

文章编号:1671-6833(2010)02-0097-05

## RFID 系统数据传输中 CRC 算法的分析与实现

范文兵, 李建华, 禹士鹏, 房传礼, 葛 峰

(郑州大学 信息工程学院, 河南 郑州 450001)

**摘 要:** 分析了基于 ISO/IEC 18000-6 协议的超高频 (UHF) RFID 系统数据传输中循环冗余校验 (CRC) 算法的原理和特点, 在经典 LFSR 电路的基础上, 采用按字节并行计算 CRC 校验码的方法, 以 CRC-CCITT 生成多项式为例, 用 Verilog HDL 语言设计实现了 8 位并行 CRC-16 电路。在 Quartus II 8.0 综合开发环境下进行时序仿真, 并在 FPGA 芯片 EP1C6Q240I7 上测试验证, 结果表明: 所设计的电路在一个时钟周期内处理 8 位数据, 符合协议规定, 满足超高频 RFID 系统的通信速率要求。

**关键词:** 射频识别; 循环冗余校验; 现场可编程门阵列; 并行电路

**中图分类号:** TN911

**文献标识码:** A

### 0 引言

射频识别 (Radio Frequency Identification, RFID) 系统采用无接触的方式进行数据传输, 在数据传输过程中很容易受到干扰, 从而导致传输错误<sup>[1]</sup>。因此, 在 RFID 系统中广泛采用循环冗余校验 (Cyclic Redundancy Check, CRC) 作为数据传输正确性校验的方法。

CRC 计算一般有软件和硬件 2 种实现方法。硬件实现又分为串行和并行 2 种。目前, 在超高频 RFID 系统中, CRC 算法多数是通过串行方法实现的<sup>[2-3]</sup>。这种串行方法在一个时钟周期内只能处理一位数据, 数据处理速度慢, 而超高频 RFID 系统信息量大、通讯速率高、识别速度快, 要求具有更高的数据处理速度。为满足其高数据处理速度要求, 笔者采用按字节并行 CRC 计算方法<sup>[4]</sup>, 结合串行 CRC 电路, 设计了一个 8 位并行 CRC-16 电路。

### 1 循环冗余码基本原理

循环冗余码利用除法及余数的原理进行错误侦测, 通过模 2 运算来建立信息位和校验位之间的约束关系。首先将所传数据看成高次多项式, 用此多项式除以预先给定的生成多项式, 其余数作为校验位附加在所传数据的尾部一并传送, 即在

一个长度为  $n$  的码组中有  $k$  个信息位和  $r$  个校验位。译码时用同样的生成多项式去除, 若余数为零, 则可判断收到的数据是正确的。CRC 数学原理描述如下<sup>[5]</sup>:

假设待校验  $k$  的位信息为

$$M = (m_{k-1}, m_{k-2}, \dots, m_2, m_1, m_0) \quad (1)$$

它所对应的信息多项式为

$$M(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \dots + m_2x^2 + m_1x + m_0 \quad (2)$$

用  $x^{n-k}$  乘以  $M(x)$ , 再用给定的  $(n, k)$  循环码生成多项式  $G(x)$  除  $x^{n-k}M(x)$ , 得

$$\frac{x^{n-k}M(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \quad (3)$$

式中:  $Q(x)$  和  $R(x)$  分别是除法的商式和余式。

则  $x^{n-k}M(x) + R(x)$  即为长度为  $n$  的最终传送数据, 其对应的码字为

$$(m_{k-1}, m_{k-2}, \dots, m_2, m_1, m_0, r_{n-k-1}, r_{n-k-2}, \dots, r_2, r_1, r_0) \quad (4)$$

在接收端, 对收到的码字进行译码, 即除以相同的生成多项式  $G(x)$ , 得

$$\begin{aligned} \frac{x^{n-k}M(x) + R(x)}{G(x)} &= \frac{Q(x)G(x) + R(x) + R(x)}{G(x)} \\ &= Q(x) + \frac{R'(x)}{G(x)} \end{aligned} \quad (5)$$

若余数  $R'(x) = 0$ , 则说明传输正确; 当余数

收稿日期: 2009-08-03; 修订日期: 2009-12-16

基金项目: 河南省 2007 年度高校青年骨干教师资助项目; 河南省教育厅自然科学基金项目 (2007510019)

作者简介: 范文兵 (1969-), 男, 河南周口人, 郑州大学副教授, 博士, 主要从事信号与信息处理、射频识别技术方面的研究, E-mail: iewbfan@zzu.edu.cn.

