

文章编号:1001-893X(2011)03-0075-04

基于 PowerPC 的 RapidIO 高速串行通信设计与实现^{*}

纪 斌,郑志国

(中国西南电子技术研究所,成都 610036)

摘 要:提出了采用基于 PowerPC 架构的 MPC8548E 为主机的高速串行 RapidIO 实现方案,详细阐述了硬件设计要点和软件初始化流程,为以 PowerPC 为处理器的嵌入式操作系统实现设备间的高速互连提供了一套行之有效的解决方案。

关键词:嵌入式操作系统;系统互联;数据传输;串行 RapidIO

中图分类号:TN919;TP336 **文献标识码:**A **doi:**10.3969/j.issn.1001-893x.2011.03.017

Design and Implementation of High-speed Serial RapidIO Based on PowerPC

Ji Bin, ZHENG Zhi-guo

(Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: A scheme is proposed to realize high-speed serial RapidIO based on PowerPC architecture CPU MPC8548. The hardware design and the flow of the software initialization are emphasized. This implementation can solve the problem of high-speed connection between equipment in embedded system based on PowerPC architecture.

Key words: embedded system; system interconnect; data transmission; serial RapidIO

在传统的嵌入式系统中,受处理器速度和性能的限制,系统对数据传输的速率要求不高。然而,随着 CPU、DSP、FPGA 和 ADC 技术的迅速发展,传统的分层共享总线已经无法满足现在高性能嵌入式系统的 IO 性能需求和信号处理高速数据传输需求。系统互联,即板间各种信号高速实时交互,成了制约嵌入式系统性能提升的一个主要瓶颈。

RapidIO 作为解决各系统板间数据高速互连的国际通用协议,得到了越来越多主流芯片厂商的支持,有广阔的应用发展空间。串行 RapidIO 的互连架构是一种高性能、低引脚数、基于包交换的交叉开关互连技术。本文提出了基于 PowerPC 的 RapidIO 高速通信协议来解决这一瓶颈的软硬件实现方式。

1 基于 PowerPC 的串行 RapidIO 的硬件设计

PowerPC 作为高端的嵌入式系统处理器,在航

空、航天、雷达终端等电子设备领域有着广泛的应用。RapidIO 协议有多种实现方式:可以通过 FPGA 加入 IPCORE 的方式,但这种方式受限于 IPCORE 对器件的限制和对协议实现的不完全,很难运用于嵌入式系统实时的主控操作;还可以通过一些 DSP 提供的 RapidIO 硬件接口实现,通过操作 DSP 提供的 RapidIO 专用寄存器能轻松进行 RapidIO 设备间的读写,但是 DSP 是单线程操作,很难加入实时操作系统对任务进行管理和调度。因此,本次设计最终选择了通过 PowerPC 提供的 RapidIO 硬核实现。本次方案设计选用 Freescale 公司针对 Serial RapidIO 推出的最新 PowerPC——MPC8548E。

由于串行 RapidIO 采用的是点对点技术,在需要 2 个以上 RapidIO 接口处理或桥接部分的系统中都需要采用一台交换机。因此,本次设计选用了 Tindar 公司针对串行 RapidIO 推出的专用交换机 Tsi745。

