

10-Bit SAR ADC 数据表 SAR10 V 1.0

Copyright © 2009-2010 Cypress Semiconductor Corporation. All Rights Reserved.

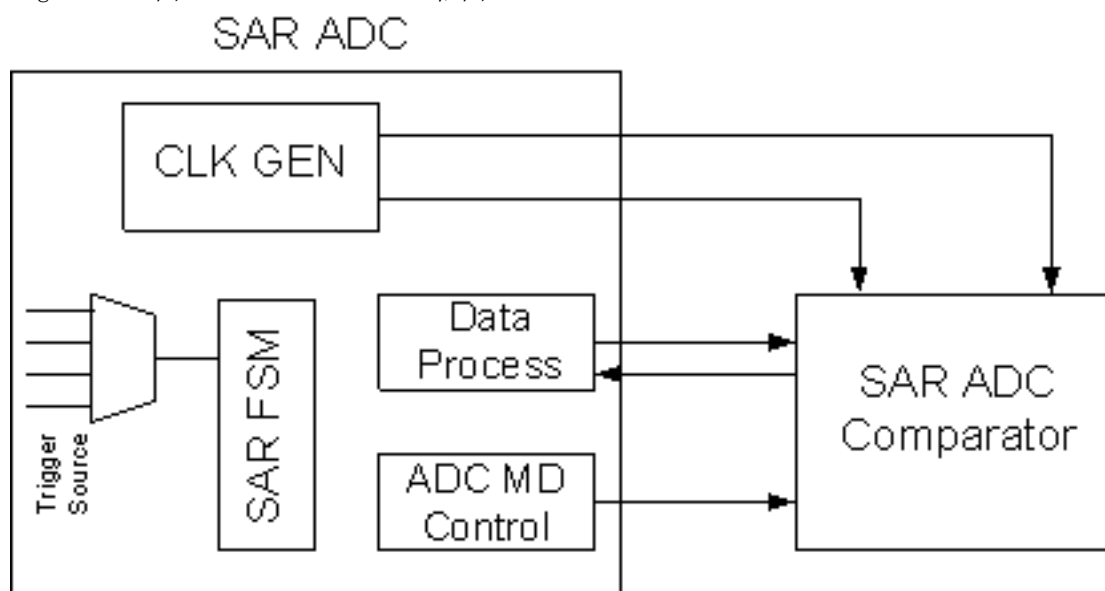
资源	PSoC® 模块		API 存储器（字节）		引脚（每个外部 I/O）
	数字	SAR	闪存	RAM	
CY8C21x45, CY8C22x45, CY8C28x45, CY8C28x43, CY8C28x13, CY8C28x03, CY8C28x52					
SAR10	0	1	240	0	0

功能和概述

- CY8C21x45、CY8C22x45 和 CY8C28x45 器件上的最佳模数转换结果。
- 10-bit 分辨率
- 单触发转换
- 自由运行转换
- 可选的转换触发器
- 可编程时钟分频器
- 转换完成后立即自动进入低功耗模式

SAR10 用户模块是一个 10-bit 逐次逼近寄存器（Successive Approximation Register, SAR）ADC 转换器，使用 SAR 模块将输入电压转换为数字代码。它为每次采样生成一个 10-bit 无符号值。此用户模块支持以下三种模数转换模式：软件触发、硬件触发和自由运行。

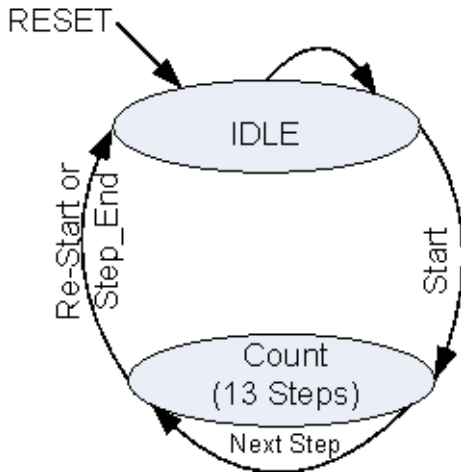
Figure 1. 图 1. 10-Bit SAR ADC 框图



功能说明

SAR10 用户模块每次转换时需要 12 个 ADC 时钟周期。前两个时钟周期对模拟输入信号进行采样。其他十个时钟周期用来进行数据转换。在转换过程中，与参考时钟相对比，某个选定 ADC 时钟周期的持续时间会拉长到其正常状态下的两倍。因此，从 ADC 有限状态机的角度看，转换需要 13 个时钟周期。有限状态机至少需要一个额外的 SYSCLK 计时单元以满足“空闲”状态的要求。每次转换必须以“空闲”开始，并返回“空闲”状态。