$R'(x) \neq 0$  时,可以判断数据传输有误,达到数据通信检错的目的。

## 2 超高频 RFID 系统 CRC 电路设计

ISO/IEC18000-6<sup>[3]</sup>标准采用CRC作为数据传输正确性校验的方法,定义了3种类型的协议:Type A、Type B和Type C。Type A和Type B都使用相同的16位循环冗余校验(CRC-CCITT)作为前向链路和返回链路的校验。另外,当Type A满足防错保护所需的电平时,对短命令采用5位循环冗余校验(CRC-5)。5位CRC校验和16位CRC校验的设计和实现方法完全类似,笔者仅以Type B协议的16位CRC校验为例进行分析设计。

与基本的CRC不同,协议中的CRC生成与校验有其特殊性,需要注意以下4点:

(1)CRC的位和字节传输规则。CRC的最高有效位首先传输,同时,每个字节的最高有效位也先传输,当最低有效位传输后,16位寄存器全部清零。

(2)在进行CRC的生成和校验之前,需将余数寄存器的值置为全“1”,即16位寄存器需要预载“FFFF”。这是为了保证所传输数据最前面的“0”能够正确传输。因为基本的CRC原理并不能检测数据串头“0”的增加或减少。余数寄存器置为全“1”后,相当于把原始数据的高16位取反来计算CRC。如果由于传输错误在数据开头出现“0”的增加或减少,经过取反后的数据就会与原来的不同,故而达到了错误检测的目的<sup>[6]</sup>。由于在数值上相当于给被除数加上了个常量 $F$ ,而在CRC的生成和校验阶段都做这样的处理,所以在算法上对最终的校验结果没有影响<sup>[6]</sup>。

(3)生成CRC码后,需要将正常计算得出的结果按位取反,然后作为最终的CRC码附在数据包的末尾传输。这是为了保证所传输数据最后面的“0”能够正确传输。因为基本的CRC原理并不能检测数据串尾“0”的增加或减少。原始的CRC码取反后,在模2运算中相当于将一个系数全为“1”的多项式加在要传输的数据低位部分,即给余数加上了一个常数“FFFF”。

(4)CRC校验时,有2种方法。

方法一:首先预载16位寄存器值“FFFF”,之后将接收到的数据以及CRC码一起送入CRC电路,若校验结果为“1D0F”,则说明传输无误。“1D0F”称为魔数(magicnumber),计算方法如

下<sup>[6]</sup>:

设 $M$ 为原始数据串, $G$ 为生成多项式, $R$ 为取反前的CRC余数,所有的加法均为模2加法,可知:

$$(Mx^{16} + R) \bmod G = 0 \quad (6)$$

现在将 $R$ 取反,记为 $\bar{R}$ ,则有:

$$\begin{aligned} x^{16}(Mx^{16} + \bar{R}) \bmod G \\ &= x^{16}[Mx^{16} + (R + x^{16} + x^{15} + \cdots + x + 1)] \bmod G \\ &= x^{16}[(Mx^{16} + R) + x^{16} + x^{15} + \cdots + x + 1] \bmod G \\ &= [x^{16}(Mx^{16} + R) + x^{16}(x^{16} + x^{15} + \cdots + x + 1)] \bmod G \end{aligned} \quad (7)$$

式(6)代入式(7),得

$$\begin{aligned} x^{16}(Mx^{16} + \bar{R}) \bmod G &= x^{16}(x^{16} + x^{15} \\ &\quad + \cdots + x + 1) \bmod G \end{aligned} \quad (8)$$

对于16位CRC校验的生成多项式来说,根据式(8),即可求得结果为“1D0F”。

方法二:预载16位寄存器值“FFFF”,然后将接收到的CRC值按位取反后附在数据后输入CRC电路,输入结束后,若16位寄存器值为“0000”,则说明数据传输正确。

