
第 18 章 12 位流水线型模数转换器 (ADC)

目录

本章包括下列主题：

18.1 简介	18-2
18.2 控制寄存器	18-5
18.3 ADC 工作原理、术语和转换序列	18-30
18.4 ADC 模块配置	18-39
18.5 更多 ADC 功能	18-53
18.6 中断	18-63
18.7 节能模式下的操作	18-65
18.8 复位的影响	18-66
18.9 传递函数	18-66
18.10 ADC 采样要求	18-67
18.11 连接注意事项	18-67
18.12 相关应用笔记	18-68
18.13 版本历史	18-69

注： 本系列参考手册章节旨在用作对器件数据手册的补充。根据不同的器件型号，本手册章节可能并不适用于所有 PIC32 器件。
请参见当前器件数据手册中“**12 位模数转换器 (ADC)**”章节开始处的注释，以查看本文档是否支持您当前使用的器件。
器件数据手册和系列参考手册的各章节可从 Microchip 网站下载 <http://www.microchip.com>。

18.1 简介

PIC32 12 位流水线型模数转换器 (Analog-to-Digital Converter, ADC) 具有以下特性：

- 12 位分辨率
- 6 级转换和 4 级处理流水线
- 最短 357 ns 的转换延时：
 - 最高 28 Msps 的总转换速率
- 外部参考电压输入引脚
- 6 个采样保持 (Sample and Hold, S&H) 电路 SH0 至 SH5：
 - 5 个专用 S&H 电路，具有单独的输入选择和单独的转换触发源选择，可实现高速转换
 - 1 个共用 S&H 电路，具有自动输入扫描模式和公共转换触发源选择
- 除内部参考电压和内部带隙温度传感器外，最多还有 48 个模拟输入源
- 每个模拟输入具有独立的 32 位转换结果寄存器
 - 转换结果的格式可以为无符号或有符号数据
- 6 个数字比较器：
 - 多个比较选项
 - 可分配给特定模拟输入
- 6 个过采样滤波器：
 - 提供较高的分辨率
 - 可分配给特定模拟输入
- 可在 CPU 休眠和空闲模式下工作

图 18-1 给出了 ADC1 模块的简化框图。图 18-2 和图 18-3 中给出了相关的 S&H 电路图。

在 5 个专用 S&H 电路中，模拟输入通过多路开关连接并切换到 S&H 电容。这些多路开关允许在 S&H 电路中进行输入和配置选择。配置设置 SHxALT<1:0> 和 SHxMOD<1:0> 用于配置多路开关，并且在转换开始时与模拟采样一起被传递至 6 级流水线转换器。转换器使用数字配置信息处理模拟采样，这可使其了解数字格式、测量模式和采样收集自哪个 ANx 输入。当转换完成时，最终结果将保存在特定模拟输入的结果缓冲区，并传递至数字滤波器和数字比较器（如果配置为使用该特定采样的数据）。

单个共用 S&H 电路的输入端集成了一个较大多路开关。当专用 S&H 电路使用单个输入（或其交替输入）并要用于对时间敏感或瞬态信号进行高速和精确的采样时，共用 S&H 连接至所有剩余输入，可使用输入扫描逻辑自动扫描大批输入，从而提供更大灵活性。

关于专用和共用 S&H 电路的详细信息在 18.3.2 “专用采样和保持”和 18.3.3 “共用采样和保持”中进行了说明。模拟输入和其与专用和共用 S&H 电路的关联构成 3 种类型的输入。专用 S&H 的输入为第 1 类，共用 S&H 的输入为第 2 类或第 3 类。18.3.1 “模拟输入”对这 3 类输入进行了说明。18.3.4 “6 级流水线转换器”介绍了流水线转换器。

图 18-1: ADC1 模块框图

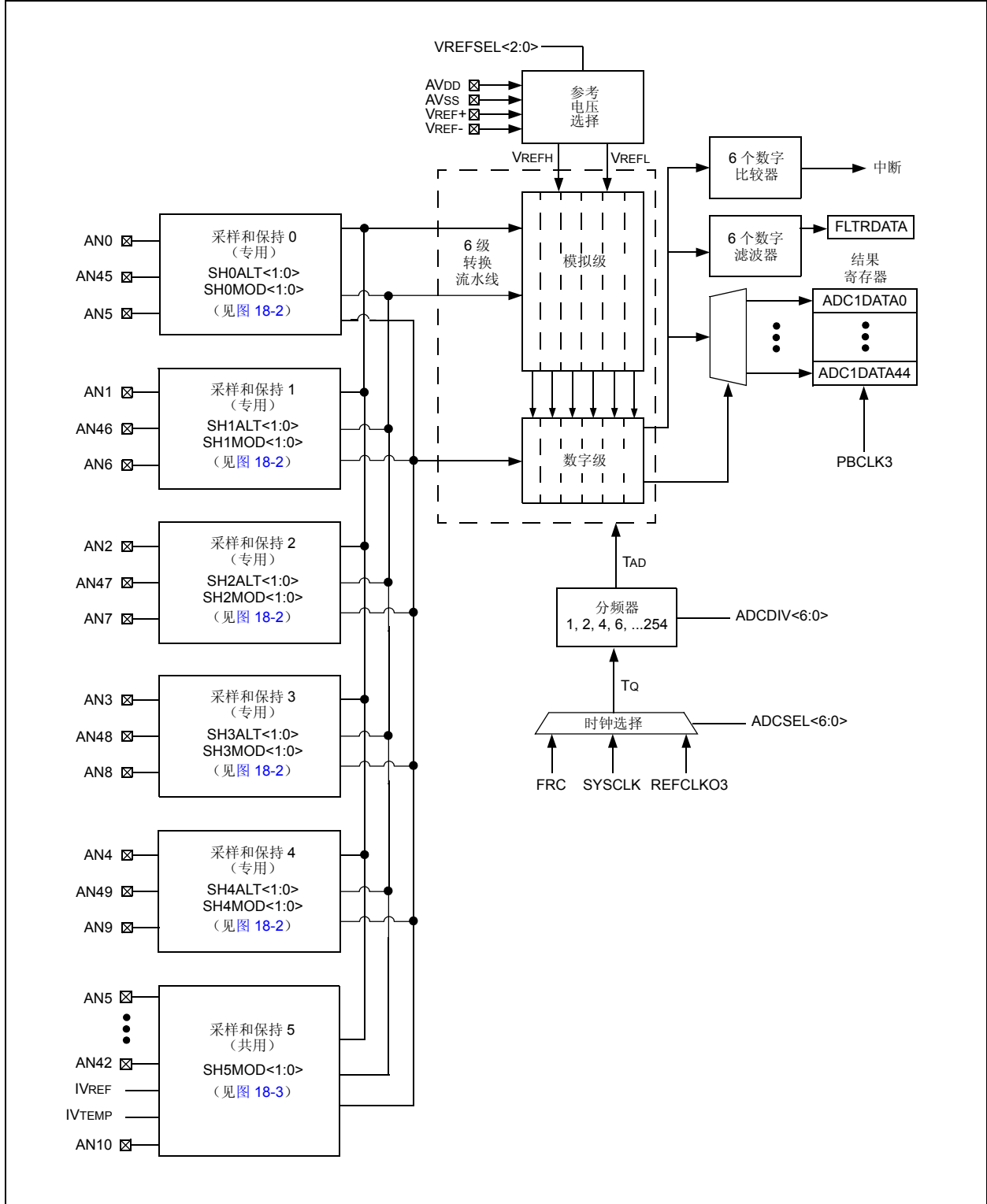


图 18-2: 专用 S&H 0-4 框图

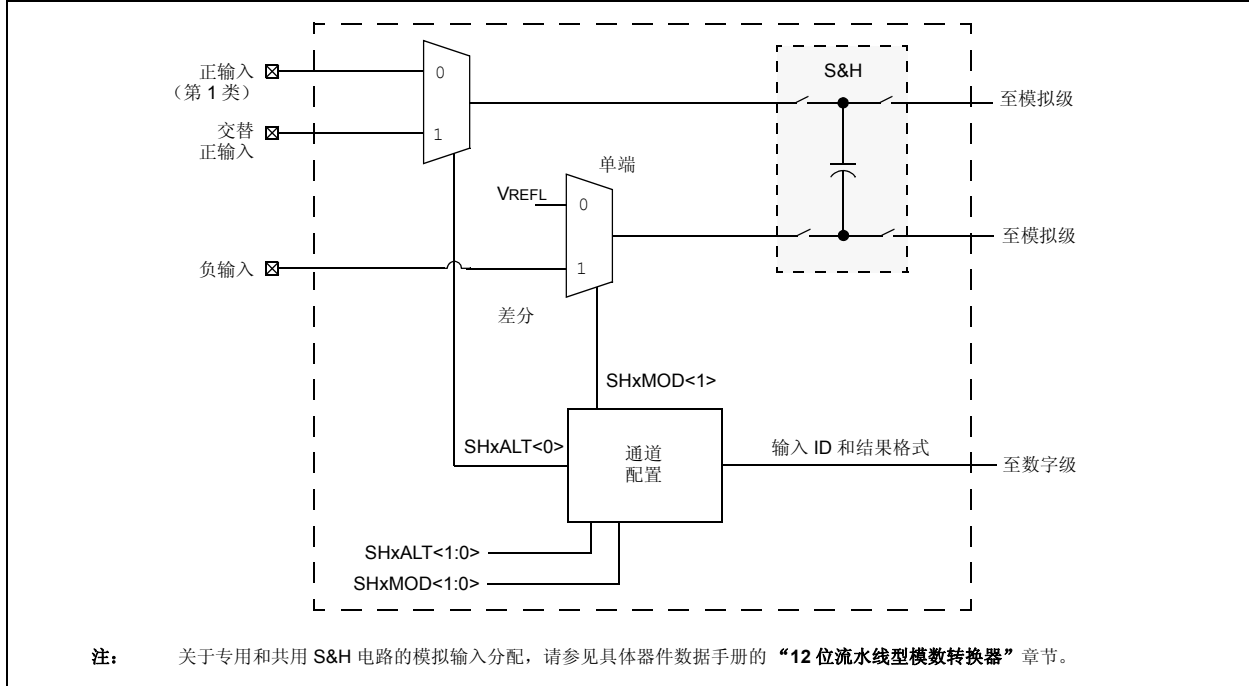
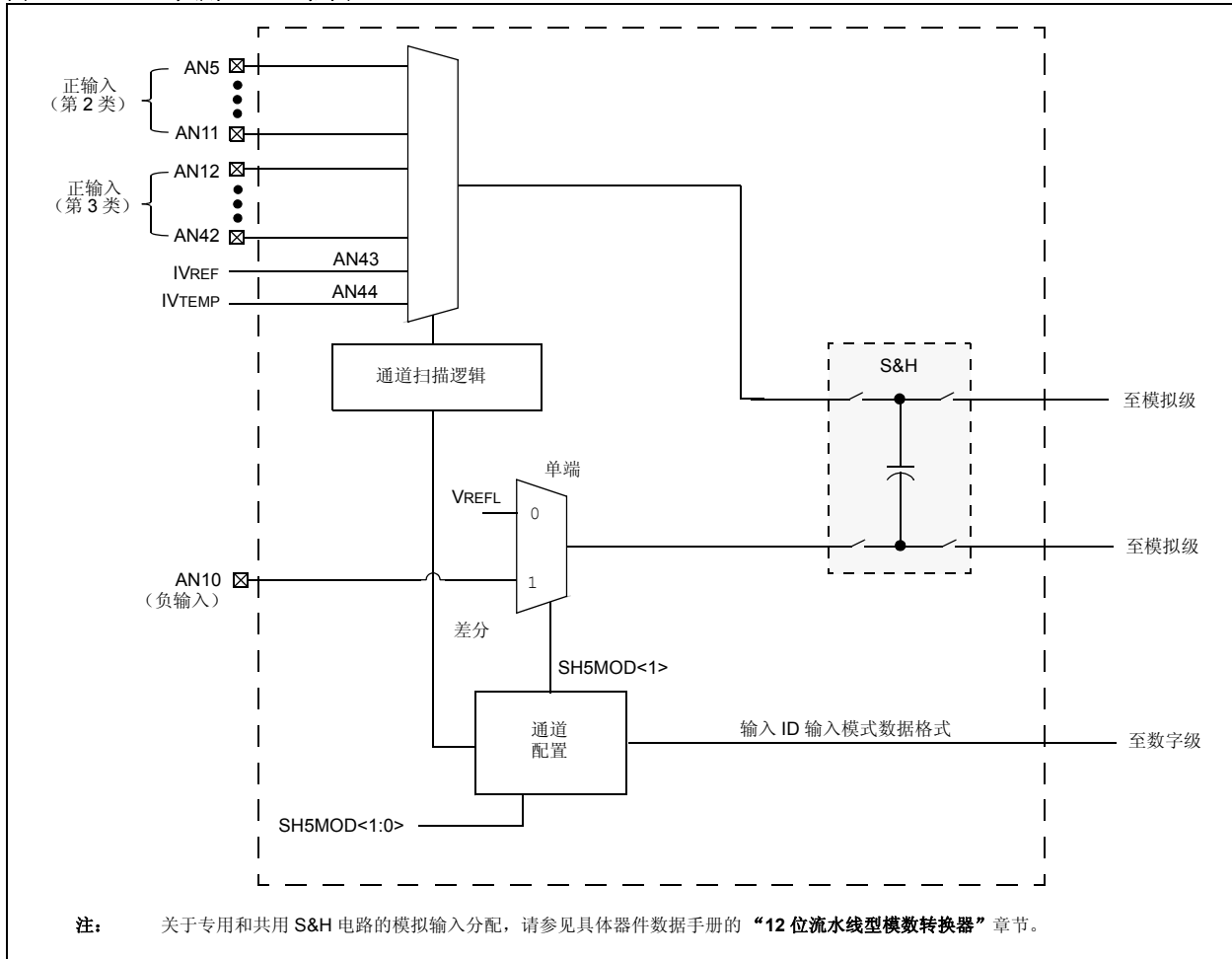


图 18-3: 共用 S&H 5 框图



18.2 控制寄存器

PIC32 12 位流水线型 ADC 模块具有以下特殊功能寄存器 (Special Function Register, SFR):

- **AD1CON1: ADC1 控制寄存器 1**

该寄存器用于控制 ADC 模块的基本操作, 包括休眠和空闲模式下的行为以及数据格式。该寄存器指定中断控制器的向量移位置。AD1CON1 的其他功能包括提前中断产生和已扫描模拟输入的触发信号选择。

- **AD1CON2: ADC1 控制寄存器 2**

该寄存器用于控制 ADC 模块的时钟源、用于共用 S&H 电路的时钟分频器和自动采样时间。该寄存器控制和优化 ADC 内核电路的性能, 还包含一个控制位, 用于将 ADC 模块置于特殊低功耗状态。

- **AD1CON3: ADC1 控制寄存器 3**

该寄存器使用户软件可明确请求特定模拟输入的转换。CAL 位可用于手动启动 ADC 校准过程。

- **AD1IMOD: ADC1 输入模式控制寄存器**

这些寄存器使用户可为 5 个专用 S&H 电路 (SH0 至 SH4) 中的每一个选择交替模拟输入。另外, AD1IMOD 寄存器还使用户可选择单端和差分模式以及有符号和无符号数据格式。

- **AD1GIRQEN1: ADC1 全局中断允许寄存器 1 和
AD1GIRQEN2: ADC1 中断允许寄存器 2**

这些寄存器用于指定哪个单独输入转换中断可以产生全局 ADC1 中断。

- **AD1CSS1: ADC1 输入扫描选择寄存器 1 和
AD1CSS2: ADC1 输入扫描选择寄存器 2**

这些寄存器用于指定要通过公共扫描触发信号扫描的模拟输入。

- **AD1DSTAT1: ADC1 数据就绪状态寄存器 1 和
AD1DSTAT2: ADC1 数据就绪状态寄存器 2**

这些寄存器包含单独模拟输入转换的中断状态。每个位表示其相关转换结果的数据就绪状态。

- **AD1CMPCONn: ADC1 数字比较器控制寄存器 n (n = 1、2、3、4、5 或 6)**

这些寄存器用于控制数字比较器的操作, 包括产生中断和要使用的比较条件。该寄存器还提供了比较器事件发生时的状态。

- **AD1CMPENn: ADC1 数字比较器使能寄存器 n (n = 1、2、3、4、5 或 6)**

这些寄存器用于选择将通过数字比较器处理的模拟输入转换结果。

- **AD1CMPn: ADC1 数字比较器寄存器 n (n = 1、2、3、4、5 或 6)**

这些寄存器包含数字比较器使用的数字比较上限值和下限值。

- **AD1FLTRn: ADC1 滤波器寄存器 n (n = 1、2、3、4、5 或 6)**

这些寄存器提供了过采样滤波器累加器的控制位和状态位, 也包含了 16 位滤波器输出数据。

- **AD1TRGR1: ADC1 输入转换控制寄存器 1**

该寄存器用于控制模拟输入 AN0 至 AN3 的触发源选择。

- **AD1TRGR2: ADC1 输入转换控制寄存器 2**

该寄存器用于控制模拟输入 AN4 至 AN7 的触发源选择。

- **AD1TRGR3: ADC1 输入转换控制寄存器 3**

该寄存器用于控制模拟输入 AN8 至 AN11 的触发源选择。

- **AD1DATAn: ADC1 数据输出寄存器 (n = 0 到 44)**

这些寄存器是模数转换输出数据寄存器。AD1DATAn 寄存器与每个模拟输入 n 相关。

- **AD1CALx: ADC1 校准寄存器 (x = 1-5)**

这些寄存器包含 ADC 校准值。

ADC 模块还具有以下用于中断控制的相关位（有关这些位的位置的更多信息，请参见具体器件数据手册的“**中断控制器**”章节）：

- 中断请求标志状态位 (AD1IF)
- 中断允许控制位 (AD1IE)
- 中断优先级控制位 (AD1IP<2:0>) 和 (AD1IS<1:0>)

