AN81623

PSoC® 3、PSoC 4 和 PSoC 5LP 数字设计最佳实践

作者： Todd Dust、Mark Ainsworth

相关芯片系列：所有 PSoC 3 和 PSoC 5LP 芯片，以及 PSoC 4 4200 系列
软件版本：PSoC Creator™ 2.2 SP1 以及更高版本
要获取完整的应用笔记列表，请单击此处。
要想获得该应用笔记的最新版本或相关项目文件，请访问网站 http://www.cypress.com/go/AN81623。

AN81623 简单介绍了数字硬件设计原理，并对 PSoC 3、PSoC 4（4200 系列）以及 PSoC 5LP 中功能强大且高度灵活的数字子系统进行了相关介绍。同时该应用笔记还介绍了使用 PSoC Creator 设计数字的最佳实践，并说明了如何使用静态时序分析（STA）报告文件。

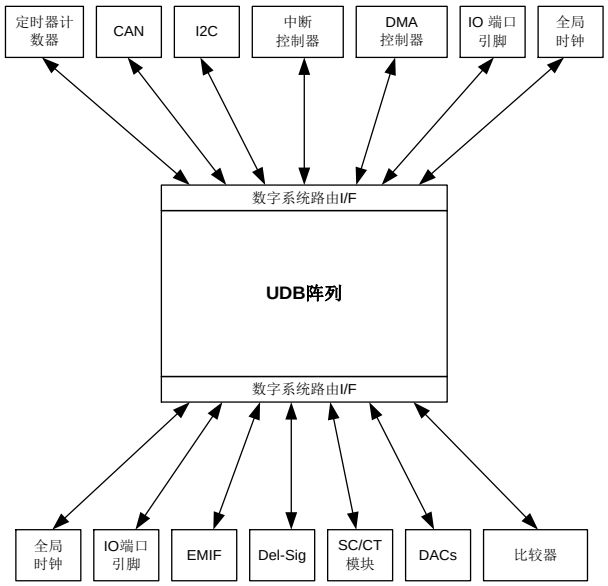
目录

简介 2
什么是数字设计？ 2
数字设计基本概念 3
数字设计方法 3
数字设计的时序主题 5
PSoC 的数字设计 7
PSoC 数字子系统 7
PSoC 3 和 PSoC 5LP 时钟 8
PSoC 中的同步 11
PSoC Creator 静态时序分析（STA） 12
PSoC 数字设计注意事项 13
主题#1：组件数据手册规范 13
主题#2：时钟使用 13
主题#3：控制寄存器 17
主题#4：请勿使用锁存器 17
主题#5：使用数字功能组件 18
主题#6：引脚的接口 19
主题#7：固定模块的接口 20
使用 PSoC Creator STA 报告 21
设置 STA 温度的条件 21
查找 STA 报告 21
STA 报告的内容 21
使用 STA 报告移除警告 28
时钟的额定频率和所需频率 33
保持、恢复和解决违规 35
总结 35
相关应用笔记 35
全球销售和设计支持 38

简介

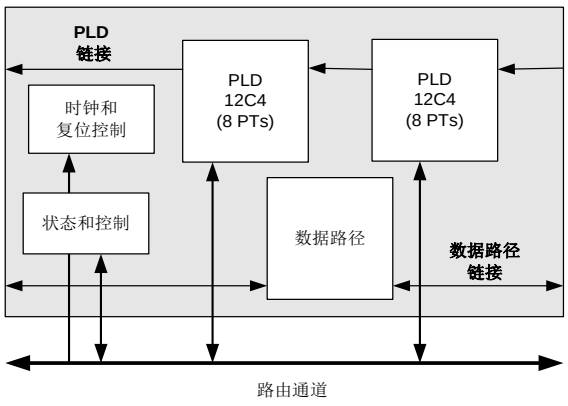
PSoC 3、PSoC 4 和 PSoC 5LP 具有一个功能强大且灵活的可编程数字外设系统。除具有一组固定功能模块（4 个定时器、I2C、USB、CAN）外，它们还提供了 24 个可编程通用数字模块（UDB）和一个被称为数字系统互连（DSI）的广泛信号布线系统。图 1 显示的是如何将这些系统放置在一些 PSoC 器件中以及使用它们的方式。

图 1. PSoC 3 和 PSoC 5LP 可编程数字架构



每个 UDB 包含：两个小型的可编程逻辑器件（PLD）、一个包含可编程 8 位 ALU 的数据路径模块以及其他寄存器和功能，如图 2 所示。

图 2. PSoC 3 和 PSoC 5LP UDB 框图



通过编程 UDB PLD 和数据路径，并在 UDB 内和 UDB 之间路由信号，您可以使复杂的定制外设明显降低 CPU 的负担。在 DSI 中额外路由允许这些外设与其他 PSoC 组件（如 I/O 引脚、DMA 和模拟系统）连接起来，如图 1 所示。

PSoC Creator IDE 提供了一个预编程外设的大型库，通过它您可以创建自己的定制设计。

通过 PSoC 的可配置性您可以在系统级优化设计。但是，它也揭示了一系列传统 MCU 用户尚不了解的新设计注意事项。本应用笔记提供了 PSoC 3 和 PSoC 5LP 的多种数字设计主题。

本应用笔记假定您已经熟悉使用 PSoC Creator 进行 PSoC 3、PSoC 4 或 PSoC 5LP 的开发应用。如果您刚刚接触这些产品，可以参阅 AN54181 — PSoC 3 入门、AN79953 — PSoC 4 入门以及 AN77759 — PSoC 5LP 入门了解它们的有关信息。如果您尚未了解 PSoC Creator，请参考 PSoC Creator 主页中的内容。

如果您尚未了解通用的数字设计原理，那么什么是数字设计？一节提供了基本概念

如果您已熟悉了数字设计原理，并想了解如何使用 PSoC Creator 在 PSoC 器件中对它们进行寻址，请参考 PSoC 数字设计和 PSoC 数字设计注意事项一节。

更多有关如何读取一个 PSoC Creator STA 报告文件，请参见使用 PSoC Creator STA 报告一节。

相关应用笔记中列出了更先进的 PSoC 数字设计资源。

注释： 本应用笔记中所显示的各示例是专为在大多数条件下产生指示的静态时序分析（STA）警告而设计的。但在某些情况下，根据路由和布局情况以及时钟频率，并非总会生成警告。对于添加时序驱动布局（TDP）功能的 PSoC Creator 2.1，便会出现这种情况。

什么是数字设计？

如果您仅工作于 MCU 中，那么不需要直接处理低级数字设计主题。这是因为现在的 MCU 是高度集成的 — 大多数（并非全部）寄存器和外设都位于带有 CPU 的芯片中。您只要对 CPU 进行写代码，以便能够按照正确的方式访问存储器或外设寄存器地址。

与传统的 MCU 不同，PSoC 3 和 PSoC 5LP 允许您将通用数字（和模拟）构建模块设计为自定义外设。但执行前，了解一些数字设计的基本概念非常有用。

本应用笔记中其余部分介绍的所有讨论内容都是基于本节所介绍的简单概念。

数字设计基本概念

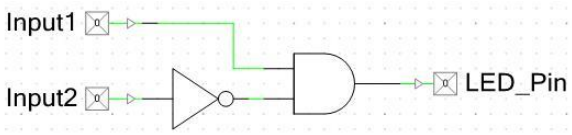
实际上数字设计过程与对 CPU 进行的写固件相同，但数字设计比 CPU 进行的更快。例如，考虑到如下情况：当一个条件为“true”（逻辑高或“1”）并且另一个条件为“false”（逻辑低或“0”）时，您需要打开指示灯 LED。那么您可以通过写入 C 代码来执行功能：

```
if (GetInput1() && !GetInput2())
{
    SetOutput(1); // turn on the LED
}
else
{
    SetOutput(0); // turn off the LED
}
```

要执行该功能需要多个 CPU 周期。

但是，您可以使用低级逻辑门来简单实现该功能，而不需要写入 CPU 代码，如图 3 所示。

图 3. 示例数字逻辑功能

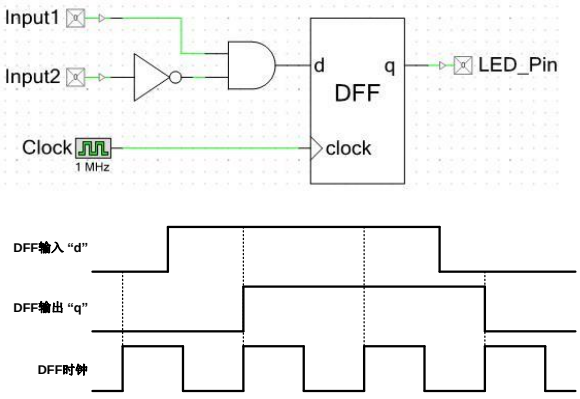


与 MCU 不同，您可以在 PSoC Creator 项目原理图中绘制图 3 中所显示的设计，然后编程它并在 PSoC 器件中直接实现。引脚、门和路由的延迟大约需要几纳秒时间。但不需要几 CPU 周期。

寄存器和 DFF

图 3 中的设计存在一个主题：如果输入快速或连续发生变化，那么您可能在输出上得到不希望的干扰。我们可以使用一个时钟寄存器（还被称为“D 型触发器”，简称“DFF”）来降低干扰的发生，如图 4 所示：

图 4. 示例时钟数字逻辑功能



只有“时钟”从 0 切换为 1（意味着它在上升沿）时，寄存器的输出“q”才发生变化。输出值与输入“d”值相同，并保持该值，直到时钟下一个上升沿到来为止。

通常，一个时钟是一个不断运行的方波，具有固定的频率和 50%的占空比。通过 PSoC Creator 您可以实现多个 PSoC 时钟，该时钟频率可高达器件频率限制 — 请参见第 8 上的 PSoC 3 和 PSoC 5LP 时钟。

使用一个时钟寄存器，您可以更好地控制事件检测的时序和响应。这种情况也被称为“同步”一个异步输入和一个已知的时钟信号。

如果发生时钟边沿时输入同时发生更改，那么在一段时间不能确定输出。这种情况被称为亚稳态 — 有关该主题的更多信息，请参考第 6 页上亚稳态和寄存器时序一节。

数字设计方法

传统 MCU 的用户可以将 MCU 分为多种等级：

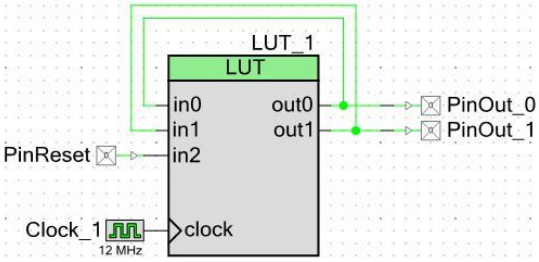
- **汇编程序**：高度优化代码，但产生低编码效率和非可移植性。通常只对特定的模块进行该操作。
- **C**：良好的编码效率和可移植性。最常用的 MCU 编码语言。
- **面向对象的语言（Java、C ++等）**：允许定义自定义对象，便于重复使用和实现较高的编码效率。通常，该语言只在存储器大小至少为 256 K 的更高级 MCU（如 32 位）中使用。

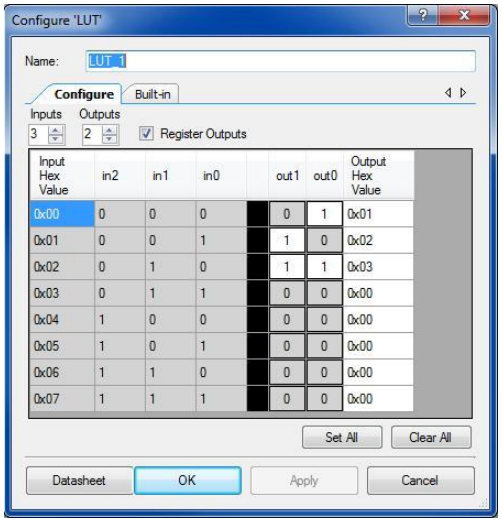
同样，数字设计可以按照多个级别执行：

- **门级**：对单独门（AND、OR、XOR、NOT）和 DFF 进行连接，从而执行逻辑功能。PSoC Creator 为所有逻辑功能提供了门符号，并在一个稍高的水平上提供了一个查询表（LUT）组件。

  使用 LUT 可以更容易设计复杂的逻辑功能，而不需要连接单独门。另外，您不要编程一个 MCU 就能很容易的设计状态机。图 5 显示的是一个具有复位引脚的 2 位计数器示例：

图 5. 基于 LUT 的状态机设计





更多有关在 PSoC 3 和 PSoC 5LP 中设计状态机的详细信息，请参见 AN62510 — 使用 PSoC 3 和 PSoC 5LP 实现状态机

- **代码级**：PSoC Creator 支持硬件设计语言 Verilog，该语言的结构和语法与 C 语言相同。
- **数据路径编程**：门、LUT 和 Verilog 级设计通常是在 PSoC UDB 的 PLD 中实现的（请参见图 2）。要编程 UDB 的其他部分（如数据路径），PSoC Creator 会提供一个数据路径配置工具。大部分基于 UDB 的设计都使用了数据路径和 PLD。
- **组件**：与面向对象编程相同，您使用 PSoC Creator 可以定义易于实施且能重复使用的自定义组件。这些组件可以使用多种 PSoC 资源，包括 UDB 数据路径和 PLD。

注释： Verilog 与数据路径编程和组件开发的详细说明超出了本应用笔记的范围。更多有关这些主题的信息，请参考相关应用笔记。

数字设计调试

固件开发必须包括调试工具和调试技术。一些用于固件的良好调试技术包括：

- 单步运行每一行代码，并在可能条件下至少观察一次其运行情况。
- 进行开发的同时进行调试，即在进行下一个模块之前编写和测试当前小模块。

同样，在数字设计中您可能需要查看门、模块或组件级的性能。使用 PSoC 实现该操作的最简单方法是路由所需信号以测试引脚，然后使用一个示波器或一个逻辑分析仪来观察引脚的活动。与固件相同，当调试数字设计时，您想应该执行下面操作：

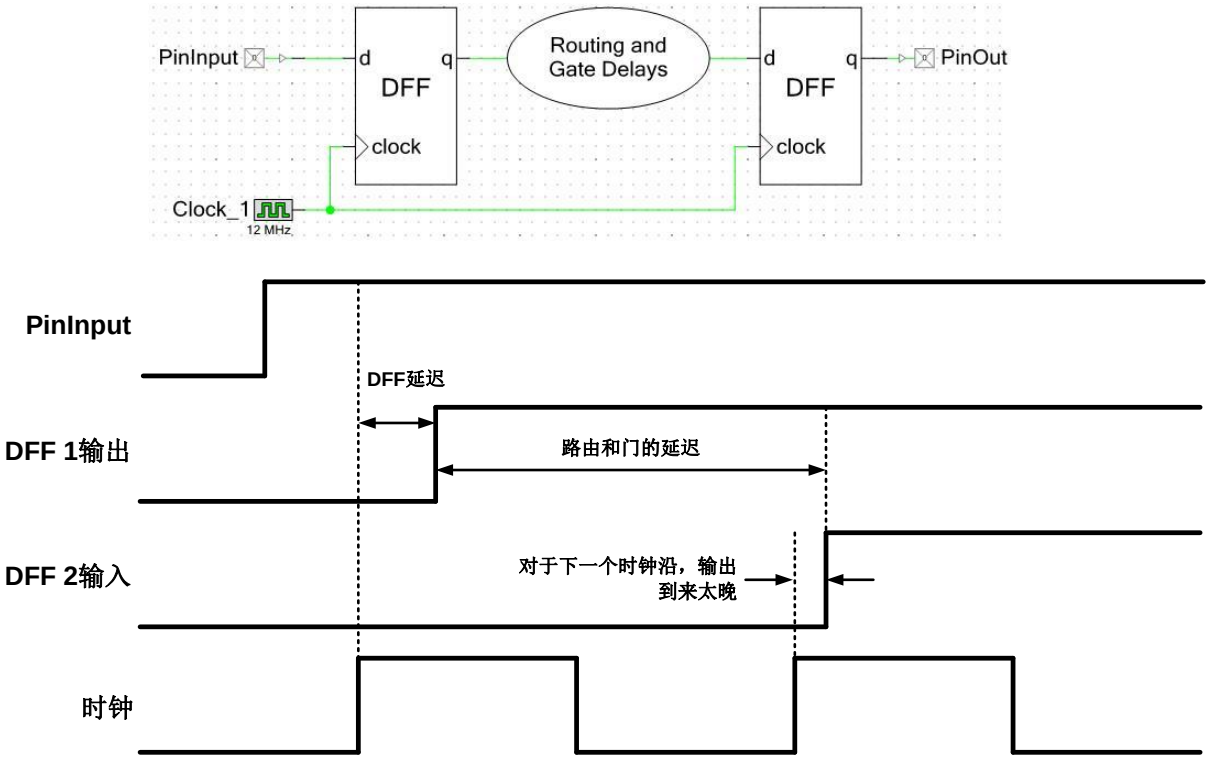
- 可能条件下至少对每个信号观察一次。
- 进行开发的同时进行调试。在这种情况下，转到设计的下一阶段前要对设计（在逻辑级或组件级）的一小型部分进行测试和运行。

