基于 FPGA 的高速串行通信嵌入式系统的研究与设计

陈国鹏，陆　达
（厦门大学 计算机科学系　福建 厦门　361005）

【摘　要】：介绍基于 IBM PowerPC 405 硬核的嵌入式系统的构建，分析基于嵌入 IBM Power PC405 硬核的 Virtex-II 系列 FPGA 设计的通信嵌入式系统的系统架构，并实现系统设计，最后下载到 Virtex-II Pro 板上的 Prom 芯片上。测试结果显示，系统结构紧凑，工作可靠稳定，实现了通信嵌入式系统。在 Windows 2000 环境下，传输速率约 4MB/秒。

【关键词】：FPGA; EDK; PowerPC 405; 嵌入式; Aurora IP 核; 高速串行通信

片上可编程系统（SOPC, System On a Programmable Chip）技术是近年来随着微电子技术的发展而出现的新兴的嵌入式计算机系统技术。由于其与传统的嵌入式计算机系统相比速度更快，集成度更高，灵活性更大，因而目前得到了越来越广泛的应用。

飞行控制计算机系统采用多机系统结构，根据工作的要求，为保障系统的可靠性工作，飞行控制计算机采用三余度容错系统，每两台计算机间通过电缆连接，进行高速串行通信。在该课题的背景下，本设计采用了基于 PowerPC405 处理器的 SOPC 解决方案，实现了基于 FPGA 的高速串行通信嵌入式系统平台的研究与设计。

1. 系统整体设计

基于 FPGA 的高速串行通信嵌入式系统平台主要要实现两个主要部分：嵌入式系统平台的搭建和高速串行收发器。嵌入式系统平台的搭建是基于 Xilinx 公司推出的 FPGA 平台 Virtex-II Pro 上的一种较为完善的 SOPC 解决方案。Xilinx 在 Virtex-II Pro FPGA 芯片中嵌入了 IBM PowerPC405 处理器硬核，提供了相应的总线架构，丰富的 IP 核资源，以及方便、高效的设计开发工具。高速串行收发器的实现是使用 Aurora 核，Aurora 核的数据传送为 32 位，串行数据传输速率为 1.25Gbps，参考时钟为 62.5MHz，Aurora 接口读写时钟频率为 31.25MHz；再设计一个接收存储器和一个发送存储器，将 Aurora 核和嵌入式系统连接起来，完整整个系统设计。系统整体结构图如图 1 所示：

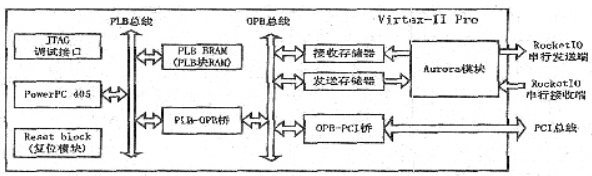

图 1 基于 FPGA 的高速串行通信嵌入式系统平台的系统结构图

根据功能模块，本系统结构可以分为：PowerPC 405 处理器嵌入式系统、Aurora IP 核的通信模块、OPB-PCI 总线桥、数据存储区。下面分别介绍各个部分的设计。

2. 基于 PowerPC 405 处理器的嵌入式系统设计

2.1 PowerPC 405 处理器简介

PowerPC 405 处理器硬核是 IBM 专门为 Xilinx 的 FPGA 芯片开发的处理器产品。嵌入 Virtex-II Pro 芯片中的 PowerPC405 处理器硬核是一种 32 位哈佛结构的 RISC 核，最高可以工作在 400Mhz 频率下。在 Virtex-II Pro 器件中，PowerPC 405 处理器内核与片内所定制的各种外设需要连接起来，而且这些外设的速度可以不一样，甚至会有较大的差别，需要利用各种功能的总线和桥接器来完成。Virtex-II Pro 器件采用 IBM 的 CoreConnect 总线技术。CoreConnect 总线架构是由 IBM 开发的一种片上总线通信连接技术。它能够将 FPGA 内各种不同的 IP 核连接到一起构成一个完整的系统。CoreConnect 总线是一个总线标准的集合，它包括处理器局部总线（Processor Local Bus: PLB）、片上外设总线（On-chip Peripheral Bus: OPB）和设备控制寄存器总线（Device Control Register Bus: DCR）。

