

多通道高速 HDLC 处理器的设计与实现

陆园琳, 乔庐峰, 王志功

(东南大学射频与光电集成电路研究所, 江苏南京 210096)

摘 要: 本文详述了由一个具有分时处理能力的 HDLC 处理器对 128 逻辑通道数据进行高速、并行、实时处理的设计与实现过程, 并讨论了其实现关键技术, 给出了系统中关键结点的功能仿真波形图。

关键词: 高级数据链路控制规程; 通信协议; 时分复用; 现场可编程门阵列

中图分类号: TP368. 1 **文献标识码:** A **文章编号:** 0372-2112 (2003) 12-1632-04

Design and Implementation of a Multi-Channel High Speed HDLC Data Processor

LU Yuanlin, QIAO Lufeng, WANG Zhigong

(Institute of RF- & OE- ICs, Southeast University, Nanjing, Jiangsu 210096, China)

Abstract: This paper describes the design of a multi-channel high speed HDLC data processor which can process 128 logic channel HDLC data simultaneously. Its logic function and communication protocol coherence have been verified successfully by real-time operation system)) via Works through FPGA. In the system, this multi-channel HDLC processor connects with 8 E1 physical links, and all 128 logic channel data separated from 256 timeslots of 8 E1 frames are processed by this single HDLC processor using time multiplex technology. Compared with other communication chips of this type, the above circuit structure takes more advantages in chip resources taking up and channel management.

Key words: HDLC; communication protocol; time division multiplexing; FPGA

1 引言

HDLC(High Level Data Link Control)协议为国际标准化组织 ISO 制订的面向比特的高级数据链路控制规程^[1]。早期, 每一个用户都拥有一个独立的 HDLC 处理器进行数据的处理。出于处理速度的考虑, HDLC 处理器主要由硬件实现。但随着数据业务多用户、大容量的发展趋势, 这种实现方式必然造成大量硬件资源的占用。因而随着 E1 中继线的大量使用^[2,3], 出现了可以同时处理最多 31 个 64K 数据链路的, 采用时分复用方式的 HDLC 处理器, 并将其设计成标准的模块。这样做可以使芯片面积大大缩小, 当需要处理多个 E1 通道时, 就采用多个相同的单元^[4,5]。但随着单一芯片上所需要承载的 E1 支路数急剧增加, 某些设计在单一芯片上实现了同时处理 32 个 E1 通道的处理能力, 此时, 这种方式占用的芯片面积仍是相当大的。为了解决上述矛盾, 我们设计了一种可以同时处理 8 个 E1 通道, 最大 128 个数据链路的芯片模块。本设计中首先对 128 个低速的 HDLC 通道的数据进行缓冲, 然后使用一个具有分时处理能力的高速 HDLC 处理器, 通过时分复用的方式进行数据的处理。该处理器全双工工作, 核心部分由收发独立的 HDLC 处理器、通道配置寄存器和通道状态存储器构成。

这样设计的模块在 ASIC 芯片资源占用和通道管理上都有明显的优势。

由于数据链路层所采用的各种数据包的结构都与 HDLC 的数据结构基本相同, 所以该 HDLC 处理器可以广泛的应用于帧中继系统、PPP 协议数据包的处理^[6]、ISDN、X. 25 数据网、骨干和边缘路由器等各种数据网环境之中。

随着深亚微米工艺技术的发展, FPGA(Field Programmable Gate Array)芯片的规模越来越大, 其单片逻辑门数已超过上百万门, 已可以实现系统集成^[7]。同时它又具有设计开发周期短、设计制造成本低、可实时在线检验等优点, 因此被广泛用于产量小于 50000 片的芯片设计之中。所以本设计中采用 Xilinx 公司的 X2S200(20 万逻辑门)实现上述的多通道高速 HDLC 处理器。

2 HDLC 帧格式

(1) 标志段(FLAG): 数据链路层的数据传输是以帧为单位的, 为了解决帧同步的问题, 即从收到的比特流中正无误的判断出一个帧的边界, HDLC 规定了每一帧都以 FLAG(0x7E)为开头和结尾。

- (2) 地址段:用于标识接收站的地址。
- (3) 控制段:是最为复杂的字段,根据它的定义可将 HDLC 帧分为信息帧、监督帧和无编号帧。
- (4) 信息段:所要传输的数据。
- (5) 帧校验序列:用于对传输数据的检错与纠错。HDLC 规程中采用循环冗余校验码(CRC)。本设计中可采用 CRC216 编码,多项式为 $1 + X^2 + X^{15} + X^{16}$,或用 CRC232 编码,多项式为 $1 + X + X^2 + X^4 + X^5 + X^7 + X^8 + X^{10} + X^{11} + X^{12} + X^{16} + X^{22} + X^{23} + X^{26} + X^{32}$ 。

