

文章编号: 1671-0576(2020)01-0044-04

基于 FPGA 的 1553B 总线编解码模块的设计与改进

何文青, 章 恺, 马超男, 席燕博, 陈伟雄

(上海无线电设备研究所, 上海 201109)

摘 要: 针对现有 1553B 总线编解码器结构设计复杂、编解码实时性较差的问题, 提出了一种改进的编解码方法。利用寄存器按位进行曼彻斯特编码, 提高了整体的编码效率。利用移位寄存器对同步头进行匹配, 实现了快速解码。仿真结果表明: 该设计简化了编解码逻辑, 完整实现了 1553B 总线的编解码功能。

关键词: 1553B 总线; 曼彻斯特码; 同步头检测; 编解码

中图分类号: TN919

文献标志码: A

DOI: 10.3969/j.issn.1671-0576.2020.01.009

Design and Improvement of Coding and Decoding Module of 1553B Bus Based on FPGA

HE Wen-qing, ZHANG Kai, MA Chao-nan, XI Yan-bo, CHEN Wei-xiong

(Shanghai Radio Equipment Research Institute, Shanghai 201109, China)

Abstract: To solve the problems of complex design and poor real-time performance of coding and decoding for 1553B bus, an improved method of coding and decoding is proposed. Registers are used for Manchester coding by bits, which improves the overall coding efficiency. Shift registers are used to match synchronization word for fast decoding. The simulation results show that the method simplifies the coding logic and decoding logic. The function of coding and decoding is realized for 1553B bus.

Key words: 1553B bus; Manchester code; synchronization word detecting; coding and decoding

0 引言

1553B 总线的全称为 MIL-STD-1553B 总线, 原是美国军方为飞机设备制定的一种用于信息传

输的总线标准。随着 1553B 总线的应用日益广泛, 国外许多公司推出了自己的 1553B 协议芯片^[1-2]。但这些芯片价格昂贵, 对于科研生产中的一般使用来说, 许多功能也显得多余。现今由于芯片技术和工艺的持续发展, 用现场可编程门阵列(FPGA)实现 1553B 总线的编解码已经可行。

近年来, 采用 FPGA 实现 1553B 总线编码或解码功能的设计方法被提出, 但实现过程均较为

收稿日期: 2020-01-05

作者简介: 何文青(1985—), 男, 硕士, 工程师, 主要从事信号处理、导引头故障定位技术研究。E-mail: ll_king3@163.com

复杂。文献[3]在 1.5 MHz 时钟上升沿来临时,将输入寄存器中的数据经过并-串转换输出,同时与 3 MHz 时钟相异或,完成曼彻斯特编码。整个编码过程需要设置两个时钟信号,较为繁琐。解码过程对接收寄存器两两分组进行解码。这样做风险较大,一旦寄存器存储错位,解码的数据将完全相反。文献[4]通过一个 2 选 1 选择器进行编码,虽较文献[3]简便,但编码过程耗时较长,在解码过程中则需要提取“00”或“11”作为标志位,且未有效利用同步头。文献[5]和文献[6]通过设计状态机对同步头锁定实现解码,状态机之间的关联和跳转逻辑稍显复杂。本文设计了通过寄存器处理、快速进行曼彻斯特编码的方法,并采用移位寄存器匹配同步头的方法完成解码,从而解决了 1553B 总线编解码器结构、逻辑复杂及编解码速度慢的问题。

1 曼彻斯特码

1553B 总线采用的曼彻斯特码(Manchester Code),又称为双相码,是通过电平跳变来表示二进制信息的数字编码码型。图 1 为相同时钟下不归零(Not Return to Zero, NRZ)码和曼彻斯特码的对照图。相比于不归零码,曼彻斯特码每一位时中间都有一次跳变^[1]。图中以两条虚线之间,即时钟信号两个上升沿之间,为一个位时。在一个位时中,不归零码为“1”时,曼彻斯特码用“10”表示。反之,不归零码为“0”时,曼彻斯特码用“01”表示。