数字设计的时序主题

数字设计中最常见的主题是时序。DFF、门和其他逻辑元素以及引脚与布线始终存在延迟。当延迟足够长时，会使一个到达太晚（有时太早）的信号无法被正确处理，这时会明显看到一个时序问题，如图 6 所示。

在图 6 中，DFF 1 的延迟加上 DFF 之间的逻辑与路由的延迟会使 DFF 2 输入的信号到达的太晚。要避免发生这种情况，您必须减慢时钟或降低 DFF 之间的延迟。

图 6. 多寄存器设计的延迟

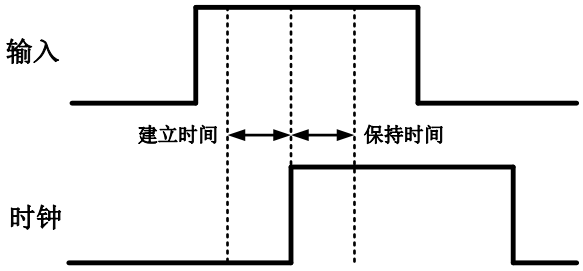


亚稳态和寄存器时序

如上所述，如果在发生时钟边沿时一个寄存器输入也同时被改变，那么不能确定输出，这种条件被称为亚稳态。

更准确地说，在发生时钟边沿之前和之后的一段时间内，输入必须稳定。寄存器指定的建立时间和保持时间定义了这个时间周期，如图 7 所示。

图 7. 寄存器建立和保持时间

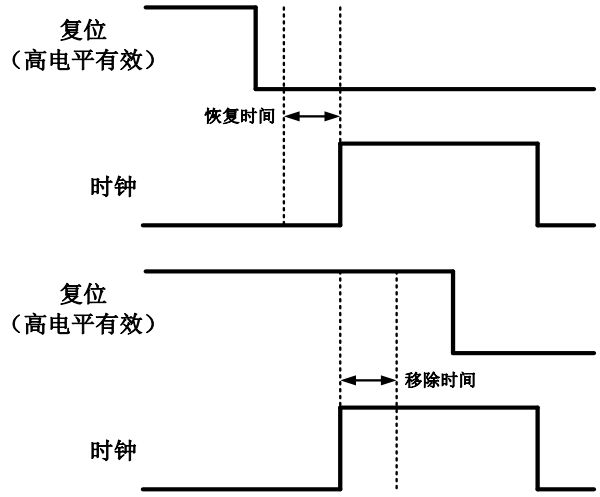


通常，在整个建立时间和保持时间内一个输入应保持其状态，否则输出信号不确定 — 一个亚稳态条件。

恢复和移除时间类似于建立和保持时间，但它们描述了与时钟相对的异步控制时序，并未描述寄存器的输入。异步控制示例是 DFF 复位输入 — 请参考第 18 页上的图 30 及相关文本。

恢复时间是一个时钟边沿到来前异步控制必须保持为无效状态的最短时间。移除时间是发生时钟边沿后控制必须保持有效状态的最短时间。请参见图 8。

图 8. 寄存器恢复时间和移除时间



静态时序分析（STA）

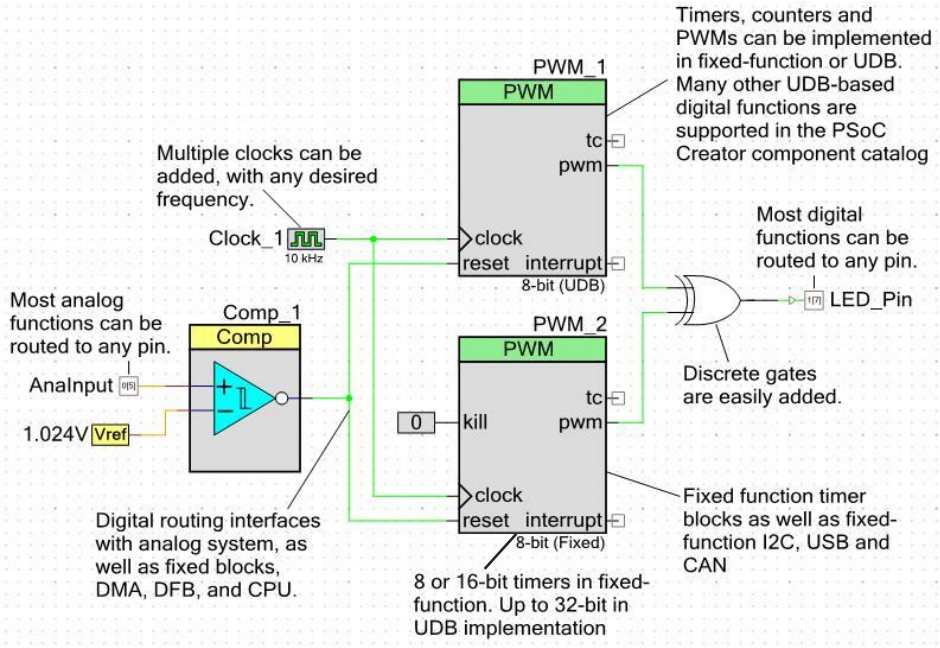
调试数字设计的一个重要部分是静态时序分析 （STA）。STA 用于评估数字设计，并计算信号输出和输入之间的延迟。根据这些延迟，它能够计算设计中所使用的每一个时钟的最大有效频率。更多有关信息，请参见 PSoC Creator 静态时序分析。

PSoC 的数字设计

虽然本应用笔记中没有详细介绍，但有必要了解 PSoC Creator 表达 PSoC 数字子系统的灵活的能力，以及它很容易便能够快速创建一个复杂的设计的能力。

例如，图 9 显示的是一个 PSoC 3 和 PSoC 5LP 的设计，它通过一个模拟电压电平来使能或关闭一个“呼吸”LED。两个 PWM 的周期稍微不同，使 LED 逐渐变得更暗或更亮。只要了解一些有关 PSoC Creator 的知识，您花费半个小时便能够在一个 PSoC 器件中建立和运行一个设计。

图 9. 对 PSoC 3 和 PSoC 5LP 进行的 PSoC Creator 数字设计示例（对 PSoC 4 是相同的）



在另一个示例中，设计一个数字系统来控制多路复用 ADC 输入，并与 DMA 进行连接，以保存 SRAM 中的数据，从而无需使用 CPU 也能够创建一个先进的模拟数据采集系统。

更多有关信息，请参考 PSoC Creator 网页，或 PSoC Creator 中的所有示例设计。

PSoC 数字子系统

如上所示，PSoC 数字子系统最重要的部分是一个 UDB 阵列（请参考图 1）。当您将一个数字组件放置在 PSoC Creator 原理图中时，将对一组数据路径、PLD 和其他 UDB 寄存器（请参见图 2）进行相关配置，从而实现所需功能。

DSI 位于 UDB 内和 UDB 周围，它是一个宽结构的可编程开关，可以连接一个 UDB 内的信号、UDB 对之间信号、通过 UDB 阵列的信号，以及 UDB 阵列与 PSoC 中多种其他模块之间的信号。在构成一个智能的自定义外设中，它提供 PSoC 自身极大的灵活性。

在 PSoC Creator 原理图中绘制数字组件间的导线时，起始点和目的点之间的一组 DSI 开关被打开，用于实现该连接。每个开关都有 1 到 2 纳秒的延迟。该延迟很短，但如果通过多个开关来路由同一个信号，那么总延迟将显著增大。

要管理该现象，PSoC Creator 具有以下功能：

- 时序驱动路由（TDR），通过路由具有关键时序的信号，来减少 DSI 开关和路由延迟
- 时序驱动布局（TDP），通过选择 UDB 资源，从而进一步优化组件时序

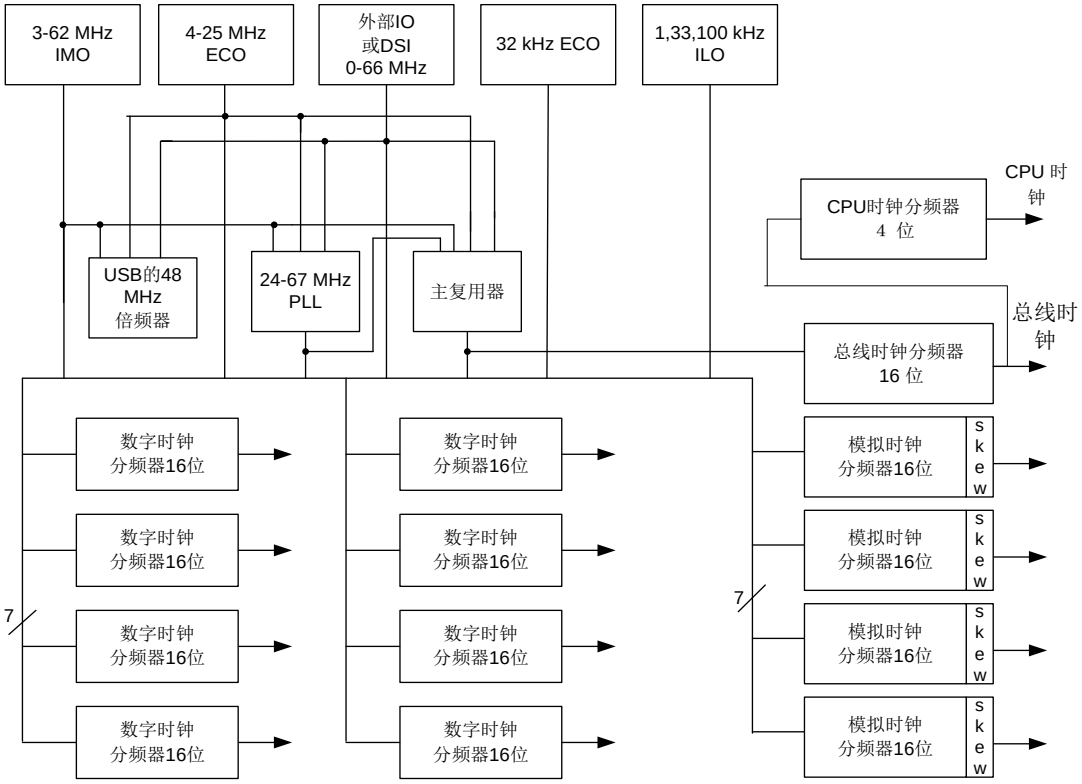
有关 PSoC 数字子系统的更多信息，请参考器件数据手册或技术参考手册（TRM）。

PSoC 3 和 PSoC 5LP 时钟

因为本应用笔记中多项内容重点介绍了时钟的同步信号，所以我们能够简单了解 PSoC 时钟系统以及它在 PSoC Creator 中是如何表示的。

PSoC 3 和 PSoC 5LP 具有一个非常灵活的时钟生成和分配系统。图 10 显示的是 PSoC 3 示例；PSoC 5LP 与此相同。PSoC 4 具有更简单的时钟系统；相关信息，请参考 PSoC 4 器件数据手册。

图 10. PSoC 3 时钟子系统

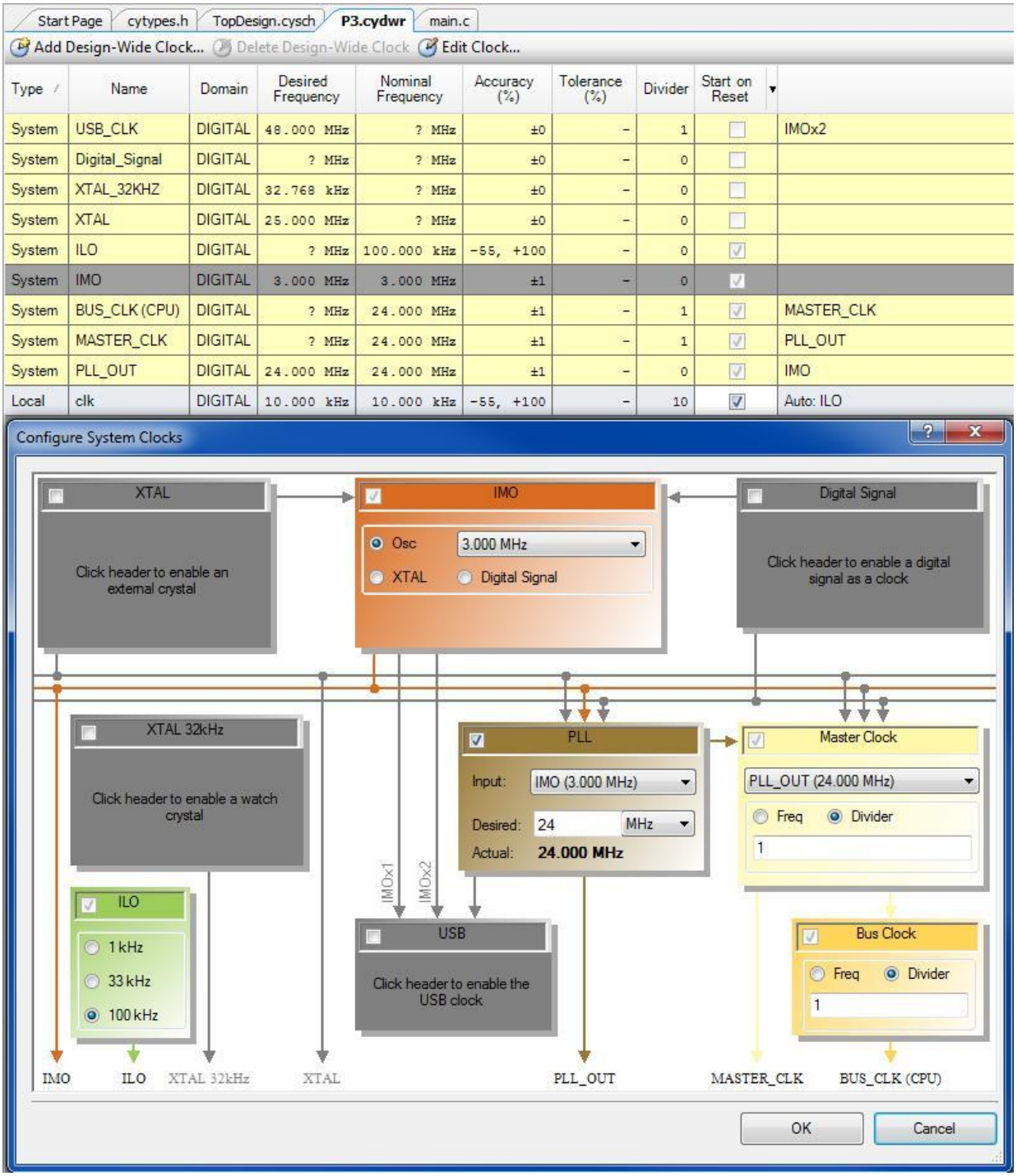


存在多种不同的时钟源，它们具有不同的频率范围、精度和电源要求。这些时钟源被路由到系统中其余部分，从而生成总线、数字和模拟时钟。对于数字系统，八个时钟分频器提供了具有所需频率的时钟。