^{*} 收稿日期:2010-11-17;修回日期:2011-01-26

Tsi745 交换机可实现与串行 RapidIO 兼容的微处理器、DSP、FPGA 及其它外围设备互联,支持40 Gbit/s 的整合带宽。

作为终端显控的嵌入式系统,本次设计还加入

了 2 个千兆以太网口以及 4 个串口控制器,作为与外部板卡连接的必要工具,RapidIO 信号通过 VPX 插座传输。整个硬件框图如图 1 所示。

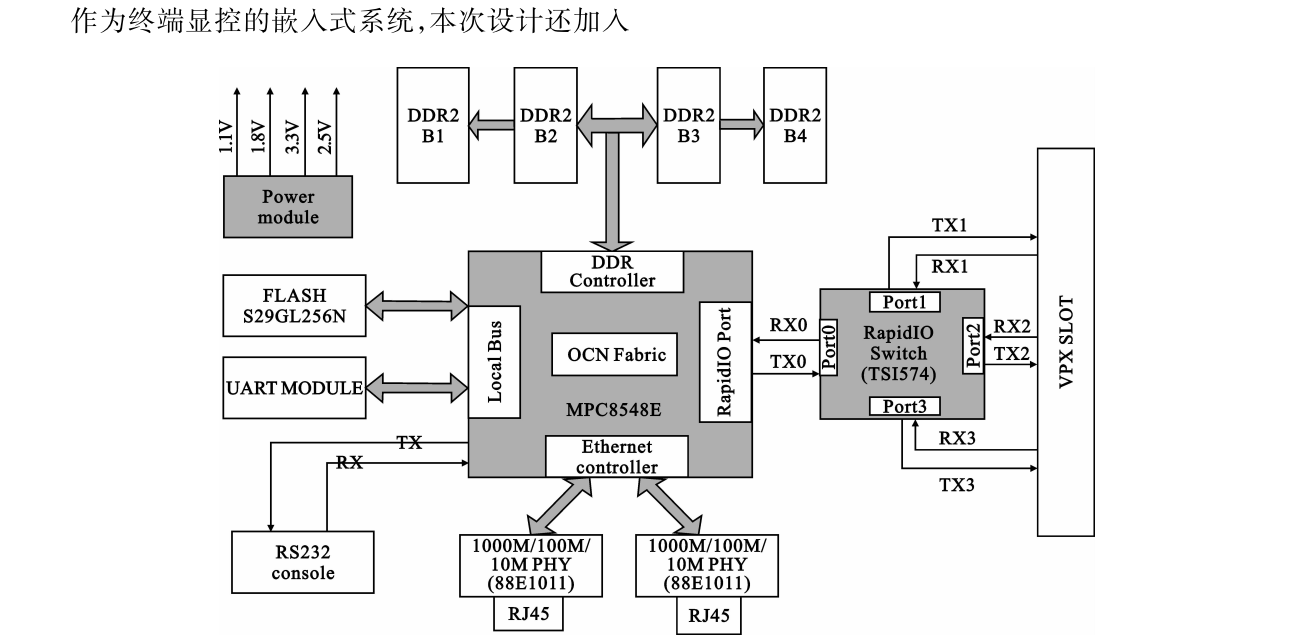


图 1 MPC8548E 硬件连接框图

Fig.1 Block diagram of MPC8548E hardware

在硬件设计方面,首先要注意电源。在上电时,采用单 5 V 供电,MPC8548E 的瞬时电流最高能到 8 A,稳定时为 2.5 A 左右,因此 MPC8548E 对于电源负载能力要求很高。MPC8548E 核心电压为 1.1 V,为了能在 1.5 GHz 的频率下正常运行,需要一个独立无干扰的电源层,本文采用 TI 公司的直流开关电源转换芯片 TPS54010,最高能提供 10 A 稳定电流。DDR2 芯片和 MPC8548E 自带的控制器的参考电压为 1.8 V,两者也需要独立回路的电源层。其它大多数器件都只需要 3.3 V 的单电源供电,其回路需要的电流也很大,采用的也是 TI 公司的开关电源转换芯片 TPS 54810 PWP,也能提供最大 10 A 的稳定电流。由于整个系统对电源的稳定性要求极高,本文采用了 MAXSIM 公司的 MAX16006 电源监控芯片,对各个电源实时监控,如果超出输入参考值的 1%,它会自动产生全局复位信号,从而保护整个电路。其次是时钟信号,本次设计需要多个不同时钟源满足各个器件的工作需求,MPC8548E 需要一个范围为 20 ~ 80 MHz 的系统时钟,作为内核 PLL 以及外围 DDR 控制器等的时钟参考。Tsi745 交换机需要两个独立的时钟,其中一路 156.25 MHz 差分时钟输入用来控制 RapidIO 信号的传输速率,一路 100 MHz 共模时钟信号用以控制 Tsi745 内部寄存器的访问,千兆以太网也需要一个 25 MHz 的时钟信号,PCB 布线时,要注意它们不要相互干扰,以致传输数据错误。然后在印制板绘制时,需要注意一些差分等长线设计。由于 DDR2 运行的频率在 533 MHz 以上,其信号线、地址线和差分时钟线都要按照自身手册要求严格等长,RapidIO 接收端需要端接一个 0.1 μ F 的电容,在高速传输中隔离直流信号。4x 的 RapidIO 信号接收和发送端也必须分别等长,并满足 50 Ω 的阻抗匹配。

