Application Note

C2000™ 器件中的 CRC 引擎

Veena Kamath, Prasanth Viswanathan Pillai, Sira Rao, and Baskaran Chidambaram

摘要

本应用报告将介绍 C2000 器件系列中提供的各种 CRC 模块。文中提供了软件示例参考，并重点说明了 CRC 模块之间的差异。此外，节 9 还讨论了为特定用例选择 CRC 模块的相关指导。

内容

表格清单

商标

1 引言

循环冗余校验 (CRC) 是一种用于通信网络和数据存储的错误检测机制。节 11 中讨论的 C2000 系列器件包含许多 CRC 模块（节 10 中讨论了不同的属性），这些模块提供了相关选项，让终端应用程序构建器为应用程序选择合适的引擎。此外，在 CLB、FSI、USB、CAN、EMAC、EtherCAT 等本文档未讨论的 C2000 器件中存在 CRC 引擎。如果数据和相应的 CRC 一起修改，该机制无法检测到这一情况。

1.1 首字母缩写词

表 1-1. 首字母缩写词

首字母缩写词	全称
BGCRC	后台循环冗余校验
CAN	控制器局域网
CLA	控制律加速器（F28x 系列器件的协处理器）
CLB	可配置逻辑块
CM	连接管理器
CRC	循环冗余校验
ECC	错误校正码
EMAC	以太网介质访问控制器
ERAD	嵌入式实时分析和诊断单元
FSI	快速串行接口
PAB	程序地址总线
GCRC	通用循环冗余校验
PROMCRC	程序 ROM CRC
PSA	并行签名分析
VCU	Viterbi、复杂数学和 CRC 单元

2 BGCRC

后台循环冗余校验 (BGCRC) 引擎可用于计算存储器区域的 CRC，而无需使用 CPU 或 CLA。然后可以使用 CRC 计算结果来检测存储器损坏。BGCRC 只需配置一次即可使用。BGCRC 在计算 CRC 后与配置的黄金值进行比较，然后可以在发现有任何错误时触发 NMI 或 CLA 任务。

BGCRC 模块能够作为后台进程从存储器中连续读取数据，并在应用程序运行时直接在后台工作。读取发生在空闲时间（此时 CPU、CLA、DMA 等其他主器件均未访问存储块），因此功能访问不受影响。该模块还在读取存储器时执行 ECC/奇偶校验。读取期间发生的任何 ECC 或奇偶校验错误都将通过设置相应的 NMI 标志并生成一个中断（如果已配置）来指示。

该模块还提供了一个用于在配置后锁定和提交寄存器值的选项。CRC 计算时间也可以使用内部看门狗进行监控。引擎需要一个周期来计算每个 32 位字的 CRC。但是，由于 BGCRC 只在空闲时间内工作，因此一个存储块所用的总时间可能因 CPU/DMA/CLA 存储器对配置的存储块的访问情况而异。

该器件在 CPU1 和 CPU2 子系统中都有用于 CPU 和 CLA 的独立 BGCRC 实例。

更多有关计算时间和该模块中其他特性的详细信息，请参阅 *TMS320F2838x 微控制器技术参考手册*。

如需获取软件支持，请参阅 C2000ware：

- driverlib 可参考 <C2000Ware 安装文件夹>\driverlib\<器件>\driverlib
- 示例可参考 <C2000Ware 安装文件夹>\driverlib\<器件>\examples\c28x\bgcrc

3 GCRC

通用循环冗余校验 (GCRC) 仅适用于 F2838x 的连接管理器 (Cortex-M) 内核。此 CRC 引擎提供了用于计算 CRC 的自定义多项式选项。它能够计算字节、半字或字数据的 CRC。配置该模块后，CPU/DMA 应向引擎提供数据。

其他特性包括：

- 能够定义数据的字节序和源数据的数据类型
- 能够反转位顺序
- 能够选择参与 CRC 计算的数据位

该模块还支持固定多项式路径，其中 CRC 配置按照以下值固定。此固定数据路径将在单个周期内计算给定数据集的 CRC。

- 多项式：0x04C11DB7
- 字节序：小端
- 位反转：否
- 数据类型：32 位
- 数据掩码：无

对于 n 个数据字节，GCRC 需要 2n+2 个周期（针对 0x04c11db7 以外的多项式）。

更多有关该模块的详细信息，请参阅 *TMS320F2838x 微控制器技术参考手册*。

如需获取软件支持，请参阅 C2000ware：

- driverlib 可参考 <C2000Ware 安装文件夹>\driverlib\f2838x\driverlib_cm
- 示例可参考 <C2000Ware 安装文件夹>\driverlib\f2838x\examples\cm\gcrc

