

基于 IEC 60044-8 标准的电子式电流互感器 数字输出编码模块的 FPGA 实现

朱 雷,盛春波,郑绳榷

(燕山大学 信息科学与工程学院,河北 秦皇岛 066004)

摘要:介绍了电子式电流互感器国际电工委员会标准 IEC 60044-8 的链路层规则,设计了数据组帧编码模块。针对帧格式中的循环冗余校验(CRC)校验码,通过详细的计算推导,设计出 8 位并行 CRC 逻辑电路并应用于现场可编程门阵列(FPGA),在 MAX+Plus II 环境下进行了仿真,与串行 CRC 相比,并行 CRC 的编码速度大为提高。在物理层,将完整的数据帧进行曼彻斯特编码后通过光纤传输至间隔层。实践证明,该方案实时性强、准确度高,具有广泛的应用价值。

关键词:电子式电流互感器;循环冗余校验;现场可编程门阵列;曼彻斯特码

中图分类号: TN 919.3+3 **文献标识码:** A **文章编号:** 1006-6047(2006)08-0067-04

电子式电流互感器是电力系统中的重要设备,研究它与变电站自动化系统的接口与通信,对于变电站的数字化有着重要的意义^[1]。

目前,对于电子式电流互感器的研制工作已向实用化阶段发展。国际电工委员会(IEC)已经制定出电子式电流互感器标准:IEC 60044-8^[2],对电子式电流互感器的数字输出作出了统一规范。

1 IEC 60044-8 链路层规则介绍

IEC 60044-8 标准中的链路层帧格式采用的是 FT 3 帧格式,此格式在 IEC 60870-5-1 中有所规定,如下所示。

字节 1~2	字节 3~18	字节 19~20	字节 21~36
起始字节	用户数据 1	核对序列 1	用户数据 2
字节 37~38	字节 39~54	字节 55~56	
核对序列 2	用户数据 3	核对序列 3	

收稿日期:2005-11-10;修回日期:2006-03-08
基金项目:国家自然科学基金资助项目(50477001)

2 数据组帧编码模块的总体设计

新型变电站的结构一般包括 3 层,即过程层、间隔层和变电站层,电流/电压互感器属于过程层^[3]。电子式电流互感器保护和测量的数据应按照 FT3 帧格式传输,传输速度为 2.5 Mbit/s,用曼彻斯特码传输,传输中应保证实时性和准确性。在过程层首先经过数字信号处理模块完成对接收到的采样数据的抽取滤波,接下来便是按照 IEC 60044-8 规约的要求对抽取滤波后的数据组帧编码,加入额定电流、电压、延迟以及状态码等。链路层的设计只要将数据加上帧头、循环冗余校验 CRC(Cyclic Redundancy Check)码完成组帧,然后在物理层将数据进行曼彻斯特编码即可。编码后的数据经过 E/O 变换后通过光纤高速传输,发送至间隔层。数据组帧编码总体的设计如图 1 所示,下面将对 CRC 校验码和曼彻斯特编码的概念和实现方法加以介绍。

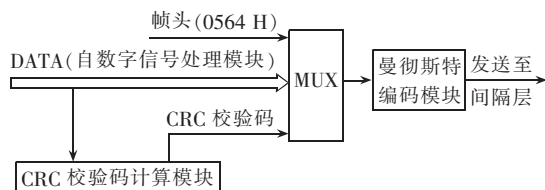


图 1 数据组帧编码设计框图

Fig.1 The design block diagram of data framing and encoding

3 循环冗余校验的基本原理

为了确保数据的可靠传输,需要在数据链路中加入差错检测码。CRC 码是在严密的代数学理论基础建立的,以其实现简单、抗干扰能力强而得到广泛应用,IEC 60044-8 标准中便是使用 CRC 校验码进行差错检测和控制。

