Section 18. 12-bit Pipelined Analog-to-Digital Converter (ADC)

HIGHLIGHTS

This section of the manual contains the following major topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all PIC32 devices.

Please refer to the note at the beginning of the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com

18.1 INTRODUCTION

The PIC32 12-bit pipelined Analog-to-Digital Converter (ADC) includes the following features:

- 12-bit resolution
- Six-stage conversion and four-stage processing pipeline
- 357 ns minimum conversion latency:
  - Up to 28 Msps total conversion rate
- External voltage reference input pins
- Six Sample and Hold (S&H) circuits, SH0 through SH5:
  - Five dedicated S&H circuits with individual input selection and individual conversion trigger selection for high-speed conversions
  - One shared S&H circuit with automatic Input Scan mode and common conversion trigger selection
- Up to 48 analog input sources, in addition to the internal voltage reference and an internal band gap temperature sensor
- Separate 32-bit conversion result register for each analog input
  - Conversion result can be formatted as unsigned or signed data
- Six digital comparators:
  - Multiple comparison options
  - Assignable to specific analog input
- Six oversampling filters:
  - Provides increased resolution
  - Assignable to specific analog input
- Operation during CPU Sleep and Idle modes

A simplified block diagram of the ADC1 module is illustrated in Figure 18-1. Diagrams of the related S&H circuits are provided in Figure 18-2 and Figure 18-3.

In each of the five dedicated S&H circuits, the analog inputs are connected through multiplexers and switches to the S&H capacitor. These multiplexers allow for input and configuration selection in the S&H circuit. The configuration settings SHxALT<1:0> and SHxMOD<1:0> configure the multiplexers and are passed to the six stage pipeline converter along with the analog sample at the start of conversion. The converter uses the digital configuration information to process the analog sample, which allows it to know the number format, the measurement mode, and which ANx input the sample was collected from. When conversion is complete, the final result is stored in the result buffer for the specific analog input and passed to the digital filter and digital comparator if configured to use data from this particular sample.

The single shared S&H incorporates a large multiplexer on the input. While the dedicated S&H uses a single input (or its alternate) and is intending for high-speed and precise sampling of time sensitive or transient signals, the shared S&H is connected to all remaining inputs and provides flexibility and automated scanning of large groups of inputs using the input scan logic.

Details regarding the dedicated and shared S&H circuits are described in **18.3.2 "Dedicated Sample and Hold"** and **18.3.3 "Shared Sample and Hold"**. The analog inputs and their association with the dedicated and shared S&H circuits form three classes of inputs. Inputs to dedicated S&H are Class 1 while inputs to the shared S&H are Class 2 or Class 3. These input types are explained in **18.3.1 "Analog Inputs"**. The Pipeline converter is described in **18.3.4 "Six-Stage Pipeline Converter"**.

Figure 18-1: ADC1 Module Block Diagram

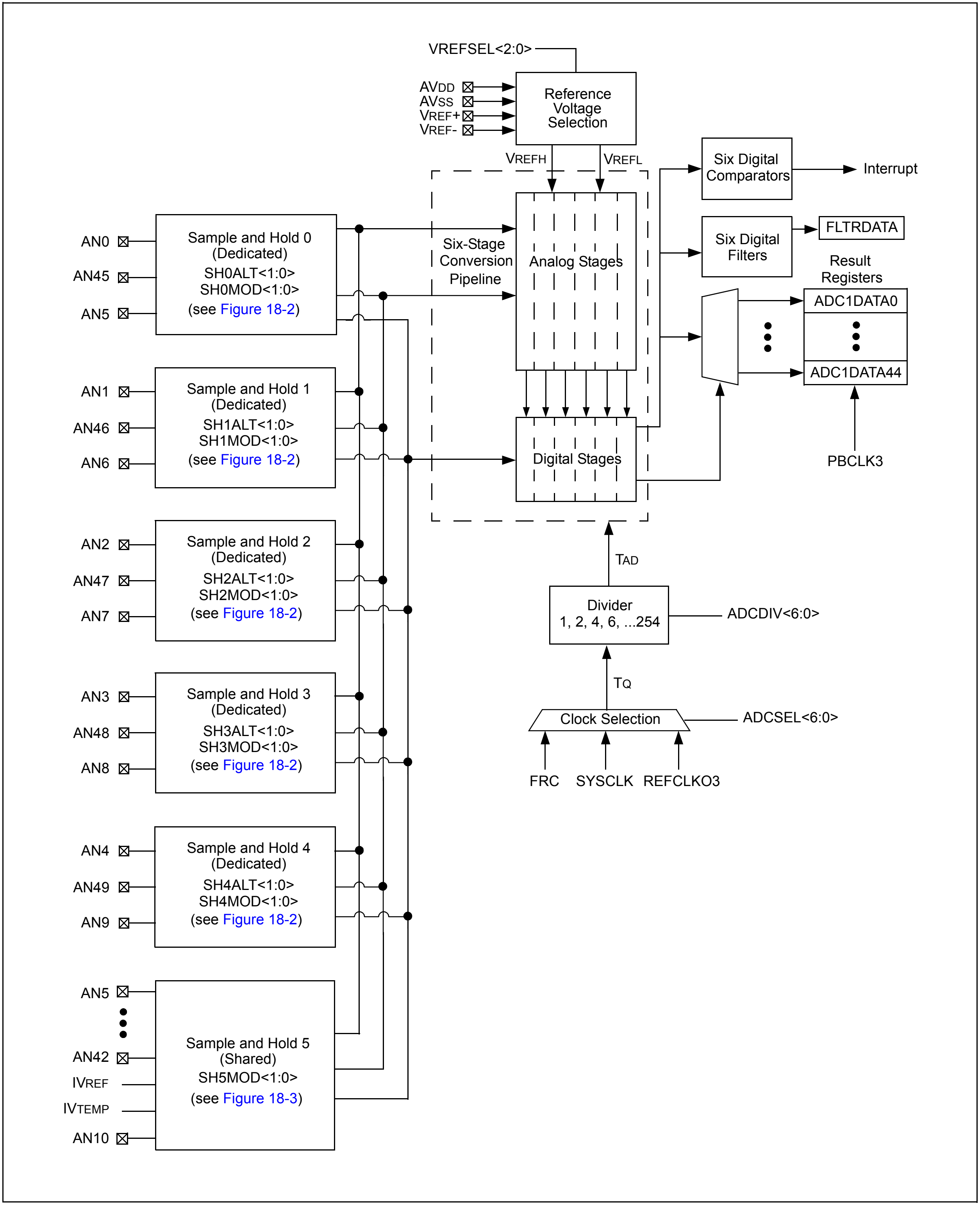
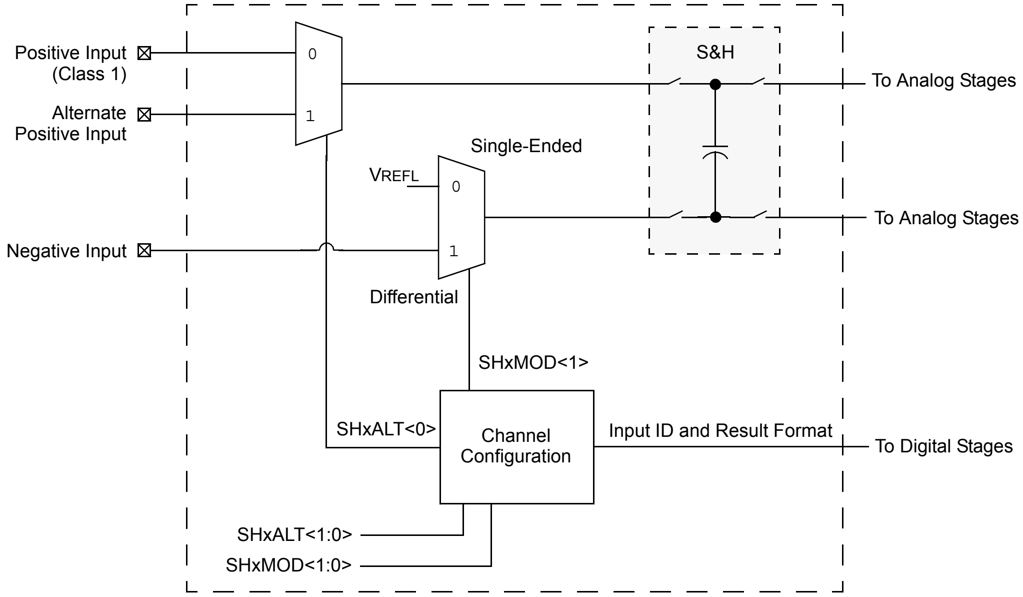
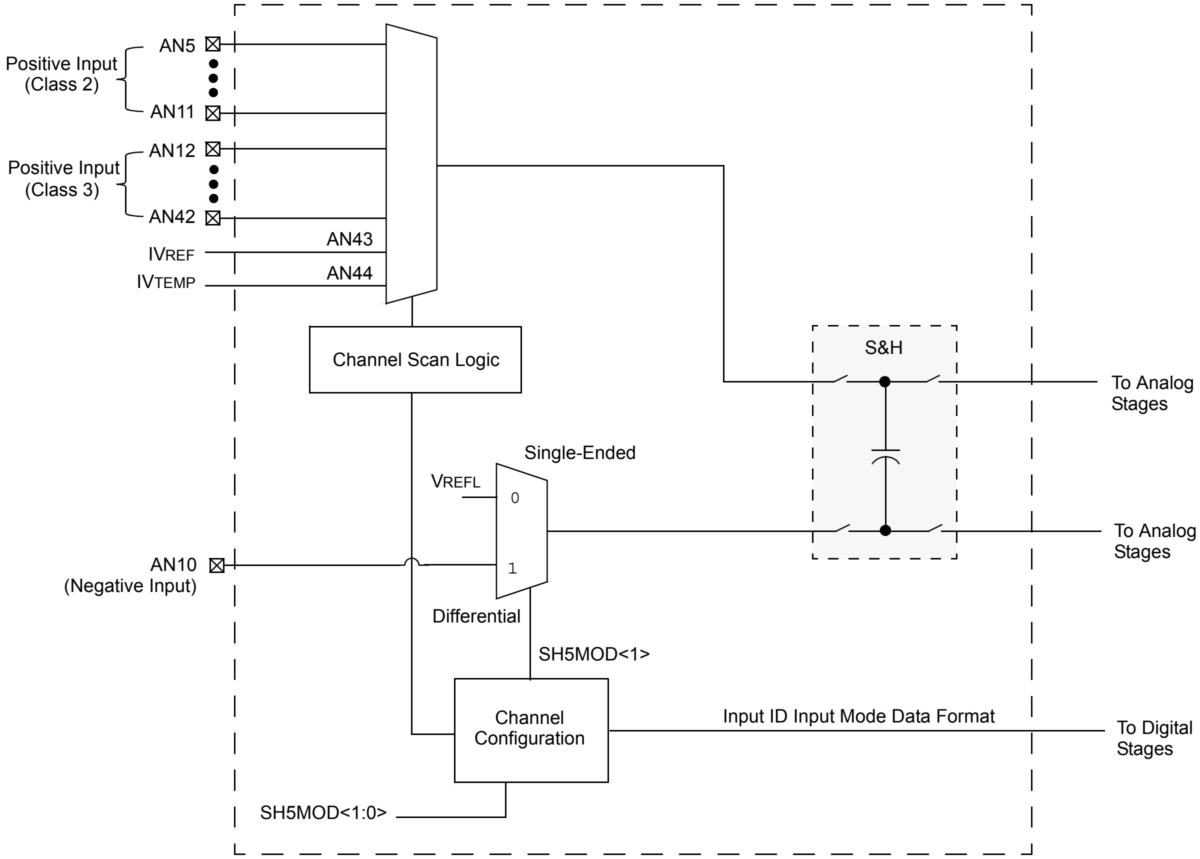


Figure 18-2: Dedicated S&H 0-4 Block Diagram



Note: Refer to the **"12-bit Pipelined Analog-to-Digital Converter"** chapter in the specific device data sheet for the analog input assignments for the dedicated and shared S&H circuits.

Figure 18-3: Shared S&H 5 Block Diagram



Note: Refer to the **"12-bit Pipelined Analog-to-Digital Converter"** chapter in the specific device data sheet for the analog input assignments for the dedicated and shared S&H circuits.

18.2 CONTROL REGISTERS

The PIC32 12-bit Pipelined ADC module has the following Special Function Registers (SFRs):

- **AD1CON1: ADC1 Control Register 1**

  This register controls the basic operation of the ADC module, including behavior in Sleep and Idle modes, and data formatting. This register specifies the vector shift amounts for the interrupt controller. Additional AD1CON1 functions include early interrupt generation and the trigger selection for the scanned analog inputs.

- **AD1CON2: ADC1 Control Register 2**

  This register controls the clock source for the ADC module, the clock divider and the auto sampling time for the shared S&H circuits. This register controls and optimizes the performance of the ADC core circuitry, and also contains a control bit for putting the ADC module into a special low-power state.

- **AD1CON3: ADC1 Control Register 3**

  This register enables the user software to explicitly request the conversion of a specific analog input. The CAL bit can be used for manually starting an ADC calibration process.

- **AD1IMOD: ADC1 Input Mode Control Register**

  These registers enable the user to select an alternate analog input for each of the five dedicated S&H circuits (SH0 to SH4). Also, the AD1IMOD register enables the user to select between single-ended and differential operation as well as select between signed and unsigned data format.

- **AD1GIRQEN1: ADC1 Global Interrupt Enable Register 1** and **AD1GIRQEN2: ADC1 Interrupt Enable Register 2**

  These registers specify which of the individual input conversion interrupts can generate the global ADC1 interrupt.

- **AD1CSS1: ADC1 Input Scan Select Register 1** and **AD1CSS2: ADC1 Input Scan Select Register 2**

  These registers specify the analog inputs to be scanned by the common scan trigger.

- **AD1DSTAT1: ADC1 Data Ready Status Register 1** and **AD1DSTAT2: ADC1 Data Ready Status Register 2**

  These registers contain the interrupt status of the individual analog input conversions. Each bit represents the data-ready status for its associated conversion result.

- **AD1CMPCONn: ADC1 Digital Comparator Control Register 'n' ('n' = 1, 2, 3, 4, 5, or 6)**

  These registers control the operation of the digital comparator, including the generation of interrupts and the comparison criteria to be used. This register also provides status when a comparator event occurs.

- **AD1CMPENn: ADC1 Digital Comparator Enable Register 'n' ('n' = 1, 2, 3, 4, 5, or 6)**

  These registers select which analog input conversion results will be processed by the digital comparator.

- **AD1CMPn: ADC1 Digital Comparator Register 'n' ('n' = 1, 2, 3, 4, 5 or 6)**

  These registers contain the high and low digital comparison values for use by the digital comparator.

- **AD1FLTRn: ADC1 Filter Register 'n' ('n' = 1, 2, 3, 4, 5, or 6)**

  These registers provide control and status bits for the oversampling filter accumulator, and also includes the 16-bit filter output data.

- **AD1TRGR1: ADC1 Input Convert Control Register 1**

  This register controls the trigger source selection for the analog inputs, AN0 through AN3.

- **AD1TRGR2: ADC1 Input Convert Control Register 2**

  This register controls the trigger source selection for the analog inputs, AN4 through AN7.

- **AD1TRGR3: ADC1 Input Convert Control Register 3**

  This register controls the trigger source selection for the analog inputs, AN8 through AN11.
- **AD1DATAn: ADC1 Data Output Register (‘n’ = 0 to 44)**

  These registers are the analog-to-digital conversion output data registers. AD1DATAn register is associated with each analog input ‘n’.
- **AD1CALx: ADC1 Calibration Register (‘x’ = 1-5)**

  These registers contains the ADC calibration value.

The ADC module also has the following associated bits for interrupt control (for more information on the location of these bits, refer to the **“Interrupt Controller”** chapter in the specific device data sheet:

- Interrupt Request Flag Status bits (AD1IF)
- Interrupt Enable Control bits (AD1IE)
- Interrupt Priority Control bits (AD1IP<2:0>) and (AD1IS<1:0>)

Table 18-1 provides a summary of all ADC Special Function Registers (SFRs). Corresponding registers appear after the summaries, which include a detailed description of each bit.

Table 18-1: ADC SFR Summary

Register Name	Bit Range	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
AD1CON1	31:16	FILTRDLY<4:0>					STRGSRC<4:0>					—	—	—	EIE<2:0>		
	15:0	ADCEN	—	ADSIDL	—	FRACT	—	—	—	—	—	—	—	—	—	—	—
AD1CON2	31:16	ADCRDY	—	—	—	—	—	—	—	SAMC<7:0>							
	15:0	—	BOOST	LOWPWR	—	—	—	ADCSEL<1:0>		—	ADCDIV<6:0>						
AD1CON3	31:16	CAL	GSWTRG	RQCONVRT	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	VREFSEL<2:0>			—	—	—	—	ADINSEL<5:0>					
AD1IMOD	31:16	—	—	—	—	—	—	SH4ALT<1:0>		SH3ALT<1:0>		SH2ALT<1:0>		SH1ALT<1:0>		SH0ALT<1:0>	
	15:0	—	—	—	—	SH5MOD<1:0>		SH4MOD<1:0>		SH3MOD<1:0>		SH2MOD<1:0>		SH1MOD<1:0>		SH0MOD<1:0>	
AD1GIRQEN1	31:16	AGIEN31	AGIEN30	AGIEN29	AGIEN28	AGIEN27	AGIEN26	AGIEN25	AGIEN24	AGIEN23	AGIEN22	AGIEN21	AGIEN20	AGIEN19	AGIEN18	AGIEN17	AGIEN16
	15:0	AGIEN15	AGIEN14	AGIEN13	AGIEN12	AGIEN11	AGIEN10	AGIEN9	AGIEN8	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0
AD1GIRQEN2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AGIEN44	AGIEN43	AGIEN42	AGIEN41	AGIEN40	AGIEN39	AGIEN38	AGIEN37	AGIEN36	AGIEN35	AGIEN34	AGIEN33	AGIEN32
AD1CSS1	31:16	CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
	15:0	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
AD1CSS2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	CSS44	CSS43	CSS42	CSS41	CSS40	CSS39	CSS38	CSS37	CSS36	CSS35	CSS34	CSS33	CSS32
AD1DSTAT1	31:16	ARDY31	ARDY30	ARDY29	ARDY28	ARDY27	ARDY26	ARDY25	ARDY24	ARDY23	ARDY22	ARDY21	ARDY20	ARDY19	ARDY18	ARDY17	ARDY16
	15:0	ARDY15	ARDY14	ARDY13	ARDY12	ARDY11	ARDY10	ARDY9	ARDY8	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0
AD1DSTAT2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	ARDY44	ARDY43	ARDY42	ARDY41	ARDY40	ARDY39	ARDY38	ARDY37	ARDY36	ARDY35	ARDY34	ARDY33	ARDY32
AD1CMPCON1	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPCON2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPCON3	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPCON4	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPCON5	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPCON6	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	AINID<4:0>					ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
AD1CMPEN1	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0

Legend: — = unimplemented, read as ‘0’.

Table 18-1: ADC SFR Summary (Continued)

Register Name	Bit Range	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
AD1CMPEN2	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
AD1CMPEN3	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
AD1CMPEN4	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
AD1CMPEN5	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
AD1CMPEN6	31:16	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
	15:0	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
AD1CMP1	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1CMP2	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1CMP3	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1CMP4	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1CMP5	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1CMP6	31:16	ADCCMPHI<15:0>															
	15:0	ADCCMPLO<15:0>															
AD1FLTR1	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1FLTR2	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1FLTR3	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1FLTR4	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1FLTR5	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1FLTR6	31:16	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY	—	—	CHNLID<5:0>					
	15:0	FLTRDATA<15:0>															
AD1TRG1	31:16	—	—	—	TRGSRC3<4:0>					—	—	—	TRGSRC2<4:0>				
	15:0	—	—	—	TRGSRC1<4:0>					—	—	—	TRGSRC0<4:0>				

Legend: — = unimplemented, read as ‘0’.

Table 18-1: ADC SFR Summary (Continued)

Register Name	Bit Range	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
AD1TRG2	31:16	—	—	—	TRGSRC7<4:0>					—	—	—	TRGSRC6<4:0>				
	15:0	—	—	—	TRGSRC5<4:0>					—	—	—	TRGSRC4<4:0>				
AD1TRG3	31:16	—	—	—	TRGSRC11<4:0>					—	—	—	TRGSRC10<4:0>				
	15:0	—	—	—	TRGSRC9<4:0>					—	—	—	TRGSRC8<4:0>				
AD1DATAn	31:16	DATA<31:16>															
	15:0	DATA<15:0>															
AD1CAL1	31:16	ADCAL<31:16>															
	15:0	ADCAL<15:0>															
AD1CAL2	31:16	ADCAL<31:16>															
	15:0	ADCAL<15:0>															
AD1CAL3	31:16	ADCAL<31:16>															
	15:0	ADCAL<15:0>															
AD1CAL4	31:16	ADCAL<31:16>															
	15:0	ADCAL<15:0>															
AD1CAL5	31:16	ADCAL<31:16>															
	15:0	ADCAL<15:0>															

Legend: — = unimplemented, read as ‘0’.

