Section 65. 12-Bit, High-Speed Pipeline A/D Converter

HIGHLIGHTS

This section of the manual contains the following major topics:

65.1 INTRODUCTION

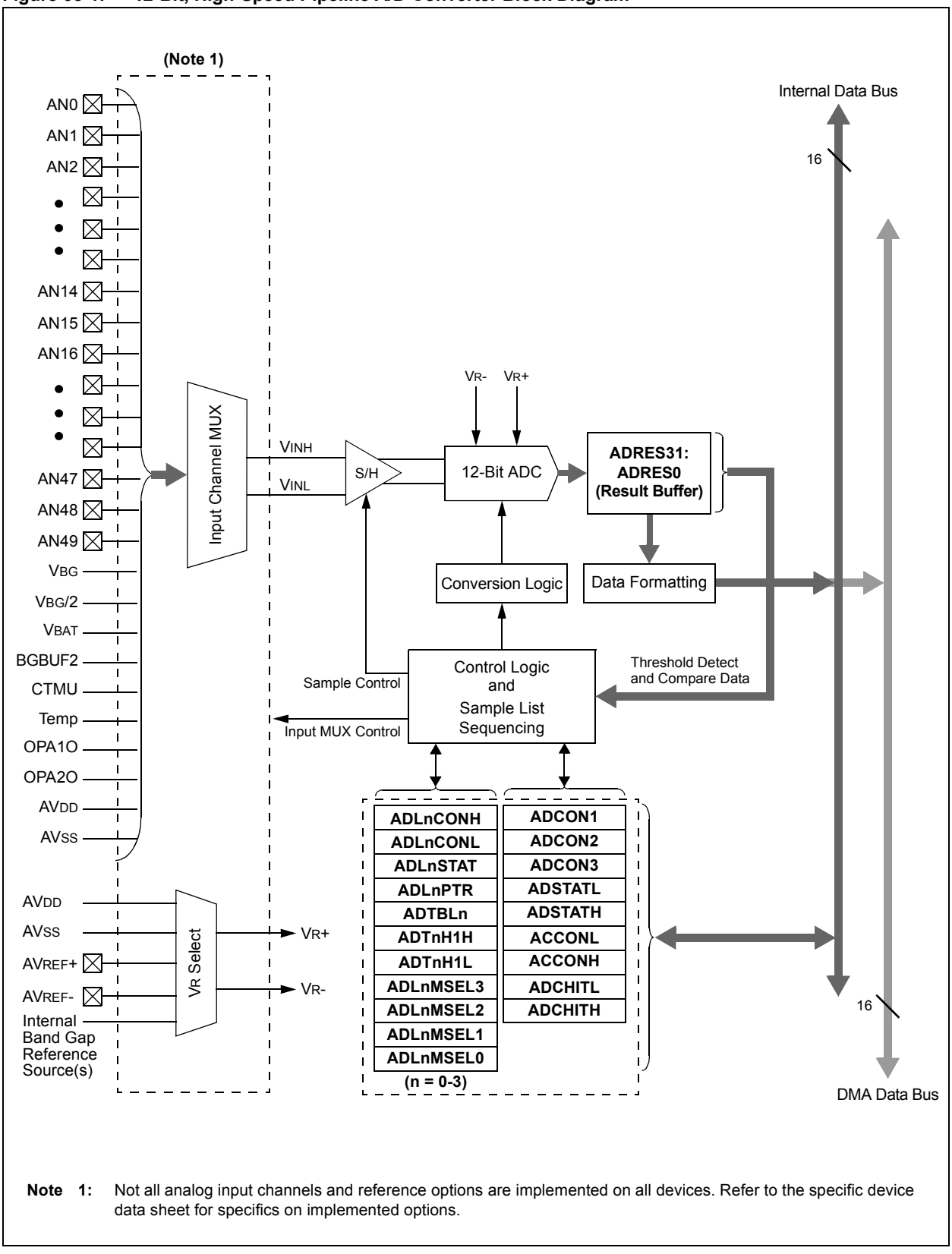
The 12-Bit, High-Speed Pipeline A/D Converter module represents a departure from previous PIC24F A/D Converters, incorporating many enhancements for speed and autonomous operation. It is capable of sampling up to once per A/D clock cycle, and supports a wide range of automatic sampling and accumulation options that reduce the need for CPU intervention during even the most complex operations. It also has tighter integration with other on-chip analog modules and a configurable results buffer.

The 12-Bit, High-Speed Pipeline A/D Converter has the following key features:

- Conversion Speeds of up to 10 Msps
- Up to 50 Analog Single-Ended, Input Channels or up to 15 Unique Differential Input Channel Pairs
- Multiple Internal Reference Input Channels
- Multiple Internal and External Voltage Reference Options
- 12-Bit Conversion Resolution
- Automated Threshold Detect (Scan and Compare) Operation to Pre-Evaluate Conversion Results
- Extended, Fully Programmable Channel Scanning Sequences from up to Four Different Sample Lists
- Automated Conversion Result Accumulation
- Selectable Conversion Trigger Source
- Internal 32-Word, Configurable Conversion Result Buffer
- Eight Options for Results Alignment
- Configurable Interrupt Generation
- Operation during CPU Sleep, Idle and Doze modes, with Extended Module-Specific Power-Saving Options

A simplified block diagram for the module is shown in Figure 65-1.

Figure 65-1: 12-Bit, High-Speed Pipeline A/D Converter Block Diagram



Note 1: Not all analog input channels and reference options are implemented on all devices. Refer to the specific device data sheet for specifics on implemented options.

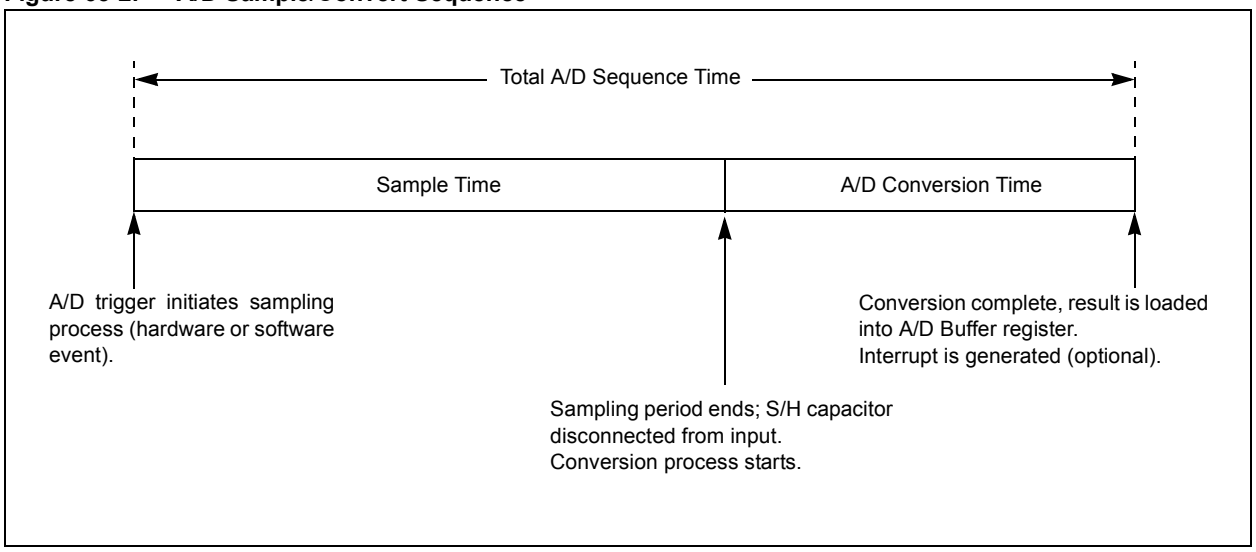
65.2 A/D TERMINOLOGY AND CONVERSION SEQUENCE

Sample time is the time that the A/D module's Sample-and-Hold (S/H) capacitor is connected to the analog input pin. The sample time is started and ended automatically by the A/D Converter's hardware, in response to a trigger event or under direct program control. There is a minimum sample time to ensure that the S/H capacitor will give sufficient accuracy for the A/D conversion.

The A/D trigger event starts the sampling time. Once this time has elapsed, the hardware automatically begins an A/D conversion. The A/D trigger sources can be taken from a variety of hardware sources or can be controlled directly in software.

Conversion time is the time required for the A/D Converter to convert the voltage held by the S/H capacitor. When the conversion is complete, the result is loaded into one of the A/D result buffers. A CPU interrupt can also be generated once the conversion result is available. Figure 65-2 shows the basic conversion sequence and the relationship between intervals.

Figure 65-2: A/D Sample/Convert Sequence



65.3 REGISTERS

Depending on the specific device, the Pipeline A/D Converter implements up to 151 registers. The majority of these are Sample List Control registers and A/D Results Buffer registers. At a minimum, only six registers need to be programmed to use the module and generate conversion results.

Refer to the specific device data sheet for details on which registers and features are actually implemented.

65.3.1 Control Registers

In its full implementation, the Pipeline A/D Converter uses 113 control registers.

Five registers control global operation:

- ADCON1 through ADCON3 (Register 65-1 through Register 65-3) control core A/D operations, including sampling clock rate, selection of voltage references, operation of charge pumps, result buffer configuration, data format and use of Sample Lists.
- ADSTATH and ADSTATL (Register 65-4 and Register 65-5) contain the status flags for top-level module functions, as well as the lower level interrupt flags.

Ninety-six registers control the selection, configuration and execution of the Sample Lists:

- ADLnCONH and ADLnCONL (n = 0 through 3) (prototype registers, Register 65-6 and Register 65-7) configure Sample List operations, including Sample List size, trigger sources, Threshold Detect and Compare mode, and interrupt generation. There is one ADLnCONL and ADLnCONH register for each of the four Sample Lists.
- ADLnSTAT (n = 0 through 3) (prototype register, Register 65-8) contains the status and interrupt flags for the Sample Lists. There is one of each register for each of the four Sample Lists.
- ADLnPTR (n = 0 through 3) (prototype register, Register 65-9) contains the Address Pointer for the Sample List. There is one register for each Sample List.
- ADTBLn (n = 0 through up to 63) (prototype register, Register 65-10) configures the sampling and channel selection setup for each item in the Sample List. All implemented registers are available to any of the four Sample Lists.
- ADLnMSELx (n = 0 through 3, x = 0 through 3) (prototypes, Register 65-17 through Register 65-20) selects the analog channels to be used with the Sample Lists for multi-channel sampling. There are four ADLnMSELx registers (ADLnMSEL0 through ADLnMSEL3) for each of the Sample Lists, for a total of 16 registers.

Twelve registers control accumulator and Threshold Detect operations:

- ACCONH and ACCONL (Register 65-11 and Register 65-12) configure the operation of the hardware accumulator.
- AD1CHITH and AD1CHITL (Register 65-13 and Register 65-14) track any positive results per Sample List item during threshold compare operations.
- ADTHnH and ADTHnL (n = 0 through 3) (prototype registers, Register 65-15 and Register 65-16) set the high and low values, respectively, for threshold compare operation on Sample Lists. There are two registers (ADTHnH and ADTHnL) for each Sample List, for a total of 8 registers.
- ADTMRPR is a 16-bit counter that defines the period for the trigger timer. See **Section 65.6.1 "Internal Trigger Timer"** for more details.

65.3.2 Data and Buffer Registers

In its full implementation, the Pipeline A/D Converter uses 35 data registers.

ADRES0 through ADRES31 are the memory-mapped result buffers. Depending on the operating and buffer modes selected, the result buffer number may correspond to a particular to an item in a Sample List, or to the item's place in the sampling order. Even though the conversion resolution is 12 bits, result buffers are a full 16 bits wide to accommodate different data formatting options. See **Section 65.5.2 "Buffer Data Formats"** for more information. Depending on the specific device, not all of the 32 result buffers may be implemented.

ACCRESH and ACCRESL store the 32-bit results of hardware Accumulator operations. See **Section 65.12 "Sample-and-Accumulate Operations"** for more details.

Register 65-1: ADCON1: A/D Control Register 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	ADSLP	FORM3	FORM2	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
PUMPEN	ADCAL(2)	—	—	—	—	—	PWRLVL(3)
bit 7							bit 0

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

bit 15 **ADON: A/D** Module Enable bit
1 = Module is enabled
0 = Module is disabled (registers are still readable and writable)

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

bit 13 **ADSIDL:** A/D Stop in Idle Control bit
1 = Halts when CPU is in Idle mode
0 = Continues to operate in CPU Idle mode

bit 12 **ADSLP:** A/D Suspend in Sleep Control bit
1 = Continues operation in Sleep mode
0 = Ignores triggers and clocks when CPU is in Sleep mode

bit 11-8 **FORM<3:0>:** Data Output Format bits
1xxx = Unimplemented, do not use
0111 = Signed Fractional (sddd dddd dddd 0000)
0110 = Fractional (dddd dddd dddd 0000)
0101 = Signed Integer (ssss sddd dddd dddd)
0100 = Integer (0000 dddd dddd dddd)
0011 = Signed Fractional (sddd dddd dddd 0000)
0010 = Fractional (dddd dddd dddd 0000)
0001 = Signed Integer (ssss sddd dddd dddd)
0000 = Integer, Raw Data (0000 dddd dddd dddd)

bit 7 **PUMPEN:** Analog Channel Switch Charge Pump Enable bit
1 = Charge pump for switches is enabled, reducing switch impedance(1)
0 = Charge pump for switches is disabled

bit 6 **ADCAL:** A/D Internal Analog Calibration bit(2)
1 = Initiates internal analog calibration
0 = No operation

bit 5-1 **Unimplemented:** Read as '0'

bit 0 **PWRLVL:** Power Level Select bit(3)
1 = Full power, maximum conversion rate; A/D clock rates from 1 MHZ to 10 MHz are allowed
0 = Low power, reduced frequency operation; A/D clock rates from 1 MHz to 2.5 MHz are allowed

Note 1: Use of the channel switch charge pump is recommended when AV_{DD} < 2.5V.

2: When set, ADCAL remains set for at least one T_{AD} and is then automatically cleared by hardware. Manually clearing the bit does not necessarily cancel the calibration routine. Calibration is complete when ADSTATH<1> = 1.

3: Whenever PWRLVL is set to '1', a delay of 50 μS should be implemented before the next A/D conversion.

