

STM32 CRC 使用详解

关键字: CRC, 生成多项式, 模 2 除法

1. 概述

在 STM32 众多外设中, CRC 是一个不起眼的小外设, 以至于在使用的时候会忽略掉。最近在学习 Class B 的时候, CRC 是 Class B 应用需要用到外设, 用来生成 ROM 校验码, 在阅读 Class B 相关文档 CRC 的部分的时候, 我出现了理解上的一些错误, 由此可见, CRC 外设虽小, 理解其细节, 也是需要时间的, 本文的目的是通过理解 CRC 外设的使用, 用文字的方式记录下来, 给后来使用者提供参考。

2. 问题描述

在阅读 RM0440 Reference Manual 文档中 CRC 章节的时候, 文中提到了对输入数据和输出数据进行 bit-reversal 的操作, 如下图所示, 开始以为是按位取反的操作, 但是按照文中给的例子, 按位取反的结果与实例不一致, 才觉得有必要深入理解一下。带着问题, 本文对 CRC 校验用到的概念以及校验方法进行较为详细的介绍。

图1. Bit-Reversal

RM0440 Cyclic redundancy check calculation unit (CRC)

The input data can be reversed, to manage the various endianness schemes. The reversing operation can be performed on 8 bits, 16 bits and 32 bits depending on the REV_IN[1:0] bits in the CRC_CR register.

For example: input data 0x1A2B3C4D is used for CRC calculation as:

- 0x58D43CB2 with bit-reversal done by byte
- 0xD458B23C with bit-reversal done by half-word
- 0xB23CD458 with bit-reversal done on the full word

按照我最初的理解, 输入数据 0x1A2B3C4D 按位取反的值都应该是 0xE5D4C3B2, 很显然, 这个理解是不对的。Bit-reversal 的正确理解是按位高低位反过来, 所以按照字节, 半字, 字为单元进行操作得到的数据是不一样的, 这个功能实际作用是解决不同硬件字节序的问题。

3. CRC 校验算法原理

在 CRC 校验会用到多项式和模 2 运算的概念, 这里先介绍一下什么是多项式和模 2 运算。

多项式是由多个单项式组成的代数式。任何一个二进制数，都可以与一个系数仅为 0 或者 1 的多项式对应。

比如二进制数 1010111 对应的多项式为 $X^6 + X^4 + X^2 + X^1 + 1$ 。

下面要谈的模 2 除法其实就是异或操作，简单来说就是同为 0，异为 1，不用进位也不用借位。不妨一起看看模 2 除法的一个具体实例。

假设被除数 $X=11110000B$ ，除数 $P=1101B$ ，那么进行模 2 除法的步骤和结果如下。

图2. 模 2 除法示例说明

1 0 1 1	//商

1 1 1 1 0 0 0	//被除数,注意首位为1
1 1 0 1	//被除数首位为1,除以除数

0 1 0 0 0 0	//余数去除首位,作为新的被除数
0 0 0 0	//被除数首位为0,除以0

1 0 0 0 0	//余数去除首位,作为新的被除数
1 1 0 1	//被除数首位为1,除以除数

1 0 1 0	//余数去除首位,作为新的被除数
1 1 0 1	//被除数首位为1,除以除数

1 1 1	//余数,此时余数位数少于除数,不能继续除了

CRC 校验的基本思想是基于二进制序列操作的.先选定固定项数的多项式作为除数 Y ，假设除数的最高项为 M ，则除数二进制位数为 $M+1$ ，原始数据 X 二进制位数为 N ，先把原始数据 X 左移 M 位，得到一个二进制位数为 $(N+M)$ 的数 $X2$ ， $X2$ 与除数 Y 进行模 2 除法运算，得到的余数 Z 就是 CRC 校验码，而 $X2$ 与余数 Z 进行模 2 加法得到数 $X3$ 为目的数据。所以数 $X3$ 与除数 Y 进行模 2 除法计算得到的余数肯定为 0. 接收方就根据余数是否为 0 来判断数据在传输过程中是否出错。

作为除数的多项式最高项是由 CRC 校验位数决定的，比如通常用到的 CRC8 的最高项为 8，CRC16 最高项为 16，CRC32 最高项为 32。

下面用一个实例解释 CRC 校验的过程。

假设原始为 10110011，使用 CRC4 校验，除数多项式为 $X^4 + X^3 + 1$ ，二进制表示为 11001。

第一步，除数的最大项为 4，所以原始数据左移 4 位得到的数是 101100110000。

第二步，进行模 2 除法运算，被除数位 101100110000，除数为 11001，得到的余数过程如下。

图3. CRC 校验码计算示例

$$\begin{array}{r}
 11010100 \\
 11001 \overline{) 101100110000} \\
 \underline{11001} \\
 11110 \\
 \underline{11001} \\
 11111 \\
 \underline{11001} \\
 11000 \\
 \underline{11001} \\
 0100
 \end{array}$$

