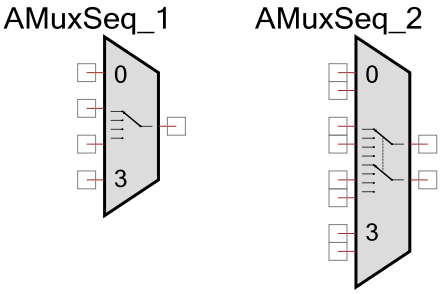
模拟定序复用器 (AMuxSeq)

1.50

特性



- 单端连接或差分连接
- 可在 2 到 32 个连接之间调整
- 由软件控制
- 连接可以来自引脚或内部源
- 无同时连接
- 双向（被动）

概述

模拟定序复用器 (AMuxSeq) 组件用于通过按连接序列顺序断开和闭合连接的方式一次将一个模拟信号连接到不同的模拟信号。AMuxSeq 主要用于时分复用。

何时使用 AMuxSeq

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 AMuxSeq 组件。由于 AMuxSeq 是被动的，它可用于复用输入或输出信号。

AMuxSeq 拥有的 API 比 AMux 的 API 要更简单、更快捷。当不需要多个同时连接且始终以同一顺序访问信号时，使用 AMuxSeq，而不是 AMux。

输入/输出连接

本节介绍 AMuxSeq 的各种输入和输出连接。I/O 列表中的星号 (*) 表示，在 I/O 说明中条件满足的情况下，该 I/O 可能不可见。

0-31 – 模拟

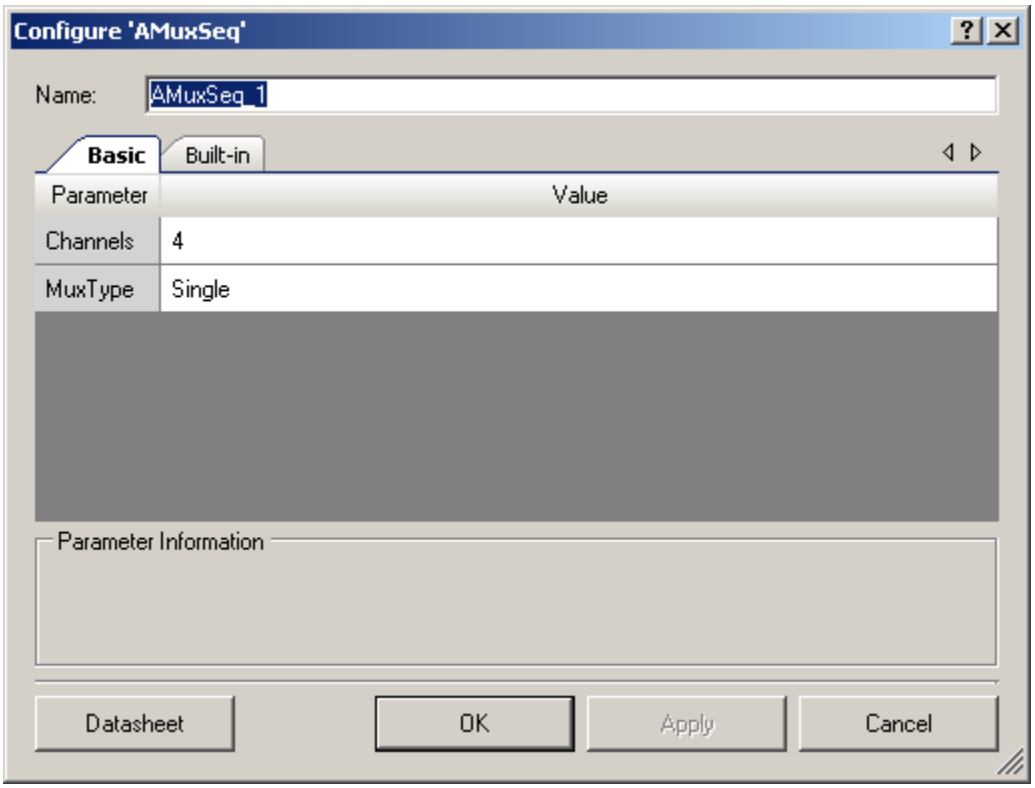
AMuxSeq 能够具有 2 到 32 种模拟可交换连接。当 **MuxType** 参数设置为 **Differential**（差分）时，显示成对连接。

common（公共端） – 模拟

“common”信号是公共端连接；它没有标注。使用 AMuxSeq_Next() 函数选择的可交换连接信号将连接到此端子。当 **MuxType** 参数设置为 **Differential**（差分）时，显示成对信号。