采用 CRC 校验时,发送方和接收方预先约定一个生成多项式 $G(x)$,该生成多项式作为除数多项式(IEC 60044-8 中所规定的生成多项式 $G(x)$ 为 $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$),将要发送的数据比特序列作为一个多项式 $f(x)$ 的系数,该多项式为被除多项式,用 $G(x)$ 除 $f(x)$ 得到一个余数多项式,该余数多项式的系数即构成了数据比特序列的 CRC 校验码,将它添加到数据序列之后一起发送到接收方。接收方同样将接收到的数据看成是一个多项式的系数,并用相同的生成多项式除该多项式,若得到的余数为 0,表示传输过程无差错,否则表示出错^[4]。发送方编码过程如下:首先,将待发送数据多项式 $f(x)$ 乘以 x^k ,其中 k 为生成多项式 $G(x)$ 的最高幂次,本文中 $k = 16$,对于二进制乘法, $f(x)x^{16}$ 意味着将 $f(x)$ 对应的发送比特序列左移 16 位。其次,用 $G(x)$ 去除以上乘积,可得到 $f(x)x^{16} / G(x) = Q(x) + R(x)/G(x)$,式中 $Q(x)$ 为商, $R(x)$ 为余数多项式。由于不带进(借)位的模 2 运算加减法是等价的,就有 $f(x)x^{16} + R(x) = G(x)Q(x)$ 。最后,令 $M(x) = f(x)x^{16} + R(x)$,将 $M(x)$ 所对应的比特序列作为一个整体发送。接收方校验时,对接收序列所对应的多项式 $M'(x)$ 作如下处理:计算 $M'(x)/G(x)$,若 $M'(x) \div G(x) = Q(x)$,余数为 0,则 $M'(x) = M(x)$,认为传输正确;若 $M'(x)/G(x)$ 余数不为 0,则认为传输有错^[5]。

4 链路层 CRC 校验码的设计与实现

随着大规模集成电路的发展,基于 EDA(Electronic Design Automation)技术的可编程逻辑器件的出现,解决了传统数字电路设计中所遇到的问题,它具有编程方式简便、速度快、可靠性高、功能强大和开发便捷的优点。本系统采用 Altera 公司生产的 EPF10K70RC240-2 芯片,EPF10K70RC240-2 属于 FLEX10K 系列现场可编程门阵列(FPGA),它具有 70 000 个典型门,器件内部具有 3 744 个逻辑单元

(LE),足以实现大规模的复杂逻辑。开发软件采用 MAX+Plus II,利用 VHDL(Very High speed integrated circuit hardware Description Language) 语言编程,可方便地实现各种数字逻辑功能^[6]。CRC 编码涉及到模 2 运算,根本上可归结为数字信号的移位和异或操作,因此,可以通过数字逻辑实现。为了有所比较,首先给出 CRC 校验码的串行实现,然后,在串行实现的基础上,通过公式递推,设计出并行运算的逻辑电路并在 FPGA 上加以实现。

4.1 串行实现

串行编码方式结构简单易于实现,根据生成多项式画出的电路结构如图 2 所示,整个电路由线性反馈移位寄存器(LFSR)和异或门组成。由图可见,串行计算时,每次输入 1 位数据,输入数据和上次异或运算的结果组成新数据,循环进行异或运算,直到所有数据都已经输入,这时移位寄存器中的状态值即为输入数据的循环冗余校验码。使用 VHDL 语言编程,通过编译、仿真,实现了串行编码的功能。

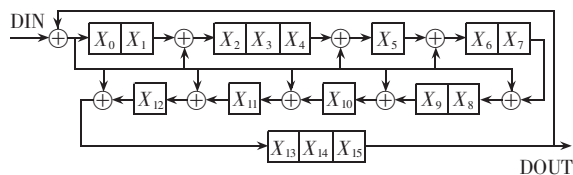


图 2 串行 CRC 编码原理图

Fig.2 Encoding principle of serial CRC

IEC 60044-8 链路层规定 16 个字节的数据码之后跟随 16 位的校验码,这里串行输入的 16 个字节的数据码。采用十六进制表示为 B16A13D230E603106460F03C1DC045C7。仿真过程中时钟周期设定为 100 ns,系统复位时,寄存器置零。通过观察仿真波形可知,16 个字节的数据码逐位输入之后,生成的校验码为 6909 H,编码时间为 128 个时钟周期。

