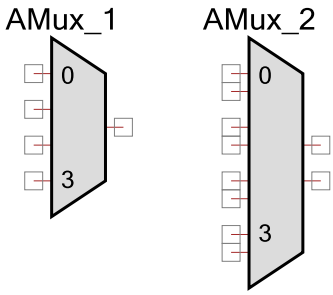
模拟复用器 (AMux)

1.50

特性



- 单端连接或差分连接
- 可在 2 到 32 种连接之间调整
- 软件可控的
- 连接可以采用引脚或内部源
- 多个同步连接
- 双向（被动）

概述

模拟复用器 (AMux) 组件可用于将一个或多个模拟信号连接到另一个通用模拟信号，也可以不连接模拟信号。同时连接多个模拟信号的能力提供了交叉交换支持，这是对传统复用器功能的扩展。

何时使用 AMux

任何时候需要将多个模拟信号复用到一个源或目标时都可使用 AMux。由于 AMux 是被动的，它可用于复用输入或输出信号。

输入/输出连接

本节介绍 AMux 的各种输入和输出连接。I/O 列表中的星号 (*) 表示，在 I/O 说明中列出的情况下，该 I/O 可能不可见。

0-31 – 模拟

AMux 能够具有 2 到 32 种模拟可交换连接。

0-32（成对）– 模拟 *

只有当 **MuxType** 参数设置为 **Differential**（差分）时，才能使用成对可交换连接。

通用 – 模拟

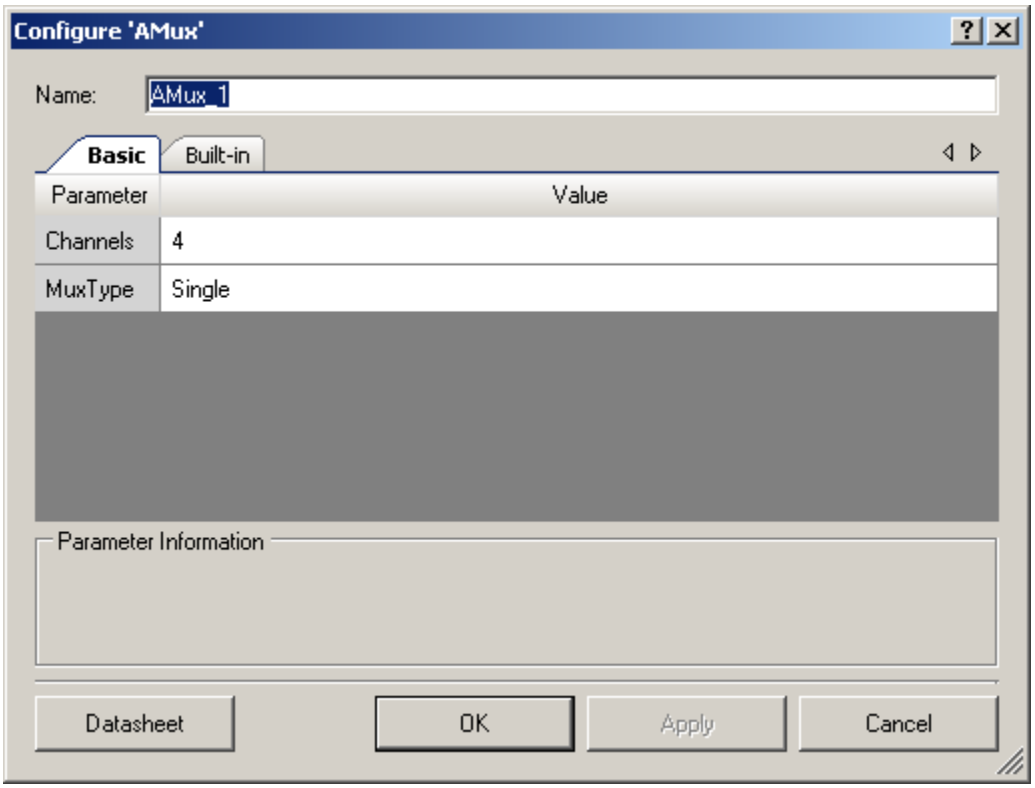
“通用”信号采用通用连接；它没有标签。使用 AMux_Select() 函数选择的通道将连接到此终端。

通用（成对）– 模拟 *

当使用差分复用器时，“通用（成对）”信号采用通用成对连接。使用 AMux_Select() 函数选择的通道将连接到此终端。

元件参数

将 AMux 组件拖动到设计中，双击它以打开 **Configure**（配置）对话框。



AMux 提供下列参数。

通道

此参数根据 **MuxType** 选择可交换连接数。介于 2 和 32 之间任意值都是有效值。

MuxType

此参数用于在**单一**可交换连接复用器和**差分**可交换连接复用器之间进行选择。当可连接信号都引用同一信号（如 V_{SSA}）时，使用**单一**。当两个或更多信号可能具有不同的信号参考时，选择 **Differential**（差分）选项。“差分”模式最常用于提供差分输入的 ADC。

资源

AMux 使用单独开关将模块和引脚连接到模拟总线。

应用程序编程接口

应用程序编程接口 (API) 子程序允许您使用软件配置组件。下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“AMux_1”分配给所提供设计中的第一个组件实例。可以将实例重命名为任何遵循标识符语法规则的唯一值。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。为增加可读性，下表中使用了实例名称“AMux”。