Register 18-1: AD1CON1: ADC1 Control Register 1

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	FILTRDLY<4:0>					STRGSRC<4:2>[1,4]		
23:16	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
	STRGSRC<1:0>		—	—	—	EIE<2:0>[2]		
15:8	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
	ADCEN[3]	—	ADSIDL	—	FRACT	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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 31-27 **FILTRDLY<4:0>:** Oversampling Digital Filter Delay bits
Specifies the sampling time for subsequent automatic triggers when using the Oversampling Digital Filter. Sample time is 1.5 + FILTRDLY<4:0> TAD.

`11111` = Sample time is 32.5 TAD
`11110` = Sample time is 31.5 TAD
•
•
•
`00001` = Sample time is 2.5 TAD
`00000` = Sample time is 1.5 TAD

bit 26-22 **STRGSRC<4:0>:** Scan Trigger Source Select bits[1,4]

`11111` = Reserved
•
•
•
`01101` = Reserved
`01100` = Comparator 2 COUT
`01011` = Comparator 1 COUT
`01010` = OCMP5
`01001` = OCMP3
`01000` = OCMP1
`00111` = TMR5 match
`00110` = TMR3 match
`00101` = TMR1 match
`00100` = INT0
`00011` = Reserved
`00010` = Reserved
`00001` = Global software trigger (GSWTRG)
`00000` = No trigger

Note 1: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available

2: The early interrupt feature should not be used if polling any of the ARDYx bits to determine if the conversion is complete. Early interrupts should be used only when all results from the ADC module are retrieved using an individual interrupt routine to fetch ADC results.

3: The ADCEN bit should be set only after the ADC module has been configured. Changing ADC Configuration bits such as ADCSEL<1:0> and ADCDIV<6:0>, when ADCEN = `1`, will result in unpredictable behavior. When ADCEN = `0`, the ADC clocks are disabled, the internal control logic is reset, and all status flags used by the module are cleared. However, the SFRs are available for reading and writing.

4: Use of the scan trigger requires set up of specific channels in the AD1TRGR1, AD1TRGR2, and/or AD1TRGR3 register.

Note: The ADC module is not available for normal operations until the ADCRDY bit (AD1CON2<31>) is set.

Register 18-1: AD1CON1: ADC1 Control Register 1 (Continued)

bit 21-19 **Unimplemented:** Read as ‘0’

bit 18-16 **EIE<2:0>:** Early Interrupt Enable bits(2)

These bits select the number of clocks prior to the actual arrival of valid data when the associated ARDYx bit is set. Since the ARDYx bit triggers an interrupt, these bits allow for early interrupt generation when an individual interrupt routine is used to fetch each ADC result.

111 = The associated data ready bit, ARDYx, is set 7 TAD clocks prior to when the data is ready
110 = The associated data ready bit, ARDYx, is set 6 TAD clocks prior to when the data is ready
101 = The associated data ready bit, ARDYx, is set 5 TAD clocks prior to when the data is ready
100 = The associated data ready bit, ARDYx, is set 4 TAD clocks prior to when the data is ready
011 = The associated data ready bit, ARDYx, is set 3 TAD clocks prior to when the data is ready
010 = The associated data ready bit, ARDYx, is set 2 TAD clocks prior to when the data is ready
001 = The associated data ready bit, ARDYx, is set 1 TAD clock prior to when the data is ready
000 = The associated data ready bit, ARDYx, is set when the data is ready

bit 15 **ADCEN:** ADC Operating Mode bit(3)

1 = Enable the ADC module
0 = Disable the ADC module

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ADSIDL:** Stop in Idle Mode bit

1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12 **Unimplemented:** Read as ‘0’

bit 11 **FRACT:** Fractional Data Output Format bit

Refer to **18.4.5 “Selecting the Format of the ADC Result”** for specific format information.

1 = Fractional
0 = Integer

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

Note 1: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available

2: The early interrupt feature should not be used if polling any of the ARDYx bits to determine if the conversion is complete. Early interrupts should be used only when all results from the ADC module are retrieved using an individual interrupt routine to fetch ADC results.

3: The ADCEN bit should be set only after the ADC module has been configured. Changing ADC Configuration bits such as ADCSEL<1:0> and ADCDIV<6:0>, when ADCEN = 1, will result in unpredictable behavior. When ADCEN = 0, the ADC clocks are disabled, the internal control logic is reset, and all status flags used by the module are cleared. However, the SFRs are available for reading and writing.

4: Use of the scan trigger requires set up of specific channels in the AD1TRGR1, AD1TRGR2, and/or AD1TRGR3 register.

Note: The ADC module is not available for normal operations until the ADCRDY bit (AD1CON2<31>) is set.

Register 18-2: AD1CON2: ADC1 Control Register 2

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	HS, HC, R-x	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	ADCRDY(1)	—	—	—	—	—	—	—
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SAMC<7:0>							
15:8	U-0	R/W-0	R/W-0	U-0	U-0	r-0	R/W-0	R/W-0
	—	BOOST	LOWPWR	—	—	—	ADCSEL<1:0>(2,3)	
7:0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	ADCDIV<6:0>(2)						

Legend:	HS = Set by Hardware	HC = Cleared by Hardware	r = Reserved
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 31 **ADCRDY:** ADC Ready bit(1)

1 = ADC module is ready for normal operation
0 = ADC is not ready for use

bit 30-24 **Unimplemented:** Read as ‘0’

bit 23-16 **SAMC<7:0>:** Sample Time for Shared S&H bits

11111111 = 256 TAD
•
•
•
00000001 = 2 TAD
00000000 = 1 TAD

This field specifies the number of ADC clock cycles allocated to the ADC sample time for the shared S&H circuit.

bit 15 **Unimplemented:** Read as ‘0’

bit 14 **BOOST:** Boost Voltage Reference bit

1 = Boost VREF
0 = Do not boost VREF

Refer to the **“Electrical Characteristics”** chapter in the specific device data sheet for the VREF specification.

Setting this bit maximizes the Signal-to-Noise Ratio (SNR) when the reference voltage, VREF (VREFH - VREFL), is less than 0.65 * (AVDD - AVSS). See **18.4.7 “Selecting the Voltage Reference Source”** for more information. Changing the state of this bit requires that the ADC module be recalibrated by setting the CAL bit (AD1CON3<31>).

bit 13 **LOWPWR:** ADC Low-power bit

1 = Force the ADC module into a low-power state. This low-power state allows nearly immediate operation after clearing this bit without requiring a calibration cycle.
0 = Exit ADC low-power state

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

bit 10 **Reserved:** Always write ‘0’ to this location

Note 1: This bit is set to ‘0’ when ADCEN (AD1CON1<15>) = 0.

2: These bits should be configured prior to enabling the ADC by setting the ADCEN bit (AD1CON1<15>) = 1.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-2: AD1CON2: ADC1 Control Register 2 (Continued)

bit 9-8 **ADCSEL<1:0>:** ADC Clock Source (T_Q) bits[2,3]

`11` = FRC Oscillator output
`10` = REFCLK3
`01` = System clock (T_{CY})
`00` = Reserved

bit 7 **Unimplemented:** Read as ‘`0`’

bit 6-0 **ADCDIV<6:0>:** ADC Input Clock Divider bits[2]

These bits divide the selected clock source to derive the desired ADC clock rate (T_{AD}).

`1111111` = 2 T_Q * (ADCDIV<6:0>) = 254 * T_Q = T_{AD}

•

•

•

`0000011` = 2 T_Q * (ADCDIV<6:0>) = 6 * T_Q = T_{AD}
`0000010` = 2 T_Q * (ADCDIV<6:0>) = 4 * T_Q = T_{AD}
`0000001` = 2 T_Q * (ADCDIV<6:0>) = 2 * T_Q = T_{AD}
`0000000` = T_Q = T_{AD}

Note 1: This bit is set to ‘`0`’ when ADCEN (AD1CON1<15>) = `0`.

2: These bits should be configured prior to enabling the ADC by setting the ADCEN bit (AD1CON1<15>) = `1`.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-3: AD1CON3: ADC1 Control Register 3

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0, HC	R/W-0, HC	R/W-0, HC	U-0	U-0	U-0	U-0	U-0
	CAL	GSWTRG	RQCONVRT	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0
	—	—	—	VREFSEL<2:0>(2)			—	—
7:0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	ADINSEL<5:0>(1)					

Legend:		HC = Cleared by Hardware	
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 31 **CAL:** Calibration bit

During the calibration process, the ADCRDY status bit in the AD1CON2 register is cleared. When calibration is complete, the CAL bit is cleared and the ADCRDY status bit is set.

1 = ADC module is performing a calibration cycle (calibration requires approximately 160 TAD clock cycles)
0 = Calibration cycle is not in progress

bit 30 **GSWTRG:** Global Software Trigger bit

1 = Trigger analog-to-digital conversion for ADC inputs that have selected the GSWTRG bit as the trigger signal, either through the associated TRGSRC<4:0> bits in the AD1TRGn registers or through the STRGSRC<4:0> bits in the AD1CON1 register.
0 = This bit is automatically cleared within 2 PBCLK3 clock cycles

bit 29 **RQCONVRT:** Individual ADC Input Conversion Request bit

This bit and its associated ADINSEL<5:0> bits enable the user to individually request an analog-to-digital conversion of an analog input without having to reprogram the TRGSRC<4:0> bits or the STRGSRC<4:0> bits. This is very useful during debugging or error handling situations where the user software needs to obtain an immediate ADC result of a specific input.

1 = Trigger the conversion of the selected ADC input as specified by the ADINSEL<5:0> bits
0 = Do not trigger the conversion; this bit is automatically cleared

bit 28-13 **Unimplemented:** Read as ‘0’

bit 12-10 **VREFSEL<2:0>:** VREF Input Selection bits(2)

VREFSEL<2:0>	VREFH	VREFL
111	Reserved	Reserved
110	Reserved	Reserved
101	Reserved	Reserved
100	Reserved	Reserved
011	VREF+	VREF-
010	AVDD	VREF-
001	VREF+	AVss
000	AVDD	AVss

bit 9-6 **Unimplemented:** Read as ‘0’

Note 1: Refer to the **“12-bit Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which analog inputs are available.

2: These bits should be configured prior to enabling the ADC module by setting the ADCEN bit (AD1CON1<15>) = 1.

Register 18-3: AD1CON3: ADC1 Control Register 3 (Continued)

bit 5-0 **ADINSEL<5:0>:** ADC Input Select bits[1]

This binary encoded bit-field selects the ADC module input to be converted when the RQCONVRT bit is set.

`111111` = Reserved
•
•
•
`101101` = Reserved
`101100` = IV_{TEMP} (internal temperature reference voltage)
`101011` = IV_{REF} (internal voltage reference)
`101010` = AN42
•
•
•
`000010` = AN2
`000001` = AN1
`000000` = AN0

Note 1: Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet to determine which analog inputs are available.

2: These bits should be configured prior to enabling the ADC module by setting the ADCEN bit (AD1CON1<15>) = `1`.

Register 18-4: AD1IMOD: ADC1 Input Mode Control Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
	—	—	—	—	—	—	SH4ALT<1:0>	
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SH3ALT<1:0>		SH2ALT<1:0>		SH1ALT<1:0>		SH0ALT<1:0>	
15:8	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	—	SH5MOD<1:0>		SH4MOD<1:0>	
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SH3MOD<1:0>		SH2MOD<1:0>		SH1MOD<1:0>		SH0MOD<1:0>	

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 31-26 **Unimplemented:** Read as ‘0’

bit 25-24 **SH4ALT<1:0>:** Analog Input to Dedicated S&H 4 (SH4) Select bits

11 = Reserved; do not use
10 = Reserved; do not use
01 = Alternate input
00 = Default Class 1 input AN4

bit 23-22 **SH3ALT<1:0>:** Analog Input to Dedicated S&H 3 (SH3) Select bits

11 = Reserved; do not use
10 = Reserved; do not use
01 = Alternate input
00 = Default Class 1 input AN3

bit 21-20 **SH2ALT<1:0>:** Analog Input to Dedicated S&H 2 (SH2) Select bits

11 = Reserved; do not use
10 = Reserved; do not use
01 = Alternate input
00 = Default Class 1 input AN2

bit 19-18 **SH1ALT<1:0>:** Analog Input to Dedicated S&H 1 (SH1) Select bits

11 = Reserved; do not use
10 = Reserved; do not use
01 = Alternate input
00 = Default Class 1 input AN1

bit 17-16 **SH0ALT<1:0>:** Analog Input to Dedicated S&H 0 (SH0) Select bits

11 = Reserved; do not use
10 = Reserved; do not use
01 = Alternate input
00 = Default Class 1 input AN0

Note 1: Alternate inputs are only available for Class 1 Inputs. Refer to the **“12-bit Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine the available Class 1 inputs and alternates for those inputs.

2: When an alternate input is selected (SHxALT<1:0> ≠ 0) the data, status and control registers for the default Class 1 input are still used. Selecting an alternate input changes the physical input source only.

3: SHxMOD<1:0> selects both the data output format and input selection for the negative input. Refer to the **“12-bit Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine the available ANx input options for the negative input.

4: Some bits may not be present on all devices. Refer to the **“12-bit Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet for availability.

Register 18-4: AD1IMOD: ADC1 Input Mode Control Register (Continued)

bit 15-12	**Unimplemented:** Read as '0'
bit 11-10	**SH5MOD<1:0>:** Input Configuration for S&H 5 (SH5) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output
bit 9-8	**SH4MOD<1:0>:** Input Configuration for S&H 4 (SH4) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output
bit 7-6	**SH3MOD<1:0>:** Input Configuration for S&H 3 (SH3) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output
bit 5-4	**SH2MOD<1:0>:** Input Configuration for S&H 2 (SH2) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output
bit 3-2	**SH1MOD<1:0>:** Input Configuration for S&H 1 (SH1) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output
bit 1-0	**SH0MOD<1:0>:** Input Configuration for S&H 0 (SH0) Select bits 11 = Differential inputs, two's complement (signed) data output 10 = Differential inputs, unipolar encoded (unsigned) data output 01 = Single ended inputs, two's complement (signed) data output 00 = Single ended inputs, unipolar encoded (unsigned) data output

Note 1: Alternate inputs are only available for Class 1 Inputs. Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet to determine the available Class 1 inputs and alternates for those inputs.

2: When an alternate input is selected (SHxALT<1:0> ≠ 0) the data, status and control registers for the default Class 1 input are still used. Selecting an alternate input changes the physical input source only.

3: SHxMOD<1:0> selects both the data output format and input selection for the negative input. Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet to determine the available ANx input options for the negative input.

4: Some bits may not be present on all devices. Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet for availability.

Register 18-5: AD1GIRQEN1: ADC1 Global Interrupt Enable Register 1

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AGIEN31	AGIEN30	AGIEN29	AGIEN28	AGIEN27	AGIEN26	AGIEN25	AGIEN24
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AGIEN23	AGIEN22	AGIEN21	AGIEN20	AGIEN19	AGIEN18	AGIEN17	AGIEN16
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AGIEN15	AGIEN14	AGIEN13	AGIEN12	AGIEN11	AGIEN10	AGIEN9	AGIEN8
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0

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 31-0 **AGIENx:** Global ADC Interrupt Enable bits[(1,2,3)]

`1` = A data ready event (transition from 0 to 1 of the ARDYx bit) will generate a Global ADC interrupt

`0` = No global interrupt is generated on a data ready event for this input

Note 1: These bits control the propagation of the ready events of one or more analog inputs to the single Global (group) interrupt. They do not affect the assertion of the individual interrupts for the data ready events. Both the individual and single global interrupt are enabled by setting the appropriate bits in the IECx register. Refer to the **“Interrupt Controller”** chapter in the specific device data sheet for details.

2: AGIENx = ANx, where ‘x’ = 0-31.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-6: AD1GIRQEN2: ADC1 Interrupt Enable Register 2

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	AGIEN44	AGIEN43	AGIEN42	AGIEN41	AGIEN40
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AGIEN39	AGIEN38	AGIEN37	AGIEN36	AGIEN35	AGIEN34	AGIEN33	AGIEN32

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 31-13 **Unimplemented:** Read as ‘0’

bit 12-0 **AGIENx:** Global ADC Interrupt Enable bits(1,2,3)

1 = A data ready event (transition from 0 to 1 of the ARDYx bit) will generate a Global ADC interrupt
0 = No global interrupt is generated on a data ready event

The Global ADC Interrupt is enabled by setting a bit in the IECx registers (refer to the **“Interrupt Controller”** chapter in the specific device data sheet for details).

Note 1: These bits control the propagation of the ready events of one or more analog inputs to the single Global (group) interrupt. They do not affect the assertion of the individual interrupts for the data ready events. Both the individual and single global interrupt are enabled by setting the appropriate bits in the IECx register. Refer to the **“Interrupt Controller”** chapter in the specific device data sheet for details.

2: AGIENx = ANx, where ‘x’ = 32-44.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-7: AD1CSS1: ADC1 Input Scan Select Register 1

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0

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 31-0 **CSSx:** ADC Input Scan Select bits[1,2,3]
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

Note 1: CSSx = ANx, where ‘x’ = 0-31.

2: In addition to setting the appropriate bits in this register, Class 1 and Class 2 analog inputs must select the STRIG input as the trigger source if they are to be scanned through the CSSx bits. Refer to the bit descriptions in the AD1TRGn register (Register 18-15) for selecting the STRIG option.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-8: AD1CSS2: ADC1 Input Scan Select Register 2

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	CSS44	CSS43	CSS42	CSS41	CSS40
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CSS39	CSS38	CSS37	CSS36	CSS35	CSS34	CSS33	CSS32

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 31-13 **Unimplemented:** Read as ‘`0`’

bit 12-0 **CSSx:** ADC Input Scan Select bits[1,2,3]
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

Note 1: CSSx = ANx, where ‘x’ = 32-44.

2: In addition to setting the appropriate bits in this register, Class 1 and Class 2 analog inputs must select the STRIG input as the trigger source if they are to be scanned through the CSSx bits. Refer to the bit descriptions in the AD1TRGn register (Register 18-15) for selecting the STRIG option.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-9: AD1DSTAT1: ADC1 Data Ready Status Register 1

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	ARDY31	ARDY30	ARDY29	ARDY28	ARDY27	ARDY26	ARDY25	ARDY24
23:16	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	ARDY23	ARDY22	ARDY21	ARDY20	ARDY19	ARDY18	ARDY17	ARDY16
15:8	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	ARDY15	ARDY14	ARDY13	ARDY12	ARDY11	ARDY10	ARDY9	ARDY8
7:0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0

Legend:	HS = Set by Hardware	HC = Cleared by Hardware	
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 31-0 **ARDYx:** Conversion Data Ready for Corresponding Analog Input Ready bits[1,2,3]

1 = This bit is set when data is ready in the buffer. An interrupt will be generated if the appropriate bit in the IECx register is set or if enabled for the ADC Global interrupt in the AD1GIRQEN register.

0 = This bit is cleared when the associated data register is read

Note 1: ARDYx = ANx, where ‘x’ = 0-31.

2: When the alternate inputs are used for the Class 1 inputs, the ready bit for the primary input will indicate the data ready status.

3: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-10: AD1DSTAT2: ADC1 Data Ready Status Register 2

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	—	—	—	ARDY44	ARDY43	ARDY42	ARDY41	ARDY40
7:0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	ARDY39	ARDY38	ARDY37	ARDY36	ARDY35	ARDY34	ARDY33	ARDY32

Legend:	HS = Set by Hardware	HC = Cleared by Hardware	
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 31-13 **Unimplemented:** Read as ‘0’

bit 12-0 **ARDYx:** Conversion Data Ready for Corresponding Analog Input Ready bits[1,2]

1 = This bit is set when data is ready in the buffer. An interrupt will be generated if the appropriate bit in the IECx register is set or if enabled for the ADC Global interrupt in the AD1GIRQEN register.

0 = This bit is cleared when the associated data register is read

Note 1: ARDYx = ANx, where ‘x’ = 32-44.

2: Selections vary by device. Refer to the **“12-bit Pipelined Analog-to-Digital Converter (ADC)”** chapter in the specific device data sheet to determine which selections are available.

Register 18-11: AD1CMPCONn: ADC1 Digital Comparator Control Register ‘n’ (‘n’ = 1, 2, 3, 4, 5, or 6)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	—	—	—	AINID<4:0>(1)				
7:0	R/W-0	R/W-0	HS, HC, R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ENDCMP	DCMPGIEN(2)	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO

Legend:	HS = Set by Hardware	HC = Cleared by Hardware	
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 31-13 **Unimplemented:** Read as ‘0’

bit 12-8 **AINID<4:0>:** Analog Input Identification (ID) bits(1)

When a digital comparator event occurs (DCMPED = 1), these bits identify the analog input which generated the event. In addition, these bits identify the last analog input to generate an event if multiple inputs are enabled and more than one event has occurred since the last reading.

bit 7 **ENDCMP:** Digital Comparator Enable bit

1 = Digital Comparator is enabled
0 = Digital Comparator is not enabled, and the DCMPED status bit is cleared

bit 6 **DCMPGIEN:** Digital Comparator Global ADC Interrupt Enable bit(2)

1 = A Digital Comparator Event (DCMPED transitions from ‘0’ to ‘1’) will generate a Global ADC interrupt.
0 = A Digital Comparator Event will not generate a Global ADC interrupt.