Register 65-2: ADCON2: A/D Control Register 2

R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-1	R/W-1
PVCFG1	PVCFG0	—	NVCFG0	—	BUFORG	ADPWR1(1)	ADPWR0(1)
bit 15							bit 8

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
BUFINT1(2)	BUFINT0(2)	—	—	—	—	REFPUMP(3)	ADHALT(1)
bit 7							bit 0

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

bit 15-14 **PVCFG<1:0>:** A/D Converter Voltage Reference Configuration for ADREF+ bits
11 = Unimplemented
10 = Internal band gap source
01 = External VREF+
00 = AVDD

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

bit 12 **NVCFG0:** A/D Converter Voltage Reference Configuration for ADREF- bit
1 = External VREF-
0 = AVSS

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

bit 10 **BUFORG:** Internal Buffer Organization Control bit
1 = Internal buffer is organized as an indexed buffer; Channel *n* conversions are stored in Register *n*
0 = Internal buffer is organized as an n-deep FIFO buffer

bit 9-8 **ADPWR<1:0>:** A/D Power Mode Select bits (between trigger events)(1)
When ADHALT = 1:
xx = Module remains powered down in all Power Managed modes
When ADHALT = 0:
11 = Module remains active in all Power-Managed modes
10 = Reserved
01 = Module is powered down in Sleep and Idle modes (when not converting)
00 = Module is always powered down when not converting

bit 7-6 **BUFINT<1:0>:** FIFO Buffer Interrupt Control bits(2)
11 = Interrupt after buffer is full and stops (hardware clears SLEN bit)
10 = Interrupt when buffer is half full
01 = Interrupt when buffer is one-quarter full
00 = No interrupt

bit 5-2 **Unimplemented:** Read as '0'

bit 1 **REFPUMP:** A/D Reference Charge Pump Control bit(3)
1 = Reference charge pump is enabled, to minimize internal sampling error
0 = Reference charge pump is disabled (suitable for all reference voltage levels)

bit 0 **ADHALT:** A/D Suspend bit(1)
1 = Ignores triggers and clocks in all modes; however, conversions still in progress will complete
0 = Operates as normal (continues on from suspend point)

Note 1: ADHALT and/or ADPWR<1:0> are not implemented on all devices; in these cases, the definitions of these bits may differ. Refer to the specific device data sheet for details.
2: BUFINT<1:0> is forced to '00' when the buffer is operating in Indexed mode (BUFORG = 1).
3: Use only when the magnitude of the A/D reference is less than (0.65 * AVDD).

Register 65-3: ADCON3: A/D Control Register 3

R/W-0	U-0	U-0	U-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
ADRC(1)	—	—	—	SLEN3	SLEN2	SLEN1	SLEN0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7(2)	ADCS6(2)	ADCS5(2)	ADCS4(2)	ADCS3(2)	ADCS2(2)	ADCS1(2)	ADCS0(2)
bit 7							bit 0

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

bit 15 **ADRC:** A/D Conversion Clock Source (TSRC) bit(1)
1 = Conversion clock is derived from FRC (TSRC = TFRC)
0 = Conversion clock is derived from the system clock (TSRC = TSYS)

bit 14-12 **Unimplemented:** Read as '0'

bit 11 **SLEN3:** A/D Sample List 3 Enable bit
1 = Sampling for this list is enabled; triggers defined by ADL3CONL<12:8> are processed
0 = Sampling for this list is disabled

bit 10 **SLEN2:** A/D Sample List 2 Enable bit
1 = Sampling for this list is enabled; triggers defined by ADL2CONL<12:8> are processed
0 = Sampling for this list is disabled

bit 9 **SLEN1:** A/D Sample List 1 Enable bit
1 = Sampling for this list is enabled; triggers defined by ADL1CONL<12:8> are processed
0 = Sampling for this list is disabled

bit 8 **SLEN0:** A/D Sample List 0 Enable bit
1 = Sampling for this list is enabled; triggers defined by ADL0CONL<12:8> are processed
0 = Sampling for this list is disabled

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Prescaler bits(2)
TAD = TSRC * (2 * ADCS<7:0>)

```
11111111
11111110
...        = Reserved
00100010
00100001

00100000 = 32 * TSRC
00011111 = 31 * TSRC
...
00000010 = 4 * TSRC
00000001 = 2 * TSRC
00000000 = TSRC
```

Note 1: This bit must be set for Sleep operation.
2: Final conversion clock frequency must be between 1 MHz and 10 MHz.

Register 65-4: ADSTATH: A/D Status Register High

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

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	PUMPST	ADREADY	ADBUSY
bit 7							bit 0

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

bit 15-3 **Unimplemented:** Read as '0'

bit 2 **PUMPST:** A/D Boost Pump Status bit
1 = The A/D boost pump is active
0 = The A/D boost pump is Idle

bit 1 **ADREADY:** A/D Analog Ready bit
1 = The analog portion of the A/D is internally calibrated and ready
0 = The analog portion of the A/D is not ready

bit 0 **ADBUSY:** A/D Busy bit
1 = A/D conversion is in progress
0 = A/D is Idle

Register 65-5: ADSTATL: A/D Status Register Low

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SLOV
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	BUFIF	ACCIF	SL3IF(1)	SL2IF(1)	SL1IF(1)	SL0IF(1)
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **SLOV:** A/D Sample List Error Event bit
1 = A buffer overflow has occurred and data has been lost
0 = No buffer overflow has occurred

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **BUFIF:** Internal Buffer Interrupt Event bit
1 = Buffer has been filled to the notification level defined by BUFINT<1:0>
0 = Internal buffer has not been filled

bit 4 **ACCIF:** Accumulator Counter Interrupt Event bit
1 = Accumulator counter has counted down to zero
0 = Accumulator counter has not reached zero

bit 3 **SL3IF:** A/D Sample List 3 Interrupt Event bit(1)
1 = An interrupt event (defined by ADL3CONH<14:13>) has occurred in Sample List 3
0 = An interrupt event has not occurred

bit 2 **SL2IF:** A/D Sample List 2 Interrupt Event bit(1)
1 = An interrupt event (defined by ADL2CONH<14:13>) has occurred in Sample List 2
0 = An interrupt event has not occurred

bit 1 **SL1IF:** A/D Sample List 1 Interrupt Event bit(1)
1 = An interrupt event (defined by ADL1CONH<14:13>) has occurred in Sample List 1
0 = An interrupt event has not occurred

bit 0 **SL0IF:** A/D Sample List 0 Interrupt Event bit(1)
1 = An interrupt event (defined by ADL0CONH<14:13>) has occurred in Sample List 0
0 = An interrupt event has not occurred

Note 1: These bits mirror the ADLIF flag bits for the corresponding ADLnSTAT registers. Changes in the ADLIF bit are simultaneously reflected in the SLxIF bits.

Register 65-6: ADLnCONH: A/D Sample List n Control Register High

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ASEN	SLINT1	SLINT0	WM1	WM0	CM2	CM1	CM0
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
CTMEN	PINTRIS	MULCHEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 7							bit 0

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

bit 15 **ASEN:** A/D Auto-Scan Enable bit
1 = Auto-Scan: Sample-and-Convert all associated inputs sequentially on every trigger event
0 = Sequential Scan: Sample-and-Convert the next associated input on trigger event

bit 14-13 **SLINT<1:0>:** Interrupt Trigger Control bits
When ASEN = 1:
11 = Interrupt after auto-scan completion only if a match occurred
10 = Interrupt after every match
01 = Interrupt after auto-scan completion
00 = No Interrupt
When ASEN = 0:
11 = Reserved
10 = Interrupt after all records in a Sample List have been converted (SLSIZE<5:0> samples)
01 = Interrupt after every sample
00 = No interrupt

bit 12-11 **WM<1:0>:** Internal Buffer Write Mode bits
11 = Reserved
10 = No conversion results saved (typically for threshold compare only)
01 = Conversion results saved when a match occurs (typically for threshold compare only)
00 = All conversion results saved to the ADRESn register associated with the conversion

bit 10-8 **CM<2:0>:** Threshold Compare Match bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = Outside Window Match: A/D Result < Low Threshold Value or A/D Result > Threshold High Value
011 = Inside Window Match: Low Threshold Value < A/D Result < Threshold High Value
010 = Greater Than Match: A/D Result > Threshold Value
001 = Less Than Match: A/D Result < Threshold Value
000 = Matching is disabled

bit 7 **CTMEN:** A/D CTMU Current Source Enable bit
1 = CTMU is enabled during sampling and used as a current source driving the selected analog input
0 = CTMU is not used as a current source driving selected analog input

bit 6 **PINTRIS:** Force Channel TRIS bit
1 = I/O pin associated with channel is in High-Impedance mode during sampling
0 = I/O pin associated with channel is driven during sampling

bit 5 **MULCHEN:** Multiple Channel Enable bit
1 = Channels selected by the ADLnMSEL registers are connected in parallel and sampled together; Sample List entries are ignored
0 = Channels selected by the ADLnMSEL registers are ignored; only one channel is selected at a time based on the Sample List

Register 65-6: ADLnCONH: A/D Sample List n Control Register High (Continued)

bit 4-0 **SAMC<4:0>:** Sample-and-Hold Capacitor Charge Time (Acquisition Time) bits

11111 = 31 TAD
11110 = 30 TAD

. . .

00001 = 1 TAD
00000 = 0.5 TAD

Register 65-7: ADLnCONL: A/D Sample List n Control Register Low

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SLEN	SAMP(1)	SLENCLR	SLTSRC4(2)	SLTSRC3(2)	SLTSRC2(2)	SLTSRC1(2)	SLTSRC0(2)
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
THSRC	—	SLSIZE5	SLSIZE4	SLSIZE3	SLSIZE2	SLSIZE1	SLSIZE0
bit 7							bit 0

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

bit 15 **SLEN:** A/D Trigger Control Enable bit
1 = Enabled: Selected trigger causes sampling of associated analog inputs
0 = Disabled: Selected trigger does not cause sampling of associated analog inputs

bit 14 **SAMP:** A/D Manual Conversion Trigger bit(1)
1 = Starts sampling selected channel
0 = Generates manual trigger event(s) as determined by SLTSRC<4:0>

bit 13 **SLENCLR:** A/D Trigger Clear bit
1 = SLEN is cleared by hardware after a trigger is generated by this Sample List
0 = SLEN is only cleared by software

bit 12-8 **SLTSRC<4:0>:** Trigger Source Select bits(2)
11111
.... = Unimplemented, do not use
10001
10000 = Timer1 A/D trigger
01111 = Comparator 3
01110 = Comparator 2
01101 = Comparator 1
01100 = Input Capture 4
01011 = Input Capture 1
01010 = Output Compare 3
01001 = Output Compare 2
01000 = Output Compare 1
00111 = Internal periodic trigger event; interval defined by the ADTMRPR register
00110 = CTMU
00101 = Timer2
00100 = Timer1 Sync
00011 = INT0
00010 = Manual Trigger Event: Triggers are generated on every ADC clock when SAMP = 0
00001 = Manual Trigger Event: Triggers are generated on every ADC clock when SAMP = 0 and ACCONH<7> = 1
00000 = Manual Trigger Event: A single trigger is generated when SAMP is cleared

bit 7 **THSRC:** Threshold List Select bit
1 = Source used for threshold compare is the Sample List Threshold register
0 = Source used for threshold compare is the Buffer Register

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

bit 5-0 **SLSIZE<5:0>:** Sample List Size bits
Specifies the number of ADTBLn registers in the Sample List (Sample List = SLSIZE<5:0> + 1).

Note 1: Applicable only with Manual Trigger modes (SLTSRC<4:0> = 00010, 00001 or 00000).
2: These assignments are typical for most devices. Refer to the specific device data sheet for specific bit mapping.

Register 65-8: ADLnSTAT: A/D Sample List n Status Register

R-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ADTACT	LBUSY	—	—	—	—	—	—
bit 15							bit 8

R-0	U-0	R/W-0, HS	U-0	U-0	U-0	U-0	U-0
ADTDLY	—	ADLIF(1)	—	—	—	—	—
bit 7							bit 0

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

bit 15 **ADTACT:** A/D Trigger Event Active bit
1 = A trigger event is asserted
0 = A trigger event is not asserted

bit 14 **LBUSY:** A/D Trigger Control Busy bit
1 = The A/D is converting a sample entry associated with this list's trigger
0 = The A/D is not busy with this trigger

bit 13-8 **Unimplemented:** Read as '0'

bit 7 **ADTDLY:** A/D Trigger Delayed Flag bit
1 = This trigger was delayed by a higher priority trigger
0 = This trigger was not delayed by a higher priority trigger

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

bit 5 **ADLIF:** A/D Sample List Interrupt Event Flag bit(1)
1 = An interrupt event (defined by ADLnCONH<14:13>) has occurred in Sample List n
0 = No interrupt event has occurred

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

Note 1: ADLIF is mirrored by the corresponding SLxIF flag bit in the ADSTATL register. Setting or clearing this bit simultaneously changes the SLxIF.

Register 65-9: ADLnPTR: A/D Sample List n Pointer Register

U-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0
—	ADNEXT6	ADNEXT5	ADNEXT4	ADNEXT3	ADNEXT2	ADNEXT1	ADNEXT0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14-8 **ADNEXT<6:0>:** A/D Pointer to Next Entry on Sample List to be Converted bits
This value is added to the start of the Sample List to determine the ADTBLn register to be used for the next trigger event.

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

Register 65-10: ADTBLn: A/D Sample Table Entry n Register

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
UCTMU	DIFF	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
bit 7							bit 0

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

bit 15 **UCTMU:** Enable CTMU During Entry Conversion bit
1 = CTMU is enabled during channel conversion for this entry
0 = CTMU is disabled during channel conversion for this entry

bit 14 **DIFF:** Differential Inputs Select bit
1 = Analog inputs are sampled as differential pairs for this entry
0 = Analog inputs are sampled as single-ended for this entry

bit 13-7 **Unimplemented:** Read as '0'

bit 6-0 **ADCH<6:0>:** A/D Channel Entry Select bits

Each seven-bit combination represents a unique combination of single-ended analog channel and voltage references or unique differential input pair.

Channel entry combinations are device-specific, depending on the total number of analog channels available. Refer to the specific device data sheet for details.

Register 65-11: ACCONH: Accumulator Control Register High

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

R/W-0, HC	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
ACEN(1)	ACIE	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ACEN:** Accumulator Enable bit(1)
- 1 = Accumulation enabled; Sample-and-Convert the current Sample List entry on trigger event and add to the contents of ACRES
- 0 = The accumulation process has not started or is complete (cleared in hardware when accumulation count is reached)

bit 6 **ACIE:** Accumulator Interrupt Enable bit
- 1 = An interrupt event is generated when the accumulator decrements to zero
- 0 = Accumulator interrupt events are disabled

bit 5-0 **Unimplemented:** Read as '0'

Note 1: To avoid unexpected or erroneous results, do not write to ACCONH or ACCONL while ACEN is set.

Register 65-12: ACCONL: Accumulator Control Register Low

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TBLSEL5	TBLSEL4	TBLSEL3	TBLSEL2	TBLSEL1	TBLSEL0
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
COUNT7	COUNT6	COUNT5	COUNT4	COUNT3	COUNT2	COUNT1	COUNT0
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **TBLSEL<5:0>:** Record to be Accumulated Pointer bits
Indicates the location of the next value to be accumulated.

bit 7-0 **COUNT<7:0>**: Accumulations To Be Completed Counter bits
Decrements on each operation.