2 RapidIO 板间通信的软件实现

在进行 RapidIO 读写前,先要在上电时对 MPC8548 和 Tsi574 进行必要的硬件配置。本文设置为 RapidIO Host 模式,通信速率为 1.25 Gbit/s。Tsi574 端需要通过 SP {n} _ MODESEL、SP _ IO _ SPEED[1:0]、SP {n} _ PWRDN 引脚配置各个端口的模式、速率以及是否使能,本文设置为 1x 模式,速率是 1.25 Gbit/s,每个端口都使能。硬件配置完成后,需要对 RapidIO 内部寄存器进行初始化配置,并完成对内部 Rapid IO 和外部 Agent 地址空间映射,最终就能像访问外部存储器一样直接对这些地址空间

操作,实现各个 RapidIO 器件的读写。由此可见,软件上的关键就是对 RapidIO 寄存器的初始化配置和地址映射,简单的流程如图 2 所示。

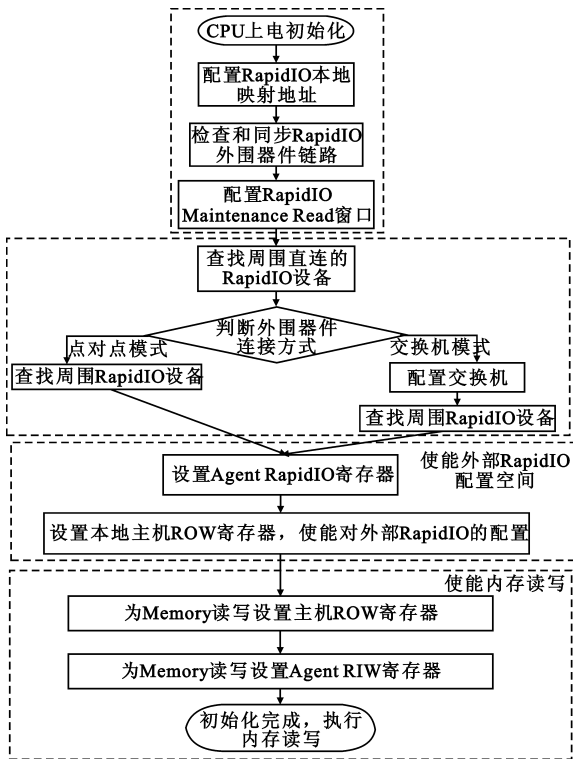


图 2 RapidIO 初始化软件流程
Fig.2 RapidIO software initialization flow

2.1 配置本地处理器

在启动 RapidIO 操作之前,首先要配置一些 MPC8548E 内部寄存器。上电后,第一个要配置的就是 TLB(Translation Lookaside Buffers)入口寄存器,为 RapidIO 读写操作设置内存空间。TLB 入口设置通过配置 MAS0(MMU Assist Register0)、MAS1、MAS2、MAS7 来实现。由于 MAS 寄存器是内部特殊寄存器,只能通过汇编指令来访问。

然后设置内部区域寄存器(Local Area Window)来重新映射可以访问的 RapidIO 读写内存空间,需要设置两个寄存器:LAWBAR(Local Access Window Base Address Register)设置 RapidIO 内存空间的基地址,如 0xC0000000; LAWAR (Local Access Window Attribute Register)设置本窗口的类型和大小,必须设置成 RapidIO 类型,大小根据实际需要确定,如 0x80C0001B,C 表示 RapidIO,1B 表示 256M。

第三步检查和同步 RapidIO 外围链路。在任何 RapidIO 操作前,都要查看有没有正确链接其它 RapidIO 器件。通过读寄存器 ESCSR(Error and Status