The Global ADC Interrupt is enabled by setting a bit in the IECx registers (refer to the **“Interrupt Controller”** chapter in the specific device data sheet for details).

bit 5 **DCMPED:** Digital Comparator Event Detected Status bit

1 = This bit is set by the digital comparator hardware when a comparison event is detected. An interrupt will be generated if the appropriate bit in the IECx register is set or if enabled for the ADC Global interrupt in the DCMPGIEN bit.
0 = This bit is cleared by reading the AINID<4:0> bits or when the ADC module is disabled

bit 4 **IEBTWN:** Between Low/High Digital Comparator Event bit(2)

1 = Generate a digital comparator event when ADCMPLO<15:0> ≤ DATA<31:0> < ADCMPHI<15:0>
0 = Do not generate a digital comparator event

bit 3 **IEHIHI:** High/High Digital Comparator Event bit(2)

1 = Generate a Digital Comparator Event when ADCMPHI<15:0> ≤ DATA<31:0>
0 = Do not generate a digital comparator event

bit 2 **IEHILO:** High/Low Digital Comparator Event bit(2)

1 = Generate a Digital Comparator Event when DATA<31:0> < ADCMPHI<15:0>
0 = Do not generate a digital comparator event

bit 1 **IELOHI:** Low/High Digital Comparator Event bit(2)

1 = Generate a Digital Comparator Event when ADCMPLO<15:0> ≤ DATA<31:0>
0 = Do not generate a digital comparator event

bit 0 **IELOLO:** Low/Low Digital Comparator Event bit(2)

1 = Generate a Digital Comparator Event when DATA<31:0> < ADCMPLO<15:0>
0 = Do not generate a digital comparator event

Note 1: A Digital Comparator Event occurred on ANx, where ‘x’ = AINID<4:0>, and ‘x’ has a range of 0-31.

2: Changing these bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

Register 18-12: AD1CMPENn: ADC1 Digital Comparator Enable Register 'n' ('n' = 1, 2, 3, 4, 5, or 6)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CMPE31	CMPE30	CMPE29	CMPE28	CMPE27	CMPE26	CMPE25	CMPE24
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CMPE23	CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0

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 31-0 **CMPE31:CMPE0:** ADC1 Digital Comparator Enable bits(1,2)

These bits enable conversion results corresponding to the Analog Input to be processed by the Digital Comparator. CMPE0 enables AN0, CMPE1 enables AN1, and so on.

Note 1: CMPEx = ANx, where 'x' = 0-31 (Digital Comparator inputs are limited to AN0 through AN31).

2: Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = `1`) can result in unpredictable behavior.

Register 18-13: AD1CMPn: ADC1 Digital Comparator Register ‘n’ (‘n’ = 1, 2, 3, 4, 5 or 6)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCMPHI<15:8>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCMPHI<7:0>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCMPLO<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCMPLO<7:0>							

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 31-16 **ADCMPHI<15:0>:** Digital Analog Comparator High Limit Value bits

These bits store the high limit value, which is used for comparisons with the analog-to-digital conversion data. The user is responsible for formatting the high limit value to match the format as specified by the SHxMOD<1:0> bits for the associated S&H and the global FRACT bit. These bits are located in the ADxMOD and ADxCON1 registers, respectively.

bit 15-0 **ADCMPLO<15:0>:** Digital Analog Comparator Low Limit Value bits

These bits store the low limit value, which is used for comparisons with the analog-to-digital conversion data. The user is responsible for formatting the low limit value to match the format as specified by the SHxMOD<1:0> bits for the associated S&H and the global FRACT bit. These bits are located in the ADxMOD and ADxCON1 registers, respectively.

Note: Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

Register 18-14: AD1FLTRn: ADC1 Filter Register 'n' ('n' = 1, 2, 3, 4, 5, or 6)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AFEN	—	—	OVRSAM<2:0>			AFGIEN	AFRDY
23:16	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	CHNLID<5:0>					
15:8	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	FLTRDATA<15:8>							
7:0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
	FLTRDATA<7:0>							

Legend:	HS = Set by Hardware	HC = Cleared by Hardware	
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 31 **AFEN:** Oversampling Filter Enable bit
1 = Oversampling filter is enabled
0 = Oversampling filter is disabled and the AFRDY bit is cleared

bit 30-29 **Unimplemented:** Read as '0'

bit 28-26 **OVRSAM<2:0>:** Oversampling Filter Ratio bits
111 = 128x (shift sum 3 bits to right, output data is in 15.1 format)
110 = 32x (shift sum 2 bits to right, output data is in 14.1 format)
101 = 8x (shift sum 1 bit to right, output data is in 13.1 format)
100 = 2x (shift sum 0 bits to right, output data is in 12.1 format)
011 = 256x (shift sum 4 bits to right, output data is 16 bits)
010 = 64x (shift sum 3 bits to right, output data is 15 bits)
001 = 16x (shift sum 2 bits to right, output data is 14 bits)
000 = 4x (shift sum 1 bit to right, output data is 13 bits)

bit 25 **AFGIEN:** Oversampling Filter Global ADC Interrupt Enable bit
1 = An Oversampling Filter Data Ready event (AFRDY transitions from '0' to '1') will generate an ADC Global Interrupt
0 = An Oversampling Filter Data Ready event will not generate an ADC Global Interrupt

bit 24 **AFRDY:** Oversampling Filter Data Ready Flag bit
1 = This bit is set when data is ready in the FLTRDATA<15:0> bits
0 = This bit is cleared when FLTRDATA<15:0> is read, or if the module is disabled

bit 23-22 **Unimplemented:** Read as '0'

bit 21-16 **CHNLID<5:0>:** Channel ID Selection bits[1]
These bits specify the analog input to be used as the oversampling filter data source.
111111 = Reserved
•
•
•
101101 = Reserved
101100 = IVTEMP
101011 = IVREF
101010 = AN42
•
•
•
000010 = AN2
000001 = AN1
000000 = AN0

bit 15-0 **FLTRDATA<15:0>:** Oversampling Filter Data Output Value bits
These bits contain the oversampling filter result.

Note 1: Selections vary by device. Refer to the **"12-bit Pipelined Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet to determine which selections are available

Register 18-15: AD1TRGR1: ADC1 Input Convert Control Register 1

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC3<4:0>				
23:16	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC2<4:0>				
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC1<4:0>				
7:0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC0<4:0>				

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 31-29 **Unimplemented:** Read as ‘0’

bit 28-24 **TRGSRC3<4:0>:** Trigger Source for Conversion of Analog Channels AN3 Select bits

11111 = Reserved
•
•
•
01101 = Reserved
01100 = Comparator 2 COUT
01011 = Comparator 1 COUT
01010 = OCMP5
01001 = OCMP3
01000 = OCMP1
00111 = TMR5 match
00110 = TMR3 match
00101 = TMR1 match
00100 = INT0
00011 = STRIG[(1)]
00010 = Reserved
00001 = Global software trigger (GSWTRG)
00000 = No trigger

bit 23-21 **Unimplemented:** Read as ‘0’

bit 20-16 **TRGSRC2<4:0>:** Trigger Source for Conversion of Analog Channels AN2 Select bits
See bits 28-24 for bit value definitions.

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

bit 12-8 **TRGSRC1<4:0>:** Trigger Source for Conversion of Analog Channels AN1 Select bits
See bits 28-24 for bit value definitions.

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

bit 4-0 **TRGSRC0<4:0>:** Trigger Source for Conversion of Analog Channels AN0 Select bits
See bits 28-24 for bit value definitions.

Note 1: Using STRIG as the trigger source specifies this input to use the Scan Trigger source for its trigger. The STRGSRC<4:0> bits (AD1CON1<26:22>), as well as the appropriate CSSx bit(s) in the AD1CSS1 and AD1CSS2 registers must be set for proper scan operation.

Register 18-16: AD1TRGR2: ADC1 Input Convert Control Register 2

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC7<4:0>				
23:16	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC6<4:0>				
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC5<4:0>				
7:0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC4<4:0>				

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 31-29 **Unimplemented:** Read as ‘0’

bit 28-24 **TRGSRC7<4:0>:** Trigger Source for Conversion of Analog Channels AN7 Select bits

11111 = Reserved
•
•
•
01101 = Reserved
01100 = C2OUT
01011 = C1OUT
01010 = OCMP5
01001 = OCMP3
01000 = OCMP1
00111 = TMR5 match
00110 = TMR3 match
00101 = TMR1 match
00100 = INT0
00011 = STRIG(1)
00010 = Reserved
00001 = Global software trigger (GSWTRG)
00000 = No trigger

bit 23-21 **Unimplemented:** Read as ‘0’

bit 20-16 **TRGSRC6<4:0>:** Trigger Source for Conversion of Analog Channels AN6 Select bits
See bits 28-24 for bit value definitions.

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

bit 12-8 **TRGSRC5<4:0>:** Trigger Source for Conversion of Analog Channels AN5 Select bits
See bits 28-24 for bit value definitions.

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

bit 4-0 **TRGSRC4<4:0>:** Trigger Source for Conversion of Analog Channels AN4 Select bits
See bits 28-24 for bit value definitions.

Note 1: Using STRIG as the trigger source specifies this input to use the Scan Trigger source for its trigger. The STRGSRC<4:0> bits (AD1CON1<26:22>), as well as the appropriate CSSx bit(s) in the AD1CSS1 and AD1CSS2 registers must be set for proper scan operation.

Register 18-17: AD1TRGR3: ADC1 Input Convert Control Register 3

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC11<4:0>				
23:16	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC10<4:0>				
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC9<4:0>				
7:0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	TRGSRC8<4:0>				

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 31-29 **Unimplemented:** Read as ‘0’

bit 28-24 **TRGSRC11<4:0>:** Trigger Source for Conversion of Analog Channels AN11 Select bits

11111 = Reserved
•
•
•
01101 = Reserved
01100 = C2OUT
01011 = C1OUT
01010 = OCMP5
01001 = OCMP3
01000 = OCMP1
00111 = TMR5 match
00110 = TMR3 match
00101 = TMR1 match
00100 = INT0
00011 = STRIG[(1)]
00010 = Reserved
00001 = Global software trigger (GSWTRG)
00000 = No trigger

bit 23-21 **Unimplemented:** Read as ‘0’

bit 20-16 **TRGSRC10<4:0>:** Trigger Source for Conversion of Analog Channels AN10 Select bits
See bits 28-24 for bit value definitions.

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

bit 12-8 **TRGSRC9<4:0>:** Trigger Source for Conversion of Analog Channels AN9 Select bits
See bits 28-24 for bit value definitions.

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

bit 4-0 **TRGSRC8<4:0>:** Trigger Source for Conversion of Analog Channels AN8 Select bits
See bits 28-24 for bit value definitions.

Note 1: Using STRIG as the trigger source specifies this input to use the Scan Trigger source for its trigger. The STRGSRC<4:0> bits (AD1CON1<26:22>), as well as the appropriate CSSx bit(s) in the AD1CSS1 and AD1CSS2 registers must be set for proper scan operation.

Register 18-18: AD1DATAn: ADC1 Data Output Register ('n' = 0 to 44)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DATA<31:24>							
23:16	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DATA<23:16>							
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DATA<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DATA<7:0>							

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 31-0 **DATA<31:0>:** Data Output Value bits

These bits are formatted as specified by the SHxMOD<1:0> bits for the FRACT bit and the associated S&H circuit.

Note: When an alternate input is used as the input source for a Class 1 input, the data output is still read from the Primary input Data Output Register.

Register 18-19: AD1CALx: ADC1 Calibration Register ('x' = 1-5)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCAL<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCAL<23:16>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCAL<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ADCAL<7:0>							

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 31-0 **ADCAL<31:0>:** Calibration Data for the ADC module bits

18.3 ADC OPERATION, TERMINOLOGY AND CONVERSION SEQUENCE

Analog-to-Digital conversion using the 12-bit ADC involves the following three steps:

1. Sampling of the input signal.
2. Capture of the input signal (holding) and transfer to the converter.
3. Conversion of the analog signal to its digital representation.

Sampling of the input signal involves charging of the capacitor in the S&H circuit. The sampling time must be adequate so that the capacitor charges to a value equal to the input voltage (see **18.10 "ADC Sampling Requirements"**). At the appropriate time, the input is disconnected from the capacitor and subsequently, the analog voltage is transferred to the converter. The converter then digitizes the analog signal and provides the result.

The converter requires a clock source and a reference voltage. The clock is referred to as the ADC clock and has a period of TAD. The clock and reference voltage sources are selectable, as well as the clock prescaling.

The 12-bit ADC uses two types of S&H circuits: dedicated and shared. Each ADC implementation includes up to five dedicated S&H circuits and a single shared S&H. Inputs connected to the S&H circuits are categorized into three types: Class 1, Class 2, and Class 3.

18.3.1 Analog Inputs

The analog input pins that are available on a device are classified into three categories: Class 1, Class 2, and Class 3. Each analog input, regardless of Class category, has its own unique data output register where the conversion result is stored.

Table 18-2: Analog Input Types

Type	S&H/Input Type	Trigger	Trigger Action
Class 1	Dedicated S&H input	Individual trigger source or channel scan trigger	Ends sampling and starts conversion
Class 2	Shared S&H input	Individual trigger source or channel scan trigger	Starts sampling/conversion sequence or begins channel scan sequence
Class 3	Shared S&H channel scan input	Channel scan trigger	Starts channel scan sequence

Class 1 inputs are associated with a dedicated S&H circuit. Each dedicated S&H has a single Class 1 input associated with it. Each Class 1 input has a unique trigger selection register.

Class 1 inputs can be part of a channel scan list, triggered by the common scan trigger source.

Class 2 inputs are used on the shared S&H, either individually triggered or as part of a channel scan list. When used individually they are triggered by their unique trigger register.

Class 3 inputs are used for channel scan exclusively. They share a common trigger source. When using channel scan it is possible to combine Class 1, Class 2, and Class 3 inputs in the scan list.

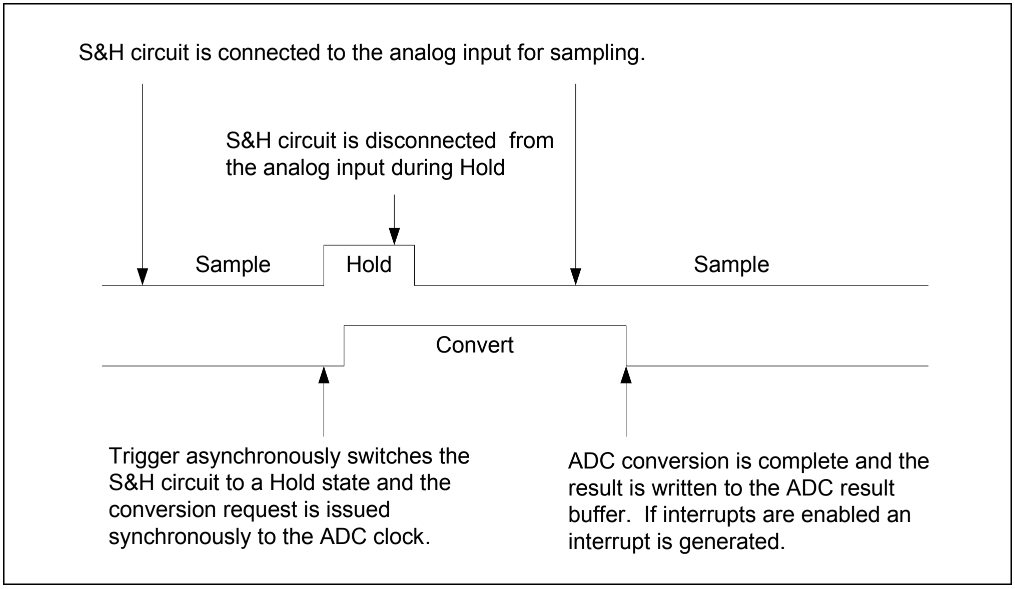
> **Note:** Each Class 1 input also has an alternate selection. This alternate input is unique from Class 1, 2, and 3 inputs. When an alternate input is used for a Class 1 input, the trigger is still selected by the Class 1 trigger register, the result data is stored in the Class 1 output register and the priority for conversion remains that of the Class 1 input.

18.3.2 Dedicated Sample and Hold

Dedicated S&H circuits, as the name implies, are dedicated to a Class 1 analog input, and enable high-speed, high-precision sampling and conversion, and are especially useful for capturing time sensitive or transient signals. They can be configured for Single-ended or Differential modes.

Each dedicated S&H continuously tracks the input signal in Sample mode until the asynchronous trigger event occurs. The trigger event causes the S&H to immediately stop sampling and enter the holding state. It is important to note that the trigger event, which ends sampling, occurs asynchronously to the ADC clock. While the S&H circuits enter the Hold state immediately, the asynchronous trigger must be synchronized to the ADC clock consuming two ADC clock edges before the conversion request is issued to the pipeline converter. If no higher priority conversion requests exits, the conversion will commence immediately; otherwise, the conversions take place by priority. The S&H remains in the holding state during the initial stage of the conversion process. When the conversion is complete, data is transferred to the result buffer and an interrupt is generated.

Figure 18-4: Dedicated S&H Conversion Sequence


S&H circuit is connected to the analog input for sampling.
S&H circuit is disconnected from the analog input during Hold
Sample
Hold
Sample
Convert
Trigger asynchronously switches the S&H circuit to a Hold state and the conversion request is issued synchronously to the ADC clock.
ADC conversion is complete and the result is written to the ADC result buffer. If interrupts are enabled an interrupt is generated.

If using a periodic trigger source with a dedicated S&H, the total sampling time is determined by the trigger rate. The trigger rate must not violate the necessary sampling time. See **18.10 "ADC Sampling Requirements"** for more information.

Note 1: There is no mechanism in the dedicated S&H to ensure the minimum sampling time has been met. The system designer must ensure that the S&H has had adequate sampling time before triggering.

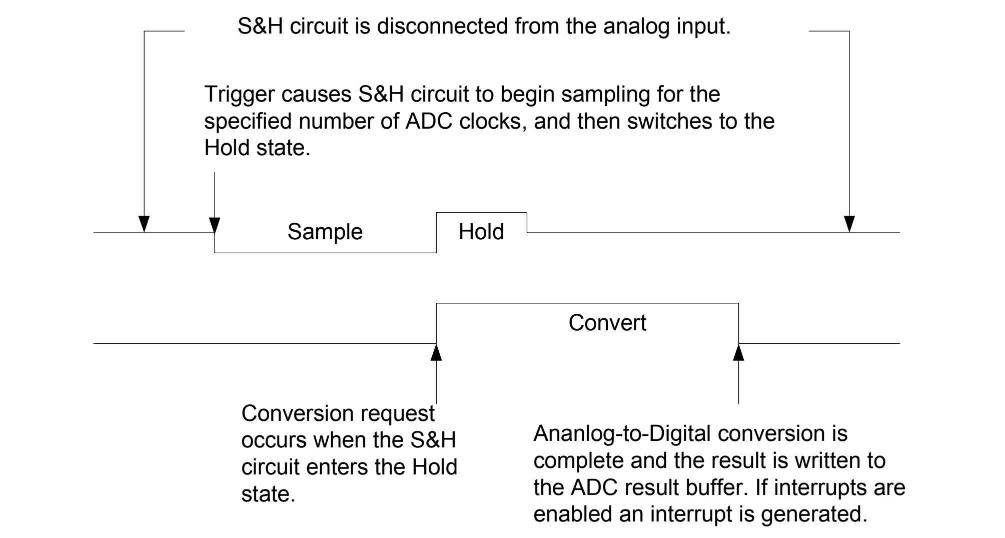
2: The alternate input provided on dedicated S&H circuits provides an alternate physical connection; however, the dedicated S&H always uses the trigger configuration and data result registers for its primary Class 1 input.

18.3.3 Shared Sample and Hold

The shared S&H circuit provides the greatest flexibility for handling a large number of Class 2 and Class 3 inputs where timing is not critical. Shared S&H circuits can also be configured for Single-ended or Differential modes; however, in Differential mode the single negative input is common to all positive input selections.

The shared S&H is shared among the majority of inputs using a multiplexer on the positive input. Unlike the dedicated S&H, the trigger event of a shared S&H starts the sampling process using a specified sample time. Once the signal has been sampled the specified number of ADC clocks, the S&H enters the hold state and the conversion request is issued. The S&H remains in the hold state during the initial stage of the conversion process.

Figure 18-5: Shared S&H Conversion Sequence


S&H circuit is disconnected from the analog input.
Trigger causes S&H circuit to begin sampling for the specified number of ADC clocks, and then switches to the Hold state.
Sample
Hold
Convert
Conversion request occurs when the S&H circuit enters the Hold state.
Ananlog-to-Digital conversion is complete and the result is written to the ADC result buffer. If interrupts are enabled an interrupt is generated.

When using the shared S&H, the SAMC<7:0> bits in the ADC1 Control Register 2 (AD1CON2<23:16>) determine the sample time. When periodically triggering a single input on the shared S&H, the trigger rate should not be less than the sample time plus 2 TAD, which is the time to transfer to the pipeline converter.