4 VCU CRC

除了 Viterbi 和复杂数学运算之外，C2000 器件上的 VCU 模块还执行 CRC 计算。该模块提供特殊指令来加速 CRC 计算，如果不使用该模块，在 C28x CPU 上可能需要几个周期进行此计算。C2000 器件上存在 3 种不同类型的 VCU 模块。要确定特定器件上可用的特定 VCU 模块（如果有），请参阅 *C2000 实时控制外设参考指南*。

- VCU Type0 或 Type1（以不同形式表示为 VCU0、VCU-I），它们完全相同。
- VCU Type2（以不同形式表示为 VCU2、VCU-II）。这是 VCU0 的较新版本。
- VCRC - 这是模块的最新版本，仅包含 CRC 功能。删除了 Viterbi 和复杂数学功能。

该模块支持 8 位、16 位、24 位（VCU0 除外）或 32 位 CRC 的计算。VCRC 模块支持用户可配置的多项式，该多项式的值和大小（1 至 32 位）都很灵活。该模块还支持用户可配置的数据大小（1 至 8 位）。

表 4-1. 不同 VCU 模块的特性

	8 位 CRC 多项式	16 位 CRC 多项式	24 位 CRC 多项式	32 位 CRC 多项式	可配置数据和多项式
VCU0	0x07	0x8005 0x1021	不可用	0x4C11DB7	不可用
VCU2	0x07	0x8005 0x1021	0x5D6DCB	0x4C11DB7 0x1EDC6F41	不可用
VCRC	0x07	0x8005 0x1021	0x5D6DCB	0x4C11DB7 0x1EDC6F41	1-8 位数据 1-32 位多项式

VCU0/VCU2 支持固定多项式（如表 4-1 所示），这些多项式的错误检测功能如下：

- 可以检测消息（任意长度）中的所有 single-bit 和 double-bit 错误
- 可以检测任何具有奇数个错误的错误模式
- 可以检测长度不超过 CRC 长度的所有突发错误，以及所有较长突发的一部分 $(1 - 2^{-n})$

更多信息，请参阅 *TMS320C28x 扩展指令集技术参考手册*。

上述功能为用户提供了多种选择来满足各种应用要求。可以对来自 ROM、RAM 或闪存的数据执行 CRC 计算。VCU 还支持位顺序，可以对“按原样”从存储器获取的数据或翻转的数据计算 CRC，后者称为“反射 CRC”。

C2000Ware 包含用于 VCU 并经过汇编优化的软件库，以及一些通过 C 语言调用的汇编函数演示软件库用法的示例。此外，为了进行比较，可以使用查找表方法（用 C 语言编写）计算 CRC，并获取链接器生成的 CRC（在链接时；这是 C2000 代码生成工具提供的功能）。

这些库支持偶校验或奇校验（字节序）。对于偶校验，CRC 输入计算从存储器中的低字节开始，而对于奇校验，则是从存储器中的高字节开始。

如需获取软件支持，请参阅 C2000ware：

- VCU0 - 可在 C2000Ware_X_XX_XX_XX\libraries\dsp\VCU\c28\source\vcu0\crc 中找到 CRC C 语言调用的汇编实现。
- VCU2 - 可在 C2000Ware_X_XX_XX_XX\libraries\dsp\VCU\c28\source\vcu2\crc 中找到 CRC C 语言调用的汇编实现。
- 示例可在以下位置找到：C2000Ware_X_XX_XX_XX\libraries\dsp\VCU\c28\examples\crc

5 ERAD CRC

ERAD 模块中的 CRC 单元会监控 CPU 总线，并在执行自检代码时计算 CRC。CRC 单元的主要目的是确保 CPU 在多次迭代中执行相同的软件测试库时保持功能不变。这些 CRC 单元会监控不同的 CPU 接口，不能用于程序或数据存储器的 CRC 计算。

更多详细信息，请参阅 *TMS320F28004x 微控制器技术参考手册* 和 *TMS320F2838x 微控制器技术参考手册*。

6 CLA CRC (PSA)

6.1 CLA PSA

并行签名分析 (PSA) 可确保控制律加速器 (CLA) 上代码执行的完整性。CLA-PSA 逻辑可用于计算 CLA 的程序地址总线 (PAB) 和数据写入数据总线 (DWDB) 的签名。黄金签名可以离线计算得出，也可以在应用程序的初始化阶段计算得出。随后可以将该计算结果与执行期间计算的签名进行比较，以确保执行代码的正确性。