Register 65-13: ADCHITH: A/D Match Hit Register High

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
CHH31	CHH30	CHH29	CHH28	CHH27	CHH26	CHH25	CHH24
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
CHH23	CHH22	CHH21	CHH20	CHH19	CHH18	CHH17	CHH16
bit 7							bit 0

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

bit 15-0 **CHH<31:16>:** A/D Conversion Match Hit bits

1 = A threshhold compare match has occurred on the corrsponding Sample List entry

0 = No match has occurred

Register 65-14: ADCHITL: A/D Match Hit Register Low

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0
bit 7							bit 0

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

bit 15-0 **CHH<15:0>:** A/D Conversion Match Hit bits

1 = A threshhold compare match has occurred on the corrsponding Sample List entry

0 = No match has occurred

Register 65-15: ADTHnH: A/D Sample Table n Threshold Value Register High

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8
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
TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0
bit 7							bit 0

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

bit 15-0 **TH<15:0>:** High Threshold Value for Windowed Compare Operations, Sample Table n bits
Value in 12-bit unsigned integer format only.

Register 65-16: ADTHnL: A/D Sample Table n Threshold Value Register Low

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8
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
TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0
bit 7							bit 0

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

bit 15-0 **TH<15:0>:** Low Threshold Value for Windowed Compare Operations, Sample Table n bits
Also serves as the comparison value for non-windowed threshold compare operations. Value in 12-bit unsigned integer format only.

Register 65-17: ADLnMSEL3: A/D Sample List n Multichannel Select Register 3

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HSC	R/W-0, HSC
—	—	—	—	—	—	MSEL49	MSEL48
bit 7							bit 0

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

bit 15-2 **Unimplemented:** Read as '0'

bit 1-0 **MSEL<49:48>:** A/D Channel Select bits

1 = Corresponding channel participates in multi-channel operations for Sample List n

0 = Channel does not participate in multi-channel operations

Register 65-18: ADLnMSEL2: A/D Sample List n Multichannel Select Register 2

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
MSEL47	MSEL46	MSEL45	MSEL44	MSEL43	MSEL42	MSEL41	MSEL40
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
MSEL39	MSEL38	MSEL37	MSEL36	MSEL35	MSEL34	MSEL33	MSEL32
bit 7							bit 0

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

bit 15-0 **MSEL<47:32>:** A/D Channel Select bits

1 = Corresponding channel participates in multi-channel operations for Sample List n

0 = Channel does not participate in multi-channel operations

Register 65-19: ADLnMSEL1: A/D Sample List n Multichannel Select Register 1

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
MSEL31	MSEL30	MSEL29	MSEL28	MSEL27	MSEL26	MSEL25	MSEL24
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
MSEL23	MSEL22	MSEL21	MSEL20	MSEL19	MSEL18	MSEL17	MSEL16
bit 7							bit 0

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

bit 15-0 **MSEL<31:16>:** A/D Channel Select bits
1 = Corresponding channel participates in multi-channel operations for Sample List n
0 = Channel does not participate in multi-channel operations

Register 65-20: ADLnMSEL0: A/D Sample List n Multichannel Select Register 0

R/W-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0	U-0
MSEL15	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **MSEL15:** A/D Channel Select bits
1 = Corresponding channel participates in multi-channel operations for Sample List n
0 = Channel does not participate in multi-channel operations

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

65.4 OPERATIONAL OVERVIEW

65.4.1 SAR vs. Pipeline Converters

The main difference between Pipeline A/D Converters and Successive Approximation Register (SAR) A/D Converters is speed. In a typical SAR-based conversion, the digital conversion portion of the result generation process is done in a serial fashion, and normally, only one analog comparator is used to convert one result bit worth of data per A/D clock cycle. A Pipeline Converter takes advantage of parallelism and uses multiple internal analog comparators to allow conversions of multiple results, at several different stages of a conversion pipeline. The parallelism of the Pipeline Converter enables it to perform conversions in a "pipelined" fashion (i.e., each conversion is tightly staged, back-to-back. The net result is the converter can generate up to one conversion result for every single A/D clock period (TAD).

The primary disadvantage of the Pipeline A/D Converter is an increased level of noise, as compared to SAR Converters. In practice, taking and averaging multiple samples is used to achieve the desired results.

An example throughput and latency comparison of this Pipeline A/D Converter versus a typical 12-bit SAR A/D Converter is shown in Table 65-1.

Table 65-1: Comparison of A/D Converter Performance Metrics

Metric	12-Bit SAR (e.g., PIC24FJ128GA310)	12-Bit Pipeline (e.g., PIC24FJ128GC010)
A/D Clock Frequency (MHz)	4 (typical)	8 (typical)
Sample-and-Hold Charge Time (TAD)	4 (typical)	0.5 (typical)
Typical Conversion Latency (TAD)	14	8.5
Throughput (results/TAD)	Approx. 0.06	1
Maximum Throughput (samples/s)	Up to 200k	Up to 10M
Internal A/D Noise (qualitative)	Very low	Moderate

65.4.2 Sampling Requirements

The analog input models of the Pipeline A/D Converter, in both single-ended and differential configurations, are shown in Figure 65-3. The total sampling time for the A/D is a function of the holding capacitor charge time, as well as input and sampling switch resistance.

During the sampling operation, the sampling switch is closed and the internal Sample-and-Hold capacitor is connected to the analog input pin, through the switch and interconnect resistances. The sampling switch remains closed for the duration that is selected by the SAMC<4:0> bits field (ADLnCONH<4:0>). While the sampling switch is closed, the Sample-and-Hold capacitor, CHOLD, begins charging to the voltage applied to the analog pin from the analog voltage source, VA.

The application should select a value for the SAMC<4:0> bits to allow a sampling duration that is sufficient to fully charge the capacitor to the same voltage as VA. The charging profile can be modelled by a second-order RC circuit, as shown in Figure 65-3. The required time is a function of the source resistance and is shown in the Table 65-2 below.

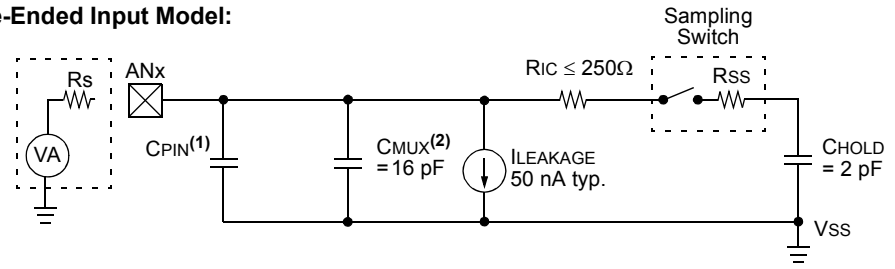
Table 65-2: Minimum Required Sampling Time

RS	Minimum Sample Time
10	5 ns
50	10 ns
100	15 ns
500	75 ns
1K	150 ns
10K	1.5 μs
100K	15 μs

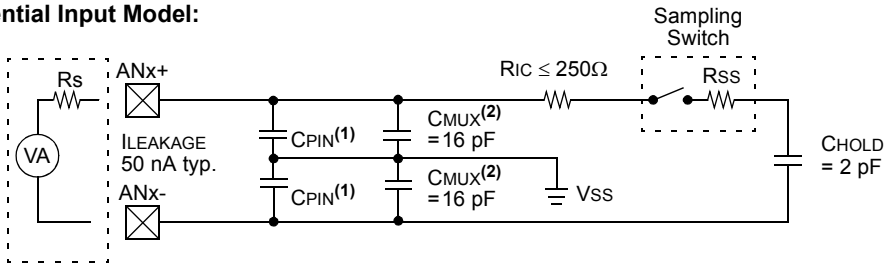
Once the sampling interval has elapsed, the sampling switch opens and the analog pipeline consumes the charge stored on the capacitor. The Pipeline A/D Converter design only requires 1/2 TAD before the capacitor is discharged and its charge is fully consumed (i.e., effectively discharged to AVSS when performing single-ended measurements or discharged between the positive and negative input channel pins when performing differential measurements). When the discharge interval has elapsed, the A/D Converter is ready to close the sampling switch again, to begin sampling the next configured channel.

Figure 65-3: 12-Bit A/D Converter Analog Input Models

Single-Ended Input Model:



Differential Input Model:



Legend:	CPIN	= Input Capacitance
	CMUX	= Multiplexor Capacitance (AN14 and higher only)
	ILEAKAGE	= Leakage Current at the pin due to various junctions
	RIC	= Interconnect Resistance
	RSS	= Sampling Switch Resistance
	CHOLD	= Sample/Hold Capacitance (from DAC)

Note 1: The CPIN value depends on the device package and is not tested. The effect of CPIN is negligible if Rs ≤ 1.5 kΩ.

2: Applicable only for analog inputs, AN14 and higher.

65.4.2.1 ANALOG SWITCH CHARGE PUMP

The 12-Bit, High-Speed Pipeline A/D Converter incorporates a built-in charge pump for use with the CMOS pass gates on the analog input channels. Similar to discrete component versions, capacitive charge pumping is employed inside the microcontroller to generate MOSFET gate voltages that are higher than AVDD. By boosting gate voltage, the worst-case effective series resistance of the pass gates can be drastically reduced. This reduces both distortion and the necessary sampling time that would otherwise be required at low analog supply voltage levels.

The analog switch charge pump is controlled by the PUMPEN bit (ADCON1<7>). It is recommended to enable the charge pump whenever AVDD is less than 2.5V. This provides the lowest analog switch resistance, reducing the time required for analog channel sampling operations.

65.4.2.2 EXTENDED SAMPLING (BEYOND 31 TAD)

The SAMC<4:0> bits (ADLnCONH<4:0>) allow the user to set a wide range of sampling times, from 0.5 TAD (default) to 31 TAD. As previously noted, the sampling time needed to charge the Sample-and-Hold capacitor depends on the characteristic impedance of the analog source being measured, as well as internal analog path capacitance. In most cases, an appropriate value can be selected with the SAMCx bits field. However, certain scenarios may require an interval longer than the maximum of 31 TAD. These include sampling exceptionally high-impedance analog sources or performing certain types of CTMU (Charge Time Measurement Unit) operations.

A simple way of doing this is to set up Sample List 0 with a size of one (SLSIZE<5:0> = 00h) and configure ADTBL0 to the channel requiring the extended sample. The A/D Converter automatically switches to the next channel to be sampled whenever the module is Idle (i.e., no trigger events are pending or have recently occurred for any of the Sample Lists). With a list size of one, the converter will remain connected indefinitely to the channel specified by ADTBL0. The selected channel will then remain connected to the Sample-and-Hold capacitor and continue to be sampled until a trigger event occurs. If a trigger event for Sample List 0 occurs, sampling will continue for the interval that is programmed by the SAMCx bits field, after which a conversion is performed.

For additional information on Sample Lists and their use, refer to **Section 65.9 “Sample Lists”**.

Keep in mind that, if other Sample Lists are configured and enabled, any trigger event that another Sample List may be programmed for will also end sampling for the target channel.

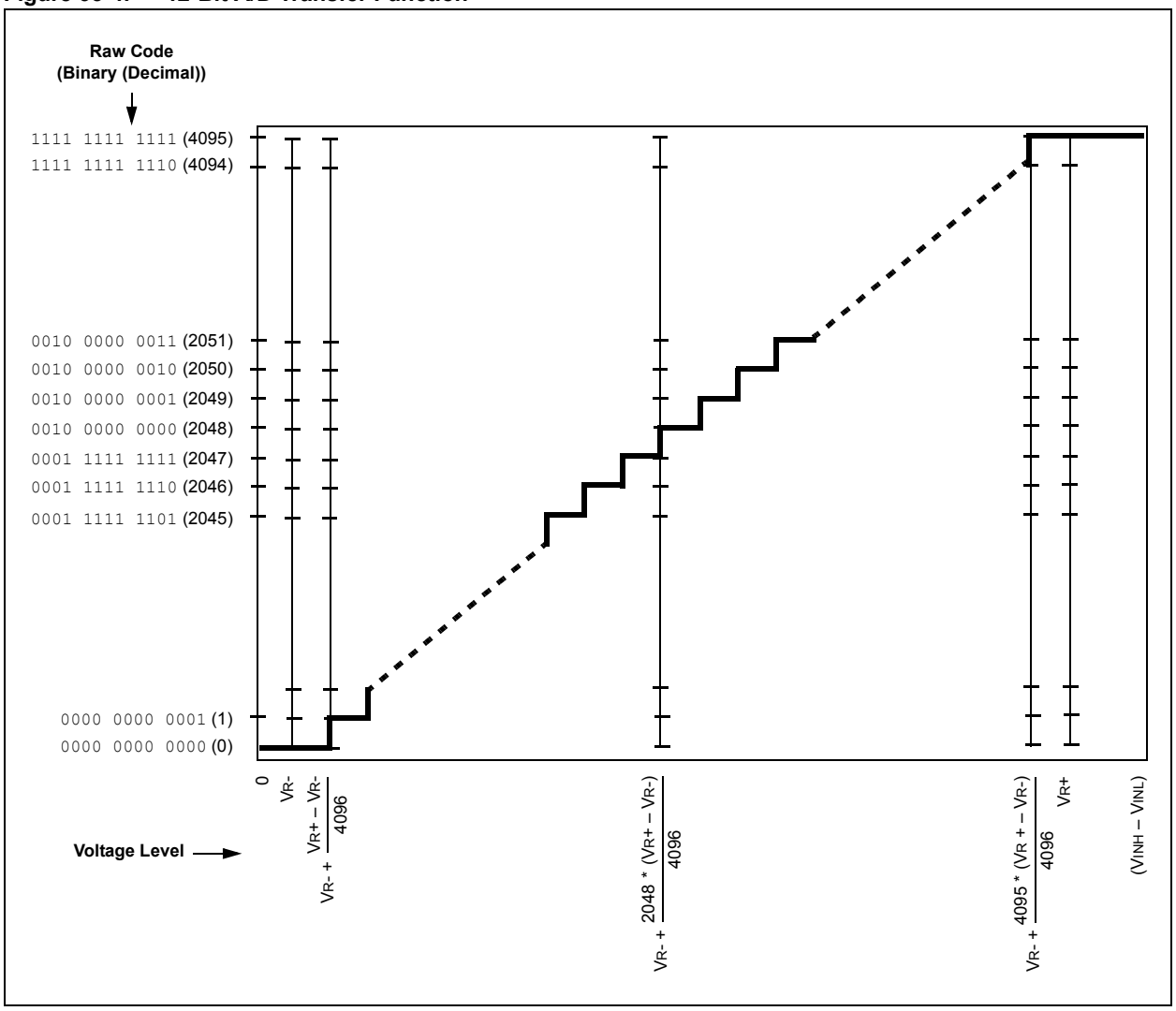
65.4.3 Transfer Function

The transfer function of the A/D Converter is shown in Figure 65-4. In both cases, the difference of the input voltages, (V_{INH} – V_{INL}), is compared to the reference, ((V_{R+}) – (V_{R-})).

