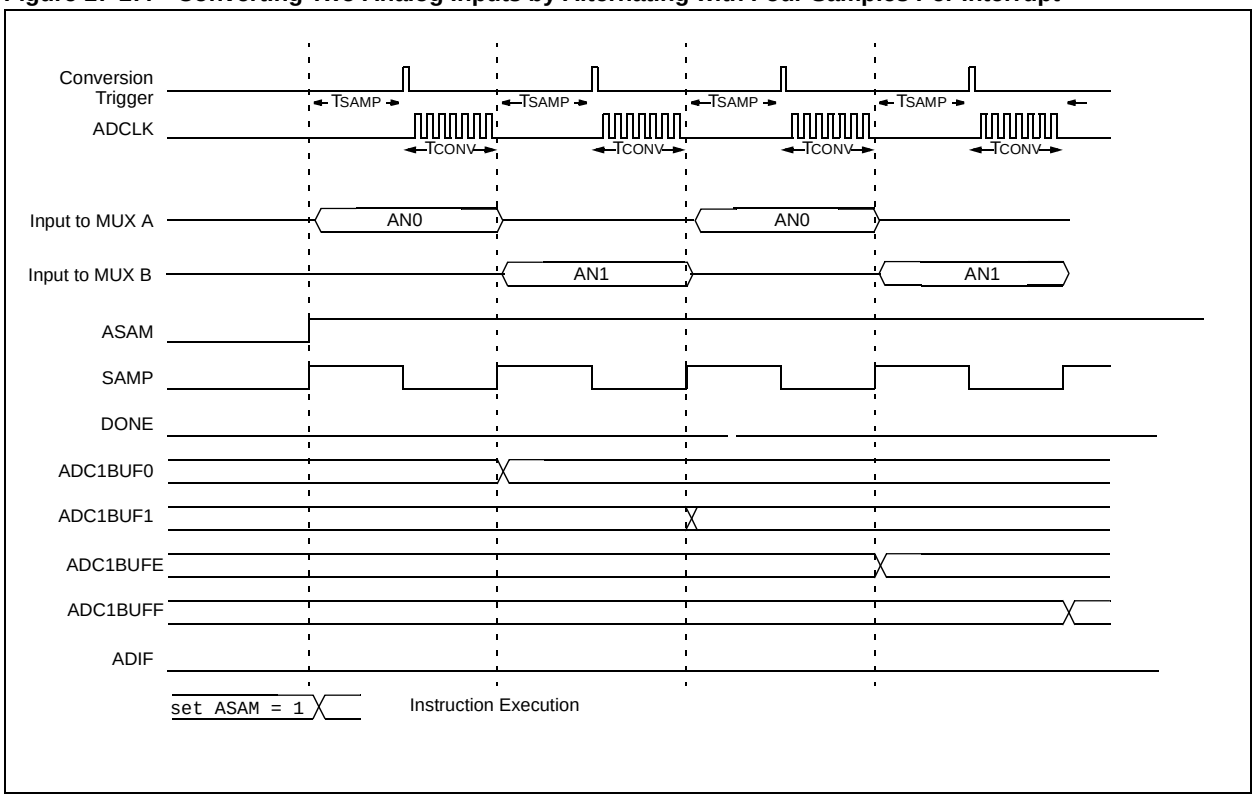


Table 17-7: Converting Two Sets of Inputs Using Alternating Input Selections

CONTROL BITS Sequence Select	OPERATION SEQUENCE
SMPI<2:0> = 0011 Interrupt on 4th sample	Sample MUX A Inputs: AN0 Convert, Write Buffer 0x0 Sample MUX B Inputs: AN1 Convert, Write Buffer 0x1 Sample MUX A Inputs: AN0 Convert, Write Buffer 0x2 Sample MUX B Inputs: AN1 Convert, Write Buffer 0x3 Interrupt; Change Buffer Sample MUX A Inputs: AN0 Convert, Write Buffer 0x8 Sample MUX B Inputs: AN1 Convert, Write Buffer 0x9 Sample MUX A Inputs: AN0 Convert, Write Buffer 0xA Sample MUX B Inputs: AN1 Convert, Write Buffer 0xB Interrupt; Change Buffer **Repeat**
—	
—	
BUFM = 1 Dual 8-word result buffers	
ALTS = 1 Alternate MUX A/B input select	
MUX A Input Select	
CH0SA<3:0> = 0000 Select AN0 for MUX A positive input	
CH0NA = 0 Select VR- for MUX A negative input	
CSCNA = 0 No input scan	
CSSL<15:0> = n/a Scan input select unused	
—	
—	
MUX B Input Select	
CH0SB<3:0> = 0001 Select AN1 for MUX B positive input	
CH0NB = 0 Select VR- for MUX B negative input	
—	
—	

Buffer Address	Buffer @ 1st Interrupt	Buffer @ 2nd Interrupt
ADC1BUF0	AN0 sample 1	
ADC1BUF1	AN1 sample 1	
ADC1BUF2	AN0 sample 2	
ADC1BUF3	AN1 sample 2	
ADC1BUF4		
ADC1BUF5		
ADC1BUF6		
ADC1BUF7		
ADC1BUF8		AN0 sample 3
ADC1BUF9		AN1 sample 3
ADC1BUFA		AN0 sample 4
ADC1BUFB		AN1 sample 4
ADC1BUFC		
ADC1BUFD		
ADC1BUFE		
ADC1BUFF		