4.2 并行实现

目前,CRC 并行运算大多是查表法及基于查表法而导出的一些方法。这些方法均需要存储长度较大的 CRC 余数表,灵活性和快速性都不高^[7]。从图 2 可看出,各移位寄存器的状态值即为 CRC 余数值,当串行运算时,当前的 CRC 余数值只与数据码的当前 1 位的输入值和前 1 状态的余数值有关^[8]。可以推断,在进行 8 位并行运算时,8 位数据码同时输入并行运算电路所产生的 CRC 余数应该与串行运算时连续 8 位数据码逐位输入,串行运算电路所产生的 CRC 余数完全相同。根据这一思想,可以从图 2 的串行运算电路推导出并行运算的逻辑关系。

设 r_i^j 、 d_j ($i = 0, 1, 2, \dots, 15$; $j = 1, 2, \dots, 8$) 分别表示移位寄存器状态值和输入数据码,其中 r_i^j ($i = 0, 1, 2, \dots, 15$) 为寄存器 i 在本次 8 位数据码依次输入串行运算电路后的状态值, r_i^0 ($i = 0, 1, 2, \dots, 15$) 为寄存器 i 在本次 8 位数据码输入串行运算电路前

的状态值。“ $\oplus$ ”表示异或运算。

$$\begin{aligned}
 r_0^8 &= r_{15}^7 \oplus d_8 = r_{14}^6 \oplus d_8 = r_{13}^5 \oplus d_8 = \\
 &(r_{12}^4 \oplus r_{15}^4 \oplus d_5) \oplus d_8 = r_{15}^4 \oplus r_{12}^4 \oplus d_8 \oplus d_5 = \\
 &r_{14}^3 \oplus (r_{11}^3 \oplus r_{15}^3 \oplus d_4) \oplus d_8 \oplus d_5 = \\
 &r_{15}^3 \oplus r_{14}^3 \oplus r_{11}^3 \oplus d_8 \oplus d_5 \oplus d_4 = \\
 &r_{14}^2 \oplus r_{13}^2 \oplus (r_{10}^2 \oplus r_{15}^2 \oplus d_3) \oplus d_8 \oplus d_5 \oplus d_4 = \\
 &r_{15}^2 \oplus r_{14}^2 \oplus r_{13}^2 \oplus r_{10}^2 \oplus d_8 \oplus d_5 \oplus d_4 \oplus d_3 = \\
 &r_{14}^1 \oplus r_{13}^1 \oplus (r_{12}^1 \oplus r_{15}^1 \oplus d_2) \oplus (r_9^1 \oplus r_{15}^1 \oplus \\
 &d_2) \oplus d_8 \oplus d_5 \oplus d_4 \oplus d_3 = \\
 &r_{14}^1 \oplus r_{13}^1 \oplus r_{12}^1 \oplus r_9^1 \oplus d_8 \oplus d_5 \oplus d_4 \oplus d_3 = \\
 &r_{13}^0 \oplus (r_{12}^0 \oplus r_{15}^0 \oplus d_1) \oplus (r_{11}^0 \oplus r_{15}^0 \oplus d_1) \oplus \\
 &r_8^0 \oplus d_8 \oplus d_5 \oplus d_4 \oplus d_3 = \\
 &r_{13}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_8^0 \oplus d_8 \oplus d_5 \oplus d_4 \oplus d_3 \quad (1)
 \end{aligned}$$

同理,可以推算出其余 15 个寄存器在本次 8 位数据码依次输入串行运算电路后的状态值,计算结果如下:

$$r_1^8 = r_{14}^0 \oplus r_{13}^0 \oplus r_{12}^0 \oplus r_9^0 \oplus d_7 \oplus d_4 \oplus d_3 \oplus d_2 \quad (2)$$

$$\begin{aligned}
 r_2^8 &= r_{15}^0 \oplus r_{14}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_8^0 \oplus d_8 \oplus \\
 &d_6 \oplus d_5 \oplus d_4 \oplus d_2 \oplus d_1 \quad (3)
 \end{aligned}$$

