

HDLC 数据帧并行搜帧解封装模块的设计与验证

钱 勇, 刘 威

(武汉大学 物理科学与技术学院, 湖北 武汉 430072)

摘 要: HDLC 信号链路是国际标准化组织(ISO)制定的高级数据链路的控制规程(High Level Data Link Control, HDLC)。遵循 HDLC 标准数据链路层规范,采用硬件描述语言 Verilog HDL 实现了一种基于并行结构的 HDLC 搜帧解封装电路,并采用 System Verilog 技术搭建验证平台,随机生成 HDLC 数据帧来验证设计正确性。使用 Modelsim 软件仿真波形,在仿真过程中,对于净荷区数据长度为 10 个字节的 HDLC 数据帧,解码器电路工作完成需要 16 个时钟周期,兼顾了处理速度和灵活性。使用 QuartusII 软件综合,在 Altera CycloneV 器件上,电路使用了 8 块自适应逻辑模块 ALM, 24 个寄存器,35 个引脚。

关键词: HDLC 协议;搜帧解封装;System Verilog;Modelsim

中图分类号: TN702

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.211472

中文引用格式: 钱勇,刘威. HDLC 数据帧并行搜帧解封装模块的设计与验证[J]. 电子技术应用, 2022, 48(1): 80-83.

英文引用格式: Qian Yong, Liu Wei. Design and verification of HDLC data frame parallel search and decapsulation module[J]. Application of Electronic Technique, 2022, 48(1): 80-83.

Design and verification of HDLC data frame parallel search and decapsulation module

Qian Yong, Liu Wei

(School of Physics Science and Technology, Wuhan University, Wuhan 430072, China)

Abstract: The HDLC signal link is the high level data link control(HDLC) developed by the international organization for standardization(ISO). The article follows the HDLC standard data link layer specification, uses the hardware description language Verilog HDL to implement a parallel structure-based HDLC frame search and decapsulation circuit, and uses System Verilog technology to build a verification platform, and randomly generates HDLC data frames to verify the correctness of the design. Using Modelsim software to simulate waveforms, during the simulation process, for HDLC data frames with a payload area of 10 bytes, the decoder circuit requires 16 clock cycles to complete the work, taking into account processing speed and flexibility. Using QuartusII software synthesis, on Altera CycloneV devices, the circuit uses 8 adaptive logic modules ALM, 24 registers, and 35 pins.

Key words: HDLC protocol; frame search and decapsulation; System Verilog; Modelsim

0 引言

大容量数据的高速传输是通信领域研究的热点问题,其中专用短程通信技术(DSRC)^[1]遵循开放系统互连(OSI)模型,它分为三层结构模型,即应用层、数据链路层、物理层。其中数据链路层在物理层和应用层中间需要上下提供接口和服务,通常人们采用高级数据链路控制协议 HDLC。HDLC 是一种用在数据链路层的高级协议,在通信领域,它的用途最广泛,其特征是有强大的差错检测 CRC 校验^[2]、流量控制和同步传输的功能,能应用于任何高速数据传输系统中。

目前解析和生成 HDLC 协议帧有很多方法,比如软件方法,使用 C 语言在 STM32 上实现 HDLC 通信接口^[3-5],市场上 HDLC 协议处理的专用 ASIC 芯片,还有 FPGA 实现方法^[6-8]。

软件编程灵活但是实现的系统性能低下,只能用于

个别路数的低速信号处理^[9]。专用 ASIC 芯片的成本较高而且由于 HDLC 协议标准文本较多,ASIC 芯片的专用性强而应用灵活性差。对比之下基于 FPGA 实现的 HDLC 解码器可以通过软件反复编程使用,可以兼顾处理速度和灵活性^[10-14]。

1 HDLC 协议简介

HDLC 协议是一种面向比特的数据链路层协议,它可以为全双工点对点数据传输提供完整的数据透明度。数据链路层协议把数据分段为块或包的数据进行逻辑传输,这种格式由起始标志开始至终止标志结束,也称为帧,帧也就是数据的传送基本单位。而数据链路层协议的代表 HDLC 协议具有以下特点:不依赖于任何一种字符编码集;有 0 比特插入机制,数据报文透明传输;有较高的数据传输效率;净荷数据结尾处有循环冗余校验码,确保可靠传输。