• • •

17.5.14.3 Example: Converting Three Analog Inputs Using Alternating Sample Mode and a Scan List

Figure 17-18, Figure 17-19, and Table 17-8 demonstrate sampling by scanning through inputs and alternating between MUX A and MUX B. When the Alternating Sample mode is selected, the first input to be sampled will be the input selected for MUX A, the second sample will be the input selected for MUX B. Then the process repeats. When scanning is combined with Alternating Input mode, the positive input to MUX A is selected by the contents of the AD1CSSL register, not CH0SA. For each sample that MUX A is selected the next item in the scan list is sampled. The positive input to MUX B is selected by CH0SB (AD1CHS<27:24>).

When ASAM (AD1CON1<2>) is clear, sampling will not resume after conversion completion, but will occur when setting the SAMP (AD1CON1<1>) bit.

Figure 17-18: Converting Three Analog Inputs Using Alternating Sample Mode and a Scan List

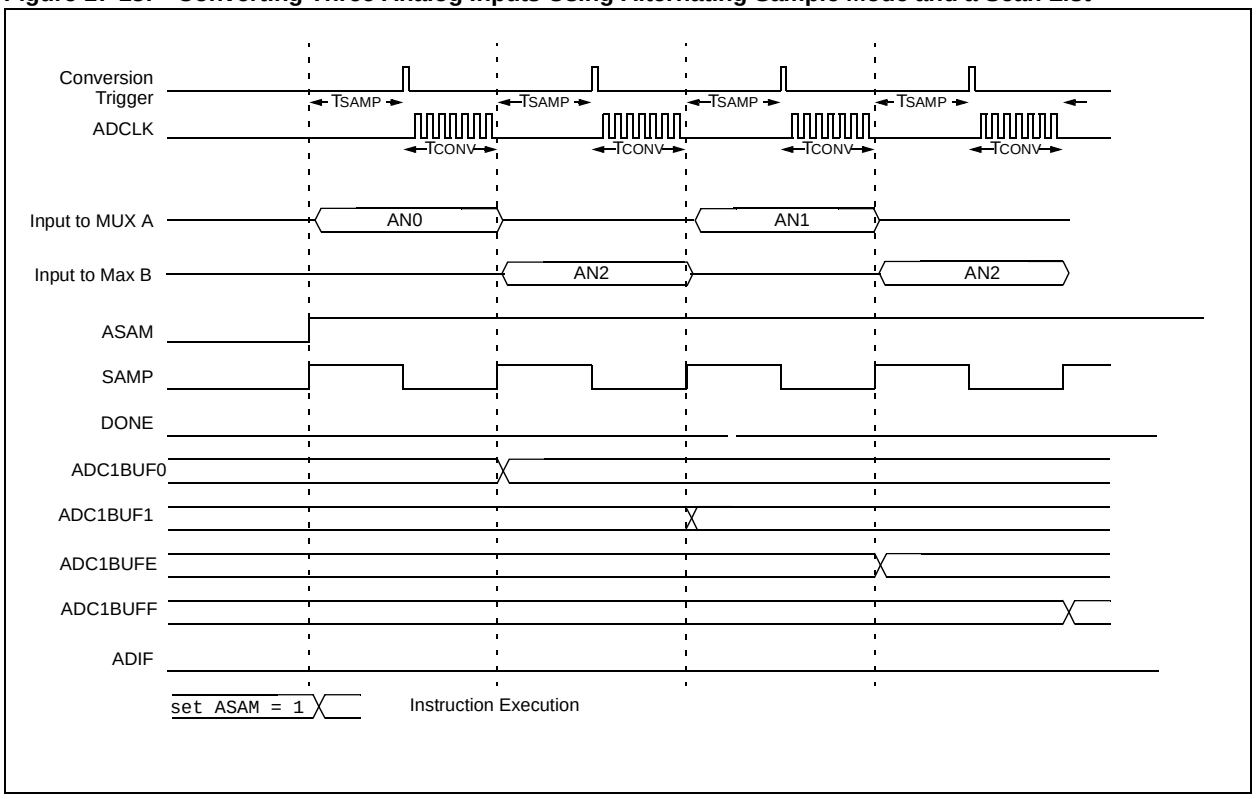


Figure 17-19: 10-Bit High-Speed A/D Converter Block Diagram For Alternating Sample and Scan

