

# HDLC 协议控制器 IP 核的设计与实现

王雅荣, 鲍民权, 邱智亮

(西安电子科技大学 ISN 国家重点实验室, 陕西 西安 710071)

**摘要:** 介绍了 HDLC 协议控制器的 IP 核方案及实现方法, 分别对发送和接收模块进行了分析, 给出了仿真波形图。该设计采用 Verilog HDL 语言进行描述, 用 ModelSim SE 6.0 进行了功能仿真。

**关键词:** HDLC 协议 IP 核 Verilog HDL 语言

HDLC(High Level Data Link Control)协议是通信领域中应用最广泛的协议之一, 它是面向位的高级数据链路控制规程, 具有差错检测功能强大、高效和同步传输的特点。目前市场上有很多专用的 HDLC 芯片, 但这些芯片功能和接口固定, 不能根据用户的需要灵活地进行改动。

IP 核是一段具有特定电路功能的硬件描述语言程序, 该程序与集成电路工艺无关, 可以移植到不同的半导体工艺中去生产集成电路芯片。利用 IP 核设计电子系统, 引用方便, 基本元件功能的修改容易。随着 CPLD/FPGA 的规模越来越大、设计越来越复杂, 使用 IP 核是一个发展趋势。

本文研究了 HDLC 协议控制器 IP 核的实现方法, 该 IP 核不仅可以用于独立的 IC 芯片, 还可以作为系统的一个子模块直接引用(如网络处理器)。

## 1 HDLC 协议简介

HDLC 是面向位的链路控制规程, 采用帧结构传输数据, HDLC 的帧结构如图 1 所示。

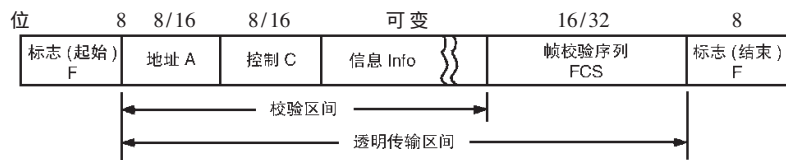


图 1 HDLC 的帧结构

每帧的起始和结束以“7E”(01111110)做标志, 以此来建立帧同步。在帧之间的空载期, 可连续发送标志字或 idle(至少连续 15 个“1”)来做填充。当帧尾收到 abort(连续 7 个到 14 个“1”), 则表示帧异常结束, 并将收到的当前帧丢弃。为了避免将数据中的“7E”误为标志, 在发送端和接收端要相应地进行“插零”及“删零”操作。即发送方在发送除标志字符外的所有信息时(包括校验位), 只要遇到连续的 5 个“1”, 就自动插入一个“0”; 反之, 接收方在接收数据时, 只要遇到连续的 5 个“1”, 就自动将其后的“0”删掉。这样使得 HDLC 具有良好的透明传输。

地址字段用于标识接收站的地址。控制字段实现了 HDLC 的许多重要功能, 根据其最前面两个位的取值, 可将 HDLC 帧划分为: 信息帧、监督帧和无编号帧。信息字段为所要传输的数据。帧校验采用 CRC 算法, 对地址、控制和信息字段进行校验, 插入的“0”不在校验范围。

16 位 CCITT-CRC 的生成多项式为:  $X^{16}+X^{12}+X^5+1$

32 位 CCITT-CRC 的生成多项式为:  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

## 2 HDLC 协议控制器 IP 核的设计与实现

图 2 是 HDLC 协议控制器的系统框图。如图所示, 要发送的数据由 Host Processor 进入发送器的 FIFO, 经过处理从 TxD 输出, RTS\_n 和 CTS\_n 是用于协调何时进行发送的一对握手信号; 接收到的数据在 CD\_n 有效时由 RxD 进入后, 经过接收器的处理放入 FIFO, 并且与 Host Processor 协调后将数据上传。

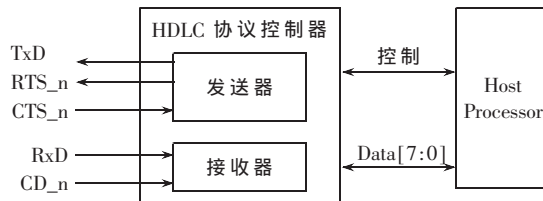


图 2 HDLC 协议控制器的系统框图

### 2.1 总体方案

本设计采用自顶向下的设计方法, 首先根据功能要求设计顶层模块, 再将各个模块逐步细化。如图 3 所示, 该 HDLC 协议控制器由发送(Tx)和接收(Rx)两大模块组成。