1.1 帧头帧尾定帧机制

在比特流数据中使用前后导码的格式,在数据流中净荷与净荷之间有插 1 处理,净荷头部有 ff7e 标志位,净荷尾部有 7eff 标志位,如果解码器模块收到多个连续的 0x7E 数据,则以最后一个 7E 作为数据净荷的前导码,表示数据的起始位置。在模块找到了前后导码位置后,就可以确定一个完整的数据帧。

在本文的并行搜帧设计中,使用一个 16 位的寄存器来缓冲输入端输入的 8 bit 数据,使用组合逻辑电路来判断 7e 在寄存器中的位置,将偏移量记作 bias_0, bias_1, bias_2, bias_3, bias_4, bias_5, bias_6, bias_7, 它们中会有一个被赋值为非零值,然后其他偏移量赋值为 0,主要设计代码如下:

```
assign bias_0=(data_cash[15:8]==8'b01111110)?8'
                                b1000_0000:8'b0000_0000;
assign bias_1=(data_cash[14:7]==8'b01111110)?8'
                                b0100_0000:8'b0000_0000;
assign bias_2=(data_cash[13:6]==8'b01111110)?8'
                                b0010_0000:8'b0000_0000;
assign bias_3=(data_cash[12:5]==8'b01111110)?8'
                                b0001_0000:8'b0000_0000;
assign bias_4=(data_cash[11:4]==8'b01111110)?8'
                                b0000_1000:8'b0000_0000;
assign bias_5=(data_cash[10:3]==8'b01111110)?8'
                                b0000_0100:8'b0000_0000;
assign bias_6=(data_cash[9:2]==8'b01111110)?8'
                                b0000_0010:8'b0000_0000;
assign bias_7=(data_cash[8:1]==8'b01111110)?8'
                                b0000_0001:8'b0000_0000;
always@(posedge clk or negedge rst_n)
begin
if(!rst_n)
    bias<=8'b0;
else
    bias<= bias_0 || bias_1 || bias_2 || bias_3 || bias_4
            || bias_5 || bias_6 || bias_7;
end
```

1.2 净荷区“0”比特插入机制

在帧头帧尾标志位之间的净荷比特数据流中,如果出现了和帧头帧尾 0x7e 一样的比特数据,那么电路模块很容易判断它为数据帧的边界,更严重地还会导致后面接收的净荷数据出错。所以一般在工程应用中,人们在使用 HDLC 协议时,会在净荷区使用 0 比特插入机制来避免这种边界检测错误。当净荷数据中出现连续 5 个 1 时,选择在 5 个 1 数据的后面人为地插入一个 0,在工程应用中解码器模块需要考虑这个数据 0 的存在,解码时删除这个数据。在工程中人们一般采用并串转换,然后对串行数据进行删 0 处理,但是在并行解码设计中,使用串行处理会使用高频率的时钟,这个对电路设计是

很大的挑战,所以在此不考虑插 0 处理。

1.3 CRC 校验机制

HDLC 数据帧的每一帧数据后都有人为加入的帧校验序列,以确保数据传输的正确性。HDLC 帧的发送端和接收端事先约定好校验公式,也就是生成多项式,本文采用循环冗余校验码 CRC 作为校验序列。使用 CRC-CCITT 算法^[13]的生成多项式为:

$$P(x)=x^{16}+x^{15}+x^5+1$$

字节序列的校验码生成过程由发送数据的一方完成,校验码的验证过程由接收数据的一方完成,实质上是一样的,都是对字节序列取余运算,解码器模块对数据序列和校验码的整体进行计算,最后得到的结果为某一特定值则数据传输正确,否则数据传输出错^[12]。

根据并行计算 8 位(单字节序列)二进制序列的数学模型(模块的输入数据端为 8 位数据)^[3,15],此时的校验码的长度为 16 位。二进制序列[b1b2b3b4b5b6b7b8]的校验码计算结果为:

$$R = \sum_{i=1}^8 R_i$$

对于 $R_i=0$ 或者 R_i' 。 R_i' 可以提前计算:

$$R_1' = 0x9188 \quad R_2' = 0x48C4$$

$$R_3' = 0x2462 \quad R_4' = 0x1231$$

$$R_5' = 0x8108 \quad R_6' = 0x4084$$

$$R_7' = 0x2042 \quad R_8' = 0x1021$$

然后对于长度为任意位比特数据的校验码计算:

$$R_i(x) = \text{Re} \left[\frac{M_i(x)}{G(x)} \right] = \text{Re} \left[\frac{x^8 R_{i-1}(x) + m_i(x)}{G(x)} \right]$$