For the 12-bit transfer function:

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

Figure 65-4: 12-Bit A/D Transfer Function



65.5 A/D RESULTS BUFFER

As conversions are completed, the module writes the results of the conversions into the A/D result buffer. This buffer is a RAM array of fixed word size, accessed through the SFR space. In most implementations, there are 32 registers in the results buffer (ADRES0 through ADRES31), corresponding to the number of ADTBLn registers and Sample List entries.

User software may attempt to read each A/D conversion result as it is generated; however, this might consume too much CPU time. To minimize software overhead, the module will fill the buffer with results and then generate an interrupt when the buffer is filled (FIFO mode).

Note:	This section describes buffer operation when ADLnCONH<12:11> = 00 (all conversions are written to the result buffer). Buffer operation is different when the Compare Only or Compare and Save modes are used with the Threshold Detect feature. For more information, see **Section 65.11 "Threshold Detect Operations"**.

65.5.1 Buffer Fill Modes

The results buffer can be configured to operate in either of two modes: a standard FIFO mode, compatible with earlier 12-bit A/D modules (default), or an Indexed mode based on the Sample Lists. The Fill mode is selected by the BUFORG bit (ADCON2<10>).

65.5.1.1 FIFO MODE

When BUFORG = 0, the results buffer operates in FIFO mode. The first conversion results, after initiating conversions, is written to ADRES0. Subsequent conversions are written to the next sequential buffer location, continuing until the process is interrupted. If allowed to continue without interrupts, the module will fill each location and then wrap around to ADRES0, continuing the process.

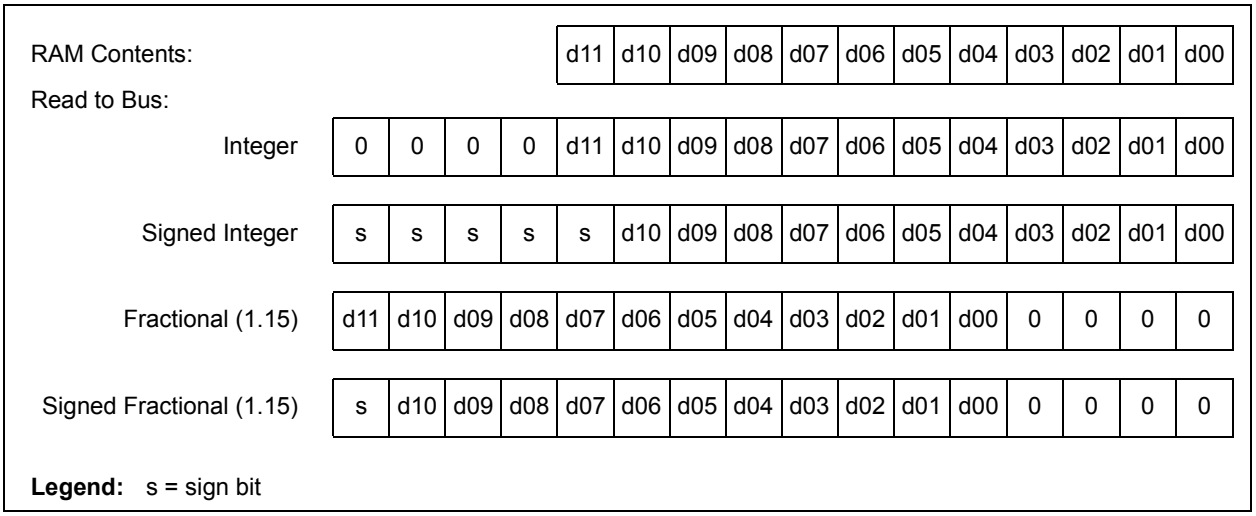
65.5.1.2 INDEXED MODE (SAMPLE LIST)

When BUFORG = 1, FIFO operation is disabled. Instead, the conversion result for each of the ADTBLn Sample List registers is written only to the ADRESn buffer location that corresponds to that register. For example, any conversions performed on the analog channel specified by ADTBL0 are stored only in ADRES0. The same holds true for ADTBL1 and ADRES1, and so on.

65.5.2 Buffer Data Formats

The results of each A/D conversion are 12 bits wide. To maintain data format compatibility, the result of each conversion is automatically converted to one of four selectable, 16-bit formats. The FORM<3:0> bits (ADCON1<11:8>) select the format. Figure 65-5 shows the data output formats that can be selected. Tables 65-3 and 65-4 show the numerical equivalents for the various conversion result codes.

Figure 65-5: A/D Output Data Formats (12-Bit)



Legend: s = sign bit

Table 65-3: Numerical Equivalents of Various Result Codes: 12-Bit Integer Formats

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

Table 65-4: Numerical Equivalents of Various Result Codes: 12-Bit Fractional Formats

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

65.6 A/D TRIGGER EVENTS

After configuring and enabling the 12-Bit, High-Speed Pipeline A/D Converter and one or more Sample Lists (described in **Section 65.9 "Sample Lists"**), the converter is ready to begin sampling and conversion operations. However, this does not actually begin until after an appropriate trigger event has been generated. A trigger event can be thought of as a momentary stimulus applied to the A/D, which may be generated manually in the application firmware (ex: by writing the SAMP bit to generate a high-to-low transition), generated internally by the ADC itself (ex: using the internal A/D trigger timer) or generated external to the A/D module (ex: due to an INT0 I/O pin event, a Timer1 interrupt, etc.).

When a trigger source external to the converter is selected, the A/D Converter is triggered upon detection of the same conditions that would normally trigger the interrupt event flag to become set for the module. For example, if using Timer1 as a trigger source, the A/D will be triggered when the Timer1 Counter and Period register match condition is detected, which also causes the TMR1IF interrupt flag to get set.

65.6.1 Internal Trigger Timer

In addition to supporting external trigger sources, the 12-Bit, High-Speed Pipeline A/D Converter has a built-in timer for generating periodic trigger events; its use is optional.

The A/D Timer Period register, ADTMRPR, is readable and writable by software, and is used to set the timer trigger frequency. The trigger timer counter is not memory mapped, and therefore, not directly readable or writable by the user. When the trigger timer count value matches the ADTMRPR value, an A/D trigger event is generated and the trigger timer count value is reset to 0000h.

The trigger timer is clocked by the same clock source as selected by the ADRC and ADCSx bits (ADCON3<15,7:0>), and increments once for every TAD. Therefore, the ADTMRPR register directly sets the number of TADs (minimum value of 1 TAD) between trigger events when using the internal ADC trigger timer.

If all Sample Lists are disabled (all SLEN bits are '0'), the trigger timer count is internally held at the reset value of 0000h. The timer count only increments when one or more SLEN bits are set.

Although only one is implemented, the trigger timer can optionally be shared by multiple Sample Lists. If multiple lists are enabled and simultaneously configured to use the timer as a trigger source, the initial trigger delay upon setting the second and subsequent SLEN bit(s) may be shorter than the ADTMRPR value would predict. This is a result of the timer counter becoming free running once the first SLEN bit is set. After the initial trigger event is generated for a Sample List, all of the subsequent trigger periods for that Sample List will match the ADTMRPR value.

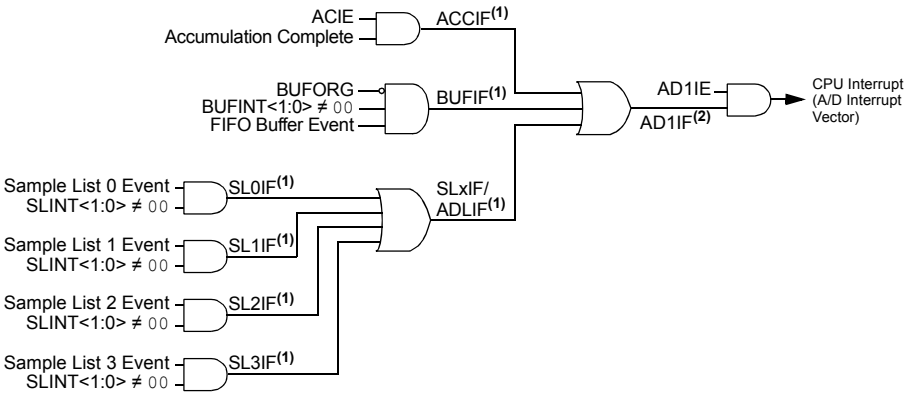
65.7 INTERRUPTS

The 12-Bit, High-Speed Pipeline A/D Converter implements its own user-configurable interrupt logic tree (Figure 65-6) to monitor three different types of module level interrupts:

- ACCIF: Accumulation Complete Interrupt
- BUFIF: FIFO mode ADRESn Buffer Level Watermark Interrupt
- ADLIF and SLxIF: Sample List n Interrupts (one each for Sample Lists 0 through 3)

These interrupts do not directly generate device level interrupts. Instead, enabling any one of these can cause the top-level AD1IF interrupt flag to be set. If the AD1IE interrupt enable bit is also set, this allows the CPU to vector to the A/D interrupt handler (assuming no higher priority interrupts are pending). If the application uses more than one of the A/D interrupt sources simultaneously, it may poll the ACCIF, BUFIF and ADLIF/SLxIF flags to determine which event caused the top level interrupt to be asserted.

Figure 65-6: Hardware Interrupt Logic for 12-Bit, High-Speed Pipeline A/D Converter



Note 1: Interrupt flag bits are set by hardware as indicated and remain set until cleared manually by the application. Clearing these bits in software does not affect the state of AD1IF or the A/D interrupt.

2: AD1IF can be cleared by software without regard to the state of the lower level interrupt flags. Once cleared, it remains cleared until a new lower level interrupt event occurs. It is NOT necessary for the SLxIF/ACCIF/BUFIF bits to be cleared for the hardware to detect a new event.

65.7.1 Accumulation Complete Interrupt (ACCIF)

The Accumulation Complete interrupt (ACCIF) is generated when all samples of an accumulation process have been successfully accumulated. The interrupt flag is set by hardware only when the accumulation completes and the Accumulate Interrupt Enable bit, ACIE (ACCONH<6>), is also set. Disabling the interrupt (ACIE = 0) means that neither ACCIF or AD1IF will be set when an accumulation completes.

The ACCIF bit can also be set or cleared by software. Manually setting or clearing ACCIF does not affect the AD1IF interrupt flag.

65.7.2 Buffer Watermark Interrupt, FIFO Mode (BUFIF)

The Buffer Level Watermark Interrupt Flag, BUFIF, is only generated and only applicable when the A/D Converter is configured to operate in the FIFO Buffer mode (BUFORG = 0). The interrupt is configured by the BUFINT<1:0> bits (ADCON2<7:6>) and can be set to occur at various levels of buffer fill. Clearing the bits disables the interrupt.

65.7.3 Sample List Interrupts (ADLIF and SLxIF)

Each Sample List has its own independent A/D Sample Interrupt Flag, ADLIF (ADLnSTAT<5>). The flag's operation is configured using the SLINT<1:0> bits (ADLnCONH<14:13>) for the corresponding list. Clearing the bits disables the interrupt for that list.

When Auto-Scan mode is used (ASEN = 1), the Sample List interrupt can be configured to be generated when the Auto-Scan is complete, when a threshold match condition occurs or only after an Auto-Scan operation with a threshold compare match condition occurs.

When Sequential mode is enabled (ASEN = 0), the Sample List interrupt can be configured to be generated either at the end of a Sample List or after completion of every single sample in the Sample List. There is no threshold compare interrupt option in Sequential mode.

For monitoring convenience, the interrupt flags in the ADLnSTAT registers are mirrored in the ADSTATL registers by the SLxIF bits (ADSTATL<3:0>). Changes to any of the ADLIF bits are simultaneously reflected by the corresponding SLxIF bits. Functionally, any ADLIF bit and its SLxIF counterpart may be treated as being identical.

65.8 INITIALIZATION

To use the 12-Bit, High-Speed Pipeline A/D Converter, the general procedure is:

1. Configure the I/O pins to be used as analog inputs by setting the corresponding bits in the ANSEL registers.
2. Set up global settings with the ADCON1, ADCON2 and ADCON3 registers:
   a) Configure the A/D clock source and rate
   b) Configure the A/D reference sources
   c) Configure the buffer storage settings, format and interrupt generation
   d) Configure other global settings, such as power-saving features
3. Set the ADON bit (ADCON1<15>) to enable the module.
4. Poll the ADREADY bit (ADSTATH<1>) until it becomes set.
5. Configure Sample List 0 settings, controlled by the ADL0CONH/L registers (but do NOT enable the Sample List at this point):
   a) Select the desired Sample List interrupt generation settings
   b) Select the Data Write mode (e.g., write all results to buffer)
   c) Configure analog sampling time (SAMC<4:0>)
   d) Select the trigger source
   e) Specify how many entries are in the Sample List (SLSIZE<5:0>)
   f) Configure other Sample List 0 specific settings
6. Initialize the ADTBLn registers to select the channels to be sampled (ADTBL0 at a minimum and subsequent registers if a longer Sample List is to be used).
7. Configure and enable A/D interrupts (if desired):
   a) Clear the ADLIF and SL0IF bits
   b) Set the interrupt priority with the ADLIP<2:0> bits
   c) Enable the ADLIE bit
8. Enable Sample List 0 by setting the SLEN bit (ADL0CONL<15>).
9. Enable the trigger source for Sample List 0 to begin sampling.
10. Wait for the appropriate A/D interrupt flag to be set (SL0IF or ADLIF) to read the results from the appropriate ADRESn registers.

A typical initialization sequence following this procedure is shown in Example 65-1 (pages 30 and 31). This procedure assumes the use of a single Sample List. To configure additional lists, repeat Steps 5 and 6, as required, before proceeding to interrupt configuration. For more information on multiple lists, see **Section 65.9.3 "Using Multiple Sample Lists"**.

When initializing and enabling the A/D Converter, the application should initialize the ADCON1, ADCON2 and ADCON3 global registers prior to setting the ADON bit. All important global settings, such as the clock source and prescaler, power level, charge pump and reference selections, should be made prior to setting ADON. This procedure ensures that the module will not be temporarily or inadvertently misconfigured during the initialization process.

Upon setting the ADON bit, the A/D Converter performs an automatic hardware self-calibration operation in the background. The self-calibration takes up to 800 TAD cycles to complete. Setting the ADON bit causes the hardware to clear the ADREADY bit (ADSTATH<1>) within one TAD, and keeps it low for the duration of the procedure. When ADREADY becomes set, the module is ready to use; the application may enable Sample List(s) and generate trigger events to start performing sampling and conversion operations. In certain devices, a second self-calibration command may be required; refer to the specific device data sheet for more information.