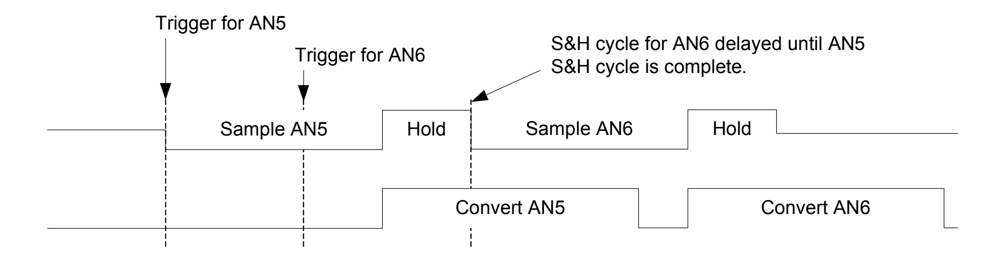
There are two operation modes for the shared S&H: Independent Class 2 Triggering and Channel Scanning. The shared S&H accommodate either or both of these modes simultaneously. There is no assurance that a Class 2 or Class 3 conversion request will be serviced immediately. Conversion requests to the pipeline converter are serviced in order or priority.

18.3.3.1 CLASS 2 TRIGGERING

When Class 2 inputs are defined with independent triggers, they are sampled and converted by the shared S&H using the sequence illustrated in Figure 18-5. When multiple Class 2 inputs with independent triggers are used it is important to understand the consequences of trigger timing.

If a conversion is underway and another Class 2 trigger occurs then the sample-hold-conversion for the new trigger will be stalled until the in-process sample-hold cycle is complete, as shown in Figure 18-6.

Figure 18-6: Multiple Independent Class 2 Trigger Conversion Sequence



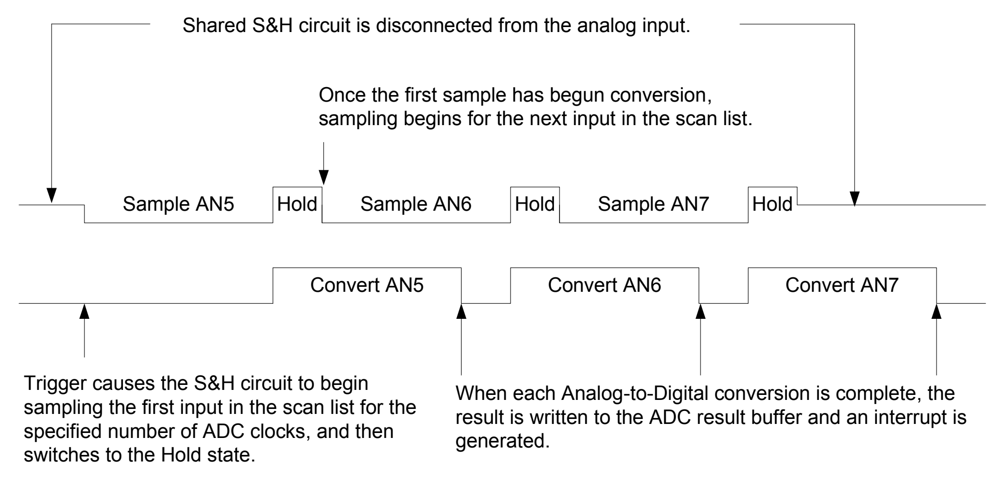
When multiple inputs to the shared S&H are triggered simultaneously, the processing order is determined by their natural priority (the lowest numbered input has the highest priority). As an example, if AN5, AN6, and AN7 are triggered using the same source, AN5 will be sampled and converted first, followed by AN6 and finally, AN7.

When using the independent Class 2 triggering on the shared S&H, the SAMC<7:0> bits (AD1CON2<23:16>) determine the sample time for all inputs while the appropriate TRGSRC<4:0> bits in the ADC1 Input Convert Control Register (see Register 18-15) determine the trigger source for each input.

18.3.3.2 CHANNEL SCAN

Channel scanning is a feature that allows an automated scanning sequence of multiple Class 1, Class 2 or Class 3 inputs. All Class 2 and Class 3 inputs are scanned using the single shared S&H. Class 1 inputs are scanned using their dedicated S&H. A single trigger source initiates a channel scan. Individual triggers, if available, are not used. When a trigger occurs, all Class 1 inputs are captured simultaneously and conversions are started based on natural priority. The trigger will also initiate sampling of the first Class 2 or 3 input in the scan list and once sampling is complete, pass it to the converter after all of the Class 1 conversions have begun. The natural input order is used; lower number inputs are sampled before higher number inputs.

Figure 18-7: Channel Scan Conversion Sequence for Three Class 2 Inputs

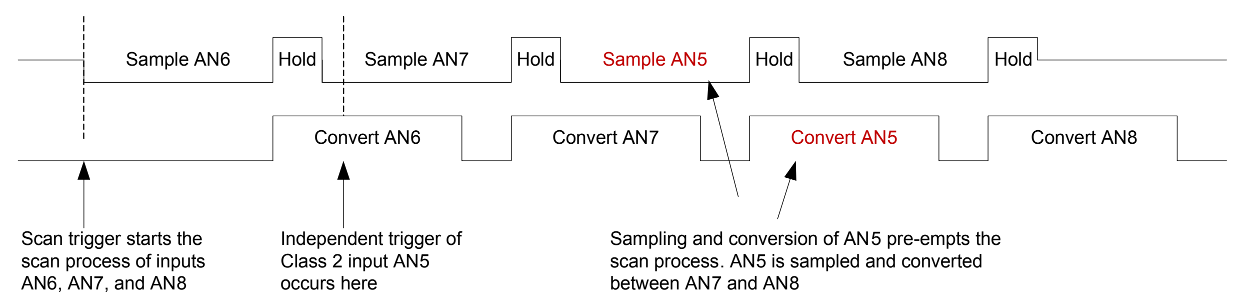

Shared S&H circuit is disconnected from the analog input.
Once the first sample has begun conversion, sampling begins for the next input in the scan list.
Sample AN5
Hold
Sample AN6
Hold
Sample AN7
Hold
Convert AN5
Convert AN6
Convert AN7
Trigger causes the S&H circuit to begin sampling the first input in the scan list for the specified number of ADC clocks, and then switches to the Hold state.
When each Analog-to-Digital conversion is complete, the result is written to the ADC result buffer and an interrupt is generated.

When using the shared S&H in scan mode, the SAMC<7:0> bits in the ADC1 Control Register 2 (AD1CON2<23:16>) determine the sample time for all inputs while the Scan Trigger Source Selection bits (STRGSRC<4:0>) in the ADC1 Control Register 1 (AD1CON1<26:22>), determine the trigger source.

To ensure predicable results, a scan should not be retriggered until sampling of all channels has completed. Care should be taken in the system design to preclude retriggering a channel scan while a scan is in progress.

Individual Class 2 triggers that occur during a channel scan will pre-empt the channel scan sequence if they are a higher priority than the sample currently being processed. In Figure 18-8, a channel scan of AN6, AN7, and AN8 is underway when an independent trigger of Class 2 input AN5 takes place. The channel scan is interrupted for the sampling and conversion of AN5.

Figure 18-8: Channel Scan Conversion Pre-empted by Class 2 Input Trigger


Sample AN6
Hold
Sample AN7
Hold
Sample AN5
Hold
Sample AN8
Hold
Convert AN6
Convert AN7
Convert AN5
Convert AN8
Scan trigger starts the scan process of inputs AN6, AN7, and AN8
Independent trigger of Class 2 input AN5 occurs here
Sampling and conversion of AN5 pre-empts the scan process. AN5 is sampled and converted between AN7 and AN8

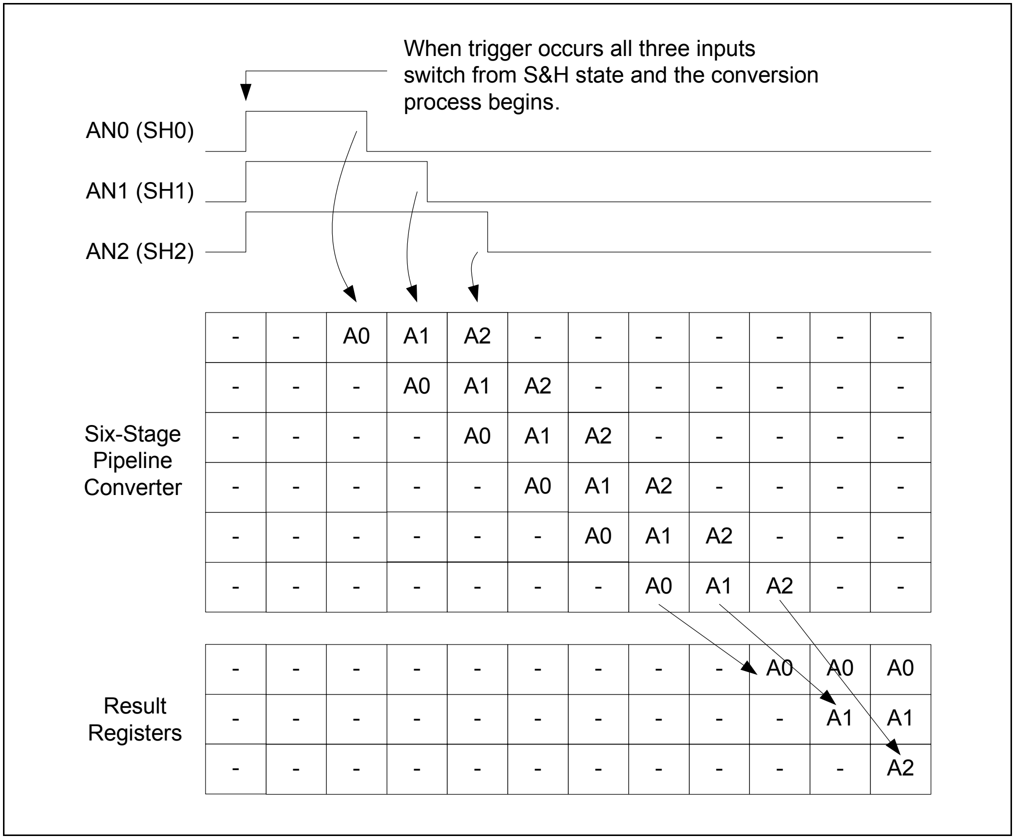
18.3.4 Six-Stage Pipeline Converter

The 12-bit ADC incorporates a six-stage pipeline converter architecture. This allows for simultaneous conversion of up to six samples, resulting in lower latencies when multiple samples need conversion. The conversion time for a sample is determined by the ADC clock rate.

When multiple inputs are ready to be converted at the same time, as would be the case when simultaneously sampling, the converter accepts the signals based on their natural priority. Lower priority inputs are blocked for pending higher priority inputs.

As shown in Figure 18-9, AN0 is used with S&H0, AN1 with S&H1, and AN2 with S&H2. All three inputs are configured to use the same trigger for simultaneous sampling. Since the natural priority is used, AN0 is transferred to the six-stage converter first, followed by AN1, and finally AN2. The S&H is held in the hold state until its sample has been passed to the converter, and then automatically resumes sampling.

Figure 18-9: Pipeline Converter (Conversion Process)



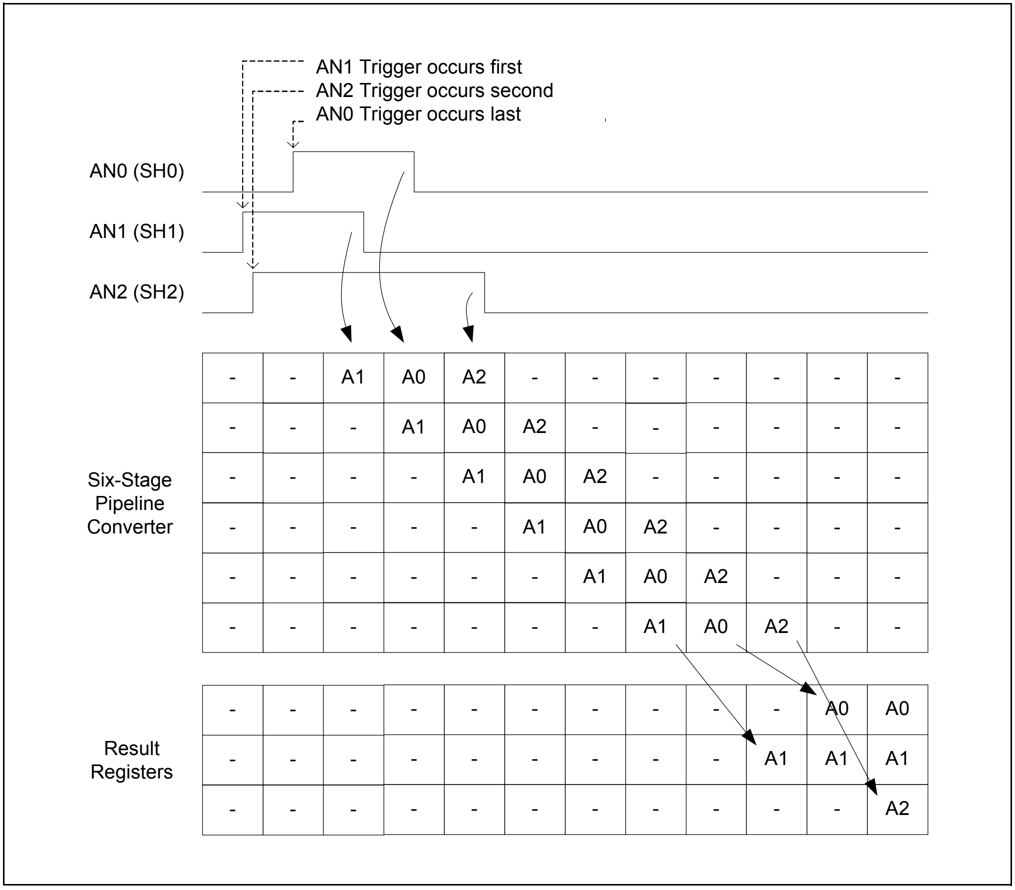
When trigger occurs all three inputs switch from S&H state and the conversion process begins.

Six-Stage Pipeline Converter	-	-	A0	A1	A2	-	-	-	-	-	-	-
	-	-	-	A0	A1	A2	-	-	-	-	-	-
	-	-	-	-	A0	A1	A2	-	-	-	-	-
	-	-	-	-	-	A0	A1	A2	-	-	-	-
	-	-	-	-	-	-	A0	A1	A2	-	-	-
	-	-	-	-	-	-	-	A0	A1	A2	-	-

Result Registers	-	-	-	-	-	-	-	-	-	A0	A0	A0
	-	-	-	-	-	-	-	-	-	-	A1	A1
	-	-	-	-	-	-	-	-	-	-	-	A2

The natural priority for input conversion is illustrated in Figure 18-10. Again, AN0 is used with S&H0, AN1 with S&H1, and AN2 with S&H2; however, each input is triggered individually. AN1 is triggered first followed by AN2, and finally AN0. The triggers for AN2 and AN0 occur while AN1 is being transferred into the converter pipeline. When the transfer for AN1 is complete, AN0 is moved into the converter pipeline even though the trigger for AN2 occurred before AN0. This is because AN0 has a higher natural priority than AN2.

Figure 18-10: Pipeline Converter (Effects of Natural Priority on the Conversion Process)



Six-Stage Pipeline Converter

-	-	A1	A0	A2	-	-	-	-	-	-	-
-	-	-	A1	A0	A2	-	-	-	-	-	-
-	-	-	-	A1	A0	A2	-	-	-	-	-
-	-	-	-	-	A1	A0	A2	-	-	-	-
-	-	-	-	-	-	A1	A0	A2	-	-	-
-	-	-	-	-	-	-	A1	A0	A2	-	-

Result Registers

-	-	-	-	-	-	-	-	-	-	A0	A0
-	-	-	-	-	-	-	-	-	A1	A1	A1
-	-	-	-	-	-	-	-	-	-	-	A2

18.3.5 Sample and Conversion Sequences

The combination of S&H and the pipeline converter establish a sample and conversion sequence and a resulting sample and conversion time. These sequences vary based on the type of S&H (dedicated or shared) and the operation mode. Basic operation is described in subsequent sections.

18.3.5.1 PIPELINE CONVERSION TIME

Conversion time is the time from the S&H entering the hold state until the availability of data in the result register. During this time, the sample is transferred from the S&H circuit to the pipeline converter. If there are no other samples pending conversion, this time is determinate. Since there is only a single pipeline converter and multiple S&H circuits, if S&H circuits are pending transfer to the converter, the sample must wait for availability of the first converter stage. As previously described, natural priority determines the order in which pending samples are transferred to the initial pipeline converter stage.

18.3.5.2 DEDICATED S&H CONVERSION TIME (CLASS 1 INPUTS)

As previously described, triggering a Class 1 input causes the dedicated S&H to enter the hold state. The conversion starts two clock periods following the ADC clock period in which the asynchronous trigger occurred. The time required from trigger until data is ready and interrupt occurs, is defined in Equation 18-1.

Equation 18-1: Dedicated S&H Conversion Time

$$T_{DCONV} = 10 \cdot T_{AD}$$

Where:

T_{AD} = Analog-to-digital conversion clock period

Note: This equation assumes there are no pending conversions. Each pending higher priority conversion will add one TAD to the conversion time.

18.3.5.3 DEDICATED S&H SAMPLE TIME (CLASS 1 INPUTS)

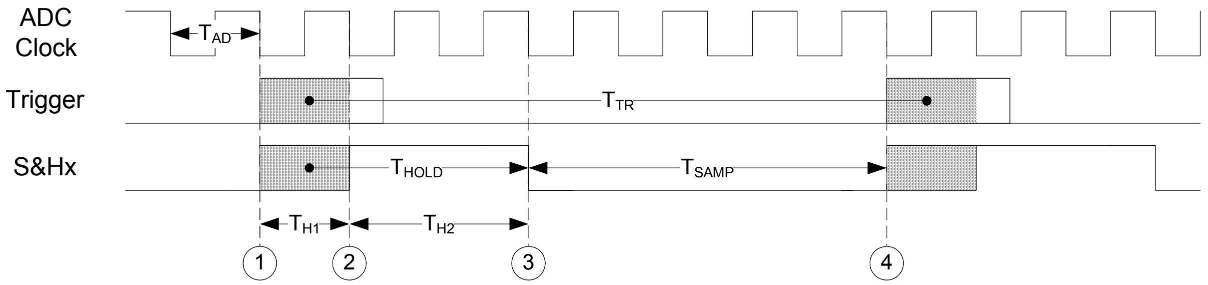
The Dedicated S&H circuits track their inputs until triggered, which starts a conversion sequence. The trigger rate establishes the sampling time. This is different from the shared S&H circuit where the trigger connects the input to the S&H, and then starts a sample period followed by a conversion sequence.

Figure 18-11 shows a simple example for a single Class 1 input that is being triggered by a periodic source.

The sampling time is the difference between the trigger rate, TTR, and the hold time, THOLD. The input can be immediately retriggered once the S&H leaves the hold state, as indicated by number three in Figure 18-11, making it possible to trigger at a rate that violates that minimum sampling time requirements. There is no mechanism built into the control logic to ensure that the minimum sampling requirements for dedicated S&H Class 1 inputs are met.

Note: When triggering Class 1 inputs, the trigger rate must allow for minimum sample time requirements to be met.

Figure 18-11: Dedicated Sample and Hold Timing



① At the time of the trigger, the dedicated S&H circuit disconnects from the input pin and switches to a hold state.

② The switch to the hold state occurs asynchronously from the ADC clock. The hold state persists for up to one clock cycle, while the ADC control logic synchronizes to the ADC clock.

③ Two clock cycles are required to transfer to the converter, and then the S&H switches back to Sample mode and waits for the next trigger

④ The S&H tracks the input signal, sampling it until the next trigger occurs where the process repeats.

Equation 18-2 can be used to determine the sample time for Class 1 inputs.

Equation 18-2: Dedicated S&H Sample Time

$$T_{SAMP} = T_{TR} - T_{HOLD}$$

Where:

T_{TR} = Trigger rate time interval (as determined by the system design)

T_{HOLD} = $TH1$ + $TH2$

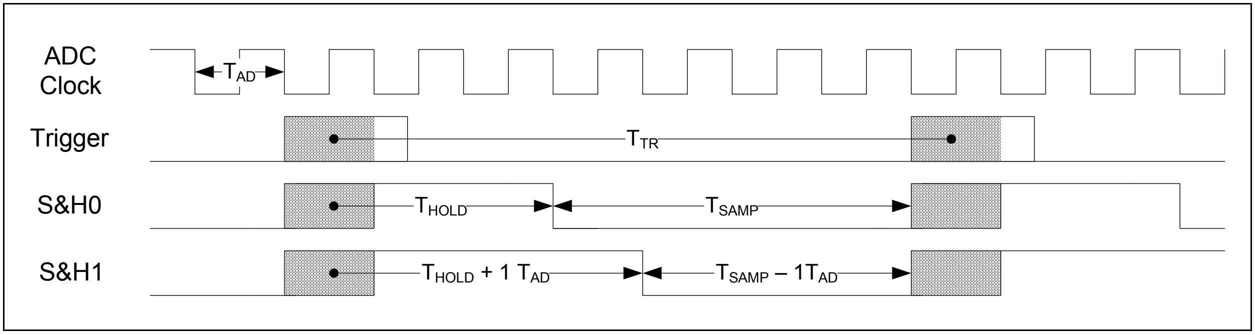
$TH1$ = 1 T_{AD} (clock synchronization)

$TH2$ = 2 T_{AD} (add 1 T_{AD} for each pending higher priority conversion)

Note: The system design must ensure that the trigger rate for Class 1 inputs does not violate the minimum sampling time requirements. Dedicated S&H Class 1 Inputs have no mechanism to establish sampling time. It is determined strictly by the trigger rate. $TH1$, the time allotted for clock synchronization, can be less than 1 T_{AD} depending on the relationship of the asynchronous trigger to the ADC clock. It is strongly recommended that 1 T_{AD} be allocated for $TH1$ in the Trigger period, T_{TR}, to ensure that the required sampling time is met regardless of the phase relationship of the trigger to ADC clock.

