

基于 SOPC 的 1553B-PCI 桥接通信模块设计

刘乐光 王升辉 李进

(北京交通大学 信息科学研究所, 北京, 100044)

摘要: 本文提出了一种基于 Altera SOPC 技术的总线桥接通信模块设计方法, 完成 1553B 总线到 PCI 总线的桥接功能。在 Altera FPGA 平台上, 自主设计了 1553B 总线控制模块, 由 Altera 提供的 IP 核生成 PCI 总线控制模块, 通过 Nios II 处理器完成总线数据交互。整个系统仅由一块 FPGA 和 1553B 总线的通信驱动电路组成, 实现了系统的最小化设计。

关键字: SOPC; 1553; PCI; Altera; Nios; IP 核; 现场可编程门阵列

Design of 1553B-PCI Bridge Communication Module Based on SOPC

Abstract: This paper describes a design of 1553B-PCI Bridge Communication Module based on SOPC. On the Altera FPGA platform, we designed 1553B bus control module with authority, created a PCI bus control module using SOPC Builder which is provided by Altera, implemented data transfer by using Nios II processor. The whole system is made up by one FPGA chip and a 1553 bus driven circuit. Therefore, the minimum system design is implemented.

Keywords: SOPC; 1553; PCI; Altera; Nios; IP Core; FPGA

1. 引言

1553B 总线是七十年代发展起来的“飞机内部时分制指令 / 响应式多路传输数据总线”, 它具有可靠性高、速度快、反应灵敏、双冗余等特点, 特别适用于控制通信种类多、数据量大、实时性要求较高、网络节点少的系统。虽然其传输速率只有 1Mbit/s, 但它并没有被后来发展起来的高速网淘汰, 而是随着微电子技术和计算机技术的发展而不断提高^[1]。

在许多应用 1553B 总线进行通信的网络中, 为了对海量的数据进行处理, 需要计算机参与计算和分析。为了将计算机系统嵌入 1553B 总线通信网络, 使之成为该网络的一个节点, 需要设计一个 1553B 现场总线到计算机局部总线的桥接通信模块来实现总线数据的交互。

PCI 总线是一种与微处理器无关的 32 / 64 位地址数据复用局部总线, 一边与处理器和存储器接口, 另一边为外设扩展提供了高速通道。33MHz、32 位的 PCI 总线可以实现 132MB/s 的数据传输速率^[2]。