6.1.1 适用于 PAB 的 PSA

交叉检查适用于 PAB 的 PSA 将确保代码执行的正确顺序。使用适用于 PAB 的 PSA 可以检测由故障导致的意外代码分支。这是使用多项式 $1 + x + x^2 + x^{22} + x^{32}$ 计算得出的。

6.1.2 适用于 DWDB 的 PSA

适用于 DWDB 的 PSA 可用于确保写入数据的完整性。如果需要计算一组存储器位置的 PSA (用于定期检查配置寄存器、静态存储器内容等的完整性) ，则需要首先将其读取到 CLA，然后写入到虚拟位置。适用于 DWDB 的 PSA 允许使用 MPSACTL.MPSA2CFG 配置来配置多项式。

- 00 – PSA ($1 + x + x^2 + x^{22} + x^{32}$)
- 01 – CRC32 ($1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$)

6.1.3 计算 PSA 时的注意事项

- 建议在事件模式下计算 PSA (MPSACTL.MPABCYC = 1 且 MPSACTL.MDWDBCYC = 1) 。在周期模式下计算 PSA 对存储器访问有很强的依赖性，而这种依赖性在实际应用中可能难以确保。
- 应在 PSA2 停止时执行多项式配置。

7 CLA-PROMCRC – CLA 程序完整性检查

CLAPROMCRC 模块计算 CLA 程序 ROM 中的可配置数据块的 CRC-32 值。当 CLA 未访问 CLA 程序总线上的 ROM 时，CLAPROMCRC 以非侵入方式计算 CRC-32。CPU 和 CLA 都没有能力计算 CLA 程序 ROM 的 CRC，这个限制由 CLAPROMCRC 单元处理。但是，CLAPROMCRC 不能用于任何其他存储块。BGCRC 引擎能够对 CLA 程序 ROM 执行 CRC。因此，在具有 BRCRC 的器件中可以使用该引擎。CLAPROMCRC 模块可用于不支持 BGCRC 的器件。

如需获取软件支持，请参阅 C2000ware：<C2000Ware>\driverlib\<器件>\examples\clapromcrc。

更多详细信息，请参阅 *TMS320F28004x 微控制器技术参考手册*。

8 使用软件计算 CRC

<C2000ware>\libraries\dsp\VCU\c28\source\common\c\crc 中提供了一些示例 C 函数 (用于 C28x) ，这些函数使用查找表计算 CRC8、CRC16 和 CRC32。在这些示例中，计算 CRC 之前会生成大小为 512 字节的查找表。此方法显著缩短了 CRC 计算时间，但需要额外的存储器。在编译器优化级别 -o2，这些函数每个 32 位字需要大约 50 个周期。

如需获取软件支持，请参阅 C2000ware：<C2000ware>\device_support\<器件>\examples\cpu1\cla_crc8。

9 针对具体用例的 CRC 建议

表 9-1 中提供了不同的用例以及哪种 CRC 类型有助于满足要求。功能安全标准 ISO 26262 和 IEC 61508 要求对各种硬件单元进行完整性检查。器件中实现的不同 CRC 块有助于满足其中的一些要求。