Command and Status Register)来获得链路的各种信息,ESCSR[PO]表示输入输出已经初始化完成,可以进行下一步操作,如果有错,就检查 ESCSR 其它位,找出错误地方,以便进行下一步修改。

第四步就是设置 Maintenance 窗口,为配置外部交换机和其它 RapidIO 从设备作准备。需要配置 RapidIO Outbound 窗口的 3 个寄存器为:ROWBAR (RapidIO Outbound Window Base Address Register)设置 Outbound 窗口的 32 位基地址,如 0x000C0000; ROWAR(RapidIO Outbound Window Attributes Register)设置 Outbound 窗口使能、读写和大小等属性,设置为 Maintenance read 和 write,如 0x8007070013; ROW-TAR (RapidIO Outbound Window Translation Address Register)设置 Outbound 窗口外部器件 ID、跳数和 Maintenance 操作高位地址,如 0x3FC00000。

2.2 查找系统内其它 RapidIO 设备

执行的第一条 RapidIO 操作指令就是通过 Maintenance read 获取外部第一个 RapidIO 设备的设备 ID,对外部 RapidIO 从设备的寄存器空间读写都需要通过 Maintenance read 或 write 指令操作。本次设计使用的是交换机 Tsi574,读出的 ID 为 0x05740002。然后对交换机进行配置,需要读出端口个数(RIO_SW_PORT Register),锁定 ID(HB-DIDLCR Register),然后更新路由查找表,通过设置 RIO_ROUTE_CFG_DESTID 和 RIO_ROUTE_CFG_PORT 确定发给主机的数据都在交换机内,且都通过正确的端口发送。

交换机配置好后,需要通过交换机查找下一级的 RapidIO 从设备,检查到正确的链路后,也需要读出其设备 ID,从而判断是哪一类器件。一般从器件的 Device ID 都是 0xFF,需要把它修改为不同的 ID,可以依次设置为 1、2、3、4。最后还要更新路由查找表,使每个端口能正确收发数据。

2.3 使能外部 RapidIO 配置空间

为了正确读写外部 RapidIO 器件的配置空间,首先要配置各个从器件的 LCSBA1CSR(Local Configuration Space Base Address 1 Command and Status Register),设置 RapidIO 在这个从器件内的起始地址。然后配置主机 Outbound 窗口 3 个寄存器:ROWBAR、ROWAR、ROWTAR,这个在设置 Maintenance 窗口步骤中已经配置过,但是要进行一些修改,ROWBAR 起始空间应从 0x000C1100 开始,ROWAR 窗口属性

要设置为 NREAD 和 NWRITE, ROWBAR 需要设置 Target ID 和 Outbound 传输地址,如 0x00401000。使能外部 Rapid 配置空间,还能在上电时通过主机启动 RapidIO 从设备。

2.4 使能内存读写

为了使能内存读写,在主机端配置好 ROWBAR、ROWAR、ROWTAR 寄存器后,还需要对各个从设备的 Inbound 窗口寄存器进行设置:RIWBAR(RapidIO Inbound Window Base Address Register),设置好从设备端的 RapidIO Inbound Window 基地址;RIWAR(RapidIO Inbound Window Attributes Register)设置 Inbound 窗口属性,包括窗口大小、读写模式和内存映射等;RIWTAR(RapidIO Inbound Window Translation Address Register),设置从设备数据输出起始地址。正常配置好这些寄存器后,就能像访问内存空间一样对 RapidIO 设备进行读写了。

3 实验结果

本次设计用于某项目中信号处理板与显控板实时数据传输通信中,要求每隔十几微秒就传输 3~4 kbyte 数据。在 Linux 操作系统中,采用 RapidIO 1x 模式,在 1.25 Gbit/s 速率下进行数据传递检验,实验框图如图 3 所示。