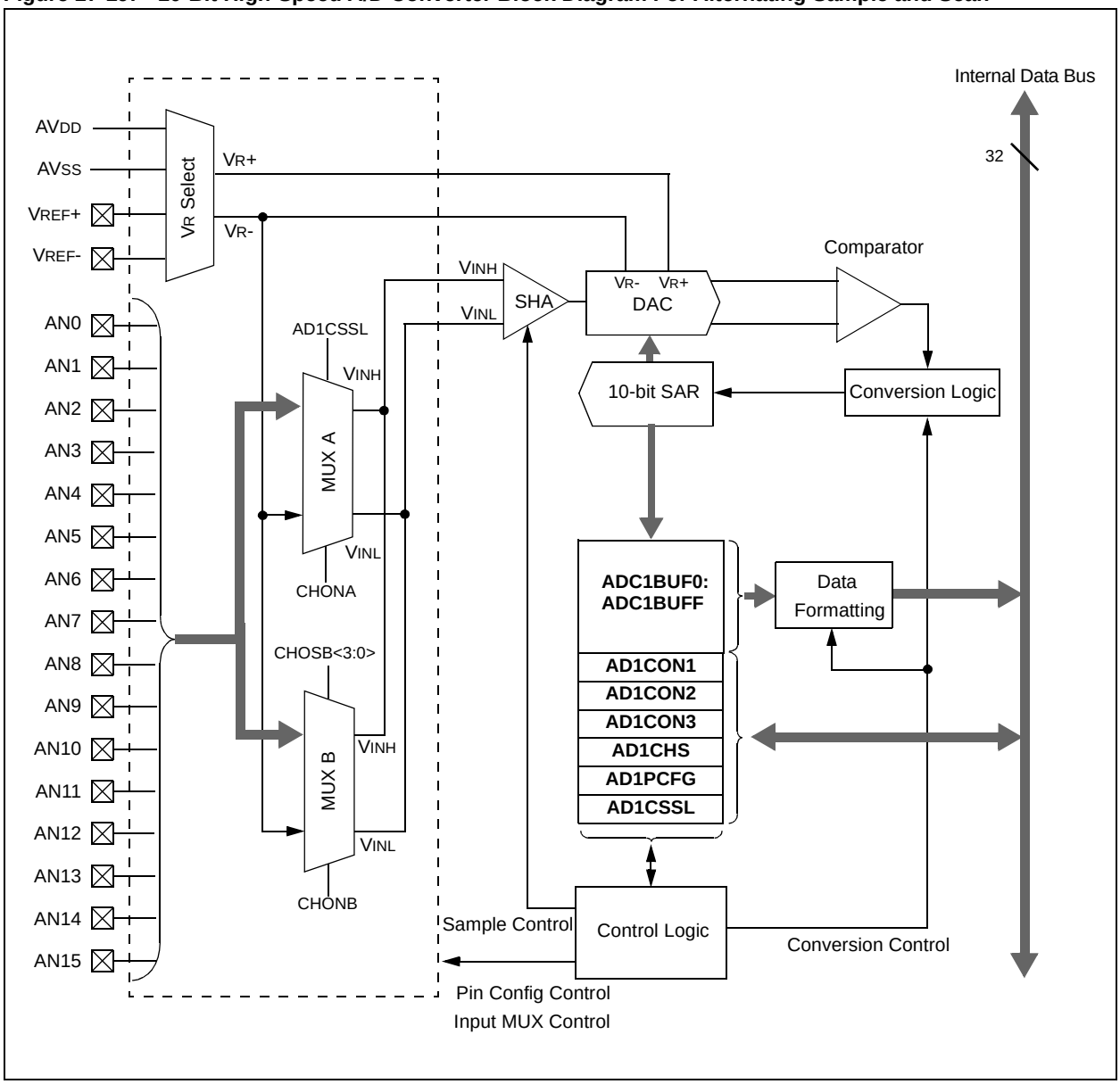


Table 17-8: Sampling Eight Inputs Using Sequential Sampling

CONTROL BITS Sequence Select	OPERATION SEQUENCE
SMPI<2:0> = 0011 Interrupt on 4th sample	Sample: AN0
—	Convert, Write Buffer 0x0
—	Sample: AN2
BUFM = 0 Single 16-word result buffer	Convert, Write Buffer 0x1
ALTS = 1 Alternate MUX A/B input select	Sample: AN1
MUX A Input Select	Convert, Write Buffer 0x2
CH0SA<3:0> = n/a Not used	Sample: AN2
CH0NA = 0 Select V_R- for CH0- input	Convert, Write Buffer 0x3
CSCNA = 1 Enable input scan	
CSSL<15:0> = n/a Scan input select scan list consisting of AN0 and AN1	Interrupt
—	**Repeat**
—	
MUX B Input Select	
CH0SB<3:0> = 0010 Select AN7 for CH0+ input	
CH0NB = 0 Select V_R- for CH0- input	
—	
—	

Buffer Address	Buffer @ 1st Interrupt	Buffer @ 2nd Interrupt
ADC1BUF0	AN0 sample 1	AN0 sample 5
ADC1BUF1	AN2 sample 2	AN2 sample 6
ADC1BUF2	AN1 sample 3	AN1 sample 7
ADC1BUF3	AN2 sample 4	AN2 sample 8
ADC1BUF4		
ADC1BUF5		
ADC1BUF6		
ADC1BUF7		
ADC1BUF8		
ADC1BUF9		
ADC1BUFA		
ADC1BUFB		
ADC1BUFC		
ADC1BUFD		
ADC1BUFE		
ADC1BUFF		

• • •

17.6 INITIALIZATION

A simple initialization code example for the ADC module is provided in Example 17-6.

