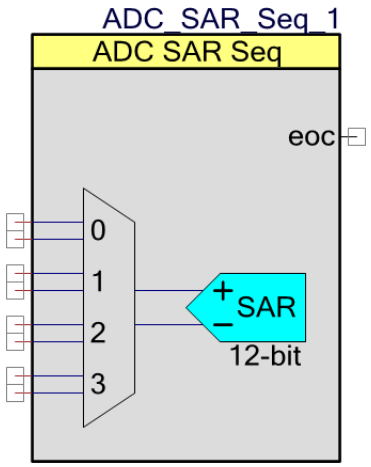
序列逐次逼近 ADC（ADC_SAR_Seq）

1.10



特性

- 支持 PSoC 5LP 器件
- 可选分辨率（8、10 或 12 位）和采样率（最高达 1 Msps）
- 自动扫描多达 64 个单端或 32 个差分通道，或仅扫描单个输入

  注意：输入通道的实际最大数量取决于特定 PSoC 器件和封装中可路由模拟 GPIO 的数量。

概述

通过序列 SAR ADC 组件，您能够在 PSoC 5LP 上轻易配置和使用不同操作模式的 SAR ADC。您还拥有原理图和固件级别的支持，可在 PSoC Creator 的设计和项目中方便的使用序列 SAR ADC。您还可以配置多个自动扫描的模拟通道，然后将每个通道的扫描结果存储到独立的 SRAM 位置内。

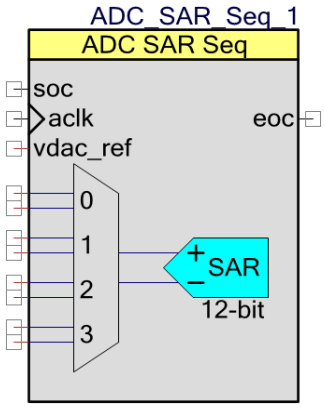
何时使用 ADC_SAR_Seq

通常，序列 SAR ADC 组件可在无 CPU 参与的情况下以高速的采样率自动实现多通道数据采样。

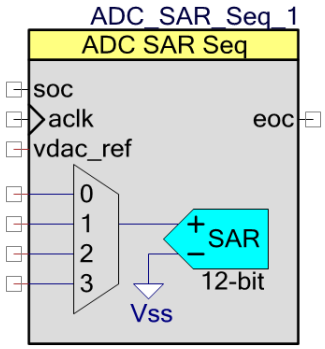
输入/输出连接

本节介绍 ADC_SAR_Seq 的输入和输出连接。I/O 列表中的星号（*）表示，在 I/O 说明部分中所列出的特定条件下，该 I/O 可能不可见。

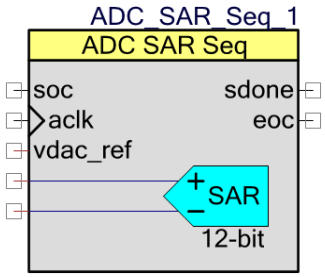
差分输入



单端输入



差分输入 – 1通道（sdone可见）



正输入 — 模拟

此输入是 ADC_SAR_Seq 的正向模拟信号输入。转换结果是一个正输入信号减去电压参考的函数。电压参考是输入信号或 V_{SSA}。

负输入 — 模拟*

显示时，此可选输入是 ADC_SAR_Seq 的负向模拟信号（或参考）输入。转换结果取决于正输入减去负输入。在将 **Input Range**（输入范围）参数设置为差分模式之一时可看到此引脚。

vdac_ref — 输入*

VDAC 参考（vdac_ref）是可选引脚。如果选择 **Vssa to VDAC*2 (Single Ended)**（Vssa 至 VDAC*2（单端））或 **0.0 +/- VDAC (Differential)**（0.0 +/- VDAC（差分））输入范围，则可看到该引脚；否则，此 I/O 会隐藏。