$$\begin{aligned}
 r_3^8 &= r_{15}^0 \oplus r_{13}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_9^0 \oplus d_7 \oplus \\
 &d_5 \oplus d_4 \oplus d_3 \oplus d_1 \quad (4)
 \end{aligned}$$

$$r_4^8 = r_{14}^0 \oplus r_{13}^0 \oplus r_{12}^0 \oplus r_{10}^0 \oplus d_6 \oplus d_4 \oplus d_3 \oplus d_2 \quad (5)$$

$$r_5^8 = r_{15}^0 \oplus r_{14}^0 \oplus r_{12}^0 \oplus r_8^0 \oplus d_8 \oplus d_4 \oplus d_2 \oplus d_1 \quad (6)$$

$$\begin{aligned}
 r_6^8 &= r_{15}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_9^0 \oplus r_8^0 \oplus d_8 \oplus \\
 &d_7 \oplus d_5 \oplus d_4 \oplus d_1 \quad (7)
 \end{aligned}$$

$$r_7^8 = r_{13}^0 \oplus r_{12}^0 \oplus r_{10}^0 \oplus r_9^0 \oplus d_7 \oplus d_6 \oplus d_4 \oplus d_3 \quad (8)$$

$$r_8^8 = r_{14}^0 \oplus r_{12}^0 \oplus r_{10}^0 \oplus r_8^0 \oplus d_8 \oplus d_6 \oplus d_4 \oplus d_2 \quad (9)$$

$$r_9^8 = r_{15}^0 \oplus r_{13}^0 \oplus r_{11}^0 \oplus r_9^0 \oplus r_1^0 \oplus d_7 \oplus d_5 \oplus d_3 \oplus d_1 \quad (10)$$

$$\begin{aligned}
 r_{10}^8 &= r_{14}^0 \oplus r_{13}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_8^0 \oplus r_2^0 \oplus \\
 &d_8 \oplus d_6 \oplus d_5 \oplus d_3 \oplus d_2 \quad (11)
 \end{aligned}$$

$$\begin{aligned}
 r_{11}^8 &= r_{15}^0 \oplus r_{14}^0 \oplus r_{13}^0 \oplus r_9^0 \oplus r_8^0 \oplus r_3^0 \oplus \\
 &d_8 \oplus d_7 \oplus d_3 \oplus d_2 \oplus d_1 \quad (12)
 \end{aligned}$$

$$\begin{aligned}
 r_{12}^8 &= r_{15}^0 \oplus r_{14}^0 \oplus r_{13}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_9^0 \oplus r_8^0 \oplus \\
 &r_4^0 \oplus d_8 \oplus d_7 \oplus d_6 \oplus d_5 \oplus d_4 \oplus d_3 \oplus d_2 \oplus d_1 \quad (13)
 \end{aligned}$$

$$\begin{aligned}
 r_{13}^8 &= r_{15}^0 \oplus r_{14}^0 \oplus r_{10}^0 \oplus r_9^0 \oplus r_8^0 \oplus r_5^0 \oplus \\
 &d_8 \oplus d_7 \oplus d_6 \oplus d_2 \oplus d_1 \quad (14)
 \end{aligned}$$

$$\begin{aligned}
 r_{14}^8 &= r_{15}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_9^0 \oplus r_6^0 \oplus \\
 &d_7 \oplus d_6 \oplus d_5 \oplus d_1 \quad (15)
 \end{aligned}$$

$$r_{15}^8 = r_{12}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_7^0 \oplus d_6 \oplus d_5 \oplus d_4 \quad (16)$$

可见,本次 CRC 值只与本次输入的 8 位数据码以及本次数据输入前寄存器的 CRC 余数值有关,由此可以建立起 8 位 CRC 并行运算的逻辑电路,原理如图 3 所示。