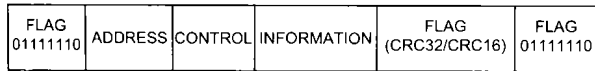


图 1 HDLC 帧格式

3 多通道 HDLC 处理器

多通道高速 HDLC 处理器的实现采用时分复用的方式,核心部分由一个收发独立且可分时处理的 HDLC 处理器、一组通道配置寄存器和一组通道状态存储器构成,其中通道状态存储器是实现时分复用的关键,每一个通道在该通道状态存储器都有一块固定的存储空间,用以存储该通道的数据处理情况,即通道状态信息。多通道高速 HDLC 处理器为每一个通道每一段需要被处理的数据分配一段长度一定的时间片(时间片的长度可根据数据的宽度成比例的变化)。每个时间片结束时,当前通道最新的状态信息(接收方向包括:处理结束时状态机所处的状态、对该通道已经处理过的数据的 CRC 校验码以及已经处理的比特数,当前时间片内已处理完但尚未来得及被输出的接收数据等;发送方向包括:处理结束时状态机所处的状态、对该通道已经处理过的数据的 CRC 校验码、在当前时间片内尚未来得及输出的传输数据等)将被存入通道状态存储器中的相应存储空间。当新的一段数据到达时,此段数据所属的通道在上一个时间片内被刷新状态信息将从通道状态存储器中读出并加载到状态机中,为新一轮的数据处理做准备。

若主机(host)发现某个通道的数据传输有误或长时间未接收到该通道的数据,能且只能在该多通道 HDLC 处理器的等待空闲状态查询该通道的状态信息。

本设计中,128 逻辑通道的 HDLC 数据流通过时分复用的方式全部由一个 HDLC 处理器进行处理,是考虑到:

- (1) 该 HDLC 处理器是以 PCI(Peripheral Component Interconnect)总线上的 33MHz 时钟驱动。
- (2) 该 HDLC 处理器与 8 个并行的 E1(2.048Mbps)物理接口相接,所以 128 逻辑通道的 HDLC 数据流是由最多 8 个物理链路的 E1 帧经时分分配所得。
- (3) 该处理器所处的系统之中对需要进行 HDLC 处理的数据有着足够大的缓冲空间,通道状态信息加载的也足够快。
- 所以一个 HDLC 处理器就足以胜任对 128 逻辑通道 HDLC 数据流的高速、并行、实时的处理工作。若希望获得更快的处理速度,可以采用几个并行的 HDLC 处理器,根据逻辑通道号,将其分配给所属的 HDLC 处理器进行并行处理。

4 单通道 HDLC 收发处理器

图 2 是单通道 HDLC 收发处理器的数据处理流程图。设计中考虑到该 HDLC 处理器面向比特,与物理端和 FIFO 的接口数据却均为并行,所以在 HDLC 数据处理之前先进行并串变换,处理之后再行串并变换。

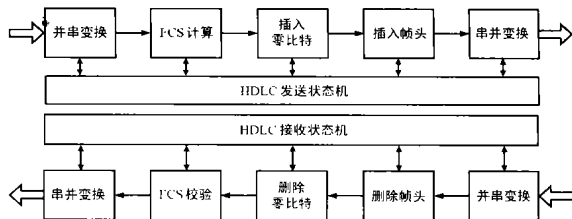


图 2 单通道 HDLC 收发处理器的数据处理流程图

4.1 HDLC 发送处理器

发送方向 HDLC 处理器对每个通道的数据进行帧头插入、CRC 计算、零比特的填充、丢弃序列的插入、帧尾的插入以及连续两帧间多个 FLAG 的插入,状态机如图 3。

为了保证数据链路的透明传输(即可以传输任意组合的比特流),HDLC 处理器在发送端对传输数据进行-0 比特填充。因为传送的数据组合有可能与 FLAG(01111110)相同,为了防止接收端误判断为帧头或帧尾,在一帧的帧头与帧尾之间每当连续传输了 5 个-1 比特之后,插入一个-0 比特,此-0 比特不进行 CRC 计算,而接收端在搜索到帧头后,每收到 5 个连-1 比特,就将其后跟随的一个-0 比特扣除(零比特的去填充)。

当发送端不能及时、完整地传输一帧 HDLC 数据时,将由发送 HDLC 处理器控制插入丢弃序列。设计中丢弃序列由 1 个-0 比特加 7 个连续的-1 比特构成(0x7F)。当物理端口急需某一通道的信息,而 FIFO 中却没有相应通道的数据时,根据通道状态存储器中的 EOP(End of Packet)标志位,发送 HDLC 处理器会为该通道插入相应的丢弃序列或 FLAG。