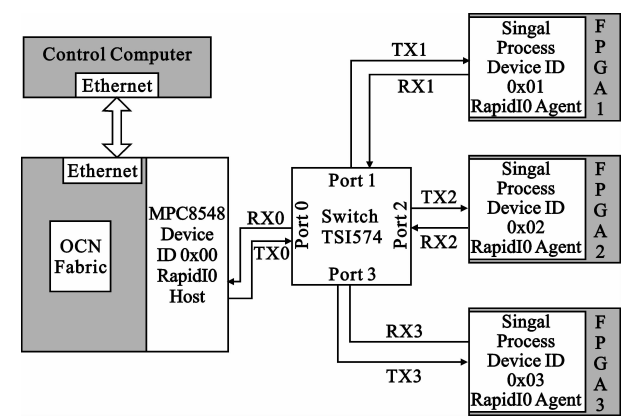


图 3 RapidIO 实验连接图
Fig.3 RapidIO experimental connection block diagram

首先在上电后,PowerPC 先配置好交换机 Tsi574,通过交换机找到 3 个 RapidIO 从设备,然后对这 3 个设备进行初始化,包括修改 Device ID、RapidIO Inbound、Outbound 寄存器和配置读写的映射空间,根据通过串口打印在控制台上的信息,一步步判

断初始化程度。RapidIO 正确配置后,待 Linux 系统正常启动后,通过上位机电脑网络发送数据到 PowerPC,然后 PowerPC 把收到的数据直接通过内存读写分发给 3 个 RapidIO 从设备,从设备收到数据后,发送接收完成中断,PowerPC 再分别读出 3 个从设备接收到的数据进行比较。实验结果表明,在 1.25 Gbit/s 速率下使用 Serial RapidIO 协议通信,4 kbyte 数据能在 1 μ s 内正确实时地传送。

参考文献:

[1] Freescale Semiconductor. MPC8548E PowerQUICC™ III Integrated Processor Family Reference Manual [M]//MPC8548ERM Datasheet. East Kilbride, USA: Freescale Semiconductor Inc,2007: 1035 – 1222.

[2] Lorraine McLuckie, Colin Cureton. Serial RapidIO Bring – Up Procedure on PowerQUICC™ III[M]. East Kilbride, USA: Freescale Semiconductor Inc,2005:3 – 38.

[3] 韦东山.嵌入式 Linux 应用开发完全手册[M].北京:人民邮电出版社,2009:384 – 389.

WEI Dong – shan. Application of embedded Linux system [M]. Beijing: People's Posts and Telecommunications Press, 2009:384 – 389. (in Chinese)

[4] 朱坚,徐光辉,朱利利.基于 Serial RapidIO 的高速实时数据采集处理系统[J].电子质量,2008(10):6 – 7.

ZHU Jian, XU Guang – hui, ZHU Li – li. High Speed Data Acquisition and Processing System Based on Serial RapidIO [J]. Electronics Quality, 2008(10):6 – 7. (in Chinese)

[5] 邓豹,赵小冬.基于串行 RapidIO 的嵌入式互连研究[J].航空计算技术,2008,38(3):123 – 126.

DENG Bao, ZHAO Xiao – dong. Research of the Embedded Interconnection Frame Based on Serial RapidIO Technology [J]. Aeronautical Computing Technique, 2008,38(3):123 – 126. (in Chinese)

[6] 王学宝,郑波祥,朱勇.基于 TSI568 的 RapidIO 交换模块设计[J].微计算机信息,2009, 25(6):104 – 106.

WANG Xue – bao, ZHENG Bo – xiang, ZHU Yong. Design of RapidIO Switch Board Base on TSI568[J]. Micro Computer Information, 2009, 25(6):104 – 106. (in Chinese)

作者简介:

纪 斌(1981 –),男,四川广汉人,工程师,主要从事雷达显控方面的工作;
JI Bin was born in Guanghan, Sichuan Province, in 1981. He is now an engineer. His research concerns radar display and control.
Email: ziyoudianzi1981@yahoo.com.cn
郑志国(1978 –),男,四川成都人,工程师,主要从事嵌入式系统研发工作。
ZHENG Zhi – guo was born in Chengdu, Sichuan Province, in 1978. He is now an engineer. His research concerns embedded system.