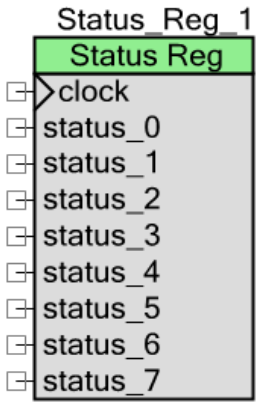
# Status Register（状态寄存器）

1.80

## 特性

- 最高达 8 位的状态寄存器
- 中断支持




## 概述

状态寄存器可让固件读取数字信号。

### 何时使用状态寄存器

当固件需要查询内部数字信号的状态时，使用状态寄存器。

## 输入/输出连接

本节描述状态寄存器的输入连接。I/O 列表中的星号 (*) 表示，在 I/O 说明列出的情况下，该 I/O 可能不可见。

### 时钟 — 输入

状态寄存器时钟。配置为透明位的对应时钟输入信号被忽略。

### status_0 - status_7 – Input *

状态寄存器输入。固件通过读取状态寄存器的方式查询输入信号。输入的数量取决于 **Inputs**（输入）参数。这些输入可能被保留为悬空，无外部连接。如果这些线路无任何连接，则组件将分配常量逻辑 0。

### status[N:0] – 输入*

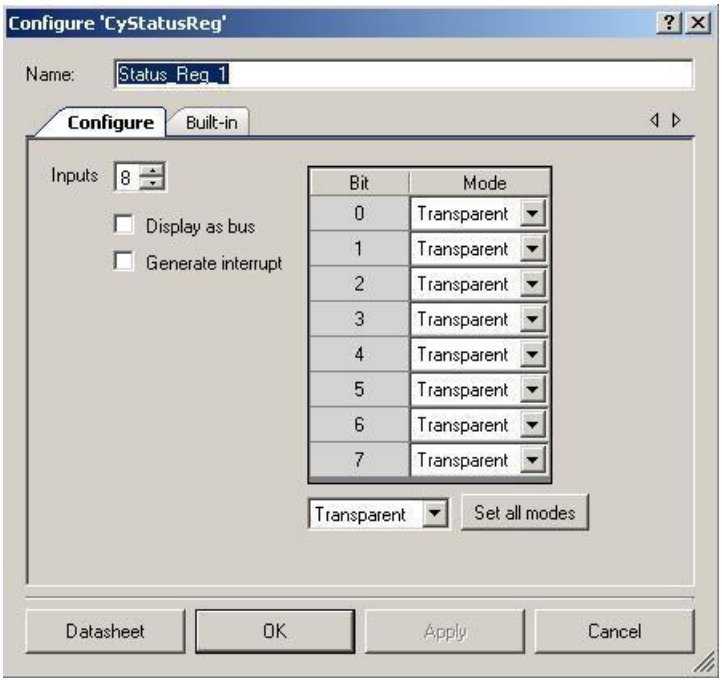
该可选输入可将各个单独的输入端并入一个单总线端。使能 **Display as bus**（显示为总线）参数后，该引脚可见。N 为输入数量 - 1。该输入可悬空，无外部连接。如果该线路无任何连接，组件将分配一个常量逻辑 0。