本文提出了一种 1553B-PCI 桥接通信模块设计方法，其功能结构框图如图 1。

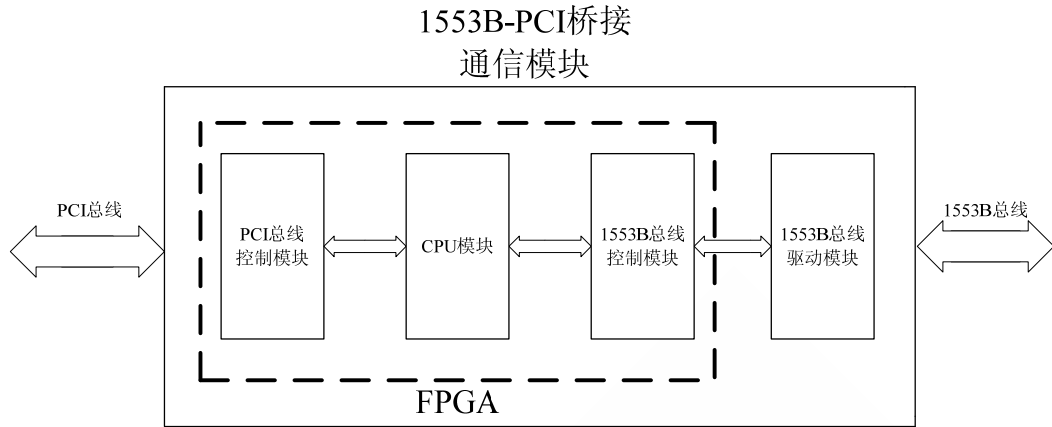


图 1 1553B-PCI 桥接通信模块功能结构框图

2. 1553B 总线控制模块设计

1553B 总线控制模块结构

1553B 总线控制模块系统结构如图 2 所示，实现的具体功能如下：

消息的正确发送：命令字/方式指令的写入、添加同步头、计时、数据字的发送、重发控制、曼彻斯特编码等^[3]。

消息的正确接收：状态字和数据字的接收、同步头识别、奇偶校验、曼彻斯特解码、中断等^[3]。

字/消息的处理：命令字提取、接口寄存器的读写、存储器及缓冲器的读写控制等。

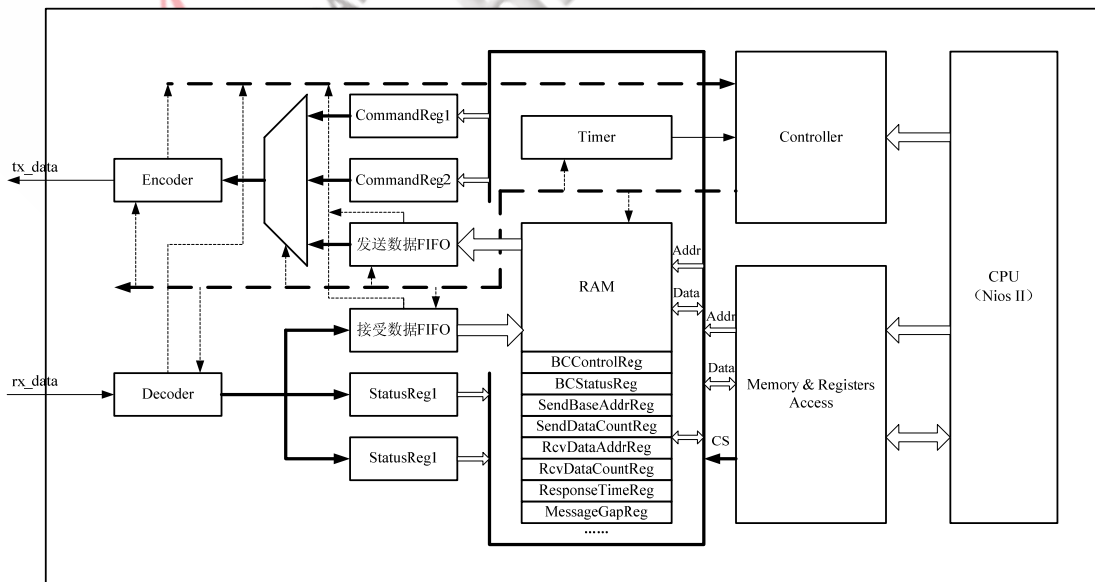


图 2 1553 总线控制器系统结构

1553B 总线控制模块主要分为数据流控制器、曼彻斯特编解码器、存储器及接口寄存器

三个子功能模块。

数据流控制器：包括状态控制器和存储器及寄存器读写控制两个部分。状态控制器是系统的状态机，控制整个系统的时序，负责状态的转移，通过不同的状态来控制数据的流向。存储器及寄存器读写控制主要功能包括两个方面。一是，根据内部状态控制器的状态来控制双口 RAM 和数据发送/接收 FIFO 的读写；二是，根据外部 CPU 的地址译码来读写内部 RAM 和接口寄存器。

曼彻斯特编解码器：包括解码器和编码器两个部分。解码器的主要功能是识别总线上传输数据的同步头，对同步头后面的曼彻斯特编码的串行数据进行接收，转换成并行数据输出，并指示数据有效、字类型、奇偶校验等信息。编码器对并行数据进行曼彻斯特编码，并且根据字的类型添加同步头，然后添加奇偶校验位，最后将编码后的数据以串行的方式发送到总线上^[4]。

存储器及接口寄存器：包括双口 RAM、数据发送/接收 FIFO 和接口寄存器三个部分。2K Bytes 的双口 RAM 用来存放发送的数据和接收的数据，外部 CPU 可以向 RAM 写入需要发送的数据，也可以读出接收的数据。数据发送/接收 FIFO 用来在进行一次消息通信的过程中做发送和接收数据的缓冲。外部 CPU 通过读写接口寄存器来了解总线的状态和控制总线数据的传输。