通过使用 PSoC Creator 设置时钟系统。用户定义的时钟被放置在原理图中（请参见第 7 页上的图 9），系统时钟设置被显示在设计范围资源（DWR）中。第 9 页上的图 11 显示的是图 9 中示例的时钟配置。

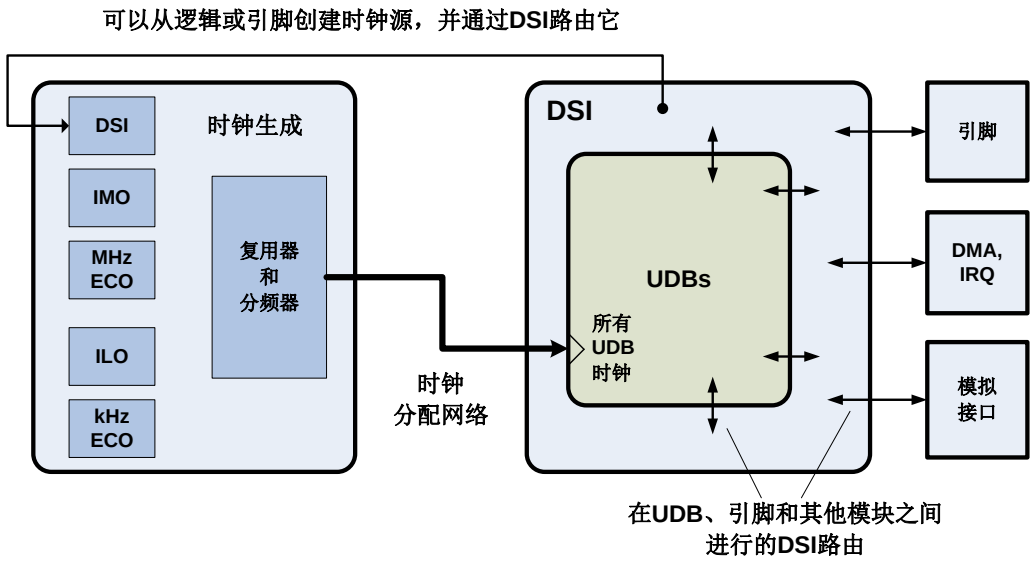
重点需要注意的是：一个主时钟和一个总线时钟以及它们的使用方法。因为总线时钟的频率通常与主时钟的相同，所以这两个术语通常可以替换使用。但是，它们具有不同的功能。主时钟是所有其他 PSoC 时钟（包括总线时钟且使用图 10 显示的分频器）的时钟源。它的频率不能小于 PSoC 中所有其他时钟的频率。总线时钟是 CPU、DMA、DFB 和其他主要模块的时钟源。

图 11. PSoC 3 和 PSoC 5LP 的示例 PSoC Creator 时钟配置



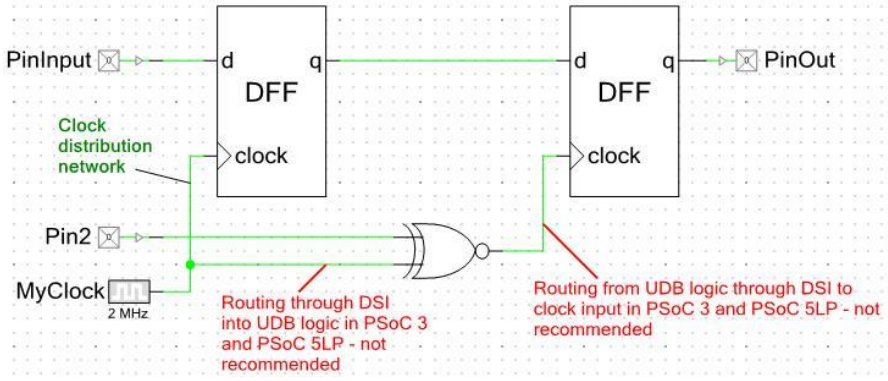
可以将任意时钟源路由到 UDB 时钟输入端。这些时钟在专用网络（与 DSI 隔离）上被分配到 UDB 和其他模块中的时钟输入端，如图 12 所示。

图 12. PSoC 3 和 PSoC 5LP 时钟分配和 DSI



在 PSoC 3 和 PSoC 5LP 中，时钟可被路由到 UDB 中的逻辑端，如图 13 所示。但通常情况下，推荐将时钟源（如时钟组件）直接连接到时钟输入端，因为这样可以获得时钟分配网络的良好性能。另外也要避免将时钟路由到门或数据输入端，并要避免将门或数据输出路由到时钟输入端。有关详细信息，请参见第 13 页上的主题#2：时钟使用和第 15 页上的不门控时钟，使用使能组件。

图 13. 时钟路由路径



更多有关 PSoC 时钟的信息，请参考 AN60631 — PSoC 3 和 PSoC 5LP 时钟资源，或者 PSoC Creator 系统参考指南中的“时钟”一节。

PSoC 中的同步

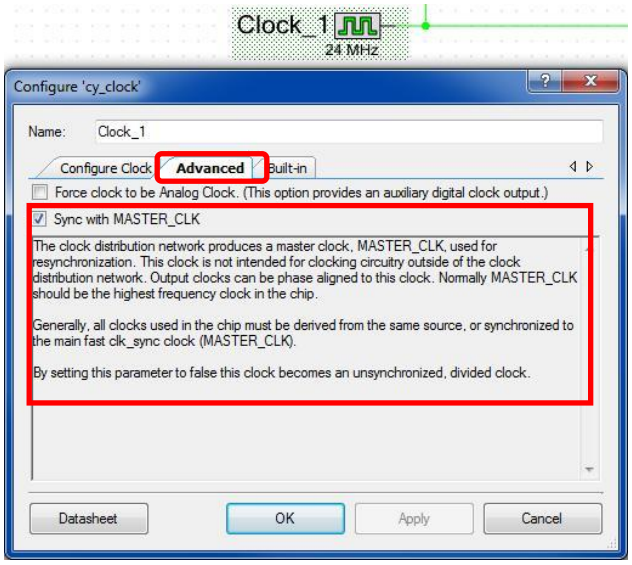
如前面所介绍的内容，一个异步数字系统会引起不希望的信号干扰。设计 PSoC 作为一个同步系统运行，从而使能数字子系统以及 CPU 与 DMA 之间的通信。通常，除了与 CPU 或 DMA 不相交互的基于 PLD 的逻辑外，不支持其它异步信号。

下面列出了一些 PSoC 中同步需要注意的实例。

PSoC 3 和 PSoC 5LP 时钟

如前面所示，八个时钟分频器用于 PSoC 3 和 PSoC 5LP，从而提供符合所需频率的数字子系统时钟。默认情况下，这些时钟与主时钟同步。通过 PSoC Creator 时钟组件配置对话框实现该选择，如图 14 所示。

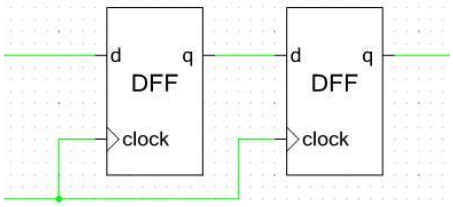
图 14. PSoC 3 和 PSoC 5LP 的时钟同步设置



引脚（所有 PSoC 器件）

进入某个输入引脚的信号为异步信号，默认情况下在路由到 DSI 之前，这些信号要经过双重同步。双重同步，如图 15 所示，可避免亚稳态现象和系统意外操作发生。

图 15. 双重同步

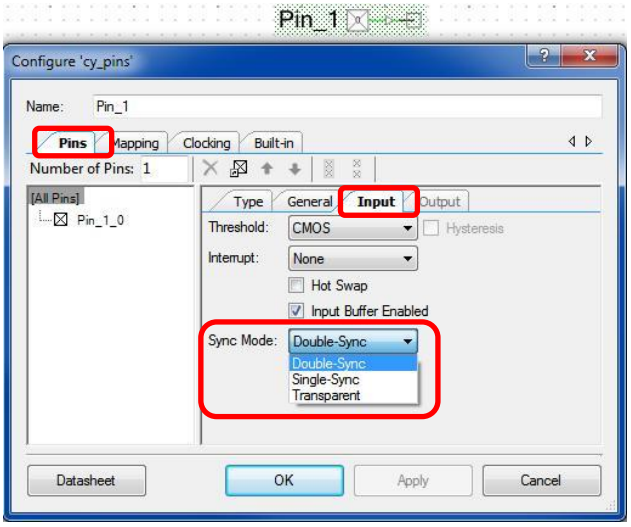


请注意，双重同步器导致的延迟等于一个到两个同步时钟周期。对于 PSoC 3 和 PSoC 5LP，同步时钟是总线时钟，其频率与主时钟相同。对于 PSoC 4，HFCLK 是同步时钟。

如果同步时钟频率不大于 33 MHz，那么将通过引脚模块中的专用电路实现双重同步。如果总线时钟高于 33 MHz，则引脚输入将通过 DSI 被路由到 UDB 电路，且双重同步在 UDB 中进行。这样，在主时钟，总线时钟或 HFCLK 发生改变（低于或高于 33 MHz）时，路由和 STA 结果也会发生改变。请参考第 30 页上有关图 50 所讨论的内容。

如前面所示，时钟源可来自 PSoC 外部，通过 I/O 引脚进入系统。如果该输入是提供给总线时钟的时钟源，那么需要关闭输入同步。通过 PSoC Creator 引脚组件配置对话框来实现该操作，如图 16 所示 — 将同步模式（Sync Mode）设置为透明（Transparent）。

图 16. 输入引脚同步



引脚输入通常要经过同步处理，这样 CPU 或 DMA 才能准确读取它们。但在某些情况下，PSoC 的外部系统可能对时序有较严格的要求。在这些情况下，同步延迟可能是不可接受，并且输入引脚同步需要被关闭。相关详细信息，请参考第 19 页上的主题 #6：引脚的接口。

引脚输出也具有同步选择要求，它可以与总线时钟（单）同步。该选择用以减少时滞，或减少多个输出之间存在的相关延迟。默认情况下，它被禁用。

USB 控制和状态寄存器

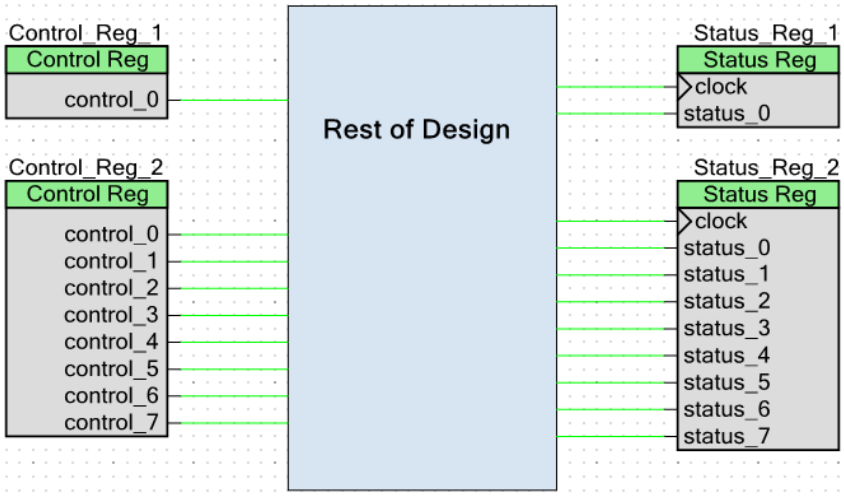
另外两个主要的 UDB 组件是控制寄存器和状态寄存器 — 每个 UDB 均有一个控制寄存器和一个状态寄存器。通过控制寄存器，CPU / DMA 可将信号驱动到数字子系统内，而 CPU / DMA 通过状态寄存器可读取来自数字子系统的信号。PSoC Creator 组件使用了这两个控制器。欲了解有关这些寄存器的更多信息，请参考技术参考手册。

除正常操作模式外，这些寄存器还提供了可用于减少时序限制的特殊模式。例如，可将状态寄存器重新配置为 4 位双重同步器 — 为了实现该操作，请使用 PSoC Creator 同步组件，如第 14 页上的图 22 所示。

另一方面，控制寄存器能在三种不同的模式下工作 — 直接、同步和脉冲。在同步和脉冲模式下，来自 CPU / DMA 的输入（即为与总线时钟同步的输入）将与其他时钟再次同步。更多详细信息，请参见第 17 页上的主题#3：控制寄存器。

注释： PSoC Creator 控制和状态寄存器组件可被配置为一个到八个控制输出或状态输入，如图 17 所示。虽然完全可以创建多个控制或状态组件，但每个组件均使用了整个 UDB 寄存器。最好将控制或状态组件与多个输出或输入配合使用。对于使用了同一个时钟的状态输入，这种方法更有效。

图 17. 单线和多线控制和状态寄存器



PSoC Creator 静态时序分析（STA）

在 PSoC Creator 项目创建时会自动创建一个 STA。并生成一个报告，介绍设计中限制每个时钟频率的重要路径。如果实际的时钟频率超过了计算出的最大频率，则报告会显示设计中的时序违规。

为了避免 STA 警告，需要了解 PSoC 中的同步和数字特性， 如下一节所介绍的内容。

PSoC 数字设计注意事项

本部分包含一系列主题，包括使用 PSoC Creator 对 PSoC 3、PSoC 4 和 PSoC 5LP 进行数字设计时的注意事项和最佳实践。

主题#1：组件数据手册规范

当您使用 PSoC Creator 组件目录中由 UDB 构成的数字组件（如计数器或 SPI）时，需要特别注意组件的时序规范。这些规范在组件数据手册中进行了详细说明，如图 18 所示。

图 18. UDB 计数器时序规范

Table 2. AC Characteristics

Parameter	Description	Min	Typ	Max[2]	Units
f_{CLOCK}	Component clock frequency				
	8-bits UDB Up Counter	–	–	39	MHz
	8-bits UDB Counter with direction	–	–	39	MHz
	16-bits Up Counter	–	–	33	MHz
	16-bits Counter with direction	–	–	33	MHz
	24-bits UDB Up Counter	–	–	29	MHz
	32-bits UDB Up Counter	–	–	26	MHz

根据组件类型，基于 UDB 的组件的频率受前面介绍的路由延迟的限制。务必保证您的时钟频率不超过数据手册所规定的最大频率。

有关的更多信息，请参考数字组件的数据手册。

主题#2：时钟使用

尽量使用低频率的时钟

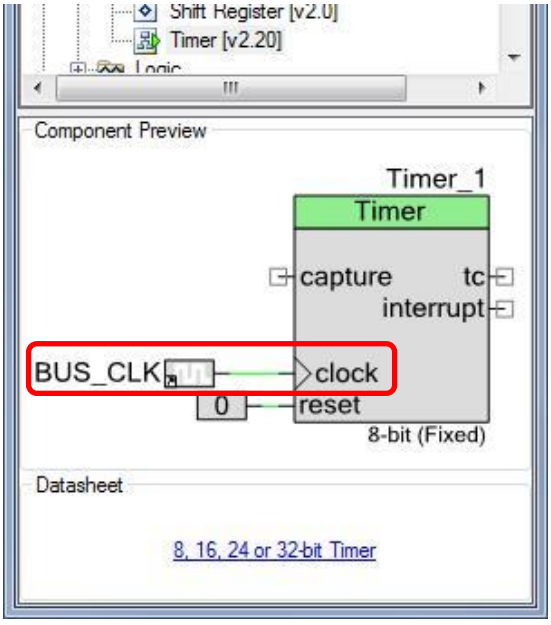
如前一主题的介绍，PSoC Creator 组件的时钟频率需要满足组件数据手册的规定。

通常，要尽量降低时钟频率，并保证该频率可满足应用的要求。这样有助于 PSoC Creator TDR 系统对您设计中的重要路径进行优化路由，并且可以降低系统噪声和功耗。

使用宏的时钟设置

在某些情况下，高频率时钟不能立即显示在设计中，如图 19 所示。PSoC Creator 组件目录中的一些宏包含了时钟组件，默认情况下它们的时钟由总线时钟提供。不使用总线时钟删除时，应该更改时钟频率。