表 18-1 汇总了所有 ADC 特殊功能寄存器 (SFR)。该汇总表之后列出了相应的寄存器，并对每个位进行了详细说明。

表 18-1: ADC SFR 汇总

寄存器名称	位范围	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>						STRGSR<4:0>						EIE<2:0>				
	15:0	ADCN	—	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	

图注: — = 未实现, 读为 0。

表 18-1: ADC SFR 汇总 (续)

寄存器名称	位范围	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>					

图注: — = 未实现, 读为 0。

表 18-1: ADC SFR 汇总 (续)

寄存器名称	位范围	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>															

图注: — = 未实现, 读为 0。

寄存器 18-1: AD1CON1: ADC1 控制寄存器 1

位范围	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
	—	—	—	—	—	—	—	—

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-27 **FILTRDLY<4:0>**: 过采样数字滤波器延时位

指定使用过采样数字滤波器时的后续自动触发的采样时间。采样时间为 $1.5 + \text{FILTRDLY}<4:0> \text{ TAD}$ 。

11111 = 采样时间为 32.5 TAD

11110 = 采样时间为 31.5 TAD

·
·
·

00001 = 采样时间为 2.5 TAD

00000 = 采样时间为 1.5 TAD

bit 26-22 **STRGSRC<4:0>**: 扫描触发源选择位 ^(1,4)

11111 = 保留

·
·
·

01101 = 保留

01100 = 比较器 2 COUT

01011 = 比较器 1 COUT

01010 = OCMP5

01001 = OCMP3

01000 = OCMP1

00111 = TMR5 匹配信号

00110 = TMR3 匹配信号

00101 = TMR1 匹配信号

00100 = INTO

00011 = 保留

00010 = 保留

00001 = 全局软件触发 (GSWTRG)

00000 = 无触发信号

- 注 1: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。
- 2: 当查询任一 ARDYx 位以确定转换是否完成时, 不应使用提前中断功能。仅当使用单独的中断程序来检索 ADC 模块的所有结果以获取 ADC 结果时, 才应使用提前中断。
- 3: ADCEN 位仅应在配置 ADC 模块之后置 1。ADCEN = 1 时更改 ADC 配置位 (如 ADCSEL<1:0> 和 ADCDIV<6:0>) 将导致不可预测的行为。当 ADCEN = 0 时, 禁止 ADC 时钟, 复位内部控制逻辑并且清零模块所使用的所有状态标志。但是, 可读写 SFR。
- 4: 使用扫描触发源需要设置 AD1TRGR1、AD1TRGR2 和 / 或 AD1TRGR3 寄存器中的指定通道。

注: ADCRDY 位 (AD1CON2<31>) 置 1 前, ADC 模块不可正常工作。

寄存器 18-1: AD1CON1: ADC1 控制寄存器 1 (续)

bit 21-19 **未实现:** 读为 0

bit 18-16 **EIE<2:0>:** 提前中断允许位 (2)

这些位用于选择有效数据实际到达之前的将相关的 ARDYx 位置 1 的时钟数。由于 ARDYx 位会触发中断，因此当使用单独的中断程序来获取每个 ADC 结果时，这些位允许产生提前中断。

111 = 在数据就绪之前的 7 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

110 = 在数据就绪之前的 6 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

101 = 在数据就绪之前的 5 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

100 = 在数据就绪之前的 4 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

011 = 在数据就绪之前的 3 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

010 = 在数据就绪之前的 2 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

001 = 在数据就绪之前的 1 个 TAD 时钟时将相关的数据就绪位 ARDYx 置 1

000 = 数据就绪时将相关的数据就绪位 ARDYx 置 1

bit 15 **ADCEN:** ADC 工作模式位 (3)

1 = 使能 ADC 模块

0 = 禁止 ADC 模块

bit 14 **未实现:** 读为 0

bit 13 **ADSIDL:** 空闲模式停止位

1 = 当器件进入空闲模式时，模块停止工作

0 = 模块在空闲模式下继续工作

bit 12 **未实现:** 读为 0

bit 11 **FRACT:** 小数数据输出格式位

关于具体格式信息，请参见第 18.4.5 节“选择 ADC 结果格式”。

1 = 小数

0 = 整数

bit 10-0 **未实现:** 读为 0

- 注 1:** 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。
- 2:** 当查询任一 ARDYx 位以确定转换是否完成时，不应使用提前中断功能。仅当使用单独的中断程序来检索 ADC 模块的所有结果以获取 ADC 结果时，才应使用提前中断。
- 3:** ADCEN 位仅应在配置 ADC 模块之后置 1。ADCEN = 1 时更改 ADC 配置位（如 ADCSEL<1:0> 和 ADCDIV<6:0>）将导致不可预测的行为。当 ADCEN = 0 时，禁止 ADC 时钟，复位内部控制逻辑并且清零模块所使用的所有状态标志。但是，可读写 SFR。
- 4:** 使用扫描触发源需要设置 AD1TRGR1、AD1TRGR2 和 / 或 AD1TRGR3 寄存器中的指定通道。

注: ADCRDY 位 (AD1CON2<31>) 置 1 前，ADC 模块不可正常工作。

寄存器 18-2: AD1CON2: ADC1 控制寄存器 2

位范围	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⁾						

图注:	HS = 由硬件置 1	HC = 由硬件清零	r = 保留
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 31 **ADCRDY:** ADC 就绪位 ⁽¹⁾

1 = ADC 模块已就绪, 可正常工作
0 = ADC 未就绪, 不可使用

bit 30-24 **未实现:** 读为 0

bit 23-16 **SAMC<7:0>:** 共用 S&H 的采样时间位

11111111 = 256 个 TAD

.

.

00000001 = 2 个 TAD

00000000 = 1 个 TAD

该位域指定分配给共用 S&H 电路的 ADC 采样时间的 ADC 时钟周期数。

bit 15 **未实现:** 读为 0

bit 14 **BOOST:** 参考电压升压位

1 = 升压 VREF
0 = 不升压 VREF

关于 VREF 规范, 请参见具体器件数据手册的“电气特性”章节。

当参考电压 VREF (VREFH - VREFL) 小于 $0.65 * (AVDD - AVSS)$ 时, 将该位置 1 可使信噪比 (Signal-to-Noise Ratio, SNR) 最大化。更多信息, 请参见 18.4.7 “选择参考电压源”。更改该位的状态时, 需要将 CAL 位 (AD1CON3<31>) 置 1 重新校准 ADC 模块。

bit 13 **LOWPWR:** ADC 低功耗位

1 = 强制 ADC 模块进入低功耗状态。低功耗状态允许清零该位后可立即工作, 而无需校准周期。
0 = 退出 ADC 低功耗状态

bit 12-11 **未实现:** 读为 0

bit 10 **保留:** 该单元始终写入 0

注 1: 当 ADCEN (AD1CON1<15>) = 0 时, 该位设置为 0。

2: 这些位应在将 ADCEN 位置 1 (AD1CON1<15> = 1) 以使能 ADC 模块之前进行配置。

3: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-2: AD1CON2: ADC1 控制寄存器 2 (续)

bit 9-8 **ADCSEL<1:0>**: ADC 时钟源 (TQ) 位 ^(2,3)

- 11 = FRC 振荡器输出
- 10 = REFCLK3
- 01 = 系统时钟 (TcY)
- 00 = 保留

bit 7 **未实现**: 读为 0

bit 6-0 **ADCDIV<6:0>**: ADC 输入时钟分频比位 ⁽²⁾

这些位用于将所选的时钟源分频以产生所需的 ADC 时钟速率 (TAD)。

$$11111111 = 2 TQ * (ADCDIV<6:0>) = 254 * TQ = TAD$$

·
·
·

$$00000111 = 2 TQ * (ADCDIV<6:0>) = 6 * TQ = TAD$$

$$00000110 = 2 TQ * (ADCDIV<6:0>) = 4 * TQ = TAD$$

$$00000011 = 2 TQ * (ADCDIV<6:0>) = 2 * TQ = TAD$$

$$00000000 = TQ = TAD$$

- 注**
- 1: 当 ADCEN (AD1CON1<15>) = 0 时, 该位设置为 0。
 - 2: 这些位应在将 ADCEN 位置 1 (AD1CON1<15> = 1) 以使能 ADC 模块之前进行配置。
 - 3: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-3: AD1CON3: ADC1 控制寄存器 3

位范围	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⁾					

图注:

R = 可读位

W = 可写位

HC = 由硬件清零

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31 **CAL:** 校准位

在校准过程中, AD1CON2 寄存器中的 ADCRDY 状态位清零。校准完成时, 清零 CAL 位并将 ADCRDY 状态位置 1。

1 = ADC 模块正在执行校准周期 (校准大约需要 160 个 TAD 时钟周期)

0 = 校准周期未在进行中

bit 30 **GSWTRG:** 全局软件触发位

1 = 为已选择 GSWTRG 位作为触发信号 (通过 AD1TRGn 寄存器中的相关 TRGSRC<4:0> 位或 AD1CON1 寄存器中的 STRGSRC<4:0> 位) 的 ADC 输入触发模数转换

0 = 该位在 2 个 PBCLK3 时钟周期内自动清零

bit 29 **RQCONVRT:** 单独 ADC 输入转换请求位

该位和其相关 ADINSEL<5:0> 位使用户可单独请求模拟输入的模数转换, 而无需重新设定 TRGSRC<4:0> 位或 STRGSRC<4:0> 位。这在调试或错误处理以下情况时非常有用: 用户软件需要立即获取特定输入的 ADC 结果。

1 = 触发由 ADINSEL<5:0> 位指定的选定 ADC 输入的转换

0 = 不触发转换, 该位自动清零

bit 28-13 **未实现:** 读为 0

bit 12-10 **VREFSEL<2:0>:** VREF 输入选择位 ⁽²⁾

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

bit 9-6 **未实现:** 读为 0

注 1: 请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定哪些模拟输入可用。

注 2: 这些位应在将 ADCEN 位置 1 (AD1CON1<15> = 1) 以使能 ADC 模块之前进行配置。

寄存器 18-3: AD1CON3: ADC1 控制寄存器 3 (续)

bit 5-0 **ADINSEL<5:0>**: ADC 输入选择位⁽¹⁾

当 RQCONVRT 位置 1 时, 该二进制编码位域选择要转换的 ADC 模块输入。

111111 = 保留

•

•

•

101101 = 保留

101100 = IVTEMP (内部温度参考电压)

101011 = IVREF (内部参考电压)

101010 = AN42

•

•

•

000010 = AN2

000001 = AN1

000000 = AN0

- 注 1: 请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定哪些模拟输入可用。
2: 这些位应在将 ADCEN 位置 1 (AD1CON1<15> = 1) 以使能 ADC 模块之前进行配置。

PIC32 系列参考手册

寄存器 18-4: AD1IMOD: ADC1 输入模式控制寄存器

位范围	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>	

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 31-26 **未实现:** 读为 0
- bit 25-24 **SH4ALT<1:0>:** 专用 S&H 4 (SH4) 的模拟输入选择位
 - 11 = 保留; 未使用
 - 10 = 保留; 未使用
 - 01 = 交替输入
 - 00 = 默认第 1 类输入 AN4
- bit 23-22 **SH3ALT<1:0>:** 专用 S&H 3 (SH3) 的模拟输入选择位
 - 11 = 保留; 未使用
 - 10 = 保留; 未使用
 - 01 = 交替输入
 - 00 = 默认第 1 类输入 AN3
- bit 21-20 **SH2ALT<1:0>:** 专用 S&H 2 (SH2) 的模拟输入选择位
 - 11 = 保留; 未使用
 - 10 = 保留; 未使用
 - 01 = 交替输入
 - 00 = 默认第 1 类输入 AN2
- bit 19-18 **SH1ALT<1:0>:** 专用 S&H 1 (SH1) 的模拟输入选择位
 - 11 = 保留; 未使用
 - 10 = 保留; 未使用
 - 01 = 交替输入
 - 00 = 默认第 1 类输入 AN1
- bit 17-16 **SH0ALT<1:0>:** 专用 S&H 0 (SH0) 的模拟输入选择位
 - 11 = 保留; 未使用
 - 10 = 保留; 未使用
 - 01 = 交替输入
 - 00 = 默认第 1 类输入 AN0

注

- 1: 交替输入仅可用于第 1 类输入。请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定可用的第 1 类输入和这些输入的交替输入。
- 2: 当选择交替输入 (SHxALT<1:0> ≠ 0) 时, 仍使用默认第 1 类输入的数据、状态和控制寄存器。选择交替输入仅更改物理输入源。
- 3: SHxMOD<1:0> 同时选择负输入的数据输出格式和输入选择。请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定可用于负输入的 ANx 输入选项。
- 4: 一些位可能并未存在于所有器件上。关于可用性, 请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节。

寄存器 18-4: AD1IMOD: ADC1 输入模式控制寄存器 (续)

bit 15-12 未实现: 读为 0

bit 11-10 SH5MOD<1:0>: S&H 5 (SH5) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

bit 9-8 SH4MOD<1:0>: S&H 4 (SH4) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

bit 7-6 SH3MOD<1:0>: S&H 3 (SH3) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

bit 5-4 SH2MOD<1:0>: S&H 2 (SH2) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

bit 3-2 SH1MOD<1:0>: S&H 1 (SH1) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

bit 1-0 SH0MOD<1:0>: S&H 0 (SH0) 的输入配置选择位

- 11 = 差分输入, 二进制补码 (有符号) 格式的数据输出
- 10 = 差分输入, 单极性编码 (无符号) 格式的数据输出
- 01 = 单端输入, 二进制补码 (有符号) 格式的数据输出
- 00 = 单端输入, 单极性编码 (无符号) 格式的数据输出

- 注 1: 交替输入仅可用于第 1 类输入。请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定可用的第 1 类输入和这些输入的交替输入。
- 2: 当选择交替输入 (SHxALT<1:0> ≠ 0) 时, 仍使用默认第 1 类输入的数据、状态和控制寄存器。选择交替输入仅更改物理输入源。
- 3: SHxMOD<1:0> 同时选择负输入的数据输出格式和输入选择。请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定可用于负输入的 ANx 输入选项。
- 4: 一些位可能并未存在于所有器件上。关于可用性, 请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节。

寄存器 18-5: AD1GIRQEN1: ADC1 全局中断允许寄存器 1

位范围	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

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-0 **AGIENx:** ADC 全局中断允许位 (1,2,3)

1 = 数据就绪事件 (ARDYx 位从 0 跳变到 1) 将产生 ADC 全局中断
 0 = 该输入发生数据就绪事件时不会产生全局中断

- 注 1:** 这些位控制一个或多个模拟输入的就绪事件的传播以产生单个全局 (组) 中断。它们不会影响数据就绪事件的单独中断的有效性。通过将 IECx 寄存器中的相应位置 1 允许单独中断和单个全局中断。详情请参见具体器件数据手册的“中断控制器”章节。
- 2:** AGIENx = ANx, 其中 x = 0-31。
- 3:** 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-6: AD1GIRQEN2: ADC1 中断允许寄存器 2

位范围	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

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-13 **未实现:** 读为 0

bit 12-0 **AGIENx:** ADC 全局中断允许位 (1,2,3)

1 = 数据就绪事件 (ARDYx 位从 0 跳变到 1) 将产生 ADC 全局中断

0 = 发生数据就绪事件时不会产生全局中断

通过将 IECx 寄存器中的某一位置 1 允许 ADC 全局中断 (详情请参见具体器件数据手册的“中断控制器”章节)。

- 注 1:** 这些位控制一个或多个模拟输入的就绪事件的传播以产生单个全局 (组) 中断。它们不会影响数据就绪事件的单独中断的有效性。通过将 IECx 寄存器中的相应位置 1 允许单独中断和单个全局中断。详情请参见具体器件数据手册的“中断控制器”章节。
- 2:** AGIENx = ANx, 其中 x = 32-44。
- 3:** 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

PIC32 系列参考手册

寄存器 18-7: AD1CSS1: ADC1 输入扫描选择寄存器 1

位范围	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

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-13 **未实现:** 读为 0

bit 12-0 **CSSx:** ADC 输入扫描选择位 (1,2,3)
 1 = 选择 ANx 进行输入扫描
 0 = 输入扫描时跳过 ANx

注 1: CSSx = ANx, 其中 x = 0-31。

2: 如果第 1 类和第 2 类模拟输入要通过 CSSx 位进行扫描, 那么除了将该寄存器中的相应位置 1 外还必须选择 STRIG 输入作为触发源。请参见 AD1TRGn 寄存器中的位说明 (寄存器 18-15) 来选择 STRIG 选项。

3: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-8: AD1CSS2: ADC1 输入扫描选择寄存器 2

位范围	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

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-13 **未实现:** 读为 0

bit 12-0 **CSSx:** ADC 输入扫描选择位 (1,2,3)
 1 = 选择 ANx 进行输入扫描
 0 = 输入扫描时跳过 ANx

注 1: CSSx = ANx, 其中 x = 32-44。

2: 如果第 1 类和第 2 类模拟输入要通过 CSSx 位进行扫描, 那么除了将该寄存器中的相应位置 1 外还必须选择 STRIG 输入作为触发源。请参见 AD1TRGn 寄存器中的位说明 (寄存器 18-15) 来选择 STRIG 选项。