4.1.2 HDLC 接收处理器

接收方向 HDLC 处理器对每个通道的数据进行帧头搜索、CRC 校验、零比特的去填充、丢弃序列的检测、帧长的监测以及帧尾的搜索。一旦检测到某一帧当前传输的信息位数达到最大帧长或检测到丢弃序列,状态机对当前帧的处理结束,并重新对新一帧进行帧头搜索,而当前帧中的剩余数据将不被处理。状态机如图 4。

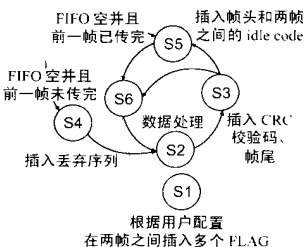


图 3 单通道 HDLC 发送处理器的状态机

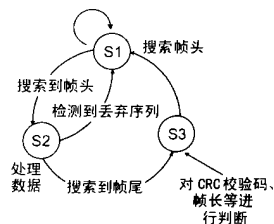


图 4 单通道 HDLC 接收处理器的状态机

5 多通道 HDLC 处理器的仿真与验证

本设计采用 Verilog HDL 语言进行 RTL 级的描述, 在 Xilinx Foundation3.1 的环境下进行语言的编译、逻辑的综合、功能的仿真、布局布线以及时序的仿真。由 Xilinx 公司的 X2S200 实现, 并在时实操作系统 VxWorks 中进行了硬件的功能验证。

5.1 关键结点的波形仿真

对通道状态存储器的正确读写操作是实现多通道高速 HDLC 处理器时分操作的关键技术之一。本设计中, 通道状态存储器由 FPGA 片内自带的单口块 RAM(Ram2-dom Access Memory)实现, 它与 128 逻辑通道 HDLC

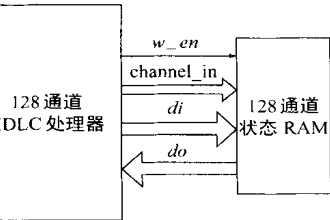


图 5 多通道 HDLC 处理器与通道状态存储器之间的接口信号关系

处理器之间的接口信号如图 5 所示: w_en 为通道状态存储器的写使能, $channel_in$ 为读写该存储器的地址, di 为存储器的输入数据, do 为存储器的输出数据。

图 6 是对通道状态存储器读写时序的仿真波形。图中, 250ns 时第一个请求到达, 通道号为 0x00, 读出 do 为 0 (通道 0x00 的初始状态值)。数据处理完后, w_en 置 1, di 将通道 0x00 的新状态值 0x00001C99088C202 写入。2175us 时通道 0x12 请求, 读出 do 为 0 (通道 0x12 的初始状态值), 数据经过处理后, w_en 置 1, di 将通道 0x12 的新状态值 0x0000361F0038C200 写入。5125us 时通道 0x00 再次请求, 读出 do 为 0x00001C99088C202。可见对通道状态存储器的读写时序是正确的。

由图 7 可见当输入数据 $data_from_tbff$ 为 0x3F 时, 输出 $dataout$ 为 11111 01100, 正确的实现了 -0- 比特的填充功能。

图 8 中 m 为移位寄存器输出使能, $m=0$ 时, 输出数据 $dataout$ 无效。可见当先后输入数据为 0x12F8 时, 输出 $dataout$ 为 00011111001000, 正确的实现了 -0- 比特的去填充功能。