图 19. 包含了总线时钟的定时器宏



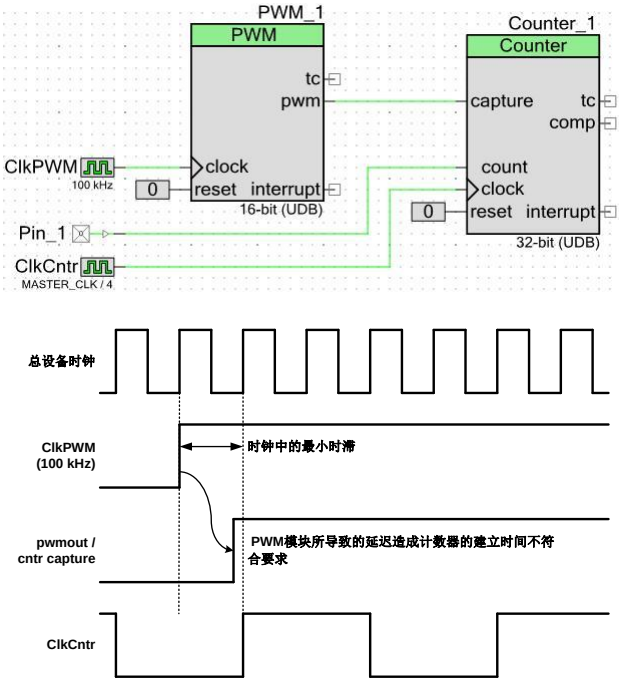
多时钟

如前面所述，PSoC 中几乎所有数字信号都是同步信号。甚至大多数时钟均与频率更高的时钟同步。时钟通常（并非始终）与主时钟同步，如图 10 所示。

虽然各时钟可能与同一个时钟源同步，但它们之间不一定彼此同步，如图 20 所示。在该示例中，一个频率计数器定期捕获 PWM 信号，两个时钟与主时钟（在 PSoC 4 中为 HFCLK）同步，但它们之间不一定同步。因此，如果主时钟频率较高，那么 PWM 组件所导致的延迟会造成计数器组件的建立时间不符合要求。

注释： 本部分和其他部分中的示例是专门为大多数情况下产生上述 STA 警告而设计的。但在某些情况下，根据路由和布局情况以及时钟频率，并非总会生成警告。

图 20. 多时钟示例



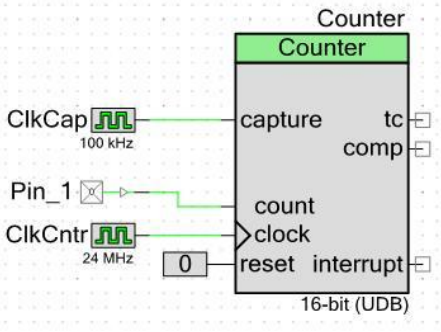

```
Warning-1366: Setup time violation found in
a path from clock (ClkPWM) to clock
(ClkCntr).
```

在这种情况下，因为两个时钟之间的最小时间差异等于一个主时钟周期，因此降低 ClkCntr 的频率是无效的。可以通过降低主时钟频率或将 PWM 输出与 ClkCntr 同步（如下一节所述）以解决该问题。您还可以参考第 28 页上的使用 STA 报告移除警告内容。

使用同步组件进行时钟同步

在某个简单的示例中，如图 21 所示，使用了一个低频时钟直接控制计数器捕获频率。这样会导致发生 STA 警告。

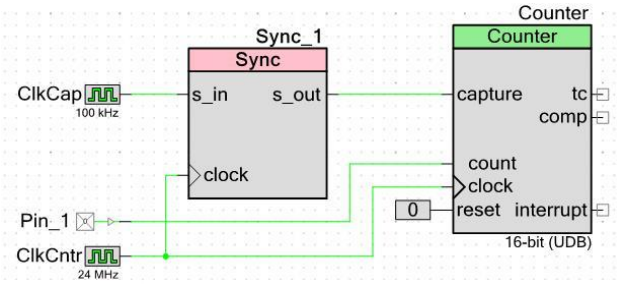
图 21. 多时钟示例（续）




```
Warning-1350: Path(s) exist between clocks
ClkCap(routed) and ClkCntr, but the clocks
are not synchronous to each other
```

如想清除该警告，需要通过 PSoC Creator 同步组件使低频率时钟与计数器同步，如图 22 所示。

图 22. 同步时钟



请注意，因为同步组件实现了双重同步（请参考第 11 页上的图 15 和第 12 页上的 UDB 控制和状态寄存器内容），同步输出占空比不等于 50%。当同步时钟频率比被同步时钟频率快很多时，可接受这种现象。但两个时钟频率几乎相同时，同步输出将不会生成时钟 — 它会处于某个 DC 电平。同步时钟频率至少要比被同步的时钟频率快两倍；理想状态快四倍。

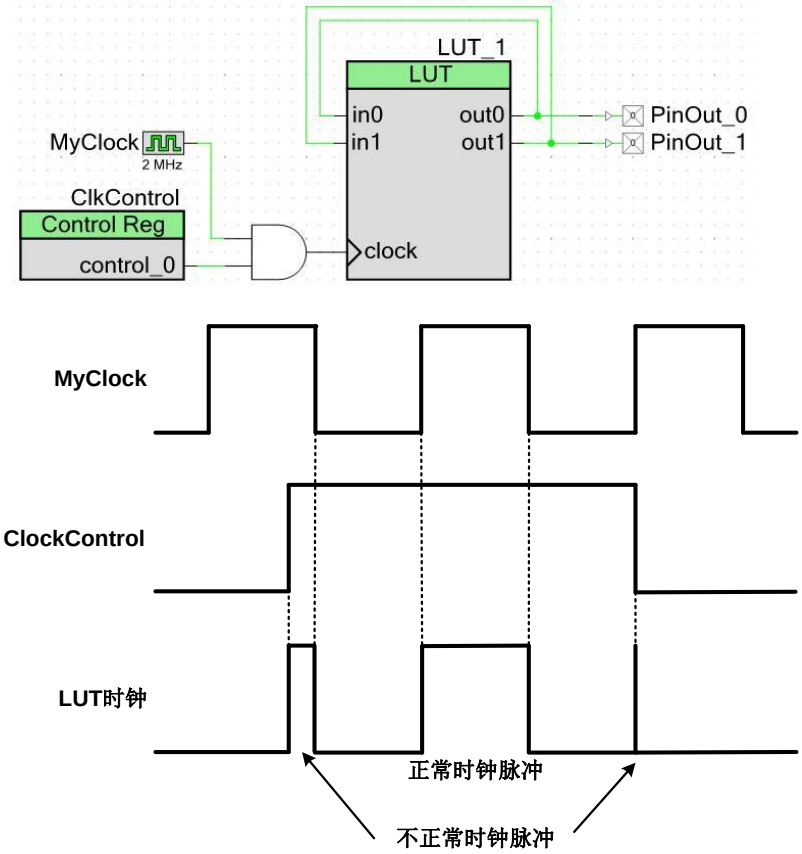
有关同步组件的更多信息，请参考同步组件数据手册。

注释： 在图 21 和图 22 中，PSoC 4 ClkCap 不能直接被连接到计数器的‘capture’终端或 Sync_1 的‘s_in’终端。因此，建议先通过使用一个切换触发器（TFF）组件来加倍 ClkCap 的频率（200 kHz）。

不门控时钟，使用使能组件

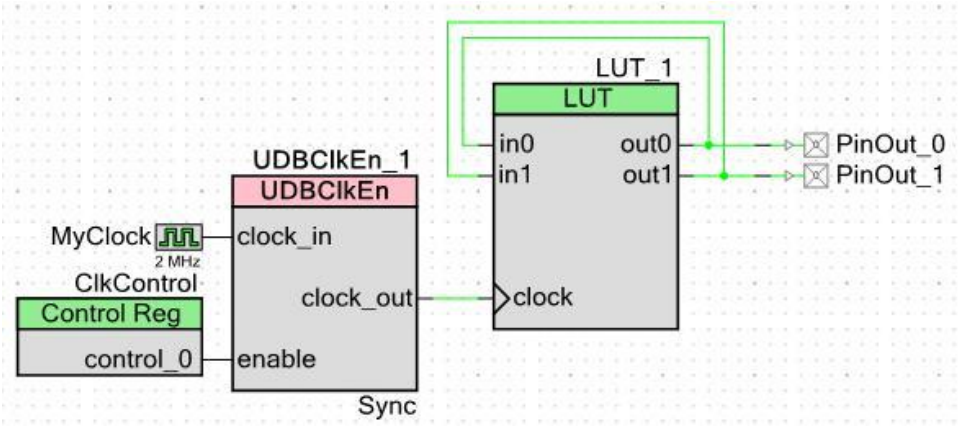
在许多设计中，可以只通过调整相应时钟的开关状态和门控操作来控制某功能。然而，图 23 中介绍了通过异步控制门控时钟可导致不正常时钟和意外结果的示例。

图 23. 门控时钟示例



如果您的功能是基于 UDB 的（如图 23 中的示例），更好的方法是使用 UDBClkEn 组件，如图 24 所示。该组件为时钟提供了一个使能信号（该信号被控制并且由电压电平触发），并能够可选地将输入时钟与总线时钟同步。有关 UDBClkEn 组件的更多信息，请参考 UDBClkEn 组件数据手册。对于 PSoC 4，时钟可以直接连接到 UDBClkEn ‘clock_in’终端。

图 24. 使用 UDBClkEn 组件



注释： 数字功能组件（请参考第 18 页上的主题#5：使用数字功能组件）中均有一个嵌入式 UDBClkEn 组件。因此，UDBClkEn 组件通常只需要与 DFF 或 LUT 组件配合使用。UDBClkEn 组件不能驱动其他 UDBClkEn 组件；尝试执行该操作会发生这样的错误：

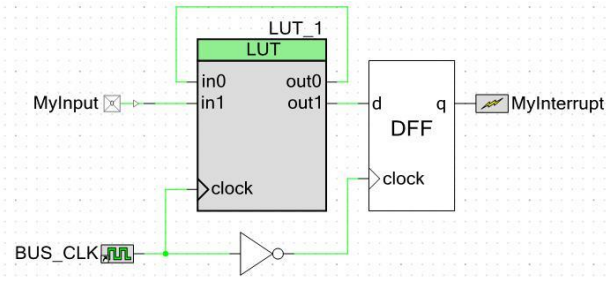
```
Error: mpr.M0096: The UDB Clock/Enable components may only drive clocks in UDB content and
\UDBClkEn_1:udbclkenable\ is connected to clock_in on ... which is not a clock input to the
UDB. (App=cydsfit)
```

反向时钟是可以接受的。

本文档并非始终使用反向时钟，如图 25 所示。这是因为，每个 UDB 具有一个时钟控制模块，可适用于反向和非反向时钟 — 反相器符号指示 UDB 使用其反向时钟输入。

通过反向时钟，您可使用较慢的时钟，因为时钟的双边沿均被使用。然而，这样可能导致更严重的时序主题，因为时钟频率将增加一倍。

图 25. 使用反向时钟的设计示例

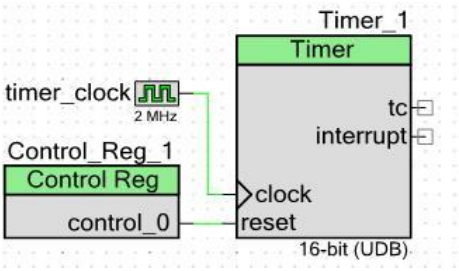


主题#3：控制寄存器

如第 12 页上 UDB 控制和状态寄存器所述，固件与数字系统交互的标准方法是通过控制寄存器。每个 UDB 有一个 8 位控制寄存器；因此，24 个 UDB 可提供高达 192 个控制位。通过 DSI，控制寄存器的输出可被路由到数字系统的任何位置。

但对于这种寄存器，需要考虑有关时钟的一些注意事项，从而可以避免发生 STA 警告。例如，在图 26 中，使用一个控制寄存器来驱动基于 UDB 的定时器的复位输入。该寄存器在默认情况下由总线时钟提供脉冲，图中没有显示该内容。总线时钟较快时，总线时钟和定时器时钟之间可能发生建立时间违规：

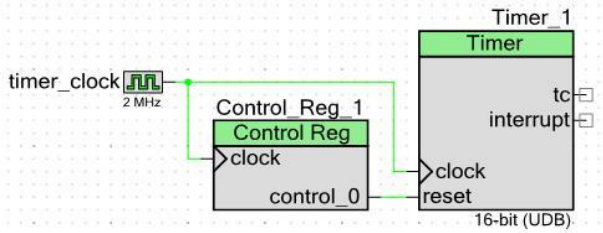
图 26. 控制寄存器引起的 STA 警告




```
Warning-1366: Setup time violation found in
a path from clock (CyBUS_CLK) to clock
(timer_clock).
```

只要将控制寄存器配置为同步模式即可解决主题，如图 27 所示：

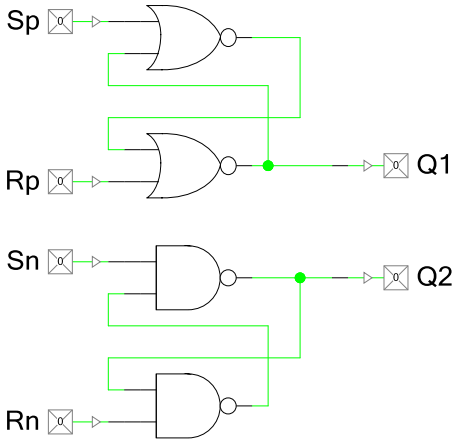
图 27. 控制寄存器的同步模式



主题#4：请勿使用锁存器

图 28 显示的是典型的置位-复位（S-R）锁存器设计。（在上面的电路中，S 和 R 是高电平有效，而在下面的电路中，S 和 R 为低电平有效。）锁存器与寄存器或 DFF 相同，但其稳定性较低，即为：输入端发生短时脉冲时，可能导致输出的意外改变。并且，当两个输出同时有效时，锁存器将处于非稳定状态。因此，建议使用寄存器替换锁存器。

图 28. 典型的 S-R 锁存器设计



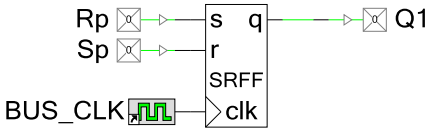
如果您使用了上述某种设计，PSoC Creator 将弹出以下的警告：

```
Warning-1361: The design contains a
combinational loop. Check the design for
unintentional latches. Breaking the loop
at ...
```

这意味着 STA 研究中不包括该环路。

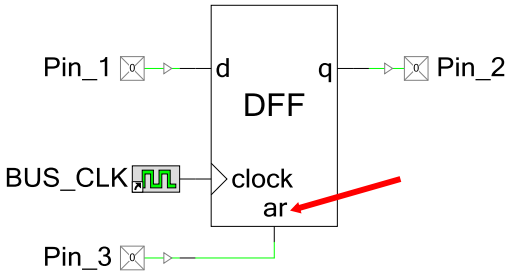
图 28 中还存在其他主题，即是异步现象（请参见有关图 3 和图 4 的讨论）并且每个锁存器使用两个 UDB PLD 宏单元。最好使用 PSoC Creator SRFF 组件，如图 29 所示。在该设计中，每一个锁存器仅使用了一个宏单元。

图 29. 同步 S-R 锁存器组件



请注意，图 29 中的锁存器可使用 LUT 组件设计得到；请参见图 5。还可以通过 DFF 组件进行设计，PSoC Creator DFF 组件的配置可带有复位和预设功能，如图 30 所示。

图 30. 带复位功能的 DFF

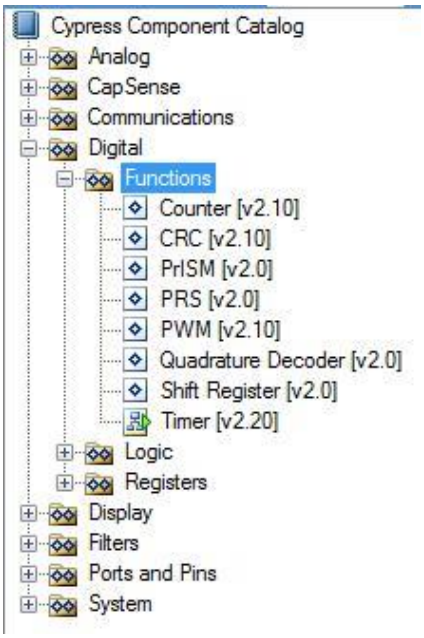


在配置对话框中，您可以选择同步或异步预设或复位。被激活时，异步输入立即生效，这样会导致亚稳态现象 — 请参见第 6 页上亚稳态和寄存器时序中所讨论的复位和移除时间。同步输入将在发生下一个时钟沿时生效，因此可以预料其状态。更多信息，请参考 DFF 组件数据手册。