Reset 是整个系统的复位信号, Txclk 是发送工作时钟, 而 Rxclk 是接收工作时钟, cpeclk 是 Host Processor 对 HDLC 协议控制器进行数据读写的时钟信号, ENT 置位时发送使能, ENR 置位时接收使能。W 是写使能信号, R 是读使能信号, A[3:0]用来控制读写的数据类型。系统采用 little-endian 模式, 即发送数据时先发送一个字节的低位后发送高位, 接收数据时先接收到一个字节的低

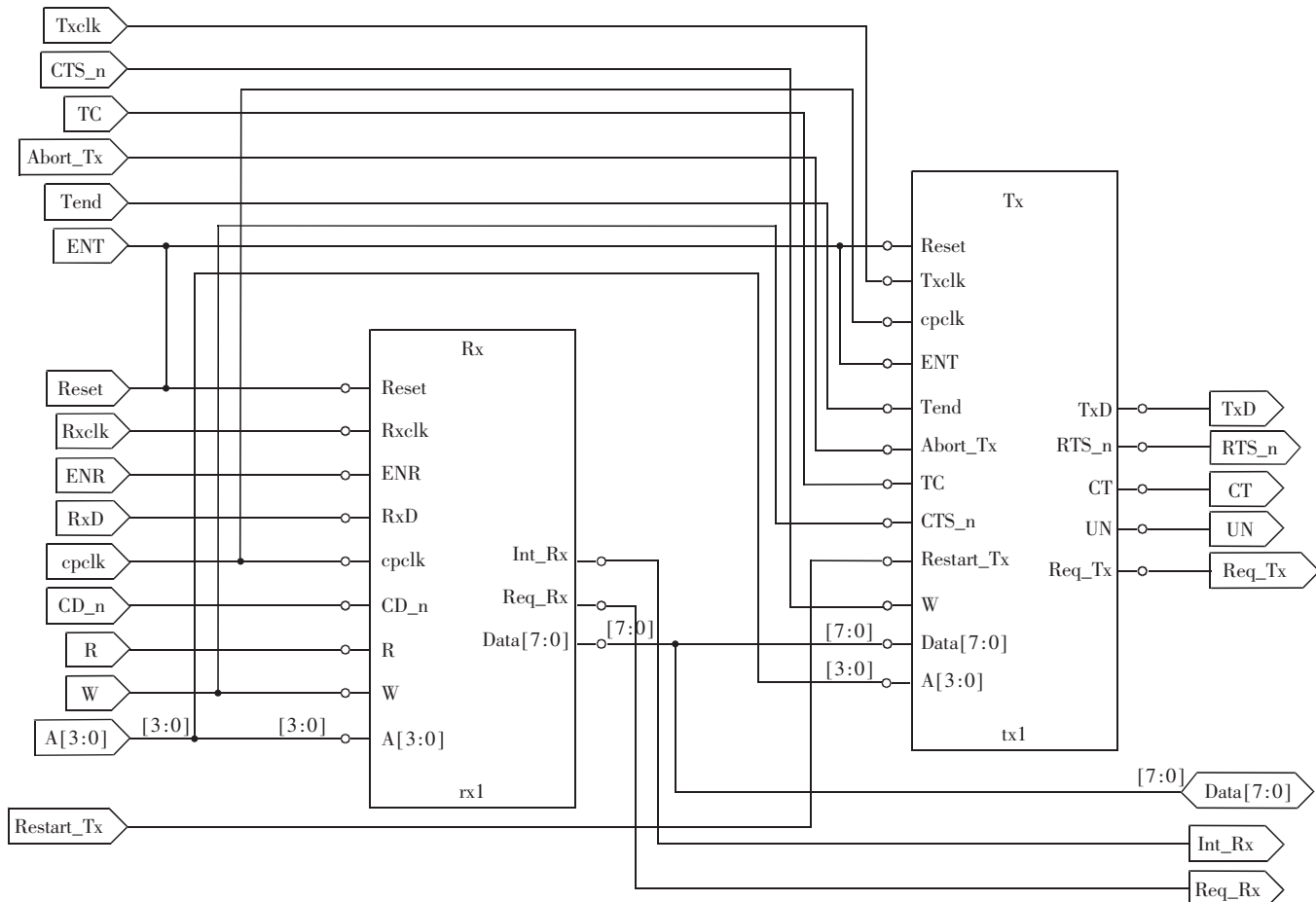


图3 HDLC协议控制器的顶层电路原理图

位后接收到高位。

该系统采用灵活的工作机制,即工作参数可以改变,这些参数在复位后通过 W 置高及 A[3:0]取相应值加载到相应的模块,之后系统按照参数进行工作,这样用户可以根据需要设置相应的工作参数。这些参数有:RTSM,NOF,FSE,MFLR,HMASK,HADDR1,HADDR2,HADDR3,HADDR4。RTSM 指示 RTS\_n 的模式,为 0 时表示:两帧之间 RTS\_n 置为无效,且在帧间发送 idle;为 1 时表示:只要发送使能,RTS\_n 就有效,且两帧之间发送 flag(0x7E)。NOF 在连续传输的两帧之间或传输一帧之前,最少应插入的 flag 个数,其值可以为 0,即两帧无间隔连续传输。FSE 为标志共享使能,若为 1 且 NOF=0,则前一帧的结束标志即为后一帧的开始标志;若 FSE=1 且 NOF>0,则两帧之间插入的标志个数为 NOF-1。MFLR 表示可接收的最大帧(指开始至结束标志之间的所有字节)长度。HMASK 及 HADDR1-HADDR4 用来进行地址匹配,当接收到一帧时,将地址域与 HADDR1-HADDR4 进行比较,其结果再与 HMASK 掩码,HMASK 中为 1 的位即为需要地址比较的位。

发送允许时,若 CTS\_n 有效(低电平有效)且发送 FIFO 有空间,则请求 Host Processor 发送数据,置 Req\_Tx 信号为高。W 为高且 A=0000 时通过 Data 总线(8 位)将