注意：只能将此引脚连接到 VDAC 组件输出。请勿将其连接到任何其他信号。

soc — 输入*

开始转换（soc）是可选引脚。如果您选择**硬件**触**发**采样模式，该输入可见。该输入的上升沿会触发 ADC 转换。此信号必须同 ADC_SAR_Seq 时钟同步。如果将 **Sample Mode**（采样模式）参数设置为 **Free Running**（自由运行）或 **Software trigger**（软件触发），则此 I/O 会隐藏。

aclk — 输入 *

如果将 **Clock Source**（时钟源）参数设置为 **External**（外部），则可看到此可选引脚；否则，此引脚会隐藏。该时钟确定了作为转换方式和分辨率函数的转换速率。

eoc — 输出

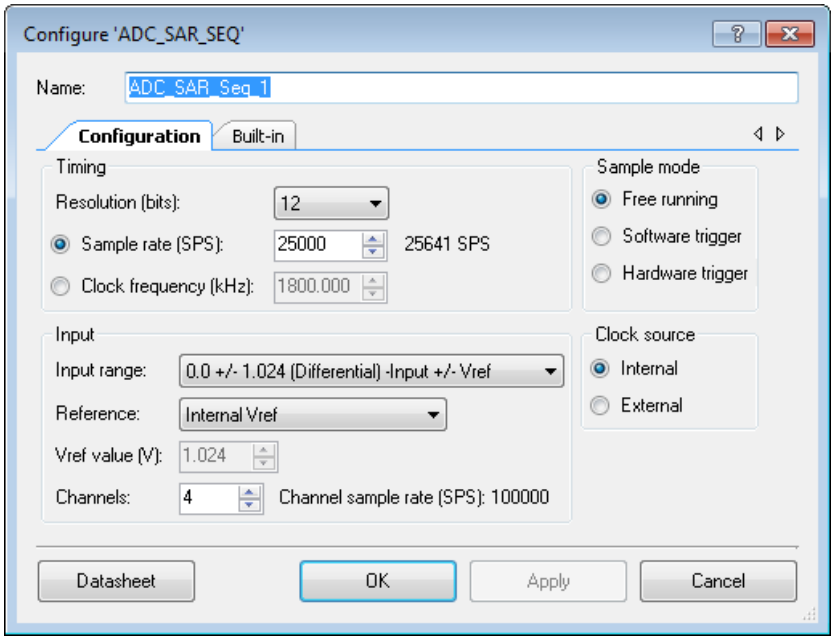
结束转换（eoc）端口输出上升沿表示一个转换周期已经完成。此时，所有通道的转换结果准备就绪，可从 SRAM 读取。也可将内部中断连接到此信号，或者可以连接自己的中断。

sdone — 输出*

该输出的上升沿表示采样已经完成。这是从 ADC SAR 组件上 EOS 输出的直接连接。需要对组件进行配置，以执行单通道操作，才能看见此信号。

组件参数

将一个 ADC_SAR_Seq 组件拖放到您的设计窗口上，并双击以打开 **Configure**（配置）对话框。



ADC_SAR_Seq 具有以下参数。粗体表示默认选项。

时序

分辨率（以位为单位）

设置 ADC_SAR_Seq 的分辨率。

分辨率	说明
12	将分辨率设置为12位
10	将分辨率设置为10位
8	将分辨率设置为8位

采样率（SPS）

此参数显示的是采样率，适用于所有通道。它以每秒采样数为单位。根据时钟频率和通道总数，可自动计算此参数。通道采样率显示在各通道的数字上下控制的附近。

时钟频率（kHz）

默认情况下，该文本框为只读（始终为灰色）区域，用于显示所选工作条件（分辨率和转换速率）所需的时钟速率。当这两个条件中的任一条件更改或两个条件都更改时，会更新该文本框。如果将 **Clock Source** 设置为 Internal（内部），则可更改该文本框。时钟频率范围介于 1 MHz 与 18 MHz 之间。占空比应为 50%。使最小脉冲宽度大于 25.5 ns。如果所设置的时钟频率不在此范围内，则 PSoC Creator 在构建过程中将发出错误警告。发生错误时，在全局资源配置时钟编辑器中更改主设备时钟。

采样模式

该参数确定了 ADC 的工作方式。

采样模式	说明
自由运行	ADC_SAR_Seq持续运行
软件触发器	StartConvert() API的调用会启动一次对所有通道进行的转换
硬件触发	SOC引脚上的一个上升沿脉冲一次对所有通道进行的转换