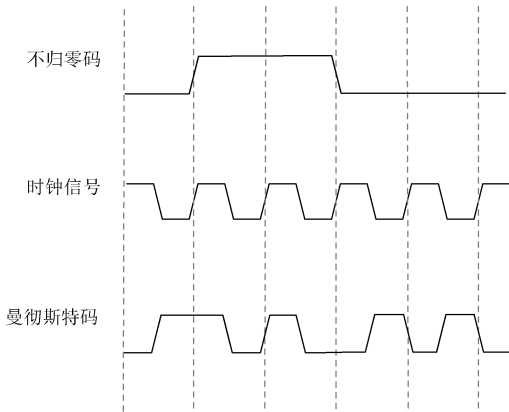


图 1 不归零码和曼彻斯特码对照图

传统的 1553B 总线编码过程,无论是采用寄

存器输出后与时钟异或,或是按位使用 2 选 1 选择器产生曼彻斯特码,都是逐位进行编码。该方式使得整个编码过程耗时较长,编码效率低下,系统资源占用率高。

传统的 1553B 总线在曼彻斯特码解码过程中通常采用读取“00”或“11”标志位或状态机方法进行解码。如果连续多位出现“10”或“01”,使用标志位方法并不能有效解码。采用状态机的方法其本身也是一种利用同步头进行匹配的方式,但是状态机跳转的逻辑表达并没有直接利用同步头进行匹配直观。

2 改进后编码模块的实现

1553B 总线的协议规定组成消息的最小单位为字,每个字由连续的 20 位组成。图 2 为数据字的组成示意图,其中包含 3 位同步头,16 位信息位和 1 位校验位^[2,7]。

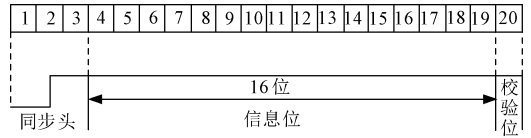


图 2 数据字组成示意图

该曼彻斯特编码模块的设计目的是使输入数据能快速完成编码并发送。主要完成以下功能:(1)编码器根据检测数据类型标志位生成同步头;(2)并行输入 16 位信息位并产生奇偶校验位,将 16 位输入信息和校验位快速进行曼彻斯特编码,并加入同步头;(3)计数器根据计数确定发送位,进行串行发送。图 3 为曼彻斯特编码模块的逻辑框图。

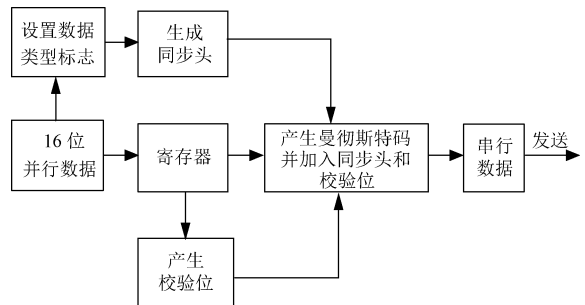


图 3 曼彻斯特编码模块逻辑图

编码模块时钟 enc_clk 采用 2 MHz,输入端

采用并行方式输入数据。根据 1553B 总线协议,数据在总线上的传输速率为 1 Mb/s,因此原来 3 位的同步头将扩展为 6 位^[2,7]。产生同步头部分,在时钟上升沿时监测同步头标志位 tx_csw(命令字或状态字)和 tx_dw(数据字)。当 tx_csw=1 且 tx_dw=0 时,同步头 sync_bits 赋值“6'b111_000”。当 tx_csw=0 且 tx_dw=1 时,同步头 sync_bits 赋值“6'b000_111”。

数据编码部分,预先设置一个 17 位寄存器 reg_in 用于存储信息位和奇偶校验位。奇偶校验位由信息位逐位异或得到。程序中设置一个使能端 en,初始化为“0”。当 enc_clk 处于上升沿时,若同步头标志 tx_csw 或 tx_dw 存在且使能端 en 为“0”,允许并行输入数据 tx_dword 存入寄存器 reg_in。16 位信息位和 1 位校验位存入寄存器 reg_in 后,进行曼彻斯特编码,并在编码后的数据头部加入同步头 sync_bits。本文采用将输入的 17 位数据分别按位取反插入到原数据后的方法,将原来的 17 位数据编码后变成 34 位数据,实现快速曼彻斯特编码。具体程序如下:

```
data_out={ sync_bits,
reg_in[0], ~reg_in[0],
reg_in[1], ~reg_in[1],
.....
reg_in[16],~reg_in[16]
};
```