During this self-calibration, the microcontroller CPU is free to execute application code other than A/D-related code.

Example 65-1: A/D Conversion and Simple Sampling (Configuration and Sample List)

```
#include <p24FJ128GC010.h>
// PIC24FJ128GC010 CONFIGURATION SETTINGS
_CONFIG4(PLLDIV_DIV2 & IOL1WAY_OFF)
_CONFIG3(BOREN_OFF)
_CONFIG2(POSCMD_HS & FNOSC_PRIPLL & FCKSM_CSECME)
_CONFIG1(FWDTEN_WDT_SW & ICS_PGx2)

// The ADC conversion result.
volatile unsigned int channel_2;
volatile unsigned int channel_38;

int main()
{
// ANALOG INPUTS CONFIGURATION
// 2 analog inputs will be sampled.
TRISBbits.TRISB2 = 1;                   // AN2(RB2)
ANSBbits.ANSB2 = 1;
TRISAbits.TRISA14 = 1;                  // AN38(RA14)
ANSAbits.ANSA14 = 1;

// GLOBAL SETTINGS
ADCON1=0;
ADCON2=0;
ADCON3=0;
// Configure the A/D voltage references.
ADCON2bits.PVCFG = 0;                   // Vref+ = AVdd
ADCON2bits.NVCFG = 0;                   // Vref- = AVss
// Configure the A/D clock.
ADCON3bits.ADRC = 0;                    // Conversion clock derived from system clock.
ADCON3bits.ADCS = 15;                   // Divide system clock by 16(TAD = 1uS @ 16 MIPS).
// Configure buffer storage settings and interrupt generation.
ADCON1bits.FORM = 0;                    // Output format is unsigned integer.
ADCON2bits.BUFORG = 1;                  // Result buffer is an indexed buffer.
ADCON2bits.BUFINT = 0;                  // No buffer interrupt.
// Configure power-saving.
ADCON1bits.PWRLVL = 0;                  // Low power, reduced frequency operation.
ADCON2bits.ADPWR = 3;                   // Module is always powered on.

// Sample List SETTINGS
ADL0CONL = 0;
ADL0CONH = 0;
ADL0CONLbits.SLSIZE = 2-1;              // Sample list length: 2 channels.
// Sampling settings.
ADL0CONHbits.ASEN = 1;                  // Enable auto-scan.
ADL0CONHbits.SLINT = 1;                 // Interrupt after auto-scan completion.
ADL0CONHbits.SAMC = 15;                 // Sample time is 15 TAD.
ADL0CONLbits.SLTSRC = 0;                // Single trigger generated when SAMP is cleared.
ADL0PTR = 0;                            // Start from the first list entry.
// Threshod compare settings.
ADL0CONHbits.CM = 0;                    // Disable threshold compare.
// Channels selection.
ADTBL0bits.ADCH = 2;                    // Channel #2.
ADTBL1bits.ADCH = 38;                   // Channel #38.
```

Example 65-1: A/D Conversion and Simple Sampling (Configuration and Sample List) (Continued)

```
// ENABLE A/D
ADCON1bits.ADON = 1;                  // Enable A/D.
while(ADSTATHbits.ADREADY == 0);      // Wait for ready flag set.
    ADCON1bits.ADCAL = 1;             // Start calibration.
while(ADSTATHbits.ADREADY == 0);
    ADL0CONLbits.SAMP = 1;            // Close sample switch.
ADL0CONLbits.SLEN = 1;                // Enable sample list.

// CONVERSION
while(1)
{
    IFS0bits.AD1IF = 0;               // Start conversion.
    ADL0CONLbits.SAMP = 0;            // Wait for the result.
    while(IFS0bits.AD1IF == 0);
    ADL0CONLbits.SAMP = 1;            // Close the sample switch.
        channel_2 = ADRES0;           // Read result for the channel #2.
        channel_38 = ADRES1;          // Read result for the channel #38.
}

}
```

65.8.1 Configuring Port Pins as Analog Inputs

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

For external analog inputs, both the ANSn register and the corresponding TRIS register bits control the operation of the A/D port pins. The port pins that will function as analog inputs must also have their corresponding TRIS bits set, specifying the pins as inputs. After a device Reset, all TRIS bits are set. If the I/O pin, associated with an A/D channel, is configured as a digital output (TRIS bit is cleared) while the pin is configured for Analog mode, the port digital output level (V_{OH} or V_{OL}) will be converted.

Note 1:	When reading a PORT register, any pin configured as an analog input reads as '0'.
2:	Analog levels on any pin that is defined as a digital input may cause the input buffer to consume current that is out of the device's specification.

65.8.2 Changing Setting with the Module Enabled

The A/D clock source and frequency, PWRLVL, and REFPUMP settings should not be modified while the A/D Converter is enabled. In order to change these settings, it is recommended to first disable the module by clearing ADON, reconfigure the module and then re-enable the module (and wait for the hardware self-calibration to complete by checking the ADREADY bit).

Although other settings may be changed while the module is active, the application should exercise special care when resizing Sample Lists or changing trigger source settings for a Sample List. Resizing a lower numbered Sample List (e.g., changing SLSIZE<5:0> for Sample List 0) will alter the locations of the ADTBLn array boundaries associated with the other Sample Lists. This may cause unexpected operation if the higher numbered list is enabled and active when the lower numbered list is resized. It is generally recommended to disable all higher numbered Sample Lists by clearing their respective SLEN bit before adjusting the size of a lower numbered Sample List.

Similarly, it is generally not recommended to change specific Sample List settings while the list is enabled (SLEN = 1). Instead, it is recommended to initialize all Sample List-specific settings (i.e., ADLnCONH and ADLnCONL values) prior to enabling the Sample List. If it becomes necessary to change settings after it has already been initialized, it is suggested to temporarily disable the Sample List by clearing the SLEN bit prior to making the change. Note that clearing a SLEN bit will not necessarily halt any ADC operations that may have already been in progress if the Sample List had already been triggered prior to clearing SLEN.

When the SLEN bit is cleared, the A/D Converter will ignore any new trigger events associated with that Sample List. However, trigger events that have already occurred may result in the current A/D operation continuing to completion. After clearing the SLEN bit, it is advisable to check the LBUSY bit (ADLnSTAT<14>) to verify that the module has ceased operations for the Sample List, before changing any of the list's settings.

65.9 SAMPLE LISTS

In many application designs, it is desirable to set up the A/D Converter to Sample-and-Convert many input channels automatically, and without CPU intervention for each conversion. In earlier PIC24F devices, the auto-scan feature allowed for multiple channels to be sampled and scanned in a fixed-channel number sequence. Unless the module was re-initialized, the same Sample-and-Convert parameters were used for each channel in the sequence, and each repetition of the sequence.

The 12-Bit, High-Speed Pipeline A/D Converter implements the same capability through the use of Sample Lists, and at the same time, provides for a much wider range of scanning and sampling options. A Sample List can be thought of as a list of instructions for the module, which tells the hardware what to do when performing automatic Sample-and-Convert operations. Among other things, it contains information, such as which analog channels to sample, what order to sample them in, which trigger source is to be used to initiate sampling, etc. It can also dictate if a group of channels is to be sampled once or repeatedly as a continuous loop.

The Pipeline A/D Converter supports up to four Sample Lists, with each list being independently configurable for a wide range of channel selection and sampling options. The settings for each list are configured by the applicable ADLnCONH and ADLnCONL registers (Register 65-6 and Register 65-7) for that list. This allows each Sample List to be independently configured with its own Sample-and-Convert settings, as well as Threshold Detect and interrupt generation settings. In addition, each Sample List is monitored by its own Status register (ADLnSTAT, Register 65-8), which shows the current state for trigger events and interrupt status.

An enabled Sample List may contain anywhere from one, to up to 64 entries (device-specific). Each entry is one ADTBLn register, which instructs the A/D Converter to perform a conversion on a specific analog input. In total, there are up to 64 ADTBLn registers implemented (ADTBL0 through ADTBL63). Refer to the specific device data sheet for the specific maximum Sample List size.

The Sample List registers can be partitioned and allocated to the various enabled Sample Lists. For example, Sample List 0 could be configured to contain 12 entries (using ADTBL0 to ADTBL11), while Sample List 1 could be configured to contain up to 52 entries. In this scenario, Sample Lists 2 and 3 would not be able to use any ADTBLn registers, and therefore, would need to remain disabled.

65.9.1 Setting Up Sample Lists

Unlike the auto-scan feature in previous devices, the use of Sample Lists is required when using the Pipeline A/D Converter. Even when performing single Sample-and-Convert operations on a single input channel, a Sample List must be set up. Typically, Sample List 0 is used.

As mentioned, the ADLnCONH and ADLnCONL registers configure the common parameters used for all sample entries in the Sample List. These include:

- Sampling Interval (acquisition time)
- Auto-Scan mode
- Trigger Event Source
- Manual Trigger Operation
- Interrupt Generation (ADLIF/SLxIF options)
- Threshold Detect and Compare Operation, and Associated Buffer Write mode
- CTMU Current Source Trigger
- I/O Pin State During Sampling
- Simultaneous Multi-Channel Sample-and-Convert

Sample entries are stored in the 64 ADTBLn registers (Register 65-10), with each sample entry stored in one register. The register encodes the analog channel (or channel pair in Differential mode) to sample, the voltage references to be used and a CTMU current source enable bit.

The size of the Sample List is determined by the SLSIZE<5:0> bits (ADLnCONL<5:0>). The actual size of the list is (SLSIZE<5:0> + 1). This means that a Sample List may contain as little as one entry (SLSIZE<5:0> = 0), or as many as 64 entries (SLSIZE<5:0> = 63).

The Sample List is enabled, and begins to Sample-and-Convert channels on trigger events, when the SLEN bit (ADLnCONL<15>) is set. It is recommended to keep SLEN = 0 until the appropriate ADTBLn register has been properly initialized for the channels to be sampled and all other settings in the ADLnCONH/ADLnCONL registers have been configured. This ensures predictable converter behavior by eliminating spurious conversions caused by trigger events received while the list is still being configured.

65.9.2 Auto-Scan Operation

In the Pipeline A/D Converter, the auto-scan functionality is based on the Sample List. These lists replace the use of Channel Scan Select registers in previous PIC24F devices. The ASEN bit (ADLnCONH<15>) determines how the Sample List is processed.

When ASEN = 1, the Sample List is in Auto-Scan mode. When a trigger event is received, the Sample List begins processing its entries, starting at the lowest ADTBLn register of the list, and automatically Samples-and-Converts each sample entry in sequence. When the list is done, the Sample List stops and waits for the next trigger event. The process is repeated for each trigger event.

When ASEN = 0, the Sample List is in Sequential mode. When a trigger event is received, the Sample List starts at the lowest ADTBLn register of the list and processes that entry. When the Sample-and-Convert process is done, it stops and waits for the next trigger event, at which time, the next sample entry is processed. The operation repeats for each trigger until the entire list of entries is processed.

65.9.3 Using Multiple Sample Lists

Sample lists may be used sequentially or any combination of lists may be enabled simultaneously. Hardware prioritization is used to arbitrate any sampling conflicts between different lists.

All of the Sample Lists share the same array of ADTBLn registers to define individual sample entries. The assignments of ADTBLn registers to individual Sample Lists are essentially dynamic, based on the list size specified by the SLSIZE<5:0> bits (ADLnCONL<5:0>). The actual size of the list is (SLSIZE<5:0> + 1). The starting point for each Sample List is the next available register from the end point of the previous list and the end point is always (SLSIZE<5:0> + 1) registers from the list's starting point. Put another way, the address of a Sample List's end point is always (SLSIZE<5:0> + n), where 'n' is the starting point's address. Sample List 0 is always the first Sample List for enumeration purposes and ADTBL0 is always its starting point.

There is no provision to independently define the starting points of Sample Lists, so the end of one list will always be directly adjacent to the beginning of the next.

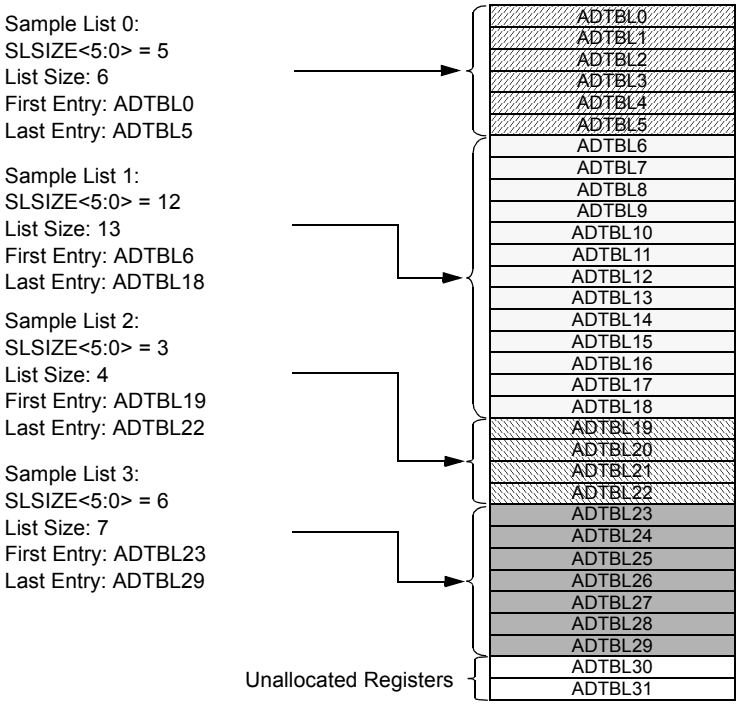
Figure 65-7 shows the relationship between the four Sample Lists that are typical for the Pipeline A/D Converter and how Sample registers are allocated to the Sample Lists. In this example, the device implements a Sample List of 32 total samples (registers). Sample List 0 starts at ADTBL0, as always; the value for SLSIZE<5:0> is 05h, meaning that Sample List 0 is six items long, extending to ADTBL5. Sample List 1 starts from the next available register, which is ADTBL6; its list size is 13 (SLSIZE<5:0> of 12, plus 1), so its end point is ADTBL18 (staring address of 6, plus SLSIZE<5:0> of 12). The other Sample Lists are allocated in a similar manner.

It is not necessary to use all of the ADTBLn registers. Any extra registers beyond the end of the enabled Sample Lists may be left unconfigured.

The module hardware does not enforce any checks to prevent more than the implemented number of sample items from being allocated; it is entirely possible to program the SLSIZE<5:0> bits field so that the total value for all enabled lists is greater than the number of available ADTBLn registers. When developing applications for the Pipeline A/D Converter, it is the user's responsibility to make certain that Sample Lists are defined correctly and that sample items are allocated to the proper Sample Lists.