输入

输入范围

该参数按给定的输入范围配置 ADC。无论输入范围的设置情况如何，连接至 PSoC 的模拟信号都必须在 V_{SSA} 与 V_{DDA} 的范围内。

输入范围	说明
0.0至2.048 V（单端） **0至Vref*2**	当使用内部参考（1.024 V）时，可用输入范围为0.0至2.048 V。ADC配置为单端输入模式下运行，同时负输入在内部连接到Vrefhi_out。如果使用外部参考电压，可用的输入范围为0.0至Vref*2。
Vssa至Vdda（单端）	此模式使用V_{DDA}/2参考；可用输入范围涵盖完整模拟电源电压。ADC置于单端输入模式，同时负输入在内部连接到Vrefhi_out。如果使用外部参考电压，可用的输入范围为0.0至Vref*2。
Vssa至VDAC*2（单端）	此模式使用VDAC参考，该参考应连接到vdac_ref引脚。可用的输入范围介于Vssa到VDAC*2伏之间。ADC配置为在单端输入模式下运行，同时负输入在内部连接到Vrefhi_out。
0.0±1.024V（差分） 负输入± Vref	此模式针对差分输入配置。当使用内部参考（1.024 V）时，输入范围为负输入±1.024 V。 例如，如果负输入连接到2.048 V，则可用输入范围为2.048±1.024 V或1.024至3.072 V。对于既需要扫描单端又需要扫描差分信号的系统，在扫描单端输入时将负输入连接到Vssa。 您可以使用外部参考源以提供更广的操作范围。可以使用相同等式“负输入±Vref”来计算可用的输入范围。
0.0±Vdda（差分） 负输入±Vdda	此模式配置用于差分输入，与输入电压成比例关系。输入范围为负输入±Vdda。对于需要扫描单端和差分信号的系统，在扫描单端输入时，将负输入连接至Vssa。如果使用外部参考电压，则可用输入范围为负输入±Vref。
0.0±Vdda/2（差分） 负输入±Vdda/2	此模式配置用于差分输入，与输入电压成比例关系。输入范围为负输入±Vdda/2。对于需要扫描单端和差分信号的系统，在扫描单端输入时，将输入连接至Vssa。如果使用外部参考电压，则可用输入范围为负输入±Vref。
0.0±VDAC（差分） 负输入±VDAC	该模式配置为用于差分输入，使用VDAC参考，该参考应连接到vdac_ref引脚。输入范围为负输入 ±VDAC。对于需要扫描单端和差分信号的系统，在扫描单端输入时，将负输入连接至Vssa。

参考电压

此参数选择 ADC_SAR 参考配置的开关。

参考电压	说明
内部Vref	使用内部参考。该选项允许的最大采样率为100,000 sps。对于较高采样率，请使用 **Internal Vref, bypassed**选项。
内部Vref，旁路	使用内部参考；必须将一个旁路电容器置于SAR1的引脚P0[2]*上或SAR0的引脚P0[4]*上。
外部Vref	在SAR1的引脚P0[2]*上或SAR0的引脚P0[4]上使用外部参考。

* 如果内部数字开关引发的噪音超过某个应用对模拟性能的要求，建议使用外部旁路电容。若要使用该选项，请将端口引脚 P0[2]或 P0[4]配置为模拟高阻引脚，并连接到 0.01 µF 至 10 µF 范围内的外部电容。

注意：ADC_SAR 和 ADC_DelSig 组件使用相同的内部参考。如果两类 ADC 同时需要使用内部参考，请使用 **Internal Vref, bypassed** 选项以获得最佳的性能。

电压参考

电压参考用于计入应用编程接口一节中讨论的 ADC 的电压转换函数。当使用内部参考时，该参数为只读。当使用外部参考时，可以编辑该值以匹配外部参考电压。

- 选择输入范围 **Vssa 至 Vdda**、**负输入+/- Vdda** 或**负输入+/- Vdda/2** 时，其值由“全局资源配置”（DWR）窗口中系统 System 选项卡内的 V_{DDA} 设置演变而来。
- 当选择输入范围 **Vssa 至 VDAC*2** 或**负输入+/- VDAC** 时，请输入 VDAC 电源电压值。

注意：输入范围和参考电压受 V_{DDA} 电压限制。

时钟源

通过该参数，可以选择 ADC_SAR 模块内部的时钟或外部时钟。

ADC_Clock	说明
内部	使用ADC_SAR_Seq的内部时钟。
外部	使用外部时钟。时钟源可以是模拟、数字或由其他组件生成。

应用编程接口

通过应用编程接口（API），您可以使用软件进行组件配置。该表列出了每个函数的接口，并进行了说明。以下各节将更详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称 “ADC_SAR_Seq _1”分配给指定设计中组件的第一个实例。您可以将该实例重命名为符合标识符语法规则的任意唯一值。实例名称会成为所有全局函数名称、变量和符号常量的前缀。出于可读性考虑，下表中使用的实例名称为 “ADC_SAR_Seq”。

