



摘要

本应用手册介绍了与 Sitara™ AM263x MCU 配合使用的双时钟比较器 (DCC) 计算工具的操作。此工具提供了要编程的预计算寄存器值，以便通过与基准时钟进行比较来检查所需的任何系统/外设时钟频率是否在特定精度百分比内，从而实现与应用代码的轻松集成。

可从以下 URL 下载本文档所述的电子表格：<https://www.ti.com/cn/lit/zip/sprad69>。

内容

1 引言.....	2
2 使用该工具的步骤.....	3
3 参考文献.....	4

商标

Sitara™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

1 引言

双时钟比较器 (DCC) 主要用于确定应用执行期间的时钟精度。精确的精度是可编程的，应针对每个应用进行计算。它还可用于确定任何可选时钟的频率，并使用另一个独立输入时钟作为参考。它使用两个独立的时钟源来检测其中一个时钟是否超出规格。

时钟频率漂移是不可避免的，这可能是由于湿度、压力和温度、老化等环境条件等多种原因造成的。当器件在现场使用几年后，就会注意到老化的影响，如果不加以监控，时钟就会开始漂移，从而导致灾难性的后果。因此，DCC 在安全关键型汽车/工业定时应用中特别有用。

DCC 包含三个计数器—counter0 (20 位)、valid0 (16 位) 和 counter1 (20 位)。最初，所有计数器都加载了用户定义的预加载值。一旦 DCC 被启用，counter0 和 counter1 即开始递减，其速率分别由 clock0 和 clock1 的频率决定。当 counter0 等于 0 (到期) 时，valid0 计数器会以 clock0 确定的速率递减。如果 counter1 在有效窗口中递减到 0，则不会产生错误，并且 clock1 被视为在用户配置的允许容差范围内良好。当频率超出允许的公差范围，counter1 不在有效窗口内递减 (clock0/clock1 不存在，clock1 在 counter0 达到 0 之前到期，或 clock1 在 counter0 和 valid0 都达到 0 后到期) 时，就会产生错误。

counter0 和 counter1 是根据 clock0 和 clock1 的频率比配置的 ($F_{clk1} \times Counter0 = F_{clk0} \times Counter1$)。valid0 计数器提供容差，并根据允许的频率误差和固有 DCC 误差 (异步和数字化误差) 进行配置。由于 clock0 和 clock1 是异步的，因此计数器的启动和停止不会同步发生。因此，在配置计数器时，必须考虑两种不同的误差来源，它们是：

- 由于 clock0 和 clock1 的异步时序导致的 DCC 误差 - 这取决于 clock0 和 clock1 的频率：
 - 如果 $F_{clk1} > F_{clk0}$ ，则为异步。误差 (以 Clock0 周期为单位) = $2 + 2 \times (F_{sysclk} / F_{clk0})$
 - 如果 $F_{clk1} < F_{clk0}$ ，则为异步。误差 (以 Clock0 周期为单位) = $2 \times (F_{clk0} / F_{clk1}) + 2 \times (F_{sysclk} / F_{clk0})$
 - 如果 F_{clk1} 未知，则为异步。误差 (以 Clock0 周期为单位) = $2 + 2 \times (F_{sysclk} / F_{clk0})$ 注意：Fsysclk 为 200MHz
- 数字化错误 - 8 个 Clock0 周期

该工具使用以下公式计算 counter0、valid0 和 counter1 种子值：

- counter0 种子 = 窗口 - 总误差
- Valid0 种子 = 2 倍总误差
- Counter1 种子 = 窗口 $\times (F_{clk1} / F_{clk0})$

其中，

- DCC 误差 (以 Clock0 周期为单位) = 异步。误差 + 8 个 Clock0 周期 (数字化误差)
- 可能的最小精度 (%) = $(100 \times \text{DCC 误差} \times (F_{clk1} / F_{clk0})) / 1048575$
- 窗口 (以 Clock0 周期为单位) = $(\text{DCC 误差}) / (0.01 \times \text{可能的最小精度 (以 \% 为单位) })$
- 允许的频率误差 (以 Clock0 周期为单位) = 窗口 $\times (\text{可能的最小精度 (以 \% 为单位) } / 100)$
- 总误差 (以 Clock0 周期为单位) = DCC 误差 + 允许的频率误差

DCC 计算工具提供了要编程的寄存器值，以便将任何给定的系统或外设时钟与可配置的参考时钟进行比较，从而确定所述时钟的频率是否在预期精度范围内。所需的精度可作为输入提供，该工具会据此计算计数器种子值。以下四个实例中的每个实例都有一个单独的计算器：DCC0、DCC1、DCC2 和 DCC3。