要发送的数据写入发送 FIFO,再通过插零、添加 CRC 校验(TC 为 0 时不添加 CRC 校验而直接发送帧尾 flag)、添加首尾 flag 将数据帧从 TxData 发送出去,其中帧间根据 RTSM 来发送 flag 或 idle。用 Tend 表示当前帧已全部写入 FIFO,当 FIFO 中数据发完时发帧尾 flag。Abort\_Tx 表示将当前帧 abort 掉,此时冲掉 FIFO,发送 0x7F,然后发送 flag 或 idle。CTS\_n 应在发送期间保持有效,否则发生 CTS lost,CT 置高,停止发送。当发送 FIFO 发生 underrun 时,UN 置高,停止发送。Abort,CTS lost 及 underrun 情况下都只有在收到 Restart\_Tx 后才能再开始发送。

接收允许时,若 CD\_n 有效(低电平有效),开始从 RxData 接收数据,并将接收到的数据进行地址匹配,只有当地址匹配时才接收该帧数据,对接收到的数据进行删零后存入接收 FIFO,同时进行 CRC 校验。当接收 FIFO 中有数据时,Req\_Rx 为高,请求 Host Processor 读取接收数据。当 R 为高,A=0011 时,Data 总线上传送的是 FIFO 中的数据;A=0000 时,传送的是接收状态信息;A=0001 时,传送的是帧长高字节部分;A=0010 时,Data 总线上传送的是帧长低字节部分。接收状态信息包括:帧超长,nonoctet-aligned(非 8 的整数倍)帧,abort,CRC 校验错误,FIFO overrun,CD lost,地址不匹配。CD\_n 应在数据接收期间保持有效,否则发生 CD lost。帧超长部分不接收进

入 FIFO, overrun 和 CD lost 情况将停止当前帧的接收。发现 nonoctet-aligned 帧时, 对数据的处理方法为在最后一个字节的非 8 位数据部分后加上一个“1”和多个“0”, 凑齐一个字节, 如图 4 所示。

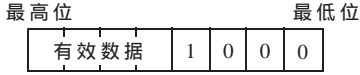


图 4 nonoctet-aligned 帧的最后一个字节填充方法

当一帧接收完全且被 Host Processor 读取或 overrun、CD lost 时, 向 Host Processor 发送 Int\_Rx, 请求 Host Processor 读取接收状态信息和帧长。