数据发送部分,在完成输入数据寄存的同一个 enc_clk 上升沿时刻,若同步头标志位为“1”,则使能端 en 变为“1”,在下一个 enc_clk 时钟上升沿触发计数器 cnt 开始计数,用于串行数据的发送。当数据送入输出寄存器 data_out 后,由计数器 cnt 确定发送位,开始在时钟上升沿时刻逐位发送数据。

3 改进后解码模块的实现

曼彻斯特解码模块通过移位寄存器准确匹配同步头完成解码,主要完成以下功能:(1)对接收的数据进行存储;(2)对同步头进行检测,并进行匹配和锁定;(3)对同步头以外的数据部分进行曼彻斯特解码,完成解码后将串行数据再次转换为并行数据,并进行奇偶校验。图 4 为曼彻斯特解

码模块的逻辑框图。

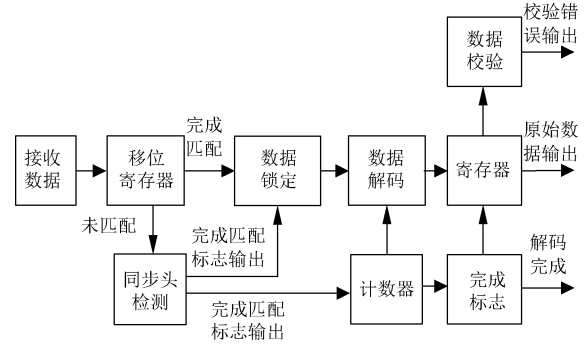


图 4 曼彻斯特解码模块逻辑图

同步头检测和锁定是正确解码的关键。为了保证采样的准确性,解码模块采用 8 MHz 的采样时钟。由于同步头并非曼彻斯特波形,理论上只需检测接收数据中是否存在与命令字/状态字匹配的数据,就可以判断是否检测到同步头。但为了避免采样时亚稳态问题或传输过程中可能存在的干扰影响,保证判定可靠,还需检测同步头之后的首位数据是否存在边缘跳变,只有两者同时成立时才能判定检测到同步头而完成匹配。因此接收端设置了 24 位的移位寄存器 sync_shift_reg 和 8 位的移位寄存器 data_shift_reg,分别用于检测同步头及同步头之后的首位曼彻斯特码数据。两个移位寄存器设置如下:

```
data_shift_reg <= { data_shift_reg[1:7],
rx_data };
sync_shift_reg <= { sync_shift_reg[1:23],
data_shift_reg[0] };
```

即,由时钟触发,data_shift_reg 接收的数据从低位输入并向高位移动,data_shift_reg 的最高位数据向 sync_shift_reg 的最低位移动,并不断向 sync_shift_reg 高位移动。结合解码模块采样时钟的设置,用于同步头检测的具体程序如下:

```
data_edge=data_shift_reg[1]^
data_shift_reg[5];
sync_csw=(sync_shift_reg==
24'hFFF_000)&.data_edge;
sync_dw=(sync_shift_reg==
24'h000_FFF)&.data_edge;
```

当移位寄存器 sync_shift_reg 内存储的数据为“24'hFFF_000”或“24'h000_FFF”,且移位寄存器 data_shift_reg 的第 2 位和第 6 位相异,即存

在边沿跳变时,则判定检测到编码模块发送过来的同步头,完成同步头匹配并锁定。

在解码环节,当同步头被匹配并锁定后,开始进行曼彻斯特解码。由于解码模块采样时钟频率为 8 MHz,是编码模块时钟频率的 4 倍。解码模块中对曼彻斯特编码进行采样,每 8 次采样代表未编码原始数据的一位。利用移位寄存器 `data_shift_reg` 以及曼彻斯特编码的特性,根据时钟信号每移位 8 次取一次移位寄存器 `data_shift_reg` 最高位的方法,将原始数据还原并依次存入一个新的寄存器。同时利用计数器,当计数器被计满,所有数据被解码还原后,对新寄存器中的数据进行串行转并行处理,然后输出。