函数	说明
ADC_SAR_Seq _Start()	使ADC_SAR_Seq通电并复位所有状态
ADC_SAR_Seq _Stop()	停止ADC_SAR_Seq转换并将功耗减少到最小值
ADC_SAR_Seq _SetResolution()	设置ADC_SAR_Seq的分辨率。
ADC_SAR_Seq _StartConvert()	开始转换
ADC_SAR_Seq _StopConvert()	停止转换
ADC_SAR_Seq _IRQ_Enable()	将内部IRQ连接至eoc。该API使能了内部ISR
ADC_SAR_Seq _IRQ_Disable()	将内部IRQ连接至eoc。该API禁用了内部ISR
ADC_SAR_Seq_IsEndConversion()	如果转换完成，将返回非零值
ADC_SAR_Seq _GetAdcResult()	返回带符号的16位转换结果，该结果被存储在ADC SAR数据寄存器内，而不是存储在结果缓冲区内
ADC_SAR_Seq _GetResult16()	返回特定通道上带符号的16位转换结果
ADC_SAR_Seq _SetOffset()	设置ADC_SAR_Seq的偏移
ADC_SAR_Seq _SetScaledGain()	设置ADC_SAR_Seq增益，其单位为每10伏计数
ADC_SAR_Seq_CountsTo_Volts()	将ADC_SAR_Seq计数转换为单位为伏的浮点电压值
ADC_SAR_Seq_CountsTo_mVolts()	将ADC_SAR_Seq计数转换为单位为毫伏的电压值
ADC_SAR_Seq_CountsTo_uVolts()	将ADC_SAR_Seq计数转换为单位为微伏的电压值
ADC_SAR_Seq _Sleep()	停止ADC_SAR_Seq操作，并保存用户配置
ADC_SAR_Seq _Wakeup()	恢复并使能用户配置
ADC_SAR_Seq _Init()	初始化自定义程序所提供的默认配置
ADC_SAR_Seq _Enable()	使能ADC_SAR_Seq的时钟，并使其通电
ADC_SAR_Seq _SaveConfig()	保存当前用户配置
ADC_SAR_Seq _RestoreConfig()	恢复用户配置

全局变量

变量	说明
ADC_SAR_Seq _initVar	该变量指示ADC是否已初始化。该变量初始化为0，并在第一次调用ADC_SAR_Seq _Start()时设置为1。这样，第一次调用ADC_SAR_Seq _Start()子程序后，组件不用重新初始化即可重启。如需重新初始化组件，调用ADC_SAR_Seq _Start()或ADC_SAR_Seq _Enable()函数之前，先调用ADC_SAR_Seq _Init()函数。
ADC_SAR_finalArray	此阵列包含每次在生成EOC脉冲和设置EOC状态后所有通道的有效转换结果内。 **注意**：直接使用此阵列的值（而不使用ADC_SAR_Seq_GetResult16()API函数）时，应该考虑一个事实，就是各通道将被倒序地扫描，从而最后通道的转换结果将放置于ADC_SAR_finalArray[0]上。 由于没有考虑到ADC_SAR_offset，请勿在任何差分模式下直接使用此阵列。相反，使用ADC_SAR_Seq_GetResult16()处理此阵列。如果直接使用它，您必须要手动处理ADC_SAR_offset。

void ADC_SAR_Seq_Start(void)

说明： 这是开始执行组件操作的首选方法。ADC_SAR_Seq_Start()函数先设置initVar变量，调用ADC_SAR_Seq_Init()函数，然后调用ADC_SAR_Seq_Enable()函数。

参数： 无

返回值： 无

其他影响： 如果已设置initVar变量，则该函数仅调用ADC_SAR_Seq_Enable()函数。

void ADC_SAR_Seq_Stop(void)

说明： 停止ADC_SAR_Seq转换并将功耗减少到最小值。

注意：该API不会断开ADC电源，但是会将电源降到最小值。该器件有一个限制，即：器件断电时，与某些模拟资源的连接会不可靠。当停止使用该资源中的组件时，这种不可靠性会在静默失败中表现出来（例如模拟组件中出现不可预见的失败结果）。

参数： 无

返回值： 无

其他影响： 无

void ADC_SAR_Seq_SetResolution(uint8 resolution)

说明： 设置GetResult16()和GetAdcResult() API的分辨率。

参数： uint8 resolution：设置分辨率

参数名称	值	说明
ADC_SAR_Seq_ADC_BITS_12	12	将分辨率设置为12位。
ADC_SAR_Seq_ADC_BITS_10	10	将分辨率设置为10位。
ADC_SAR_Seq_ADC_BITS_8	8	将分辨率设置为8位。