3: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-9: AD1DSTAT1: ADC1 数据就绪状态寄存器 1

位范围	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

图注: HS = 由硬件置 1 HC = 由硬件清零
 R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-0 **ARDYx:** 相应模拟输入的转换数据就绪位 (1,2,3)
 1 = 当数据在缓冲区中就绪时, 该位置 1。当 IECx 寄存器中相应位置 1 或在 AD1GIRQEN 寄存器中允许 ADC 全局中断时, 将产生中断。
 0 = 读取相关的数据寄存器时, 该位清零。

- 注 1:** ARDYx = ANx, 其中 x = 0-31。
2: 当交替输入用于第 1 类输入时, 主输入的就绪位将指示数据就绪状态。
3: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-10: AD1DSTAT2: ADC1 数据就绪状态寄存器 2

位范围	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

图注: HS = 由硬件置 1 HC = 由硬件清零
 R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-13 **未实现:** 读为 0
 bit 12-0 **ARDYx:** 相应模拟输入的转换数据就绪位 (1,2)
 1 = 当数据在缓冲区中就绪时, 该位置 1。当 IECx 寄存器中的相应位置 1 或者在 AD1GIRQEN 寄存器中允许 ADC 全局中断时, 将产生中断。
 0 = 读取相关的数据寄存器时, 该位清零

- 注 1:** ARDYx = ANx, 其中 x = 32-44。
2: 选择因器件而异。请参见具体器件数据手册的“12 位流水线型模数转换器 (ADC)”章节以确定哪些选择可用。

寄存器 18-11: AD1CMPCONn: ADC1 数字比较器控制寄存器 n (n = 1、2、3、4、5 或 6)

位范围	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

图注:	HS = 由硬件置 1	HC = 由硬件清零
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

bit 31-13 **未实现:** 读为 0

bit 12-8 **AINID<4:0>:** 模拟输入识别 (ID) 位 ⁽¹⁾

发生数字比较器事件 (DCMPED = 1) 时, 这些位可识别产生事件的模拟输入。另外, 如果使能多个输入并且自上次读取之后发生了多个事件, 这些位可识别最后一个产生事件的模拟输入。

bit 7 **ENDCMP:** 数字比较器使能位

1 = 使能数字比较器
0 = 未使能数字比较器, 且 DCMPED 状态位清零

bit 6 **DCMPGIEN:** 数字比较器全局 ADC 中断允许位 ⁽²⁾

1 = 数字比较器事件 (DCMPED 从 0 跳变到 1) 将产生 ADC 全局中断。
0 = 数字比较器事件不会产生 ADC 全局中断。

将 IECx 寄存器中的某一位置 1 允许 ADC 全局中断 (详情请参见具体器件数据手册的“中断控制器”章节。)

bit 5 **DCMPED:** 检测到数字比较器事件状态位

1 = 检测到比较器事件时, 该位由数字比较器硬件置 1。当 IECx 寄存器中的相应位置 1 或者通过 DCMPGIEN 位允许 ADC 全局中断时, 将产生中断。
0 = 读取 AINID<4:0> 位或禁止 ADC 模块时, 该位清零。

bit 4 **IEBTWN:** 介于上限 / 下限间的数字比较器事件位 ⁽²⁾

1 = 当 ADCMPLO<15:0> ≤ DATA<31:0> < ADCMPHI<15:0> 时发生数字比较器事件
0 = 不发生数字比较器事件

bit 3 **IEHIHI:** 高于上限的数字比较器事件位 ⁽²⁾

1 = ADCMPHI<15:0> ≤ DATA<31:0> 时发生数字比较器事件
0 = 不发生数字比较器事件

bit 2 **IEHILO:** 低于上限的数字比较器事件位 ⁽²⁾

1 = 当 DATA<31:0> < ADCMPHI<15:0> 时发生数字比较器事件
0 = 不发生数字比较器事件

bit 1 **IELOHI:** 高于下限的数字比较器事件位 ⁽²⁾

1 = ADCMPLO<15:0> ≤ DATA<31:0> 时发生数字比较器事件
0 = 不发生数字比较器事件

bit 0 **IELOLO:** 低于下限的数字比较器事件位 ⁽²⁾

1 = DATA<31:0> < ADCMPLO<15:0> 时发生数字比较器事件
0 = 不发生数字比较器事件

注 1: ANx 上发生数字比较器事件, 其中 x = AINID<4:0>, 并且 x 的范围为 0-31。

注 2: 使能数字比较器 (ENDCMP = 1) 时更改这些位会导致不可预测的行为。

寄存器 18-12: AD1CMPENn: ADC1 数字比较器使能寄存器 n (n = 1、2、3、4、5 或 6)

位范围	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

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-0 **CMPE31:CMPE0:** ADC1 数字比较器使能位 (1,2)

这些位使对应模拟输入的转换结果由数字比较器处理。CMPE0 使能 AN0, CMPE1 使能 AN1, 以此类推。

- 注** **1:** CMPE_x = AN_x, 其中 x = 0-31 (数字比较器输入限制为 AN0 至 AN31)。
 2: 使能数字比较器 (ENDCMP = 1) 时更改寄存器中的位可导致不可预测的行为。

寄存器 18-13: AD1CMPn: ADC1 数字比较器寄存器 n (n = 1、2、3、4、5 或 6)

位范围	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>							

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 31-16 **ADCMPHI<15:0>**: 数字模拟比较器上限值位
 这些位存储上限值, 该值用于与模数转换数据进行比较。用户将上限值格式化以与相关 S&H 电路的 SHxMOD<1:0> 位和全局 FRACT 位所指定的格式匹配。这些位分别位于 ADxMOD 和 ADxCON1 寄存器中。
- bit 15-0 **ADCMPLO<15:0>**: 数字模拟比较器下限值位
 这些位存储下限值, 该值用于与模数转换数据进行比较。用户将下限值格式化以与相关 S&H 电路的 SHxMOD<1:0> 位和全局 FRACT 位所指定的格式匹配。这些位分别位于 ADxMOD 和 ADxCON1 寄存器中。

注: 使能数字比较器 (ENDCMP = 1) 时更改该寄存器中的这些位会导致不可预测的行为。

PIC32 系列参考手册

寄存器 18-15: AD1TRGR1: ADC1 输入转换控制寄存器 1

位范围	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>				

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-29 **未实现:** 读为 0

bit 28-24 **TRGSRC3<4:0>:** 模拟通道 AN3 转换触发源选择位

- 11111 = 保留
- .
- .
- .
- 01101 = 保留
- 01100 = 比较器 2 COUT
- 01011 = 比较器 1 COUT
- 01010 = OCMP5
- 01001 = OCMP3
- 01000 = OCMP1
- 00111 = TMR5 匹配信号
- 00110 = TMR3 匹配信号
- 00101 = TMR1 匹配信号
- 00100 = INTO
- 00011 = STRIG⁽¹⁾
- 00010 = 保留
- 00001 = 全局软件触发 (GSWTRG)
- 00000 = 无触发信号

bit 23-21 **未实现:** 读为 0

bit 20-16 **TRGSRC2<4:0>:** 模拟通道 AN2 转换触发源选择位
 关于位值定义, 请参见 bit 28-24。

bit 15-13 **未实现:** 读为 0

bit 12-8 **TRGSRC1<4:0>:** 模拟通道 AN1 转换触发源选择位
 关于位值定义, 请参见 bit 28-24。

bit 7-5 **未实现:** 读为 0

bit 4-0 **TRGSRC0<4:0>:** 模拟通道 AN0 转换触发源选择位
 关于位值定义, 请参见 bit 28-24。

注 1: 使用 STRIG 作为触发源将指定该输入使用扫描触发源进行触发。必须设置 STRGSRC<4:0> 位 (AD1CON1<26:22>) 以及 AD1CSS1 和 AD1CSS2 寄存器中的相应的 CSSx 位以进行适当的扫描操作。

寄存器 18-16: AD1TRGR2: ADC1 输入转换控制寄存器 2

位范围	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>				

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

bit 31-29 **未实现:** 读为 0

bit 28-24 **TRGSRC7<4:0>:** 模拟通道 AN7 转换触发源选择位

11111 = 保留

•
•
•

01101 = 保留

01100 = C2OUT

01011 = C1OUT

01010 = OCMP5

01001 = OCMP3

01000 = OCMP1

00111 = TMR5 匹配信号

00110 = TMR3 匹配信号

00101 = TMR1 匹配信号

00100 = INT0

00011 = STRIG⁽¹⁾

00010 = 保留

00001 = 全局软件触发 (GSWTRG)

00000 = 无触发信号

bit 23-21 **未实现:** 读为 0

bit 20-16 **TRGSRC6<4:0>:** 模拟通道 AN6 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

bit 15-13 **未实现:** 读为 0

bit 12-8 **TRGSRC5<4:0>:** 模拟通道 AN5 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

bit 7-5 **未实现:** 读为 0

bit 4-0 **TRGSRC4<4:0>:** 模拟通道 AN4 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

注 1: 使用 STRIG 作为触发源将指定该输入使用扫描触发源进行触发。必须设置 STRGSRC<4:0> 位 (AD1CON1<26:22>) 以及 AD1CSS1 和 AD1CSS2 寄存器中相应的 CSSx 位以进行适当的扫描操作。

寄存器 18-17: AD1TRGR3: ADC1 输入转换控制寄存器 3

位范围	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>				

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-29 **未实现:** 读为 0

bit 28-24 **TRGSRC11<4:0>:** 模拟通道 AN11 转换触发源选择位

- 11111 = 保留
- .
- .
- .
- 01101 = 保留
- 01100 = C2OUT
- 01011 = C1OUT
- 01010 = OCMP5
- 01001 = OCMP3
- 01000 = OCMP1
- 00111 = TMR5 匹配信号
- 00110 = TMR3 匹配信号
- 00101 = TMR1 匹配信号
- 00100 = INTO
- 00011 = STRIG⁽¹⁾
- 00010 = 保留
- 00001 = 全局软件触发 (GSWTRG)
- 00000 = 无触发信号

bit 23-21 **未实现:** 读为 0

bit 20-16 **TRGSRC10<4:0>:** 模拟通道 AN10 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

bit 15-13 **未实现:** 读为 0

bit 12-8 **TRGSRC9<4:0>:** 模拟通道 AN9 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

bit 7-5 **未实现:** 读为 0

bit 4-0 **TRGSRC8<4:0>:** 模拟通道 AN8 转换触发源选择位

关于位值定义, 请参见 bit 28-24。

注 1: 使用 STRIG 作为触发源将指定该输入使用扫描触发源进行触发。必须设置 STRGSRC<4:0> 位 (AD1CON1<26:22>) 以及 AD1CSS1 和 AD1CSS2 寄存器中相应的 CSSx 位以进行适当的扫描操作。

第 18 章 12 位流水线型模数转换器 (ADC)

寄存器 18-18: AD1DATAn: ADC1 数据输出寄存器 (n = 0 到 44)

位范围	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>							

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-0 **DATA<31:0>**: 数据输出值位
 这些位的格式由相关 S&H 电路的 SHxMOD<1:0> 位和 FRACT 位指定。

注: 当交替输入用作第 1 类输入的输入源时, 数据输出仍然从主输入数据输出寄存器中读取。

寄存器 18-19: AD1CALx: ADC1 校准寄存器 (x = 1-5)

位范围	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>							

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 31-0 **ADCAL<31:0>**: ADC 模块的校准数据位

18.3 ADC 工作原理、术语和转换序列

使用 12 位 ADC 进行模数转换包括以下三个步骤：

1. 采样输入信号。
2. 捕捉输入信号（保持）并将其传输到转换器。
3. 将模拟信号转换为其数字表示。

对输入信号的采样包括对 S&H 电路中的电容充电。采样时间必须足够以便电容充电到其值等于输入电压（见 18.10 “ADC 采样要求”）。在适当的时间，输入与电容断开连接，随后模拟电压传送到转换器。然后转换器将模拟信号转换为数字信号并提供相应的结果。

转换器需要一个时钟源和一个参考电压。该时钟被称为 ADC 时钟并且周期为 TAD。可以选择时钟和参考电压源，也可以选择时钟预分频比。

12 位 ADC 使用两种类型的 S&H 电路：专用和共用。每个 ADC 实现包含了最多 5 个专用 S&H 电路和单个共用 S&H 电路。连接到 S&H 电路的输入分为 3 类：第 1 类、第 2 类和第 3 类。

18.3.1 模拟输入

器件上可用的模拟输入引脚分为 3 类：第 1 类、第 2 类和第 3 类。每个模拟输入，不论分类类别，都有自己唯一的数据输出寄存器，用于存储转换结果。

表 18-2: 模拟输入类型

类型	S&H/ 输入类型	触发信号	触发动作
第 1 类	专用 S&H 输入	独立触发源或通道扫描触发源	结束采样并启动转换
第 2 类	共用 S&H 输入	独立触发源或通道扫描触发源	启动采样 / 转换序列或开始通道扫描序列
第 3 类	共用 S&H 通道扫描输入	通道扫描触发源	启动通道扫描序列

第 1 类输入与专用 S&H 电路相关。每个专用 S&H 有单个第 1 类输入与其相关。每个第 1 类输入具有唯一的触发选择寄存器。

第 1 类输入可以是通道扫描列表的一部分，通过公共扫描触发源触发。

第 2 类输入用在共用 S&H 电路上，可被单独或作为通道扫描列表的一部分触发。当单独使用时，通过其唯一的触发寄存器触发。

第 3 类输入专用于通道扫描。它们共用一个公共触发源。使用通道扫描时，扫描列表中可组合第 1 类、第 2 类和第 3 类输入。

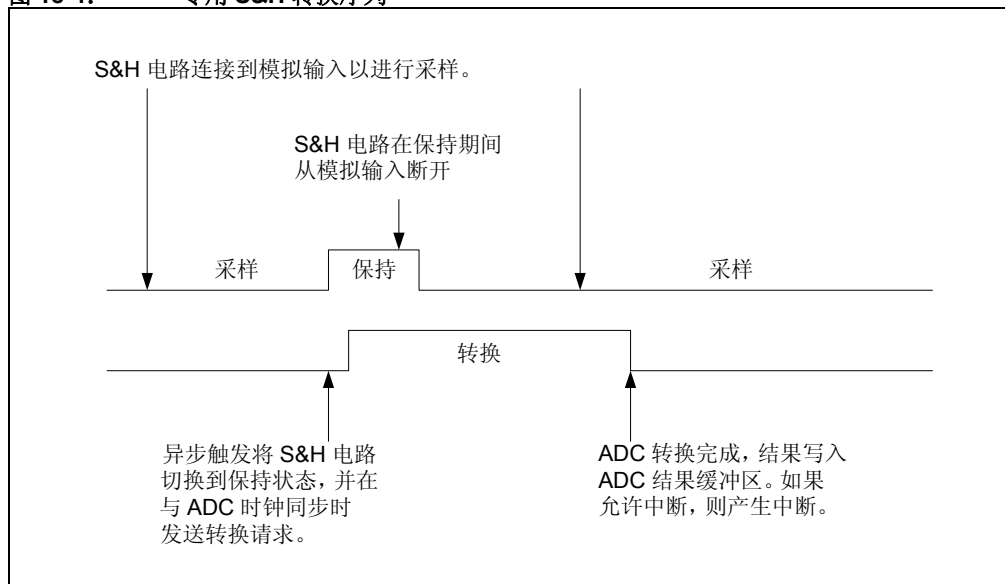
注： 每个第 1 类输入还有一个备用选择。该交替输入独立于第 1 类、第 2 类和第 3 类输入。当交替输入用于第 1 类输入时，仍然通过第 1 类触发寄存器选择触发信号，结果数据存储在第 1 类输出寄存器中，转换的优先级保持第 1 类输入的优先级。

18.3.2 专用采样和保持

顾名思义，专用 S&H 电路专用于第 1 类模拟输入，并使能高速、高精度采样和转换，对于捕捉时间敏感或瞬态信号尤其有用。可以配置为单端或差分模式。

每个专用 S&H 在采样模式下持续跟踪输入信号，直到发生异步触发事件。触发事件会导致 S&H 立即停止采样并进入保持状态。需要注意的是，结束采样的触发事件与 ADC 时钟异步发生。S&H 电路立即进入保持状态时，异步触发必须与 ADC 时钟同步，消耗 2 个 ADC 时钟边沿，然后将转换请求发送到流水线转换器。如果不存在高优先级转换请求，将立即开始转换；否则，转换根据优先级进行。S&H 在转换过程中的初始阶段始终处于保持状态。当转换完成时，数据传送到结果缓冲区并且产生中断。