其中 $M_i(x)$ 为多字节的序列对应的多项式, $G(x)$ 为生成多项式, $m_i(x)$ 为 $M_i(x)$ 比 $M_{i-1}(x)$ 多的 8 比特数据, $R_{i-1}(x)$ 为 $M_{i-1}(x)$ 序列的多项式除法余数。当前的 CRC 余数的值只和信息码的当前 1 个字节的输入值 $m_i(x)$ 和前一状态计算出的余数值 $R_{i-1}(x)$ 有关。则并行 CRC 运算逻辑框图如图 1 所示。

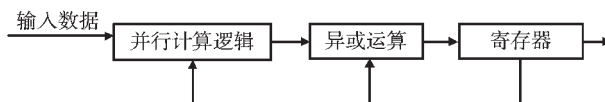


图 1 并行 CRC 运算逻辑框图

2 整体的解码器电路系统

整体的模块框图如图 2 所示,解码器电路主要由 3 个模块构成,8 bit 数据首先进入 search 模块,搜索到帧头帧尾标志给出偏移量信号到下级模块,dec_frame 模块负责把帧头帧尾信号去除,给出 vld 指示信号和净荷数据信号到下级模块,最后是 CRC_dec 模块,对 8 bit 的净荷数据进行并行 CRC 校验计算,向外部输出计算结果,如果传输过程可靠的话最终计算出 CRC_out 为确定值。

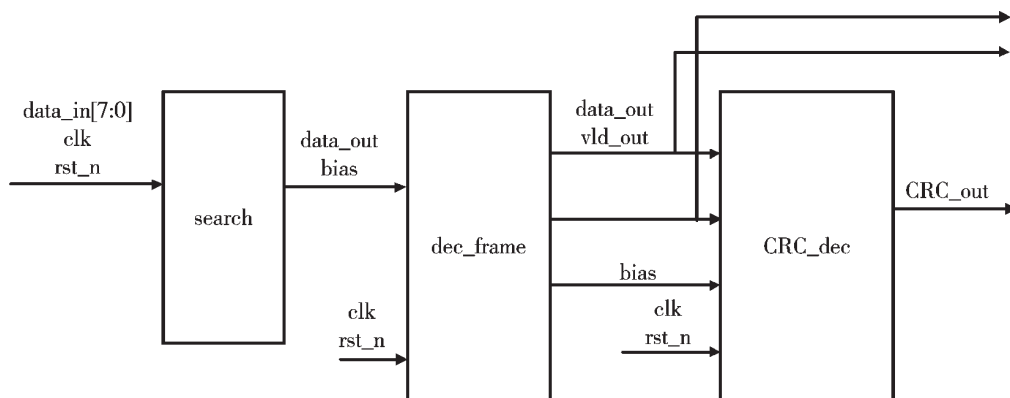


图2 HDLC 解码器框图

综合是设计流程中的重要环节,综合结果的优劣直接影响布局布线结果的最终效能。好的综合器能够使设计占用芯片的物理面积最小,工作效率最高,这也是评定综合器优劣的两个重要指标。HDLC 解码器模块的 Verilog 语言综合由 Altera 公司的集成开发工具 QuartusII 完成,图3所示最后的实现平台采用 Altera 公司的 CycloneV 器件,综合结果使用了8块 ALM,24个寄存器,35个引脚。

Flow Summary	
Flow Status	Successful - Wed Mar 03 13:15:00 2021
Quartus II 64-Bit Version	13.0.0 Build 156 04/24/2013 SJ Full Version
Revision Name	top_design
Top-level Entity Name	top_design
Family	Cyclone V
Device	5CGXFC7C6F23C7
Timing Models	Preliminary
Logic utilization (in ALMs)	8 / 56,480 (< 1 %)
Total registers	24
Total pins	35 / 268 (13 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total HSSI PMA TX ATT Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