返回值： 无

其他影响： 在转换周期过程中不能更改ADC分辨率。建议的最佳实践是使用ADC_SAR_Seq_StopConvert()停止转换，更改分辨率，然后使用ADC_SAR_Seq_StartConvert()重新启动转换。

如果决定在调用此API之前不停止转换，请使用ADC_SAR_Seq_IsEndConversion()等到转换完成，然后再更改分辨率。

如果在转换过程中调用ADC_SAR_Seq_SetResolution()，则在当前转换完成之前，分辨率将不会更改。对于另外6 + “新分辨率（以位为单位）”个时钟周期，数据在新分辨率中不可用。在调用ADC_SAR_Seq_SetResolution()函数之后，可能需要添加此时钟周期数的延迟，数据才能再次有效。

通过在ADC_SAR_Seq计数和施加的输入电压之间计算正确转换，可影响ADC_SAR_Seq_CountsTo_Volts()、 ADC_SAR_Seq_CountsTo_mVolts()和ADC_SAR_Seq_CountsTo_uVolts()。计算取决于分辨率、输入范围和电压参考。

void ADC_SAR_Seq_StartConvert(void)

说明： 该函数将强制ADC启动一个转换。在自由运行模式中，ADC_SAR_Seq连续运行。在软件触发模式中，该函数还充当软件版本的SOC，并且必须通过ADC_SAR_Seq_StartConvert()触发每个转换。在硬件模式下，该函数不可用。

参数： 无

返回值： 无

其他影响： 通过调用ADC_StartConvert()函数，可禁用外部SOC引脚。

void ADC_SAR_Seq_StopConvert(void)

说明： 该函数将强制ADC停止转换。如果当前正在执行转换，将会完成该转换，并且没有启动其他转换。该函数仅适用于自行运行模式。

参数： 无

返回值： 无

其他影响： 在自由运行和软件触发模式中，该函数将软件版本的SOC设置低级别并将SOC源切换为硬件SOC输入（硬件触发模式中）。

void ADC_SAR_Seq_IRQ_Enable(void)

说明： 转换结束后，该函数将使能中断。必须启动全局中断，以实现ADC中断。若要使能全局中，请在使能任何中断之前，在*main.c*文件中使能全局中断宏“CYGlobalIntEnable;”。

参数： 无

返回值： 无

其他影响： 使能中断。读取结果将清除中断。

void ADC_SAR_Seq_IRQ_Disable(void)

说明： 转换结束后，禁用中断。

参数： 无

返回值： 无

其他影响： 无

uint32 ADC_SAR_Seq_IsEndConversion(uint8 retMode)

说明： 根据retMode参数的设置情况，会立即返回转换状态或不会返回转换状态直到转换完成（锁住）。

参数： uint8 retMode：检查转换返回模式。下表说明了可用选项。

选项	说明
ADC_SAR_SEQ_RETURN_STATUS	立即返回状态。如果返回值为零，则表示转换未完成，应重试该函数，直至返回非零值。
ADC_SAR_SEQ_WAIT_FOR_RESULT	在ADC_SAR_Seq转换完成之前不返回结果。

返回值： uint32：如果返回非零值，则最后一次转换已完成。如果返回值为零，则ADC_SAR_Seq仍在计算最后的结果。

其他影响： 该函数可读取转换结束的状态，并在读取后会清除。

int16 ADC_SAR_Seq_GetAdcResult(void)

说明： 获取SAR DATA寄存器中存在的数据，而不是结果缓冲器中的数据。

参数： 无

返回值： int16：最后一次ADC转换结果。

其他影响： 将ADC计数转换为2的补码形式。

int16 ADC_SAR_Seq_GetResult16(uint16 chan)

说明： 返回通道“chan”的转换结果。

参数： 无

返回值： int16：返回转换数据如带符号的16位整数电压值

其他影响： 将ADC计数转换为2的补码形式。

void ADC_SAR_SeqSetOffset(int32 offset)

说明： 设置ADC_SAR_Seq_CountsTo_Volts()、 ADC_SAR_Seq_CountsTo_mVolts()和ADC_SAR_Seq_CountsTo_uVolts()所用的ADC_SAR_Seq偏移，以便在计算电压转换，从给定读数中减去该偏移量。

参数： int32 offset：如果该输入短路或连接到相同输入电压，则该值为测量值。

返回值： 无

其他影响： 通过减去给定偏移，可影响ADC_SAR_Seq_CountsTo_Volts()、ADC_SAR_Seq_CountsTo_mVolts()和ADC_SAR_Seq_CountsTo_uVolts()。

void ADC_SAR_Seq_SetScaledGain(int32 adcGain)