2.2 PowerPC 405 硬件系统结构与外部设备概述

在设计中，主要是在 Virtex-II Pro 器件内部构建一个以 PowerPC 405 处理器硬核为中心的嵌入式计算机应用系统，在 FPGA 内部实现系统的总线架构、数据存储、地址译码、外设接口等系统部件和功能。各功能部件在 FPGA 内部都以 IP 核的形式构建并连接。整个系统的结构框图如图 1 所示。其中，JTAG 调试接口连接 PowerPC 405 处理器核和 JTAG 链，用于 JTAG 调试；Reset block 控制 FPGA 内各模块的 reset 信号的输入输出；PowerPC 处理器通过 PLB 总线和 OPB 总线与各外设 IP 核相连，PLB 总线和 OPB 总线之间通过 PLB-OPB 桥相连；OPB-PCI 桥实现了 OPB 总线和 CPCI 总线之间的通信；为了存储应用程序，系统使用了 16K 字节 PLB 总线块 RAM；此外还有二个 8K 字节的 OPB 总线块 BRAM 用于存储通信数据，其中，接收存储器用于存储 Aurora 核接收到的数据，而发送存储器则用于存储待发送的数据；Aurora 模块实现了设备双方通过 Aurora 链路层协议通信，通信双方以帧方式进行数据传输，帧大小设定为 7K，并在通信中采用了 Aurora 核的 NFC 流控制机制。

2.3 系统地址分配

PowerPC 405 处理器提供了 4G 的地址空间用于访问处理器总线上的存储器和 IO 设备。表 1 给出了本设计中设备的地址范围。

设备	起始地址	结束地址
PLB 块 RAM	0XFFFFC000	0XFFFFFFFF
接收存储器	0X81000000	0X81001FFF
发送存储器	0X81100000	0X81101FFF
OPB-PCI 桥	0X80800000	0X8089FFFF

表 1 系统地址分配

由于 Aurora 模块并没有连接在 OPB 总线上，因此，系统并没有给 Aurora 模块分配地址。应用程序的数据和代码都存于 PLB 块 RAM 中，当系统上电或复位时，PowerPC 405 处理器将从地址 0XFFFFFFFC 开始执行 Reset 向量。

3. Aurora IP 核的通信模块

3.1 Aurora 链路层协议简介

Xilinx 公司的 Aurora 链路层协议是一个免费、开放、可扩展、低成本、高带宽的高速串行通信链路层协议，具有简洁的帧结构，适合在点对点串行链路中传输数据。

Aurora 协议描述了用户协议数据单元（用户 PDU, User Protocol Data Unit）是如何在 Aurora 通道中传输的。一个 Aurora 通道由一个或多个 Aurora Lane 组成，每一个 Aurora Lane 都是由一个 MGT（multi-gigabit transceiver）与另一个 MGT 连接成的全双工串行收发器。通过 Aurora 通道进行数据传输的单元称为 Aurora 通道协议数据单元（通道 PDU）。Aurora 接口通过用户接口从用户程序接收数据和控制信号，并向用户程序发送数据和控制信号。

基金项目：国家高技术研究发展计划（863 计划）项目（2003AA755024）

Aurora 协议具有以下特性：支持 622Mbps 至 10Gbps 的传输速率；任意个数的通道绑定可用来支持更高的带宽；支持全双工或单工通信；内部流控制；支持 8B/10B 或 64B/66B 编码。Aurora 协议适用于任何需要进行点对点串行传输的场合，包括：芯片到芯片之间的数据传输；板级传输；流数据接口。