数据流控制器设计

数据流控制器通过状态机来控制整个总线控制器的时序和数据流向，采用有限状态机设计，状态转移如图 3 所示：

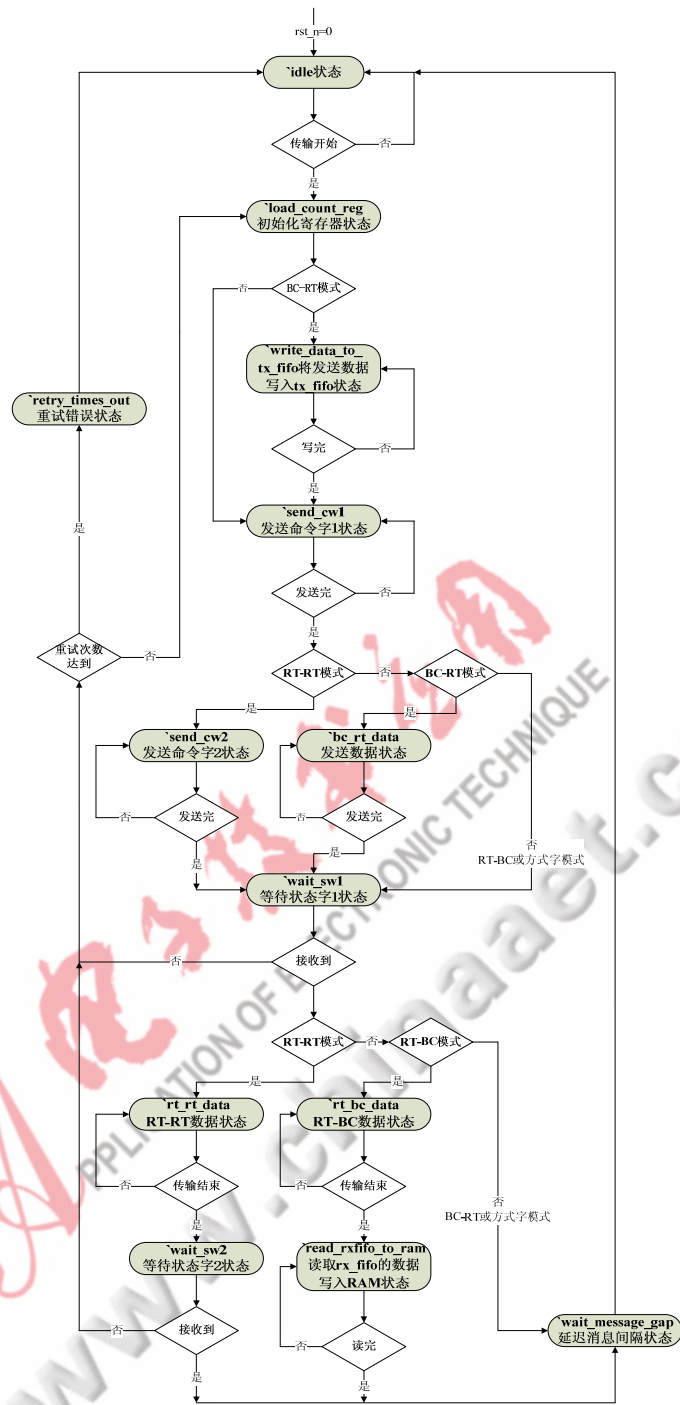


图3 数据流控制器状态机状态转移图

状态机共有 13 个状态，每个状态的具体操作如下：

idle: 等待状态，在系统复位和一次消息传输结束后进入等待状态，在 BC_start 信号有效时，跳出此状态开始一次消息的发送。

load_count_reg: 装载发送数据个数、接收数据个数等计数器，进行发送前的初始化。

write_data_to_tx_fifo: 当消息是 BC (Bus Controller) -RT (Remote Terminator) 类型时，BC 需要发送数据到总线上去。从 RAM 中读取对应地址空间的数据个数到发送 FIFO 中，准

备发送。

`send_cw1: 发送第一命令字, 无论是何种消息都需要发送第一命令字。

`send_cw2: 当是 RT-RT 消息时, 发送第二命令字。

`bc_rt_data: 当是 BC-RT 消息时, 在发送完第一命令字后, 读取发送 FIFO 中的数据使用编码器发送到总线上去。

`wait_sw1: 等待第一状态字, 并且启动定时器, 在设置时间内 ResponseTimeReg(0~65535us)接收到的状态字写入第一状态字寄存器 RxStatusWordReg1 中。如果超时未接收到状态字, 并且重试次数未超过时, 进行重试重新发送。如果重试次数超过则进入重试错误状态`retry_times_out。

`rt_rt_data: 当是 RT-RT 消息时, 等待 RT-RT 之间的数据传输完毕。

`wait_sw2: 当是 RT-RT 消息时, 在 RT-RT 之间的数据传输完毕后, 等待第二状态字, 并且启动定时器, 在设置时间内 ResponseTimeReg(0~65535us)接收到的状态字写入第二状态字寄存器 RxStatusWordReg2 中。如果超时未接收到状态字, 并且重试次数未超过时, 进行重试重新发送。如果重试次数超过则进入重试错误状态`retry_times_out。