说明： 为以下电压转换函数设置每10伏电压的ADC_SAR_Seq增益计数。默认情况下，该值由参考和输入范围设置设定。该值仅可用于进一步校准具有已知输入的ADC_SAR_Seq，或仅在ADC_SAR_Seq使用外部参考的情况下使用。要校准该增益，向ADC输入提供参考电压并通过使用万用表测量此电压。使用以下公式，以计算增益系数：

$$adcGain = \frac{counts \times 10}{V_{measured}}$$

其中，**counts**（计数）是从ADC_SAR_Seq_GetResult16() 值返回的值，$V_{measued}$使用万用表进行测量，它单位为伏特。

参数： int32 adcGain：每10伏电压的ADC_SAR_Seq增益计数。

返回值： 无

其他影响： 通过在ADC计数和施加的输入电压之间计算正确转换，可影响ADC_SAR_Seq_CountsTo_Volts()、 ADC_SAR_Seq_CountsTo_mVolts()和ADC_SAR_Seq_CountsTo_uVolts()。

float32 ADC_SAR_Seq_CountsTo_Volts(int16 adcCounts)

说明： 将ADC_SAR_Seq输出转换为单位为伏的浮点电压值。例如，如果测得的ADC_SAR_Seq输出为0.534 V，则返回值为0.534。电压计算取决于电压参考值。当Vref基于Vdda时，用于Vdda的值针对“全局资源配置”（DWR）的System（系统）选项卡中项目设置。

参数： int16 adcCounts：ADC_SAR_Seq转换的结果。

返回值： float32：返回单位为伏的电压值

其他影响： 无

int32 ADC_SAR_Seq_CountsTo_mVolts(int16 adcCounts)

说明： 将ADC_SAR_Seq输出转换为以mV为单位的32位整数电压值。例如，如果测得的ADC_SAR_Seq输出为0.534 V，则返回值为534。电压计算取决于电压参考值。当Vref基于Vdda时，用于Vdda的值针对“全局资源配置”（DWR）的System（系统）选项卡中项目设置。

参数： Int32 adcCounts：ADC_SAR_Seq转换的结果。

返回值： int32：以mV为单位的电压值

其他影响： 无

int32 ADC_SAR_Seq_CountsTo_uVolts(int16 adcCounts)

说明： 将ADC_SAR_Seq输出转换为以mV为单位的32位整数电压值。例如，如果测得的ADC_SAR_Seq输出为0.534 V，则返回值为534000。电压计算取决于电压参考值。当Vref基于Vdda时，用于Vdda的值针对“全局资源配置”（DWR）的System（系统）选项卡中项目设置。

参数： int16 adcCounts：ADC转换的结果

返回值： int32：以µV为单位的电压值

其他影响： 无

void ADC_SAR_Seq_Sleep(void)

说明： 这是组件准备进入睡眠模式时的首选子程序。ADC_SAR_Seq_Sleep()例程保存当前组件状。然后该程序调用ADC_SAR_Seq_Stop()函数，并调用ADC_SAR_Seq_SaveConfig()保存硬件配置。

在调用CyPmSleep()或CyPmHibernate()函数之前调用ADC_SAR_Seq_Sleep()函数。有关功耗管理函数的详细信息，请参考*《系统参考指南》*。

参数： 无

返回值： 无

其他影响： 无

void ADC_SAR_Seq_Wakeup(void)

说明： 该函数是将组件恢复到调用ADC_SAR_Seq_Sleep()时状态的首选子程序。ADC_SAR_Seq_Wakeup()函数调用ADC_SAR_Seq_RestoreConfig()函数以恢复配置。如果组件在调用ADC_SAR_Seq_Sleep()函数前已启用，则ADC_SAR_Seq_Wakeup()函数将重新启用组件。

参数： 无

返回值： 无

其他影响： 调用ADC_SAR_Seq_Wakeup()函数前未调用ADC_Sleep()或ADC_SAR_Seq_SaveConfig()函数可能会产生意外行为。

void ADC_SAR_Seq_Init(void)

说明： 根据自定义程序“Configure”对话框中的设置初始化或恢复组件。无需调用ADC_SAR_Seq_Init()，因为ADC_SAR_Seq_Start()子程序会调用该函数并是开始组件操作的首选方法。

参数： 无

返回值： 无

其他影响： 根据自定义程序“Configure”对话框中的内容，设置所有寄存器。

void ADC_SAR_Seq_Enable(void)