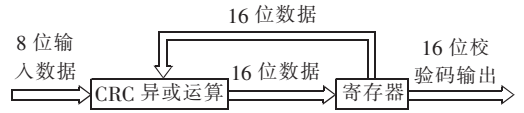


图 3 8 位并行 CRC 编码原理图

Fig.3 The schematic diagram of 8-bit parallel CRC encoding

使用 VHDL 语言编程,通过编译、仿真,实现了 8 位并行 CRC 编码的功能。为了与前面的串行电路相比较,16 个字节的输入数据码与串行输入的数据码相同。同样,设定时钟周期为 100 ns,第 1 个时钟周期内系统处于复位状态,寄存器置零。第 2 个时钟周期开始输入并行数据,8 位并行数据码与寄存器初始值按照前面推导出的逻辑关系进行异或运算,所得 16 位 CRC 余数值存入寄存器,供下次并行运算使用。当下一组 8 位数据来临时,再将 8 位数据与寄存器当前的余数值进行异或运算,所得新余数存入寄存器。依此类推,当 16 个字节的输入数据码全部输入完毕,生成的校验码为 6909H,这与串行编码的结果保持一致。

实际上,通过对前面串行编码仿真波形的观察,可以看出从第 1 位输入数据码开始,每 8 位数据码串行输入后所产生 CRC 余数值都与这里并行编码仿真波形中的相对应部分完全一致,从而进一步证实了并行实现的可靠性和正确性。同时,这里编码只用了 16 个时钟周期,是串行编码所需时间的 1/8,满足了电力通信快速性的要求,优势十分明显。

5 曼彻斯特码的实现

曼彻斯特码又称数字双相码,是一种时钟自同步编码技术,每个二进制码分别利用 2 个具有不同相位的二进制新码所取代的码^[9]。编码规则之一是:在每个数据码元间隔的中间以下降沿代表数据“1”,上升沿代表数据“0”。时钟(CLK)、数据和曼彻斯特码的波形关系如图 4 所示。

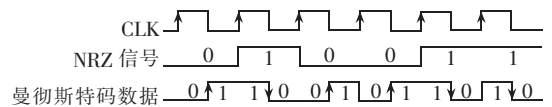


图 4 曼彻斯特码波形

Fig.4 Manchester code waveform

对于二进制通信,代表“0”码与“1”码的 2 个信号波形间的相关系数越小,则判决错误的概率越小。曼彻斯特码代表“0”码与“1”码的信号相关系数为 -1,因而收端判决比较容易,错误的概率很小。

曼彻斯特码的编码过程相对解码较为简单,从图 4 中看出只要将 NRZ 码与时钟信号异或倒相即可得到曼彻斯特码^[10]。不过由于数据和 CLK 的位置不完全一致,而且数据信号的上升沿和下降沿不理想,可能会产生尖峰脉冲,使输出信号出现毛刺,这在实际中是需要考虑的。如图 5 所示,本文采用改

进后的电路,将 NRZ 信号与 CLK 异或非后,还要用一个 2 倍时钟频率的信号通过 D 触发器整形。由于在 2 倍时钟之后使用了非门,所以是采用时钟的下降沿去取样编码后的数据,即可有效地消除毛刺信号。仿真结果表明,所设计的电路能够实现对信号的正确编码。

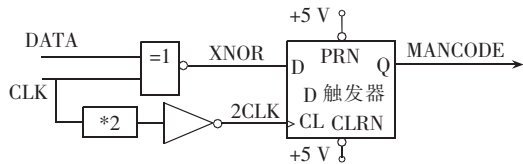


图 5 曼彻斯特编码电路

Fig.5 Manchester encoding circuit

6 结语

按照 IEC 60044-8 标准的要求,本文设计了电子式电流互感器数字输出编码模块,介绍了 CRC 校验码的基本原理,结合标准所规定的生成多项式,在串行设计基础上,给出了详细的并行设计过程,通过 VHDL 语言编程,在 FPGA 中实现,具有很高的实时性,实践证明完全是可行的。在物理层,设计了曼彻斯特编码电路,将完整的数据帧进行曼彻斯特编码后通过光纤传输至间隔层。所设计的编码模块满足了 IEC 60044-8 标准所规定的快速性的要求,达到了预定的目标,可方便地应用于电力通信系统,具有广泛的推广应用价值。

参考文献:

- [1] 李伟,尹项根,韩小涛. 基于 IEC 60044-7/8 的光电式互感器在变电站自动化系统中的应用[J]. 电力自动化设备,2003,23(5): 39-42.
LI Wei,YIN Xiang-gen,HAN Xiao-tao. Application of optical transducer based on IEC 60044-7/8 protocols in substation automation system[J]. Electric Power Automation Equipment, 2003,23(5):39-42.
- [2] International Electrotechnical Commission. IEC 60044-8 Instru-

ment transformers part 8:electronic current transducers[S]. [S.1.]: IEC,2002.