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 VR+ and VR-. The start of acquisition, as well as start of conversion (conversion trigger), are performed manually in software. The CH0 SHA is used for conversions. Scanning of inputs is disabled, and an interrupt occurs after every acquisition/convert sequence (1 conversion result). The ADC conversion clock is TPB/2.

Since acquisition is started manually by setting the SAMP bit (AD1CON1<1>) after each conversion is complete, the auto-sample time bits, SAMC<4:0> (AD1CON3<12:8>), are ignored. Moreover, since the start of conversion (i.e., end of acquisition) is also triggered manually, the SAMP bit needs to be cleared each time a new sample needs to be converted.

Example 17-6: ADC Initialization Code Example

```
AD1PCFG = 0x0000;             /* Configure ADC port
                                 all input pins are analog */

AD1CON1 = 0x2208;             /* Configure sample clock source and Conversion Trigger mode.
                                 Unsigned Fractional format, Manual conversion trigger,
                                 Manual start of sampling, Simultaneous sampling,
                                 No operation in IDLE mode.  */

AD1CON2 = 0x0000;             /* Configure ADC voltage reference
                                  and buffer fill modes.
                                 VREF from AVDD and AVSS,
                                 Inputs are not scanned,
                                 Interrupt every sample */

AD1CON3 = 0x0000;             /* Configure ADC conversion clock */

AD1CHS = 0x0000;              /* Configure input channels,
                                 CH0+ input is AN0.
                                 CHO- input is VREFL (AVss)

AD1CSSL = 0x0000;             /* No inputs are scanned.
                                 Note: Contents of AD1CSSL are ignored when CSCNA = 0 */

IFS1CLR = 2;                  /*Clear ADC conversion interrupt*/

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

IEC1SET = 2;                  /* Enable ADC conversion interrupt*/

AD1CON1SET = 0x8000;          /* Turn on the ADC module */
AD1CON1SET = 0x0002;          /* Start sampling the input */
DelayNmSec(100);             /* Ensure the correct sampling time has elapsed before
                                 starting a conversion.*/

AD1CON1CLR = 0x0002;          /* End Sampling and start Conversion*/
      :                       /* The DONE bit is set by hardware when the convert sequence
                                 is finished. */
      :                       /* The ADIF bit will be set. */
```

Example 17-7: Converting 1 Channel at 400 ksps, Auto-Sample Start, 2 TAD Sampling Time Code Example

```
AD1PCFG = 0xFFFB;                       //  all PORTB = Digital; RB2 = analog
AD1CON1 = 0x00E0;                       //  SSRC bit = 111 implies internal
                                        //  counter ends sampling and starts
                                        //  converting.
AD1CHS  = 0x00020000;                   //  Connect RB2/AN2 as CH0 input
                                        //  in this example RB2/AN2 is the input
AD1CSSL = 0;
AD1CON3 = 0x0203;                       //  Sample time = 2Tad

AD1CON2 = 0x6004;                       //  Select external VREF+ and VREF- pins
                                        //  Interrupt after every 2 samples
AD1CON1bits.ADON = 1;                   //  turn ADC ON
while (1)                               //  repeat continuously
{
  ADCValue = 0;                             //  clear value
  ADC16Ptr = &ADC1BUF0;                     //  initialize ADC1BUF0 pointer
  IF1bits.AD1IF = 0;                        //  clear ADC interrupt flag
  AD1CON1bits.ASAM = 1;                     //  auto start sampling
                                            //  for 31Tad then go to conversion
  while (!IFS0bits.ADIF);                   //  conversion done?
  AD1CON1bits.ASAM = 0;                     //  yes then stop sample/convert
  for (count = 0; count <2; count++)
  {                                         //  average the two
       ADCValue = ADCValue + *ADC16Ptr++;
       ADCValue = ADCValue >> 1;
  }
}                                           //  repeat
```


17.7 INTERRUPTS

The ADC has a dedicated interrupt bit AD1IF and a corresponding interrupt enable/mask bit AD1IE. These bits are used to determine the source of an interrupt and to enable or disable an individual interrupt source. The priority level of each of the channels can also be set independently of the other channels.

The AD1IF is set when the condition set by the Samples Per Interrupt bit SMPI<3:0> (AD1CON2<5:2>) is met. The AD1IF bit will then be set without regard to the state of the corresponding AD1IE bit. The AD1IF bit can be polled by software if desired.

The AD1IE bit controls the interrupt generation. If the AD1xIE bit is set, the CPU will be interrupted whenever an event defined by SMPI<3:0> occurs and the corresponding AD1IF bit will be set (subject to the priority and sub priority as outlined below).

It is the responsibility of the routine that services a particular interrupt to clear the appropriate Interrupt Flag bit before the service routine is complete.

The priority of the ADC interrupt can be set independently via the AD1IP<2:0> (IPC6<28:26>) bits. This priority defines the priority group that interrupt source will be assigned to. The priority groups range from a value of 7, the highest priority, to a value of 0, which does not generate an interrupt. An interrupt being serviced will be preempted by an interrupt in a higher priority group.

The subpriority bits allow setting the priority of a interrupt source within a priority group. The values of the subpriority, AD1xIS<1:0> (IPC6<25:24>), range from 3, the highest priority, to 0 the lowest priority. An interrupt with the same priority group but having a higher subpriority value will preempt a lower subpriority interrupt that is in progress.