Figure 2. 图 2. 10-bit SAR ADC 状态机



SAR10 用户模块拥有三种模数转换模式：

- **软件触发模式：** 每次向 SAR_CR0_REG 寄存器（如果 SAR 已启用）中的“开始”位写入 1 时，都会触发一次新的转换。任何未完成的转换将中断，并立即开始新的转换。转换完成后状态机将返回“空闲”状态。
- **自由运行模式：** 除非您禁用 ADC，否则转换将反复运行。在向 SAR_CR0_REG 寄存器中的“开始”位写入 1 时，软件触发仍然可用，并会开始一次新转换。
- **硬件触发模式：** 也称为自动触发模式或自动调整模式。从四个硬件触发源中选择一个，并使用该硬件源触发转换的“开始”位。除使用的触发源不同外，该触发与软件触发模式非常相似。

如果启用自动触发模式，ADC 将在该模式下运行。如果禁用自动触发模式，并设置了自由运行位，那么 ADC 将连续运行。如果将自动触发模式和自由运行位全部禁用，ADC 将以单触发模式运行。在单触发模式中，ADC 在每次向 SAR_CR0_REG 寄存器的“开始”位写入 1 时运行一次。

SAR10 采样率基于下表：

Table 1. ADC 采样率与时钟选择

SYSCLK (IM0)	最快		最慢	
	时钟设置	实际 SPS	时钟设置	实际 SPS
24 MHz	SYSCLK/12	152.8 KSPS	SYSCLK/64	28.8 KSPS

直流和交流电气特性

其他电气特性，请参阅您的 PSoC 器件数据表。

交流电气特性

下表列出了在以下电压和温度范围内许可的最大和最小规范：分别为 4.75V 到 5.25V，且 $T_A = -40^{\circ}\text{C}$ 到 $+85^{\circ}\text{C}$ ；或 3.0V 到 3.6V，且 $T_A = -40^{\circ}\text{C}$ 到 $+85^{\circ}\text{C}$ 。典型参数适用于 25°C 且电压为 5V 和 3.3V 的情况，仅供设计指导之用。

Table 2. 交流 SAR10 ADC 规范

符号	说明	最小值	典型值	最大值	单位	注
F_{INSAR10}	SAR10 ADC 的输入时钟频率	–	–	2.0	MHz	
F_{SSAR10}	SAR10 ADC 的采样率 SAR10 ADC 分辨率 = 10 位	–	–	152.8	ksps	采样期间包含 13 个 ADC 输入时钟，并至少需要一个额外的 SysClk 计时单元。

直流规范

下表列出了在以下电压和温度范围内许可的最大和最小规范：分别为 4.75V 到 5.25V，且 $T_A = -40^{\circ}\text{C}$ 到 $+85^{\circ}\text{C}$ ；或 3.0V 到 3.6V，且 $T_A = -40^{\circ}\text{C}$ 到 $+85^{\circ}\text{C}$ 。典型参数适用于 25°C 且电压为 5V 和 3.3V 的情况，仅供设计指导之用。

Table 3. 直流 SAR10 ADC 规范

符号	说明	最小值	典型值	最大值	单位	注
$\text{INL}_{\text{SAR10}}$	积分非线性度	–2.5	–	2.5	LSB	10-bit 分辨率
$\text{DNL}_{\text{SAR10}}$	微分非线性度	–1.5	–	1.5	LSB	10-bit 分辨率
$I_{\text{VREFSAR10}}$	在配置为 SAR10 ADC 的 VREF 输入时，P22[5] 的输入电流。	–	–	0.5	mA	在此配置下，会禁用内部电压参考缓冲区。
$V_{\text{VREFSAR10}}$	在配置为 SAR10 ADC 的外部电压参考时，P22[5] 的输入参考电压。	3.0	–	4.95	V	当 VREF 在 SAR10 ADC 内部缓冲时，P22[5] 的电压电平（当配置为外部参考电压时）必须至少比 Vdd 引脚的芯片供电电压电平小 300 mV。 ($V_{\text{VREFSAR10}} < (V_{\text{dd}} - 300 \text{ mV})$)。