图 18-4: 专用 S&H 转换序列



如果将周期性触发源用于专用 S&H，总采样时间由触发率决定。触发率不得违反必要的采样时间。更多信息，请参见 18.10 “ADC 采样要求”。

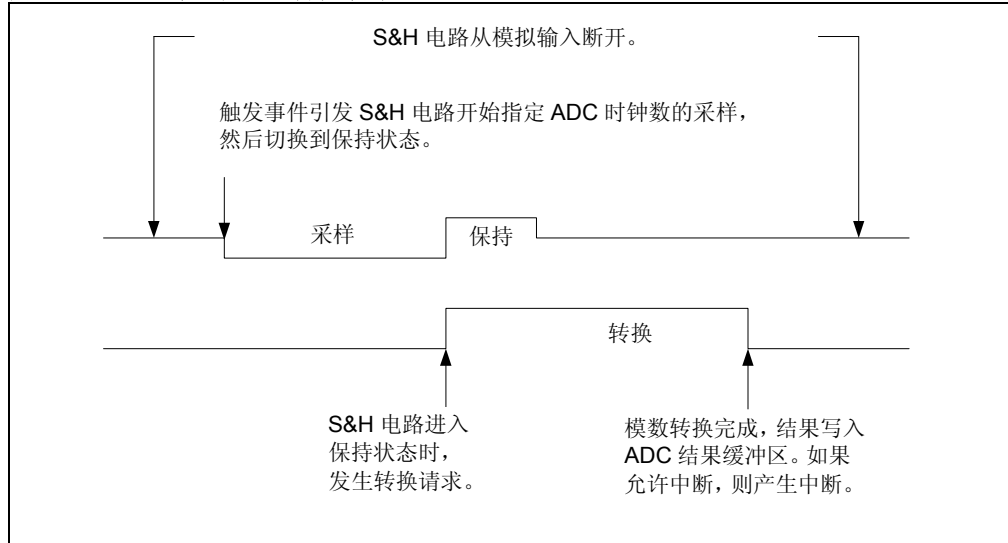
- 注 1:** 专用 S&H 中没有机制可确保满足最小采样时间。系统设计人员必须确保 S&H 在触发前有足够的采样时间。
- 2:** 专用 S&H 电路提供的交替输入提供了备用物理连接；但是，专用 S&H 始终使用其主第 1 类输入的触发配置和数据结果寄存器。

18.3.3 共用采样和保持

共用 S&H 电路为处理大量对时序要求并不严格的第 2 类和第 3 类输入提供了很大的灵活性。同样也可将共用 S&H 电路配置为单端或差分模式；但是，在差分模式下，单个负输入为所有正输入选择所共用。

共用 S&H 在正输入上使用多路开关以实现较多输入之间的共用。不同于专用 S&H，共用 S&H 的触发事件启动使用指定采样时间的采样过程。采样到指定 ADC 时钟数量的信号后，S&H 进入保持状态，并发送转换请求。S&H 在转换过程中的初始阶段始终处于保持状态。

图 18-5: 共用 S&H 转换序列



当使用共用 S&H 时，ADC1 控制寄存器 2 中的 SAMC<7:0> 位 (AD1CON2<23:16>) 用于确定采样时间。当周期性地触发共用 S&H 上的单个输入时，触发率应不少于采样时间加 2 TAD，这是传送到流水线转换器所需的时间。

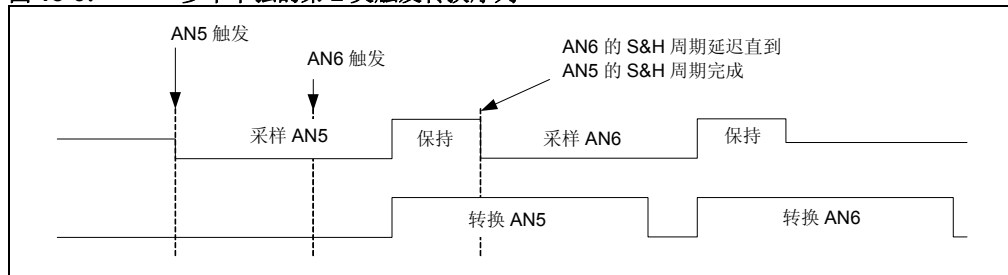
共用 S&H 有两种工作模式：独立的第 2 类触发和通道扫描。共用 S&H 可在任一模式下工作，也可同时在两个模式下工作。无法保证第 2 类或第 3 类转换请求将立即得到处理。流水线转换器按顺序或优先级处理转换请求。

18.3.3.1 第 2 类触发

当第 2 类输入定义为使用单独触发信号时，这些输入通过共用 S&H 使用图 18-5 中所示的序列来采样和转换。当使用多个采用单独触发信号的第 2 类输入时，了解触发时序的结果非常重要。

如果正在进行转换而发生另一个第 2 类触发，然后新触发的采样保持转换将暂停，直到进行中的采样 / 保持周期完成，如图 18-6 所示。

图 18-6: 多个单独的第 2 类触发转换序列



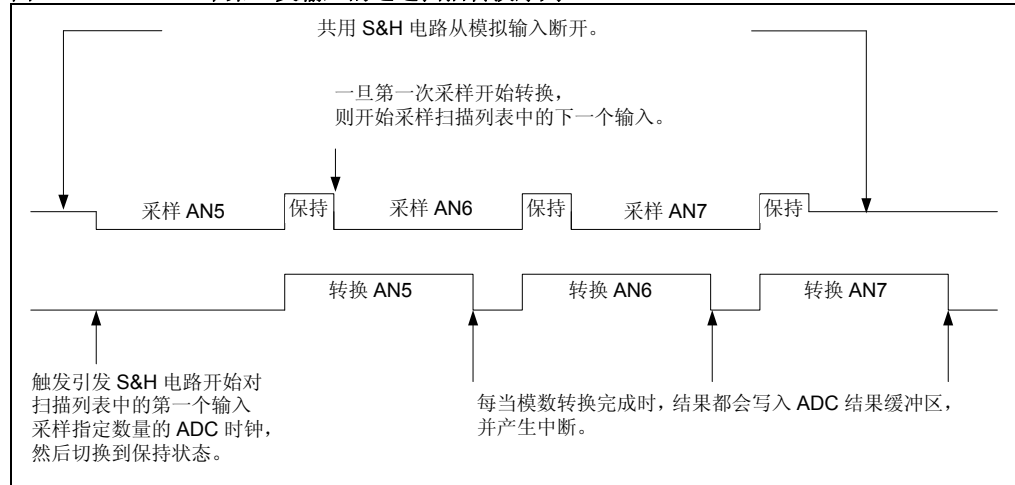
当共用 S&H 的多个输入同时触发时，处理顺序由其天然优先级决定（编号最低的输入具有最高的优先级）。例如，如果 AN5、AN6 和 AN7 使用相同的触发源，AN5 将首先采样和转换，之后是 AN6，最后是 AN7。

在共用 S&H 上使用独立第 2 类触发时，SAMC<7:0> 位 (AD1CON2<23:16>) 决定所有输入的采样时间，而 ADC1 输入转换控制寄存器中相应的 TRGSRC<4:0> 位（见寄存器 18-15）决定每个输入的触发源。

18.3.3.2 通道扫描

通道扫描是一种功能，支持多个第 1 类、第 2 类或第 3 类输入的自动扫描序列。所有第 2 类和第 3 类输入使用单个共用 S&H 扫描。第 1 类输入使用其专用 S&H 扫描。单个触发源启动一个通道扫描。不使用单独触发（如果可用）。当发生触发时，同时捕捉所有第 1 类输入，并基于自然优先级启动转换。触发也将启动对扫描列表中首个第 2 类或第 3 类输入的采样，而且采样完成时，在所有的第 1 类转换开始后将采样传递给转换器。使用自然输入顺序；编号较低的输入在编号较高的输入之前采样。

图 18-7: 3 个第 2 类输入的通道扫描转换序列

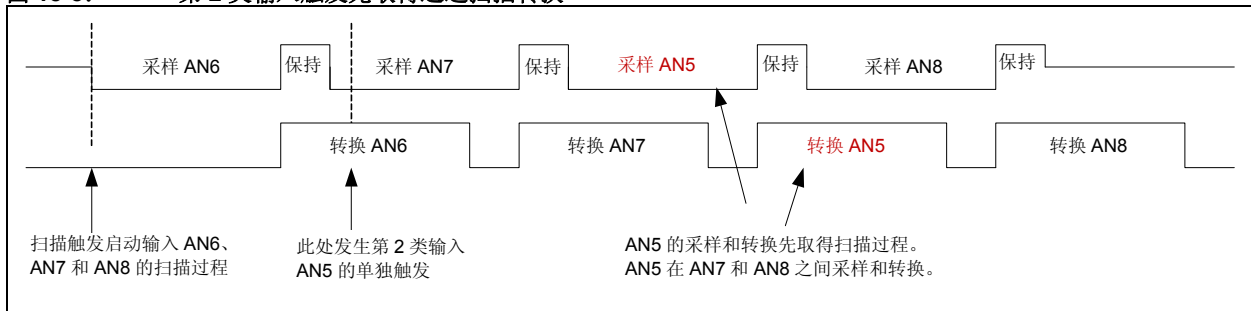


当共用 S&H 工作在扫描模式下时，ADC1 控制寄存器 2 中的 SAMC<7:0> 位 (AD1CON2<23:16>) 决定所有输入的采样时间，而 ADC1 控制寄存器 1 (AD1CON1<26:22>) 中的扫描触发源选择位 (STRGSRC<4:0>) 决定触发源。

为了确保可预测的结果，不应再触发扫描直到完成所有通道的采样。进行系统设计时应注意避免在扫描进行时再触发通道扫描。

如果扫描期间发生的单独第 2 类触发的优先级比当前正在处理的采样高，那么将先取得通道扫描序列。在图 18-8 中，发生第 2 类输入 AN5 的独立触发时，AN6、AN7 和 AN8 的通道扫描正在进行。通道扫描被中断，用于进行 AN5 的采样和转换。

图 18-8: 第 2 类输入触发先取得通道扫描转换



18.3.4 6 级流水线转换器

12 位 ADC 具有一个 6 级流水线转换器架构。这支持同时对最多 6 个采样进行转换，从而在需要转换多个采样时可产生较低延时。采样的转换时间由 ADC 时钟速率决定。

当多个输入准备同时转换时，与同时采样的情况一样，转换器基于其自然优先级接收信号。阻止较低优先级输入，使其等待较高优先级输入。

如图 18-9 所示，S&H0 使用 AN0，S&H1 使用 AN1，S&H2 使用 AN2。所有 3 个输入配置为使用相同的触发信号进行同时采样。由于使用自然优先级，首先将 AN0 传输至 6 级转换器，接下来是 AN1，最后是 AN2。保持 S&H 在保持状态，直到已将其采样传递至转换器，然后自动恢复采样。

图 18-9: 流水线转换器（转换过程）

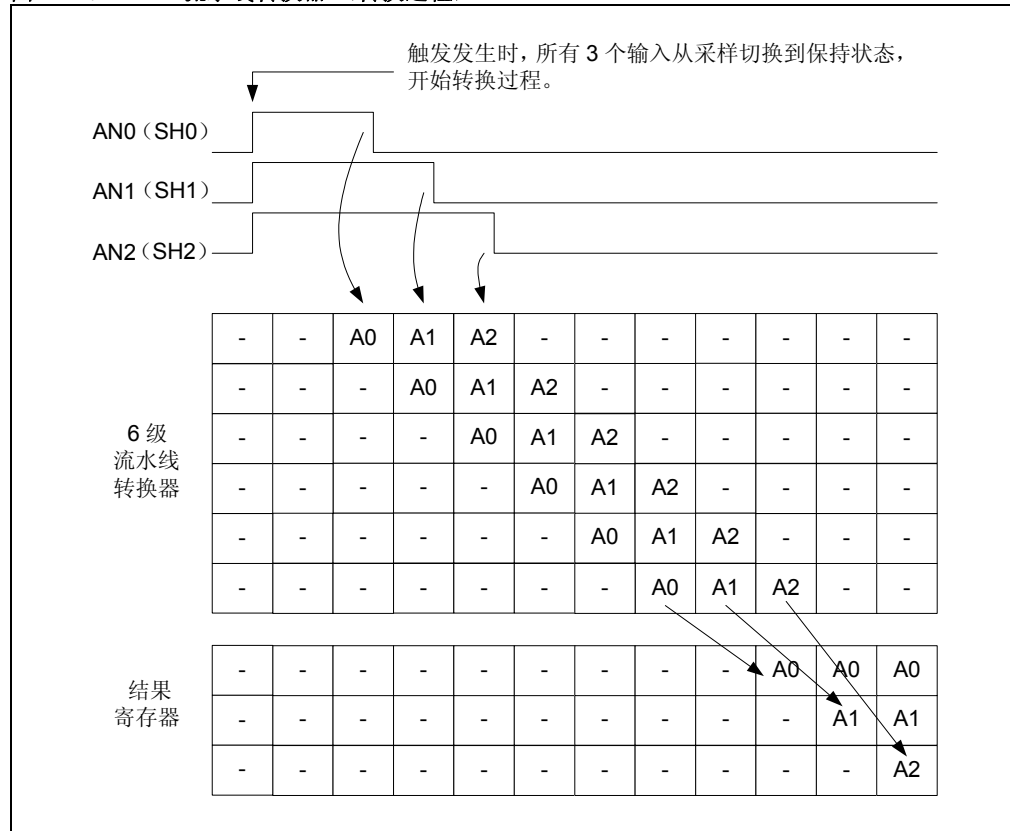
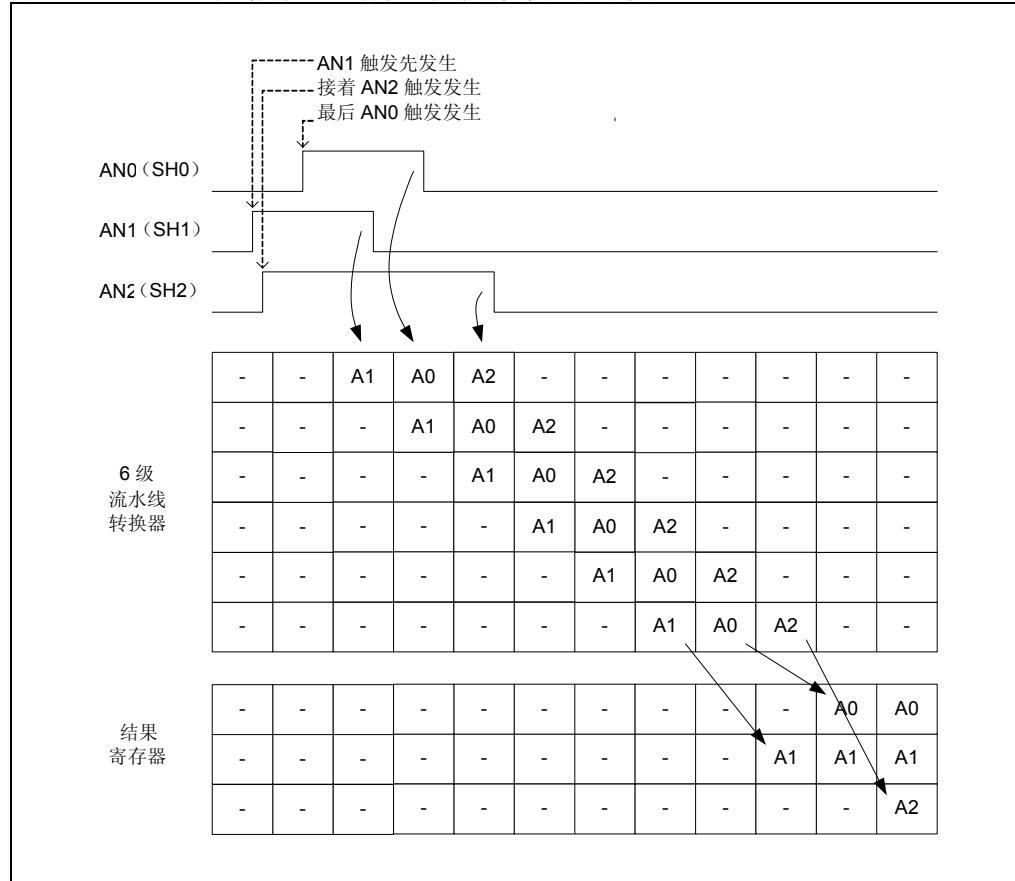


图 18-10 显示了输入转换的自然优先级。同样，S&H0 使用 AN0，S&H1 使用 AN1，S&H2 使用 AN2；但是，每个输入单独触发。首先触发 AN1，接下来是 AN2，最后是 AN0。AN2 和 AN0 的触发发生在将 AN1 传输至转换器流水线时发生。当完成 AN1 传输时，AN0 被移入转换器流水线，即使 AN2 的触发发生在 AN0 之前。这是因为 AN0 的自然优先级比 AN2 高。

图 18-10: 流水线转换器 (转换过程中自然优先级的影响)



18.3.5 采样和转换序列

S&H和流水线转换器组合建立了采样和转换序列以及产生的采样和转换时间。这些序列随着S&H类型（专用或共用）和工作模式的不同而变化。在后续章节中对基本工作原理进行了说明。

18.3.5.1 流水线转换时间

转换时间是从 S&H 进入保持状态直到结果寄存器中存在数据的时间。在这期间，采样从 S&H 电路传输至流水线转换器。如果没有其他采样等待转换，则该时间是确定的。由于只有单个流水线转换器而有多 S&H 电路，如果 S&H 电路等待传输至转换器，采样必须等待首个转换器级可用。如前所述，自然优先级决定等待采样传输至初始流水线转换器级的顺序。

18.3.5.2 专用 S&H 转换时间（第 1 类输入）

如前所述，触发第 1 类输入会导致专用 S&H 进入保持状态。紧接着发生异步触发的 ADC 时钟，转换启动两个时钟周期。公式 18-1 定义了从触发到数据就绪中断发生所需的时间。

公式 18-1: 专用 S&H 转换时间

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

其中:

T_{AD} = 模数转换时钟周期

注: 该公式假设没有转换等待处理。每个等待较高优先级的转换时间将加 1 个 T_{AD} 。

18.3.5.3 专用 S&H 采样时间 (第 1 类输入)