### intr – 输出*

使能 **Generate interrupt**（产生中断）参数后，符号上会显示该可选引脚。该选项仅在所选的输入数量少于 8 个时有效。

## 组件参数

拖拉状态寄存器到你的设计，双击状态寄存器以打开 **Configure**（配置）对话框。




### 输入

输入端的数量 (1-8)。默认值为 **8**。

### 显示为总线

该参数将输入显示为总线，而非各个单独的端。该选项在默认情况下为未选中。

## 产生中断

该参数会在符号上显示中断输入。该选项在默认情况下为未选中。中断仅在输入数量少于 8 时有效。

## 设置所有模式

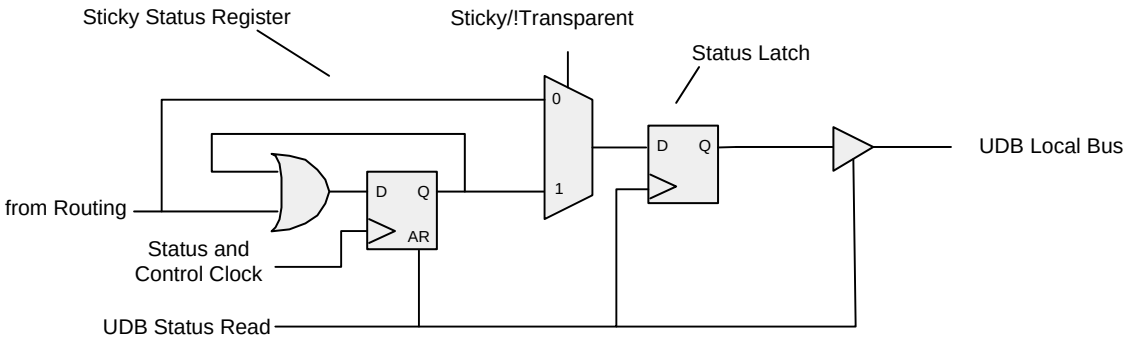
该按钮会将所有位设置为 **Transparent**（透明）或者 **Sticky**（粘滞）模式，具体情况取决于该按钮左手边组合框中所选的模式。

## 模式

这些参数用于设置状态寄存器的特定位，以在寄存后保持高电平，直到执行读取时为止。该读取清除所有寄存值。这些设置是：

- **透明 –** 默认情况下，CPU 读写寄存器时透明地读取相关布线网络(routing net)的状态，它与模块时钟是异步的。此模式可用于 UDB 内部计算和寄存 的瞬变状态。
- **粘滞（清除读取） –** 此模式下，在状态和控制时钟的每个周期上进行相关布线网络的采样。如果在给定采样中信号为高电平，则被捕获到状态位，并保持高电平，而不考虑相关布线的后续状态。当 CPU 固件读取状态寄存器时，该位被清除。状态寄存器清除不依赖于任何模式，即使在模块时钟被禁用时也同样发生；它基于总线时钟，并作为读取操作的一部分而发生。

**图 1. 透明与粘滞模式的行为特性**




## 中断掩码

该选项仅在选中 **Generate interrupt**（产生中断）时显示在配置对话框中。使用这些参数，可为状态寄存器中的每个位设置中断掩码值。默认情况下，中断掩码值为 0。

# 应用程序编程接口

应用程序编程接口 (API) 子程序允许你使用软件配置组件。

默认情况下，PSoC Creator 将实例名称“Status_Reg_1”分配给设计中状态寄存器的第一个实例。你可以将组件重命名为任何唯一值，只要遵循标识符的语法规则。实例名称会成为每个全局函数名称、变量和常量符号的前缀。为了便于阅读，下列函数中使用的实例名称为“StatusReg”。

| 函数 | 说明 |
|---|---|
| StatusReg_Read() | 读取状态寄存器的当前值 |
| StatusReg_InterruptEnable() | 使能状态寄存器中断 |
| StatusReg_InterruptDisable() | 禁用状态寄存器中断 |
| StatusReg_WriteMask() | 写入的值分配给掩码寄存器 |
| StatusReg_ReadMask() | 从掩码寄存器返回当前中断掩码值 |