`rt_bc_data: 当是 RT-BC 消息时, 解码器从总线上接收的数据需要写入到接收数据 FIFO 中做缓冲。

`read_rxfifo_to_ram: 当是 RT-BC 消息时, 在接收数据完毕后, 需要将接收数据 FIFO 中的数据写入到 RAM 中对应的地址空间中。

`retry_times_out: 当重试次数超过时, 将状态寄存器 BCStatusReg 中的重试出错标志位置位。

`wait_message_gap: 在一次消息传输结束后, 延迟设置的消息间隔时间 MessageGapReg(0~65535us), 延迟时间到达后进入等待状态, 等待系统发起下一次消息。

曼彻斯特编码器

编码器接收来自发送数据 FIFO、第一命令寄存器 CommandWordReg1 和第二命令字寄存器 CommandWordReg2 的 16 位数据, 根据数据的类型添加不同的同步头, 在数据末端添加奇校验位, 然后以串行的方式在设置的波特率下送入发送器, 将数据传输到 1553B 总线上。

编码器模块如图 4 所示。

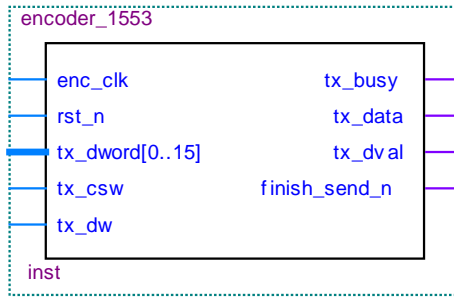


图 4 曼彻斯特编码器

图 5 为编码器发送数据字 16'h00ff 的 Modelsim 仿真波形:

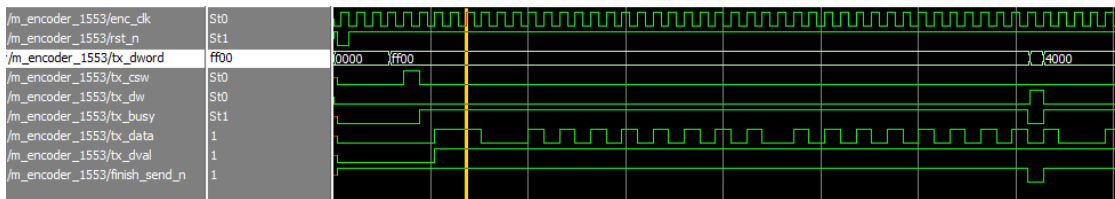


图 5 编码器发送数据字 16'h00ff 的 Modelsim 仿真波形

曼彻斯特解码器

曼彻斯特解码器为 1553B 总线数据接收部分，首先需要对同步头进行提取，判断接收数据类型，将接收的串行数据转换成并行数据，并进行奇校验。

解码器模块如图 6 所示。

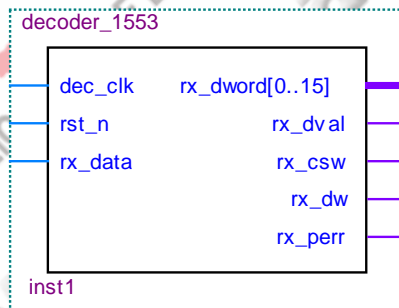


图 6 曼彻斯特解码器

图 7 为解码器对串行数据解码后得到 16'haaaa 的 ModelSim 仿真波形。

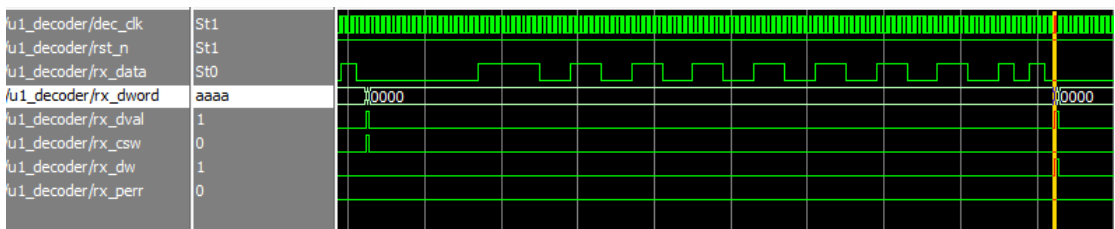


图 7 解码器解码 16'haaaa 的 Modelsim 仿真波形

接口寄存器

外部 CPU 通过接口寄存器来了解总线状态和控制总线数据传输，接口寄存器的描述如

表 1 所示。

Register	Address	Bits	Description
BCControlReg	11'b1000000000	6:5	响应超时重试次数
		4:1	消息类型
		0	总线传输开始位
BCStatusReg	11'b1000000001	6:3	状态机状态
		2	接收数据奇校验出错
		1	重试次数超出出错位
		0	总线忙信号位
CommandWordReg1	11'b1000000010	15:0	第一命令字寄存器
CommandWordReg2	11'b1000000011	15:0	第二命令字寄存器
RxStatusWordReg1	11'b1000000100	15:0	第一状态字寄存器
RxStatusWordReg2	11'b1000000101	15:0	第二状态字寄存器
SendDataAddrReg	11'b1000000110	15:0	发送数据基址寄存器
SendDataCountReg	11'b1000000111	15:0	发送数据个数寄存器
RcvDataAddrReg	11'b1000001000	15:0	接收数据基址寄存器
RcvDataCountReg	11'b1000001001	15:0	接收数据个数寄存器
ResponseTimeReg	11'b1000001010	15:0	响应时间寄存器 (us)
MessageGapReg	11'b1000001011	15:0	消息间隔时间寄存器(us)

表 1 接口寄存器定义

3. PCI 总线控制模块设计

基于 IP 核技术的 PCI 系统设计

PCI 总线控制器的配置非常灵活，可以根据具体的应用配置为 Master/Target 设备或者 Target 设备。Master/Target 设备可以主动访问其他 PCI 总线设备，Target 设备则只能在 PCI 主设备的控制下进行工作。根据使用的总线宽度不同，又可以配置为 32 位总线系统或 64 位总线系统。

Altera 的 PCI 编译器为用户提供了很多选项来创建用户自定义的、高性能的 PCI 接口设计。该 PCI 编译器包含了 pci_mt64 (Master/Target, 64 位), pci_mt32 (Master/Target, 32 位), pci_t64 (Target, 64 位) 和 pci_t32 (Target, 32 位) 功能，同时提供 Verilog HDL 语言

和 VHDL 语言的测试平台，并且有相应的参考设计^[5]。

在 Quartus 软件中，使用者可以通过两种方式来创建自己的 PCI 系统^[5]：

MegaWizard Plug-in Manager 方式：此种方式允许用户选择特定的 PCI MegaCore 功能，设置具体参数，生成设计文件，并手动整合参数化的 PCI MegaCore 功能到用户的系统中。

SOPC Builder 方式：此种方式允许用户使用元件搭接的方式来自动生成一个完整的 PCI 系统。SOPC Builder 通过 Avalon 总线桥将 PCI 总线模块连接到系统中，使用者可以非常容易地创建包含一个或多个 Avalon 总线设备的系统。

使用 SOPC Builder 方式创建 PCI 系统，可以使用 Altera 提供的大量的已有的设计模块，通过 Avalon 总线自动连接成一个完整的系统，并且能够同时生成仿真环境。使用这种方式进行设计可以大大缩短开发的周期，本设计就是采用了 SOPC Builder 方式创建 PCI 控制模块。