图3 QuartusII 软件综合报告

3 使用 System Verilog 搭建验证平台

在使用 Verilog 编写测试文件时,设计者难以对测试环境进行分层设计,这不利于多个测试用例同时测试。而 SystemVerilog 技术则采用面向对象编程(Object Oriented Programming, OOP) 的思想来完成验证平台的分层建模。与硬件描述语言 HDL 相比,SV 语言具有以下性质:(1)可以产生受约束的随机激励;(2)功能覆盖率统计;(3)面向对象编程具有更高的层次结构;(4)支持多线程运行及线程之间可通信;(5)支持 HDL 的数据类型;(6)集成了事件仿真器来方便对逻辑设计施加控制。

使用接口连接待测试的解码器模块和验证平台。在接口中声明了信号位宽和类型,同一时钟域的信号放在

同一个 clocking 里面,drv 是 driver 的时钟域,mon 是 monitor 的时钟域。在 driver 驱动器类里面,测试平台实现对设计 DUT 的信号驱动,通过带有时钟块的接口,在时钟的上升沿或者下降沿触发对信号的驱动。接口 Interface 定义如下:

```
interface pkt_if(input clk, rst_n);
    logic [7:0] data_in;
    logic [7:0] data_out;
    logic vld_out;
    logic [15:0] crc_out;
```

```
.....
```

```
endinterface
```

搭建验证平台编写了 environment.sv, interface.sv, pkt_data.sv, pkt_driver.sv, pkt_gen.sv, test.sv, top.sv 等文件,在 pkt_data.sv 文件里使用随机 bit 队列来生成一个长度随机、内容随机的净荷数据,然后调用队列 queue 的 push_back 方法来插入帧头帧尾、帧间间隔和 CRC 校验码,其中两个字节的 CRC 校验码由软件代码计算得到。

```
rand bit [7:0] data_q[$];
constraint data_size_cons{
    data_q.size() == pkt_len;
};
```

在搭建好验证平台后,根据应用中的对设计的硬件规范,制定验证计划,从需求说明书中提取测试点,根据测试点设计测试用例,编写用例代码,完成仿真过程。跑完测试用例后,需要收集功能覆盖率。对于这个设计,测试用例有:(1)冒烟用例,数据格式正确的随机长度 HDLC 帧;(2)CRC 校验编码错误的 HDLC 错帧;(3)HDLC 数据域长度异常帧;(4)HDLC 帧头帧尾标志缺失的错帧。

4 Modelsim 波形仿真结果与分析

Modelsim 波形仿真结果如图4~图6所示。

将随机的8位比特数据流输入到设计的模块中,首先经过第一个模块 search,当搜索到帧头7e结构时,模块给出偏移量 bias 到下级 dec_frame 模块,最后数据进入 CRC_dec 模块,给出计算结果。在图中的 data_in 数据流中,发送到待测设计的数据帧帧头帧尾都是8位对齐