## uint8 StatusReg_Read (void)

**说明：** 读取状态寄存器的值。

**参数：** 无

**返回值：** 返回状态寄存器的当前值。

**副作用：** 无

## void StatusReg_InterruptEnable (void)

**说明：** 使能状态寄存器中断。默认行为特性是禁用。其仅在如果状态寄存器产生中断时有效。

**参数：** 无

**返回值：** 无

**副作用：** 无

## void StatusReg_InterruptDisable (void)

**说明：** 禁用状态寄存器中断。其仅在如果状态寄存器产生中断时有效。

**参数：** 无

**返回值：** 无

**副作用：** 无

### void StatusReg_WriteMask (uint8 mask)

**说明：** 写当前掩码值分配给状态寄存器。其仅在如果状态寄存器产生中断时有效。

**参数：** 掩码：写入掩码寄存器的值。

**返回值：** 无

**副作用：** 无

### uint8 StatusReg_ReadMask (void)

**说明：** 读分配给状态寄存器的当前中断掩码值。其仅在如果状态寄存器产生中断时有效。

**参数：** 无

**返回值：** 返回中断掩码的当前值。

**副作用：** 无

## DMA

可以使用 DMA 组件从状态寄存器直接读取数据。可以使用 DMA 向导按如下所示配置 DMA 操作

| DMA 向导中 DMA 源/目的地的名称 | 方向 | DMA 请求信号 | DMA 请求类型 | 说明 |
|---|---|---|---|---|
| StatusReg_Status_PTR | 源 | 无 | 无 | 存储状态寄存器值。 |

## MISRA 符合性(compliance)

本节介绍了本组件与 MISRA-C:2004 的符合性和偏离(deviation)。定义了两种类型的偏离：

- 项目偏离 - 适用于所有 PSoC Creator 组件的偏离
- 特定偏离 - 仅适用于此组件的偏离

本节提供了有关组件特定偏离的信息。*系统参考指南*的“MISRA 符合性”章节中介绍项目偏离以及有关 MISRA 符合性验证环境的信息。

状态寄存器组件没有任何特定偏离。

# 固件源代码示例

PSoC Creator 在“Find Example Project (查找示例项目)”对话框中提供了大量包括原理图和代码的实例项目。对于组件的具体示例，请打开 Component Catalog (组件目录)中的对话框或原理图中的组件实例。对于通用的示例，请打开 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（筛选选项）可缩小可选项目的列表。

有关更多信息，请参见 PSoC Creator 帮助中的“Find Example Project（查找示例项目）”主题。

# 资源

状态寄存器组件使用 UDB 阵列的一个状态单元。

# API 存储器使用

根据编译器、器件(device)、所用 API 数量和组件配置的不同，组件内存使用会出现显著变化。下表提供给定组件配置所有可用 API 的存储器使用。

相关编译器的测量已经完成，优化配置了大小的设置。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使用。

| 配置 | PSoC 3 (Keil_PK51) | | PSoC 4 (GCC) | | PSoC 5LP (GCC) | |
|---|---|---|---|---|---|---|
| | 闪存（字节） | RAM（字节） | 闪存（字节） | RAM（字节） | 闪存（字节） | RAM（字节） |
| 默认值 | 46 | 0 | 92 | 0 | 92 | 0 |

# 组件更改

本节列出了以前版本相比的组件主要更改。

| 版本 | 更改说明 | 更改原因/影响 |
|---|---|---|
| 1.80a | 更新了PSoC 4 内存使用的数据手册。 | |
| 1.80 | 增加 MISRA 符合性章节。 | 此组件没有任何特定偏离。 |
| 1.70 | 增加了 PSoC 5LP 支持。 | |
| | 实现了 StatusReg_InterruptEnable()、StatusReg_InterruptDisable()、StatusReg_WriteMask() 和 StatusReg_ReadMask() API。 | 支持中断功能。 |
| | 更新了“Configure（配置）”对话框。 | 增加了 Display as bus（显示为总线）、Generate interrupt（产生中断）、Set all modes（设置所有模式）和 Interrupt mask parameters（中断掩码参数），以及次要设计更改。 |
| | 为输入端增加了中断引脚和显示为总线选项。还实现了 DMA 功能和调试窗口支持。 | 将输入端作为总线，并支持中断产生。 |
| 1.60 | 更新了 Configure（配置）对话框。 | 更改了位显示，解决了次要的 Configure（配置）对话框问题 |
| 1.50.b | 数据手册编辑 | |
| 1.50.a | 数据手册编辑 | |
| 1.50 | 更新了“Configure（配置）”对话框。 | 创建了定制接口。增加“Set All”（“设置所有项”）按钮，并更改了输入数字栏以允许键盘输入。更新对话框，使其符合公司标准。 |

© 赛普拉斯半导体公司，2010-2013。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，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。也不根据专利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明确的书面协议，否则赛普拉斯产品不保证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。此处引用的所有其他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。赛普拉斯保留在不另行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的赛普拉斯软件许可协议限制。