The code example shown in Example 65-2 (pages 34 through 36) demonstrates the use of multiple Sample Lists.

Figure 65-7: Example of Allocating ADTBLn Registers to Multiple Sample Lists



Example 65-2: Multiple Sample List Initialization (Input Configuration)

```
// PIC24FJ128GC010 CONFIGURATION SETTINGS
_CONFIG4(PLLDIV_DIV2 & IOL1WAY_OFF)
_CONFIG3(BOREN_OFF)
_CONFIG2(POSCMD_HS & FNOSC_PRIPLL & FCKSM_CSECME)
_CONFIG1(FWDTEN_WDT_SW & ICS_PGx2)

volatile unsigned int channel_0;
volatile unsigned int channel_1;
volatile unsigned int channel_2;
volatile unsigned int channel_3;
volatile unsigned int channel_4;
volatile unsigned int channel_5;

int main()
{
// ANALOG INPUTS CONFIGURATION
// 2 analog inputs for list #0.
TRISBbits.TRISB0 = 1;                // AN0(RB0)
ANSBbits.ANSB0 = 1;
TRISBbits.TRISB1 = 1;                // AN1(RB1)
ANSBbits.ANSB1 = 1;
// 3 analog inputs for list #1.
TRISBbits.TRISB2 = 1;                // AN2(RB2)
ANSBbits.ANSB2 = 1;
TRISBbits.TRISB3 = 1;                // AN3(RB3)
ANSBbits.ANSB3 = 1;
TRISBbits.TRISB4 = 1;                // AN4(RB4)
ANSBbits.ANSB4 = 1;
// 1 analog input for list #2.
TRISBbits.TRISB5 = 1;                // AN5(RB5)
ANSBbits.ANSB5 = 1;
```

Example 65-2: Multiple Sample List Initialization (Global and Sample List Configuration) (Continued)

```
// GLOBAL SETTINGS
ADCON1=0;
ADCON2=0;
ADCON3=0;
// Configure the A/D voltage references.
ADCON2bits.PVCFG = 0;                   // Vref+ = AVdd
ADCON2bits.NVCFG = 0;                   // Vref- = AVss
// Configure the A/D clock.
ADCON3bits.ADRC = 0;                    // Conversion clock derived from system clock.
ADCON3bits.ADCS = 15;                   // Divide system clock by 16(TAD = 1uS @ 16 MIPS).
// Configure buffer storage settings and interrupt generation.
ADCON1bits.FORM = 0;                    // Output format is raw integer.
ADCON2bits.BUFORG = 1;                  // Result buffer is indexed registers.
ADCON2bits.BUFINT = 0;                  // No buffer interrupt.
// Configure power-saving.
ADCON1bits.PWRLVL = 0;                  // Low power, reduced frequency operation.
ADCON2bits.ADPWR = 3;                   // Module is always powered.

// SAMPLE LIST #0 SETTINGS
ADL0CONL = 0;
ADL0CONH = 0;
// Sample list length.
ADL0CONLbits.SLSIZE = 2-1;              // 2 channels are in the list.
// Sampling settings.
ADL0CONHbits.ASEN = 1;                  // Enable auto-scan.
ADL0CONHbits.SLINT = 1;                 // Interrupt after auto-scan completion.
ADL0CONHbits.SAMC = 15;                 // Sample time is 15 TAD.
ADL0CONLbits.SLTSRC = 0;                // Single trigger generated when SAMP is cleared.
ADL0PTR = 0;                            // Start from the first list entry.
// Threshod compare settings.
ADL0CONHbits.CM = 0;                    // Disable threshold compare.
// Channels selection.
ADTBL0bits.ADCH = 0;                    // Channel #0.
ADTBL1bits.ADCH = 1;                    // Channel #1.
// SAMPLE LIST #1 SETTINGS
ADL1CONL = 0;
ADL1CONH = 0;
// Sample list length.
ADL1CONLbits.SLSIZE = 3-1;              // 3 channels are in the list.
// Sampling settings.
ADL1CONHbits.ASEN = 1;                  // Enable auto-scan.
ADL1CONHbits.SLINT = 1;                 // Interrupt after auto-scan completion.
ADL1CONHbits.SAMC = 15;                 // Sample time is 15 TAD.
ADL1CONLbits.SLTSRC = 0;                // Single trigger generated when SAMP is cleared.
ADL1PTR = 0;                            // Start from the first list entry.
// Threshod compare settings.
ADL1CONHbits.CM = 0;                    // Disable threshold compare.
// Channels selection.
ADTBL2bits.ADCH = 2;                    // Channel #2.
ADTBL3bits.ADCH = 3;                    // Channel #3.
ADTBL4bits.ADCH = 4;                    // Channel #4
// SAMPLE LIST #2 SETTINGS
ADL2CONL = 0;
ADL2CONH = 0;
// Sample list length.
ADL2CONLbits.SLSIZE = 1-1;              // 1 channel is in the list.
// Sampling settings.
ADL2CONHbits.ASEN = 1;                  // Enable auto-scan.
ADL2CONHbits.SLINT = 1;                 // Interrupt after auto-scan completion.
ADL2CONHbits.SAMC = 15;                 // Sample time is 15 TAD.
ADL2CONLbits.SLTSRC = 0;                // Single trigger generated when SAMP is cleared.
ADL2PTR = 0;                            // Start from the first list entry.
// Threshod compare settings.
ADL1CONHbits.CM = 0;                    // Disable threshod compare.
// Channels selection.
ADTBL5bits.ADCH = 5;                    // Channel #5.
```

Example 65-2: Multiple Sample List Initialization (Conversion) (Continued)

```
// ENABLE A/D
ADCON1bits.ADON = 1;                    // Enable A/D.
while(ADSTATHbits.ADREADY == 0);        // Wait for ready flag set.
ADCON1bits.ADCAL = 1;                   // Start calibration.
while(ADSTATHbits.ADREADY == 0);
ADL0CONLbits.SAMP = 1;                  // Close sample switch.
ADL0CONLbits.SLEN = 1;                  // Enable sample lists.
ADL1CONLbits.SLEN = 1;
ADL2CONLbits.SLEN = 1;

// CONVERSION
while(1)
{
    ADSTATLbits.SL0IF = 0;
    ADSTATLbits.SL1IF = 0;
    ADSTATLbits.SL2IF = 0;
    ADL0CONLbits.SAMP = 0;              // Start conversion.
    while(                              // Wait for the result.
         (ADSTATLbits.SL0IF == 0)||
         (ADSTATLbits.SL1IF == 0)||
         (ADSTATLbits.SL2IF == 0)
    );
    ADL0CONLbits.SAMP = 1;              // Close the sample switch.
        channel_0 = ADRES0;             // Read result for the channel #0.
        channel_1 = ADRES1;             // Read result for the channel #1.
        channel_2 = ADRES2;             // Read result for the channel #2.
        channel_3 = ADRES3;             // Read result for the channel #3.
        channel_4 = ADRES4;             // Read result for the channel #4.
        channel_5 = ADRES5;             // Read result for the channel #5.
}

}
```

65.9.3.1 SIMULTANEOUS TRIGGERING OF MULTIPLE SAMPLE LISTS

If the application configures and enables more than one Sample List simultaneously, it is possible for two or more Sample Lists to be triggered simultaneously, or close enough in time that the Pipeline A/D Converter is still busy processing one Sample List trigger event when the second trigger event arrives. In this scenario, the analog pipeline can only begin sampling and measuring one channel at a time, as the analog core only provides one Sample-and-Hold capacitor, and only one analog conversion pipeline.

In these instances, the A/D Converter uses fixed priority hardware arbitration to handle simultaneous, or near simultaneous, triggers from multiple lists. Arbitration is based on the list number, with the lowest numbered Sample List that is enabled being processed first. The next highest number enabled list is processed next, and so on.

One trigger event can be queued for each enabled list. If additional trigger events are generated after a trigger event has already been queued up, the additional trigger(s) will be ignored. Once the A/D Converter has processed the queued trigger event for a Sample List, the list may be triggered again.

As an example, if Sample Lists 0 through 2 are all simultaneously enabled, and all are configured to trigger on external INT0 falling edge events, all three Sample Lists will be triggered on an INT0 high-to-low transition. In this scenario, the hardware will:

1. Begin sampling and converting the analog channel selected by Sample List 0, while queuing up one trigger each for Sample List 1 and Sample List 2.
2. Begin sampling and converting the analog channel selected by Sample List 1 once Sample List 0 has been handled.
3. Begin sampling and converting the analog channel selected by Sample List 2 once Sample List 1 has been handled.

If Auto-Scan mode is enabled on a Sample List, the converter will finish sampling all entries in the list prior to moving on to next queued trigger.

65.10 MULTI-CHANNEL OPERATION

The A/D Converter includes a hardware feature to enable multiple analog channels at once (effectively shorting them together through the resistance of the internal MUX). The shorted channels are passed to the S/H cap, which results in a single result. The primary use of this mode is in capacitive touch key arrays, where shorting multiple buttons will indicate an approaching touch (proximity detector). Note that if multiple input channels are enabled, the result is not summed; they are shorted.

This operation is not to be confused with simultaneous sampling, with multiple S/H and A/D Converters. This module has a single S/H and a single A/D Converter, and can only sample one voltage at a time.

Multi-channel operation is implemented individually for each of the Sample Lists. Operation is controlled by the MULCHEN bit (ADLnCONH<5>). Setting MULCHEN allows all of the analog channels selected by the ADLnMSELx registers (Register 65-17 to Register 65-20) to be enabled simultaneously and sampled as a group. Setting or clearing any of the MSELx bits in these registers, respectively, includes or excludes the corresponding analog input in simultaneous sampling.

Note:	Only Analog Input Channels 15 and higher can be simultaneously enabled with the MULCHEN bit. Channel select bits, corresponding to Analog Channels AN0 through AN15, are not provided in the ADLnMSEL0 register.

In addition, setting the MULCHEN bit disables other processing for the Sample List. The SLSIZEx bits and any associated ADTBLn Sample List registers are ignored. Instead, the enabled channels are sampled as a group and converted according to the other bit settings defined in the corresponding ADLnCONL register. Threshold Detect and compare operations may also be performed with the resulting conversion, as dictated by the settings in the ADLnCONH register.

The results of a conversion of a multi-channel operation are stored at the starting address of the Sample List. For Sample List 0, this is always ADRES0. The starting address of other Sample Lists is determined by the current Sample List configuration, as described in **Section 65.9.3 "Using Multiple Sample Lists"**.

65.11 THRESHOLD DETECT OPERATIONS

Threshold Detect extends the capabilities of multi-channel scanning and Sample List operations, by allowing the user to define match conditions, based on the conversion results, and generate an interrupt based on these conditions. This feature is implemented using dedicated digital threshold comparison hardware. During normal operation, Threshold Detect can potentially reduce the amount of CPU time spent on processing A/D interrupts. For low-power applications, this can allow the CPU to remain inactive for longer periods, waking only when specific analog conditions are met.

In the Pipeline A/D Converter, Threshold Detect includes two options for managing comparison thresholds: using dedicated Threshold registers (ADTHnH/ADTHnL) or the ADRESn buffer itself. Threshold Detect also uses the ADCHITH/ADCHITL registers to indicate when threshold match conditions have been met. Independently selectable comparison and buffer storage settings make a wide range of operating combinations possible.

65.11.1 Operating Modes

The operation of Threshold Detect is mostly controlled by the ADLnCONH register. This allows each Sample List to use a different Threshold Detect configuration or to not use it at all.

The Compare Mode bits, CM<2:0> (ADLnCONH<10:8>), select the type of comparison to be performed. Four types are available:

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

The Write Mode bits, WM<1:0> (ADLnCONH<12:11>), determine the disposition of the conversion. Three options are available:

- Discard the conversion data after the comparison has been performed
- Store the conversion data after the comparison has been performed
- Store the conversion data without comparison

65.11.2 Setting Comparison Thresholds

The reference values for Threshold Detect operations can be stored in one of two register sets. The selection is controlled by the THSRC bit (ADLnCONL<7>). When this bit is set, Threshold Detect operations use the ADTHnH and ADTHnL registers (Register 65-15 and Register 65-16) as references for the high and low threshold values, respectively. For non-windowed (Greater Than or Less Than) comparisons, ADTHnL stores the single reference value. Each Sample List has its own pair of Threshold registers.

When THSRC = 0, the A/D Converter uses the ADRESn buffer as a source for reference(s) used during comparison operations. As the ADRESn registers can only be written to in software when the A/D Converter is disabled, the application must preload the ADRESn buffer with the desired reference threshold value(s), prior to enabling the A/D Converter. If it becomes necessary to change the threshold value(s) after the module has been enabled, the application must first disable the module before changing the value(s).

When the Threshold Detect mode is configured for non-windowed (Greater Than or Less Than) comparisons (CM<2:0> = 010 or 001), the single threshold value used for comparison is stored in the ADRESn register that corresponds to the ADTBLn Sample List entry. When the Threshold Detect mode is configured for windowed comparisons (CM<2:0> = 100 or 011), adjacent ADRESn registers are paired, with pairs aligned on the even numbered register. For example, ADRES0 is paired with ADRES1, ADRES2 is paired with ADRES3, etc. The even numbered buffer stores the low threshold value, while the odd numbered buffer stores the high threshold value.

Channel pairing is not enforced by hardware, so paired buffer locations can be used for other purposes. However, storing any other data in a particular buffer location may result in misleading comparison evaluations. In general, it is recommended that the ADTHnH/L registers be used to store threshold values, rather than the ADRESn registers. If the ADRESn registers are to be used to store threshold values, particularly in Windowed modes, it is recommended that FIFO Buffer mode (BUFORG = 0) not be used with Threshold Detect operations.

65.11.3 Compare Hit Registers

To indicate if a Threshold Detect event has occurred, the A/D module uses two registers to record match events. These registers are referred to as the Compare Hit registers, and are designated ADCHITL and ADCHITH. The registers map their individual bits sequentially to each of the (up to) 32 Sample List entries.

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

Depending on the event, more than one Compare Hit bit (CHHx) may be set. The significance of a set bit must be interpreted by the application in the context of the Compare mode selected.

65.11.4 Comparison Mode Examples

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

For simplicity, the examples show only 16 ADRESn registers and one Compare Hit register (ADCHITL). The principles described here apply equally to the balance of ADRESn registers and the ADCHITH register.

Additionally, these examples illustrate operation when the ADRESn buffer is used for threshold storage (THSRC = 0). As previously noted, the use of ADTHnH/L registers to store comparison values (THSRC = 1) is the preferred method. When the ADTHnH/L registers are used to store threshold values, the ADRESn registers may contain any data.

Note:	When using any Comparison mode, always use indexed buffer storage (BUFORG = 1); otherwise, the threshold values for other channels may be overwritten, resulting in unpredictable comparisons.