主题#5：使用数字功能组件

PSoC Creator 所生成的大多数 STA 警告均与组件目录中数字功能文件夹所列出的组件相关；请参见图 31。这些组件的特殊点在于它们均有时钟输入。

图 31. 数字功能组件



UDB 数据路径，PLD 和其他模块的时钟来自多种时钟源。数字功能组件利用了该特性，并且与它们的时钟输入（而并非与总线时钟）同步。这样，当异步信号与其他输入连接时会导致 STA 警告，如使用同步组件进行时钟同步内容所述 — 请参见第 14 页上的图 21 和图 22。

本主题不针对固定功能定时器模块，虽然它们均有时钟并由计数器，定时器和 PWM 组件所支持。更多信息，请参考组件的 TRM 或数据手册。

主题#6：引脚的接口

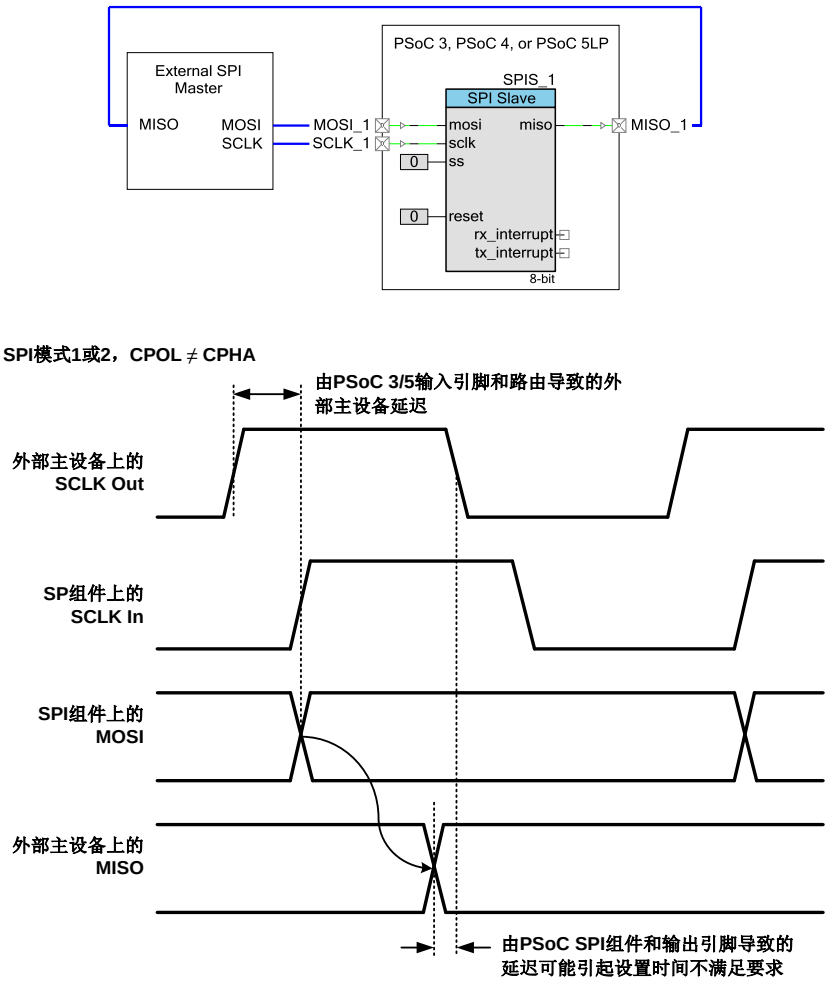
PSoC3 和 PSoC5LP I/O 引脚可以连接到 PsoC 中供应/接收电源的各个模块，其中包括 CPU、DMA、模拟和 LCD。它们可通过 DSI 进行连接，因而可通过数字结构路由。这样会使引脚面对与其他组件相同的同步主题。引脚输入和输出可能是同步或异步的，彼此之间完全独立。

如第 11 页上引脚部分所述，引脚输入信号可能与总线时钟进行双重同步。默认设置是使能同步，但在某些特殊情况下，需要关闭输入同步。

最好的示例是 PSoC Creator SPI 从设备（SPIS）组件，如图 32 所示。PCB 路由、PSoC 引脚、DSI 路由和 UDB 导致的延迟构成了主设备的 SCLK 和 MOSI 输出与 MISO 输入之间的延迟。这样会导致读取主设备的 MISO 输入不正确。

PSoC Creator SPI 主设备（SPIM）组件也面对同样的主题。

图 32. SPIS 组件的 SPI 时序



用于降低 PSoC Creator SPIS 组件的延迟的方法均不大有效。但是，同步 MISO、SCLK 和 MOSI 引脚会加大延迟。对于输入引脚，这种现象更为明显 — 如第 11 页上引脚所述的内容，当输入引脚被双重同步时，将引起较大的延迟。通常，对于 SPI 引脚，应关闭同步。

所有 PSoC Creator SPI 组件均作为宏使用，其中包括引脚组件，它们的同步事先均被关闭。如果您没有使用宏，而是直接使用了 SPIM 或 SPIS 组件，那么需要保证连接到这些组件的引脚都关闭了同步。

主题#7：固定模块的接口

如图 1 所示，PSoC 3 和 PSoC 5LP 的 UDB 系统和固定数字以及模拟子系统之间存在多种接口。图 33 显示了固定模块和基于 UDB 组件（在此是计数器）之间的接口示例。请注意，定时器组件是基于功能固定的定时器/计数器/PWM（TCPWM）模块。

此时，PSoC Creator STA 将不包括在模拟和数字子系统间路由的信号。此外，如果发生了时序问题，计数器的输入端可能无法正确捕获比较器和 ADC 的输出。

此外，还可能发生 STA 警告，提示固定模块时钟与 UDB 时钟之间存在异步路径，如：

```
Asynchronous path(s) exist from
"Clock_1(fixed-function)" to "Clock_1".
```

这是由于时钟网络和 DSI 之间存在的时序差异；有关更多信息，请参见第 10 页上的图 12 和相关介绍。通常，您应该将同步组件添加到固定模块输出；这些模块输出被路由到基于 UDB 的组件，如图 34 所示。

图 33. 固定模块/UDB 接口

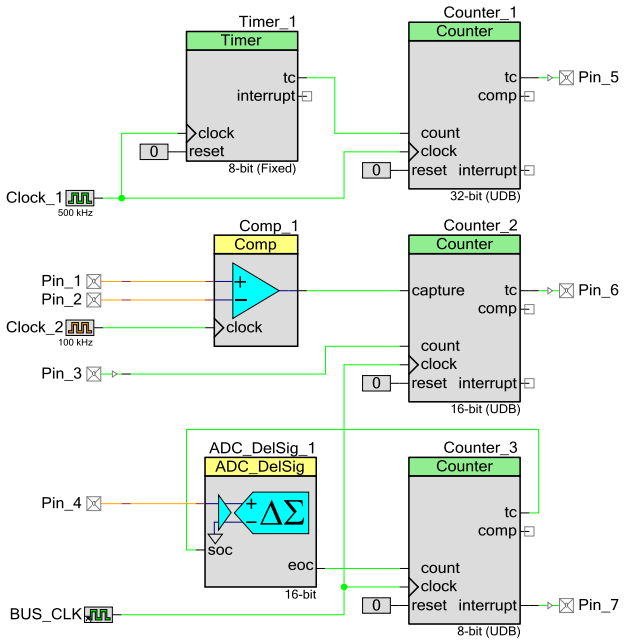
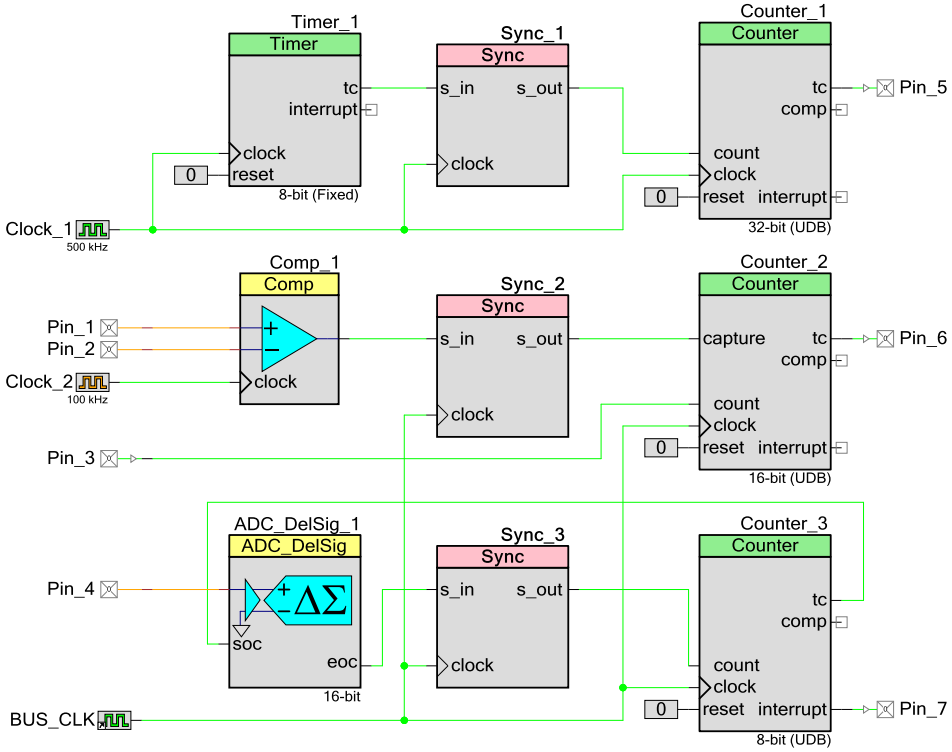


图 34. 固定模块与同步组件接口



使用 PSoC Creator STA 报告

如前面的介绍，PSoC Creator STA 功能评估一个数字设计，并生成关于每个时钟最大许可频率的报告，并查看是否存在潜在时序问题。如果在编译项目过程中您得到一个 STA 警告，那么需要重新查看 STA 报告，从而全面了解和解决该警告。

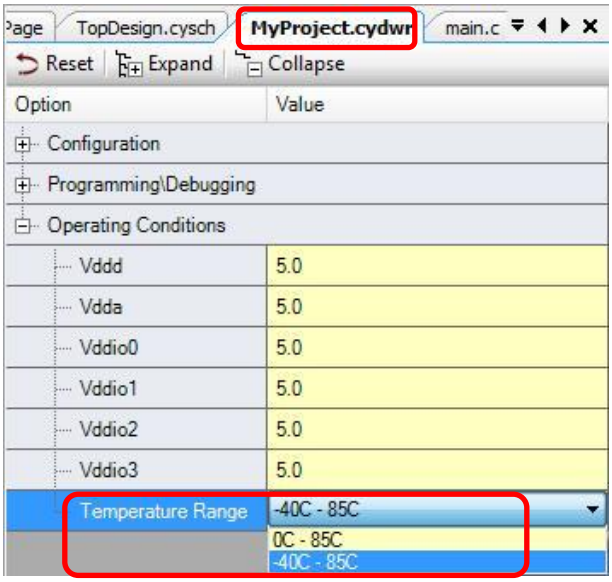
本节介绍了使用 STA 报告的策略和最佳实践。

注释： 虽然基于 PSoC 4 的设计会发出 STA 警告，由于 PSoC 3 和 PSoC 5LP 上的数字和时钟系统更加复杂，所以更有可能是它们引起的 STA 警告。因此，本节将介绍基于 PSoC 3 和 PSoC 5LP 的设计。

设置 STA 温度的条件

PSoC 器件运行于一个广泛的供电电压和温度范围内，并在选择 TDR 限制时会考虑到整个温度规范。但是，如果您的设计运行的温度环境更易受控，那么更容易满足 TDR 要求，并能够实现更加灵活的路由。您设计的温度范围是一项 PSoC Creator 项目系统的设置内容，如图 35 所示。

图 35. 器件温度范围设置

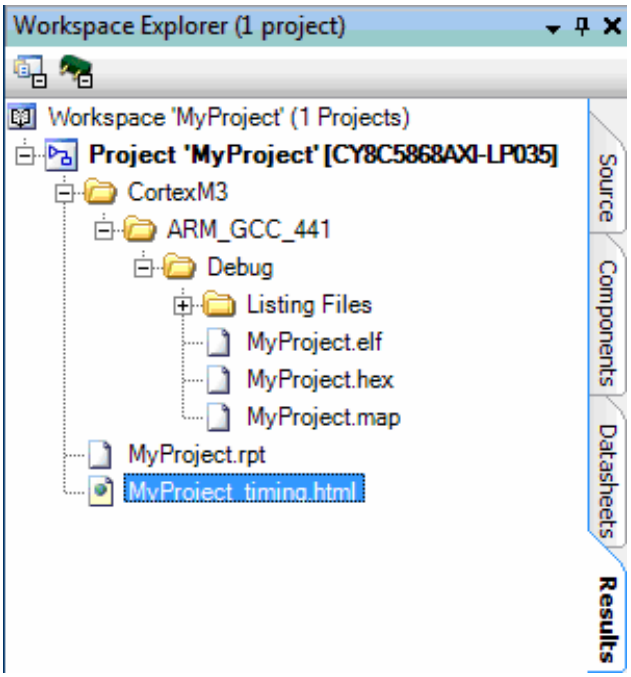


如果您为项目选择较小的温度范围，TDR 工具可以容易查找符合时序的路由解决方案。这样会影响在编译项目时您是否会被 STA 警告。

查找 STA 报告

STA 报告是一个自动生成的 HTML 文件，并可以在 PSoC Creator 项目的“Result”选项卡中找到，如图 36 所示。

图 36. PSoC Creator 项目 STA 报告文件



STA 报告的内容

除了时序违规和时钟汇总外，STA 报告文件还具备某些内容显示了时钟和其他信号之间建立、保持、恢复和移除时间的详细分析。更多有关 STA 报告的部分的信息，请参考 PSoC Creator 帮助文件“静态时序分析”部分的内容。

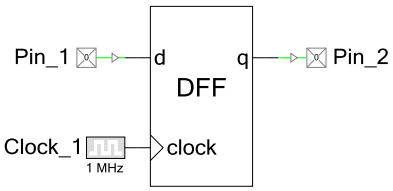
下面的这些示例详细显示了设计和 STA 报告数据间的关系。

示例 1

从一个简单的示例开始，如第 22 页上的图 37 所示。在该示例中，所有组件和时钟设置都是默认针对 PSoC 3 和 PSoC 5LP 进行的：

- 主时钟和总线时钟的频率都是 24 MHz
- Pin_1 具有被打开的输入同步（请参见第 11 页上的引脚一节）
- Pin_2 具有被关闭的输出同步
- Clock_1 与主时钟同步

图 37. STA 报告示例 1



Static Timing Analysis

Project : Fig36
Build Time : 07/22/13 17:10:53
Device : CY8C3866AXI-040
Temperature : -40C - 85/125C
Vdda : 5.00
Vddd : 5.00
Vio0 : 5.00
Vio1 : 5.00
Vio2 : 5.00
Vio3 : 5.00
Voltage : 5.0
Vusb : 5.00