放置

SAR10 用户模块会占用 SAR10 模块。单个配置中不能放置多个 SAR10 用户模块。

参数和资源

ADC 时钟

“ADC 时钟 ” 参数设置 SAR 模块的当前时钟源。此参数包含以下选择：

ADC 时钟	说明
二分频	ADC 时钟由系统时钟二分频得来。
四分频	ADC 时钟由系统时钟四分频得来。
六分频	ADC 时钟由系统时钟六分频得来。
八分频	ADC 时钟由系统时钟八分频得来。
DivideBy12	ADC 时钟由系统时钟十二分频得来。
DivideBy16	ADC 时钟由系统时钟十六分频得来。
DivideBy32	ADC 时钟由系统时钟三十二分频得来。
DivideBy64	ADC 时钟由系统时钟六十四分频得来。

运行模式

“ 运行模式 ” 参数确定用户模块在自由运行模式还是单触发模式下运行。此参数包含以下选择：

运行模式	说明
单触发	仅执行一次模数转换。
自由运行	反复执行模数转换。

输入

“ADC 输入通道选择 ” 参数确定当前输入源。此参数包含以下选择：

ADC 输入通道选择	说明
Port_0_0	端口 0 的引脚 0 用作 SAR10 模块的输入源。
Port_0_1	端口 0 的引脚 1 用作 SAR10 模块的输入源。
Port_0_2	端口 0 的引脚 2 用作 SAR10 模块的输入源。
Port_0_3	端口 0 的引脚 3 用作 SAR10 模块的输入源。
Port_0_4	端口 0 的引脚 4 用作 SAR10 模块的输入源。
Port_0_5	端口 0 的引脚 5 用作 SAR10 模块的输入源。

ADC 输入通道选择	说明
Port_0_6	端口 0 的引脚 6 用作 SAR10 模块的输入源。
Port_0_7	端口 0 的引脚 7 用作 SAR10 模块的输入源。
AnalogMuxBus_0	AnalogMuxBus_0 用作 SAR10 模块的输入源。
AnalogMuxBus_1	AnalogMuxBus_1 用作 SAR10 模块的输入源。

自动触发全局启用

“ 自动触发全局启用 ” 参数启用硬件触发模式。此参数包含以下选择：

自动触发全局启用	说明
禁用	模数转换由软件触发驱动。
启用	模数转换由外部模块触发信号驱动。

选择自动触发源

“ 选择自动触发源 ” 设定当前自动触发源。此参数包含以下选择：

选择自动触发源	说明
TGL	将低 8 位数字路径用作自动触发源。
TGH	将高 8 位数字路径用作自动触发源。
TG16Bit	将高 / 低 8 位数字组合路径用作自动触发源。
TGINCMP	将 GIE 或内部比较器用作自动触发源

分辨率

分辨率设定当前分辨率模式。此参数包含以下选择：

分辨率	说明
8 位	设定为左对齐模式。返回 8 位结果。
10 位	设定为右对齐模式。返回 10 位结果。

Note * 此参数仅可用于 CY8C28x45 器件。

应用程序编程接口

应用程序编程接口 (API) 函数作为用户模块的一部分提供，从而能够采用更高级的方式处理模块。本节指定每个函数的接口，以及内置文件所提供的相关常量。

每次放置用户模块时，都会为其分配一个实例名称。默认情况下，PSoC Designer 会为指定项目中此用户模块的第一个实例分配 SAR10_1。可将该值更改为符合标识符语法规则的任意唯一值。分配的实例名称成为每个全局函数名称、变量和常量符号的前缀。为简便起见，在以下说明中将实例名称缩写为 SAR10。

Note ** 在这里，如同所有用户模块 API 中的一样，A 和 X 寄存器的值可以通过调用 API 函数来更改。如果在调用后需要 A 和 X 的值，则调用函数负责在调用前保留 A 和 X 的值。此“寄存器易失”策略是针对提高效率的目的选择，自 PSoC Designer 1.0 版起已强制使用此策略。C 编译器自动遵循此要求。汇编语言编程人员也必须确保他们的代码符合此策略。虽然一些用户模块 API 函数可以保留 A 和 X 不变，但是无法保证它们将来也会如此。