2 使用该工具的步骤

通过工具或器件技术参考手册，确定可用于监控所需时钟源的 DCC 实例。

1. 在 (1) 输入部分中：

- a. 在所使用 DCC 实例部分的 EXT_REFCLK 频率输入字段中输入作为外部参考时钟提供的时钟频率。
- b. 使用提供的下拉菜单输入要用作比较参考时钟的时钟源 DCCCLKSSRC0 (参考时钟)。可用的选项有 EXT_REFCLK、RCCLK10M (10MHz)、RCCLK32K (32KHz) 和 XTALCLK (25MHz)。
- c. 使用提供的下拉菜单输入要用于验证的时钟源 DCCCLKSSRC1 (待验证的时钟)。可用的选项包括 EXT_REFCLK、R5SS0_CLK (400MHz)、R5SS1_CLK (400MHz)、RCCLK32K (32KHz)、SYSCLK (200MHz) 和 XTALCLK (25MHz)。
- d. 输入您的应用所需的时钟精度 %。如果未提供输入，该工具将使用默认值 0.2 或可能的最小精度 (以较高者为准)。
- e. 该工具根据作为输入提供的时钟源计算异步误差、DCC 误差、可能的最低精度、窗口、允许的频率误差、总误差 (上一节中定义的公式)。

备注

最小可能精度不能大于 48 (超过该值计数器种子值变为零)，在这种时钟源组合中，系统会显示“ERROR”。

2. 从 (2) 寄存器配置部分：

- a. 使用前一步计算的 Counter0 种子值的十六进制来对 DCCCNTSEED0 (@0x00000008) 进行编程。
- b. 使用前一步计算的 Counter1 种子值的十六进制来对 DCCCNTSEED1 (@0x00000010) 进行编程。
- c. 使用前一步计算的 Valid0 种子值的十六进制来对 DCCVALIDSEED0 (@0x0000000C) 进行编程。
- d. 使用相应的 (A000 + 所选时钟源 0 的十六进制索引) 对 DCCCLKSRC0 (@0x00000028) 进行编程。
- e. 使用相应的 (A000 + 所选时钟源 1 的十六进制索引) 对 DCCCLKSRC1 (@0x00000024) 进行编程。
- f. 完成上述所有配置后，使用 0x0000AAAA 设置 DCCGCTRL (@0x00000000)，以在 SINGLE_SHOT 模式下设置比较，启用 DONE 信号，启用误差信号并开始比较。

备注

对于连续模式，将位 11:8 配置为 0101。

- g. 在 DCC 完成阶段结束时，如果时钟处于所配置的容差范围内，DCC 会生成到 R5F 的 DCC_done。寄存器 DCCSTAT(@0x00000014) 中的 Done 位 (位 1) 将被设定为 1 来表示完成。

如果 DCC 在频率测量中检测到错误，它会生成到 ESM 的 DCCx_error (错误信令模块)，而不是生成到 R5F 的 DCC_done 中断。寄存器 DCCSTAT(@0x00000014) 中的 Error 状态 (位 0) 将被设定为 1 来表示错误条件。在这种情况下，ESM 可以配置为生成到 R5F 的中断或器件错误引脚，以采取进一步操作。有关配置和设置中断的详细信息，请参阅器件技术参考手册。

如果未使用中断模式，配置 SINGLE_SHOT 模式后，R5F 可配置为持续轮询 DCCSTAT(@0x00000014) 寄存器中的 Error 状态 (位 0) 和 Done 位 (位 1)。

3. 如果出现错误，可以读取 counter0、counter1 和 valid0 值（分别从地址偏移 0x00000018、0x00000020 和 0x0000001C），以确定错误类型（clock0/clock1 不存在，clock1 在 counter0 达到 0 之前到期，或 clock1 在 counter0 和 valid0 都达到 0 后到期）。

以上示例适用于 DCC0。按照类似的过程使用任何其他 DCC 实例 DCC1、DCC2 和 DCC3。

3 参考文献

- 德州仪器 (TI) : [AM263x Sitara™ 微控制器数据表](#)
- 德州仪器 (TI) : [AM263x Sitara™ 微控制器技术参考手册](#)
- 德州仪器 (TI) : [AM263x Sitara 处理器技术参考手册寄存器附录](#)
- AM263x MCU Plus SDK 自述文件指南 : <https://www.ti.com/tool/MCU-PLUS-SDK-AM263X>

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司