As noted in Equation 18-2, TH2 is increased by 1 TAD for each pending higher priority conversion. Simultaneously triggering multiple Class 1 Inputs will always create a pending higher priority conversion. Figure 18-12 shows the effect of an added pending conversion where the sample time for S&H 1 is shorter than S&H 0 by one TAD period. In these situations, it may become necessary to increase the trigger interval, TTR, so that the lowest priority input has adequate sample time.

Figure 18-12: Effects of Simultaneous Sampling Multiple Class 1 Inputs



18.3.5.4 SHARED S&H CONVERSION TIMES (CLASS 2 INPUTS)

As previously described, triggering a Class 2 input on a shared S&H circuit starts the sampling process. On completion of the sampling process the S&H will enter the hold state and the conversion will commence. The SAMC value determines the sample time. The time required from trigger until data is ready and interrupt occurs is defined in Equation 18-3.

Equation 18-3: Shared Sample and Hold Conversion Time

$$T_{SCONV} = T_{SSAM} + T_{CONV}$$

$$T_{SCONV} = (SAMC + 1) \cdot T_{AD} + 10 \cdot T_{AD}$$

$$T_{SCONV} = (SAMC + 11) \cdot T_{AD}$$

Where:

T_{AD} = Analog-to-digital conversion clock period

$SAMC$ = SAMC<7:0> bits (AD1CON2<23:16>)

Note: This equation assumes there are no pending conversions. Each pending higher priority Class 2 conversion will add one TAD to the conversion time plus the sample time (SAMC + 1) * TAD.

If there are higher priority Class 1 conversions pending at the completion of a shared S&H sampling period, each higher priority dedicated conversion will delay conversion of the shared S&H by one TAD.

18.3.5.5 CHANNEL SCAN CONVERSION TIME (CLASS 2 AND CLASS 3 INPUTS)

As previously described, the scan trigger starts the sequential sampling and conversion process of the Class 1, Class 2, or Class 3 inputs configured for channel scan. The total time to complete the channel scan is based on the ADC clock rate, the number of inputs being scanned, and the sampling time. For a scan list consisting of Class 2 and Class 3 inputs, the time required from trigger until the last scanned input's data is ready and interrupt occurs is defined in Equation 18-4.

Equation 18-4: Channel Scan Sample and Conversion Time

$$T_{SCAN} = (N \cdot (SAMC + 4) + 6) \cdot T_{AD}$$

Where:

N = Number of scanned channels

T_{AD} = Analog-to-digital conversion clock period

$SAMC$ = SAMC<7:0> bits (AD1CON2<23:16>)

Note: This equation assumes there are no pending conversions. Each pending higher priority Class 1 conversion will add one TAD to the conversion time. Each pending higher priority Class 2 conversion will add one TAD conversion time plus (SAMC + 1) * TAD sample time.

18.4 ADC MODULE CONFIGURATION

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

To configure the ADC module, perform the following steps:

1. Configure the analog port pins, as described in **18.4.1 "Configuring the Analog Port Pins"**.
2. Select the analog inputs to the ADC MUXs, as described in **18.4.2 "Selecting the ADC Multiplexer Analog Inputs"**.
3. Select the format of the ADC result, as described in **18.4.5 "Selecting the Format of the ADC Result"**.
4. Select the conversion trigger source, as described in **18.4.6 "Selecting the Conversion Trigger Source"**.
5. Select the voltage reference source, as described in **18.4.7 "Selecting the Voltage Reference Source"**.
6. Select the scanned inputs, as described in **18.4.8 "Selecting the Scanned Inputs"**.
7. Select the analog-to-digital conversion clock source and prescaler, as described in **18.4.9 "Selecting the Analog-to-Digital Conversion Clock Source and Prescaler"**.
8. Specify any additional acquisition time, if required, as described in **18.10 "ADC Sampling Requirements"**.
9. Turn on the ADC module, as described in **18.4.10 "Turning ON the ADC"**.
10. Perform ADC calibration, as described in **18.4.12 "ADC Calibration"**.
11. Configure the ADC interrupts (if required), as described in **18.6 "Interrupts"**.

Note: Steps 1 through 8 can be performed in any order. Steps 9 through 11 must be the final steps in every case, and must be performed in the sequence specified above.

18.4.1 Configuring the Analog Port Pins

The ANSELx registers for the I/O ports associated with the analog inputs are used to configure the corresponding pin as a digital pin. A pin is configured as analog input when the corresponding ANSELx bit = `1`. When the ANSELx bit = `0`, the pin is set to digital control. When configured for analog input, the associated port I/O digital input buffer is disabled so that it does not consume current. The ANSELx registers are set when the device comes out of Reset, causing the ADC input pins to be configured as analog inputs by default.

The TRISx registers control the digital function of the port pins. The port pins that are required as analog inputs must have their corresponding bit set in the specific TRISx register, configuring the pin as an input. If the I/O pin associated with an ADC input is configured as an output by clearing the TRISx bit, the port's digital output level (VOH or VOL) will be converted. After a device Reset, all of the TRISx bits are set. For more information on port pin configuration, refer to the **"I/O Ports"** chapter of the specific device data sheet.

Note: When reading a PORT register that shares pins with the ADC, any pin configured as an analog input reads as a '`0`' when the PORT latch is read. Analog levels on any pin that is defined as a digital input but not configured as an analog input, may cause the input buffer to consume current that exceeds the device specification.

Example 18-1: ADC Initialize

```
int main(int argc, char** argv) {

    int result[3];

    /* Set up the CAL registers */
    AD1CAL1 = DEVADC1;        // Copy the configuration data to the AD1CAL2 = DEVADC2;
                              // AD1CALx special function registers.
    AD1CAL3 = DEVADC3;
    AD1CAL4 = DEVADC4;
    AD1CAL5 = DEVADC5;

    /* Configure AD1CON1 */
    AD1CON1 = 0;              // No AD1CON1 features are enabled including: Stop-in-Idle, early
                              // interrupt, filter delay Fractional mode and scan trigger source.

    /* Configure AD1CON2 */
    AD1CON2 = 0;              // Boost, Low-power mode off, SAMC set to min, set up the ADC Clock
    AD1CON2bits.ADCSEL = 1;   // 1 = SYSCLK, 2 REFCLK03, 3 FRC
    AD1CON2bits.ADCDIV = 4;   // TQ = 1/200 MHz; Tad = 4* (TQ * 2) = 40 ns; 25 MHz ADC clock

    /* Configure AD1CON3 */
    AD1CON3 = 0;              // ADINSEL is not configured for this example. VREFSEL of ‘0’
                              // selects AVDD and AVSS as the voltage reference.

    /* Configure AD1MOD */
    AD1MOD = 0;               // All channels configured for default input and single-ended
                              // with unsigned output results.

    /* Configure AD1GIRGENx */
    AD1GIRQEN1 = 0;           // No global interrupts are used.
    AD1GIRQEN2 = 0;

    /* Configure AD1CSSx */
    AD1CSS1 = 0;              // No channel scanning is used.
    AD1CSS2 = 0;

    /* Configure AD1CMPCONx */
    AD1CMPCON1 = 0;           // No digital comparators are used. Setting the AD1CMPCONx
    AD1CMPCON2 = 0;           // register to ‘0’ ensures that the comparator is disabled. Other
    AD1CMPCON3 = 0;           // registers are “don’t care”.
    AD1CMPCON4 = 0;
    AD1CMPCON5 = 0;
    AD1CMPCON6 = 0;

    /* Configure AD1FLTRx */
    AD1FLTR1 = 0;             // No oversampling filters are used.
    AD1FLTR2 = 0;
    AD1FLTR3 = 0;
    AD1FLTR4 = 0;
    AD1FLTR5 = 0;
    AD1FLTR6 = 0;

    /* Set up the trigger sources */
    AD1TRG1 = 0;              // Initialize all sources to no trigger.
    AD1TRG2 = 0;
    AD1TRG3 = 0;

    AD1TRG1bits.TRGSRC0 = 1;  // Set AN0 to trigger from software.
    AD1TRG1bits.TRGSRC1 = 1;  // Set AN1 to trigger from software.
    AD1TRG1bits.TRGSRC2 = 1;  // Set AN2 to trigger from software.

    /* Turn the ADC on, start calibration */
    AD1CON1bits.ADCEN = 1;

    /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);
```

Example 18-1: ADC Initialize (Continued)

```
 /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);

    while (1) {

        /* Trigger a conversion */
        AD1CON3bits.GSWTRG = 1;

        /* Wait the conversions to complete */
        while (AD1DSTAT1bits.ARDY0 == 0);
        /* fetch the result */
        result[0] = AD1DATA0;
        while (AD1DSTAT1bits.ARDY1 == 0);
        /* fetch the result */
        result[1] = AD1DATA1;
        while (AD1DSTAT1bits.ARDY2 == 0);
        /* fetch the result */
        result[2] = AD1DATA2;

        /*
       * Process results here
       *
       * Note 1: Loop time determines the sampling time since all inputs are Class 1.
       * A delay may be needed to meet sample time requirements.
       * Note 2: The first 5 samples may have reduced accuracy.
       *
       */
    }

    return (1);
}
```

18.4.2 Selecting the ADC Multiplexer Analog Inputs

Each S&H has two inputs referred to as the positive and negative inputs. Input selection options vary for the dedicated and shared S&H are described in the following sections.

Each dedicated S&H is dedicated to a single Class 1 positive input. This means that all control, status and data for that S&H are accessed through the registers associated with its dedicated analog (ANx) input. For example, if the S&H1 for ADC1 uses AN0 as its dedicated input, readings obtained from S&H1 will always be read from AD1DATA0, the global interrupt is enabled with AGIEN0, interrupt status is read using ARDY0, the digital comparator is enabled with CMPE0, and the trigger source is set by TRGSC0. This is true of all dedicated Class 1 S&H circuits, regardless of the following settings.

To provide greater flexibility to the dedicated S&H circuits, a single alternate input is provided for each one. This alternate input can be chosen using the SHxALT<1:0> bits in the ADC1 Input Mode Control Registers AD1IMOD as follows:

- SH0ALT<1:0> (AD1IMOD<17:16>)
- SH1ALT<1:0> (AD1IMOD<19:18>)
- SH2ALT<1:0> (AD1IMOD<21:20>)
- SH3ALT<1:0> (AD1IMOD<23:22>)
- SH4ALT<1:0> (AD1IMOD<25:24>)

An example (for a device with five S&H circuits) is shown in Table 18-3:

Table 18-3: Positive Input Selections

Module	Default Selections (SHxALT<0> = 0)	Alternate Selections (SHxALT<0> = 1)

ADC1	SH0 = AN0
	SH1 = AN1
	SH2 = AN2
	SH3 = AN3
	SH4 = AN4
		SH0 = AN45
		SH1 = AN46
		SH2 = AN47
		SH3 = AN48
		SH4 = AN49

Note: The selections shown in Table 18-3 are for illustration purposes only. Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet for actual channel assignments on a device.

18.4.3 Positive Input Selection for the Shared Sample and Hold

For the shared S&H circuit, the positive input is shared among all Class 2 and Class 3 inputs. Input connection of the analog input ANx to the S&H circuit is automatic for either Class 2 input trigger or during a channel scan of Class 2 and or Class3 inputs. Selecting inputs for Channel scanning is discussed separately in **18.4.8 "Selecting the Scanned Inputs"**.

18.4.4 Negative Input Single-ended and Differential Options (Shared and Dedicated Sample and Hold)

Negative input selection is determined by the setting of the SHxMOD<1:0> bits of the AD1IMOD register. The SHxMOD<1:0> bits allow the inputs to be rail-to-rail, and either single-ended or differential. These bits also scale the internal ADC analog inputs and reference voltages and configure the digital result to line up with the expected full-scale output range.

Table 18-4: Input Configuration

SHxMOD<1:0> Setting	Input Configuration	Input Voltage (see Note 1)		Output
`11`	Differential 2’s Complement	Minimum Input	VINP – VINN = -VREF	-2048
		Maximum Input	VINP – VINN = VREF	+2047
`10`	Differential Unipolar	Minimum Input	VINP – VINN = -VREF	0
		Maximum Input	VINP – VINN = VREF	+4095
`01`	Single-ended 2’s Complement	Minimum Input	VINP = VINN	-2048
		Maximum Input	VINP – VINN = VREF	+2047
`00`	Single-ended Unipolar	Minimum Input	VINP = VINN	0
		Maximum Input	VINP – VINN = VREF	+4095

Legend: VINP = Positive S&H input; VINN = Negative S&H input;
VREF = VREFH – VREFL

Note 1: For proper operation and to prevent device damage, input voltage levels should not exceed the limits listed in the **“Electrical Characteristics”** chapter of the specific device data sheet.

The SHxMOD<1:0> bits are defined in the AD1IMOD register as follows:

- SH0MOD<1:0> (AD1IMOD<1:0>)
- SH1MOD<1:0> (AD1IMOD<3:2>)
- SH2MOD<1:0> (AD1IMOD<5:4>)
- SH3MOD<1:0> (AD1IMOD<7:6>)
- SH4MOD<1:0> (AD1IMOD<9:8>)
- SH5MOD<1:0> (AD1IMOD<11:10>)

18.4.5 Selecting the Format of the ADC Result

The data in the ADC Result register can be read in any of the four supported data formats. The user can select from unsigned integer, signed integer, unsigned fractional, or signed fractional. Integer data is right-justified and fractional data is left-justified.

- The integer/fractional data format selection is specified globally for all ADC inputs using the Fractional Data Output Format bit, FRACT (AD1CON1<11>)
- The signed/unsigned data format selection can be independently specified for each individual S&H circuit using the SHxMOD<1:0> bits, as shown in **18.4.4 “Negative Input Single-ended and Differential Options (Shared and Dedicated Sample and Hold)”**

Figure 18-13 illustrates how a result is formatted.

Figure 18-13: ADC Output Data Formats

Analog-to-Digital Conversion Result

d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

Unsigned Integer

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

Signed Integer

S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

Fractional

d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Signed Fractional

S	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Example 18-2: ADC Class 2 Configuration and Fractional Format

```
int main(int argc, char** argv) {

    int result[3];

    /* Set up the CAL registers */
    AD1CAL1 = DEVADC1;          // Copy the configuration data to the
    AD1CAL2 = DEVADC2;          // AD1CALx special function registers.
    AD1CAL3 = DEVADC3;
    AD1CAL4 = DEVADC4;
    AD1CAL5 = DEVADC5;

    /* Configure AD1CON1 */
    AD1CON1 = 0;                // No AD1CON1 features are enabled including:
                                // Stop-in-Idle, early interrupt, filter delay
                                // and scan trigger source
    AD1CON1bits.FRACT = 1;      // use Fractional output format

    /* Configure AD1CON2 */
    AD1CON2 = 0;                // Boost, Low-power mode off
                                // Set up the ADC Clock
    AD1CON2bits.ADCSEL = 1;     // 1 = SYSCLK, 2 REFCLK03, 3 FRC
    AD1CON2bits.ADCDIV = 4;     // TQ = 1/200 MHz; TAD = 4 * (TQ * 2) = 40 ns;
                                // 25 MHz ADC clock
    AD1CON2bits.SAMC = 9;       // Sample time set to 10 * TAD = 400 ns for
                                // Class 2 and 3 inputs.

    /* Configure AD1CON3 */
    AD1CON3 = 0;                // ADINSEL is not configured for this example.
                                // VREFSEL of 0 selects AVDD and AVSS as the
                                // voltage reference.

    /* Configure AD1MOD */
    AD1MOD = 0;                 // All channels configured for default input
                                // and single-ended with unsigned output results

    /* Configure AD1GIRGENx */
    AD1GIRQEN1 = 0;             // No global interrupts are used.
    AD1GIRQEN2 = 0;

    /* Configure AD1CSSx */
    AD1CSS1 = 0;                // No channel scanning is used.
    AD1CSS2 = 0;

    /* Configure AD1CMPCONx */
    AD1CMPCON1 = 0;             // No digital comparators are used.
    AD1CMPCON2 = 0;             // Setting the AD1CMPCONx register to ‘0’ ensures that
    AD1CMPCON3 = 0;             // the comparator is disabled. Other registers
    AD1CMPCON4 = 0;             // are “don't care”.
    AD1CMPCON5 = 0;
    AD1CMPCON6 = 0;

    /* Configure AD1FLTRx */
    AD1FLTR1 = 0;                // No oversampling filters are used.
    AD1FLTR2 = 0;
    AD1FLTR3 = 0;
    AD1FLTR4 = 0;
    AD1FLTR5 = 0;
    AD1FLTR6 = 0;

    /* Set up the trigger sources */
    AD1TRG1 = 0;                // Initialize all to no trigger.
    AD1TRG2 = 0;
    AD1TRG3 = 0;

    AD1TRG2bits.TRGSRC5 = 1;    // Set AN5 to trigger from software.
    AD1TRG2bits.TRGSRC6 = 1;    // Set AN6 to trigger from software.
    AD1TRG2bits.TRGSRC7 = 1;    // Set AN7 to trigger from software.
```

Example 18-2: ADC Class 2 Configuration and Fractional Format (Continued)

```
 /* Turn the ADC on, start calibration */
    AD1CON1bits.ADCEN = 1;

    /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);

    while (1) {

        /* Trigger a conversion */
        AD1CON3bits.GSWTRG = 1;

        /* Wait the conversions to complete */
        while (AD1DSTAT1bits.ARDY5 == 0);
        /* fetch the result */
        result[0] = AD1DATA5;
        while (AD1DSTAT1bits.ARDY6 == 0);
        /* fetch the result */
        result[1] = AD1DATA6;
        while (AD1DSTAT1bits.ARDY7 == 0);
        /* fetch the result */
        result[2] = AD1DATA7;

        /*
         * Process results here
         *
         * Note 1: The first 5 samples may have reduced accuracy
         * Note 2. The output format is fractional as described in
        * 18.4.5 "Selecting the Format of the ADC Result"
         */
    }

    return (1);
}
```

18.4.6 Selecting the Conversion Trigger Source

Class 1 and Class 2 inputs to the ADC module can be triggered for conversion either individually, or as part of a channel scan sequence. Class 3 inputs can only be triggered as part of a channel scan sequence. Individual or channel scan triggers can originate from an on-board timer or output compare peripheral event, from external digital circuits connected to INT0, from external analog circuits connected to an analog comparator, or through software by setting a trigger bit in a SFR.

Note 1: When conversion triggers for multiple Class 1 or Class 2 analog inputs occur simultaneously, they are prioritized according to a natural order priority scheme based on the analog input used. AN0 has the highest priority, AN1 has the next highest priority, etc. Therefore, time sensitive or transient analog signals should be sampled and converted using Class 1 S&H circuits, with higher natural priority.

2: When selecting trigger sources for Class 1 inputs, ensure that sampling times are not violated.

18.4.6.1 TRIGGER SELECTION FOR CLASS 1 AND CLASS 2 INPUTS

For each one of the Class 1 and Class 2 inputs, the user application can independently specify a conversion trigger source. The individual trigger source for an analog input ‘n’ is specified by the TRGSRCn<4:0> bits located in registers AD1TRG1 through AD1TRG3. Refer to the **“12-bit Analog-to-Digital Converter (ADC)”** chapter of the specific device data sheet for more information on the available conversion trigger options. For example, these trigger sources may include:

- **General Purpose (GP) Timers:** When a period match occurs for the 32-bit timer, Timer3/2 or Timer5/4, or the 16-bit Timer1, Timer3 or Timer5, a special ADC trigger event signal is generated by the timer. This feature does not exist for other timers. For more information, refer to **Section 14. “Timers”** (DS61105) and the specific device data sheet.
- **Output Compare:** The Output Compare peripherals, OC1, OC3, and OC5, can be used to generate an ADC trigger then the output transitions from a low to high state. For more information, refer to **Section 16. “Output Compare”** (DS61111), and the specific device data sheet.
- **Comparators:** The analog Comparators can be used to generate an ADC trigger when the output transitions from a low state to a high state. For more information, refer to **Section 19. “Comparator”** (DS61110), and the **“Comparator”** chapter in the specific device data sheet.
- **External INT0 Pin Trigger:** In this mode, the ADC module starts a conversion on an active transition on the INT0 pin. The INT0 pin may be programmed for either a rising edge input or a falling edge input to trigger the conversion process.
- **Global Software Trigger:** The ADC module can be configured for manually triggering a conversion for all inputs that have selected this trigger option. The user can manually trigger a conversion by setting the Global Software Trigger bit, GSWTRG (AD1CON3<30>).