- [3] 谭文恕. 变电站自动化系统的结构和传输规约[J]. 电网技术, 1998,22(8):1-4.
TAN Wen-shu. Configuration and transmission protocol of sub-station automation system[J]. Power System Technology,1998,22(8):1-4.
- [4] STALLINGS W. 数据与计算机通信[M]. 王海,张娟,蒋慧,等译. 北京:电子工业出版社,2001.
- [5] RAMABADRAN T V,GAITONDE S S. A tutorial on CRC computations[J]. IEEE Micro,1988,8(4):62-75.
- [6] 杨刚,龙海燕. 现代电子技术——VHDL 与数字系统设计[M]. 北京:电子工业出版社,2004.
- [7] 朱荣华. 一种 CRC 并行计算原理及实现方法[J]. 电子学报, 1999,27(4):143-145.
ZHU Rong-hua. The principle and implementation of a parallel CRC computing[J]. Acta Electronica Sinica,1999,27(4):143-145.
- [8] 黄海波,刘磊,鲍黎波. 循环冗余校验编译码的并行处理研究及其 FPGA 实现[J]. 湖北汽车工业学院学报,2003,17(4):27-30.
HUANG Hai-bo,LIU Lei,BAO Li-bo. The parallel processing study and FPGA realization for codec of cyclic redundancy check [J]. Journal of Hubei Automotive Industries Institute,2003,17(4): 27-30.
- [9] 高嵩,何宁,王国琿,等. 曼彻斯特码多路数据传输系统[J]. 西安工业学院学报,2003,23(4):294-298.
GAO Song,HE Ning,WANG Guo-hui,et al. Multiple data transmission system with Manchester code[J]. Journal of Xi'an Institute of Technology,2003,23(4):294-298.
- [10] 陈新坤,周东,余敬东. Manchester 编码器的 FPGA 设计与实现 [J]. 电子科技大学学报,2003,32(3):324-327.
CHEN Xin-kun,ZHOU Dong,YU Jing-dong. Design and FPGA implementation of Manchester encoder[J]. Journal of UEST of China,2003,32(3):324-327.

(责任编辑:汪仪珍)

作者简介:

朱雷(1979-),男,江苏金坛人,硕士研究生,从事电力通信及信号处理方面的研究(E-mail:zhulei0814@163.com);
盛春波(1981-),男,山东邹城人,硕士研究生,从事数据采集及信号处理方面的研究;
郑绳樵(1937-2006),男,浙江衢州人,教授,博士研究生导师,从事电流、电压互感器及电力通信方面的研究。

FPGA realization of digital output encoding module for electronic current transformers based on IEC 60044-8 standard

ZHU Lei,SHENG Chun-bo,ZHENG Sheng-xuan

(Yanshan University,Qinhuangdao 066004,China)

Abstract: The link layer rule of IEC 60044-8 standard for electronic current transformers is introduced. The data framing and encoding module is designed. For the CRC(Cyclic Redundancy Check) code in frame format,an 8-bit parallel CRC logical circuit is designed through detailed calculation and derivation and implemented in FPGA(Field Programmable Gate Array),which is simulated under MAX+Plus II. Compared with serial CRC,the encoding efficiency of parallel CRC is considerably higher. In the physical layer,the total data frame is encoded on Manchester code and then transmitted to the bay level using fibers. The result proves that the designed scheme has a wide practical value for its better real-time performance and higher precision.

This project is supported by the National Natural Science Foundation of China(50477001).

Key words: electronic current transformer; CRC; FPGA; Manchester code