### 2.1 串行CRC电路设计

经典的串行硬件实现采用线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)<sup>[7]</sup>以及一些触发器和门电路完成。图1<sup>[3]</sup>为ISO/IEC18000-6附录A中提供的CRC-16(生成多项式 $G(x) = x^{16} + x^{12} + x^5 + 1$ )电路范例,它是由参考文献[7]中的LFSR2电路改进而来。

在图1中,从左至右依次为CRC寄存器 $Q_0$ 至 $Q_{15}$ ( $Q_{15}$ 为MSB, $Q_0$ 为LSB),异或门实现生成多项式 $G(x)$ 的反馈逻辑关系,其原理就是输入多项式 $M(x)$ 对生成多项式 $G(x)$ 进行模2运算。图中每个寄存器的输入值都是由当前输入数据位和最后一级寄存器的输出( $Q_{15}$ )相异或,再和对应的生成多项式 $G(x)$ 的系数相与,之后再与前一寄存器的输出相异或得到。若对应的生成多项式 $G(x)$ 的系数为“0”,则相应的寄存器输入等于前一寄存器的输出,即只对数据进行移位操作,可省去异或门。由于该电路采用的生成多项式 $G(x)$ 对应的系数为“0001,0000,0010,0001”,因生成多项式最高幂次项系数是固定的“1”,可略去固定,因此可省去一组与门,且只有 $Q_0$ 、 $Q_5$ 和 $Q_{12}$ 3个寄存器前使用了异或门。

CRC生成时,首先预载16位寄存器值“FFFF”,然后当每一个时钟来临时输入一位数

据,经过相应的异或和移位运算,当所有数据输入完成后,16 位寄存器的值即为未反转的 CRC 值,反转后即 为协议所规定的 CRC 校验码,可附加在命令或数据后发送出去。

图 1 为一同步时序逻辑电路,设输入数据为  $D$ ,其状态方程如下:

$$\begin{cases} Q_0^{n+1} = Q_{15} \oplus D \\ Q_5^{n+1} = Q_4 \oplus Q_{15} \oplus D \\ Q_{12}^{n+1} = Q_{11} \oplus Q_{15} \oplus D \\ Q_i^{n+1} = Q_{i-1} \end{cases} \quad (9)$$

其中:  $i = 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15$ 。

根据以上逻辑关系,很容易用 Verilog HDL 语言描述,并在 FPGA 中实现。

### 2.2 并行 CRC 电路设计

串行电路一个时钟周期内只能处理一位数据,为提高处理速度,只能提高时钟频率,但这样做就会增加开发的难度和成本。RFID 系统在读写器与标签的一次双向数据通信中需要进行编码和

解码 2 次 CRC 计算,要求系统具有较高的数据处理速度,所以采用并行 CRC 算法。

并行方式需要确定每次并行处理的位数。由于超高频 RFID 系统中需要进行 CRC 校验的数据位数均为 8 的倍数,这里设计 8 位并行度的 CRC - CCITT 电路。

并行 CRC 电路与串行 CRC 电路计算结果应当是等效的,即一个时钟周期内 8 位数据同时输入并行运算电路产生的 CRC 余数与 8 位数据逐位输入串行运算电路产生的 CRC 余数相同。基于这一点,并结合式(9)的逻辑关系,采用代入的方法可由串行算法推导出每次处理 8 位数据后的 CRC 余数与当前输入数据和 CRC 寄存器前一状态的关系。

图 2 显示了该方法的推导过程,寄存器  $Q_0 \sim Q_{15}$  分别对应图 1 中相应寄存器,CRC 项为对应寄存器中的 CRC 值, $C_{15}$  是 CRC 码的最高位, $D_7$  是输入 8 位数据的最高位。开始计算前,16 个寄存器全置“1”,即每一位 CRC 都为“1”。以后的每一