表 9-1. 用例和推荐的 CRC 模块

否	用例	推荐(1) 使用的 CRC 类型
1	为了确保代码执行的完整性（例如，检测代码中的意外分支，由于故障而导致的错误程序数据），请对程序总线使用 CRC。根据获取的程序数据计算出的 CRC 将适用于获取单元、地址生成单元和存储器互连。	1.对于 C28x，使用 ERAD CRC 2.对于 CLA，使用 CLA-PSA
2	配置寄存器包含决定器件行为的重要内容。需要定期检查配置寄存器的完整性。CRC 模块可用于更快地完成此测试。	1.对于 C28x，使用 ERAD CRC 2.对于 C28x（针对不支持 ERAD CRC 的器件），使用 VCUCRC 3.对于 CLA，使用 CLA PSA 4.对于具有固定多项式 (0x04C11DB7) 的 CM 子系统，使用 GCRC
3	与配置寄存器类似，需要检查静态 SRAM/ROM 内容的完整性，以确保它们没有错误。	1.对于 C28x 和 CLA，使用 BGCRC 2.对于 C28x（针对不支持 ERAD CRC 的器件），使用 VCUCRC 3.对于 CM 子系统（多项式 - 0x04C11DB7），使用 GCRC 4.CLA PROMCRC 可用于不可使用 BGCRC 的器件。
4	与配置寄存器和 SRAM/ROM 存储器类似，需要检查静态闪存内容的完整性。	1.对于 C28x，使用 ERAD CRC 2.对于 CM 子系统（多项式：0x04C11DB7），使用 GCRC 3.对于 C28x（针对不支持 ERAD CRC 的器件），使用 VCUCRC
5	很多时候，用户可能需要使用硬件不支持的多项式来计算 CRC。VCUCRC 和 GCRC 模块提供的可配置 CRC 支持有助于以高效的方式实现这一点。	1.对于 C28x，使用 VCUCRC（配置相应的多项式） 2.如果多项式在 CLA-PSA 中可用，则可以使用多项式。否则，需要使用软件 3.配置了适当多项式的 GCRC
6	通信接口以反向通道安全方法来实现端到端安全，以确保传输数据的完整性。器件中的硬件加速器可用于实现此目的并降低对 CPU MIPS 的影响。	1.对于 C28x，使用 VCUCRC 2. 对于 CLA，使用基于软件的计算方法 3.对于 CM 子系统，使用 GCRC

(1) 其他 CRC 模块可酌情使用，此处提到的选项只是建议的选项。

10 CRC 模块比较

表 10-1. CRC 引擎的属性

CRC 引擎的属性	BGCRC	GCRC	VCU CRC	ERAD CRC	CLA PSA
可访问的内核	C28x、CLA	CM	C28x	C28x	CLA
使用的 CRC 多项式	固定值 = 0x04C11DB7	用户可编程（高达 32 位的多项式）	• 8 位 - 0x07 • 16 位 - CRC16 802.15.4、0x8005、CRC-CCITT、0x1021 • 24 位 - 0x5d6dcb • 32 位 - CCITT-32、0x04C11DB7、0x1EDC6F41 VCRC 支持用户可配置的数据大小和可配置的多项式（值和大小（高达 32 位））	固定多项式	PAB：1 + x + x^2 + x^{22} + x^{32} (0x00400007) DWDB：可配置多项式 PSA 模式：1 + x + x2 + x22 + x32 (0x00400007) CRC32 模式：1 + x + x2 + x4 + x5 + x7 + x8 + x10 + x11 + x12 + x16 + x22 + x23 + x26 + x^{32} (0x04C11DB7) CRC-16-IBM：1 + x2 + x15 + x16 (0x8005) CRC16-CCITT：1 + x5 + x12 + x16 (0x1021)
种子值	用户可编程	用户可编程	用户可编程	用户可编程	用户可编程
输入大小	256 字节的倍数 最小：256 字节 最大：256KB	任意大小	任意大小	-	任意大小
反转位顺序（反射输入）	否	是	是	否	否
用于 CRC 计算的周期数	每个 32 位字 1 个周期	固定多项式：每个 32 位字 1 个周期 其他：n 字节数据为 2n+2 个周期	固定多项式：每字节 1 个周期 可配置的多项式：1-8 位为 3 个周期	1	1
数据访问模式	直接访问存储器	CPU/DMA 需要在 GCRC 中将数据逐一写入特定寄存器	直接访问存储器	监控总线访问	监控总线访问

11 CRC 引擎与器件映射表

表 11-1. CRC 引擎与器件映射表

器件/ CRC 引擎	BGCRC	GCRC	VCU CRC	CLA PSA	CLAPROMCRC	ERAD CRC
F2837xD/F2837xS/ F2807x	不适用	不适用	C28x (VCU2)	不适用	不适用	不适用
F28004x	不适用	不适用	C28x (VCU0)	CLA	CLA	不适用
F2838x	C28x、CLA	CM	C28x (VCRC)	CLA	不适用	C28x
F28002x	C28x	不适用	C28x (VCRC)	不适用	不适用	C28x

12 参考文献

- *TMS320F2838x 微控制器技术参考手册*
- *TMS320C28x 扩展指令集技术参考手册*
- *TMS320F28004x 微控制器技术参考手册*
- *C2000 实时控制外设参考指南*

13 修订历史记录

Changes from Revision * (January 2020) to Revision A (May 2023)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式	1

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出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