3.2 Aurora IP 核功能描述

Aurora IP（intellectual property）核是 Xilinx 公司提供的基于 Aurora 协议和 Rocket I/O 多吉比特收发器的高速串行设计方案。该 IP 核内嵌了 Rocket I/O 模块，在内部实现了 Aurora 协议，并提供了简单灵活的用户接口。使用该 IP 核，用户可以将 Rocket I/O 复杂的控制结构转换为简单的用户接口，通过 Rocket I/O 多吉比特收发器收发数据，无需自己设计有关串行接口所涉及的 8b/10b 编解码、同步、收发速率匹配等问题。

Aurora IP 核的功能结构图如图 2 所示：

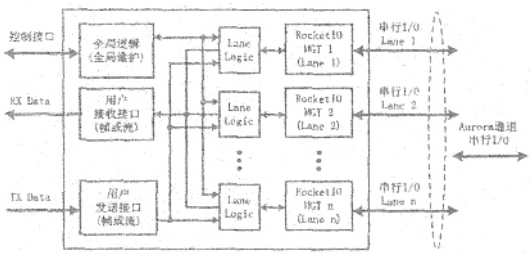

图 2 Aurora IP 核的功能结构图

(1) Lane Logic：每一个该模块的实例驱动一个 MGT，负责其初始化、处理控制字符的编码解以及错误检测的工作。

(2) 全局逻辑：全局逻辑模块负责通道的初始化、绑定以及验证工作。

(3) 用户接收接口：用户接收接口负责从通道中取出数据送至用户程序。

(4) 用户发送接口：用户发送接口负责把用户程序的数据送给通道。

3.3 Aurora 通信模块的实现

Aurora 模块主要实现将发送存储器中的数据帧通过 Aurora 核发送出去，并将 Aurora 核接收到的数据帧存入接收存储器。Aurora 模块由发送存储器控制器、接收存储器控制器、发送状态机、接收状态机和 Aurora 核组成。图 3 分别显示了发送状态机和接收状态机。

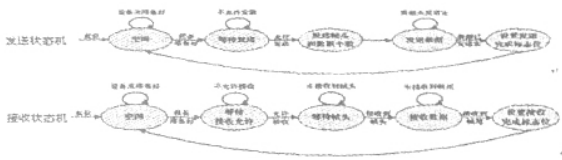

图 3 Aurora 发送状态机和 Aurora 接收状态机

4. OPB- PCI 桥

在本设计中，主机通过 PCI 总线访问 OPB 总线上的接收存储器和发送存储器，OPB- PCI 桥成为 PCI 总线上的一个目标设备和 OPB 总线上的主设备，工作方式为从桥。OPB- PCI 桥需要把 PCI 总线的地址转化为 OPB 总线上的地址，这是通过设置 OPB- PCI 桥的 C_PCIBAR2IPIFBAR 和 C_PCIBAR_LEN 二个参数来实现的，C_PCIBAR2IPIFBAR 表示 PCI 总线要访问的 OPB 总线上的设备起始地址，而 C_PCIBAR_LEN 则表示 PCI 端访问的地址空间大小。由于在 OPB 总线上有接收和发送二个存储器，因此，还需要把 OPB- PCI 桥上的 PCI 存储器个数设置为 2，这是通过将参数 C_PCIBAR_NUM 设置为 2 实现的。参数设置如表 2 所示。OPB- PCI 桥的其它参数采用系统默认即可。

参数名称	参数值
C_INCLUDE_PCI_CONFIG	0：工作方式为从桥
C_PCIBAR_NUM	2：二个存储器空间
C_PCIBAR2IPIFBAR_0	0X81000000：接收存储器的起始地址
C_PCIBAR_LEN_0	20：PCI 端访问的空间为 1M
C_PCIBAR2IPIFBAR_1	0X81100000：发送存储器的起始地址
C_PCIBAR_LEN_1	20：PCI 端访问的空间为 1M