第三步，根据第二步计算得到的 CRC 校验码为模 2 除法的余数 0100，添加到原始数据后面，得到的数据为 101100110100，这个数据就是发送给目标端的数据。

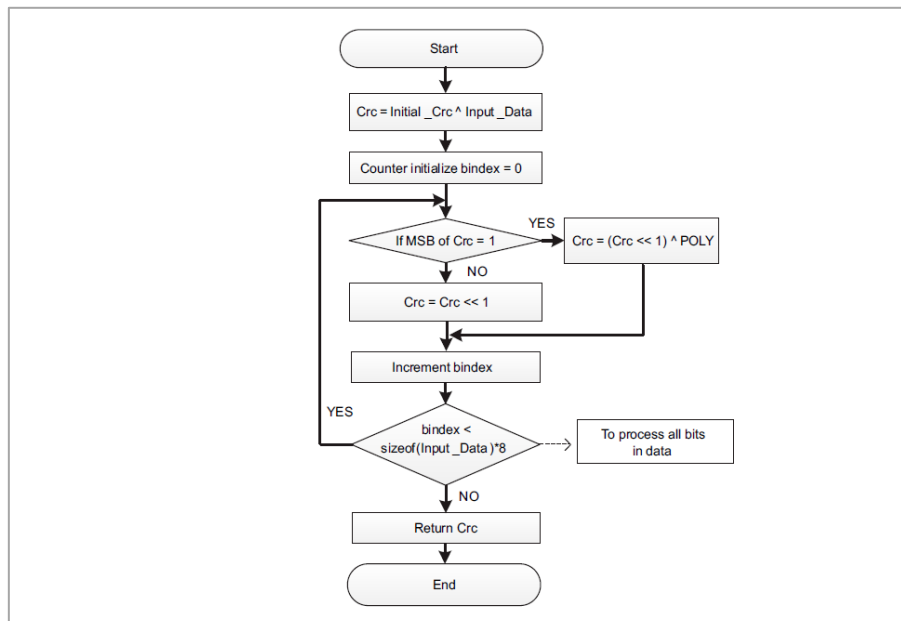
第四步，目标端对接收到的数据同样进行模 2 除法运算，如果得到的余数为 0，则表示数据没有错，否则表示数据有误。

4. STM32 中的 CRC 外设

通过上述内容理解了 CRC 校验算法的原理，就很容易理解如何 STM32 CRC 外设及其使用了。

图 4 是 STM32 CRC 外设对单个数据进行校验的过程，如果对一组数据进行校验，当前数据产生的校验码会作为下一个数据的初始校验码，循环执行图 4 的校验流程，最终得到的校验码为这一组数据的校验码。

图4. STM32 CRC algorithm flow



在 STM32 CRC 外设主要有 CRC_DR,CRC_IDR,CRC_CR,CRC_INIT,CRC_POL 共五个寄存器，其中 CRC_DR 对应着图 4 流程图中 input_data,同时它也用来存储 CRC 校验码。CRC_INIT 对应着 Initial_Crc，POLY 对应 CRC_POLY，CRC_CR 是控制寄存器。

5. 小结

本文重点介绍了 CRC 校验的实现原理以及 CRC 校验中用到的模 2 运算和多项式的基本概念。

参考文献

文件编号	文件标题	版本号	发布日期
RM0444	STM32G0x1 advanced Arm-based 32-bit MCUs	5	2020-11-20
AN4187	Using the CRC peripheral in the STM32 family	1	2013-06-06
	IEC60335 ClassB ST MCU 软件库介绍		

版本历史（内部参考）

(仅供内部参考，评审过程中，评审员需要删除这部分：版本历史（内部参考）)

日期	版本	变更	作者
2021年11月20日	0.1	初始版本	Rusk RUAN
2021年12月22日	0.2	Reviewed	Leo ZHANG
2022年1月13日	0.3	局部字词调整、格式及版本维护	Miler Shao

版本历史

日期	版本	变更
YYYY年MM月DD日	1.0	详细变更信息

重要通知 - 请仔细阅读

意法半导体公司及其子公司 (“ST”) 保留随时对 ST 产品和 / 或本文档进行变更的权利, 恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用, ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定, 将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息, 请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章, 旨在交流与分享, 并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致, 请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档 (包括代码, 电路图 etc) 信息, 我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2020 STMicroelectronics - 保留所有权利