说明： 激活硬件并开始执行组件操作。无需调用ADC_SAR_Seq_Enable()，因为ADC_SAR_Seq_Start()子程序会调用该函数，这是开始组件操作的首选方法。

参数： 无

返回值： 无

其他影响： 无

void ADC_SAR_Seq_SaveConfig(void)

说明： 此函数会保存组件配置以及非保留寄存器。它还保存Configure对话框中定义的或通过相应API修改的当前组件参数值。该函数由ADC_SAR_Seq_Sleep()函数调用。

参数： 无

返回值： 无

其他影响： 保留所有ADC_SAR_Seq配置寄存器。不会执行该函数，以备将来使用。在此显示了该函数，以便使API在整个组件中保持一致。

void ADC_SAR_Seq_RestoreConfig(void)

说明： 该函数会恢复组件配置和非保留寄存器。它还将组件参数值恢复为在调用ADC_SAR_Seq_Sleep()函数之前的值。

参数： 无

返回值： 无

其他影响： 调用该函数前未调用ADC_SAR_Seq_Sleep()或ADC_SAR_Seq_SaveConfig()函数可能会产生意外行为。不会执行该函数，以备将来使用。在此提供该函数，以便使API在整个组件中保持一致。

MISRA 合规性

本节介绍了MISRA-C:2004合规性和本器件的偏差情况。有两种差异的类型：

- 项目偏差 — 适用于所有 PSoC Creator 组件的偏差
- 特定偏差 — 仅适用于该组件的偏差

本节介绍了有关组件特定偏差的信息。*系统参考指南*的“MISRA 合规性”章节中介绍了项目偏差以及有关 MISRA 合规性验证环境的信息。

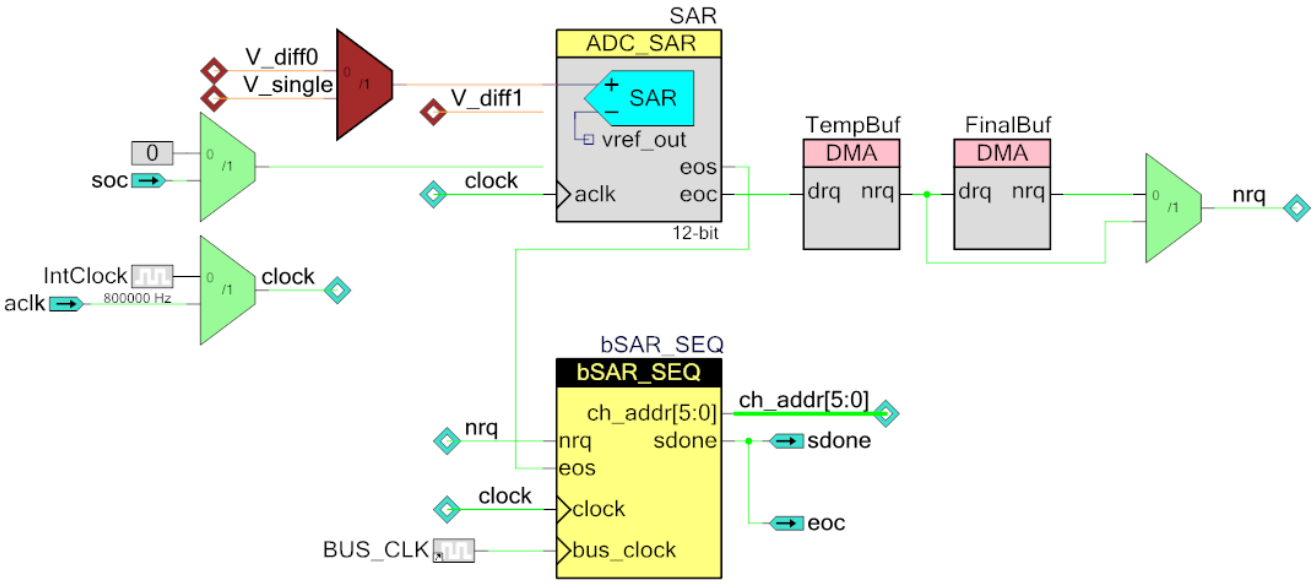
ADC_SAR_Seq 器件具有这些特定偏差：

MISRA-C:2004规则	规则类别（必须/建议）	规则说明	偏差说明
10.1	R	在下面情况下，整型表达式的数值不应隐式转换为不同的底层类型。 a）没有转换到相同符号的更宽的整数类型，或 b）表达式是复合的，或 c） 表达式不是一个常量，而是函数参数，或 d） 表达式不是一个常量，而是返回表达式。	DMA组件提供了各个通用整数类型的定义。
13.2	A	对一个非零值进行显性测试，除非操作数是有效的布尔（Boolean）值。	DMA组件提供了对各个通用整数类型的定义。通过对这些定义进行“逻辑或”（OR）运算，可以得到各函数参数。
17.4	R	数组索引是唯一允许的指针运算形式。	使用数组索引形式的指针运算访问DMA结构。