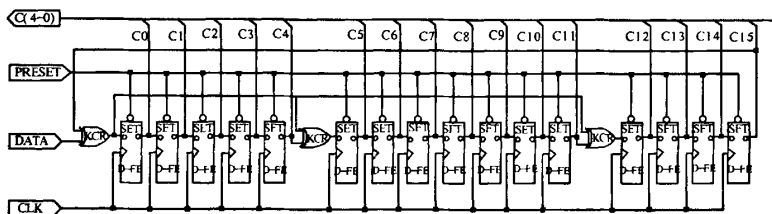


图 1 串行 CRC - 16 LFSR 电路  
Fig.1 Serial CRC - 16 LFSR circuit

寄存器	$Q_0$	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	$Q_7$	$Q_8$	$Q_9$	$Q_{10}$	$Q_{11}$	$Q_{12}$	$Q_{13}$	$Q_{14}$	$Q_{15}$
CRC	$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$	$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$
步骤	数据	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或	异或
1	$D_7$	$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$	$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$
2	$D_6$	$D_7$	$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$	$C_{11}$	$C_{12}$	$C_{13}$
...	...	依次递推														
8	$D_0$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
		$C_8$	$C_9$	$C_{10}$	$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$D_8$	$D_9$
		$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$	$C_0$	$C_1$
		$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$	$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$	$D_0$	$D_1$	$D_2$
		$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$	$C_{15}$				$C_4$	$C_5$	$C_6$

图 2 并行 8 位 CTC - 16 逻辑关系推导过程  
Fig.2 The derivative process of 8 - bit parallel CTC - 16 logical relations

行表示当前输入一位数据后 CRC 寄存器中新的内容,CRC 的最终值为输入  $D_0$  后各个项的因子相互异或.

根据图 2 推导出的逻辑关系,很容易用 Verilog HDL 语言设计实现.

### 3 仿真与分析

在 Altera 公司的 Quartus II 8.0 综合开发环境下,以 Type B 协议中的“SUCCESS”命令为例,分别对串行电路和并行电路进行时序仿真.

从读写器发送到标签的数据帧由图 3 所示,但是仅“SUCCESS”命令的命令码(09H)用于循环冗余校验.

CRC 码的生成和校验可以用同样的电路完成,所不同的只是进行校验时输入数据中包含有 CRC 码.设计的 CRC 电路端口定义如图 4 所示.

帧头检测	帧头	起始分隔符	SUCCESS 命令	CRC-16
2 个字节	9 个曼彻斯特码 0	11 00 11 10 10	09H	CRC-16

图 3 Type B 协议中的“SUCCESS”命令

Fig. 3 SUCCESS command in Type B protocol

端口	clk	reset	enable	start	data in	erc_out
L/O	input	input	input	input	input	output
说明	时钟	复位	使能	启动	输入数据	输出 CRC

图 4 CRC 电路端口定义

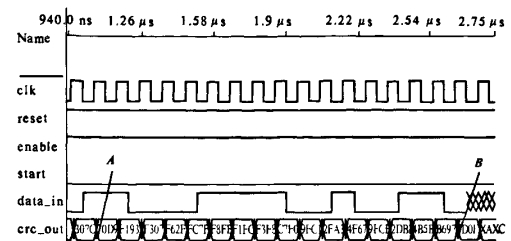
Fig. 4 Port definition of CRC circuit

串行电路的时序仿真波形如图 5 所示.图 5 (a)采用方法一校验,在 start 启动信号后,按位依次输入 24 位待校验数据“098F26”其中前 8 位为信息位“09”,后 16 位为 CRC-16 码,B 点处输入数据结束,CRC 寄存器值为“1D0F”,结果正确,传输无误.图 5 (b)采用方法二进行校验,输入 24 位数据为“0970D9”,在 B 点得到校验结果“0000”,传输无误.

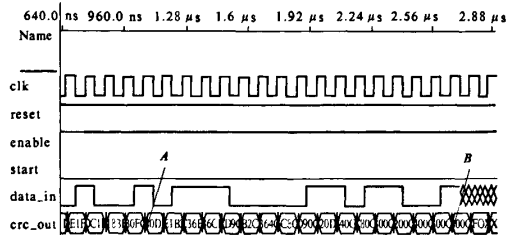
并行电路时序仿真波形如图 6 所示.图 6 (a)采用方法一校验,输入数据为“098F26”,在 A 点,即 start 信号后第一个时钟上升沿,采样高 8 位数据并同时计算 CRC 码,之后经过 2 个时钟周期,在 B 点得到校验结果“1D0F”,说明传输正确.图 6 (b)采用方法二校验,输入数据为“0970D9”,在 B 点得到正确的校验结果“0000”.