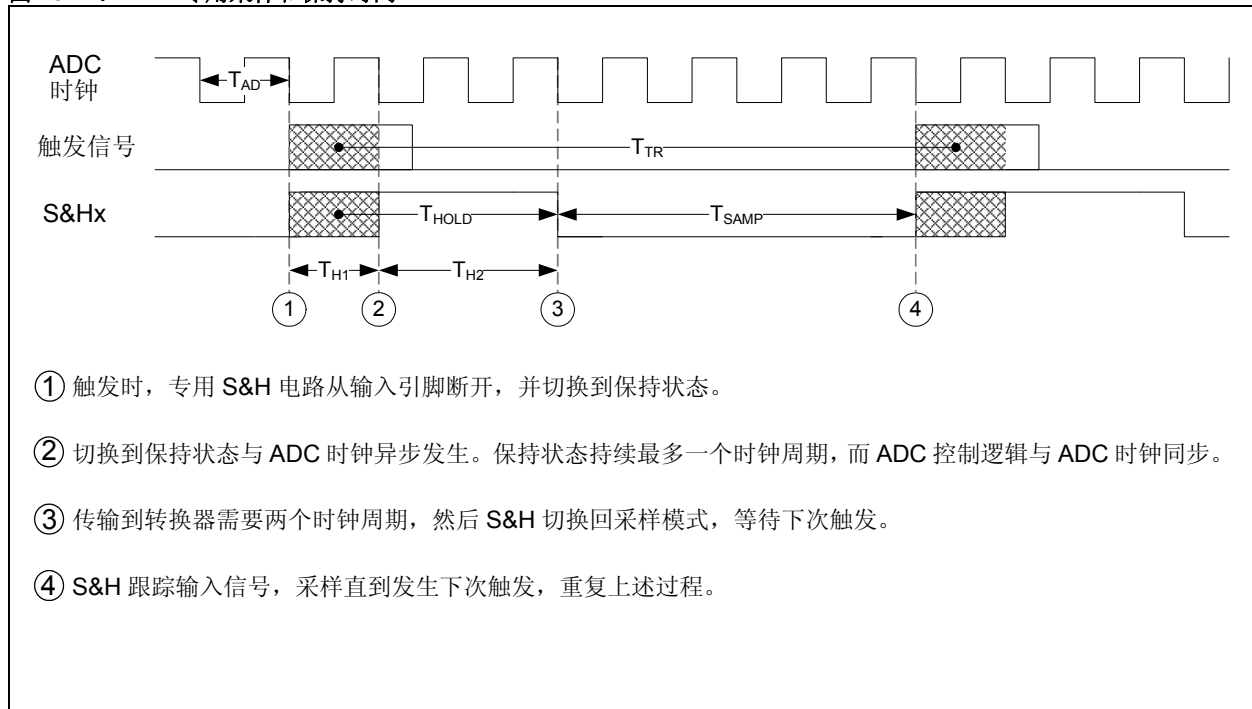
专用 S&H 电路跟踪输入直至发生触发, 这将启动一个转换序列。触发率建立采样时间。这与共用 S&H 电路不同, 触发将输入连接至 S&H, 然后紧接着转换序列启动一个采样周期。

图 18-11 给出了通过周期性源触发的单个第 1 类输入的简单示例。

采样时间为触发率 T_{TR} 和保持时间 T_{HOLD} 之间的差值。一旦 S&H 离开保持状态, 可立即再次触发输入, 如图 18-11 中的数字 3 指示, 使其以不违反最小采样时间要求的速率触发。控制逻辑未构建任何机制来确保满足专用 S&H 第 1 类输入的最小采样时间要求。

注: 触发第 1 类输入时, 触发率必须考虑满足最小采样时间要求。

图 18-11: 专用采样和保持时间



公式 18-2 可用于确定第 1 类输入的采样时间。

公式 18-2: 专用 S&H 采样时间

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

其中:

T_{TR} = 触发率时间间隔 (由系统设计决定)

$T_{HOLD} = TH1 + TH2$

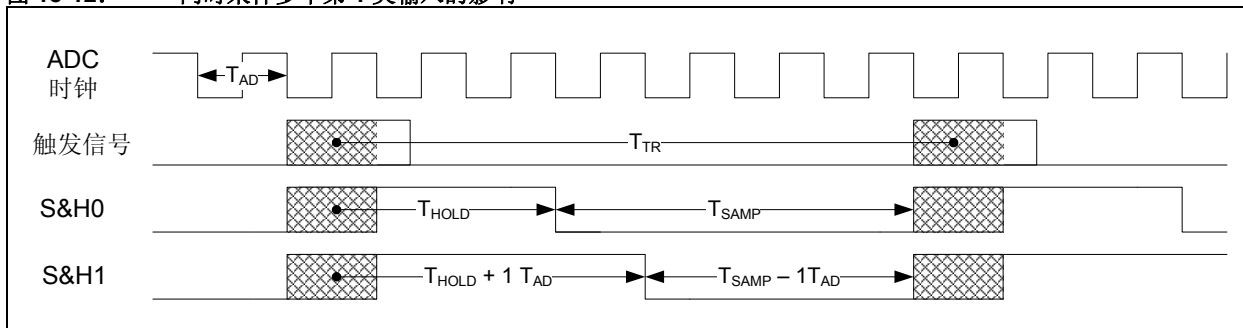
$TH1 = 1 T_{AD}$ (时钟同步)

$TH2 = 2 T_{AD}$ (每个等待较高优先级转换增加 1 个 T_{AD})

注: 系统设计必须确保第 1 类输入的触发率不会违反最小采样时间要求。专用 S&H 第 1 类输入没有采用任何机制来确立采样时间。采样时间完全由触发率严格决定。分配给时钟同步的时间 $TH1$, 可以少于 1 个 T_{AD} , 具体取决于异步触发与 ADC 时钟的关联。强烈建议在触发周期 T_{TR} 中将 1 个 T_{AD} 分配给 $TH1$, 确保所需的采样时间均得到满足, 无论触发与 ADC 时钟的相位关系如何。

如公式 18-2 所述, 对于每个等待较高优先级转换, $TH2$ 增加 1 个 T_{AD} 。同时触发多个第 1 类输入将始终创建一个等待较高优先级转换。图 18-12 显示了增加一个等待转换的影响, 其中 S&H 1 的采样时间比 S&H 0 少一个 T_{AD} 周期。在这些情况下, 很有必要增加触发间隔 T_{TR} , 以便最低优先级输入有足够的采样时间。

图 18-12: 同时采样多个第 1 类输入的影响



18.3.5.4 共用 S&H 转换时间（第 2 类输入）

如前文所述，在共用 S&H 电路上触发第 2 类输入将启动采样过程。采样过程完成时，S&H 将进入保持状态，并开始转换。SAMC 值决定采样时间。公式 18-3 定义了从触发到数据就绪并发生中断所需的时间。

公式 18-3: 共用采样和保持转换时间

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

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

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

其中:

T_{AD} = 模数转换时钟周期

$SAMC = SAMC<7:0>$ 位 (AD1CON2<23:16>)

注: 该公式假设没有等待处理的转换。每个等待较高优先级第 2 类转换将在转换时间与采样时间 $(SAMC + 1) \cdot T_{AD}$ 的和上加 1 个 T_{AD} 。

如果在共用 S&H 采样周期完成时有较高优先级第 1 类转换等待处理，每个较高优先级专用转换将延迟 1 个 T_{AD} 转换。

18.3.5.5 通道扫描转换时间（第 2 类和第 3 类输入）

如前文所述，扫描触发信号启动配置为通道扫描的第 1 类、第 2 类或第 3 类输入的顺序采样和转换过程。完成通道扫描的总耗用时间基于 ADC 时钟率、扫描的输入数和采样时间。公式 18-4 针对由第 2 类和第 3 类输入组成的扫描列表定义了从触发到最后一个扫描输入的数据就绪且发生中断所需的时间。

公式 18-4: 通道扫描采样和转换时间

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

其中:

N = 扫描通道数

T_{AD} = 模数转换时钟周期

$SAMC = SAMC<7:0>$ 位 (AD1CON2<23:16>)

注: 该公式假设没有等待处理的转换。每个等待较高优先级第 1 类转换的转换时间将加 1 个 T_{AD} 。每个等待较高优先级第 2 类转换将在转换时间与 $(SAMC + 1) \cdot T_{AD}$ 采样时间的和上加上 1 个 T_{AD} 。

18.4 ADC 模块配置

ADC 模块的操作通过特定寄存器中的位设置指定。以下指令汇总了操作和设置。每个配置步骤的选项和详细信息将在后续章节中提供。

要配置 ADC 模块，请执行以下步骤：

1. 配置模拟端口引脚，如 18.4.1 “配置模拟端口引脚” 中所述。
2. 选择 ADC MUX 的模拟输入，如 18.4.2 “选择 ADC 多路开关模拟输入” 中所述。
3. 选择 ADC 结果格式，如 18.4.5 “选择 ADC 结果格式” 中所述。
4. 选择转换触发源，如 18.4.6 “选择转换触发源” 中所述。
5. 选择参考电压源，如 18.4.7 “选择参考电压源” 中所述。
6. 选择扫描输入，如 18.4.8 “选择扫描输入” 中所述。
7. 选择模数转换时钟源和预分频比，如 18.4.9 “选择模数转换时钟源和预分频比” 中所述。
8. 指定任意额外采集时间（如果需要），如 18.10 “ADC 采样要求” 中所述。
9. 启动 ADC 模块，如 18.4.10 “开启 ADC” 中所述。
10. 执行 ADC 校准，如 18.4.12 “ADC 校准” 中所述。
11. 配置 ADC 中断（如果需要），如 18.6 “中断” 中所述。

注： 步骤 1 至 8 可以按任意顺序执行，但在任何情况下，步骤 9 至 11 都必须是最后的几个步骤，并且必须以上述指定顺序执行。

18.4.1 配置模拟端口引脚

与模拟输入相关的 I/O 端口的 ANSELx 寄存器，用于将相应的引脚配置为数字引脚。当相应的 ANSELx 位 = 1 时，引脚被配置为模拟输入。当 ANSELx 位 = 0 时，引脚被设置为数字控制引脚。当引脚被配置为模拟输入时，相关的端口 I/O 数字输入缓冲区禁止，因此不消耗电流。器件退出复位状态时，ANSELx 寄存器置 1，这使得 ADC 输入引脚默认配置为模拟输入。

TRISx 寄存器用于控制端口引脚的数字功能。对于需要作为模拟输入的端口引脚，具体 TRISx 寄存器中其对应的位必须置 1。如果通过将 TRISx 位清零将与 ADC 输入相关的 I/O 端口配置为输出，端口的数字输出电平 (VOH 或 VOL) 将被转换。在器件复位后，所有 TRISx 位均置 1。关于端口引脚配置的更多信息，请参见具体器件数据手册的 “I/O 端口” 章节。

注： 如果某个 PORT 寄存器与 ADC 共用引脚，则读取 PORT 锁存器时，配置为模拟输入的任何引脚都读为 0。任何定义为数字输入、但未配置为模拟输入的引脚上的模拟电平都可能导致输入缓冲器消耗的电流超出器件规范。

例 18-1: ADC 初始化

```
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);
}
```

例 18-1: ADC 初始化 (续)

```
/* 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 选择 ADC 多路开关模拟输入

每个 S&H 都具有两个输入，称为同相输入和反相输入。下面的章节对专用和共用 S&H 的输入选择选项的不同进行了说明。

每个专用 S&H 专用于单个第 1 类正输入。这意味着该 S&H 的所有控制、状态和数据均通过与其专用模拟 (ANx) 输入相关的寄存器进行访问。例如，如果用于 ADC1 的 S&H1 使用 AN0 作为其专用输入，从 S&H1 获取的读数将始终从 AD1DATA0 读取，通过 AGIEN0 允许全局中断，使用 ARDY0 读取中断状态，通过 CMPE0 使能数字比较器，且触发源通过 TRGSC0 设置。这对所有专用第 1 类 S&H 电路均适应，而无论以下设置为何。

为使专用 S&H 电路更加灵活，为每个专用 S&H 电路提供了单个交替输入。可以使用 ADC1 输入模式控制寄存器 AD1IMOD 中的 SHxALT<1:0> 位选择交替输入，如下所述：

- 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>)

表 18-3 给出了具有 5 个 S&H 电路的器件示例：

表 18-3: 正输入选择

模块	默认选择 (SHxALT<0> = 0)	备用选择 (SHxALT<0> = 1)
ADC1	SH0 = AN0 SH1 = AN1 SH2 = AN2 SH3 = AN3 SH4 = AN4	SH0 = AN45 SH1 = AN46 SH2 = AN47 SH3 = AN48 SH4 = AN49

注： 表 18-3 所示的选择仅用于说明。关于器件上实际的通道分配，请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节。

18.4.3 共用采样和保持的正输入选择

对于共用 S&H 电路，所有第 2 类和第 3 类输入之间共用正输入。触发第 2 类输入时或第 2 类或第 3 类输入的通道扫描期间，模拟输入 ANx 自动连接至 S&H 电路。选择通道扫描的输入将单独在 18.4.8 “选择扫描输入”中讨论。

18.4.4 负输入单端和差分选项 (共用和专用采样和保持)

负输入选择由 AD1IMOD 寄存器中 SHxMOD<1:0> 位的设置决定。SHxMOD<1:0> 位允许轨到轨的输入，且可为单端或差分。这些位也可用于测量内部 ADC 模拟输入和参考电压，并且配置数字结果以符合期望的满量程输出范围。

表 18-4: 输入配置

SHxMOD<1:0> 设置	输入配置	输入电压 (见注 1)		输出
11	差分二进制补码	最小输入	$V_{INP} - V_{INN} = -V_{REF}$	-2048
		最大输入	$V_{INP} - V_{INN} = V_{REF}$	+2047
10	差分单极性	最小输入	$V_{INP} - V_{INN} = -V_{REF}$	0
		最大输入	$V_{INP} - V_{INN} = V_{REF}$	+4095
01	单端二进制补码	最小输入	$V_{INP} = V_{INN}$	-2048
		最大输入	$V_{INP} - V_{INN} = V_{REF}$	+2047
00	单端单极性	最小输入	$V_{INP} = V_{INN}$	0
		最大输入	$V_{INP} - V_{INN} = V_{REF}$	+4095

图注: V_{INP} = S&H 正输入; V_{INN} = S&H 负输入;
 $V_{REF} = V_{REFH} - V_{REFL}$

注 1: 为确保正常工作和防止器件损坏，输入电平不应超过具体器件数据手册的“电气特性”章节列出的限制值。

SHxMOD<1:0> 位在 AD1IMOD 寄存器中定义如下:

- 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>)

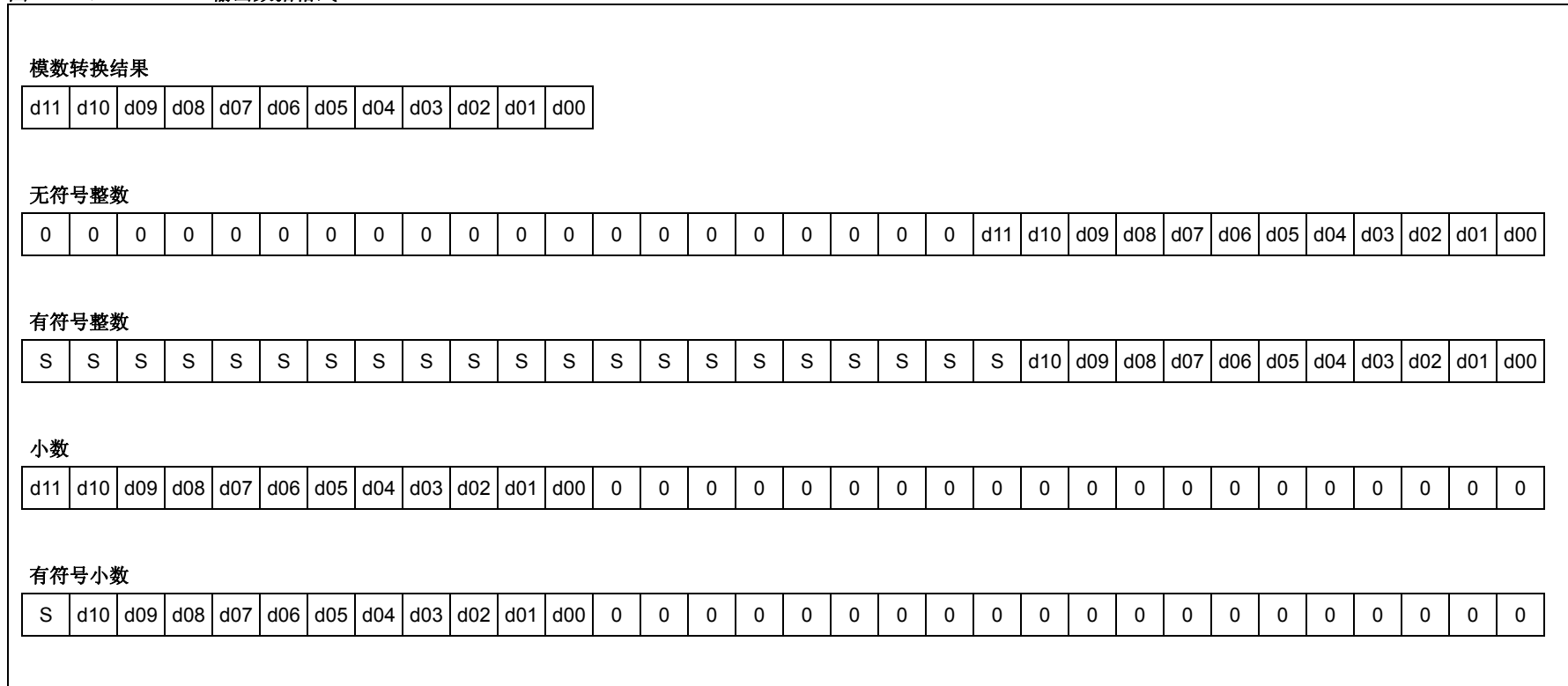
18.4.5 选择 ADC 结果格式

ADC 结果寄存器中的数据可以使用四种支持的数据格式读取。用户可以选择无符号整数、有符号整数、无符号小数或有符号小数。整数数据右对齐，而小数数据左对齐。

- 使用小数数据输出格式位 **FRACT** (**AD1CON1<11>**) 为所有 ADC 输入全局指定整数 / 小数数据格式选择
- 可使用 **SHxMOD<1:0>** 位单独为每个独立 S&H 电路指定有符号 / 无符号数据格式选择，如 18.4.4 “负输入单端和差分选项（共用和专用采样和保持）” 所示

图 18-13 给出了各种格式下的结果。

图 18-13: ADC 输出数据格式



例 18-2: ADC 第 2 类配置和小数格式