元件参数

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



AMuxSeq 提供下列参数。

通道

此参数用于选择输入或成对输入的个数（根据 **MuxType** ）。介于 2 和 32 之间任意值都是有效值。

MuxType

此参数用于选择单端输入（**Single**）与差分输入 **Differential**。**当输入信号都是相对于同一参考信号（如 V_{SSA}）时，使用 Single**。当两个或更多信号可能相对于不同的参考源时，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

AMuxSeq 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“AMuxSeq_1”分配给提供的设计中的第一个组件实例。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“AMuxSeq”。

函数	说明
AMuxSeq_Init()	断开所有通道的连接
AMuxSeq_Start()	断开所有通道的连接
AMuxSeq_Stop()	断开所有通道的连接
AMuxSeq_Next()	断开之前通道的连接并按照顺序连接到下一个通道。
AMuxSeq_DisconnectAll()	断开所有通道的连接
AMuxSeq_GetChannel()	返回当前连接的通道。如果没有连接通道，返回 –1。

void AMuxSeq_Init(void)

说明： 断开所有通道的连接。下次调用 AMuxSeq_Next() 时，选择第一个通道。

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 AMuxSeq_Start(void)

说明： 断开所有通道的连接。下次调用 AMuxSeq_Next() 时，选择第一个通道。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMuxSeq_Stop(void)

说明： 断开所有通道的连接。下次调用 AMuxSeq_Next() 时，选择第一个通道。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMuxSeq_Next(void)

说明： 断开之前通道的连接并按照顺序连接到下一个通道。第一次调用 AMuxSeq_Next() 时或者在 AMuxSeq_Init(), AMuxSeq_Start()、AMuxSeq_Enable()、AMuxSeq_Stop() 或 AMuxSeq_DisconnectAll() 之后调用此函数时，它将连接通道 0。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

void AMuxSeq_DisconnectAll(void)

说明： 此函数用于断开所有通道的连接。下次调用 AMuxSeq_Next() 时，选择第一个通道。

参数： None（无）

Return Value（返回值）： None（无）

Side Effects（副作用）： None（无）

int8 AMuxSeq_GetChannel(void)

说明：	返回当前连接的通道。如果没有连接通道，返回 –1。
参数：	None（无）
Return Value（返回值）：	当前通道或 –1。
Side Effects（副作用）：	None（无）

固件源代码示例

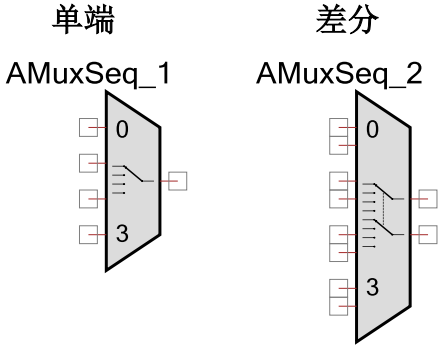
PSoC Creator 在“Find Example Project（查找示例项目）”对话框中提供了很多包括原理图和代码示例的示例项目。要获取组件特定的示例，请打开组件目录中的对话框或原理图中的组件实例。要获取通用的示例，请打开 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（滤波器选项）可缩小可选项目的列表。

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

AMuxSeq 由固件而不是硬件控制。一次只能有一个信号连接到通用信号。

下面显示了 AMuxSeq 配置为单端和差分的流程。



性能

连续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。性能取决于设计中复用器的精确配置，但是对输入的数量无影响，因为每次调用都是断开一个输入并连接另一个输入。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。编译器优化被配置为 PSoC Creator 捆绑的编译器所能提供的最高级别。对于 PSoC 3，Keil 编译器设置为优化级别 5（针对大小或速度）。对于 PSoC 5，GNU 编译器设置为针对大小或速度的优化。

表 1。性能

函数	优化	PSoC 3 (µs)	PSoC 5 (µs)
Next	Size	8.8	1.4
	Speed	2.4	1.4

直流和交流电气特性

AMuxSeq 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 AMuxSeq_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并说明排序。
	添加了 AMuxSeq_GetChanne()l API 函数。	用于获取当前连接的通道。
	为没有参数的函数添加了缺少的‘void’。 将 AMux 通道变量的类型从无符号整数更改为了有符号整数，因为 -1 用于说明没有选择通道。	这些更改消除了使用 MDK 和 RVDS 编译器进行编译期间显示的有关过期声明的警告。

© 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。