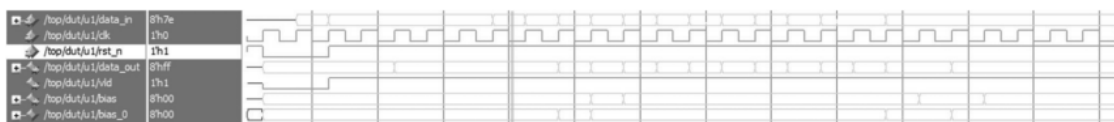


图 4 搜帧模块波形图

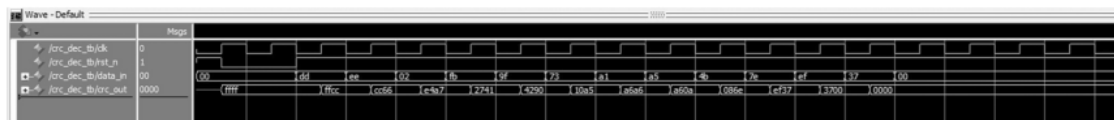


图 5 CRC 校验模块波形图

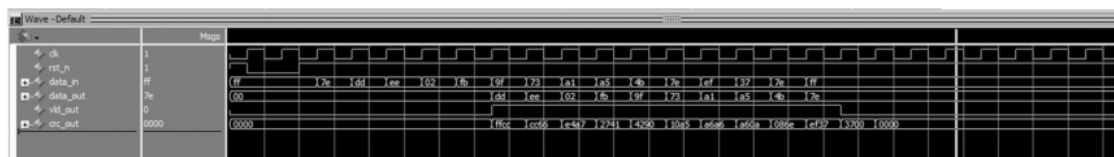


图 6 整体模块波形图

的,所以搜帧时 `bias_0` 会被赋值为 `1000_0000`。

使用 System Verilog 搭建验证平台可以满足测试激励的验证完备性,同时测试结果也具有可观性^[11, 16]。在使用 SV 语言搭建的验证平台去仿真时,使用 rand bit 产生随机比特数据,约束净荷数据长度为 10 个字节,在验证平台内部使用软件计算 CRC 验证码的方法,计算出 16 位 CRC 校验码为 16'hcf37,把两个字节的校验码插入净荷数据尾部,使用解码器电路解码,在仿真波形图中可以看出,净荷数据输入电路模块除去帧头帧尾后经过 4 个时钟周期输出到 data_out 端,对于一个净荷区 10 个字节长度的 HDLC 协议帧,从帧头 7e 输入模块到净荷尾部输出电路模块且 CRC 校验结束总共需要 16 个时钟周期,在 100 MHz 的时钟下需要耗时 160 ns。

5 结论

高级数据链路控制协议(HDLC)是数据链路层中最常用的协议,它具有强大的差错检测功能和高可靠、高效率、透明传输的特点,能应用于任何面向比特的高速数据传输系统中。本文使用 Verilog HDL 语言设计了基于 HDLC 协议的并行解码器,并搭建了基于 SV 语言的验证平台,输入端的净荷数据波形和输出端波形相同,CRC 校验结果为确定值,从而验证了模块对 HDLC 数据帧解封装的正确性。

参考文献

- [1] 邓昌晟, 刘昱, 李海洋, 等. ETC 系统中 HDLC 协议解码控制器的 Verilog HDL 实现[J]. 微型机与应用, 2017, 36(19): 30-33.
- [2] 张伟. 基于 FPGA 的 HDLC 协议控制系统实现[J]. 指挥信息系统与技术, 2011, 2(2): 48-52.
- [3] 李永强, 郭少雷. 基于 STM32F103 的 HDLC 通信接口设计与实现[J]. 光电技术应用, 2018, 33(5): 44-48.
- [4] 黄致绮. 基于 FPGA 平台的 HDLC 协议的应用与研究[J]. 中国新通信, 2017, 19(1): 98.

- [5] 阎肖鹏,余义德,张洪昶.一种基于单片机的 HDLC 通信卡设计[J].自动化技术与应用,2016,35(10):156-159.
- [6] 田地,杨伟.基于 FPGA 的列车网络接口单元设计[J].数字技术与应用,2016,9(1):188.
- [7] 王琼,李增有.基于 FPGA 技术的异步转同步 HDLC 通信卡实现方法[J].机电信息,2016,21(77):141-143.
- [8] 顾秋萍.基于 PCI 总线的串行通讯控制器的设计[D].南京:东南大学,2016.
- [9] 姜育生,闵卫锋.基于 HDLC 协议的光网络通信分析与实现[J].电子设计工程,2020,28(7):161-164.
- [10] 刘晨曦,杜振环,吴涛,等.用于机车通信的 HDLC 控制器 IP 核设计[J].铁道机车与动车,2019(8):23-25.
- [11] 张凡,王小力.基于形式验证的 HDLC 协议验证[J].中国新通信,2017,19(19):22.
- [12] 彭磊.基于 HDLC 协议的机载通信研究[D].武汉:华中师范大学,2017.
- [13] 张必英.基于 FPGA 的 HDLC 协议控制器的设计[D].哈尔滨:哈尔滨工程大学,2005.
- [14] 党成,黄海生,李鑫,等.基于 E1 链路的 HDLC 成帧解帧处理电路[J].光通信技术,2021,45(2):28-31.
- [15] 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.
- [16] 邱玉泉,曾维,刘世伟,等.基于 System Verilog 的同步 FIFO 的验证平台搭建[J].无线电通信技术,2017,43(6):64-66.

(收稿日期:2021-03-08)

作者简介：

钱勇(1997-),男,硕士研究生,主要研究方向:集成电路设计。

刘威(1979-),通信作者,男,教授,主要研究方向:集成电路、人工智能,E-mail: wliu@whu.edu.cn。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所