PCI 总线控制模块连接如图 8 所示：

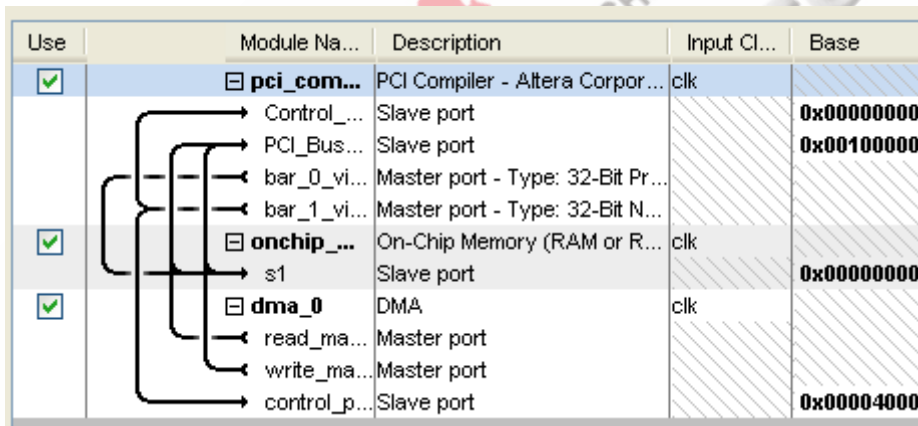


图 8 PCI 控制模块的 SOPC Builder 连接

端口信号如图 9 所示，_pci_compiler 信号为标准 PCI 总线信号，直接连接到 PCI 总线。

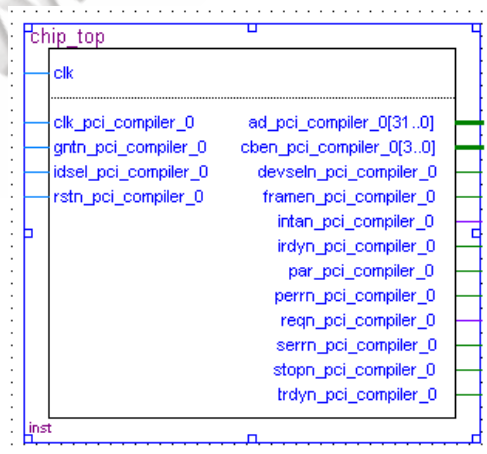


图 9 PCI 控制模块的端口信号

ModelSim 仿真结果如图 10: (配置空间读写单字节数据 0x97)

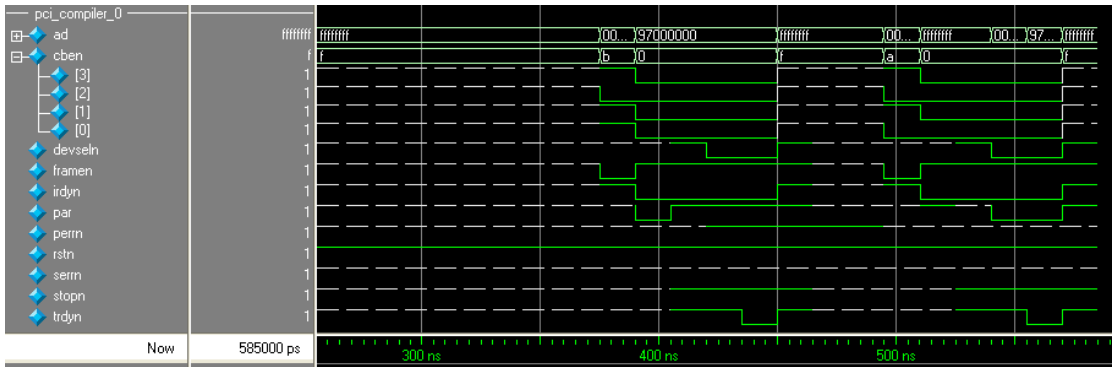


图 10 PCI 总线控制模块 ModelSim 仿真图

4. CPU 模块设计

CPU 模块控制逻辑设计

CPU 模块主要完成总线数据的交互，其功能是通过分别对 1553B 总线控制模块和 PCI 总线控制模块读写数据实现的。PCI 总线控制模块挂接在 Avalon 总线上，可以直接通过总线命令读写数据。对 1553B 总线控制模块的操作通过接口寄存器实现，CPU 模块的控制逻辑设计如图 11 所示：