The priority group and subpriority bits allow more than one interrupt source to share the same priority and subpriority. If simultaneous interrupts occur in this configuration the natural order of the interrupt sources within a priority/subgroup pair determine the interrupt generated. The natural priority is based on the vector numbers of the interrupt sources. The lower the vector number the higher the natural priority of the interrupt. Any interrupts that were overridden by natural order will then generate their respective interrupts based on priority, subpriority, and natural order after the interrupt flag for the current interrupt is cleared.

After an enabled interrupt is generated, the CPU will jump to the vector assigned to that interrupt (refer to Table 17-9). The vector number for the interrupt is the same as the natural order number. The IRQ number is not always the same as the vector number due to some interrupts sharing a single vector. The CPU will then begin executing code at the vector address. The users code at this vector address should perform an operations required, such as reloading the duty cycle, clear the interrupt flag, and then exit. Refer to **Section 8. "Interrupts"** for vector address table details and for more information on interrupts.

Table 17-9: ADC Interrupt Vectors for Various Offsets with EBASE = 0x8000:0000

Interrupt	Vector/Natural Order	IRQ Number	Vector Address IntCtl.VS = 0x01	Vector Address IntCtl.VS = 0x02	Vector Address IntCtl.VS = 0x04	Vector Address IntCtl.VS = 0x08	Vector Address IntCtl.VS = 0x10
ADC	27	32	8000 0560	8000 08C0	8000 0F80	8000 1D00	8000 3800

Example 17-8: ADC Interrupt Configuration Code Example

```
IPS6SET = 0x0014;                       //  Set Priority to 5
IPS6SET = 0x0003;                       //  Set Sub Priority to 3
                                        //
IFS1CLR = 0x0002;                       // Ensure the interrupt flag is clear
IEC1SET = 0x0002;                       // Enable ADC interrupts
```

17.8 I/O PIN CONTROL

The pins used for analog input can also be used for digital I/O. Configuring a pin for analog input requires three steps. Any digital peripherals that share the desired pin must be disabled. The pin must be configured as a Digital input, by setting the corresponding TRIS bit to a '1', to disable the output driver. Then the pin must be placed in Analog mode by setting the corresponding bit in the AD1PCFG register.

Table 17-10: Pins Associated with the ADC Module

Pin Name	Module Control	Controlling Bit Field	Pin Type	Buffer Type	TRIS	Description
AN0	ON	AD1PCFG<0>	A	—	Input	Analog input
AN1	ON	AD1PCFG<1>	A	—	Input	Analog input
AN2	ON	AD1PCFG<2>	A	—	Input	Analog input
AN3	ON	AD1PCFG<3>	A	—	Input	Analog input
AN4	ON	AD1PCFG<4>	A	—	Input	Analog input
AN5	ON	AD1PCFG<5>	A	—	Input	Analog input
AN6	ON	AD1PCFG<6>	A	—	Input	Analog input
AN7	ON	AD1PCFG<7>	A	—	Input	Analog input
AN8	ON	AD1PCFG<8>	A	—	Input	Analog input
AN9	ON	AD1PCFG<9>	A	—	Input	Analog input
AN10	ON	AD1PCFG<10>	A	—	Input	Analog input
AN11	ON	AD1PCFG<11>	A	—	Input	Analog input
AN12	ON	AD1PCFG<12>	A	—	Input	Analog input
AN13	ON	AD1PCFG<13>	A	—	Input	Analog input
AN14	ON	AD1PCFG<14>	A	—	Input	Analog input
AN15	ON	AD1PCFG<15>	A	—	Input	Analog input
$V_{REF}+$	ON	AD1CON2<15:13>	P	—	—	Positive voltage reference
$V_{REF}-$	ON	AD1CON2<15:13>	P	—	—	Negative voltage reference

Legend:	ST = Schmitt Trigger input with CMOS levels	A = Analog
	I = Input	P = Power
	O = Output	

17.9 OPERATION DURING SLEEP AND IDLE MODES

Note:	In this manual, a distinction is made between a power mode as it is used in a specific module, and a power mode as it is used by the device, e.g., Sleep mode of the Comparator and SLEEP mode of the CPU. To indicate which type of power mode is intended, uppercase and lowercase letters (Sleep, Idle, Debug) signify a module power mode, and all uppercase letters (SLEEP, IDLE, DEBUG) signify a device power mode.

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

17.9.1 CPU SLEEP Mode Without RC ADC Clock

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

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

ADC register contents are not affected by the device entering or leaving SLEEP mode.

17.9.2 CPU SLEEP Mode With RC ADC Clock

The ADC module can operate during SLEEP mode if the ADC clock source is set to the internal ADC RC oscillator (ADRC = 1). This reduces the digital switching noise from the conversion. When the conversion is completed, the DONE bit will be set and the result loaded into the ADC result buffer, ADC1BUF.

If the ADC interrupt is enabled (AD1IE = 1), the device will wake up from SLEEP when the ADC interrupt occurs. Program execution will resume at the ADC Interrupt Service Routine if the ADC interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the WAIT instruction that placed the device in SLEEP mode.