表 2 OPB-PCI 桥主要参数设置

5. 接收存储器、发送存储器

本设计中，接收存储器、发送存储器采用 8K 双端口片上块 RAM 连接到 OPB 总线上。块 RAM 的用户侧信号定义如下：

- BRAM_Rst: reset 信号，高有效
- BRAM_Clk: 时钟信号
- BRAM_EN: BRAM 使能信号，高有效
- BRAM_WEN: 写使能信号，高有效
- BRAM_Addr: 地址信号，32 位地址，低 13 位有效
- BRAM_Din: 输入数据总线，32 位
- BRAM_Dou: 输出数据总线，32 位

在 Aurora 模块中通过控制这些信号线实现对块 RAM 的数据存取。接收存储器和发送存储器的 8K 空间分配如表 3 和表 4 所示：

地址偏移	作 用
0X00	当值为 0X00 时，表示接收存储器中接收到了一帧数据
0X0C	接收到的数据帧大小
0X10	数据存储的起始地址，从该地址到存储器的结束地址用来存储数据。

表 3 接收存储器空间分配

地址偏移	作 用
0X04	当值为 0X01 时，表示发送存储器中有一帧数据待发送
0X0C	发送的数据帧大小
0X10	数据存储的起始地址，从该地址到存储器的结束地址用来存储数据。

表 4 发送存储器空间分配

6. Xilinx SOPC 集成开发环境 EDK

6.1 EDK 概述

Xilinx 公司提供的 EDK(Embedded Development Kit)设计工具，是一个专门用于 FPGA 内部 32 位嵌入式处理器的集成化开发工具包，并提供硬件和软件的协同设计能力，从而能极大地缩短设计周期。

在 EDK 工具包中集成了硬件平台产生器(Platgen)、硬件仿真模型产生器(SimGen)、软件平台产生器(Libgen)、应用软件编译工具(GNU Compiler)软件调式工具(GNU Debugger)等等。用户可以通过集成在 EDK 中的平台工作室（Xilinx Platform Studio, XPS)在 Windows 的图形界面下，方便地调用各种工具，完成整个 SOPC 系统的开发，并且软硬件的开发可以同时进行。

此外，EDK 也提供了各种丰富的 IP 核，包括总线接口、外部存储控制器、通用 IO、中断控制等等。利用这些现有的资源，用户可以非常方便地构造自己的嵌入式平台。如果已有的 IP 核不能满足用户的特定需求，用户也可以开发自己的 IP 核。

6.2 EDK 系统描述文件

在 EDK 集成开发环境中，通过 MHS(Microprocessor Hardware Specification) 文件、MSS(Microprocessor Software Specification) 文件和 MVS （Microprocessor Verification Specification) 文件完整地描述了 SOPC 系统的软硬件结构和仿真验证模型。MHS 文件：用于完整描述 Virtex- II Pro FPGA 的硬件系统结构，主要定义当前 SOPC 设计的处理器类型、总线结构、外设接口、中断处理和地址空间。MSS 文件：用于完整描述嵌入式处理器的平台 FPGA 的软件系统结构。它主要定义当前 SOPC 系统设计的软件库、驱动程序和文件系统。MVS 文件：用于说明当前嵌入式处理器的平台 FPGA 的硬件仿真模型。

6.3 EDK 开发流程

在 EDK 的集成开发环境 XPS 中，FPGA 设计者可以采用工程形式管理基于 PowerPC405 处理器硬核的 SOPC 软硬件系统，并调用多种设计开发工具。

在 XPS 环境下，设计者通过创建或导入 MHS 文件来建立包括 PowerPC405 处理器硬核在内的整个 SOPC 的硬件系统，并添加用户自己开发的 IP 核，最后，产生当前 FPGA 硬件系统的网表文件和约束文件。在硬件开发完成后，可以根据需要使用硬件仿真模型产生器(SimGen)产生仿真模型。使用仿真软件，如

ModelSim 进行硬件仿真。在无误后，可以生成.bit 流文件，将位流文件转换成 .mcs 的配置文件，下载到 FPGA 的配置 PROM 中完成整个设计开发过程。