函数	说明
AMux_Init()	断开所有通道的连接
AMux_Start()	断开所有通道的连接
AMux_Stop()	断开所有通道的连接
AMux_Select()	断开所有通道的连接，然后连接“chan”
AMux_Connect()	连接“chan”信号，但是不断开其他通道的连接
AMux_Disconnect()	仅断开“chan”信号的连接
AMux_FastSelect()	断开 AMux_Select() 或 AMux_FastSelect() 函数选择的上一通道，然后连接新信号“chan”
AMux_DisconnectAll()	断开所有通道的连接

void AMux_Init(void)

说明： 断开所有通道的连接。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： 所有寄存器将复位为其初始值。

void AMux_Start(void)

说明： 断开所有通道的连接。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMux_Stop(void)

说明： 断开所有通道的连接。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMux_Select(uint8 chan)

说明： AMux_Select() 函数首先断开所有其他通道的连接，然后连接给定的通道。

参数： chan：要连接到通用终端的通道。

Return Value（返回值）： None（无）

Side Effects（副作用）： 使用 AMux_Select()，将断开通过 AMux_Connect() 或 AMux_FastSelect() 实现的连接。

void AMux_FastSelect(uint8 chan)

说明： 此函数首先断开 AMux_FastSelect() 或 AMux_Select() 函数进行的上一连接，然后连接给定的通道。AMux_FastSelect() 函数类似于 AMux_Select() 函数，但速度快一些，这是因为选择的仅是上一通道而不是选择所有可能的通道。

参数： chan：要连接到通用终端的通道

Return Value（返回值）： None（无）

Side Effects（副作用）： 如果在调用 AMux_FastSelect() 之前使用 AMux_Connect() 函数选择了通道，将不断开 AMux_Connect() 选择的通道。这在必须连接并行信号时非常有用。

void AMux_Connect(uint8 chan)

说明： 此函数将给定通道连接到通用信号，不影响其他连接。

参数： chan：要连接到通用终端的通道

Return Value（返回值）： None（无）

Side Effects（副作用）： 调用函数 AMux_Select() 后，会断开在将通过的通道连接到 AMux_Select() 命令之前使用 AMux_Connect() 函数连接的任何通道。

void AMux_Disconnect(uint8 chan)

说明： 仅断开指定通道与通用终端的连接。

参数： uint8 chan：要与通用终端断开连接的通道

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMux_DisconnectAll(void)

说明： 断开所有通道的连接。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

固件源代码示例

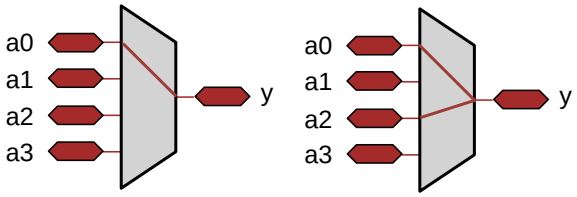
PSoC Creator 在“查找示例项目”（Find Example Project）对话框中提供了很多包括原理图和代码示例的示例项目。要获取组件特定的示例，请打开组件目录中的对话框或原理图中的组件实例。要获取通用的示例，请打开 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（滤波器选项）可缩小可选项目的列表。

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。

功能描述

AMux 与大多数硬件复用器不一样。下面是 AMux 与标准固定硬件复用器的两个不同之处。第一，它是独立开关的集合；第二，它由固件而不是硬件控制。

由于这两点不同，AMux 更加灵活，允许多个信号同时连接到通用信号。可以在任何给定时间将两个或更多信号连接到通用信号。



性能

模拟复用器由软件控制，因此切换性能取决于提供的 API 的执行时间。此性能取决于设计中复用器的精确配置。表 1 旨在提供有关切换性能的指南。

所有的性能测量都是在 CPU 频率为 48 MHz 的情况下测得的。性能测量与 CPU 频率几乎呈现线性关系。配置了编译器优化，以向与 PSoC 器件捆绑在一起的编译器提供最大的优化。对于 PSoC 3，编译器设置为针对大小处于优化级别 5 的 Keil 优化。对于 PSoC 5，编译器设置为针对大小的 GNU 优化。

表 1. 性能

函数	复用器单输入	PSoC 3 (µs)	PSoC 5 (µs)
Connect	2	2.9	1.2
	4	4.9	1.8
Disconnect	2	2.8	1.2
	4	4.9	1.8
Select	2	16.3	4.9
	4	30.5	8.5
FastSelect	2	7.2	3.0
	4	11.4	4.2

直流和交流电气特性

AMux 将在所有有效供电电压下运行。

组件更改

本节介绍组件与以前版本相比的主要更改。

版本	更改说明	更改/影响原因
1.50.c	在数据手册中添加了“性能”一节	
1.50.b	对数据表进行了少量编辑和更新	
1.50.a	对数据表进行了少量编辑和更新	
1.50	添加了 AMux_Init 函数。	为了符合公司标准，并提供 API 以便无需启动组件即可初始化或恢复组件。
1.20.a	向组件中添加了信息，以说明它与芯片修订版的兼容性。	如果组件在不兼容的芯片上使用，该工具将报告错误或发生警告。如果发生此情况，请更新到支持您的目标器件的修订版。
1.20	更新了符号图片。	更新的目的是为了符合公司标准。

© 赛普拉斯半导体公司，2010-2012。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。。

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。