Expand All | Collapse All | Show All Paths | Hide All Paths

+ Timing Violation Section

No Timing Violations

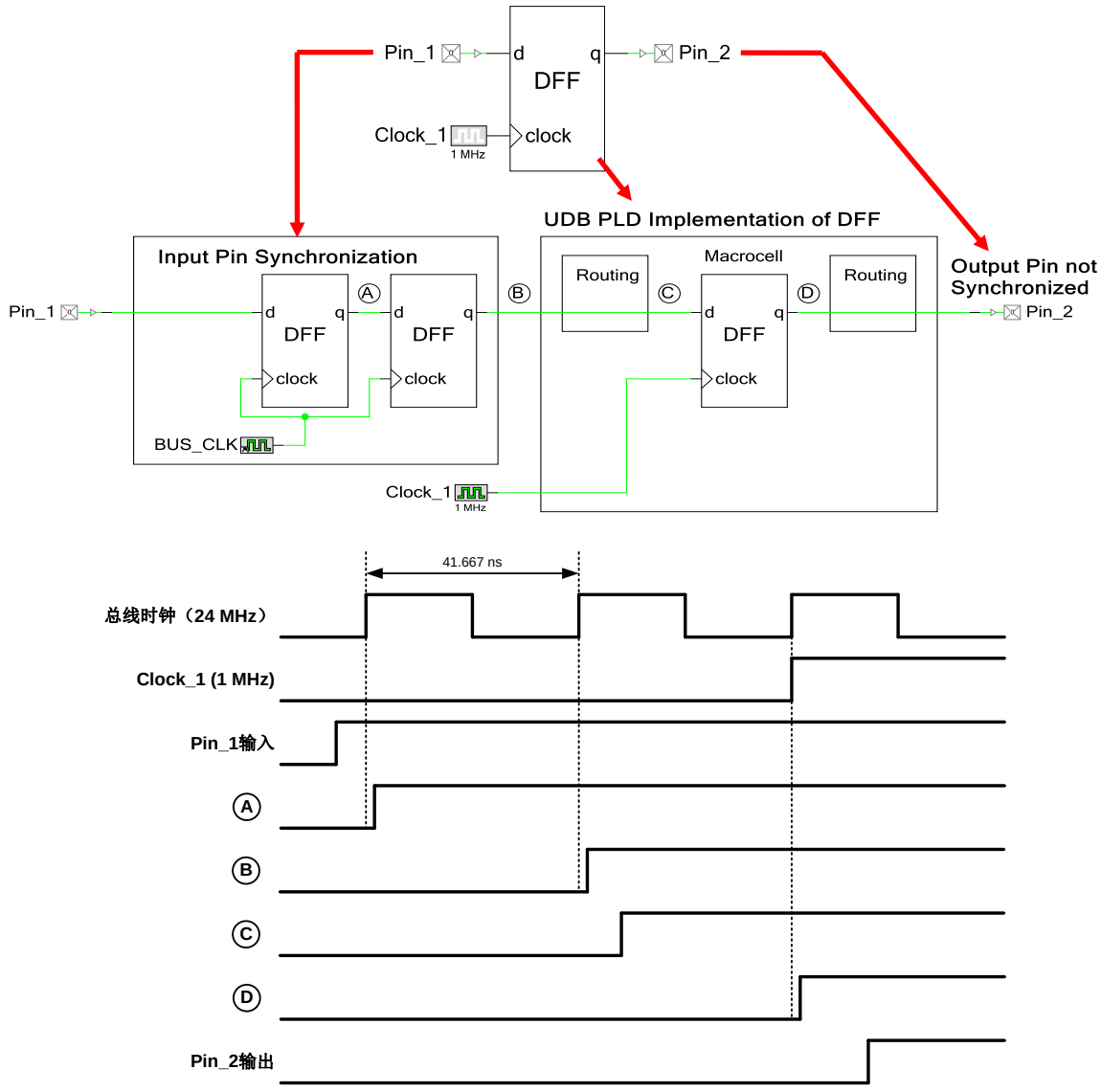
+ Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	N/A	
Clock_1	CyMASTER_CLK	1.000 MHz	1.000 MHz	69.677 MHz	
CyBUS_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	69.677 MHz	
CyPLL_OUT	CyPLL_OUT	24.000 MHz	24.000 MHz	N/A	

打开 STA 报告文件，它会显示时序违规汇总和具有时钟速度违规的时钟列表。在该示例中，不存在任何违规。报告文件显示总线时钟和 Clock_1 的频率高达 69.677 MHz（实际上可能高于器件频率的限制），但不会发生时钟速率违规。

查看 STA 报告时序详细信息前，需要查看原理图中各设计之间的关系以及 PSoC 器件中实际发生的情况。在第 23 页上的图 38 中，输入引脚同步模块显示：当输入引脚同步模块被打开时，输入引脚信号会通过两个由总线时钟提供脉冲的寄存器。然后，通过 DSI 从该引脚上将信号路由到某个在原理图中实现 DFF 的 UDB 宏单元。最后，通过 DSI，宏单元输出被路由到非同步的输出引脚上。

图 38. 示例#1 硬件实现



有关 STA 报告中的延迟的详细信息，请点击“Expand All”（全部展开）（请参见图 37）。第 24 页上的图 39 显示的是展开报告和小节中编号的意义：

- **Setup Subsection（设置小节）**，“Path Delay Requirement”（路径延迟要求）：41.6667 ns，相应于 24 MHz 总线时钟
- 在相同小节中的表内，“Delay”（延迟）为：14.352 ns，它是从 Pin_1 输出到宏单元输入计算得到的延迟，是由路由和门延迟引起的
- 在同一表中，“Slack”（余量）为：27.315 ns，或 41.667 和 14.352 的差 — 宏单元的计算建立时间
- **Clock to Output Section（时钟输入至数据输出节）** “延迟”为 22.883 ns，它是从宏单元输出到 Pin_2 的延迟，是路由延迟引起的。

点击“Show All Paths”（显示所有路径）可以进一步展开报告，但通常不需要展开。在图 39 中，**Hold Subsection**（保持小节）内容被隐藏 — 有关保持违规的信息，请参考第 35 页上的保持、恢复和移除违规一节。

图 39. 展开的 STA 报告#1 以及相应时序

Static Timing Analysis

Project : MyProject
Build Time : 07/23/13 15:55:08
Device : CY8C3866AXI-040
Temperature : -40C - 85/125C
Vdda : 5.00
Vddd : 5.00
Vio0 : 5.00
Vio1 : 5.00
Vio2 : 5.00
Vio3 : 5.00
Voltage : 5.0
Vusb : 5.00

Expand All | Collapse All | Show All Paths | Hide All Paths

– Timing Violation Section

No Timing Violations

– Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	N/A	
Clock_1	CyMASTER_CLK	1.000 MHz	1.000 MHz	69.677 MHz	
CyBUS_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	69.677 MHz	
CyPLL_OUT	CyPLL_OUT	24.000 MHz	24.000 MHz	N/A	

– Register to Register Section

– Setup Subsection

– Source Clock : CyBUS_CLK : Positive edge(Required Frequency 24 MHz)

– Destination Clock : Clock_1 : Positive edge(Required Frequency 1 MHz)

Path Delay Requirement : 41.6667ns(24 MHz)
Affects clock : CyMASTER_CLK

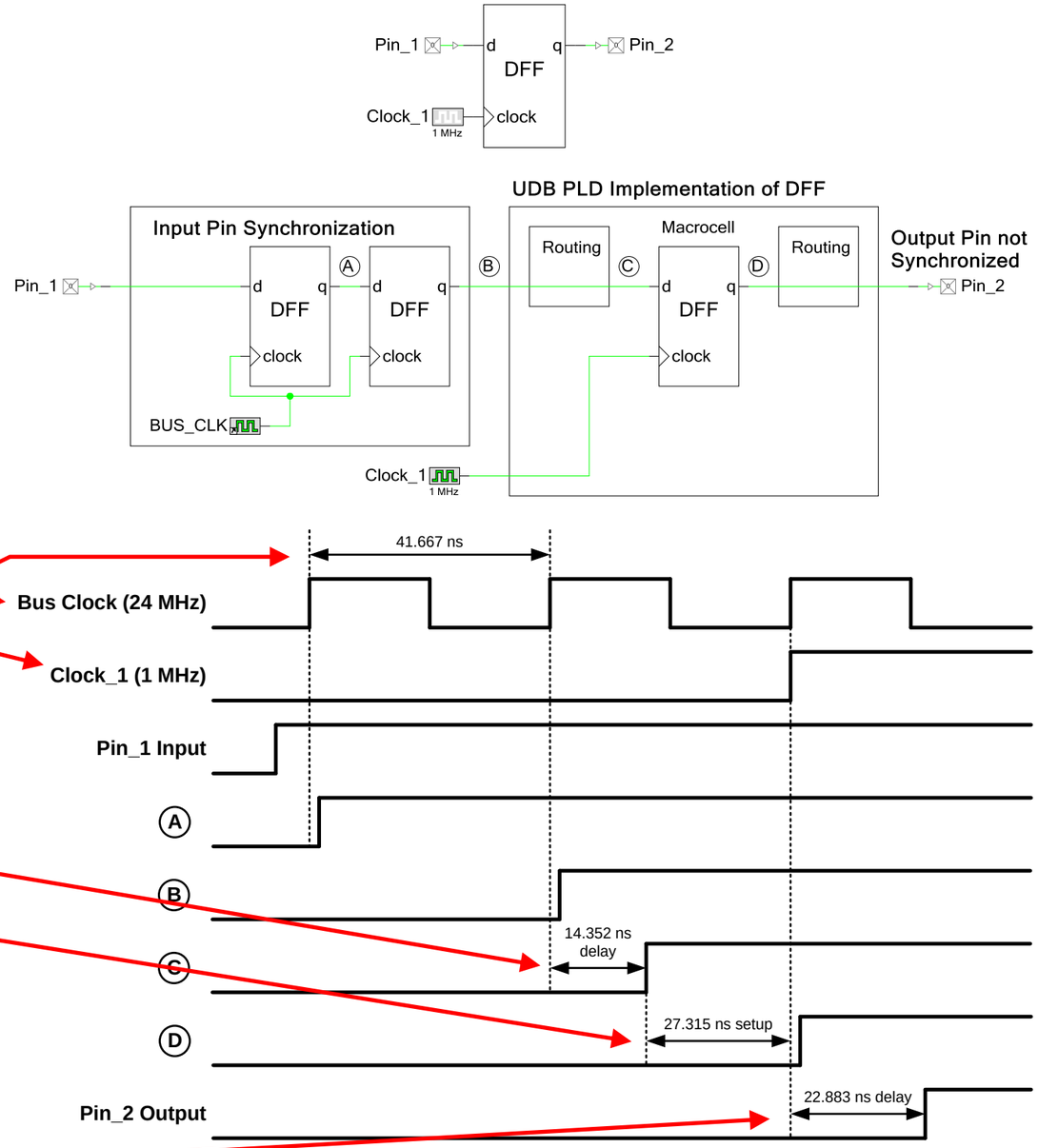
Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
Pin_1(0)/fb	cydff_1/main_0	69.677 MHz	14.352	27.315	

+ Hold Subsection

– Clock To Output Section

– Clock_1

Source	Destination	Delay (ns)
cydff_1/q	Pin_2(0)_PAD	22.883



如果您点击报告中各表内任意一行，该行会被展开，以显示更多信息。在图 40 中，**Clock to Output Section** 被展开，以显示从 Clock_1 通过 DFF 到 Pin_2 产生延迟的每一部分。

展开的报告显示大部分延迟（16.2 ns）是由自身引脚引起的。较小延迟（5.4 ns）是从宏单元输出通过 DSI 再路由给引脚的；较小延迟是由自身 DFF 引起的。

图 40. 示例#1 展开的“时钟输入至数据输出”章节

- Clock_1

Source	Destination	Delay (ns)
cydff_1/q	Pin_2(0)_PAD	22.883

Type	Location	Fanout	Instance/Net	Source	Dest	Delay (ns)
macrocell1	U(3,4)	1	cydff_1	cydff_1/clock_0	cydff_1/q	1.250
Route		1	cydff_1	cydff_1/q	Pin_2(0)/pin_input	5.433
iocell	P0[1]	1	Pin_2(0)	Pin_2(0)/pin_input	Pin_2(0)/pad_out	16.200
Route		1	Pin_2(0)_PAD	Pin_2(0)/pad_out	Pin_2(0)_PAD	0.000
Clock					Clock path delay	0.000

示例#2

现在，我们通过将 DFF 替换为基于 UDB 的 8 位计数器来更改示例 1，如图 41 所示：

图 41. STA 报告示例 2

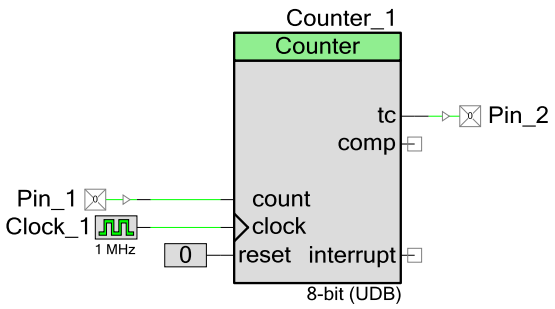


图 42 显示了相应 STA 报告文件的一部分。虽然没有时序违规，但 **Clock Summary Section**（时钟汇总节）显示现在总线时钟的容限已经很小；最大频率已经降低为~35 MHz。

图 42. 示例#2 的 STA 报告

Static Timing Analysis

Project : MyProject
Build Time : 07/23/13 16:15:15
Device : CY8C3866AXI-040
Temperature : -40C - 85/125C
Vdda : 5.00
Vddd : 5.00
Vio0 : 5.00
Vio1 : 5.00
Vio2 : 5.00
Vio3 : 5.00
Voltage : 5.0
Vusb : 5.00

Expand All | Collapse All | Show All Paths | Hide All Paths

- Timing Violation Section

No Timing Violations

- Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	N/A	
Clock_1	CyMASTER_CLK	1.000 MHz	1.000 MHz	35.688 MHz	
CyBUS_CLK	CyMASTER_CLK	24.000 MHz	24.000 MHz	35.688 MHz	
CyPLL_OUT	CyPLL_OUT	24.000 MHz	24.000 MHz	N/A	

+ Register to Register Section

+ Clock To Output Section

要了解原因，请查看第 26 页上图 43 中的 **Register to Register Section（寄存器到寄存器节）**、**Setup Subsection（设置小节）**中介绍的内容：

图 43. 示例#2 展开的设置小节

- **Register to Register Section**

  - **Setup Subsection**

    - **Source Clock : Clock_1 : Positive edge(Required Frequency 1 MHz)**

      - **Destination Clock : Clock_1 : Positive edge(Required Frequency 1 MHz)**

Path Delay Requirement : 1000ns(1 MHz)

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
\Counter_1:CounterUDB:sCTRLReg:SyncCtl:ctrlreg\/control_7	\Counter_1:CounterUDB:sC8:counterdp:u0\/cs_addr_1	45.304 MHz	22.073	977.927	
\Counter_1:CounterUDB:count_stored_i\/q	\Counter_1:CounterUDB:sC8:counterdp:u0\/cs_addr_1	48.256 MHz	20.723	979.277	
\Counter_1:CounterUDB:sC8:counterdp:u0\/z0_comb	\Counter_1:CounterUDB:sC8:counterdp:u0\/cs_addr_0	51.454 MHz	19.435	980.565	
\Counter_1:CounterUDB:sC8:counterdp:u0\/z0_comb	\Counter_1:CounterUDB:sSTSReg:rstSts:stsreg\/status_3	61.501 MHz	16.260	983.740	
Net_22/q	\Counter_1:CounterUDB:sSTSReg:rstSts:stsreg\/status_3	70.507 MHz	14.183	985.817	
\Counter_1:CounterUDB:prevCompare\/q	\Counter_1:CounterUDB:sSTSReg:rstSts:stsreg\/status_0	92.902 MHz	10.764	989.236	
\Counter_1:CounterUDB:sC8:counterdp:u0\/z0_comb	Net_22/main_0	94.073 MHz	10.630	989.370	
\Counter_1:CounterUDB:sC8:counterdp:u0\/z0_comb	\Counter_1:CounterUDB:sSTSReg:rstSts:stsreg\/status_1	99.661 MHz	10.034	989.966	

      - **Source Clock : CyBUS_CLK : Positive edge(Required Frequency 24 MHz)**

        - **Destination Clock : Clock_1 : Positive edge(Required Frequency 1 MHz)**

Path Delay Requirement : 41.6667ns(24 MHz)
Affects clock : CyMASTER_CLK

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
Pin_1(0)/fb	\Counter_1:CounterUDB:sC8:counterdp:u0\/cs_addr_1	35.688 MHz	28.021	13.646	
Pin_1(0)/fb	\Counter_1:CounterUDB:count_stored_i\/main_0	69.677 MHz	14.352	27.315	

UDB 数据路径和 PLD 的组合构成一个计数器组件。图 43 中的 **Source Clock: Clock_1** 表显示的是 UDB 组件内部连接到计数器是由计数器时钟输入（这里为 Clock_1）提供的时钟脉冲。该表也显示：由于 Clock_1 的频率只为 1 MHz，因此余量为 1 µs — 该值是一个非常合适的容限。

Source Clock: CyBUS_CLK 表中显示的是更有意思的数据。请注意，要求该表的路径延迟等于 24 MHz 总线时钟的周期大小。第 14 页上的多时钟会介绍原因。

该表显示输入引脚（由总线时钟同步）被路由到计数器内的两个 UDB 组件。这两个 UDB 组件中一个延迟大于 27 ns，这样可使余量低于 14 ns。这样会导致总线时钟被限制成 35 MHz。

最后，我们可以将 8 位计数器替换为 32 位计数器，并且为了进行演示，需要将总线时钟频率增大到 28 MHz。重新建立后，我们会得到一个建立时间违规，如图 44 所示。

图 44. 32 位计数器时序违规

- **Timing Violation Section**

Note: If your design will only ever run at typical room temperatures, selecting the narrower temperatur
range in the system DWR for your application helps the tool to find timing-compliant routing solutions.