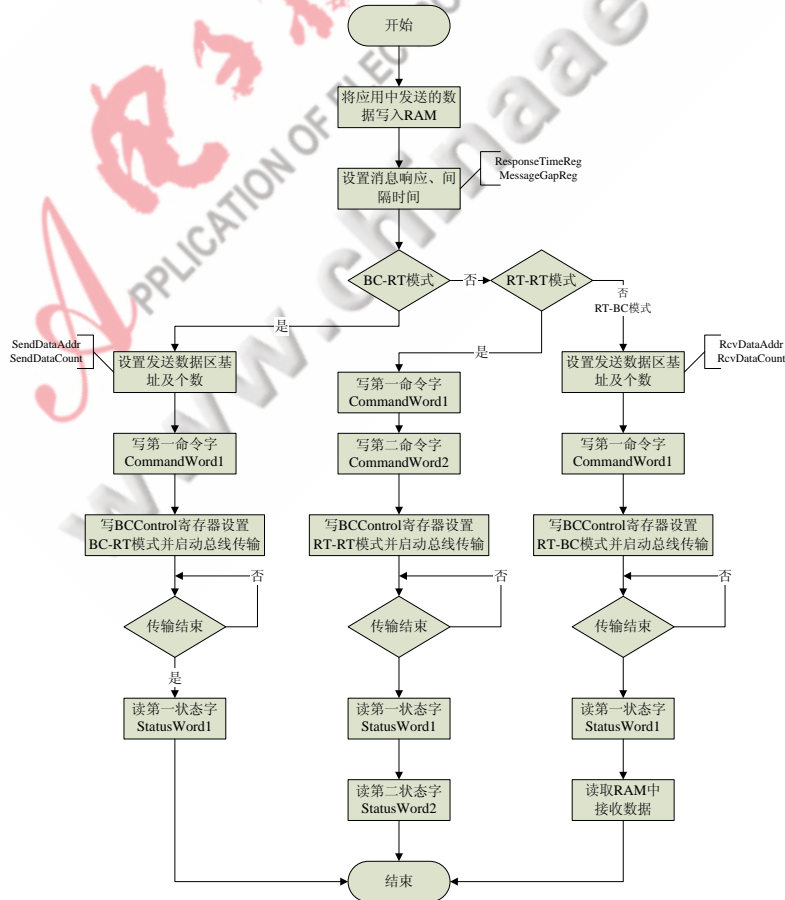


图 11 CPU 模块控制逻辑

Nios II CPU 设计

Nios II 是 Altera 特有的基于通用 FPGA 构架的软 CPU 内核，是一个可灵活配置的通用 32 位 RISC 嵌入式处理器。Nios II 处理器内核有 3 种类型，它们分别是快速型（Nios II/f, Fast）、经济型（Nios II/e, Economy）和标准型（Nios II/s, Standard）^[6]，用户可以根据不同的设计要求来选择具体使用那种处理器内核。

1553B 总线控制器模块的 CPU 接口信号包括读写控制信号和地址、数据信号。读写控制信号包括片选 READY_N 信号、写有效 WE_N 信号和读有效信号 OE_N；地址为 11bit 宽的 ADDRESS[10..0]，数据为 16bit 宽的 data[15..0]。这些接口信号的控制时序与 SRAM 的控制时序相同，因此可以使用 SOPC Builder 中 SRAM 读写控制器 IP 核来控制自主设计的 1553B 总线控制器模块，在 Nios II 中集成 SRAM 控制器 IP 核与 1553B 总线控制模块相连。