由以上仿真波形可以看出,对同一“SUCCESS”命令进行校验,串行电路需要 24 个时钟周

期,而该并行电路每次处理 8 位数据,仅需 3 个时钟周期,效率是原串行电路的 8 倍.

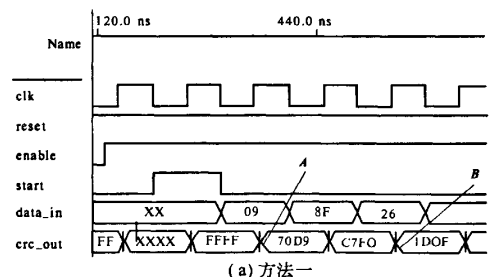


(a) 方法一

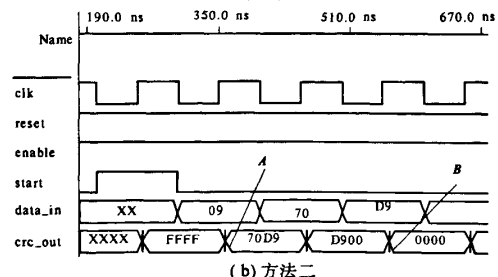


(b) 方法二

图 5 串行电路校验波形  
Fig. 5 Serial circuit waveform



(a) 方法一



(b) 方法二

图 6 并行电路校验波形  
Fig. 6 Parallel circuit waveform

### 4 结束语

笔者在分析 RFID 系统数据传输中 CRC 校验算法的基础上,设计了串行和并行 CRC 电路.在 FPGA 芯片 EP1C6Q240I7 上进行测试验证,结果表明:该电路可以配合 RFID 其它电路工作,与串行电路相比,8 位并行 CRC 电路在一个时钟周期

内处理8位数据,满足了超高频RFID系统的通信速率要求。同时,笔者使用的设计方法,避开了繁琐的数学推导,方法简单、易于硬件实现、可移植性强。

### 参考文献:

- [1] 周晓光,王晓华,王伟. 射频识别系统(RFID)设计仿真与应用[M]. 北京:人民邮电出版社,2008:170-171.
- [2] 陈亦灏. 符合EPC C1G2标准的数字基带关键技术研究[D]. 上海:华东师范大学信息学院,2008.
- [3] ISO/IEC Information Centre. ISO/IEC 18000-6 Information technology - Radio frequency identification for item management - Part 6: Parameters for air interface communications at 860 MHz to 960 MHz[S]. Switzerland: International Organization for Standardization, 2004.
- [4] PEREZ A. Byte - Wise CRC Calculations[J]. IEEE Micro,1983,3(3):40-50.
- [5] 康东,石喜勤,李勇鹏. 射频识别(RFID)核心技术与典型应用开发案例[M]. 北京:人民邮电出版社,2008:48-53.
- [6] 俞迅. 32位CRC校验码的并行算法及硬件实现[J]. 信息技术,2007,31(4):71-74.
- [7] CAMPOBELLO G, PATANE G, RUSSO M. Parallel CRC realization[J]. IEEE Transactions on Computers, 2003, 52(10):1312-1319.

## Analysis and Implementation of CRC in RFID System

FAN Wen-bing, LI Jian-hua, YU Shi-peng, FANG Chuan-li, GE Zheng

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** This paper analyzes the principle and characteristic of CRC calculations in UHF RFID system data communications based on the ISO/IEC 18000-6 standard. Using a byte-wise method for parallel computing CRC and the Verilog HDL language, and taking CRC-CCITT generator polynomial for example, we designed a byte-oriented parallel CRC-16 circuit based on the classical LFSR circuit. Timing simulation in Quartus II 8.0 integrated development environment, and verifying it in the FPGA chip EP1C6Q240I7, the results show that this circuit we designed can processing 8 bit data per clock cycle. It is in agreement with the ISO/IEC 18000-6 standard and meet the requirements of communication rate in UHF RFID systems.

**Key words:** radio frequency identification; CRC; FPGA; parallel circuit