```

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.
}

```

例 18-2: ADC 第 2 类配置和小数格式 (续)

```
/* 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“选择 ADC 结果格式”
     */
}

return (1);
}
```

18.4.6 选择转换触发源

ADC 模块的第 1 类和第 2 类输入可触发为单独地或作为通道扫描序列的一部分进行转换。第 3 类输入只可作为通道扫描序列的一部分触发。单独或通道扫描触发信号可产生于片内定时器或输出比较外设事件，产生于连接到 INTO 的外部数字电路，产生于连接到模拟比较器的外部模拟电路，或通过软件将 SFR 中的触发位置 1 产生。

- 注 1:** 当多个第 1 类或第 2 类模拟输入的转换触发同时发生时，根据基于使用的模拟输入的自然顺序优先级机制确定优先级。AN0 具有最高优先级，AN1 具有下一个最高优先级，以此类推。因此，应使用具有较高自然优先级的第 1 类 S&H 电路采样和转换时间敏感或瞬态模拟信号。
- 2:** 为第 1 类输入选择触发源时，用户应确保没有违反采样时间要求。

18.4.6.1 第 1 类和第 2 类输入的触发信号选择

用户应用程序可单独为每个第 1 类和第 2 类输入指定转换触发源。模拟引脚 n 的独立触发源通过位于寄存器 AD1TRG1 至 AD1TRG3 的 TRGSRCn<4:0> 位指定。关于可用转换触发信号选项的更多信息，请参见具体器件数据手册中的“12 位模数转换器 (ADC)”章节。例如，这些触发源可包括：

- **通用 (General Purpose, GP) 定时器:** 当 32 位定时器、Timer3/2 或 Timer5/4，或 16 位 Timer1、Timer3 或 Timer5 发生周期匹配，定时器将产生一个特殊 ADC 触发事件信号。其他定时器不具备该特性。更多信息，请参见第 14 章“定时器” (DS61105) 和具体器件数据手册。
- **输出比较:** 输出比较外设 OC1、OC3 和 OC5 可用于产生 ADC 触发信号，然后输出从低电平状态跳变到高电平状态。更多信息，请参见第 16 章“输出比较” (DS61111) 和具体器件数据手册。
- **比较器:** 当输出从低电平状态跳变到高电平状态时，模拟比较器可用于产生 ADC 触发信号。更多信息，请参见第 19 章“比较器” (DS61110) 和具体器件数据手册的“比较器”章节。
- **外部 INTO 引脚触发:** 在该模式下，ADC 模块在 INTO 引脚上发生有效跳变时启动转换。INT0 引脚可以设定为使用上升沿输入或下降沿输入触发转换过程。
- **全局软件触发:** ADC 模块可以配置为手动触发所有选择该触发选项的输入的转换。用户可以通过设置全局软件触发位 GSWTRG (AD1CON3<30>) 手动触发转换。

18.4.6.2 第 1 类、第 2 类和第 3 类输入的扫描触发选择 (仅 SH3)

所有可用的模拟输入均可配置为自动通道扫描。第 1 类输入通过其专用 S&H 电路采样。第 2 类和第 3 类输入通过共用 S&H 电路采样。使用 STRGSRC<4:0> 位 (AD1CON1<26:22>) 为所有选择为扫描的输入选择单个转换触发源。每次转换触发时，ADC 模块按照自然优先级对用户指定通道扫描列表指定的所有输入进行转换。对于第 1 类输入，采样在触发时结束。对于第 2 类和第 3 类输入，触发信号以自然优先级顺序启动顺序采样 / 转换过程。

可用于通道扫描的触发选项与可用于独立触发第 1 类和第 2 类输入的选项相同。作为通道扫描一部分的任一第 1 类或第 2 类输入必须在 TRGSRCn<4:0> 位中选择 STRIG 选项作为其触发源。

18.4.6.3 用户请求的独立转换触发

用户可在程序执行期间的任何时候明确请求对任何所选模拟输入的单个转换，而无需更改ADC的触发源配置。要转换的输入通过ADC输入选择位ADINSEL<5:0> (AD1CON3<5:0>)指定。独立ADC输入转换请求位RQCONVRT (AD1CON3<29>)用于触发该转换。该位在两个外设时钟周期后自动清零，从而允许用户软件在需要时触发另一个转换。

18.4.7 选择参考电压源

用户应用程序可以为ADC模块选择参考电压，参考电压可以是内部或外部电压。VREF输入选择位VREFSEL<2:0> (AD1CON3<12:10>)用于选择模数转换的参考电压。参考电压高电压 (VREFH)和参考电压低电压 (VREFL)可作为内部AVDD和AVSS电压轨，也可作为VREF+和VREF-输入引脚电压。外部ADC参考电压可用于降低转换器中的噪声。

加到外部参考电压引脚上的电压必须符合特定的规范。关于电气规范，请参见具体器件数据手册中的“电气特性”章节。

参考电压升压位BOOST (AD1CON2<14>)，应该在选择的参考电压 (VREFH - VREFL)之间的差小于 $0.65 * (AVDD - AVSS)$ 时置1。该位置1时，将最大限度提高使用小参考电压轨的模数转换的SNR。

注： 外部VREF+和VREF-引脚可以与其他模拟外设共用。更多信息请参见具体器件数据手册的“引脚图”章节。此外，VREF+和VREF-引脚的ANSELx位必须设置为模拟模式。

18.4.8 选择扫描输入

要将某个模拟输入包含在扫描列表中，输入扫描选择寄存器 (AD1CSS1或AD1CSS2)中其指定的位必须设置为1。AD1CSS1和AD1CSS2寄存器中的每个位对应于不同的模拟输入。如果输入为第1类或第2类输入，该输入的触发源通过位于AD1TRG1至AD1TRG3寄存器的TRGSRcn<4:0>位指定，且必须设置为STRIG选项，表示输入作为扫描过程的一部分触发。

每个扫描转换触发信号 (如STRGSRc<4:0>位 (AD1CON1<26:22>)所指定)用于启动扫描转换序列。如18.3“ADC工作原理、术语和转换序列”中所述，扫描序列以输入的自然优先级顺序采样和转换每个扫描输入。

注： AD1CSS1和AD1CSS2位仅指定S&H电路的正输入。通道的负输入在扫描序列期间不改变。

例 18-3: ADC 通道扫描

```

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.STRGSR = 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;

```

例 18-3: ADC 通道扫描 (续)

```
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 选择模数转换时钟源和预分频比

ADC 模块可以使用内部快速 RC (Fast RC, FRC) 振荡器输出或系统时钟 (SYSCLK) 作为转换时钟源 (T_Q)。在一些器件上, 参考时钟可用作转换时钟源。请参见具体器件数据手册的“12 位模数转换器 (ADC)”章节以确定哪些参考时钟输出源可用。

当 ADCSEL<1:0> 位 (AD1CON2<9:8>) = 11 时, 内部 FRC 振荡器用作 ADC 时钟源。当使用内部 FRC 振荡器时, ADC 模块可在休眠和空闲模式下继续工作。

注: 建议要求精确 ADC 采集时序的应用使用 SYSCLK 作为 ADC 的时钟源。

当模数转换时钟源 (T_Q) 位 ADCSEL<1:0> (AD1CON2<9:8>) = 01 时, SYSCLK 用作转换时钟源。当 ADCSEL<1:0> 位 = 00 时, 禁止 ADC 时钟。

模数转换时钟周期 (T_{AD}) 是应用 ADC 输入时钟预分频比位 ADCDIV<6:0> (AD1CON2<6:0>) 之后所选时钟源周期。计算得到 T_{AD} 值的范围是从 1 * T_Q 到 254 * T_Q。

为了正确进行模数转换, 不得超出 T_{AD} 限制值。ADC 模块支持的 ADC 时钟频率范围为 1 MHz 至 28 MHz。

ADC 模块完成模数转换可达的最高速率（有效转换吞吐量）为 28 Msps。但是，单个输入通道的最高转换速率取决于采样时间要求。可能的最快转换速率可以通过使用多个第 1 类输入采样和转换模拟信号来实现。

注： 最小转换延时（从触发到结果可用的时间）是 $10 * T_{AD}$ 。实际延时可能是 $11 * T_{AD}$ ，因为触发信号与 ADC 时钟异步。

如公式 18-5 所示， T_{AD} 值受 $ADCDIV<6:0>$ 控制位（ $AD1CON2<6:0>$ ）和器件指令时钟周期 T_{CY} 的影响。

公式 18-5: A/D 转换时钟周期

对于 $ADCDIV = 0$

$$T_{AD} = T_Q$$

对于 $ADCDIV > 0$

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

无论哪种情况：

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

对于共用 S&H 电路， $SAMC<7:0>$ 位（ $AD1CON2<23:16>$ ）指定 T_{AD} 时钟周期数，在转换触发和启动转换间应采样模拟输入的周期数。

18.4.10 开启 ADC

当 ADC 工作模式位 $ADCEN$ （ $AD1CON1<15>$ ）设置为 1 时，模块处于工作模式，处于完全供电状态，可使用所有的功能。

当 $ADCEN$ 为 0 时，模块被禁止。电路的数字和模拟部分被关闭，以最大限度地节省电流消耗。

为了从关闭状态返回工作模式，用户必须等待模拟级稳定下来。关于稳定时间的更多信息，请参见具体器件数据手册的“电气特性”章节。

注： 在 ADC 模块使能时，建议不要对用于控制 ADC 时钟、通道分配、通道扫描、参考电压选择、S&H 电路工作模式和中断配置的 ADC 控制位进行写操作。

18.4.11 ADC 状态位

ADC 模块包含 $ADRDY$ （ $AD1CON2<31>$ ）状态位，指示其当前状态。该位指示 ADC 模块已就绪，可开始转换。 $ADCRDY$ 位仅在 ADC 模块已使能且校准后置 1。用户应用程序在 $ADCRDY$ 位置 1 前不应执行任何 ADC 操作。

18.4.12 ADC 校准

使能 ADC 模块之前，用户必须从配置存储器读取 ADC 校准数据（更多信息，请参见具体器件数据手册中的“**特殊功能**”章节），然后将该数据写入到校准寄存器 AD1CAL1 至 AD1CAL5。

当执行自校准时，SH0 的输入模式必须设置为差分输入模式中的一种。可以通过设置 SH0MOD<1:0>（AD1MOD<1:0>）位 = 10 来使用差分单极性编码格式，或通过设置 SH0MOD<1:0>（AD1MOD<1:0>）位 = 11 来使用差分二进制补码编码格式。自校准在每次 ADC 使能或 ADC 使能后启动校准周期时发生。通过设置 ADCEN（AD1CON1<15>）位 = 1 来使能 ADC。在 ADC 使能后通过设置 CAL（AD1CON3<31>）位 = 1 来启动校准周期。校准后，SH0 的输入模式可能会根据应用需要而变化。

以下情况需要执行重新校准：

- 使能 ADC 模块后参考电压发生变化
- 使能 ADC 模块后更改 BOOST 位设置

18.5 更多 ADC 功能

本节描述了 ADC 模块的一些其他特性，如数字比较器和过采样滤波器。

18.5.1 数字比较器

ADC 模块具有数字比较器，可用于监视选定模拟输入转换结果并当转换结果不在用户指定的限制之内时产生中断。仍需要转换触发信号来启动转换。当转换完成时比较自动发生。通过置 1 数字比较器模块使能位 ENDCMP (AD1CMPCON<7>) 使能该可选功能。

用户应用程序使用中断，该中断在模数转换结果高于或低于 AD1CMP 寄存器中指定的上限和下限值时产生。上限值和下限值分别在 ADCMPHI<15:0> 位 (AD1CMP<31:16>) 和 ADCMPLO<15:0> 位 (AD1CMP<15:0>) 中指定。

AD1CMPENx 寄存器中的 CMPEx 位 (x = 0 至 31) 用于指定哪个模拟输入被数字比较器监视 (针对前 32 个模拟输入 ANx, 其中 x = 0 至 31)。AD1CMPCON 寄存器指定将产生中断的比较条件，如下所示：

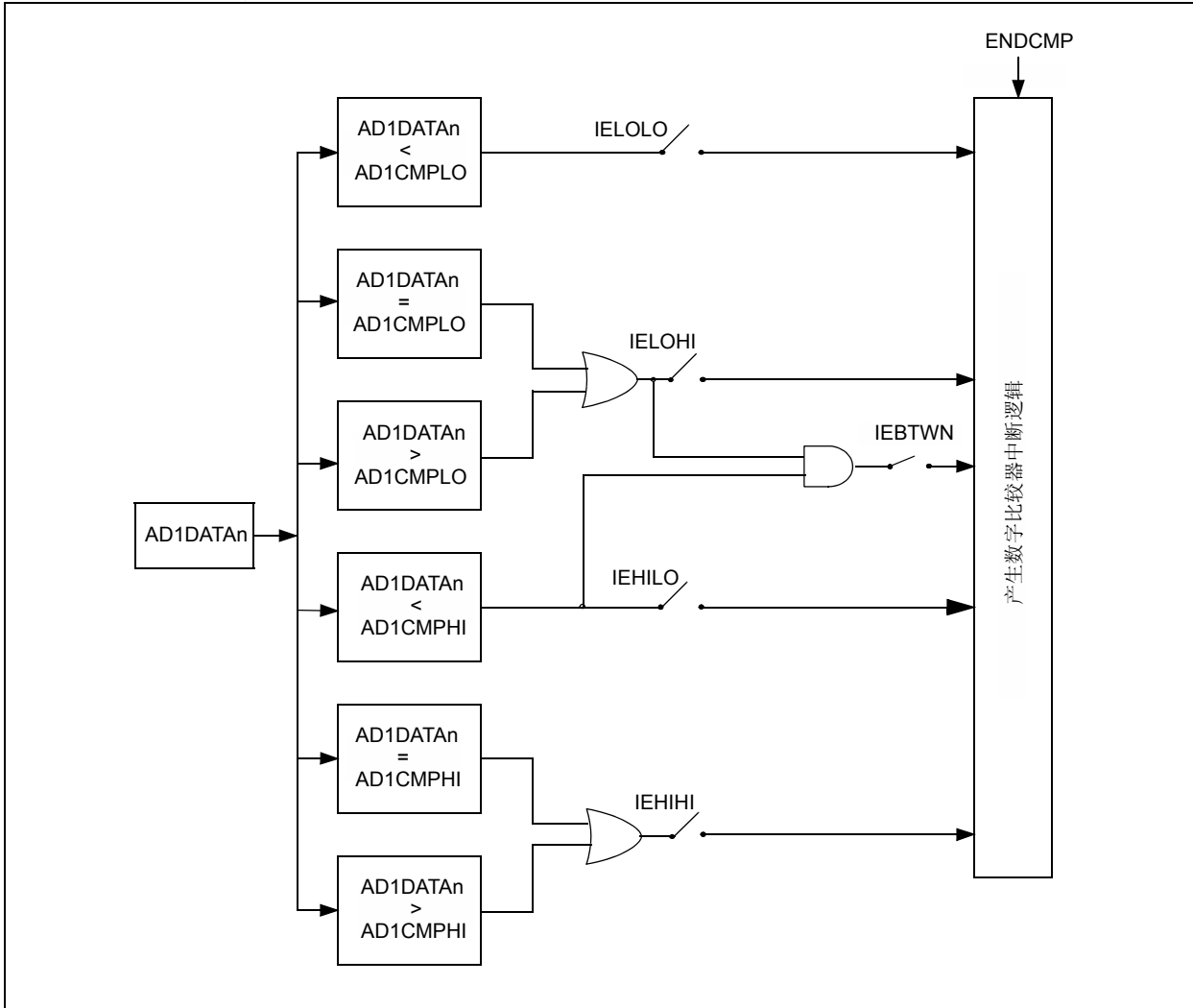
- IEBTWN = 1 时，当 $AD1CMPLO \leq AD1DATAn < AD1CMPHI$ 时产生中断
- IEHIHI = 1 时，当 $AD1CMPHI \leq AD1DATAn$ 时产生中断
- IEHILO = 1 时，当 $AD1DATAn < AD1CMPHI$ 时产生中断
- IELOHI = 1 时，当 $AD1CMPLO \leq AD1DATAn$ 时产生中断
- IELOLO = 1 时，当 $AD1DATAn < AD1CMPLO$ 时产生中断