If the ADC interrupt is not enabled, the ADC module will then be disabled, although the ON bit will remain 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 ADC ON bit should be set in the instruction prior to the WAIT instruction.

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

17.9.3 ADC Operation During CPU IDLE Mode

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

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

17.9.4 Effects of Freeze on ADC Operation

If Freeze mode is entered while the ADC is performing a conversion the result of the conversion will be lost.

While in Freeze mode the ADC registers can be read. Any writes to the ADC register while in Freeze mode will not take effect until the device exits Freeze mode.

Note:	The FRZ bit is readable and writable only when the CPU is executing in Debug Exception mode. In all other modes, the FRZ bit reads as '0'. If FRZ bit is changed during DEBUG mode, the new value does not take effect until the current Debug Exception mode is exited and re-entered. During the Debug Exception mode, the FRZ bit reads the state of the peripheral when entering DEBUG mode.

17.10 EFFECTS OF VARIOUS RESETS

17.10.1 $\overline{\text{MCLR}}$ Reset

Following a $\overline{\text{MCLR}}$ event all the ADC control registers (AD1CON1, AD1CON2, AD1CON3, AD1CHS, AD1PCFG, and AD1CSSL) are reset to a value of 0x00000000. This disables the ADC and sets the analog input pins to Analog Input mode. Any conversion that was in progress will terminate and the result will not be written to the result buffer.

The values in the ADC1BUF registers are initialized during a $\overline{\text{MCLR}}$ Reset. ADC1BUF0...ADC1BUFF will contain 0x00000000.

17.10.2 Power-on Reset

Following a POR event all the ADC control registers (AD1CON1, AD1CON2, AD1CON3, AD1CHS, AD1PCFG, and AD1CSSL) are reset to a value of 0x00000000. This disables the ADC and sets the analog input pins to Analog Input mode.

The values in the ADC1BUF registers are initialized during a Power-on Reset. ADC1BUF0...ADC1BUFF will contain 0x00000000.

17.10.3 Watchdog Timer Reset

Following a Watchdog Timer (WDT) Reset all the ADC control registers (AD1CON1, AD1CON2, AD1CON3, AD1CHS, AD1PCFG, and AD1CSSL) are reset to a value of 0x00000000. This disables the ADC and sets the analog input pins to Analog Input mode. Any conversion that was in progress will terminate and the result will not be written to the result buffer.

The values in the ADC1BUF registers are initialized after a WDT Reset.

ADC1BUF0...ADC1BUFF will contain 0x00000000.

17.11 DESIGN TIPS

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

Answer: The following tips can be helpful for optimizing performance:

1. Make sure you are meeting all of the timing specifications. If you are turning the module off and on, there is a minimum delay you must wait before taking a sample. If you are changing input channels, there is a minimum delay you must wait for this as well. Also, there is TAD, which is the time selected for each bit conversion. This is selected in AD1CON3 and should be within a certain range as specified in 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 to charge the sample capacitor can affect accuracy. If the input signal does not change too quickly, try putting a 0.1 μF capacitor on the analog input. This capacitor will charge to the analog voltage being sampled and supply the instantaneous current needed to charge the 4.4 pF internal holding capacitor.
3. Put the device into SLEEP mode before the start of the A/D conversion. The RC clock source selection is required for conversions in SLEEP mode. This technique increases accuracy because digital noise from the CPU and other peripherals is minimized.

Question 2: ***Do you know of a good reference on ADCs?***

Answer: The following handbook can assist with a good understanding of A/D conversions:

Analog Devices, Inc., and Scheingold, D. H., ed. *Analog-Digital Conversion Handbook.* 3rd ed., Englewood Cliffs, NJ: Prentice Hall, 1986. ISBN 0-13-032848-0.