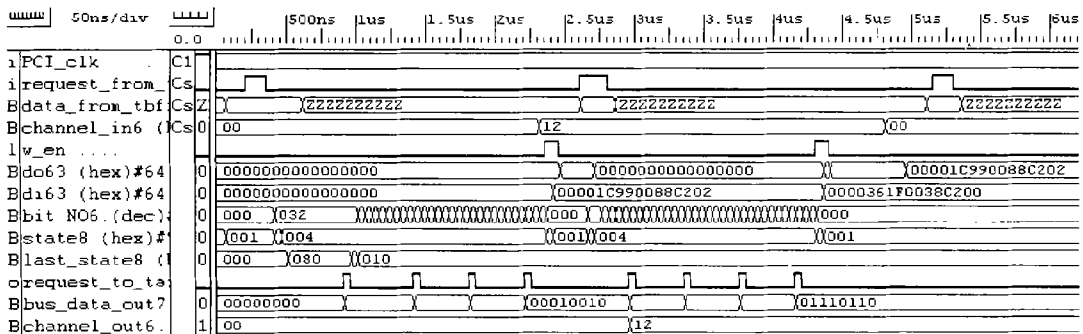


图 6 对通道状态存储器读写的仿真波形

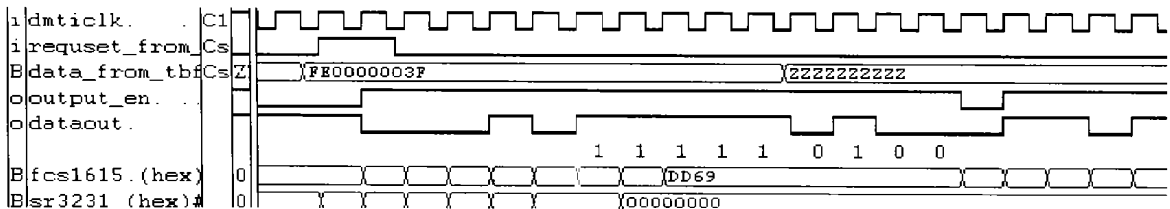


图 7 -0- 比特的填充的仿真波形

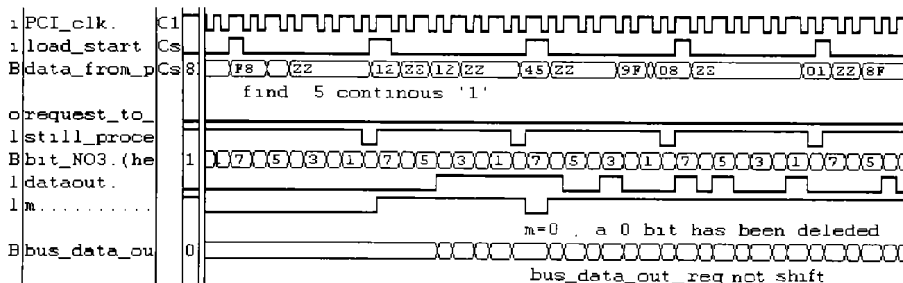


图 8 -0- 比特的去填充的仿真波形

5.1.2 占用资源的比较

在实际系统中, 本模块将与 8 个并行的 E1 (21048Mbps) 物理接口相接。设计中, 综合考虑了 8 个物理端口, 通过时隙

分配表实现 8 个物理链路 (E1 帧) 与最多 128 个逻辑通道之间的相互转换, 并仅由一个 HDLC 处理器通过时分复用的方式实现对 128 通道的数据处理 (方法 1)。而某些同类型通信芯

片如 Rockwell 的 Bt8474 和 Dallas 的 DS3134, 每一个物理链路 (E1 帧) 通过时隙分配表实现与最多 31 个逻辑通道之间的映射关系, 并有一个独立的 HDLC 处理器进行数据的处理, 即若处理 8 个 E1 帧, 需要 8 个 HDLC 处理器(方法 2)。

表 1 是在数据处理接收方向对上述两种实现方法所占 FPGA 资源的比较(Xilinx 公司 FPGA 的每一个可配置逻辑块包含两个 slice), 显然采用本文中的设计方法(方法 1), 通过减少 HDLC 处理器的个数, 可以大大减少 FPGA 的占用资源, 从而减小 ASIC 的芯片面积。

表 1 两种实现方法所占 FPGA 的资源比较

	SLICE		BLOCK RAM (Kbit)
	HDLC 处理器	物理接口	
方法 1	481	2147	20
方法 2	3792= 474x8	2152= 269x8	39

6 结束语

本课题是江苏省科技攻关项目/ 基于 PCI 的高速多通道 HDLC 链路控制器0 的核心模块之一, 目前已通过 FPGA 实现了逻辑功能的验证, 并与 Rockwell 公司的通信芯片 8474 顺利实现了对通。

参考文献:

[1] 谢希仁. 计算机网络(第二版)[M]. 大连: 大连理工大学出版社, 2000. 59- 63.

[2] 郭世满, 叶奕和, 钱德馨. 数字通信))) 原理、技术及其应用[M]. 北京: 人民邮电出版社, 1994. 77- 81.

[3] 肖定中, 肖萍萍. 数字通信终端及复接设备[M]. 北京: 北京邮电学院出版社, 1991. 157- 161.

[4] PM7364(Frame Engine and Data Link Manager)Data Sheet[Z]. PMC2 Sierra, Inc.

[5] Bt8474(Multichannel Synchronous Communications Controller)[Z]. Rockwell Semiconductors Systems.

[6] RFC21662/ PPP in HDLC2like Framing0 Internet Engineering Task Force[S]. July 1994.

[7] 宋万杰, 罗丰, 吴顺君. CPLD 技术及其应用[M]. 西安: 西安电子科技大学出版社, 1999. 1- 2.

作者简介:



陆园琳 女, 1977 年 7 月生于江苏南京, 2000 年至 2002 年, 作为东南大学电路与系统专业的一名硕士研究生, 在东南大学射频与光电集成电路研究所从事超大规模数字集成电路、通信芯片设计工作。



乔庐峰 男, 1971 年 11 月生于河南南乐, 1993 年毕业于南京通信工程学院, 1997 年在该校获得硕士学位, 2000 年至今, 在东南大学射频与光电集成电路研究所攻读博士学位, 主要从事与计算机网络接入、SDH 传输系统有关的超大规模集成电路设计工作。