18.4.6.2 SCANNED TRIGGER SELECTION FOR CLASS 1, CLASS 2 AND CLASS 3 INPUTS (SH3 ONLY)

All available analog inputs can be configured for automatic channel scanning. Class 1 inputs are sampled using their dedicated S&H circuit. Class 2 and Class 3 inputs are sampled using the shared S&H circuit. A single conversion trigger source is selected for all the inputs selected for scanning using the STRGSRC<4:0> bits (AD1CON1<26:22>). On each conversion trigger, the ADC module starts converting in the natural priority, all inputs specified in the user-specified channel scan list. For Class 1 inputs, sampling ends at the time of the trigger. For Class 2 and Class 3 inputs, the trigger initiates a sequential sample/conversion process in the natural priority order.

The trigger options available for channel scan are identical to those available for independent triggering of Class 1 and Class 2 inputs. Any Class 1 or Class 2 inputs that are part of the channel scan must have the STRIG option selected as their trigger source in the TRGSRCn<4:0> bits.

18.4.6.3 USER-REQUESTED INDIVIDUAL CONVERSION TRIGGER

The user can explicitly request a single conversion of any selected analog input at any time during program execution, without changing the trigger source configuration of the ADC. The input to be converted is specified by the ADC Input Select bits, ADINSEL<5:0> (AD1CON3<5:0>). The Individual ADC Input Conversion Request bit, RQCONVRT (AD1CON3<29>) is used to trigger the conversion. This bit is automatically cleared after two peripheral clock cycles, thereby allowing the user software to trigger another conversion if needed.

18.4.7 Selecting the Voltage Reference Source

The user application can select the voltage reference for the ADC module and the voltage reference can be internal or external. The VREF Input Selection bits, VREFSEL<2:0> (AD1CON3<12:10>), select the voltage reference for analog-to-digital conversions. The upper voltage reference (VREFH) and the lower voltage reference (VREFL) may be the internal AVDD and AVSS voltage rails or the VREF+ and VREF- input pins. The external ADC voltage reference may be used to reduce noise in the converter.

The voltages applied to the external reference pins must comply with certain specifications. Refer to the **"Electrical Characteristics"** chapter in the specific device data sheet for the electrical specifications.

The Voltage Reference Boost bit, BOOST (AD1CON2<14>), should be set when the difference between the selected reference voltages (VREFH - VREFL) is less than 0.65 * (AVDD - AVSS). Setting this bit maximizes the SNR for analog-to-digital conversions using small reference voltage rails.

Note: The external VREF+ and VREF- pins can be shared with other analog peripherals. Refer to the **"Pinout Diagrams"** section of the specific device data sheet for more information. In addition, the ANSELx bits for the VREF+ and VREF- pins must be set to Analog mode.

18.4.8 Selecting the Scanned Inputs

To include an analog input in the scan list, its specified bit in the Input Scan Select register (AD1CSS1 or AD1CSS2) must be set to '1'. Each bit in the AD1CSS1 and AD1CSS2 registers corresponds to a different analog input. If the input is a Class 1 or Class 2 input, the trigger source for that input, specified by the TRGSRCn<4:0> bits located in the AD1TRG1 through AD1TRG3 registers, must be set to the STRIG option, indicating that they are triggered as part of the scan process.

Every scan conversion trigger, as specified by the STRGSRC<4:0> bits (AD1CON1<26:22>), starts a scan conversion sequence. As described in **18.3 "ADC Operation, Terminology and Conversion Sequence"**, the scan sequence samples and converts each scanned input in their natural priority.

Note: The AD1CSS1 and AD1CSS2 bits specify only the positive input to the S&H circuits. The negative input of the channel does not change during the scanning sequence.

Example 18-3: ADC Channel Scan

```
int main(int argc, char** argv) {

     int result[3];

    /* Set up the CAL registers */
    AD1CAL1 = DEVADC1;          // Copy the configuration data to the
    AD1CAL2 = DEVADC2;          // AD1CALx special function registers.
    AD1CAL3 = DEVADC3;
    AD1CAL4 = DEVADC4;
    AD1CAL5 = DEVADC5;

    /* Configure AD1CON1 */
    AD1CON1 = 0;                // No AD1CON1 features are enabled including:
                                // Stop-in-Idle, early interrupt, filter delay
                                // and Fractional mode.
    AD1CON1bits.STRGSRC = 1;    // Software trigger initiates scan.

    /* Configure AD1CON2 */
    AD1CON2 = 0;                // Boost, low power mode off.
                                // Set up the ADC Clock
    AD1CON2bits.ADCSEL = 1;     // 1 = SYSCLK, 2 REFCLK03, 3 FRC
    AD1CON2bits.ADCDIV = 4;     // TQ = 1/200 MHz; TAD = 4 * (TQ * 2) = 40 ns;
                                // 25 MHz ADC clock.
    AD1CON2bits.SAMC = 9;       // Sample time set to 10 * TAD = 400 ns for
                                // Class 2 and 3 Inputs.

    /* Configure AD1CON3 */
    AD1CON3 = 0;                // ADINSEL is not configured for this example.
                                // VREFSEL of ‘0’ selects AVDD and AVSS as the
                                // voltage reference.

    /* Configure AD1MOD */
    AD1MOD = 0;                 // All channels configured for default input
                                // and single-ended with unsigned output results

    /* Configure AD1GIRGENx */
    AD1GIRQEN1 = 0;             // No global interrupts are used.
    AD1GIRQEN2 = 0;

    /* Configure AD1CSSx */
    AD1CSS1 = 0;                // Clear all bits
    AD1CSS2 = 0;
    AD1CSS1bits.CSS0 = 1;       // AN0 (Class 1) set for scan
    AD1CSS1bits.CSS5 = 1;       // AN5 (Class 2) set for scan
    AD1CSS2bits.CSS43 = 1;      // AN43 (Class 3) set for scan

    /* Configure AD1CMPCONx */
    AD1CMPCON1 = 0;             // No digital comparators are used.
    AD1CMPCON2 = 0;             // Setting the AD1CMPCONx register to ‘0’ ensures that
    AD1CMPCON3 = 0;             // the comparator is disabled. Other comparator
    AD1CMPCON4 = 0;             // registers are “don't care”.
    AD1CMPCON5 = 0;
    AD1CMPCON6 = 0;

    /* Configure AD1FLTRx */
    AD1FLTR1 = 0;                // No oversampling filters are used.
    AD1FLTR2 = 0;
    AD1FLTR3 = 0;
    AD1FLTR4 = 0;
    AD1FLTR5 = 0;
    AD1FLTR6 = 0;

    /* Set up the trigger sources */
    AD1TRG1 = 0;                // Initialize all sources to no trigger.
    AD1TRG2 = 0;
    AD1TRG3 = 0;
```

Example 18-3: ADC Channel Scan (Continued)

```
    AD1TRG1bits.TRGSRC0 = 3;    // Set AN0 (Class 1) to trigger from scan source
    AD1TRG2bits.TRGSRC5 = 3;    // Set AN5 (Class 2) to trigger from scan source
                                // AN43 (class 3) always uses scan trigger source

    /* Turn the ADC on, start calibration */
    AD1CON1bits.ADCEN = 1;

    /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);

    while (1) {

        /* Trigger a conversion */
        AD1CON3bits.GSWTRG = 1;

        /* Wait the conversions to complete */
        while (AD1DSTAT1bits.ARDY0 == 0);
        /* fetch the result */
        result[0] = AD1DATA0;
        while (AD1DSTAT1bits.ARDY5 == 0);
        /* fetch the result */
        result[1] = AD1DATA5;
        while (AD1DSTAT2bits.ARDY43 == 0);
        /* fetch the result */
        result[2] = AD1DATA43;

        /*
         * Process results here
         *
         * Note: The first 5 samples may have reduced accuracy.
         *
         */
    }
    return (1);
}
```

18.4.9 Selecting the Analog-to-Digital Conversion Clock Source and Prescaler

The ADC module can use the internal Fast RC (FRC) oscillator output or the System Clock (SYSCLK) as the conversion clock source (TQ). On some devices, the reference clock can be used as the conversion clock source. Refer to the **"12-bit Analog-to-Digital Converter (ADC)"** chapter in the specific device data sheet to determine which reference clock output source may be available.

When the ADCSEL<1:0> bits (AD1CON2<9:8>) = `11`, the internal FRC oscillator is used as the ADC clock source. When using the internal FRC oscillator, the ADC module can continue to function in Sleep and Idle modes.

Note: It is recommended that applications that require precise timing of ADC acquisitions use SYSCLK as the clock source for the ADC.

When the Analog-to-Digital Conversion Clock Source (TQ) bits, ADCSEL<1:0> (AD1CON2<9:8>) = `01`, SYSCLK is used as the conversion clock source. When the ADCSEL<1:0> bits = `00`, the ADC clock is disabled.

The analog-to-digital conversion clock period (TAD) is the period of the selected clock source after the prescaler ADC Input Clock Divider bits, ADCDIV<6:0> (AD1CON2<6:0>), are applied. The resultant value of TAD can range from 1 * TQ to 254 * TQ.

For correct analog-to-digital conversions, the TAD limits must not be exceeded. ADC clock frequencies from 1 MHz to 28 MHz are supported by the ADC module.

The maximum rate at which analog-to-digital conversions may be completed by the ADC module (effective conversion throughput) is 28 Msps. However, the maximum rate at which a single input channel can be converted is dependent on the sampling time requirements. The fastest possible conversion rates can be achieved by sampling and converting an analog signal through multiple Class 1 inputs.

> **Note:** The minimum conversion latency, (time from trigger to availability of result) is 10 * TAD. Actual latency could be as much as 11 * TAD since the trigger is asynchronous to the ADC clock.

Equation 18-5 provides the TAD value as a function of the ADCDIV<6:0> control bits (AD1CON2<6:0>) and the device instruction cycle clock period, TCY.

Equation 18-5: A/D Conversion Clock Period

For $ADCDIV = 0$

$$T_{AD} = T_Q$$

For $ADCDIV > 0$

$$T_{AD} = 2\ T_Q * ADCDIV$$

For either case:

$$35.71\ ns \leq T_{AD} \leq 1.0\ \mu s$$

For the shared S&H circuits, the SAMC<7:0> bits (AD1CON2<23:16>) specify the number of TAD clock cycles for which an analog input should be sampled between the conversion trigger and the start of conversion.

18.4.10 Turning ON the ADC

When the ADC Operating Mode bit, ADCEN (AD1CON1<15>), is set to ‘1’, the module is in Active mode and is fully powered and functional.

When the ADCEN bit is ‘0’, the module is disabled. The digital and analog portions of the circuit are turned off for maximum current savings.

In order to return to Active mode from a disabled state, the user must wait for the analog stages to stabilize. Refer to the **“Electrical Characteristics”** chapter in the specific device data sheet for more information on the stabilization time.

> **Note:** Writing to ADC control bits that control the ADC clock, channel assignments, channel scanning, voltage reference selection, S&H circuit operating modes, and interrupt configuration is not recommended while the ADC module is enabled.

18.4.11 ADC Status Bits

The ADC module includes the ADRDY status bit (AD1CON2<31>), which indicates its current state. This bit indicates that the ADC module is ready to begin a conversion. The ADCRDY bit is only set after the ADC module has been enabled and calibrated. The user application should not perform any ADC operations until the ADCRDY bit is set.

18.4.12 ADC Calibration

Prior to enabling the ADC module, the user must read the ADC calibration data from the Configuration memory (refer to the **"Special Features"** chapter in the specific device data sheet for more information), and then write this data into the calibration registers AD1CAL1 through AD1CAL5.

When performing a self-calibration, the input mode for SH0 must be set to one of the differential input modes. Use either the differential unipolar encoded format by setting the SH0MOD<1:0> bits (AD1MOD<1:0>) = `10` or the differential two's compliment encoded format by setting the SH0MOD<1:0> bits = `11`. Self-calibration occurs each time the ADC is enabled or by initiating a calibration cycle after the ADC is enabled. The ADC is enabled by setting the ADCEN bit (AD1CON1<15>) = `1`. A calibration cycle is initiated after the ADC is enabled by setting the CAL bit (AD1CON3<31>) = `1`. After calibration the input mode for SH0 may be changed based on the application need.

The following conditions require recalibration to be performed:

- A change in reference voltage after enabling the ADC module
- Changing the BOOST bit setting after enabling the ADC module

18.5 ADDITIONAL ADC FUNCTIONS

This section describes some additional features of the ADC module such as a digital comparator and an oversampling filter.

18.5.1 Digital Comparator

The ADC module features digital comparators that can be used to monitor selected analog input conversion results and generate interrupts when a conversion result is not within the user-specified limits. Conversion triggers are still required to initiate conversions. The comparison occurs automatically once the conversion is complete. This optional feature is enabled by setting the Digital Comparator Module Enable bit, ENDCMP (AD1CMPCON<7>).

The user application makes use of an interrupt that is generated when the analog-to-digital conversion result is higher or lower than the specified high and low limit values in the AD1CMP register. The high and low limit values are specified in the ADCMPHI<15:0> bits (AD1CMP<31:16>) and the ADCMPLO<15:0> bits (AD1CMP<15:0>).

The CMPEx bits (‘x’ = 0 through 31) in the AD1CMPENx registers are used to specify which analog inputs are monitored by the digital comparator (for the first 32 analog inputs, ANx, where ‘x’ = 0 through 31). The AD1CMPCON register specifies the comparison conditions that will generate an interrupt, as follows:

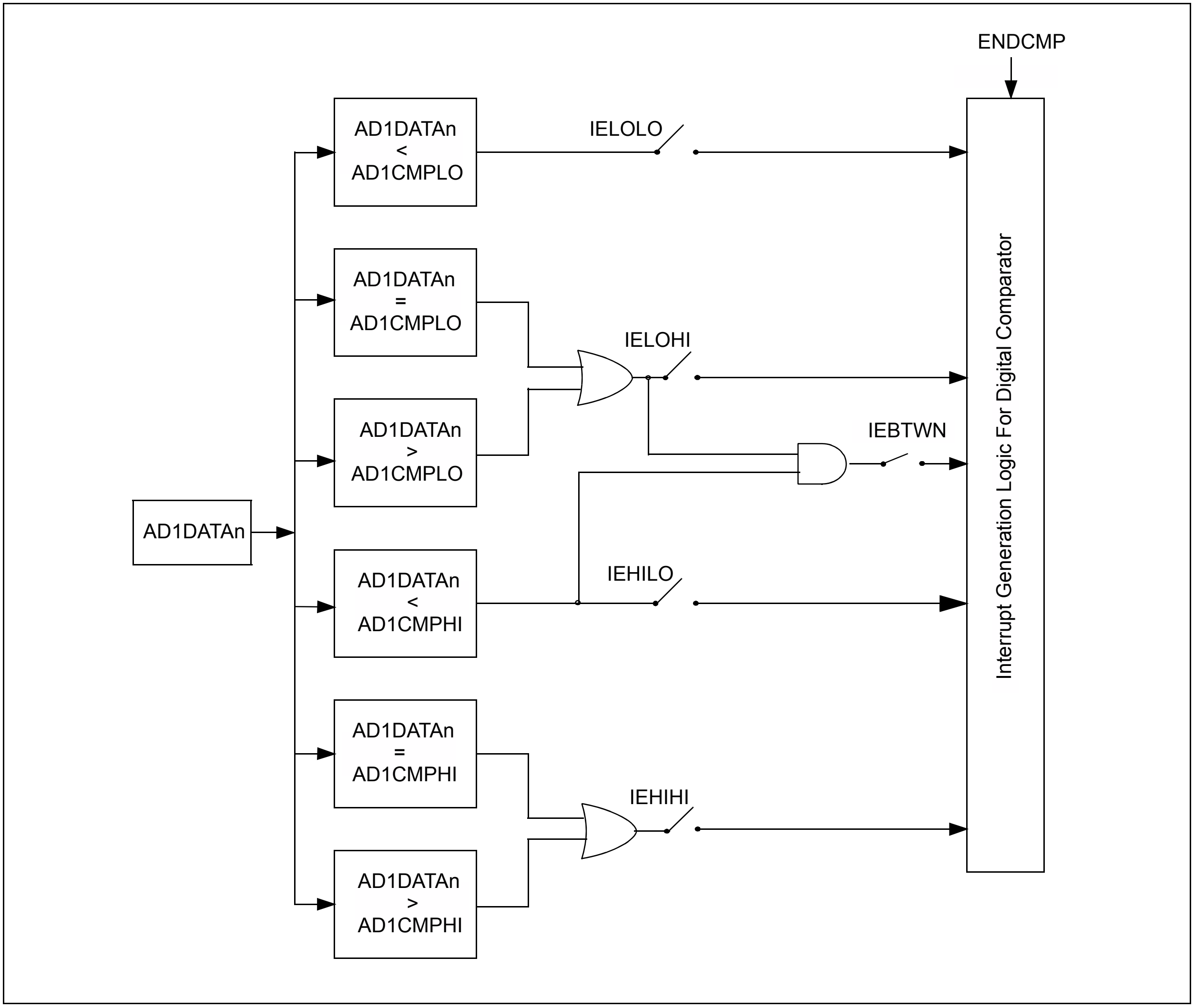
- When IEBTWN = 1, interrupt is generated when AD1CMPLO ≤ AD1DATAn < AD1CMPHI
- When IEHIHI = 1, interrupt is generated when AD1CMPHI ≤ AD1DATAn
- When IEHILO = 1, interrupt is generated when AD1DATAn < AD1CMPHI
- When IELOHI = 1, interrupt is generated when AD1CMPLO ≤ AD1DATAn
- When IELOLO = 1, interrupt is generated when AD1DATAn < AD1CMPLO

The comparator event generation is illustrated in Figure 18-14. When the ADC module generates a conversion result, the conversion result is provided to the comparator and is formatted based on the selected data format stored in the specific result register. The comparator uses the SH0MOD<1:0> (AD1IMOD<1:0>), SH1MOD<1:0> (AD1IMOD<3:2>), SH2MOD<1:0> (AD1IMOD<5:4>), SH3MOD<1:0> (AD1IMOD<7:6>), SH4MOD<1:0> (AD1IMOD<9:8>), or SH5MOD<1:0> (AD1IMOD<11:10>) bits (depending on the S&H circuit used) to determine the data format used and appropriately select whether the comparison should be signed or unsigned. The global ADC setting, which is specified by the FRACT bit (AD1CON1<11>), is also used to set the fractional or integer format. The digital magnitude comparator compares the ADC result with the high and low limit values (depending on the selected comparison criteria) in the AD1CMP register.

Depending on the comparator results, a digital comparator interrupt event may be generated. If a comparator event occurs, the Digital Comparator Interrupt Event Detected status bit, DCMPIF(AD1CMPCON<15>), is set, and the Analog Input Identification (ID) bits, AINID<4:0> (AD1CMPCON<12:8>), are automatically updated so that the user application knows which analog input generated the interrupt event.

> **Note 1:** The Digital Comparator module supports only the first 32 analog inputs (AN0 through AN31).
>
> **2:** The user software must format the values contained in the ADxCMPn registers to match converted data format as either signed or unsigned, and fractional or integer.