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

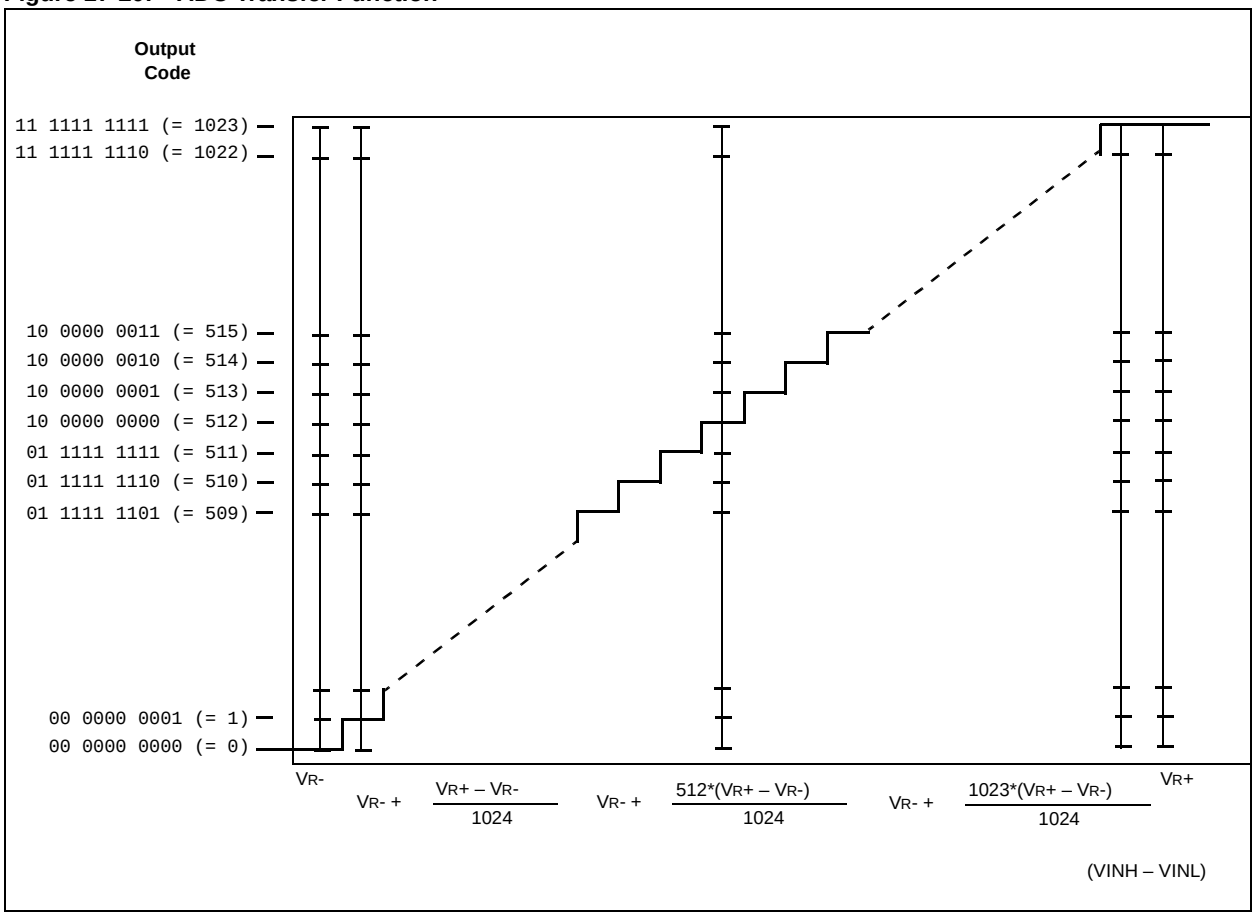
Answer: This configuration is not recommended. The buffer will contain the results of the first 16 samples (or 8, if a Dual Buffer mode is used) in the conversion sequence. The remaining items in the conversion sequence will be ignored.

17.11.1 Transfer Function

The ideal transfer function of the A/D converter is shown in Figure 17-20. The difference of the input voltages, (V_{INH} – V_{INL}), is compared to the reference, (V_{R+} – V_{R-}).

- The first code transition occurs when the input voltage is (V_{R+} – V_{R-L}/2048) or 0.5 LSb.
- The `00 0000 0001` code is centered at (V_{R+} – V_{R-}/1024) or 1.0 LSb.
- The `10 0000 0000` code is centered at (512*(V_{R+} – V_{R-})/1024).
- An input voltage less than (1 × (V_{R+} – V_{R-L})/2048) converts as `00 0000 0000`.
- An input greater than (2045 × (V_{R+} – V_{R-})/2048) converts as `11 1111 1111`.