Violation	Source Clock	Destination Clock	Slack(ns)
Setup			
	CyBUS_CLK	Clock_1	-2.705

- **Clock Summary Section**

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	28.000 MHz	28.000 MHz	N/A	
Clock_1	CyMASTER_CLK	1.000 MHz	1.000 MHz	26.029 MHz	
CyBUS_CLK	CyMASTER_CLK	28.000 MHz	28.000 MHz	26.029 MHz	Frequency
CyPLL_OUT	CyPLL_OUT	28.000 MHz	28.000 MHz	N/A	

时钟汇总章节中更加详细介绍了违规的原因，该节显示总线时钟频率为 28 MHz 对组件来说已经过高。

第 27 页上的图 45 显示了违规的详细信息。一个 32 位计数器需要四个 UDB 数据路径被链接起来。因此，同步到总线时钟的输入引脚被路由到 UDB 组件中多个不同的目的时钟。对于用于这些目的其中一个时钟，路由延迟和其他延迟为 38.419 ns，该值大于 35.7143 ns 周期的总线时钟，因此会导致建立时间违规。改变 Clock_1 频率无助于解决这些问题。您需要降低总线时钟频率或将引脚 1 与 Clock_1 同步，该问题才能得到解决。

图 45. 32 位计数器的时序违规的详情

– Source Clock : CyBUS_CLK : Positive edge(Required Frequency 28 MHz)

– Destination Clock : Clock_1 : Positive edge(Required Frequency 1 MHz)

Path Delay Requirement : 35.7143ns(28 MHz)

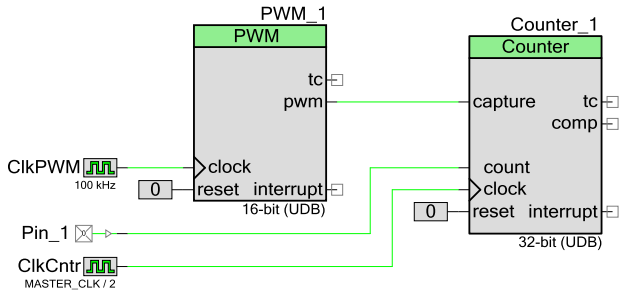
Affects clock : CyMASTER_CLK

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	26.029 MHz	38.419	-2.705	SETUP
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	28.483 MHz	35.109	0.605	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	28.484 MHz	35.108	0.606	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	30.431 MHz	32.861	2.853	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u1\/ci	31.448 MHz	31.799	3.915	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	31.449 MHz	31.798	3.916	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/cs_addr_1	33.805 MHz	29.581	6.133	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/cs_addr_1	33.807 MHz	29.580	6.134	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u0\/cs_addr_1	35.064 MHz	28.519	7.195	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u1\/cs_addr_1	35.066 MHz	28.518	7.196	

使用 STA 报告移除警告

作为最后一项操作，现在我们使用前面章节中介绍的方法并通过一个中等复杂的示例来了解和解决一系列的 STA 警告。首先，我们查看图 20 中显示的设计，图 46 中再一次显示了该设计，但在这里进行了一处更改：现在 ClkCntr 频率是主时钟经过 2 分频后得到的。

图 46.图 20 中的多时钟示例



请注意，在该设计中，两种时钟都与主时钟同步；特别将计数器时钟 ClkCntr 的频率大小设置为与主时钟进行二分频后得到的频率一样大。当主时钟频率足够高时（这种情况下为 60 MHz），可以检测到两个 STA 建立时间违规：

```
Warning-1366: Setup time violation found in
a path from clock (ClkCntr) to
clock(ClkCntr)

Warning-1366: Setup time violation found in
a path from clock (CyBUS_CLK) to clock
(ClkCntr)
```

图 47 显示在 STA 报告的时序违规节中这些相同的警告

图 47.图 46 的 STA 报告

Static Timing Analysis

Project : Fig19
Build Time : 07/23/13 09:27:41
Device : CY8C3866AXI-040
Temperature : -40C - 85/125C
Vdda : 5.00
Vddd : 5.00
Vio0 : 5.00
Vio1 : 5.00
Vio2 : 5.00
Vio3 : 5.00
Voltage : 5.0
Vusb : 5.00

Expand All | Collapse All | Show All Paths | Hide All Paths

- Timing Violation Section

Note: If your design will only ever run at typical room temperatures, selecting the narrower temperature range in the system DWR for your application helps the tool to find timing-compliant routing solutions.

Violation	Source Clock	Destination Clock	Slack(ns)
Setup			
	ClkCntr	ClkCntr	-2.533
	CyBUS_CLK	ClkCntr	-14.795

- Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	60.000 MHz	60.000 MHz	70.952 MHz	
ClkCntr	CyMASTER_CLK	30.000 MHz	30.000 MHz	27.882 MHz	Frequency
Clk_PWM	CyMASTER_CLK	100.000 kHz	100.000 kHz	42.411 MHz	
CyBUS_CLK	CyMASTER_CLK	60.000 MHz	60.000 MHz	31.784 MHz	Frequency
CyPLL_OUT	CyPLL_OUT	60.000 MHz	60.000 MHz	N/A	

要删除这些警告，首先应检查计数器时钟的建立违规情况，因此要检查源时钟和目的时钟是否都是 ClkCntr。要检查该操作，请点击**时序违规节**中的“ClkCntr 至 ClkCntr”一行。这样可以打开报告中的详细信息，如图 48 所示：

图 48. “ClkCntr 至 ClkCntr”的 STA 设置违规

- **Register to Register Section**

  - **Setup Subsection**

    - **Source Clock : ClkCntr : Positive edge(Required Frequency 30 MHz)**

      - **Destination Clock : ClkCntr : Positive edge(Required Frequency 30 MHz)**

Path Delay Requirement : 33.3333ns(30 MHz)

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
\Counter_1:CounterUDB:sC32:counterdp:u0\/z0	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	27.882 MHz	35.866	-2.533	SETUP
\Counter_1:CounterUDB:sC32:counterdp:u1\/z0	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	29.039 MHz	34.436	-1.103	SETUP
\Counter_1:CounterUDB:sC32:counterdp:u2\/z0	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	30.298 MHz	33.006	0.327	
\Counter_1:CounterUDB:sCTRLReg:SyncCtl:ctrlreg\/control_7	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	30.714 MHz	32.558	0.775	
\Counter_1:CounterUDB:sC32:counterdp:u0\/z0	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	30.716 MHz	32.556	0.777	
\Counter_1:CounterUDB:sC32:counterdp:u3\/z0_comb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	31.670 MHz	31.576	1.757	
\Counter_1:CounterUDB:sC32:counterdp:u0\/z0	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	31.742 MHz	31.504	1.829	
\Counter_1:CounterUDB:count_stored_i\/q	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	32.050 MHz	31.201	2.132	
\Counter_1:CounterUDB:sC32:counterdp:u1\/z0	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	32.127 MHz	31.126	2.207	
\Counter_1:CounterUDB:sC32:counterdp:u1\/z0	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	33.251 MHz	30.074	3.259	

因为计数器是 32 位，所以需要使用四个 UDB 数据路径来实现它。图 48 显示通过一些 UDB 的延迟对于频率为 30 MHz 的计数器时钟（ClkCntr）来讲太久了，我们必须要将 ClkCntr 的频率降低到 27.882 MHz 以下。因为 ClkCntr 是主时钟进行二分频后得到的，所以我们可以通过将主时钟频率降低为 55 MHz，这样便能够使 ClkCntr 的频率变为 27.5 MHz。

这样便解决了第一个警告，但仍存在其他设置违规：

```
Warning-1366: Setup time violation found in a path from clock (CyBUS_CLK) to clock (ClkCntr)
```

要想删除该警告，我们可以检查“CyBUS_CLK 到 ClkCntr”节的 STA 报告 — 点击**时序违规节**中的相关行。图 49 中显示了详细信息：

图 49. 主时钟频率为 55 MHz 时的 STA 设置违规

- **Register to Register Section**

  - **Setup Subsection**

    + **Source Clock : ClkCntr : Positive edge(Required Frequency 27.5 MHz)**

    + **Source Clock : Clk_PWM : Positive edge(Required Frequency 100 kHz)**

    - **Source Clock : CyBUS_CLK : Positive edge(Required Frequency 55 MHz)**

      - **Destination Clock : ClkCntr : Positive edge(Required Frequency 27.5 MHz)**

Path Delay Requirement : 18.1818ns(55 MHz)
Affects clock : CyMASTER_CLK

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	31.758 MHz	31.488	-13.306	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	35.489 MHz	28.178	-9.996	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	35.528 MHz	28.147	-9.965	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	39.005 MHz	25.638	-7.456	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u1\/ci	40.212 MHz	24.868	-6.686	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	40.263 MHz	24.837	-6.655	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u3\/cs_addr_1	44.431 MHz	22.507	-4.325	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u2\/cs_addr_1	44.727 MHz	22.358	-4.176	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u0\/cs_addr_1	46.322 MHz	21.588	-3.406	SETUP
Pin_1(0)_SYNC/out	\Counter_1:CounterUDB:sC32:counterdp:u1\/cs_addr_1	46.389 MHz	21.557	-3.375	SETUP

第一，请注意该图中的“路径延迟要求”是 55 MHz 的主时钟周期，不是 ClkCntr 的周期。第 14 页上的多时钟介绍了原因。路由到计数器 UDB 的延迟相对于 Pin_1 来讲太久。可以通过下面两种解决方案解决该主题：

1. 降低主时钟频率。如果您整个系统时序不需要高速主时钟，则这种方法是首先推荐的，因为它还能降低噪声和功耗。图 49 显示的是必须将主时钟频率调整为小于 FMax 列中的最小值（即 31.758 MHz）。但如果总线时钟频率低于 33 MHz 会导致 Pin_1 采用另一种同步方法 — 请参考第 11 页上的引脚章节 — 这样会导致不同的路由和另一个时序违规。图 50 显示实际上需要更大幅度降低主时钟频率，该频率要低于 26.036 MHz。

图 50. 主时钟频率为 31 MHz 时的设置违规

- Register to Register Section

- Setup Subsection

+ Source Clock : ClkCntr : Positive edge(Required Frequency 15.5 MHz)

+ Source Clock : Clk_PWM : Positive edge(Required Frequency 100 kHz)

- Source Clock : CyBUS_CLK : Positive edge(Required Frequency 31 MHz)

- Destination Clock : ClkCntr : Positive edge(Required Frequency 15.5 MHz)

Path Delay Requirement : 32.2581ns(31 MHz)
Affects clock : CyMASTER_CLK

Source	Destination	FMax	Delay (ns)	Slack (ns)	Violation
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	26.036 MHz	38.409	-6.151	SETUP
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	28.491 MHz	35.099	-2.841	SETUP
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	28.492 MHz	35.098	-2.840	SETUP
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/ci	30.440 MHz	32.851	-0.593	SETUP
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u1\/ci	31.457 MHz	31.789	0.469	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/ci	31.458 MHz	31.788	0.470	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u2\/cs_addr_1	33.817 MHz	29.571	2.687	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u3\/cs_addr_1	33.818 MHz	29.570	2.688	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u0\/cs_addr_1	35.077 MHz	28.509	3.749	
Pin_1(0)/fb	\Counter_1:CounterUDB:sC32:counterdp:u1\/cs_addr_1	35.078 MHz	28.508	3.750	

完成该操作（通过将 PLL 设置为 52 MHz，然后对它进行二分频而得到 26 MHz 主时钟）后，STA 不会产生任何警告，如图 51 所示：

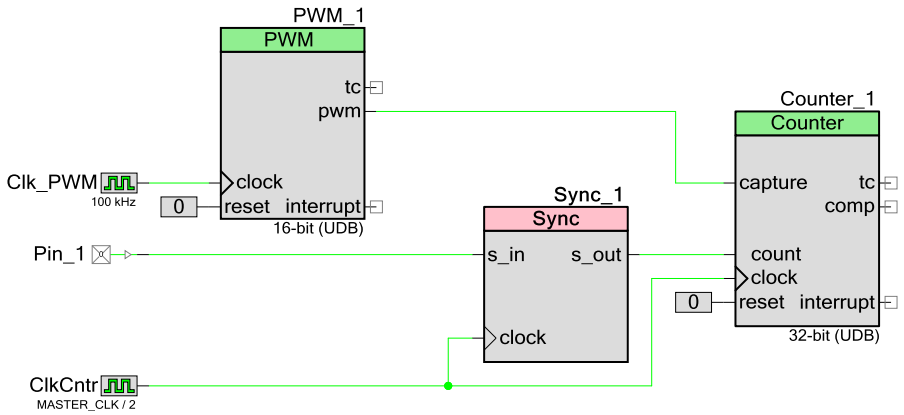
图 51. 23 MHz 主时钟频率的时钟汇总

- Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	26.000 MHz	26.000 MHz	76.225 MHz	
ClkCntr	CyMASTER_CLK	13.000 MHz	13.000 MHz	26.036 MHz	
Clk_PWM	CyMASTER_CLK	100.000 kHz	100.386 kHz	40.125 MHz	
CyBUS_CLK	CyMASTER_CLK	26.000 MHz	26.000 MHz	26.036 MHz	
CyPLL_OUT	CyPLL_OUT	52.000 MHz	52.000 MHz	N/A	

2. 您可以将主时钟频率调整为一个足够低的值，从而防止所有 STA 警告。如果您需要一个高速的主时钟或总线，最好通过添加一个 PSoC Creator 同步组件，实现 Pin_1 与 ClkCntr 之间同步。请参考图 52。

图 52.图 20 中已经显示的多时钟示例，但添加了一个同步组件



这样操作后生成的 STA 报告显示，现在的 ClkCntr 可以高达 28 MHz，如图 53 所示：

图 53. 主时钟频率为 26 MHz 并且添加了同步组件时的时钟汇总

- Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	26.000 MHz	26.000 MHz	77.149 MHz	
ClkCntr	CyMASTER_CLK	13.000 MHz	13.000 MHz	28.988 MHz	
Clk_PWM	CyMASTER_CLK	100.000 kHz	100.386 kHz	42.411 MHz	
CyBUS_CLK	CyMASTER_CLK	26.000 MHz	26.000 MHz	N/A	
CyPLL_OUT	CyPLL_OUT	52.000 MHz	52.000 MHz	N/A	

因为 ClkCntr 频率可增大到 28 MHz，所以主时钟频率可以高达 56 MHz，如 图 54 所示。

图 54. 主时钟频率为 56 MHz 且添加了同步组件时的时钟汇总

- Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	56.000 MHz	56.000 MHz	75.228 MHz	
ClkCntr	CyMASTER_CLK	28.000 MHz	28.000 MHz	28.988 MHz	
Clk_PWM	CyMASTER_CLK	100.000 kHz	100.000 kHz	42.411 MHz	
CyBUS_CLK	CyMASTER_CLK	56.000 MHz	56.000 MHz	N/A	
CyPLL_OUT	CyPLL_OUT	56.000 MHz	56.000 MHz	N/A	