图 18-14 说明了比较器事件的生成。ADC 模块生成转换结果时，该转换结果被提供给比较器且根据具体结果寄存器上存储的选定数据格式进行格式化。比较器使用 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>) 或 SH5MOD<1:0> (AD1IMOD<11:10>) 位 (具体取决于所使用的 S&H 电路) 来确定使用的数据格式，并相应地选择比较是否应该有符号或无符号。全局 ADC 设置由 FRACT 位 (AD1CON1<11>) 指定，还用于设置小数或整数格式。数值比较器将 ADC 结果和 AD1CMP 寄存器中的上限值和下限值 (具体取决于所选的比较条件) 进行比较。

根据比较器结果，可产生数字比较器中断事件。如果发生比较器事件，检测到数字比较器中断事件状态位 DCMPIF (AD1CMPCON<15>) 将置 1，且模拟输入识别 (ID) 位 AINID<4:0> (D1CMPCON<12:8>) 自动更新，从而用户应用程序知道哪个模拟输入产生中断事件。

- 注**
- 1: 数字比较器模块仅支持前 32 个模拟输入 (AN0 至 AN31)。
 - 2: 用户软件必须将 ADxCMPn 寄存器中包含的值格式化为有符号或无符号小数或整数形式以与已转换数据的格式匹配。

图 18-14: 数字比较器



例 18-4: ADC 数字比较器

```

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;
}

```

例 18-4: ADC 数字比较器 (续)

```
/* 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 过采样数字滤波器

ADC 模块支持最多 6 个过采样数字滤波器。过采样数字滤波器由一个累加器和一个抽取器（取样器）组成，共同作用构成一个低通滤波器。通过以高于所需的采样率对模拟输入采样，然后经过过采样数字滤波器处理数据，从而以降低转换吞吐量为代价提高 ADC 模块的有效分辨率。使用 4x 过采样产生 1 位额外分辨率，16x 过采样产生 2 位额外分辨率，64x 过采样产生 3 位额外分辨率，而 256x 过采样提供 4 位额外分辨率。

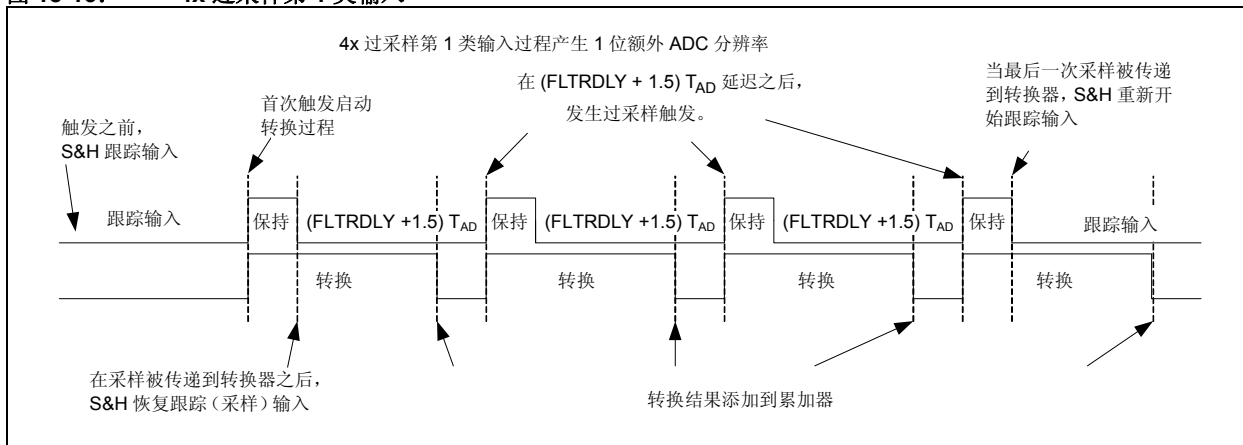
用户应用程序应配置以下位以执行过采样转换：

- 通过 ADC1 滤波器寄存器 (AD1FLTRn<28:26>) 中的过采样滤波器过采样率位 (OVSAM<2:0>) 选择过采样量。
- 设置后续采样的采样时间：
 - 如果使用第 1 类输入，使用 AD1CON1 寄存器 (AD1CON1<31:27>) 中的 (FILTRDLY<4:0>) 位选择重复转换的采样时间
 - 如果使用第 2 类或第 3 类输入，使用 AD1CON2 寄存器 (AD1CON2<23:16>) 中的 (SAMC<7:0>) 位选择采样时间
- 通过配置通道 ID 选择 (CHNLID<5:0>) 位 (AD1FLTRn<21:16>) 选择要被过采样的特定模拟输入
- 如果需要，通过将累加器滤波器全局中断允许 (AFGIEN) 位 (AD1FLTRn<25>) 置 1，将过采样滤波器中断事件包含在全局 ADC 中断中
- 将过采样滤波器累加器使能 (AFEN) 位 (AD1FLTRn<31>) 置 1 使能过采样滤波器

配置过采样数字滤波器后，等待外部转换触发信号以启动过采样过程。用于过采样通道的触发信号导致累加器清零和启动第一次转换。当每个转换请求在转换流水线中正在被转换时，滤波器开始新的采样。每个转换请求均被传递到流水线转换器后，根据 FILTRDLY<4:0> 或 SAMC<7:0> 位的值分别启动新的针对第 1 类、第 2 类和第 3 类输入的采样。将继续该过程直到所需数量的采样（4、8、16、32、64、128 或 256）被转换。对已转换的采样进行求和后，输出传输到过采样滤波器数据输出位 FLTRDATA<15:0> (AD1FLTRn<15:0>) 并且累加器滤波器数据就绪中断标志位 AFRDY (AD1FLTRn<24>) 置 1。

图 18-15 给出了针对第 1 类输入的 4x 过采样。触发之前，第 1 类输入 S&H 跟踪输入信号。触发时启动第一次转换。采样被传输到转换器后，S&H 继续跟踪并启动滤波器延时定时器 (FILTRDLY<4:0>)。滤波器延时定时器到期时，发生新的转换序列。每个采样在转换后被添加至累加器。序列重复直到累加得到 OVSAM<2:0> 位指定的采样数量。最后一个采样被转换后，其值添加至累加器。结果左移，然后保存在 FLTRDATA<15:0> 位中。

图 18-15: 4x 过采样第 1 类输入



在采样输入时，较低优先级的输入可以传输到转换器，且不会破坏过采样时序。较高优先级转换请求会破坏过采样时序，并会产生不可预测的结果。当较高优先级 ADC 转换请求没有产生预期中断时，用户应用程序应该安排发生过采样滤波器启动触发。

图 18-16 给出了使用第 2 类或第 3 类输入的 4x 过采样。触发第 2 类或第 3 类输入将启动 SAMC 所定义时间长度的采样。过采样逻辑产生的再触发信号使用 SAMC<4:0> 位，而不是 FLTRDLY<4:0> 位，设置采样时间。

由于第 2 类和第 3 类输入都使用共用 S&H，过采样阻止较低优先级第 2 类和第 3 类触发信号。较高优先级第 2 类和第 3 类触发将完全破坏过采样过程，因此应完全避免。

图 18-16: 4x 过采样第 2 类输入

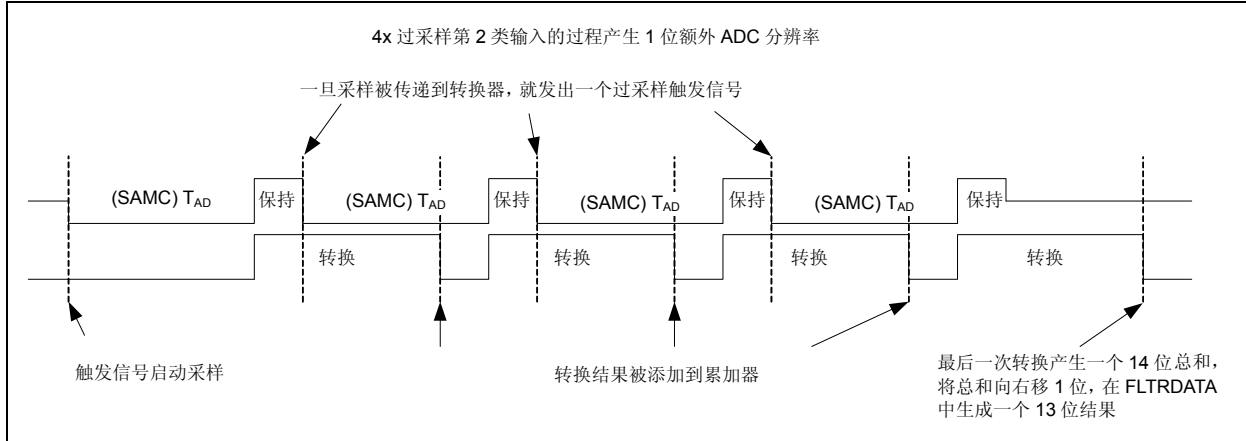
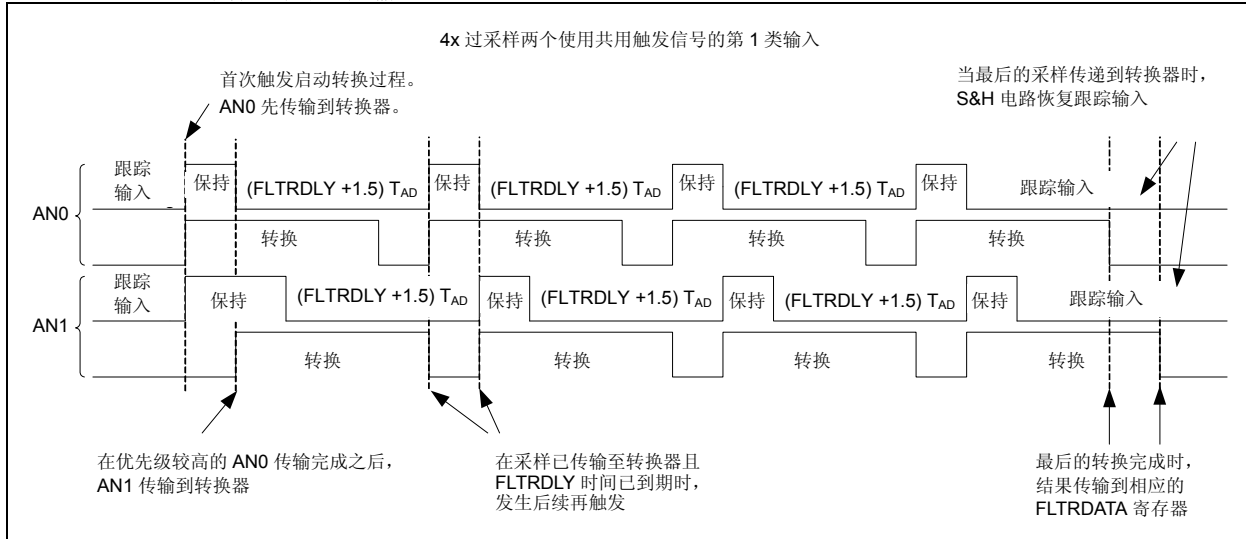


图 18-17 给出了两个第 1 类输入的同时（共用触发信号）4x 过采样。在首次触发时，两个 S&H 电路同时进入保持状态。较高优先级 AN0 首先传输至转换器，紧接着是 AN1 输入。对于过采样过程的剩余阶段，转换时序相差 1 个 T_{AD}。当完成过采样过程时，每个模拟输入创建唯一中断。

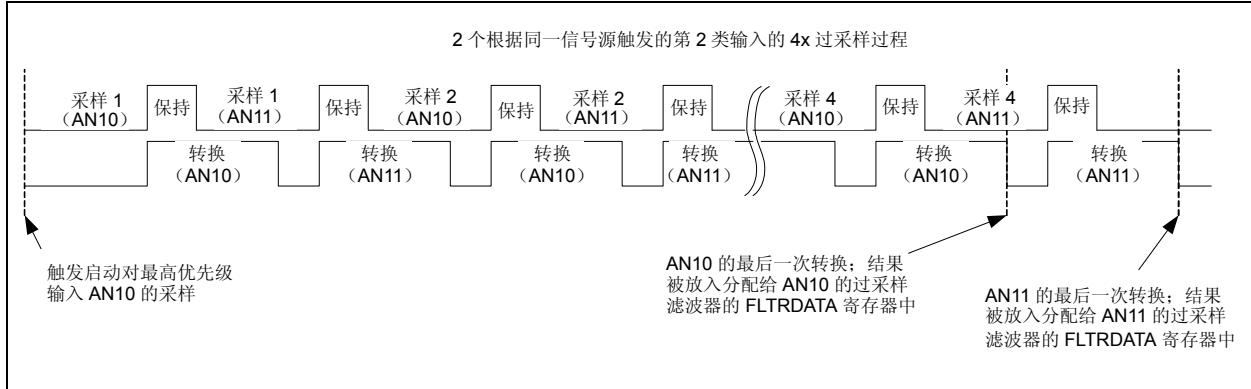
图 18-17: 过采样两个第 1 类输入



当对连接到共用 S&H 的模拟输入使用过采样功能时，系统将交错采样，最高优先级过采样输入和另一个通道交替进行。图 18-18 给出了使用 AN10 和 AN11 的此类配置。在该情况下，如果两个输入 AN10 和 AN11 配置为使用共用独立触发信号，或者设置为两个输入扫描列表，可以获得相同的结果。

由于使用单个共用 S&H，采样必须在两个输入间交替，与采用相同设置过采样单个通道相比，这将使采样率降低 50%。

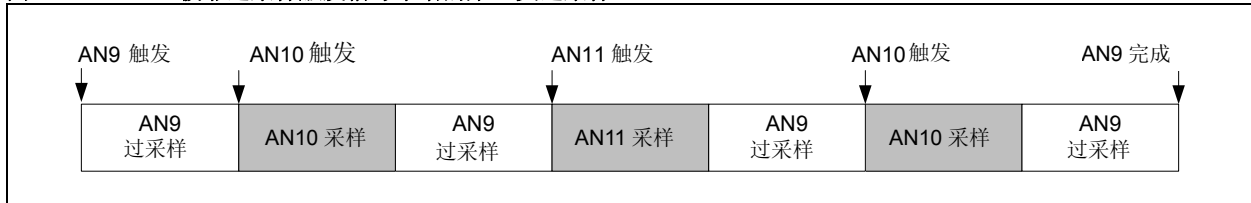
图 18-18: 过采样两个第 2 类输入



需要重点注意的是共用 S&H 上的过采样输入将与另一个较低优先级通道交替。因此，可同时过采样最多两个第 2 类或第 3 类输入。

在过采样第 2 类输入期间，当较低优先级非过采样第 2 类输入被触发时也应用该交错方法，如图 18-19 所示。在此例中，AN9 的过采样与两个 AN10 采样和单个 AN11 采样交错进行。

图 18-19: 被非过采样触发信号中断的第 2 类过采样



如果在过采样过程中正在过采样单个第 2 类输入时，触发较低优先级过采样输入，会产生以下影响，如图 18-20 所示。

图 18-20: 过采样 2 个使用单独触发信号的第 2 类输入



使用过采样时在共用 S&H 上门控输入触发信号来确保过采样没有被破坏很重要，否则会发生意外结果。

例 18-5: ADC 数字过采样滤波器

```
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.TRGSR0 = 1;      // Set AN0 to trigger from software
}
```

例 18-5: ADC 数字过采样滤波器 (续)

```

/* 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 过采样转换时间

公式 18-6 和公式 18-7 介绍了使用第 1 类和第 2 类输入的过采样时序，表示从触发到转换结果可用的时间。

公式 18-6: 第 1 类输入的过采样时间

过采样输入的单个触发总转换时间:

$$T_{C1OVS_CONV} = (N \cdot (FILTDLY + 4) + 8) T_{AD} \quad (\text{见注 1})$$

再触发期间使用的采样时间 (见注 2):

$$T_{C1OVS_SAMP} = (FILTDLY + 1.5) T_{AD}$$

过采样输入触发间的最短时间:

$$T_{C1OVS_TRG_MIN} = (N \cdot (FILTDLY + 4)) T_{AD} \quad (\text{见注 1})$$

其中:

N = 由 OVSAM<2:0> 位 (AD1FLTR<28:26>) 确定的采样总数

T_{AD} = 模数转换时钟周期

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

- 注 1:** 该公式假设在过采样转换序列期间没有发生任何较高优先级第 1 类输入的触发。每个发生的较高优先级第 1 类转换将需要加上 1 个 T_{AD} 至公式。
- 注 2:** 首次采样的采样时间由触发率决定，通常所有第 1 类输入都是如此。该公式说明了后续过采样再触发的采样时间。

公式 18-7: 第 2 类输入的过采样时间

总转换时间:

$$T_{C2OVS_CONV} = (N \cdot (SAMC + 4) + 7)T_{AD}$$

采样时间:

$$T_{C2OVS_SAMP} = (SAMC + 1)T_{AD}$$

其中:

N = 由 OVSAM<2:0> 位 (AD1FLTR<28:26>) 确定的采样总数

T_{AD} = 模数转换时钟周期

$SAMC$ = SAMC<7:0> 位 (AD1CON2<23:16>)

注: 该公式假设较高优先级第 1 类模拟通道上没有等待处理的转换。

18.6 中断

每个 ADC 模块支持从各种中断源触发中断，可单独或全局处理。提前中断功能还可用于补偿中断服务延时。

产生允许的中断之后，CPU 将跳转到为该中断分配的向量处。CPU 将在向量地址处开始执行代码。该向量地址处的用户软件应执行所需的操作（如处理数据结果和清零中断标志），然后退出。关于中断的更多信息和向量地址表的详细信息，请参见《PIC32 系列参考手册》中的第 8 章“中断”（DS61108）和具体器件数据手册中的“中断控制器”章节。