### 2.2 发送模块 Tx

图 5 是 HDLC 协议控制器中发送模块 Tx 的模块细化图。

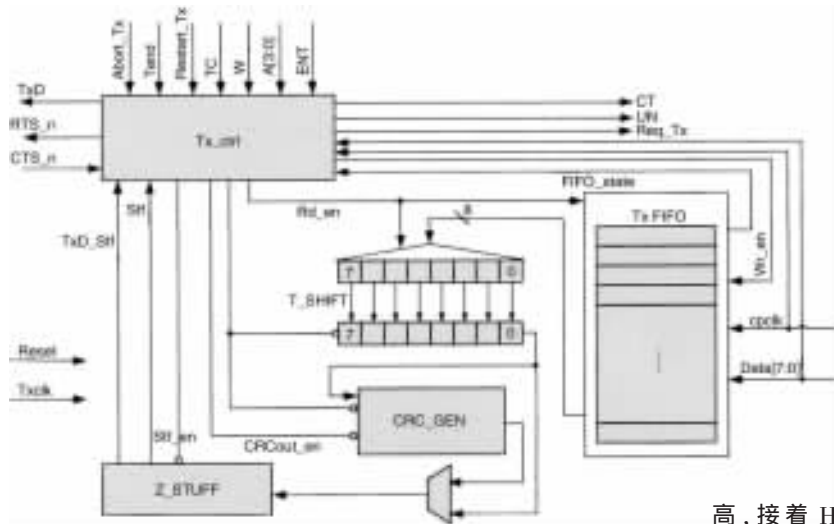


图 5 HDLC 发送模块

如图 5 所示, 发送模块可以细化为五个部分: 发送控制模块 Tx\_ctrl, Tx\_FIFO, 并串转换 T\_SHIFT, CRC 产生 CRC\_GEN, 插零 Z\_STUFF。

Tx\_ctrl 与外部交互, 控制着整个发送过程。要发送的数据进入 Tx\_FIFO, 在 Tx\_ctrl 的控制下, Tx\_FIFO 中的数据进入 T\_SHIFT 进行并串变换, CRC\_GEN 对 T\_SHIFT 输出数据进行 CRC 校验, Tx\_ctrl 控制输出并串转换的数据或 CRC 校验码进入 Z\_STUFF 进行插零处理, 插零后的数据进入 Tx\_ctrl 加上首尾 flag 由 TxD 发送出去。

### 2.3 接收模块 Rx

图 6 是 HDLC 协议控制器中接收模块 Rx 的模块细化图。

接收模块主要可细化为八个模块: 接收缓存 R\_BUFFER, 接收状态检测 State\_DETECT, 删零 Z\_UNSTUFF, 地址检验 ADDR\_CHK, CRC 校验 CRC\_CHK, 串并

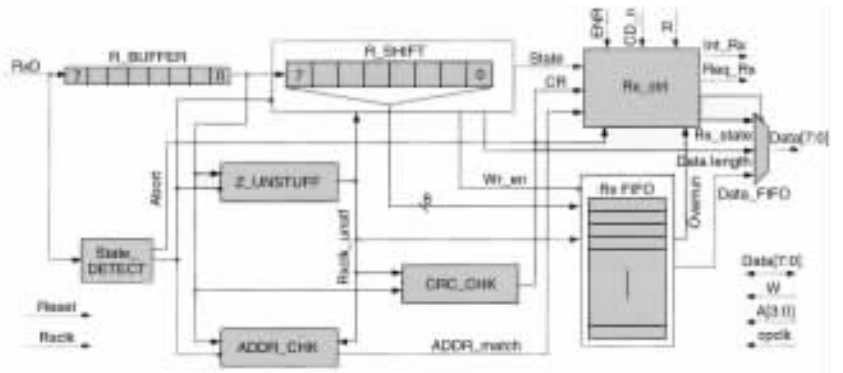


图 6 HDLC 接收模块

变换 R\_SHIFT, 接收控制 Rx\_ctrl, Rx\_FIFO。

Rx\_ctrl 通过与外部交互, 控制着整个接收过程。接收到的数据从 RxD 进入 State\_DETECT 和 R\_BUFFER, State\_DETECT 检测 flag、idle、abort 及数据, R\_BUFFER 引入了 8 个 RxClock 的时延确保在检测到 flag 以后再行删零, 删零后的数据进行地址校验和 CRC 校验, 并将结果反馈给 Rx\_ctrl, R\_SHIFT 将删零后的数据进行串并变换以字节形式写入 Rx\_FIFO, 并统计帧长、检测帧是否超长及是否为 8 位的整数倍, Rx\_ctrl 根据 A[3:0] 的不同从 Data 输出接收到的数据、状态信息或帧长。