7. 功能仿真、板级验证及性能分析

7.1 Aurora 数据通信模块的功能仿真

功能仿真主要是将写好的源代码做编译，以检查语法上是否正确，再依照设计者输入的信号产生所要的输出，来验证功能上是否正确，本仿真是在 Modelsim 下进行

图 4 上部分显示了 Aurora 模块发送数据的过程。当信号 tx_sof_i=0 有效时，表示 Aurora 模块开始发送一帧数据，这时 Aurora 模块将发送 tx_d_i 上的数据；当 tx_eof_i=0 有效时，Aurora 模块将发送帧尾信息以结束数据的发送。在这个过程中，Aurora 发送了一帧数据。

图 4 下部分显示了 Aurora 模块接收数据的过程。当 rx_sof_out=0 有效时表示 Aurora 模块接收到了帧头信息并开始接收数据，rx_d_i 为接收到的数据；当 rx_eof_out=0 有效时，表示 Aurora 模块接收到了帧尾信息，应停止接收数据。在这个过程中，Aurora 接收了一帧数据。

对照图 4 输出结果可以看出，Aurora 模块在功能上是正确的。

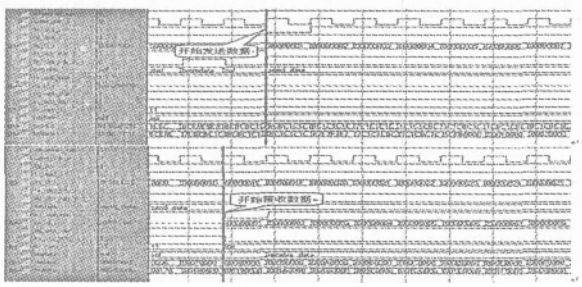

图 4 Aurora 通信模块的仿真

7.2 板级验证

在测试时，两个开发板分别插在两台主机的 PCI 插槽上，两个开发板上的两个光模块通过光纤互连。主机上的操作系统为 Windows 2000。二台主机通过通信卡进行通信的过程，分为发送数据和接收数据二个过程。在数据传输过程中，发送存储器和接收存储器充当了数据缓冲区的功能。

发送数据时：主机首先通过 PCI 总线读取发送存储器偏移地址 0X04 中的内容。若其值为 1，表示设备忙，主机必须等待设备空闲再发送数据；否则若为 0，表示设备空闲可以发送数据，此时，主机进行如下的操作：主机将要发送的数据长度 X 写入发送存储器偏移地址 0X0C，接着把要发送的一帧数据写入发送存储器从偏移地址 0X10 开始的 X 个位置，待数据全部写入后，通过把数值 1 写到发送存储器偏移地址 0X04 通知 Aurora 模块发送数据。之后，Aurora 模块从发送存储器偏移地址 0X04 读到值 1，得知有一帧数据等待发送，便进行如下操作：首先，读取发送存储器偏移地址 0X0C 获得要发送的数据长度 X，然后按照取得的长度，从偏移地址 0X10 处读取 X 个数据，与数据长度合在一起组成一帧数据通过 Aurora 核发送出去。待全部数据发送完成后，Aurora 模块向发送存储器偏移地址 0X04 写入数值 0 通知主机发送已完成，可以继续发送下一帧数据。

接受数据时：Aurora 模块通过读取接收存储器偏移地址 0X00 的内容可以知道是否允许接收数据，当其值为 1 时，表示允许接收数据；值为 0 则不允许接收数据，此时可能是主机不允许接收数据或接收存储器内有数据，应忽略所有接收到的数据。因此，当 Aurora 模块接收到对方发送来的帧头信息时，首先读取接收存储器偏移地址 0X00 中的内容，若其值为 1，则进行如下的操作：将接收到的第一个数据 Y (Y 为发送的数据长度)写入接收存储器偏移地址 0X0C，接着依次将接收到的数据写入接收存储器偏移地址 0X10 起始的 Y 个位置，直到接收到帧尾信息才停止接收数据。接收完一帧数据后，Aurora 模块将值 0 写入接收存储器偏移地址 0X00 通知主机已接收到一帧数据。之后，主机通过 PCI 总线查询到接收存储器偏移地址 0X00 内容为 0 得知已接收到一帧数据，便可将数据取走。