Figure 17-20: ADC Transfer Function

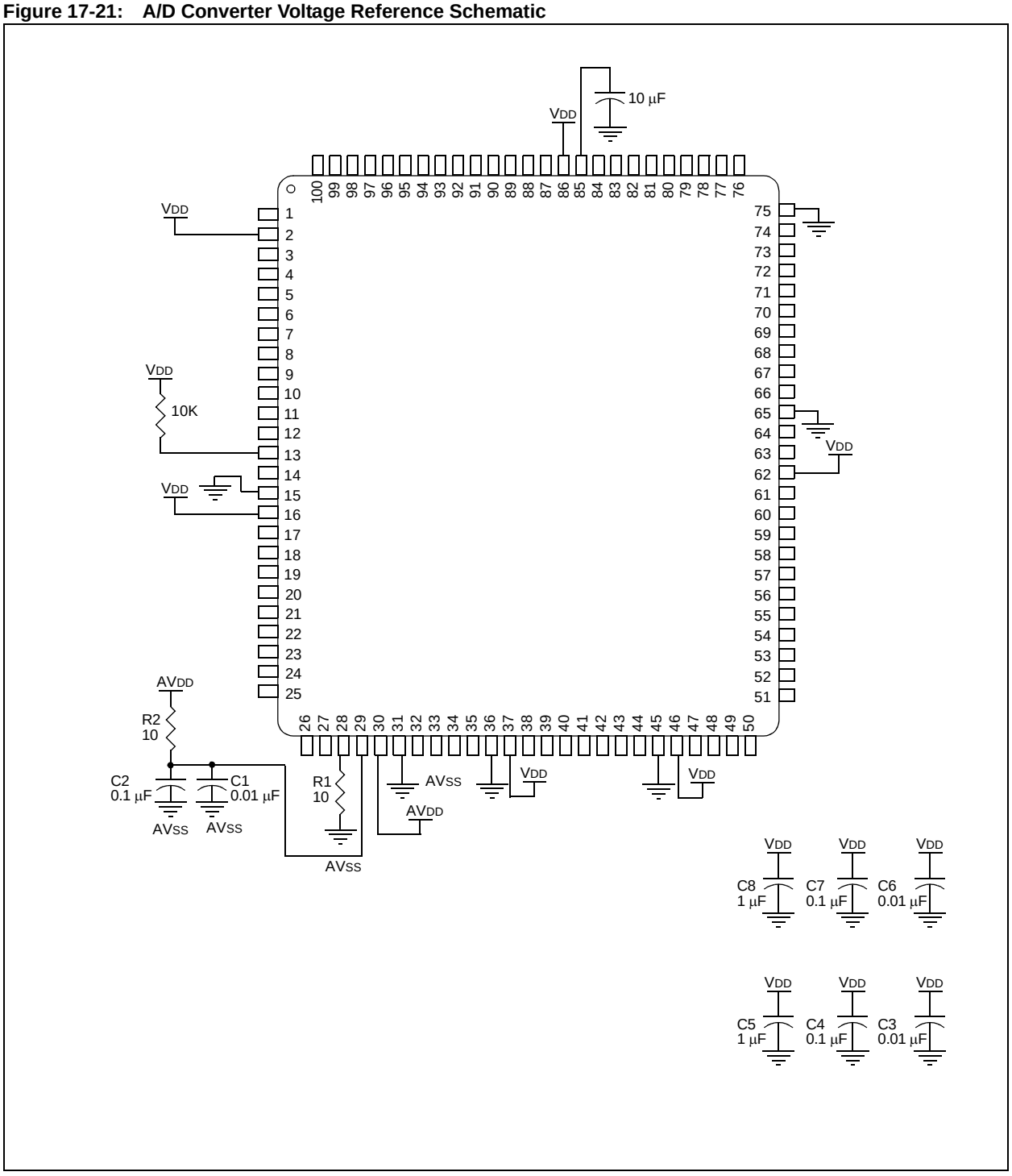


17.11.2 ADC Accuracy/Error

Refer to **Section 17.12 "Related Application Notes"** for a list of documents that discuss ADC accuracy.

The following figure depicts the recommended circuit for the conversion rates above 400 ksps. The PIC32MX is shown as an example.

Figure 17-21: A/D Converter Voltage Reference Schematic



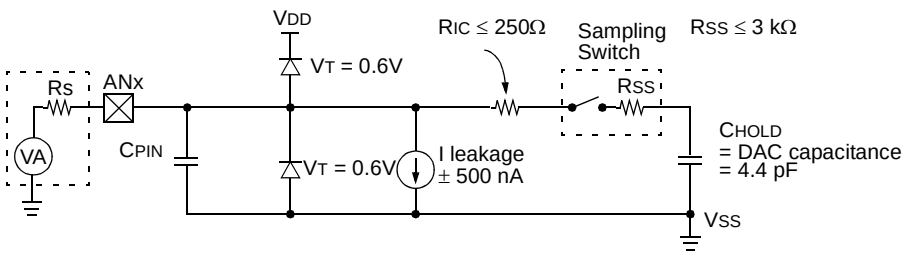
17.11.3 ADC Sampling Requirements

The analog input model of the 10-bit A/D converter is shown in Figure 17-22. The total acquisition time for the A/D conversion 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 analog output source impedance (RS), the interconnect impedance (RIC), and the internal sampling switch (RSS) impedance combine to directly affect the time required to charge the CHOLD. The combined impedance of the analog sources must therefore be small enough to fully charge the holding capacitor within the chosen sample time. After the analog input channel is selected (changed), this acquisition 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 acquisition time. For more details, see the device electrical specifications.

Figure 17-22: 10-Bit A/D Converter Analog Input Model



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

Legend:

CPIN = input capacitance

VT = threshold voltage

RSS = sampling switch resistance

RIC = interconnect resistance

RS = source resistance

CHOLD = sample/hold capacitance

I leakage = leakage current at the pin due to various junctions

17.11.4 Connection Considerations

Since the analog inputs employ ESD (Electrostatic Discharge) 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 acquisition 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.

17.12 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 PIC32MX device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the 10-bit A/D Converter module are:

Title	Application Note #
Using the Analog-to-Digital (A/D) Converter	AN546
Four Channel Digital Voltmeter with Display and Keyboard	AN557
Understanding A/D Converter Performance Specifications	AN693
Using the dsPIC30F for Sensorless BLDC Control	AN901
Using the dsPIC30F for Vector Control of an ACIM	AN908
Sensored BLDC Motor Control Using the dsPIC30F2010	AN957
An Introduction to AC Induction Motor Control Using the dsPIC30F MCU	AN984

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the PIC32MX family of devices.

17.13 REVISION HISTORY

Revision A (October 2007)

This is the initial released version of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised U-0 to r-x.

Revision D (June 2008)

Revised Register 17-1 note; Revised Registers 17-13, 17-17, 17-21, 17-25, 17-26; Revised Equation 17-1; Added Section 17.5.6; Revised Tables 17-4, 17-5, 17-6, 17-7, 17-8; Delete Section 17.11.5 (500 KSPS Configuration Guideline); Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (AD1CON1 Register).