65.11.4.1 SIMPLE COMPARISONS (GREATER AND LESS THAN RESULTS)

When the Compare Mode bits (CM<2:0>) are programmed as '010' or '001', the converter compares the sampled value to see if it is greater than (CM<2:0> = 010), or less than (CM<2:0> = 001), the threshold value in either of the ADTHnL/ADTHnH registers or in the buffer location. If the condition is met, both of the following occur:

- The Compare Hit bit (CHHx) for the corresponding channel is set.
- If the Write Mode bits, WM<1:0> (ADLnCONH<12:11>), are programmed to '01', the converted value is written to the buffer. When THSRC = 0, the converted value replaces the threshold value already in the ADRESn register.
  If WM<1:0> = 10, the converted value is discarded.

If THSRC = 1, the values in the ADTHnL and ADTHnH registers remain unchanged, regardless of the configuration of the WMx bits. The Compare Hit bits are still set as previously described.

The changes to the result buffer and the Compare Hit register are shown in Figure 65-8. Note that they are the same for both types of simple comparison.

Figure 65-8: Simple Comparison Operations (Greater Than and Less Than), THSRC = 0

Before Conversion and Comparison

ADRES15	—
ADRES14	—
ADRES13	—
ADRES12	—
ADRES11	—
ADRES10	—
ADRES9	—
ADRES8	—
ADRES7	—
ADRES6	—
ADRES5	—
ADRES4	—
ADRES3	—
ADRES2	Threshold Value
ADRES1	—
ADRES0	—

After Conversion and Comparison

	Compare Only ('10')	Compare and Store ('01')
ADRES15	—	—
ADRES14	—	—
ADRES13	—	—
ADRES12	—	—
ADRES11	—	—
ADRES10	—	—
ADRES9	—	—
ADRES8	—	—
ADRES7	—	—
ADRES6	—	—
ADRES5	—	—
ADRES4	—	—
ADRES3	—	—
ADRES2	Threshold Value	**Conversion Value**
ADRES1	—	—
ADRES0	—	—

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

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

65.11.4.2 INSIDE WINDOW COMPARISON

When the Compare Mode bits, CM<2:0>, are programmed as '010' and THSRC = 0, the converter compares the sampled value to see if it falls between the threshold values in the buffer register pair. Since the value in the odd register location is always the greater value of the two thresholds, the condition is met when:

Threshold 2 > Converted Value > Threshold 1

In this case, both of the following occur:

- The Compare Hit bit (CHHx) for the corresponding channel is set; the Compare Hit bit for the paired buffer remains cleared.
- If WM<1:0> are programmed to '01', the converted value is written to the buffer. When THSRC = 0, the converted value replaces the threshold value already in the ADRESn register. If WM<1:0> = 10, the converted value is discarded.

If THSRC = 1, the values in the ADTHnL and ADTHnH registers are used for threshold comparison and remain unchanged, regardless of the configuration of the WMx bits. The Compare Hit bits are still set as previously described.

The changes to the result buffer and the Compare Hit register are shown in Figure 65-9.

Figure 65-9: Inside Window Comparison Operation, THSRC = 0

Before Conversion and Comparison

ADRES15	—
ADRES14	—
ADRES13	—
ADRES12	—
ADRES11	—
ADRES10	—
ADRES9	—
ADRES8	—
ADRES7	—
ADRES6	—
ADRES5	—
ADRES4	—
ADRES3	Threshold 2
ADRES2	Threshold 1
ADRES1	—
ADRES0	—

After Conversion and Comparison

	Compare Only ('10')	Compare and Store ('01')
ADRES15	—	—
ADRES14	—	—
ADRES13	—	—
ADRES12	—	—
ADRES11	—	—
ADRES10	—	—
ADRES9	—	—
ADRES8	—	—
ADRES7	—	—
ADRES6	—	—
ADRES5	—	—
ADRES4	—	—
ADRES3	Threshold 2	Threshold 2
ADRES2	Threshold 1	**Conversion Value**
ADRES1	—	—
ADRES0	—	—

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

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

65.11.4.3 OUTSIDE WINDOW COMPARISON

When the Compare Mode bits CM<2:0> are programmed as '011', and THSRC = 0, the converter compares the sampled value to see if it falls outside of the threshold values in the buffer register pair. Again, since the value in the odd buffer location is always the greater value of the two thresholds, the condition is met when either:

Converted Value > Threshold 2

or

Threshold 1 > Converted Value

In these cases, the following occurs:

- The Compare Hit bit (CHHx) for the corresponding channel is set.
- If the converted value is greater than Threshold 2, the CHHx bit for the odd buffer is also set. If it is less than Threshold 1, the odd buffer bit remains '0'.
- If WM<1:0> is programmed to '01' and THSRC = 0:
  - If the converted value is above Threshold 2, the converted value is written to the odd buffer, replacing the upper threshold value.
  - If the converted value is below Threshold 1, the converted value is written to the even buffer, replacing the lower threshold value.
- If WM<1:0> = 10, the converted value is discarded.

If THSRC = 1, the values in the ADTHnL and ADTHnH registers remain unchanged, regardless of the configuration of the WMx bits. The Compare Hit bits are still set as previously described.

The changes to the result buffer and the Compare Hit register are shown in Figure 65-10 (over the upper threshold) and Figure 65-11 (under the lower threshold).

Figure 65-10: Outside Window Comparison Operation (Over Threshold 2), THSRC = 0

Before Conversion and Comparison

ADRES15	—
ADRES14	—
ADRES13	—
ADRES12	—
ADRES11	—
ADRES10	—
ADRES9	—
ADRES8	—
ADRES7	—
ADRES6	—
ADRES5	—
ADRES4	—
ADRES3	Threshold 2
ADRES2	Threshold 1
ADRES1	—
ADRES0	—

After Conversion and Comparison

	Compare Only ('10')	**Compare and Store ('01')**
ADRES15	—	—
ADRES14	—	—
ADRES13	—	—
ADRES12	—	—
ADRES11	—	—
ADRES10	—	—
ADRES9	—	—
ADRES8	—	—
ADRES7	—	—
ADRES6	—	—
ADRES5	—	—
ADRES4	—	—
ADRES3	Threshold 2	**Conversion Value**
ADRES2	Threshold 1	Threshold 1
ADRES1	—	—
ADRES0	—	—

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

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

Figure 65-11: Outside Window Comparison Operation (Under Threshold 1), THSRC = 0

Before Conversion and Comparison

ADRES15	—
ADRES14	—
ADRES13	—
ADRES12	—
ADRES11	—
ADRES10	
ADRES9	—
ADRES8	—
ADRES7	—
ADRES6	—
ADRES5	—
ADRES4	—
ADRES3	Threshold 2
ADRES2	Threshold 1
ADRES1	—
ADRES0	—

After Conversion and Comparison

	Compare Only ('10')	Compare and Store ('01')
ADRES15	—	—
ADRES14	—	—
ADRES13	—	—
ADRES12	—	—
ADRES11	—	—
ADRES10		
ADRES9	—	—
ADRES8	—	—
ADRES7	—	—
ADRES6	—	—
ADRES5	—	—
ADRES4	—	—
ADRES3	Threshold 2	Threshold 2
ADRES2	Threshold 1	**Conversion Value**
ADRES1	—	—
ADRES0	—	—

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

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

Note that when a Windowed Comparison mode is selected and THSRC = 0, nothing prevents a conversion from another operation from being stored in the paired channel location. In the previous examples, if a Sample List includes an entry for ADTBL3, a Threshold Detect operation on that entry would be tested against the high threshold value already stored there. This could result in the threshold value being overwritten and/or the CHH3 bit being set.

For this reason, users must always carefully consider the allocation and use of Sample Lists and the ADRESn buffers when using Windowed Compare modes. Wherever possible, use the ADTHnL and ADTHnH registers to store threshold values. If that is not possible, exclude the odd ADRESn registers from Threshold Detect operations, and convert and test the corresponding Sample List items in a separate routine.

65.12 SAMPLE-AND-ACCUMULATE OPERATIONS

In many applications, it is desirable to repeatedly sample an analog signal and sum the results. One use for sample accumulation is oversampling, where the average of many samples over time can greatly improve the quality of the result data by effectively removing circuit noise. Oversampling can be performed by software, but this can result in excessive CPU utilization.

The Pipeline A/D Converter provides a hardware facility to perform automated Sample-and-Accumulate operations without CPU intervention. The accumulate function is user-configurable, and is controlled by the ACCONH/ACCONL registers. Accumulated sums, up to 32 bits wide, are stored in the ACRESH/ACRESL registers.

To use the accumulation hardware feature:

1. Configure the A/D Converter's global settings and enable the module (see **Section 65.8 "Initialization"**).
2. Configure and enable a Sample List with a single entry selecting the desired channel to accumulate.
3. Configure the desired accumulation settings:
   a) Configure the interrupt settings
   b) Configure the number of samples to include in the accumulated result before completing the operation and generating an accumulation complete interrupt (if enabled)
   c) Set TBLSEL<5:0> (ACCONL<13:8>) to point to the correct ADTBLn register
4. Clear or preload the ACRESH/ACRESL values (if desired).
5. Enable the accumulation feature by setting the ACEN bit (ACCONH<7>).
6. Generate repeated trigger events for the Sample List configured in Step 2. For each sample result generated using the selected ADTBLn register, the value is added to the ACRESH/ACRESL accumulator result.

The accumulation feature is only intended to accumulate unsigned right justified conversion results. Operation with other data formatting modes is not supported.

The total number of individual samples that will be added/accumulated to the ACRESH/ACRESL registers is determined by the COUNT<7:0> bits (ACCONL<7:0>), which are configured by software prior to setting ACEN. The hardware automatically decrements the COUNTx value after each conversion on the specified channel. Once COUNT<7:0> reaches 00h, the hardware automatically clears ACEN and generates an ACCIF interrupt event (if enabled).

The total number of samples that are accumulated in a single accumulation procedure is equal to the value of the COUNT<7:0> bits prior to starting the process. This allows the accumulator to provide any oversampling ratios of 1:1 to 1:256, depending on the initial value of the COUNTx bits. When COUNTx is preloaded with 00h prior to an accumulation, the hardware performs exactly 256 additions before clearing ACEN.

The A/D Converter does not automatically clear the ACRESH/ACRESL value at the start of an accumulation process. For sample sizes of 256 or lower, the application will need to clear ACRESH/ACRESL prior to starting a new accumulation procedure. Any new result values will be added to the existing ACRESH/ACRESL value. This allows the A/D Converter to accumulate sample sizes greater than 256 for higher oversampling ratios.

If a higher oversampling or accumulation ratio is desired, the application firmware may re-initialize the desired COUNT<7:0> bits field value and set ACEN again.

Note:	Once ACEN is cleared by hardware at the end of an accumulate round, the module hardware requires a small settling time before the next accumulation process may be started. Before setting ACEN to initiate another accumulate round, wait for at least 1 TAD.

Once the accumulation process is complete, the result sum stored in ACRESH/ACRESL can be divided in firmware, by the number of samples in the sum to get an average result, with the same number of bits as the converter's native (12-bit) resolution. If the sample size is an exact power of two, the division operation can be performed quickly by right bit shift operation(s) on the result. Otherwise, the `DIV.UD` instruction may be used. For those using the MPLAB® C30 or XC16 compilers, this instruction can be employed with the `__builtin_divud()` function.

Example 65-3 (pages 45 and 46) demonstrates the use of multiple Sample Lists.

Example 65-3: Sample-and-Accumulate (Configuration and Sample List)

```
#include <p24FJ128GC010.h>
// PIC24FJ128GC010 CONFIGURATION SETTINGS
_CONFIG4(PLLDIV_DIV2 & IOL1WAY_OFF)
_CONFIG3(BOREN_OFF)
_CONFIG2(POSCMD_HS & FNOSC_PRIPLL & FCKSM_CSECME)
_CONFIG1(FWDTEN_WDT_SW & ICS_PGx2)

#define ACC_NUMBER_SAMPLES    16         // Number of samples to accumulate.

// These variables contain the results.
volatile unsigned int  channel_2;
volatile unsigned int  channel_38;
volatile unsigned long accumulator;

int main()
{
// ANALOG INPUTS CONFIGURATION
// 2 analog inputs will be sampled.
TRISBbits.TRISB2 = 1;                    // AN2(RB2)
ANSBbits.ANSB2 = 1;
TRISAbits.TRISA14 = 1;                   // AN38(RA14)
ANSAbits.ANSA14 = 1;

// GLOBAL SETTINGS
ADCON1=0;
ADCON2=0;
ADCON3=0;
// Configure the A/D voltage references.
ADCON2bits.PVCFG = 0;                    // Vref+ = AVdd
ADCON2bits.NVCFG = 0;                    // Vref- = AVss
// Configure the A/D clock.
ADCON3bits.ADRC = 0;                     // Conversion clock derived from system clock.
ADCON3bits.ADCS = 15;                    // Divide system clock by 16(TAD = 1uS @ 16 MIPS).
// Configure buffer storage settings and interrupt generation.
ADCON1bits.FORM = 0;                     // Data output format is a raw integer.
ADCON2bits.BUFORG = 1;                   // Result buffer is an indexed buffer.
ADCON2bits.BUFINT = 0;                   // No buffer interrupt.
// Configure power-saving.
ADCON1bits.PWRLVL = 0;                   // Low power, reduced frequency operation.
ADCON2bits.ADPWR = 3;                    // Module is always powered.

// SAMPLE LIST SETTINGS
ADL0CONL = 0;
ADL0CONH = 0;
// Sample list length.
ADL0CONLbits.SLSIZE = 2-1;               // 2 channels are in the list.
// Sampling settings.
ADL0CONHbits.ASEN = 1;                   // Enable auto-scan.
ADL0CONHbits.SLINT = 1;                  // Interrupt after auto-scan completion.
ADL0CONHbits.SAMC = 15;                  // Sample time is 15 TAD.
ADL0CONLbits.SLTSRC = 0;                 // Single trigger generated when SAMP is cleared.
// Start from the first list entry.
ADL0PTR = 0;
// Threshod compare settings.
ADL0CONHbits.CM = 0;                     // Disable threshod compare.
// Channels selection.
ADTBL0bits.ADCH = 2;                     // Channel #2.
ADTBL1bits.ADCH = 38;                    // Channel #38.
```

Example 65-3: Sample-and-Accumulate (Accumulator and Conversion) (Continued)

```
// ACCUMULATOR SETTINGS
// Set initial value for the accumulator.
ACRESL = 0;
ACRESH = 0;
ACCONLbits.COUNT = ACC_NUMBER_SAMPLES;    // Accumulate 16 samples.
ACCONLbits.TBLSEL = 1;                    // Accumulate channel #38 from entry #1.
ADSTATLbits.ACCIF = 0;                    // Reset accumulator interrupt flag.
ACCONHbits.ACIE = 1;                      // Enable the accumulator interrupt.
ACCONHbits.ACEN = 1;                      // Accumulator enabled.

// ENABLE A/D
ADCON1bits.ADON = 1;                      // Enable A/D.
while(ADSTATHbits.ADREADY == 0);          // Wait for ready flag set.
ADCON1bits.ADCAL = 1;                     // Start calibration.
while(ADSTATHbits.ADREADY == 0);
ADL0CONLbits.SAMP = 1;                    // Close sample switch.
ADL0CONLbits.SLEN = 1;                    // Enable sample list.

// CONVERSION
while(1)
{
    IFS0bits.AD1IF = 0;
    ADL0CONLbits.SAMP = 0;                    // Start conversion.
    while(IFS0bits.AD1IF == 0);               // Wait for the result.
    ADL0CONLbits.SAMP = 1;                    // Close the sample switch.
        channel_2 = ADRES0;                   // Read result for the channel #2.
        channel_38 = ADRES1;                  // Read result for the channel #38.
    if(ADSTATLbits.ACCIF != 0)
    {
        accumulator = ACRESH;                  // Read result from accumulator.
        accumulator <<= 16;
        accumulator |= ACRESL;
        ACRESL = 0;                            // Set initial value for the accumulator.
        ACRESH = 0;
        ACCONLbits.COUNT = ACC_NUMBER_SAMPLES; // Accumulate samples again.
        ACCONHbits.ACEN = 1;                   // Reenable accumulator.
        ADSTATLbits.ACCIF = 0;                 // Reset accumulator interrupt flag.
        //
        // PROCESS ACCUMULATOR DATA HERE
        //
        }
}

}
```

65.13 EFFECTS OF A RESET

If a Reset occurs, the A/D Converter will abandon any operations that may currently be in progress and the A/D Control registers will reset to their default states. This will have the effect of disabling the A/D module. In order to begin using the A/D again, following a Reset event, the firmware will need to fully re-initialize and re-enable the module, Sample Lists, trigger source/settings, etc.