18.6.1 中断源

每个 ADC 能够根据以下事件产生中断：

- ANx 数据就绪事件——在完成模拟输入源（ANx）的转换时，与该输入相关的 ARDYx 位将在 AD1STATn 寄存器中置 1。每个 ARDYx 位能够在置 1 时产生唯一中断。
- 数字比较器事件——当转换的比较条件符合配置和使能的数字比较器时，DCMPED 位将在 AD1CMPCONn 寄存器中置 1。每个数字比较器能够在其 DCMPED 位置 1 时产生唯一中断。
- 过采样滤波器数据就绪事件——当过采样滤波器完成累加 / 抽取过程并且已保存结果，AFRDY 位将在 AD1FLTRn 寄存器中置 1。每个过采样滤波器能够在 AFRDY 位置 1 时产生唯一中断。

18.6.2 中断允许、优先级和向量

前面提到的每个 ADC 事件在其相关的中断允许位 IE 置 1 时，将产生中断。中断标志位 IF、优先级位 IP<2:0> 和子优先级位 IS<1:0> 也与每个事件相关。关于如何允许中断以及中断优先级的更多信息，请参见《PIC32 系列参考手册》中的第 8 章“中断”（DS61108）。前面列出的每个 ADC 事件也有相关的中断向量。关于与每个单独中断相关的向量位置和控制 / 状态位的更多信息，请参见具体器件数据手册中的“中断控制器”章节。

18.6.3 单独和全局中断

使用前面提到的单独中断可以显著优化服务多个 ADC 事件，通过保持每个中断服务程序（Interrupt Service Routine, ISR）专注于有效处理特定事件。此外，可根据执行的任务很容易地将不同的 ISR 分开，从而使用户软件更容易实现和维护。

有些情况下需要单个 ISR 服务多个中断事件。为了实现这项功能，每个 ADC 事件可以进行逻辑或运算以创建单个全局 ADC 中断。当使能 ADC 事件以产生全局中断时，将指向单个中断程序。该单个全局 ISR 负责查询确定中断源并进行相应处理。

对于 ADC 事件：

- 仅当相应的 AGIENx 位（位于 AD1GIRQEN1 或 AD1GIRQEN2 寄存器）置 1 时，每个 ANx 数据就绪中断事件才包含在全局 ADC 中断中
- 仅当 DCMPGIEN 位（AD1CMPCON<6>）置 1 时，数字比较器中断事件才包含在全局 ADC 中断中
- 仅当 AFGIEN 位（AD1FLTRn<25>）置 1 时，过采样滤波器中断事件才包含在全局 ADC 中断中

如 18.6.2 “中断允许、优先级和向量”所述，使用全局中断需要配置其唯一的 IE、IF、IP 和 IS 位以及配置其中断向量。

ADC 中断可配置为单独和全局，或同时使用这两种（一些单独处理，其他位于全局 ISR 中）。

18.6.4 提前中断

AD1CON1 寄存器中的 EIE<2:0> 位在完成转换之前允许中断产生。即使输入仍在进行转换，处理器应用软件可使用“抢先”开始执行进入 ISR。提前中断可提高系统的吞吐量，使用与中断相关的处理器开销重叠完成 ADC 转换。使用 EIE<2:0> 位可降低从采样模拟信号至用户应用软件可使用该数据时的延时。

使用 EIE<2:0> 位将导致在 AD1DATAx 寄存器中的数据实际可用之前置 1 AD1STATx 寄存器中的 ARDYx 位。存储在 EIE<2:0> 位中的值决定在完成转换之前的将该位置 1 的 TAD 时钟周期数。EIE<2:0> 位设置适用于 ADC 的所有 ARDYx 位。如果应用程序使用查询来确定转换是否完成，不应使用提前中断。这包括在全局 ISR 内查询。此外，请注意 EIE<2:0> 位设置不适用于过采样滤波器数据就绪信号 AFRDY。

注： 仅当使用单独的中断程序来检索数据时才应使用提前中断功能，而查询 ARDYx 位时不应使用提前中断，即使在全局中断服务程序中。
--

18.7 节能模式下的操作

休眠和空闲节能模式将 CPU、总线和其他外设的数字活动减到最少，从而有助于将转换噪声降至最小。

18.7.1 休眠模式

当器件进入休眠模式时，SYSCLK 和所有以 SYSCLK 为时钟源运行的外设停止。当选择 SYSCLK 作为时钟源或选择 REFCLK3 作为 ADC 时钟源，并且 SYSCLK 用作 REFCLK3 时钟源时，ADC 也停止。

当 SYSCLK 是时钟源（直接或间接）并且在转换过程中进入休眠模式时，转换会中止。在从休眠模式退出时，转换器不会继续进行已部分完成的转换。器件进入或退出休眠模式不会影响 ADC 寄存器的内容。

如果 ADC 时钟源由休眠模式期间活动的 SYCLK 之外的时钟源提供，ADC 模块可在休眠模式下工作。FRC 时钟源是用于在休眠期间工作的理想选择，但是 REFCLK3 时钟源也可使用（原因是它有一个输入时钟在休眠模式下仍工作）。

休眠模式期间的 ADC 操作会降低转换的数字开关噪声。当转换完成时，该模拟输入的 ARDYx 状态位将置 1，结果将装入相应的 ADC 结果寄存器 (AD1DATAn)。

如果允许任何 ADC 中断 (AD1IE = 1)，ADC 中断发生时器件将从休眠模式唤醒。如果 ADC 中断的优先级高于当前 CPU 优先级，则程序执行将在 ADC ISR 执行后恢复。否则，程序将从将器件置为休眠模式的 WAIT 指令之后的指令继续执行。

为了将数字噪声对 ADC 模块操作的影响降至最低，用户必须选择转换触发源以确保 ADC 转换可在休眠模式下进行。例如，外部中断引脚 (INT0) 转换触发选项 (TRGSRCn<4:0> = 00100) 可用于在器件处于休眠模式下执行采样和转换。

注： 为了使 ADC 模块可以在休眠模式下工作，必须将 ADC 时钟源设置为内部 FRC (ADCSEL<1:0> 位 (AD1CON2<9:8>) = 11)。另外，可以使用 REFCLK3 时钟源；然而，用于 REFCLK3 的时钟源必须在休眠期间工作。对 ADC 时钟配置进行任何更改都需要禁止 ADC。

18.7.2 空闲模式下的 ADC 操作

对于 ADC，空闲模式停止位 ADSIDL (AD1CON1<13>) 用于选择 ADC 模块在空闲模式下是停止还是继续工作。如果 ADSIDL = 0，当器件进入空闲模式时，ADC 模块将继续正常工作。如果允许任何 ADC 中断，ADC 中断发生时器件将从空闲模式唤醒。如果 ADC 中断的优先级大于当前 CPU 优先级，则程序执行将在 ADC ISR 执行后恢复。否则，程序将从将器件置于空闲模式的 WAIT 指令之后的指令处继续执行。

如果 ADSIDL = 1，在空闲模式下 ADC 模块将停止。如果器件在一次转换过程中进入空闲模式，转换将中止。在从空闲模式退出时，转换器不会继续进行已部分完成的转换。

18.7.3 ADC 低功耗模式

通过将 ADC 低功耗位 LOWPWR (AD1CON2<13>) 置 1 将 ADC 模块置于低功耗状态。与使用 ADCEN 位 (AD1CON1<15>) 禁止和重新使能 ADC 模块相比，使用该低功耗模式提供显著较快的模块重启。这是因为使用 ADCEN 位 (AD1CON1<15>) 禁止和重新使能 ADC 模块需要执行 ADC 校准长序列。而在清零 LOWPWR 位 (AD1CON2<13>) 后重启 ADC 模块仅需要 2 * TAD 周期，因为 ADC 偏压发生器在 ADC 低功耗模式下不关闭。

注： 从 ADC 低功耗模式退出后紧接着进行的前 5 个转换的精度很可能比器件数据手册中规定的精度低。

18.8 复位的影响

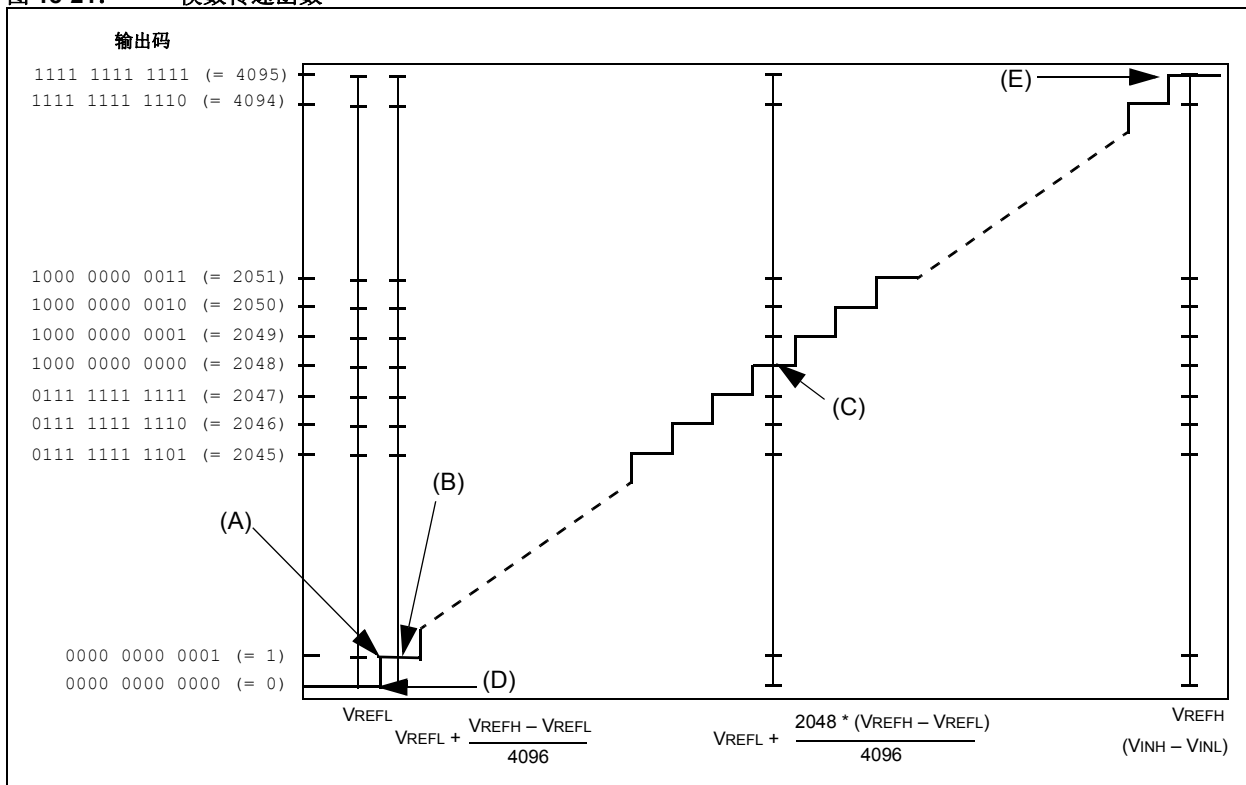
发生任何复位事件后，所有 ADC 控制和状态寄存器将复位为其默认值，且控制位处于无效状态。这会禁止 ADC 模块并且将模拟输入引脚设置为模拟输入模式。任何正在进行的转换将终止，结果不会被写入结果缓冲区。在器件复位期间，AD1DATAn 寄存器中的值将初始化为 0x00000000。

18.9 传递函数

图 18-21 给出了 12 位 ADC 的典型传递函数。对输入电压差值 ($V_{INH} - V_{INL}$) 与参考电压差值 ($V_{REFH} - V_{REFL}$) 进行比较。

- 当输入电压为 $(V_{REFH} - V_{REFL}/8192)$ 或 0.5 LSb 时，发生第一次代码转换 (A)
- 00 0000 0001 代码中点位于 $(V_{REFH} - V_{REFL}/4096)$ 或 1.0 LSb (B)
- 10 0000 0000 代码中点位于 $(2048 * (V_{REFH} - V_{REFL})/4096)$ (C)
- 小于 $(1 * (V_{REFH} - V_{REFL})/8192)$ 的输入电压会转换为 00 0000 0000 (D)
- 大于 $(8192 * (V_{REFH} - V_{REFL})/8192)$ 的输入电压会转换为 11 1111 1111 (E)

图 18-21: 模数传递函数



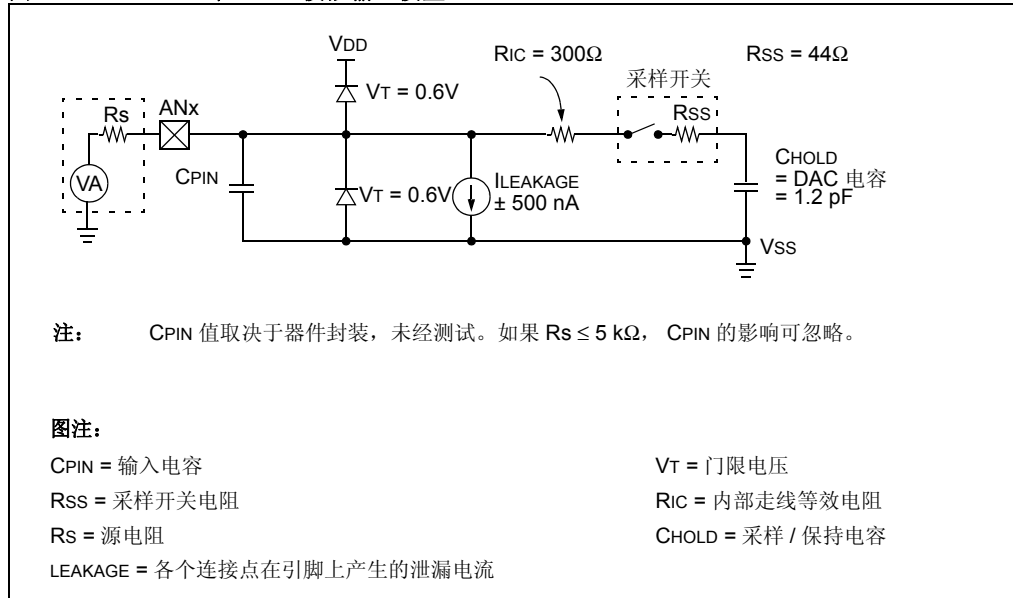
18.10 ADC 采样要求

图 18-22 给出了 12 位 ADC 的模拟输入模型。模数转换的总采集时间是内部电路稳定时间和保持电容充电时间的函数。

为了使 ADC 模块达到规定的精度，必须让充电保持电容 (CHOLD) 充分充电至模拟输入引脚上的电压。模拟输出信号源阻抗 (R_s)、片内走线等效电阻 (R_{ic}) 和内部采样开关阻抗 (R_{ss}) 共同直接影响着电容 CHOLD 充电所需的时间。因此，模拟信号源的合并阻抗必须足够小，以便在选择的采样时间内对保持电容充分充电。在每次采样操作前，内部保持电容将处于放电状态。

两次转换之间应留出至少 1 个 TAD 时间段作为采集时间。更多信息，请参见具体器件数据手册的“电气特性”章节。

图 18-22: 12 位 ADC 模拟输入模型



18.11 连接注意事项

因为模拟输入采用静电放电 (Electrostatic Discharge, ESD) 保护，它们通过二极管连接到 VDD 和 VSS。这就要求模拟输入电压必须介于 VDD 和 VSS 之间。如果输入电压超出此范围 0.3V 以上 (任一方向上)，就会有一个二极管正向偏置，如果超过输入电流规范可能会损坏器件。

有时可以通过外接一个 RC 滤波器来对输入信号进行抗混叠滤波。应选择合适的 R (电阻式) 元件以确保达到采集时间要求。任何通过高阻抗连接到模拟输入引脚上的外部元件 (如电容和齐纳二极管等) 在引脚上的泄漏电流都应极小。

18.12 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改即可使用，但在使用时会受到一定限制。当前与 12 位流水线型模数转换器（ADC）模块相关的应用笔记有：

标题	应用笔记编号
《理解 A/D 转换器的性能参数》	AN693
《采用过采样技术实现更高的 ADC 分辨率》	AN1152

注： 如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

18.13 版本历史

版本 A (2013 年 5 月)

这是本文档的初始版本。

版本 B (2013 年 11 月)

该版本包含以下更新：

- 更新了共用 S&H 5 框图 (见图 18-3)
- 在 ADC1 控制寄存器 1 (AD1CON1) 中向 STRGSRC<4:0> 位添加了注 4 (见寄存器 18-1)
- 更新了 ADC1 输入模式控制寄存器 (AD1MOD) 中的 SHxALT 位 (见表 18-1 和寄存器 18-4)
- 添加了示例代码 (见例 18-1 至例 18-5)
- 对整个文档的文本和格式进行了少量更新

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应尽的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

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

商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、FlashFlex、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rPIC、SST、SST 徽标、SuperFlash 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、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 和 Z-Scale 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

GestIC 和 ULPP 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2013-2014, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62077-842-5

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器 and 模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **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

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-3666
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

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

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

印度 India - Pune
Tel: 91-20-3019-1500

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

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

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

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

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

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

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

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

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

欧洲

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

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

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

德国 Germany - Dusseldorf
Tel: 49-2129-3766400

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

德国 Germany - Pforzheim
Tel: 49-7231-424750

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

意大利 Italy - Venice
Tel: 39-049-7625286

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

波兰 Poland - Warsaw
Tel: 48-22-3325737

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

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

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