### 3 仿真与验证

图 7 是从系统复位到一帧开始发送的仿真波形图。由图可见, 复位后 W 置高, A 取 0000 到 1101 将各个工作参数从 Data 加载, 随后 ENT 有效, 发送请求 Req\_Tx 置高, 接着 Host processor 将要发送数据写入, 待发送数据是 0xFFFFDFCFBFAF9, 以 Tend 置位告知一帧结束。开始 RTS\_n 无效, TxD 是 idle 状态, 当发送器准备好可以发送时, RTS\_n 有效, TxD 先发一个 flag, 接着输出 1111101110111110..., 从而正确实现了“0”位的填充。

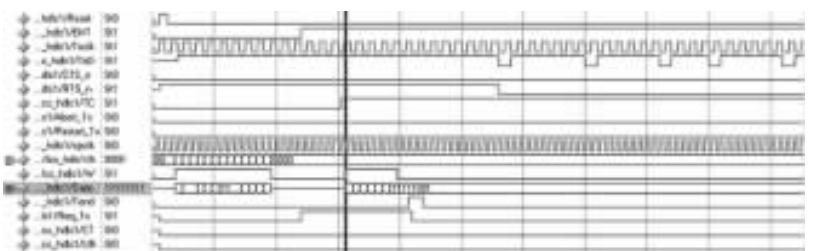


图 7 发送仿真结果

图 8 是接收地仿真波形图。如图, 如果 ENR 有效, RxD 接收到 flag 后开始接收数据, FIFO 中有数据且地址匹配时 Req\_Rx 置位, 随后 R 置位, A 取 0011 来读取数据, 接收到的数据是 1111101110111110110001001..., 输出

(下转第 40 页)