此组件有以下嵌入式组件，即：ADC_SAR、DMA、中断和硬件 AMUX。MISRA 合规性与特定偏差的相关信息，请参见相应组件数据手册。

功能说明

该框图介绍了 ADC SAR 组件如何对来自硬件 AMUX 的模拟信号进行采样和转换。第一个 DMA 通道逐次将结果从 SAR ADC 转移到临时 RAM 缓冲区内。当所有通道的完整数据集均位于临时 RAM 缓冲区内时，会通过一个脉冲段将临时 RAM 缓冲区中的内容传输到结果（SRAM）缓冲区内。可使用临时缓冲区收集数据，从而每次扫描后将结果缓冲区进行更新。因此您拥有必要的时间来实现一个将最后结果移动到结果缓冲区的扫描序列。



寄存器

状态寄存器

ADC_SAR_SEQ_STATUS_REG

位	7	6	5	4	3	2	1	0
值	未使用							EOC

- EOC — 结束转换。当所有通道的一个转换周期完成时，可设置该寄存器。

资源

ADC_SAR_Seq 使用安装在硅片内固定模块 SAR 上相应的 SAR ADC 组件。

配置	资源类型					
	数据路径单元	宏单元	状态单元	Control/ Count7单元	DMA通道	中断
默认值	-	45	1	2	2	1

API 的内存使用量

该组件根据设置的不同，内存使用量也显著不同，它取决于编译器、设备所使用的 API 数量以及组件配置。该表说明了在默认组件配置中可用的所有 API 的内存使用量。

通过使用释放模式中的相应编译器来完成测量操作。在该模式中，存储器大小得到优化。对于特定设计，可以分析编译器生成的映射文件，以确定存储器的内存使用量。

配置	PSoC 5LP（GCC）	
	闪存字节	SRAM（字节）
默认值	1468	38

直流和交流电的电气特性

这些值表示了预计性能，它们基于初始特性数据。除非另有指定，否则运行条件均为：

- Fclk = 1-18 MHz

  注意：总线时钟频率至少是组件时钟频率的两倍时，才能保证得到所需采样率。

- 输入范围 = $\pm V_{REF}$
- 10 µF 的旁路电容

更多有关信息，请参考 ADC SAR 组件数据手册中的相关内容。

组件勘误表

本节列出了组件的已知问题。

赛普拉斯ID	组件版本	问题	解决方案
191257	v1.10	在没有修正PSoC Creator 3.0 SP1中的版本编号时进行更改这组件。更多信息，请参见基础知识文章KBA94159（网页地址：www.cypress.com/go/kba94159）。	解决方案是不必要的。设计不受任何影响。

组件更改

本节列出了组件与先前版本相比的主要更改的内容。

版本	更新内容	更改原因/影响
1.10.a	编辑数据手册并将其添加到组件勘误章节。	文档的组件被更改，但设计不受任何影响。
1.10	更新了校准增益，以得到更好的分辨率。 添加了新的ADC_SAR_Seq_SetScaledGain() API函数。该函数将设置增益系数具有大于12位分辨率的输入范围。	根据ADC SAR组件的变化 （增益系数(ADC_countsPerVolt)在10位分辨率的Vssa 至Vdda输入范围下）来添加新的API函数。这样会影响了ADC_SAR_Seq_CountsTo_Volts()、ADC_SAR_Seq_CountsTo_mVolts()和ADC_SAR_Seq_CountsTo_uVolts() API。
1.0	首次组件发行	

©赛普拉斯半导体公司，2013 - 2014。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路以外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不会根据专利权或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订了明确的书面协议，否则赛普拉斯产品不保证产品能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

PSoC®是赛普拉斯半导体公司的注册商标，PSoC Creator™和 Programmable System-on-Chip™是赛普拉斯半导体公司的商标。该处引用的所有其它商标或注册商标归其各自所有者所有。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途之外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于合理预计可能发生运转异常和故障，并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用于赛普拉斯软件许可协议的限制。