65.14 OPERATION IN SLEEP AND IDLE MODES

65.14.1 Operation in Sleep Modes

The Pipeline A/D Converter can continue to operate and collect measurements in Sleep mode when all of the following conditions are true:

1. The module is configured to use the FRC oscillator as the clock source (e.g., ADRC = 1).
2. The ADSLP bit (ADCON1<12>) is set.
3. An asynchronous trigger event is generated (INT0 event, TMR1 in Asynchronous mode, etc.).

In order to do this, the FRC oscillator needs to remain active in Sleep mode. This happens automatically when the A/D Converter is enabled (ADON = 1), FRC is selected as the clock source (ADRC = 1) and ADSLP is set.

If continuous FRC operation results in Sleep current consumption that is too high for the intended application, a “hurry up and wait” solution may be implemented to minimize the amount of time that the FRC is running. Such a solution involves:

1. Allowing the device to Sleep most of the time with the A/D Converter disabled (ADON = 0).
2. Waking up periodically (Timer1 interrupt, WDT, etc.) and performing a conversion:
   a) Setting ADON to enable the module
   b) Taking the A/D measurement(s)
   c) Clearing ADON to disable the module
3. Re-entering Sleep mode.

65.14.2 Operation in Idle Mode

When the ADSIDL bit is clear, the A/D Converter will continue to operate normally while the microcontroller is in Idle mode. If interrupts are configured and enabled, the module will continue to generate interrupt events when conversions are complete.

65.15 REGISTER MAP

The register map for the 12-Bit, High-Speed Pipeline A/D Converter is shown in Table 65-5. Note this list represents the map for a complete implementation; some devices may have a less complete feature set. Refer to the device data sheet for the device-specific register map.

Table 65-5: 12-Bit, High-Speed Pipeline A/D Converter Register Map

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADCON1	ADON	—	ADSIDL	ADSLP	FORM3	FORM2	FORM1	FORM0	PUMPEN	ADCAL	—	—	—	—	—	PWRLVL	0000
ADCON2	PVCFG1	PVCFG0	—	NVCFG0	—	BUFORG	ADPWR1	ADPWR0	BUFINT1	BUFINT0	—	—	—	—	RFPUMP	ADHALT	0000
ADCON3	ADRC	—	—	—	SLEN3	SLEN2	SLEN1	SLEN0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
ADTMRPR	A/D Trigger Timer Period Value																0000
ADSTATH	—	—	—	—	—	—	—	—	—	—	—	—	—	PUMPST	ADREADY	ADBUSY	0000
ADSTATL	—	—	—	—	—	—	—	SLOV	—	—	BUFIF	ACCIF	SL3IF	SL2IF	SL1IF	SL0IF	0000
ADL0CONH	ASEN	SLINT1	SLINT0	WM1	WM0	CM2	CM1	CM0	CTMEN	PINTRIS	MULCHEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	0000
ADL0CONL	SLEN	SAMP	SLENCLR	SLTSRC4	SLTSRC3	SLTSRC2	SLTSRC1	SLTSRC0	THSRC	—	SLSIZE5	SLSIZE4	SLSIZE3	SLSIZE2	SLSIZE1	SLSIZE0	0000
ADL1CONH	ASEN	SLINT1	SLINT0	WM1	WM0	CM2	CM1	CM0	CTMEN	PINTRIS	MULCHEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	0000
ADL1CONL	SLEN	SAMP	SLENCLR	SLTSRC4	SLTSRC3	SLTSRC2	SLTSRC1	SLTSRC0	THSRC	—	SLSIZE5	SLSIZE4	SLSIZE3	SLSIZE2	SLSIZE1	SLSIZE0	0000
ADL2CONH	ASEN	SLINT1	SLINT0	WM1	WM0	CM2	CM1	CM0	CTMEN	PINTRIS	MULCHEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	0000
ADL2CONL	SLEN	SAMP	SLENCLR	SLTSRC4	SLTSRC3	SLTSRC2	SLTSRC1	SLTSRC0	THSRC	—	SLSIZE5	SLSIZE4	SLSIZE3	SLSIZE2	SLSIZE1	SLSIZE0	0000
ADL3CONH	ASEN	SLINT1	SLINT0	WM1	WM0	CM2	CM1	CM0	CTMEN	PINTRIS	MULCHEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	0000
ADL3CONL	SLEN	SAMP	SLENCLR	SLTSRC4	SLTSRC3	SLTSRC2	SLTSRC1	SLTSRC0	THSRC	—	SLSIZE5	SLSIZE4	SLSIZE3	SLSIZE2	SLSIZE1	SLSIZE0	0000
ADL0PTR	—	ADNEXT6	ADNEXT5	ADNEXT4	ADNEXT3	ADNEXT2	ADNEXT1	ADNEXT0	—	—	—	—	—	—	—	—	0000
ADL1PTR	—	ADNEXT6	ADNEXT5	ADNEXT4	ADNEXT3	ADNEXT2	ADNEXT1	ADNEXT0	—	—	—	—	—	—	—	—	0000
ADL2PTR	—	ADNEXT6	ADNEXT5	ADNEXT4	ADNEXT3	ADNEXT2	ADNEXT1	ADNEXT0	—	—	—	—	—	—	—	—	0000
ADL3PTR	—	ADNEXT6	ADNEXT5	ADNEXT4	ADNEXT3	ADNEXT2	ADNEXT1	ADNEXT0	—	—	—	—	—	—	—	—	0000
ADL0STAT	ADTACT	LBUSY	—	—	—	—	—	—	ADTDLY	—	ADLIF	—	—	—	—	—	0000
ADL1STAT	ADTACT	LBUSY	—	—	—	—	—	—	ADTDLY	—	ADLIF	—	—	—	—	—	0000
ADL2STAT	ADTACT	LBUSY	—	—	—	—	—	—	ADTDLY	—	ADLIF	—	—	—	—	—	0000
ADL3STAT	ADTACT	LBUSY	—	—	—	—	—	—	ADTDLY	—	ADLIF	—	—	—	—	—	0000
ADTBL0	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL1	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
…																	
ADTBL62	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL63	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000

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

Table 65-5: 12-Bit, High-Speed Pipeline A/D Converter Register Map (Continued)

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADRES0	A/D Result Register 0																0000
ADRES1	A/D Result Register 1																0000
...																	
ADRES30	A/D Result Register 30																0000
ADRES31	A/D Result Register 31																0000
ACCONH	—	—	—	—	—	—	—	—	ACEN	ACIE	—	—	—	—	—	—	0000
ACCONL	—	—	TBLSEL5	TBLSEL4	TBLSEL3	TBLSEL2	TBLSEL1	TBLSEL0	COUNT7	COUNT6	COUNT5	COUNT4	COUNT3	COUNT2	COUNT1	COUNT0	0000
ACRESH	A/D Accumulation High Result Register (bits 31-16)																0000
ACRESL	A/D Accumulation Low Result Register (bits 15-0)																0000
ADCHITH	CHH31	CHH30	CHH29	CHH28	CHH27	CHH26	CHH25	CHH24	CHH23	CHH22	CHH21	CHH20	CHH19	CHH18	CHH17	CHH16	0000
ADCHITL	CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8	CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0	0000
ADTH0H	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH0L	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH1H	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH1L	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH2H	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH2L	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH3H	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADTH3L	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	0000
ADL0MSEL3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSEL49	MSEL48	0000
ADL0MSEL2	MSEL47	MSEL46	MSEL45	MSEL44	MSEL43	MSEL42	MSEL41	MSEL40	MSEL39	MSEL38	MSEL37	MSEL36	MSEL35	MSEL34	MSEL33	MSEL32	0000
ADL0MSEL1	MSEL31	MSEL30	MSEL29	MSEL28	MSEL27	MSEL26	MSEL25	MSEL24	MSEL23	MSEL22	MSEL21	MSEL20	MSEL19	MSEL18	MSEL17	MSEL16	0000
ADL0MSEL0	MSEL15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
ADL1MSEL3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSEL49	MSEL48	0000
ADL1MSEL2	MSEL47	MSEL46	MSEL45	MSEL44	MSEL43	MSEL42	MSEL41	MSEL40	MSEL39	MSEL38	MSEL37	MSEL36	MSEL35	MSEL34	MSEL33	MSEL32	0000
ADL1MSEL1	MSEL31	MSEL30	MSEL29	MSEL28	MSEL27	MSEL26	MSEL25	MSEL24	MSEL23	MSEL22	MSEL21	MSEL20	MSEL19	MSEL18	MSEL17	MSEL16	0000
ADL1MSEL0	MSEL15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
ADL2MSEL3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSEL49	MSEL48	0000
ADL2MSEL2	MSEL47	MSEL46	MSEL45	MSEL44	MSEL43	MSEL42	MSEL41	MSEL40	MSEL39	MSEL38	MSEL37	MSEL36	MSEL35	MSEL34	MSEL33	MSEL32	0000
ADL2MSEL1	MSEL31	MSEL30	MSEL29	MSEL28	MSEL27	MSEL26	MSEL25	MSEL24	MSEL23	MSEL22	MSEL21	MSEL20	MSEL19	MSEL18	MSEL17	MSEL16	0000
ADL2MSEL0	MSEL15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
ADL3MSEL3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSEL49	MSEL48	0000
ADL3MSEL2	MSEL47	MSEL46	MSEL45	MSEL44	MSEL43	MSEL42	MSEL41	MSEL40	MSEL39	MSEL38	MSEL37	MSEL36	MSEL35	MSEL34	MSEL33	MSEL32	0000
ADL3MSEL1	MSEL31	MSEL30	MSEL29	MSEL28	MSEL27	MSEL26	MSEL25	MSEL24	MSEL23	MSEL22	MSEL21	MSEL20	MSEL19	MSEL18	MSEL17	MSEL16	0000
ADL3MSEL0	MSEL15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000

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

65.16 DESIGN TIPS

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

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

1. Make sure you are meeting all of the timing specifications. If you are turning the module off and on, there is a minimum delay you must wait before taking a sample. When taking a measurement, ensure that enough time is allowed (SAMC<4:0> bits setting) to adequately charge the Sample-and-Hold capacitor to the analog source voltage. Last but not least, make certain that the A/D Converter is correctly configured. The A/D Converter has both a minimum and maximum TAD clock frequency that it may be operated at. Operating the A/D Converter outside of the specified range can introduce significant additional error or otherwise prevent the module from working correctly.
2. Try to minimize analog source impedance wherever possible. High-impedance analog sources are more susceptible to injected noise from other nearby sources, are affected more by I/O pin leakage and require longer A/D Converter sample capacitor charging times. In addition, other global settings, such as RFPUMP and PUMPEN, should be configured correctly to optimize the signal path, and the converter's internal operation.
3. Make sure that the analog source signals being sampled by the A/D Converter are adequately separated or shielded from digital, or other high-dV/dt signals at the PCB design level. The converter's Sample-and-Hold capacitor is very small, often of the same order of magnitude or smaller than the PCB parasitic capacitance between two parallel traces routed next to each other on a typical PCB. When a nearby adjacent trace (e.g., a digital clock or data line) toggles, this can inject a small capacitive charge into analog signal traces nearby. The severity and quantity of the injected charge depends on the physical separation distance, and the length that the two traces run parallel on the PCB. On PCBs with long parallel trace lengths, this digital I/O pin toggling can inject enough charge to change the A/D Converter measurement by many Least Significant bit counts. This potential source of noise can be mitigated by increasing the trace separation distance and/or adding a shielded "dummy" trace in between (that can be connected to AVSS or some other pure DC low-impedance net).

   Additionally, adding a small additional capacitance (e.g., 100 pF) onto the analog net can, in some cases, help to reduce the AC impedance of the analog signal, thereby absorbing some of the injected charge and resulting in a more stable signal. Intentionally slew rate limiting nearby digital (or otherwise high-dV/dt) traces can also help. Typically, this can be accomplished by adding some additional resistance (typically <1 kΩ) directly in-series with the digital trace, located on the output pin/driver that is driving the trace.
4. The pipeline architecture of this converter generates a higher average noise floor than traditional SAR converters. To circumvent this, in most applications, this converter is operated at a higher speed and multiple samples are taken. This can either be in the form of successive samples (effectively a moving average filter) or groups of readings are taken in sets and averaged (e.g., eight samples per reading).

Question 2: ***Can you recommend a good reference on A/D Converters?***

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

65.17 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent, and could be used with modification and possible limitations. The current application notes related to the 12-Bit, High-Speed Pipeline A/D Converter are:

Title	Application Note #
Understanding A/D Converter Performance Specifications	AN693

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

65.18 REVISION HISTORY

Revision A (March 2013)

Original version of this document.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-62077-082-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

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

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

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

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

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

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

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

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

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

ASIA/PACIFIC

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

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

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

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

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

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

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

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

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

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

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

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

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

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

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

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

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

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

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

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

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

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

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

EUROPE

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

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

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

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

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

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

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

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

11/29/12