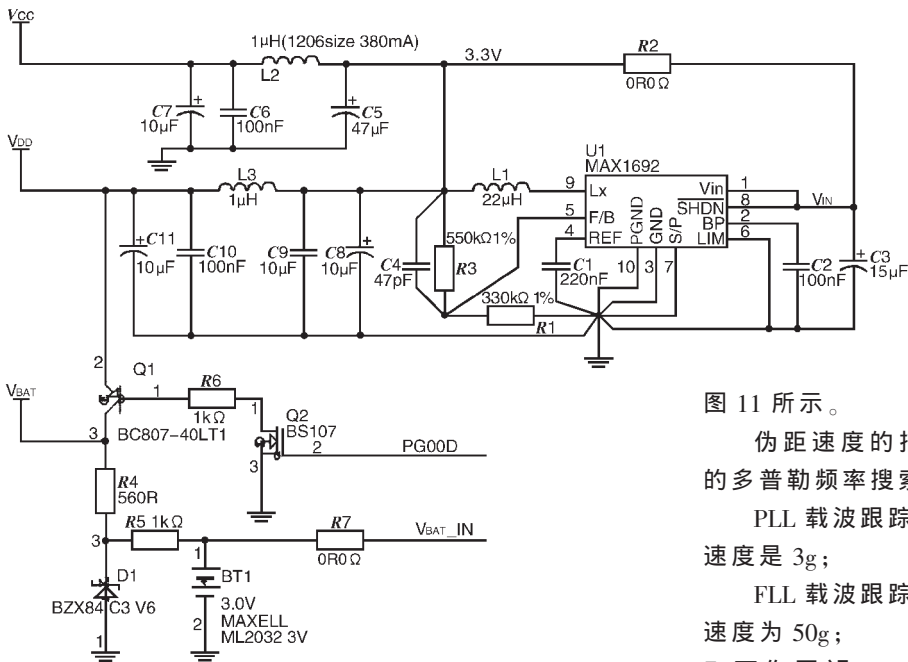


图 10 电源设计电路原理图

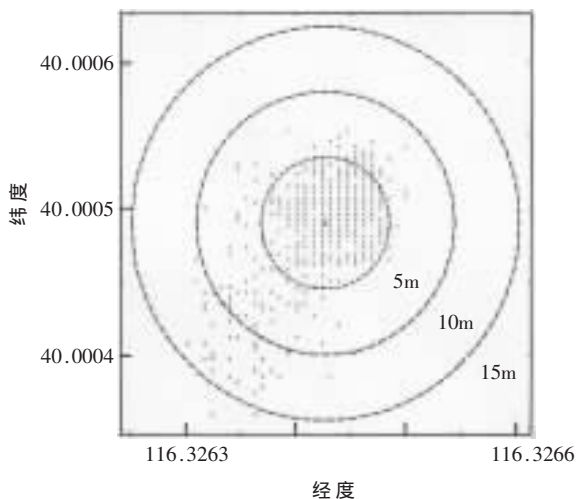


图 11 星载 GPS 接收机地面工作模式的定位结果

数字电路的器件布置在六层 PCB 板上, 双面布置器件。

#### 4 测试结果及分析

测试定位时间为 80 秒。冷启动情况下, 该指标与接收机搜索 GPS 卫星顺序及 GPS 卫星可视情况有关。接收机首先分配 12 个相关器通道搜索 PRN1 到 PRN12 卫星, 当这些卫星多于 3 颗可见时, 在约 30 秒时就能实现定位解算。定位结果如图 11 所示。

伪距速度的捕获范围与应用软件中设置的  $\pm 35\text{kHz}$  的多普勒频率搜索范围吻合;

PLL 载波跟踪环路动态特性差, 能够跟踪的最大加速度是 3g;

FLL 载波跟踪环路动态特性好, 能够跟踪的最大加速度为 50g;

#### 5 工作展望

当前, 系统已经具有完整的接收机硬件和软件平台, 且已经能够在地面应用情况下良好工作。针对航天应用, 还将进行如下工作:

- (1) 验证卫星在轨高速运动情况下的接收机捕获跟踪解算功能;
- (2) 改善定位算法, 提高定位精度;
- (3) 热循环、热真空、振动试验、抗辐射的考虑等。

#### 参考文献

[1] GP4020 GPS baseband processor design manul.Zarlink (Mitel) Semiconductor Company, <http://www.zarlink.com>  
 [2] GP2000 GPS receiver hardware design.Zarlink(Mitel) Semiconductor, <http://www.zarlink.com>  
 [3] ARM Limited.ARM software development toolkit version 2.50 reference guide.

(收稿日期: 2006-08-03)

(上接第 35 页)

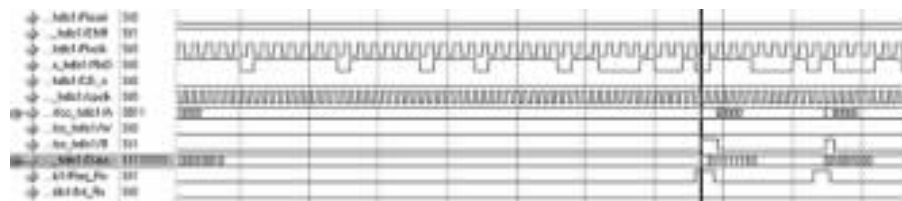


图 8 接收仿真结果

Data 是 11111111\_11111110\_01001000, 从而正确完成了“0”位的填充。

本文提出了一种 HDLC 协议控制器 IP 核的设计方案, 采用 Verilog HDL 语言进行 RTL 级的描述, 用软件

ModelSim SE 6.0 进行了功能仿真。经验证, 该设计在各种情况下都工作正常。限于篇幅, 各种情况的仿真结果不能在这里一一给出。该协议控制器简单灵活, 有效可行, 能够很好地满足一些系统的要求。

#### 参考文献

[1] 谢希仁. 计算机网络. 北京: 电子工业出版社, 2003.  
 [2] BHASKER J 著, 徐振林译. Verilog HDL 硬件描述语言. 北京: 机械工业出版社, 2003.  
 [3] 夏宇闻. Verilog 数字系统设计教程. 北京: 北京航空航天大学出版社, 2003.

(收稿日期: 2006-07-26)