另外，如果您需要一个最大速率的主时钟和总线时钟（高达 66 MHz），那么您可以将 ClkCntr 频率设置为主时钟经过三分频（而不是二分频）后的频率（即 22 MHz），该频率小于最大频率要求。

异步时钟交叉

STA 报告会引起另一种违规，即“异步”，如图 55 所示。这些“异步”违规通常是因为固定模块输出被路由到基于 UDB 组件的输入端引起的。处理这种异步情况最简单方法是在设计中添加同步组件，如主题#7 一节中所介绍的内容。

图 55. 异步时序违规

Static Timing Analysis

Project : Fig33
Build Time : 02/03/14 09:44:25
Device : CY8C3866AXI-040
Temperature : -40C - 85/125C
Vdda : 5.00
Vddd : 5.00
Vio0 : 5.00
Vio1 : 5.00
Vio2 : 5.00
Vio3 : 5.00
Voltage : 5.0
Vusb : 5.00

Expand All | Collapse All | Show All Paths | Hide All Paths

– Timing Violation Section

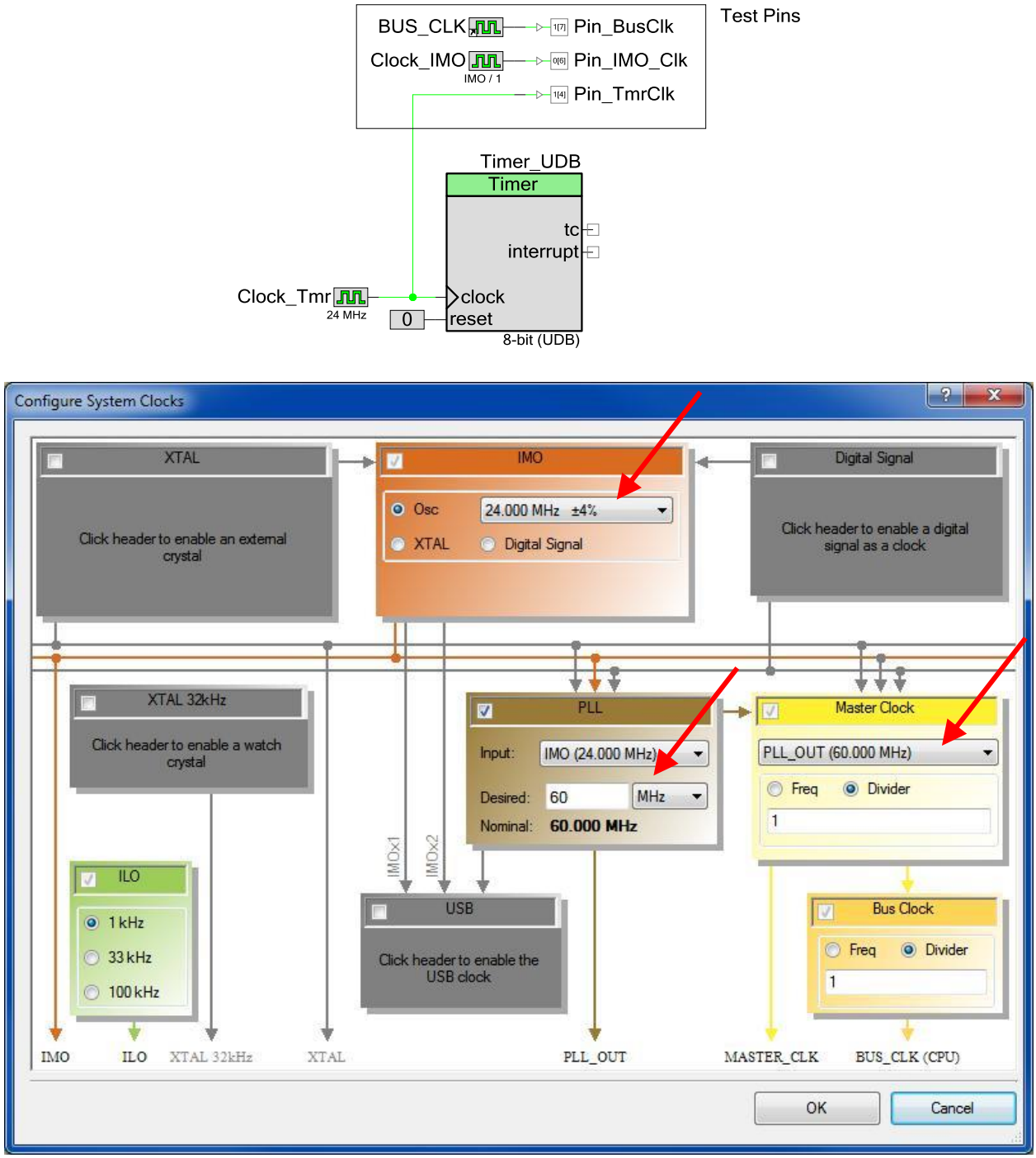
Note: If your design will only ever run at typical room temperatures, selecting the narrower temperature range in the system DWR for your application helps the tool to find timing-compliant routing solutions.

Violation	Source Clock	Destination Clock	Slack(ns)
Async			
	Clock_1(fixed-function)	Clock_1	
	Clock_2(fixed-function)	CyBUS_CLK	
	\ADC_DelSig_1:DSM4\/dec_clock	CyBUS_CLK	

时钟的额定频率和所需频率

查看前面各图，并注意：在 STA 时钟总结部分，所有时钟在相应的额定频率列和所需频率列中的值都相同。这是典型情况，但这些频率也可能存在差别。当主时钟不是另一个时钟的整数倍时，将发生这种情况，如图 56 所示。IMO 为 Clock_Tmr 提供时钟源，并且该主时钟的频率为 60 MHz；它的频率是定时器时钟频率的 2.5 倍。

图 56. 引起不同额定频率和所需频率的设计



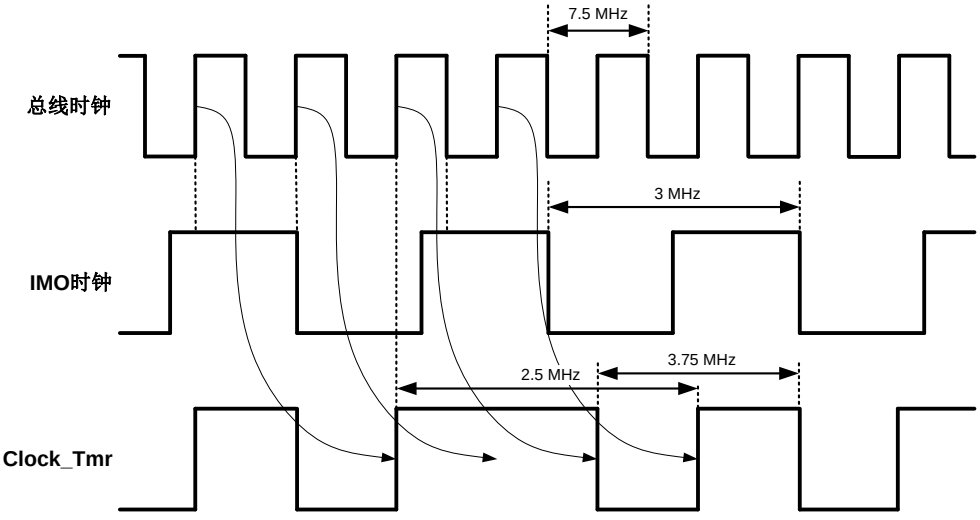
在这种设计中，频率得到降低时仍可以得到同样的效果，如第 33 页上图 56 中各测试引脚所示。使用一个示波器更容易印证这一点。图 57 显示的是一个类似的示例：其中， Clock_Tmr 时钟的频率为 3 MHz，主时钟频率为 7.5 MHz（60 MHz 经过 8 分频后得到）。

图 57. 不同的额定频率以及所需频率，示例#2

Type	Name	Domain	Desired Frequency	Nominal Frequency	Accuracy (%)	Tolerance (%)	Divider	Start on Reset	
System	USB_CLK	DIGITAL	48.000 MHz	? MHz	±0	-	1	☐	IMOx2
System	Digital Signal	DIGITAL	? MHz	? MHz	±0	-	0	☐	
System	XTAL 32kHz	DIGITAL	32.768 kHz	? MHz	±0	-	0	☐	
System	XTAL	DIGITAL	25.000 MHz	? MHz	±0	-	0	☐	
System	ILO	DIGITAL	? MHz	1.000 kHz	-50, +100	-	0	☑	
System	IMO	DIGITAL	3.000 MHz	3.000 MHz	±1	-	0	☑	
System	BUS_CLK (CPU)	DIGITAL	? MHz	7.500 MHz	±1	-	1	☑	MASTER_CLK
System	MASTER_CLK	DIGITAL	? MHz	7.500 MHz	±1	-	8	☑	PLL_OUT
System	PLL_OUT	DIGITAL	60.000 MHz	60.000 MHz	±1	-	0	☑	IMO
Local	Clock_IMO	DIGITAL	? MHz	3.000 MHz	±1	-	1	☑	IMO
Local	Clock_Tmr	DIGITAL	3.000 MHz	3.000 MHz	±1	-	1	☑	Auto: IMO
Local	Clock_UDB_1	DIGITAL	? MHz	7.500 MHz	±1	-	0	☑	BUS_CLK

在这两种情况下，定时器时钟都由 IMO 提供脉冲，但通常需要与 UDB 一起使用时，它必须与总线时钟同步。图 58 显示的是各个测试引脚上的时序框图；请注意，由于 Clock_Tmr 与总线时钟双重同步，因此它会被右移两个总线时钟周期。

图 58. 时序图，不同的额定频率和所需频率



由于 Clock_Tmr（其脉冲起源于 IMO）与总线时钟同步，而总线时钟不是该 IMO 的整数倍，因此，Clock_Tmr 的输出会有两个不同的频率：2.5 MHz 和 3.75 MHz — 这两个频率都不等于 IMO 的频率。该效应被称为**同步抖动**。

这些内容显示在 STA 时钟总结部分，如图 59 所示。请注意，要区分 PSoC Creator Clocks 选项卡（第 34 页上的图 57）中的术语“所需频率”和“额定频率”与 STA 报告（图 59）中的术语“额定频率”和“所需频率”。额定频率在这两种显示中是相同的。所需频率：是指使用该频率运行的路径必须满足时序的频率。它是时钟的最高频率 — 在这种情况下为 3.75 MHz。可能会发生一种情况：额定频率小于最大频率、所需频率高于最大频率，这样会引起没有明显解决方案的时序违规。

图 59. 显示不同额定频率和所需频率的 STA

+ Clock Summary Section

Clock	Domain	Nominal Frequency	Required Frequency	Maximum Frequency	Violation
Clock_IMO	Clock_IMO	3.000 MHz	3.000 MHz	N/A	
Clock_IMO (routed)	Clock_IMO (routed)	3.000 MHz	3.000 MHz	N/A	
Clock_Tmr (routed)	Clock_Tmr (routed)	3.000 MHz	3.000 MHz	N/A	
CyBUS_CLK (routed)	CyBUS_CLK (routed)	7.500 MHz	7.500 MHz	N/A	
CyILO	CyILO	1.000 kHz	1.000 kHz	N/A	
CyIMO	CyIMO	3.000 MHz	3.000 MHz	N/A	
CyMASTER_CLK	CyMASTER_CLK	7.500 MHz	7.500 MHz	N/A	
Clock_Tmr	CyMASTER_CLK	3.000 MHz	3.750 MHz	55.723 MHz	
CyBUS_CLK	CyMASTER_CLK	7.500 MHz	7.500 MHz	N/A	
CyPLL_OUT	CyPLL_OUT	60.000 MHz	60.000 MHz	N/A	

一般情况下，不推荐使用同步抖动来驱动定时器、计数器、PWM 或其他与时序有关的电路。构建完一个设计后，您需要检查 STA 时钟总结部分的内容，并确保不会发生这种情况。如果发生这种情况，可以使用下面两种方法进行处理：

- 调整主时钟和总线时钟的频率，使其等于其他时钟源频率的整数倍。最佳实践：最少要将该频率置为其他时钟源频率的四倍 — 请参考第 14 页上讨论部分的内容。在上述情况中，可以将主时钟的频率更改为 12 MHz，即等于 Clock_Tmr 频率的四倍。
- 为该时钟选择一个脉冲源，它可以是主时钟或与主时钟同步的其他时钟。这时，Clock_Tmr 的脉冲来自主时钟，并非 IMO。请注意，这样会引起您的时钟频率发生改变，从而需要改变您设计中其他部分。

保持、恢复和解决违规

前面的示例用于介绍设置违规，因为这类违规是目前最常见的。其他类型的违规（如第 6 页上描述的亚稳态和寄存器时序）会发生在下面的条件中：

- 通常保持违规与从 DSI 路由的时钟有关。因此，发生延迟或时滞时也会影响这两者，请参考图 23。这样会延迟时钟，并会引起一个保持时间违规；请参考图 7。
- 恢复和解决违规与异步复位或预设机制相关联；请参考图 30。如果将它们修改为同步，则通常可以删除警告。

总结

本应用笔记显示了数字设计中的某些普遍原则，并介绍了如何将它应用到 PSoC 3、PSoC 4 和 PSoC 5LP 器件中。它还展示了一套最佳实践的数字设计主题。

最后部分提供了 STA 警告的详细说明以及如何删除这些警告的示例。

使用本应用笔记中介绍的信息，您可以在 PSoC 器件中实现各种稳健、高性能的数字设计。Verilog、数据路径编程和组件开发的详细说明超出了本应用笔记的范围。更多有关这些主题的信息，请参考 PSoC Creator 软件中所包含的组件开发套件工具以及附件文档。另请参考 PSoC Creator 设计指南。

相关应用笔记

- AN82250 — 实现可编程逻辑 — 介绍内容
- AN82156 — PSoC 3 和 PSoC 5LP 使用 UDB Datapath（数据路径）设计 PSoC Creator 组件
- AN54181 — PSoC 3 入门
- AN79953 — PSoC 4 入门
- AN77759 — PSoC 5LP 入门
- AN60631 — PSoC 3 和 PSoC 5LP 时钟资源
- AN72382 — 使用 PSoC 3 和 PSoC 5LP GPIO 引脚
- AN62510 — 使用 PSoC 3 和 PSoC 5LP 实现状态机功能

关于作者

姓名： Todd Dust

职务： 高级应用工程师

背景： 西雅图太平洋大学电气工程学士学位

姓名： Mark Ainsworth

职务： 首席应用工程师

背景： 雪城大学计算机工程学士学位，华盛顿大学电子工程硕士学位

文档修订记录

文档标题：AN81623 — PSoC® 3、PSoC 4 和 PSoC 5LP 数字设计最佳实践

文档编号： 001-97876

版本	ECN	变更者	提交日期	变更说明
**	4802435	SNYQ	06/23/2015	本文档版本号为 Rev**，译自英文版 001-81623 Rev *D。
*A	5791936	XITO	07/04/2017	本文档版本号为 Rev*A，译自英文版 001-81623 Rev *E。

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。如果想要查找离您最近的办事处，请访问赛普拉斯所在地。

产品

ARM® Cortex® 微控制器	cypress.com/arm
汽车级产品	cypress.com/automotive
时钟与缓冲器	cypress.com/clocks
接口	cypress.com/interface
物联网	cypress.com/iot
存储器	cypress.com/memory
微控制器	cypress.com/mcu
PSoC	cypress.com/psoc
电源管理 IC	cypress.com/pmic
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线连接	cypress.com/wireless

PSoC®解决方案

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

赛普拉斯开发者社区

论坛 | WICED IoT 论坛| 项目 | 视频 | 博客 | 培训 | 组件

技术支持

cypress.com/support

PSoC 是赛普拉斯半导体公司的注册商标且 PSoC Creator 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。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© 赛普拉斯半导体公司，2012-2017 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。