SAR10_Start

说明：

启用 SAR10 模块，并开始模数转换。

C 原型：

```
void SAR10_Start(void);
```

汇编：

```
lcall SAR10_Start
```

参数：

无

返回值：

无

副作用：

请参阅 API 一节开头的注意 **。

SAR10_Stop

说明：

禁用 SAR10 模块，并停止模数转换。

C 原型：

```
void SAR10_Stop(void);
```

汇编：

```
lcall SAR10_Stop
```

参数：

无

返回值：

无

副作用：

请参阅 API 一节开头的注意 **。

SAR10_EnableInt

说明:

启用 SAR10 模块中断。

C 原型:

```
void SAR10_EnableInt(void);
```

汇编:

```
lcall SAR10_EnableInt
```

参数:

无

返回值:

无

副作用:

请参阅 API 一节开头的注意 **。

SAR10_DisableInt

说明:

禁用 SAR10 模块中断。

C 原型:

```
void SAR10_DisableInt(void);
```

汇编:

```
lcall SAR10_DisableInt
```

参数:

无

返回值:

无

副作用:

请参阅 API 一节开头的注意 **。

SAR10_Trigger

说明:

触发 SAR10 以转换一个实例。如果 SAR10 用户模块在自动触发模式下运行，此功能将不会影响到 ADC 采样工作。

C 原型:

```
void SAR10_Trigger(void);
```

汇编:

```
lcall SAR10_Trigger
```

参数:

无

返回值:

无

副作用:

请参阅 API 一节开头的注意 **。

SAR10_fIsDataAvailable**说明:**

检查采样数据的可用性。

C 原型:

```
BYTE SAR10_fIsDataAvailable(void);
```

汇编:

```
lcall SAR10_fIsDataAvailable
```

参数:

无

返回值:

如果数据已经转换或可以读取则返回非零值。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_iGetData**说明:**

返回最后转换的数据。在获取该数据前，应调用 SAR10_fIsDataAvailable() 以确保数据有效。

C 原型:

```
INT SAR10_iGetData(void);
```

汇编:

```
lcall SAR10_iGetData
```

参数:

无

返回值:

返回转换结果。在汇编程序中，LSB 在 X 中返回，MSB 在累加器中返回。

副作用:

请参阅 API 一节开头的注意 **。

此功能仅以右对齐模式在 CY8C28x45 器件上返回正确结果。

SAR10_bGetData

说明:

返回最后转换的数据的最高 8 位。在获取该数据前，应调用 SAR10_fIsDataAvailable() 以确保数据有效。

C 原型:

```
BYTE SAR10_bGetData(void);
```

汇编:

```
lcall SAR10_bGetData
```

参数:

无

返回值:

返回转换结果。在汇编程序中，结果位于累加器中。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_SetADCChannel

说明:

选择 SAR10 输入源。

C 原型:

```
void SAR10_SetADCChannel(BYTE bChannel);
```

汇编:

```
mov A,bChannel
lcall SAR10_SetADCChannel
```

参数:

bChannel 为输入源。符号名以 C 和汇编语言形式提供。相关值在以下表格中给出:

符号名	值	说明
SAR10_CHS_P00	0x00	Port_0_0 是 SAR10 模块的输入源。
SAR10_CHS_P01	0x08	Port_0_1 是 SAR10 模块的输入源。
SAR10_CHS_P02	0x10	Port_0_2 是 SAR10 模块的输入源。
SAR10_CHS_P03	0x18	Port_0_3 是 SAR10 模块的输入源。
SAR10_CHS_P04	0x20	Port_0_4 是 SAR10 模块的输入源。
SAR10_CHS_P05	0x28	Port_0_5 是 SAR10 模块的输入源。

符号名	值	说明
SAR10_CHS_P06	0x30	Port_0_6 是 SAR10 模块的输入源。
SAR10_CHS_P07	0x38	Port_0_7 是 SAR10 模块的输入源。
SAR10_CHS_AMUX0	0x60	AnalogMuxBus_0 是 SAR10 模块的输入源。
SAR10_CHS_AMUX1	0x68	AnalogMuxBus_1 是 SAR10 模块的输入源。

返回值:

无。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_SetTriggerSrc

说明:

选择 SAR10 自动触发源。

C 原型:

```
void SAR10_SetTriggerSrc(BYTE bSrc);
```

汇编:

```
mov A,bSrc
lcall SAR10_SetTriggerSrc
```

参数:

bSrc: 是触发源。符号名以 C 和汇编语言形式提供。相关值在以下表格中给出:

符号名	值	说明
SAR10_SRC_TGRL	0x00	低 8 位数字路径是自动触发源。
SAR10_SRC_TGRH	0x10	高 8 位数字路径是自动触发源。
SAR10_SRC_TGR16	0x20	高 / 低 8 位数字组合路径是自动触发源。
SAR10_SRC_TGRINCP	0x30	GIE 或内部比较器是自动触发源。

返回值:

无。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_EnableAutoTrigger

说明:

SAR10 自动触发功能的全局启用控制。

C 原型:

```
void SAR10_EnableAutoTrigger(BYTE bMode);
```

汇编:

```
mov A,bMode
lcall SAR10_EnableAutoTrigger
```

参数:

bMode 启用或禁用自动触发模式。符号名以 C 和汇编语言形式提供。相关值在以下表格中给出:

符号名	值	说明
SAR10_AUTOTGR_ENABLE	0x01	启用自动触发模式。
SAR10_AUTOTGR_DISABLE	0x00	禁用自动触发模式。

返回值:

无。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_SetClk

说明:

设置 ADC 采样率与时钟选择。将系统时钟设置为 24 MHz 时, SAR10_SYSCLK_2 设置所产生的采样率对于可靠操作来说过快。请勿在系统时钟为 24 MHz 时使用 SAR10_SYSCLK_2。

C 原型:

```
void SAR10_SetClk(BYTE bClkMode);
```

汇编:

```
mov A,bClkMode
lcall SAR10_SetClk
```

参数:

bClkMode 是时钟源。符号名以 C 和汇编语言形式提供。相关值在以下表格中给出:

符号名	值	说明
SAR10_SYSCLK_2	0x00	系统时钟二分频。
SAR10_SYSCLK_4	0x02	系统时钟四分频。
SAR10_SYSCLK_6	0x04	系统时钟六分频。
SAR10_SYSCLK_8	0x06	系统时钟八分频。

符号名	值	说明
SAR10_SYSCLK_12	0x08	系统时钟十二分频。
SAR10_SYSCLK_16	0x0A	系统时钟十六分频。
SAR10_SYSCLK_32	0x0C	系统时钟三十二分频。
SAR10_SYSCLK_64	0x0E	系统时钟六十四分频。

返回值:

无。

副作用:

请参阅 API 一节开头的注意 **。

SAR10_SetRunMode

说明:

将 ADC 设置为自由运行或单触发模式。

C 原型:

```
void SAR10_SetRunMode (BYTE bRunMode);
```

汇编:

```
mov A,bRunMode
lcall SAR10_SetRunMode
```

参数:

bRunMode 是运行模式。符号名以 C 和汇编语言形式提供。相关值在以下表格中给出:

符号名	值	说明
SAR10_ONESHOT	0x00	单触发
SAR10_FREERUN	0x08	自由运行

返回值:

无。

副作用:

请参阅 API 一节开头的注意 **

固件源代码示例

此处所示 C 代码向您说明如何使用 SAR10 用户模块。

```
#include <m8c.h>           // part specific constants and macros
#include "PSoC_API.h"      // PSoC API definitions for all User Modules

int iResult;
void main(void)
{
    SAR10_SetClk(SAR10_SYSCLK_64); // Set clock source - system clock/64
    SAR10_SetRunMode(SAR10_ONESHOT); // Set running method - one-shot
    SAR10_SetADCChannel(SAR10_CHS_P05); // Set Port_0_5 as input
    SAR10_EnableInt(); // Enable SAR10 interrupt
    SAR10_Start(); // Start conversion

    M8C_EnableGInt; // Enable global interrupt

    while(1)
    {
        SAR10_Trigger(); //Trigger new sample
        while(SAR10_fIsDataAvailable()==0); //Wait while data is not ready
        iResult = SAR10_iGetData(); // Read result
    }
}
```