Figure 18-14: Digital Comparator



Example 18-4: ADC Digital Comparator

```
int main(int argc, char** argv) {

    int eventFlag, result;

    /* Set up the CAL registers */
    AD1CAL1 = DEVADC1;            // Copy the configuration data to the
    AD1CAL2 = DEVADC2;            // AD1CALx special function registers.
    AD1CAL3 = DEVADC3;
    AD1CAL4 = DEVADC4;
    AD1CAL5 = DEVADC5;

    /* Configure AD1CON1 */
    AD1CON1 = 0;                  // No AD1CON1 features are enabled including:
                                  // Stop-in-Idle, early interrupt, filter delay
                                  // fractional mode and scan trigger source

    /* Configure AD1CON2 */
    AD1CON2 = 0;                  // Boost, low power mode off
                                  // Setup the ADC Clock
    AD1CON2bits.ADCSEL = 1;       // 1 = SYSCLK, 2 REFCLK03, 3 FRC
    AD1CON2bits.ADCDIV = 4;       // TQ = 1/200 MHz; TAD = 4* (TQ * 2) = 40 ns;
                                  // 25 MHz ADC clock
    AD1CON2bits.SAMC = 9;         // Sample time set to 10 * TAD = 400 ns for
                                  // Class 2 and 3 inputs.

    /* Configure AD1CON3 */
    AD1CON3 = 0;                  // ADINSEL is not configured for this example.
                                  // VREFSEL of ‘0’ selects AVDD and AVSS as the
                                  // voltage reference.

    /* Configure AD1MOD */
    AD1MOD = 0;                   // All channels configured for default input
                                  // and single-ended with unsigned output results.

    /* Configure AD1GIRGENx */
    AD1GIRQEN1 = 0;               // No global interrupts are used.
    AD1GIRQEN2 = 0;

    /* Configure AD1CSSx */
    AD1CSS1 = 0;                  // No channel scanning is used.
    AD1CSS2 = 0;

    /* Configure AD1CMPCONx */
    AD1CMPCON1 = 0;               // Set up Comparator 1, clear all AD1CMPCON1 bits.
    AD1CMP1bits.ADCMPHI = 0xC00;  // High limit is a 3072 result.
    AD1CMP1bits.ADCMPLO = 0x500;  // Low limit is a 1280 result.
    AD1CMPCON1bits.IEBTWN = 1;    // Create an event when the measured result is
                                  // >= low limits and < high limit.
    AD1CMPEN1 = 0;                // Clear all enable bits
    AD1CMPEN1bits.CMPE5 = 1;      // set the bit corresponding to AN5
    AD1CMPCON1bits.ENDCMP = 1;    // enable comparator

    AD1CMPCON2 = 0;               // Digital comparators 2 through 6 are not used.
    AD1CMPCON3 = 0;               // Setting the AD1CMPCONx to ‘0’ ensures that
    AD1CMPCON4 = 0;               // the comparator is disabled. Other registers
    AD1CMPCON5 = 0;               // are “don't care”.
    AD1CMPCON6 = 0;

    /* Configure AD1FLTRx */
    AD1FLTR1 = 0;                // No oversampling filters are used.
    AD1FLTR2 = 0;
    AD1FLTR3 = 0;
    AD1FLTR4 = 0;
    AD1FLTR5 = 0;
    AD1FLTR6 = 0;
```

Example 18-4: ADC Digital Comparator (Continued)

```
 /* Set up the trigger sources */
    AD1TRG1 = 0;                // Initialize all sources to no trigger.
    AD1TRG2 = 0;
    AD1TRG3 = 0;

    AD1TRG2bits.TRGSRC5 = 1;    // Set AN5 to trigger from software.

    /* Turn the ADC on, start calibration */
    AD1CON1bits.ADCEN = 1;

    /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);

    while (1) {

        /* Trigger a conversion */
        AD1CON3bits.GSWTRG = 1;

        /* Wait the conversions to complete */
        while (AD1DSTAT1bits.ARDY5 == 0);
        result = AD1DATA5;
        /* Note: It is not necessary to fetch the result for the digital
         * comparator to work. In this example we are triggering from
         * software so we are using the ARDY5 to gate our loop. Reading the
         * data clears the ARDY bit.
         */

        /* See if we have a comparator event*/
        if (AD1CMPCON1bits.DCMPED == 1) {

            eventFlag = 1;

            /*
             * Process results Here
             *
             * Note: The first 5 samples may have reduced accuracy.
             *
             */
        }
    }

    return (1);
}
```

18.5.2 Oversampling Digital Filter

The ADC module supports up to six oversampling digital filters. The oversampling digital filter consists of an accumulator and a decimator (down-sampler), which function together as a low-pass filter. By sampling an analog input at a higher-than-required sample rate, and then processing the data through the oversampling digital filter, the effective resolution of the ADC module can be increased at the expense of decreased conversion throughput. Using 4x oversampling yields one extra bit of resolution, 16x oversampling yields two extra bits of resolution, 64x oversampling provides three extra bits of resolution, and 256x oversampling provides four extra bits of resolution.

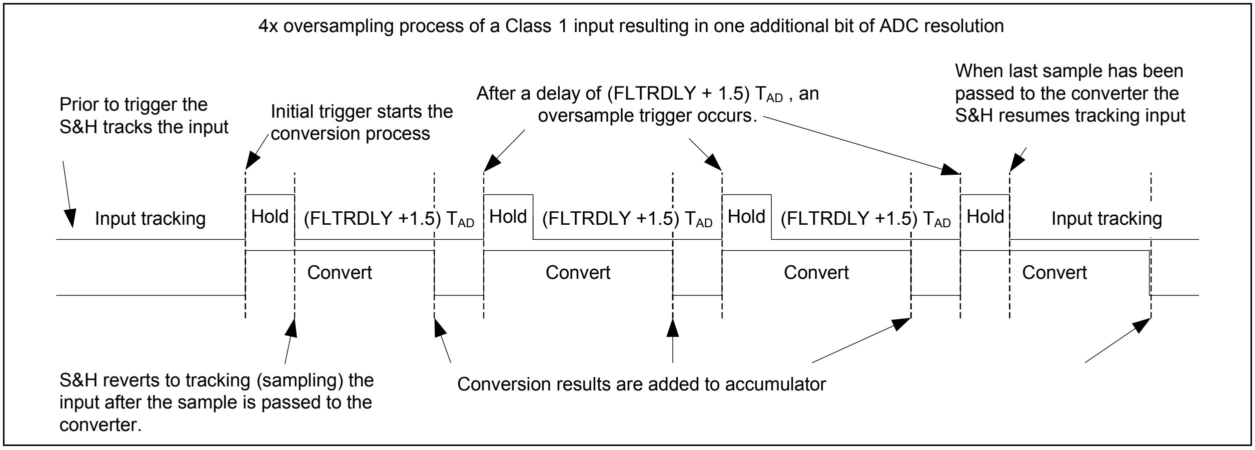
The user application should configure the following bits to perform an oversampling conversion:

- Select the amount of oversampling through the Oversampling Filter Oversampling Ratio (OVRSAM<2:0>) bits in the ADC1 Filter register (AD1FLTRn<28:26>)
- Set the sample time for subsequent samples:
  - If using Class 1 inputs, select the sample time of the recurring conversions using the (FILTRDLY<4:0>) bits in the AD1CON1 registers (AD1CON1<31:27>)
  - If using Class 2 or Class 3 inputs, select the sample time using the (SAMC<7:0>) bits in the ADC1 Control register 2 (AD1CON2<23:16>)
- Select the specific analog input to be oversampled by configuring the Channel ID Selection (CHNLID<5:0>) bits (AD1FLTRn<21:16>)
- If needed, include the oversampling filter interrupt event in the global ADC interrupt, by setting the Accumulator Filter Global Interrupt Enable (AFGIEN) bit (AD1FLTRn<25>)
- Enable the oversampling filter by setting the Oversampling Filter Accumulator Enable (AFEN) bit (AD1FLTRn<31>)

Once the oversampling digital filter is configured, it waits for an external conversion trigger to initiate the oversampling process. The trigger signal for the channel to be oversampled causes the accumulator to be cleared and initiates the first conversion. When each conversion request is in the process of being converted in the conversion pipeline, the filter begins a new sample. Once each conversion request has been passed to the pipeline converter, a new sample is initiated based on the values of the FILTDLY<4:0> or SAMC<7:0> bits for Class 1, Class 2, and Class 3 inputs, respectively. This process continues until the required number of samples (4, 8, 16, 32, 64, 128, or 256) has been converted. When the converted samples have been summed, the output is transferred to the Oversampling Filter Data output bits, FLTRDATA<15:0> (AD1FLTRn<15:0>) and the Accumulator Filter Data Ready Interrupt Flag bit, AFRDY (AD1FLTRn<24>), is set.

Figure 18-15 illustrates 4x oversampling on a Class 1 input. Prior to the trigger, the Class 1 input S&H is tracking the input signal. The trigger starts the first conversion. Once the sample has been transferred to the converter, the S&H resumes tracking and starts a filter delay timer (FILTRDLY<4:0>). When the filter delay timer expires a new conversion sequence occurs. After each sample is converted it is added to the accumulator. The sequence repeats until the number of samples specified by the OVRSAM<2:0> bits have been accumulated. When the last sample has been converted, its value is added to the accumulator. The result is left shifted and then stored in the FLTRDATA<15:0> bits.

Figure 18-15: 4x Oversampling of a Class 1 Input


4x oversampling process of a Class 1 input resulting in one additional bit of ADC resolution
Prior to trigger the S&H tracks the input
Initial trigger starts the conversion process
After a delay of (FLTRDLY + 1.5) T_{AD} , an oversample trigger occurs.
When last sample has been passed to the converter the S&H resumes tracking input
Input tracking
Hold
(FLTRDLY +1.5) T_{AD}
Hold
(FLTRDLY +1.5) T_{AD}
Hold
(FLTRDLY +1.5) T_{AD}
Hold
Input tracking
Convert
Convert
Convert
Convert
S&H reverts to tracking (sampling) the input after the sample is passed to the converter.
Conversion results are added to accumulator

When input is being sampled, lower priority inputs can be transferred to the converter without disrupting oversampling timing. Higher priority conversion requests can disrupt the oversampling timing and can produce unexpected results. The user application should arrange the initiation trigger for the oversampling filters to occur while there are no expected interruptions from higher priority ADC conversion requests.

Figure 18-16 illustrates 4x Oversampling using a Class 2 or Class 3 input. Triggering a Class 2 or Class 3 input initiates sampling for the length of time defined by SAMC. Retriggers generated by the oversampling logic use the SAMC<4:0> bits, not the FLTRDLY<4:0> bits, to set the sample time.

Since Class 2 and Class 3 inputs all use the shared S&H, oversampling blocks lower priority Class 2 and Class 3 triggers. Higher priority Class 2 and Class 3 triggers will completely disrupt the oversampling process, and therefore, should be avoided completely.

Figure 18-16: 4x Oversampling of a Class 2 Input

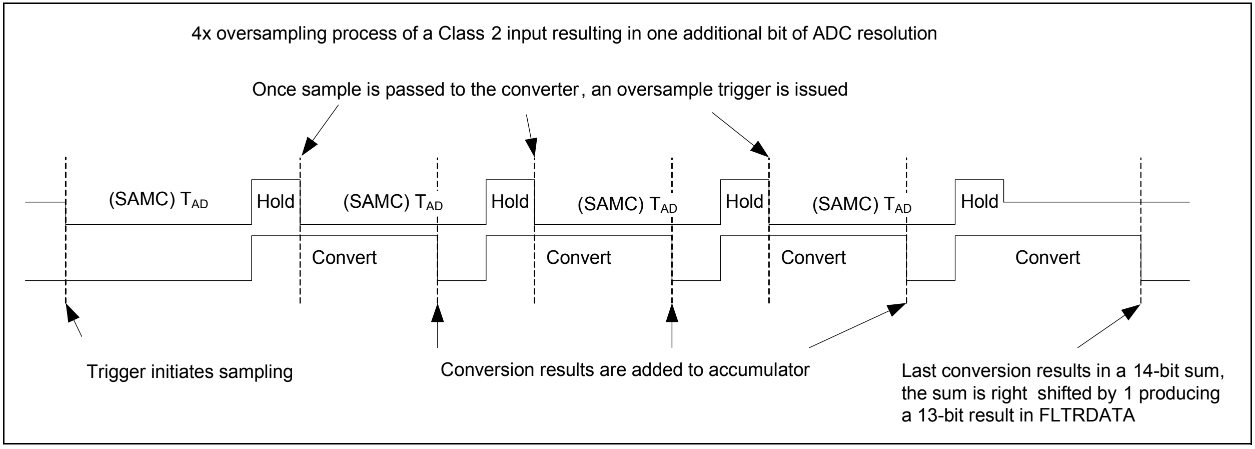
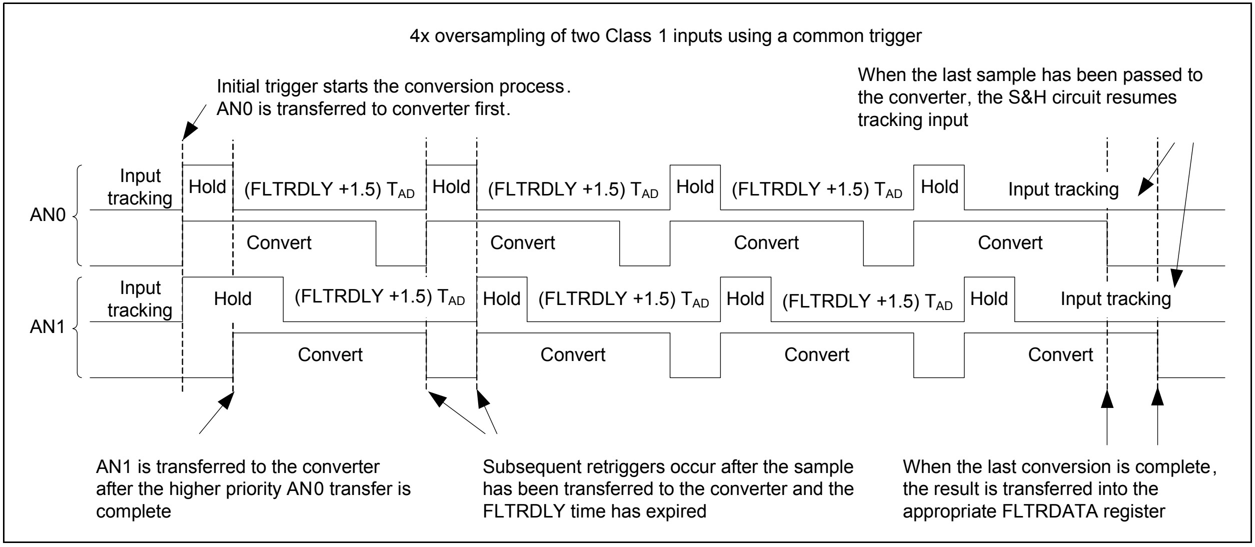

4x oversampling process of a Class 2 input resulting in one additional bit of ADC resolution
Once sample is passed to the converter, an oversample trigger is issued
(SAMC) TAD
Hold
(SAMC) TAD
Hold
(SAMC) TAD
Hold
(SAMC) TAD
Hold
Convert
Convert
Convert
Convert
Trigger initiates sampling
Conversion results are added to accumulator
Last conversion results in a 14-bit sum, the sum is right shifted by 1 producing a 13-bit result in FLTRDATA

Figure 18-17 illustrates 4x oversampling of two Class 1 inputs simultaneously (common trigger). At the initial trigger, both S&H circuits enter the hold state simultaneously. The higher priority AN0 is transferred to the converter first, followed immediately by the AN1 input. For the remainder of the oversampling process the conversions are skewed by 1 TAD. Each analog input creates a unique interrupt when the oversampling process is complete.

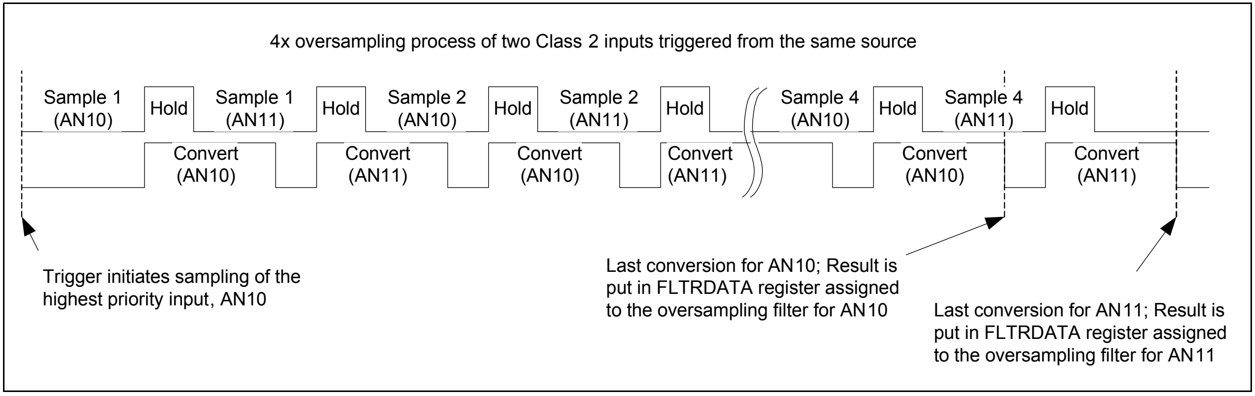
Figure 18-17: Oversampling Two Class 1 Inputs


4x oversampling of two Class 1 inputs using a common trigger
Initial trigger starts the conversion process. AN0 is transferred to converter first.
When the last sample has been passed to the converter, the S&H circuit resumes tracking input
AN0
Input tracking
Hold
(FLTRDLY +1.5) TAD
Hold
(FLTRDLY +1.5) TAD
Hold
(FLTRDLY +1.5) TAD
Hold
Input tracking
Convert
Convert
Convert
Convert
AN1
Input tracking
Hold
(FLTRDLY +1.5) TAD
Hold
(FLTRDLY +1.5) TAD
Hold
(FLTRDLY +1.5) TAD
Hold
Input tracking
Convert
Convert
Convert
Convert
AN1 is transferred to the converter after the higher priority AN0 transfer is complete
Subsequent retriggers occur after the sample has been transferred to the converter and the FLTRDLY time has expired
When the last conversion is complete, the result is transferred into the appropriate FLTRDATA register

When using the oversampling feature on the analog inputs connected to the shared S&H, the system will interleave sampling, alternating the highest priority oversampled input with one other channel. Figure 18-18 illustrates this configuration using AN10 and AN11. In this case, the same results are obtained if the two inputs, AN10 and AN11, are configured with a common independent trigger, or if they are set up as a two-input scan list.

Since a single shared S&H is used, sampling must alternate between the two inputs, which reduces the sample rate by 50 percent in comparison with oversampling a single channel with the same settings.

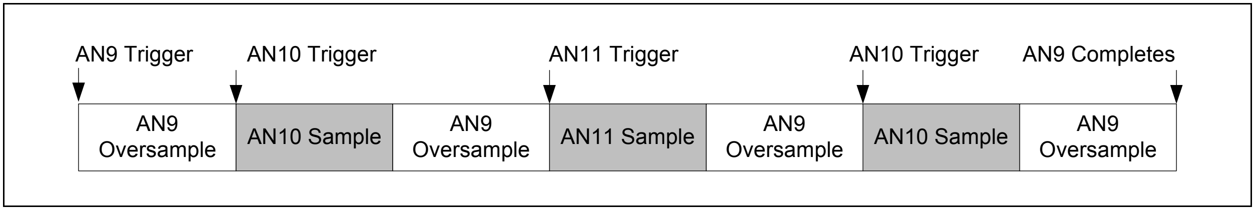
Figure 18-18: Oversampling Two Class 2 Inputs


4x oversampling process of two Class 2 inputs triggered from the same source
Sample 1 (AN10)
Hold
Sample 1 (AN11)
Hold
Sample 2 (AN10)
Hold
Sample 2 (AN11)
Hold
Sample 4 (AN10)
Hold
Sample 4 (AN11)
Hold
Convert (AN10)
Convert (AN11)
Convert (AN10)
Convert (AN11)
Convert (AN10)
Convert (AN11)
Trigger initiates sampling of the highest priority input, AN10
Last conversion for AN10; Result is put in FLTRDATA register assigned to the oversampling filter for AN10
Last conversion for AN11; Result is put in FLTRDATA register assigned to the oversampling filter for AN11

An oversampled input on the shared S&H will alternate with one other lower priority channel. Therefore, a maximum of two Class 2 or Class 3 inputs can be oversampled at one time.

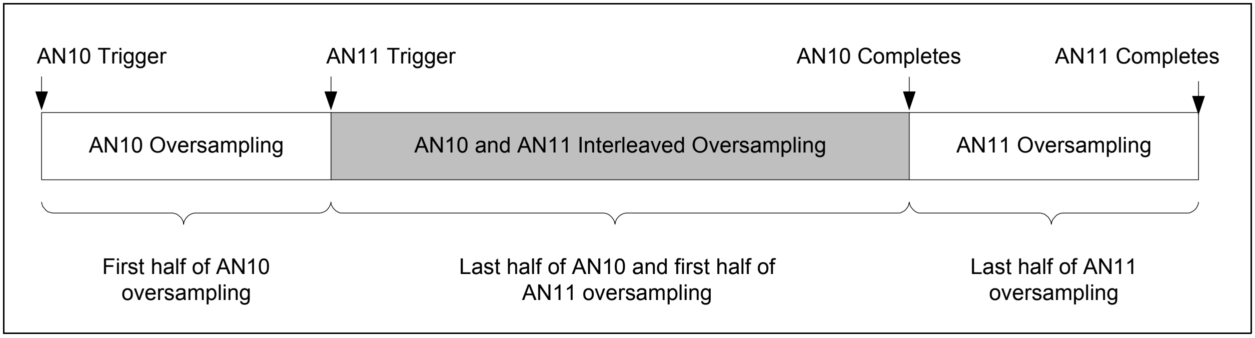
This interleaving also applies when lower priority non-oversampled Class 2 inputs are triggered during oversampling of a Class 2 input, as shown in Figure 18-19. Here, the oversampling of AN9 is interleaved with two AN10 samples and a single AN11 sample.

Figure 18-19: Class 2 Oversampling Interrupted by Non-oversampled Trigger



If oversampling a single Class 2 input during the oversampling process, a lower priority, oversampled input is triggered, the following effect will take place, as shown in Figure 18-20.

Figure 18-20: Oversampling Two Class 2 Inputs with Independent Triggers


AN10 Trigger
AN11 Trigger
AN10 Completes
AN11 Completes
AN10 Oversampling
AN10 and AN11 Interleaved Oversampling
AN11 Oversampling
First half of AN10 oversampling
Last half of AN10 and first half of AN11 oversampling
Last half of AN11 oversampling

It is important to gate the triggers of inputs on the shared S&H when using oversampling to ensure that the oversampling is not disrupted, otherwise unexpected results could occur.