4 仿真验证

4.1 编码模块

图 5 为编码模块的仿真图。编码时钟信号 `enc_clk` 频率为 2 MHz,复位信号 `rst_n` 低电平有效;`tx_csw` 和 `tx_dw` 分别代表命令/状态字和数据字标志,当 `tx_csw` 或 `tx_dw` 为“1”时,代表发送数据为命令/状态字或为数据字;`tx_dword` 为并行发送的数据内容;`tx_data` 为经过曼彻斯特编码后的数据内容。

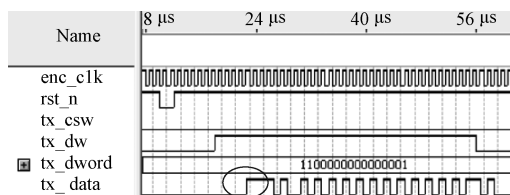


图 5 编码模块仿真结果

由图 5 可知,当 `tx_dw` 为“1”时,同步头输出为“000111”(图 5 中画圈处)。当输入数据为“11000000000000001”,经奇偶校验和曼彻斯特编码后,输出数据为“10100101010101010101010101010101”,实现了编码模块的功能。

4.2 解码模块

图 6 为解码模块的仿真图。解码时钟信号 `dec_clk` 频率为 8 MHz;`sync_csw` 和 `sync_dw` 分别为命令/状态字和数据字同步头的匹配标志;`rx_csw` 和 `rx_dw` 分别为接收端命令/状态字和数

据字标志;`dword_out` 为解码过程中的数据变化,可以检验解码过程中数据的情况;`data_sample` 为解码过程的采样时钟;`rx_dword_out` 为最终接收并解码后的并行数据;`rx_done` 和 `rx_perr` 分别为接收完成标志和校验错误标志。当 `sync_dw` 输出高电平脉冲时,表示数据字同步头匹配完成,可以进行解码;随后采样时钟 `data_sample` 为上升沿时,按照解码规则采样移位寄存器中的数据,并存入新的寄存器中;最终采集的数据将全部送给 `rx_dword_out` 进行并行输出。当 `rx_dword_out` 输出最终解码的数据“11000000000000001”后,数据字标志 `rx_dw` 输出高电平脉冲表示输出为数据内容,同时接收完成标志 `rx_done` 输出高电平脉冲表示完成解码数据接收。校验错误标志 `rx_perr` 为低电平时表示没有错误。

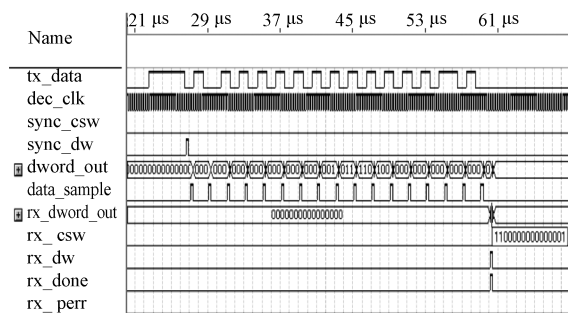


图 6 解码模块仿真结果

5 结束语

本文对 1553B 总线的编码和解码模块进行了设计与改进。改进后的编码模块舍弃了传统的逐位进行曼彻斯特编码的方式,采用寄存器按位统一编码的方式,提高了整体的编码效率。解码模块有效利用移位寄存器对数据进行检测,完成同步头匹配,实现了快速解码并提高了解码的可靠性。系统采用硬件描述语言实现,经软件仿真测试符合设计要求,为后续系统的设计与改进提供了基础。

参考文献

- [1] 刘士全, 隽扬, 蔡洁明, 等. 1553B 总线应用发展研究[J]. 电子与封装, 2013, 13(12): 12-15.