同一代码用汇编语言表示为:

```
include "m8c.inc"          ; part specific constants and macros
include "memory.inc"       ; Constants & macros for SMM/LMM and Compiler
include "PSoC_API.inc"     ; PSoC API definitions for all User Modules

export _main
area bss (RAM, REL)
_iResult:
iResult: BLK    2
area text (ROM, REL)
_main:
mov A, SAR10_SYSCLK_64
call SAR10_SetClk ;Set clock source - system clock/64
mov A, SAR10_ONESHOT
call SAR10_SetRunMode ;Set running method - one-shot
mov A, SAR10_CHS_P05
call SAR10_SetADCChannel ;Set Port_0_5 as input
call SAR10_EnableInt ;Enable SAR10 interrupt
call SAR10_Start ;Start conversion
M8C_EnableGInt ;Enable global interrupt
.ReadADCData:
call SAR10_Trigger ;Trigger new sample
.Wait:
call SAR10_fIsDataAvailable
cmp A, 0
jz .Wait ;Wait while data is not ready
call SAR10_iGetData ;Read result
mov [_iResult+1], X ;Get MSB of result
```

```
mov [_iResult], A ;Get LSB of result
jmp .ReadADCData
```

配置寄存器

Table 4. SAR_CR0_REG

位	7	6	5	4	3	2	1	0
值	0	输入选择				就绪	开始	启用

此寄存器用于调整 SAR10 ADC。您可以对输入进行设置，开始 ADC 采样，以及启用 SAR10 模块。SAR_CR0_REG 同时还代表已转换数据的就绪状况。

“启用”位通过调用 SAR10_Start 或 SAR10_Stop API 例程进行更改。

“开始”位通过调用 SAR10_Trigger API 例程进行修改，并开始单独的 ADC 转换。

“就绪”位通过调用 SAR10_fIsDataAvailable API 例程进行读取，并确定已转换数据的就绪状况。

“输入选择”位确定输入源。这些位值取决于在器件编辑器中选择了哪个“ADC 输入通道选择”参数。该值可以在运行时通过调用 SAR10_SetADCChannel API 进行更改。

Table 5. SAR_CR1_REG

位	7	6	5	4	3	2	1	0
值	0	0	触发源		时钟源			自动触发

此寄存器用于调整 SAR10 ADC。您可以用它来设置时钟和触发源，以便设置自动触发模式。

“触发源”位确定外部触发信号源。这些位值取决于在器件编辑器中选择了哪个“选择自动触发源”参数。该值也可以在运行时通过调用 SAR10_SetTriggerSrc API 进行更改。

“时钟源”位确定时钟源。这些位值取决于在器件编辑器中选择了哪个“ADC 输入通道选择”参数。该值也可以在运行时通过调用 SAR10_SetC1k API 进行更改。

“自动触发”位确定自动触发读取模式。此位值取决于在器件编辑器中选择了哪个“自动触发全局启用”参数。该值也可以在运行时通过调用 SAR10_EnableAutoTrigger API 进行更改。

Table 6. SAR_CR2_REG

位	7	6	5	4	3	2	1	0
值	0	0	0	0	自由运行	0	0	0

此寄存器用于调整 SAR10 ADC，尤其可用于设置重复运行模式。

“自由运行”位确定自由运行读取模式。此位值取决于在器件编辑器中的用户模块参数下选择了哪个“运行模式”参数。该值也可以通过调用 SAR10_SetRunMode API 进行更改。

Table 7. SAR_DH_REG

位	7	6	5	4	3	2	1	0
值	高数据							

此寄存器包含 CY8C21x45、CY8C22x45 器件和 CY8C28x45 器件的 ADC 转换数据的最高 8 位，数据为左对齐模式。

此寄存器包含 CY8C28x45 器件的 ADC 转换数据的最高 2 位，数据为右对齐模式。

“高数据”位通过调用 SAR10_iGetData. 读取

Table 8. SAR_DL_REG

位	7	6	5	4	3	2	1	0
值							低数据	

此寄存器包含 CY8C21x45、CY8C22x45 和 CY8C28x45 器件的 ADC 转换数据的 2 个最低有效位，数据为左对齐模式。

此寄存器包含 CY8C28x45 器件的 ADC 转换数据的 8 个最高有效位，数据为右对齐模式。

“低数据”位通过调用 SAR10_iGetData. 读取

Table 9. SAR_CR3_REG

位	7	6	5	4	3	2	1	0
值	LALIGN							

版本历史记录

版本	创作者	说明
1.0	DHA	初始版本

Note PSoC Designer 5.1 在所有的用户模块数据表中提供版本历史记录。本数据表详细介绍了当前和先前用户模块版本之间的区别。