Nios II 与自主设计的 1553B 总线控制模块的连接如图 12 所示。

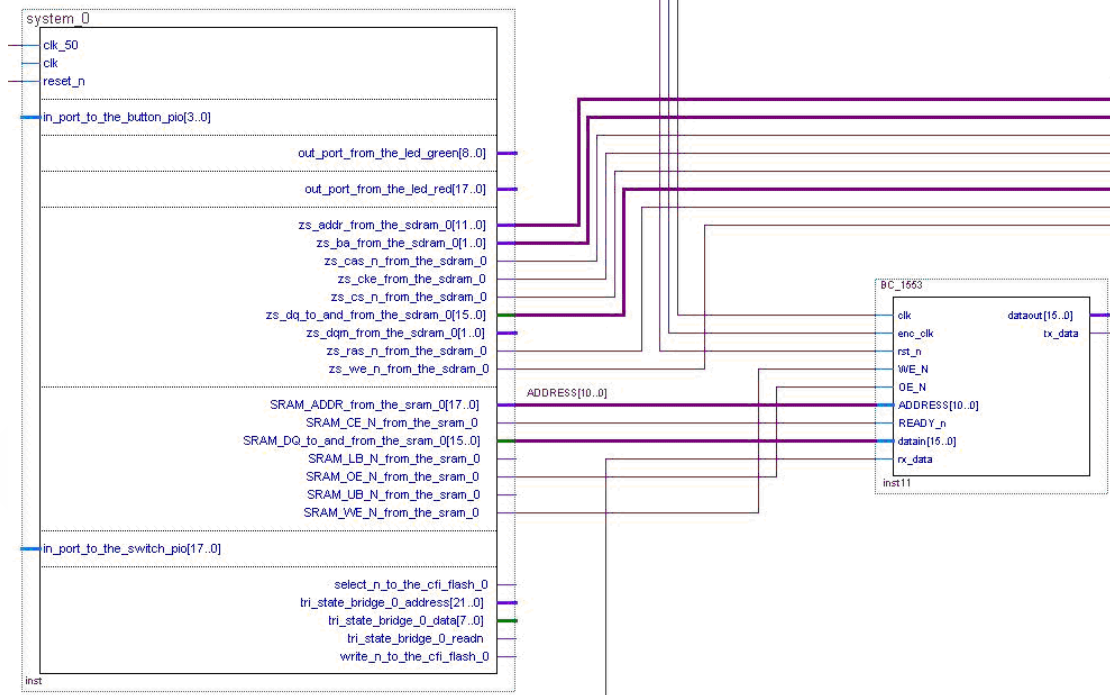


图 12 Nios II 与 1553B 总线控制模块连接图

ModelSim 仿真

仿真功能：使用 Verilog^[7]语言编写测试用例实现 Nios II 的控制逻辑，BC 接收 RT 的 4 个数据，然后从 RAM 中读取接收到的数据。

向 RAM 中地址为 0x0000-0x0003 空间分别写入 0001, 0002, 0003, 0004, 共 4 个需要发送的数据，并且配置相关寄存器：

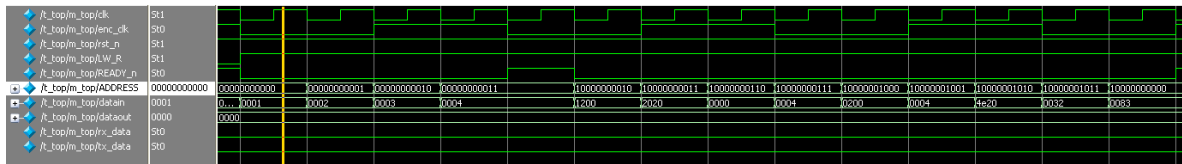


图 13 写入数据，配置寄存器

总线传输阶段，内部数据流控制器完成发送和接收的数据的存放，内部编解码器完成数据到总线上的发送和总线上数据的接收：

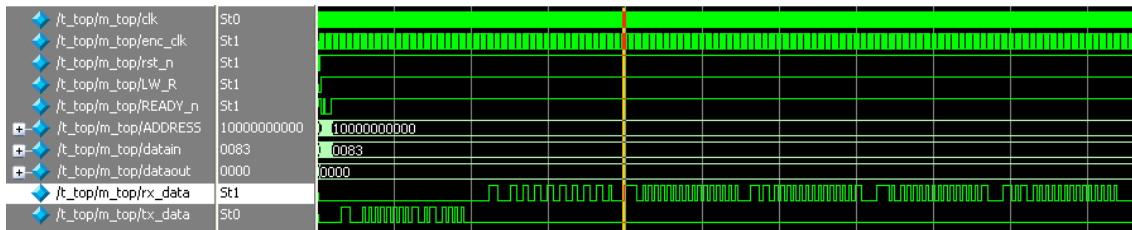


图 14 数据收发

5. 小结

本文使用基于 SOPC 的可重用技术完成了一个 1553B-PCI 桥接通信模块设计。在本设计中，采用 Altera 提供的 IP 核进行设计缩短了开发的周期，Nios II 处理器的使用实现了软件和硬件两个层面的可重用，同时自主设计的 1553B 总线控制模块可以根据不同的应用来进行裁剪，这都使得整个系统配置的灵活性大大提高。

本设计中的三个模块都是以软核的形式在 FPGA 中实现的，因此可以方便的在各种 FPGA 芯片中移植。

参考文献：

- [1] 杨春英，崔风波. 基于 PCI 接口的 1553B 总线模块的设计与实现. 情报指挥控制系统与仿真技术, 2002
- [2] 李桂山，陈金鹏. PCI 局部总线及其应用. 西安电子科技大学出版社, 2003
- [3] DDC Corporation. Mil-Std-1553 Designer's Guide. 6th edition. 2003
- [4] 王建伟. 基于 FPGA 的 1553B 总线接口设计. [电子科技大学硕士论文], 2004
- [5] Altera. PCI Compiler User Guide. Version: 9.0. Document Date: March 2009
- [6] Altera. Nios II Processor Reference Book. 2005
- [7] 夏宇闻. Verilog 数字系统设计教程. 北京航空航天大学出版社, 2003

原创性声明：作者声明，本论文提到的设计方法及内容是本人在指导教师的指导下进行的研

研究工作获得的研究成果。尽我所知，文中除特别标注和指出的地方外，不包含其他人已经发表或撰写过的研究成果及专利性技术。