Example 18-5: ADC Digital Oversampling Filter

```
int main(int argc, char** argv) {

    int result;

    /* Set up the CAL registers */
    AD1CAL1 = DEVADC1;          // Copy the configuration data to the
    AD1CAL2 = DEVADC2;          // AD1CALx special function registers.
    AD1CAL3 = DEVADC3;
    AD1CAL4 = DEVADC4;
    AD1CAL5 = DEVADC5;

    /* Configure AD1CON1 */
    AD1CON1 = 0;                // No AD1CON1 features are enabled including:
                                // Stop-in-Idle, early interrupt,
                                // Fractional mode and scan trigger source
    AD1CON1bits.FILTRDLY = 5;   // sample time between oversampling triggers
                                // is 6.5 * TAD = 260 ns.

    /* Configure AD1CON2 */
    AD1CON2 = 0;                // Boost, Low-power mode off, SAMC set to min,
                                // Set up the ADC Clock
    AD1CON2bits.ADCSEL = 1;     // 1 = SYSCLK, 2 REFCLK03, 3 FRC
    AD1CON2bits.ADCDIV = 4;     // TQ = 1/200 MHz; TAD = 4 * (TQ * 2) = 40 ns;
                                // 25 MHz ADC clock

    /* Configure AD1CON3 */
    AD1CON3 = 0;                // ADINSEL is not configured for this example.
                                // VREFSEL of ‘0’ selects AVDD and AVSS as the
                                // voltage reference.

    /* Configure AD1MOD */
    AD1MOD = 0;                 // All channels configured for default input
                                // and single-ended with unsigned output results.

    /* Configure AD1GIRGENx */
    AD1GIRQEN1 = 0;             // No global interrupts are used.
    AD1GIRQEN2 = 0;

    /* Configure AD1CSSx */
    AD1CSS1 = 0;                // No channel scanning is used.
    AD1CSS2 = 0;

    /* Configure AD1CMPCONx */
    AD1CMPCON1 = 0;             // No digital comparators are used.
    AD1CMPCON2 = 0;             // Setting the AD1CMPCONx register to ‘0’ ensures that
    AD1CMPCON3 = 0;             // the comparator is disabled. Other registers
    AD1CMPCON4 = 0;             // are “don't care”.
    AD1CMPCON5 = 0;
    AD1CMPCON6 = 0;

    /* Configure AD1FLTRx */
    AD1FLTR1 = 0;               // Clear all bits for Filter 1
    AD1FLTR1bits.CHNLID = 0;    // Use AN0 as the source
    AD1FLTR1bits.OVRSAM = 1;    // 16x oversampling
    AD1FLTR1bits.AFEN = 1;      // Enable filter 1
    AD1FLTR2 = 0;               // Filters 2 through 6 are not used.
    AD1FLTR3 = 0;
    AD1FLTR4 = 0;
    AD1FLTR5 = 0;
    AD1FLTR6 = 0;

    /* Set up the trigger sources */
    AD1TRG1 = 0;                // Initialize all sources to no trigger.
    AD1TRG2 = 0;
    AD1TRG3 = 0;

    AD1TRG1bits.TRGSRC0 = 1;    // Set AN0 to trigger from software
```

Example 18-5: ADC Digital Oversampling Filter (Continued)

```
    /* Turn the ADC on, start calibration */
    AD1CON1bits.ADCEN = 1;

    /* Wait for calibration to complete */
    while (AD1CON2bits.ADCRDY == 0);

    while (1) {

        /* Trigger a conversion */
        AD1CON3bits.GSWTRG = 1;

        /* Wait for the oversampling process to complete */
        while (AD1FLTR1bits.AFRDY == 0);
        /* fetch the result */
        result = AD1FLTR1bits.FLTRDATA;

        /*
         * Process result Here
         *
         * Note 1: Loop time determines the sampling time for the first sample.
         * remaining samples sample time is determined by FLTRDLY in AD1CON1.
         * Note 2: The first 5 samples may have reduced accuracy.
         *
         */
    }

    return (1);
}
```

18.5.3 Oversampling Conversion Times

Equation 18-6 and Equation 18-7 describe the timing for oversampling using Class 1 and Class 2 inputs. They represent the time from trigger to the conversion result availability.

Equation 18-6: Oversampling Time for Class 1 Input

Total Conversion Time for a single trigger of an oversampled input:

$$T_{C1OVS\_CONV} = (N \cdot (FILTDLY + 4) + 8)T_{AD}$$ (see **Note 1**)

Sample Time used during retriggering (see **Note 2**):

$$T_{C1OVS\_SAMP} = (FILTDLY + 1.5)T_{AD}$$

Minimum time between triggers of an oversampled input:

$$T_{C1OVS\_TRG\_MIN} = (N \cdot (FILTDLY + 4))T_{AD}$$ (see **Note 1**)

Where:

N = Total number of samples determined by the OVRSAM<2:0> bits (AD1FLTR<28:26>)

T_{AD} = Analog-to-digital conversion clock period

$FILTDLY$ = FILTDLY<4:0> bits (AD1CON1<31:27>)

Note 1: This equation assumes that no triggers of any higher priority Class 1 inputs occur during the oversample conversion sequence. Each higher priority Class 1 conversion which takes place will add one TAD to the equation.

2: The sample time for the initial sample is determined by the trigger rate, as is the case for all Class 1 inputs. This equation describes the sample time for subsequent oversampling retriggers.

Equation 18-7: Oversampling Time for Class 2 Input

Total Conversion Time:

$$T_{C2OVS\_CONV} = (N \cdot (SAMC + 4) + 7)T_{AD}$$

Sample Time:

$$T_{C2OVS\_SAMP} = (SAMC + 1)T_{AD}$$

Where:

N = Total number of samples determined by the OVRSAM<2:0> bits (AD1FLTR<28:26>)

T_{AD} = Analog-to-digital conversion clock period

$SAMC$ = SAMC<7:0> bits (AD1CON2<23:16>)

Note: This equation assumes there are no pending conversions from higher priority Class 1 analog channels.

18.6 INTERRUPTS

Each ADC module supports interrupts triggered from a variety of sources, which can be processed individually or globally. An early interrupt feature is also available to compensate for interrupt servicing latency.

After an enabled interrupt is generated, the CPU will jump to the vector assigned to that interrupt. The CPU will then begin executing code at the vector address. The user software at this vector address should perform the required operations, such as processing the data results, clearing the interrupt flag, and then exit. For more information on interrupts and the vector address table details, refer to the **Section 8. “Interrupts”** (DS61108) in the *“PIC32 Family Reference Manual”* and the **“Interrupt Controller”** chapter in the specific device data sheet.

18.6.1 Interrupt Sources

Each ADC is capable of generating interrupts from the following events:

- ANx Data Ready Event – Upon a completion of a conversion from an analog input source (ANx) the ARDYx bit associated with that input will be set in the AD1STATn register. Each of the ARDYx bits is capable of generating a unique interrupt when set.
- Digital Comparator Event – When a conversion's comparison criteria are met by a configured and enabled digital comparator, the DCMPED bit will be set in the AD1CMPCONn register. Each of the digital comparators is capable of generating a unique interrupt when its DCMPED bit is set.
- Oversampling Filter Data Ready Event – When an oversampling filter has completed the accumulation/decimation process and has stored the result, the AFRDY bit will be set in the AD1FLTRn register. Each of the Oversampling Filters is capable of generating a unique interrupt when its AFRDY bit is set.

18.6.2 Interrupt Enabling, Priority and Vectoring

Each of the ADC events previously mentioned will generate an interrupt when its associate Interrupt Enable bit, IE, is set. An Interrupt Flag bit, IF, priority bits, IP<2:0>, and sub-priority bits IS<1:0> are also associated with each of the events. For more information on how to enable and prioritize interrupts, refer to **Section 8. “Interrupts”** (DS61108) in the *“PIC32 Family Reference Manual”*. Each of the ADC events previously listed also has an associated interrupt vector. Refer to the **“Interrupt Controller”** chapter in the specific device data sheet for more information on the vector location and control/status bits associated with each individual interrupt.

18.6.3 Individual and Global Interrupts

The use of the individual interrupts previously listed can significantly optimize the servicing of multiple ADC events, by keeping each Interrupt Service Routine (ISR) focused on efficiently handling a specific event. In addition, different ISRs can be easily segregated according to the tasks performed, thereby making user software easier to implement and maintain.

There may be cases where it is desirable to have a single ISR service multiple interrupt events. To facilitate this, each ADC event can be logically “ORed” to create a single global ADC interrupt. When an ADC event is enabled for a global interrupt, it will vector to a single interrupt routine. It will be the responsibility of this single global ISR to determine the source of the interrupt through polling and process it accordingly.

For the ADC Events:

- Each ANx Data Ready interrupt event is included in the Global ADC Interrupt only if the corresponding AGIENx bit (located in the AD1GIRQEN1 or AD1GIRQEN2 register) is set
- The Digital Comparator interrupt event is included in the Global ADC Interrupt only if the DCMPGIEN bit (AD1CMPCON<6>) is set
- The Oversampling Filter interrupt event is included in the Global ADC Interrupt only if the AFGIEN bit (AD1FLTRn<25>) is set

Use of the Global Interrupt requires configuration of its own unique IE, IF, IP and IS bits as well as configuration of its interrupt vector as described in **18.6.2 “Interrupt Enabling, Priority and Vectoring”**.

Interrupts for the ADC can be configured as individual or global, or utilize both where some are processed individually and others in the global ISR.

18.6.4 Early Interrupts

The EIE<2:0> bits in the AD1CON1 register enable the generation of the interrupts prior to completion of the conversion. Even though the input is still in the conversion process, the processor application software can use the "head-start" to begin execution of the entry into the ISR. The early interrupt can improve the throughput of a system by overlapping the completion of the ADC conversion with the processor overhead associated with an interrupt. The use of the EIE<2:0> bits can reduce the latency from the moment the analog signal was sampled until the point in time when the user application software can use the data.

Use of the EIE<2:0> bits will cause the ARDYx bits in the AD1STATx register to be set prior to the data actually being available in the AD1DATAx register. The value stored in the EIE<2:0> bits determines the number of TAD clocks cycles that the bit will be set prior to the completion of the conversion. The EIE<2:0> bits setting applies to all ARDYx bits of the ADC. Early interrupts should not be used if the application uses polling to determine if a conversion is complete. This includes polling inside a Global ISR. In addition, note that the EIE<2:0> bits setting does not apply to the Oversampling Filter Data Ready signal, AFRDY.

> **Note:** The early interrupt feature should only be used when the data is retrieved using an individual interrupt routine and not when polling the ARDYx bits, even in the Global interrupt service routine.

18.7 OPERATION DURING POWER-SAVING MODES

The power-saving modes, Sleep and Idle, are useful for reducing the conversion noise by minimizing the digital activity of the CPU, buses and other peripherals.

18.7.1 Sleep Mode

When a device enters Sleep mode, the SYSCLK and all peripherals that operate from the SYSCLK source are halted. This includes the ADC when the SYSCLK is selected for the clock source or when REFCLK3 is selected for the ADC clock source, and SYSCLK is used as the REFCLK3 source.

When the SYSCLK is the source, (directly or indirectly) and Sleep mode occurs during a conversion, the conversion is aborted. The converter will not resume a partially completed conversion on exiting from Sleep mode. The ADC register contents are not affected by the device entering or leaving Sleep mode.

The ADC module can operate during Sleep mode if the ADC clock source is derived from a source other than SYCLK that is active during sleep mode. The FRC clock source is a logical choice for operation during sleep, however the REFCLK3 clock source can also be used provided it has an input clock that is operational during sleep mode.

ADC operation during Sleep mode reduces the digital switching noise from the conversion. When the conversion is completed, the ARDYx status bit for that analog input will be set and the result will be loaded into the corresponding ADC Result register (AD1DATAn).

If any of the ADC interrupts is enabled (AD1IE = 1), the device will wake-up from Sleep mode when the ADC interrupt occurs. The program execution will resume at the ADC ISR, if the ADC interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the WAIT instruction that placed the device in Sleep mode.

To minimize the effects of digital noise on the ADC module operation, the user must select a conversion trigger source that ensures that the A/D conversion will take place in Sleep mode. For example, the external interrupt pin (INT0) conversion trigger option (TRGSRCn<4:0> = 00100) can be used for performing sampling and conversion while the device is in Sleep mode.

Note: For the ADC module to operate in Sleep mode, the ADC clock source must be set to Internal FRC (ADCSEL<1:0> bits (AD1CON2<9:8>) = 11). Alternately, the REFCLK3 source can be used; however, the clock source used for REFCLK3 must operate during Sleep. Any changes to the ADC clock configuration require that the ADC be disabled.

18.7.2 ADC Operation during Idle Mode

For the ADC, the Stop in Idle Mode bit, ADSIDL (AD1CON1<13>), specifies whether the ADC module will stop on Idle or continue on Idle. If ADSIDL = 0, the ADC module will continue normal operation when the device enters Idle mode. If any of the ADC interrupts are enabled, the device will wake-up from Idle mode when the ADC interrupt occurs. The program execution will resume at the ADC ISR if the ADC interrupt is greater than the current CPU priority. Otherwise, execution will continue from the instruction after the WAIT instruction that placed the device in Idle mode.

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

18.7.3 ADC Low-power Mode

The ADC module can be placed in a low-power state by setting the ADC Low-power bit, LOWPWR (AD1CON2<13>). Using this low-power mode provides a significantly faster module restart compared to disabling and re-enabling the ADC module using the ADCEN bit (AD1CON1<15>). This is because disabling and re-enabling the ADC module using the ADCEN bit (AD1CON1<15>) requires a long ADC calibration sequence to be performed. Whereas, restarting the ADC module after clearing the LOWPWR bit (AD1CON2<13>) only requires 2 * T_{AD} cycles as the ADC bias generators are not turned off in ADC Low-power mode.

Note: The first five conversions following the exit from ADC Low-power mode may be subject to lower accuracy than specified in the device data sheet.

18.8 EFFECTS OF RESET

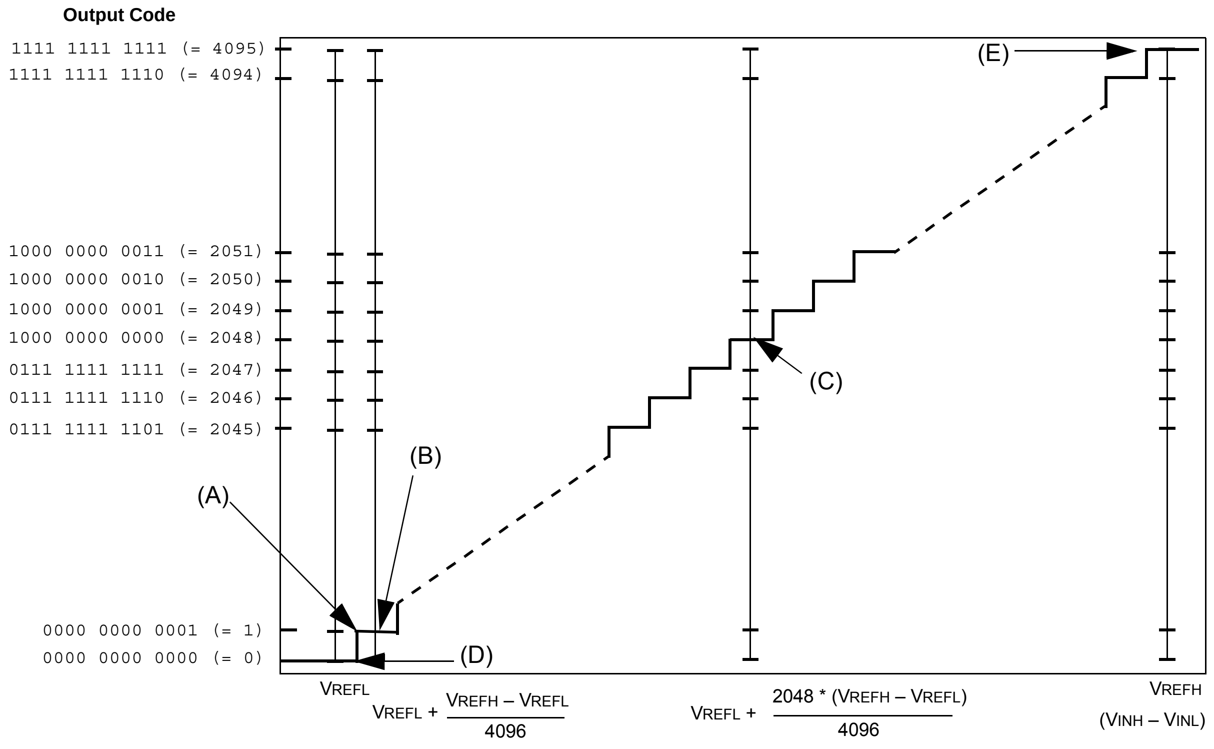
Following any Reset event, all the ADC control and status registers are reset to their default values with control bits in a non-active state. This disables the ADC module and sets the analog input pins to Analog Input mode. Any conversion that was in progress will terminate and the result will not be written to the result buffer. The values in the AD1DATAn registers are initialized to 0x00000000 during a device Reset.

18.9 TRANSFER FUNCTION

A typical transfer function of the 12-bit ADC is illustrated in Figure 18-21. The difference of the input voltages (VINH - VINL) is compared with the reference (VREFH - VREFL).

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

Figure 18-21: Analog-to-Digital Transfer Function



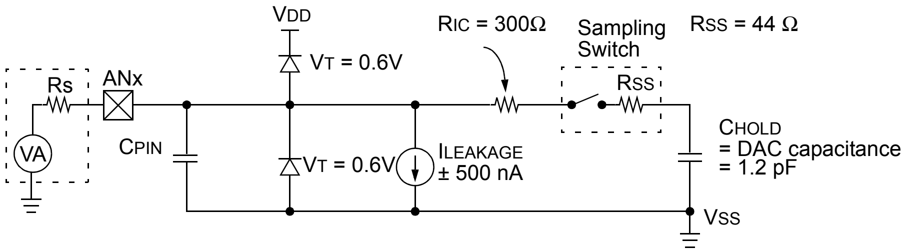
18.10 ADC SAMPLING REQUIREMENTS

The analog input model of the 12-bit ADC is illustrated in Figure 18-22. The total acquisition time for the analog-to-digital conversion is a function of the internal circuit settling time and the holding capacitor charge time.

For the ADC module to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the voltage level on the analog input pin. The analog output source impedance (RS), the interconnect impedance (RIC), and the internal sampling switch (RSS) impedance combine to directly affect the time required to charge the CHOLD. The combined impedance of the analog sources must therefore be small enough to fully charge the holding capacitor within the selected sample time. The internal holding capacitor will be in the discharged state prior to each sample operation.

At least 1 TAD time period should be allowed between conversions for the acquisition time. Refer to the **"Electrical Characteristics"** chapter in the specific device data sheet for more information.

Figure 18-22: 12-bit ADC Analog Input Model



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

Legend:

CPIN = Input capacitance

RSS = Sampling switch resistance

RS = Source resistance

ILEAKAGE = Leakage current at the pin due to various junctions

VT = Threshold voltage

RIC = Interconnect resistance

CHOLD = Sample/hold capacitance

18.11 CONNECTION CONSIDERATIONS

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

An external RC filter is sometimes added for anti-aliasing of the input signal. The R (resistive) component should be selected to ensure that the acquisition time is met. Any external components connected (through high-impedance) to an analog input pin (capacitor, Zener diode, and so on) should have very little leakage current at the pin.

18.12 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC32 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 Pipelined Analog-to-Digital Converter (ADC) module are:

Title	**Application Note #**
Understanding A/D Converter Performance Specifications	AN693
Achieving Higher ADC Resolution Using Oversampling	AN1152

Note: Please visit the Microchip Web site (www.microchip.com) for additional Application Notes and code examples for the PIC32 family of devices.

18.13 REVISION HISTORY

Revision A (May 2013)

This is the initial released version of the document.

Revision B (November 2013)

This revision includes the following updates:

- The Shared S&H 5 Block Diagram was updated (see Figure 18-3)
- Note 4 was added to the STRGSRC<4:0> bits in the ADC1 Control Register 1 (AD1CON1) (see Register 18-1)
- The SHxALT bits in the ADC1 Input Mode Control Register (AD1MOD) were updated (see Table 18-1 and Register 18-4)
- Added example code (see Example 18-1 through Example 18-5)
- Minor updates to text and formatting were incorporated throughout the document

NOTES:

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

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

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

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

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

═ ISO/TS 16949 ═

Trademarks

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

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

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

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

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

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

♻ Printed on recycled paper.

ISBN: 978-1-62077-653-7

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

Austin, TX
Tel: 512-257-3370

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
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

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

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

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
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: 91-20-3019-1500

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: 886-7-213-7830

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 - Dusseldorf
Tel: 49-2129-3766400

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

Germany - Pforzheim
Tel: 49-7231-424750

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

Italy - Venice
Tel: 39-049-7625286

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

Poland - Warsaw
Tel: 48-22-3325737

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

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

10/28/13