逻辑在线分析仪 Chipscope 能够实时读取程序在 FPGA 中运行时的信号值，图 5 给出的是程序运行时的部分时序过程，上部分为主机发送数据的部分时序图，下部分为接收数据的部分时序图。可以看出发送端发送的数据被接收端正确接收。

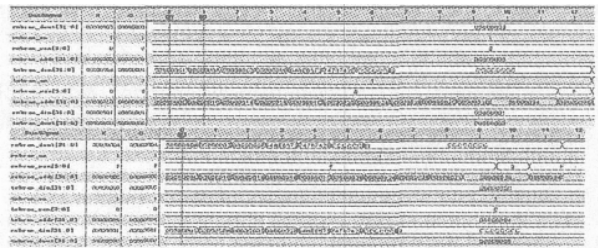

图 5 程序运行时 chipscope 抓取出的部分发送和接收图

7.3 性能分析

测试在 Windows 2000 操作系统下进行，传输时间是从一台主机发送文件到另一台主机接收结束。表 5 为 4 个测试文件的发送和接收情况：

文件大小(M 字节)	A->B 传输时间(秒)	B->A 传输时间(秒)	传输速率(M 字节/秒)
15.0	3.64	4.16	4.12 / 3.60
29.8	7.30	7.50	4.08 / 3.97
156	38.70	39.41	4.03 / 3.96
400	97.66	108.9	4.10 / 3.67

表 5 文件发送和接收

从表 5 上述实验数据可以看出，传输速率大致为 4MB/s。

设计中通信卡采用光纤进行通信，光纤的传输速率为 1.25Gbps，由于 RocketIO 采用 8B/10B 进行编码，因而实际上传输数据的最高速率应为 125MB/s。造成实际数据传输速率和理想传输速率不一致的主要因素是 PCI 总线的工作方式，在系统设计中，PCI 总线的工作方式采用轮询方式，没有使用中断，并且不支持 DMA 工作方式，这大大降低了数据的传输速率；其次，PCI 总线是共享式总线，由多个设备共用总线；另外，使用的操作系统不是实时性操作系统，对 PCI 总线的数据交易实时性支持较差。

结束语

高速串行通信以其自身的优点，在通信技术领域的应用越来越广泛。为了确保较高的安全性能，一些对控制系统要求很高的领域如飞机控制系统，一般采用多处理机容错系统，每台处理机又要求能够高速通信。在这个背景下，本文设计了基于 FPGA 的高速串行通信嵌入式系统。此系统工作可靠，结构紧致，可以扩展到多个节点间的环形通信系统。

参考文献：

1. 王诚，薛小刚，钟信潮. FPGA/CPLD 设计工具：Xilinx ISE 使用详解[M].北京：人民邮电出版社，2005.
2. 徐欣，于红旗，易凡，卢启中.基于 FPGA 的嵌入式系统设计<Xilinx Edition>[M].北京：机械工业出版社，2005.
3. 潘松，王国栋. VHDL 实用教程[M].成都：电子科技大学出版社，2001.
4. Xilinx Corp. EDK 7.1 PowerPC Tutorial in Virtex- 4. Xilinx, 2005.
5. Xilinx Corp. LogiCORE? Aurora v2.3. Xilinx. 2005.
6. Aurora Protocol Specification. Xilinx. 2003.
7. Aurora IP Overview: Aurora Reference Design. Xilinx. 2004.
8. LogiCORETM PCI Design Guide Version 3.0. Xilinx. 2003.
9. Virtex- II Pro? Platform FPGA User Guide. Xilinx. 2004.
10. 吴德铭，陆达. 高速通信中基于 FPGA 的 PCI 总线接口研究与设计[J]. 计算机应用，2005.
11. 孙永明，林琦. 1.5Gbps 高速串行数据恢复电路的标准单元实现计[J]. 计算机研究与发展，2005.
12. 杨刚，周宗仪. 基于 Rocket I/O 模块的高速 I/O 设计[J]. 电子应用技术，2004.