

A Multifunctional Serial-Port Server Design Based on Zynq

Haiwei Jiang, Lantu Guo, Xiaoming Gou, Yuebao Zhang

China Research Institute of Radiowave Propagation, Qingdao Shandong
Email: jhw200451@sina.com

Received: May 6th, 2019; accepted: May 21st, 2019; published: May 28th, 2019

Abstract

A multifunctional serial-port server, which bridges the gap between the computer and the platform under test, is designed for the development and tests of hardware platforms with multiple RS422 serial ports. The server enables the transformation from multiple serial ports to the network interface, which is driven by the Xilinx Zynq core, together with some peripheral devices. It can be automatically reconfigured. Besides, it can be set to work at non-standard baud rate.

Keywords

Serial-Port Server, Zynq, Auto-Reconfiguration, Non-Standard Baud Rate

一种基于Zynq的多功能串口服务器设计

姜海卫, 郭兰图, 苟晓鸣, 张跃宝

中国电波传播研究所, 山东 青岛
Email: jhw200451@sina.com

收稿日期: 2019年5月6日; 录用日期: 2019年5月21日; 发布日期: 2019年5月28日

摘要

为满足具有多个RS422串口的硬件产品平台研发和调试, 本文研制一种多RS422串口服务器, 连接计算机与被调试产品。该服务器以Xilinx的Zynq为核心, 配合相应的外围接口芯片, 实现了网口到多串口的扩展, 串口特性的动态可重配, 以及实现了串口的非标准波特率。

关键词

串口服务器, Zynq, 动态可重配, 非标准波特率

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

RS422 串行总线采用差分传输的方式, 其传输速率最大可达到 10 Mbps。因其总线的可靠性、通信协议的简单性以及对硬件要求不高, 所以在当前各类工业系统平台上具有广泛应用。

当前, 现有的大多数串口服务器采用 USB 转串口的模式, 该模式存在两点不足: 无法实现较长距离的数据传输和无法完成非标准波特率的串口数据传输[1] [2] [3]。

针对此状况, 本文研制一套基于 Xilinx 公司的 Zynq 芯片为核心的多功能串口服务器。该服务器由以太网口转换出多个 RS422 串口(以下简称 422), 既通过网线实现了长距离数据传输, 又充分利用 Zynq 的灵活架构, 完成了标准和非标准波特率兼顾的 422 串口数据传输。

2. 组成框图及原理

2.1. 组成框图

多功能串口服务器组成框图如图 1 所示。

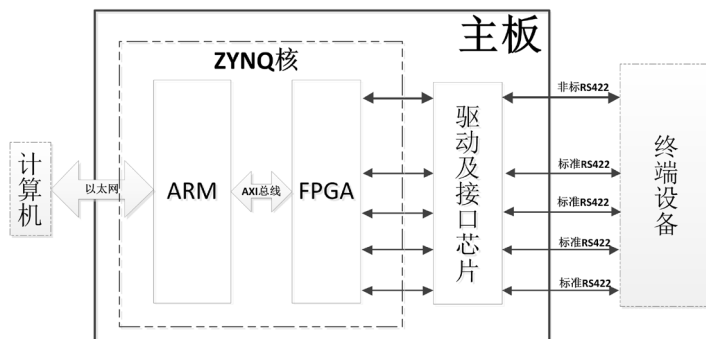


Figure 1. Multifunctional serial-port server structure

图 1. 多功能串口服务器组成框图

2.2. 工作原理

多功能串口服务器以 Zynq 为主控核心, 主要功能是通过网口接收计算机的指令, 并将指令解码分发各 422 端口, 同时将各 422 端口的返回数据打包, 再通过网口反馈给计算机。

当前, Zynq 作为 Xilinx 公司新一代的 FPGA, 内部除了常规的 FPGA 的资源外, 还集成了 ARM 硬核, 这使得其既具有通信和控制上的灵活性, 同时也具有逻辑上的实时性。基于该平台的串口服务器, 充分利用了 Zynq 内部的这两种资源, 并通过 AXI (Advanced Extensible Interface)总线完成两种资源之间的数据交互, 实现了波特率的动态重配, 同时也实现了特殊波特率通信。

首先,计算机通过以太网与 Zynq 中的 ARM 进行通信,然后 ARM 通过内部 AXI 总线与 FPGA 进行数据交互,最终再由 FPGA 来完成对外的 422 数据输入输出。

数据下传时,ARM 通过以太网获得计算机端数据,经解码和校验正确后,会根据数据包中格式字的不同被分为三种类型:指令包、标准波特率 422 数据包和非标准波特率 422 数据包。其中,指令包用来对串口的通信波特率、校验等进行配置;标准波特率 422 数据包则在 ARM 中进行打包,然后通过 AXI 总线传输给 FPGA,并调用 FPGA 中的 IP 模块 axi_uart16550,将数据进行串行化并输出;非标准波特率 422 数据包同样也在 ARM 中进行打包并通过 AXI 总线传输,但由于 FPGA 中的 axi_uart16550 只能完成标准波特率的数据传输,所以,针对所需要的非标波特率,需在 FPGA 内采用自定义模块,根据协议来完成数据的串行化,并分配给相应的端口输出。

数据上传时,首先由 FPGA 从各个端口接收串行数据,同样,包括标准和非标准波特率 422 串行数据,标准波特率的由 IP 模块 axi_uart16550 完成串并转换并通过 AXI 总线传输到 ARM,非标准波特率则采用用户自定义的串并转换模块完成转换,再通过 AXI 总线传输给 ARM。上传数据在 ARM 汇总后,进行重新打包,再通过以太网传输至计算机。

3. 设计实现

3.1. 硬件及接口设计实现

硬件系统采用 Xilinx 公司的 Zynq 系列 XC7Z020 为核心,配合相应的外围驱动及接口芯片构建。框图如图 2 所示。

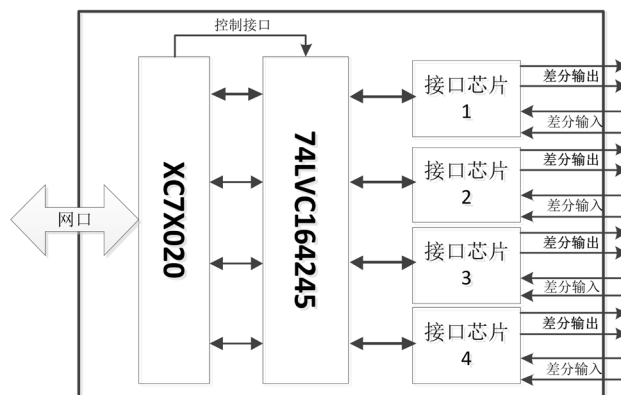


Figure 2. Hardware structure

图 2. 系统硬件框图

Zynq 系列作为 Xilinx 公司新一代的 FPGA,其内部由两个主要部分组成:一个由双核 ARM Cortex-A9 构成的处理系统和一个等价于一片 FPGA 的可编程逻辑部分。另外还具有集成的存储器、各种外设和高速通信接口等[4]。

FPGA 部分适合用来实现高速逻辑、算术和数据流子系统,而 ARM 支持软件程序和操作系统,这使得任何被设计的系统功能可以恰当地在一块芯片上做出硬件和软件之间做出划分。FPGA 和 ARM 之间的链接采用了高速的工业标准的高级可扩展接口 AXI 连接方式,如图 3 所示。

为提高 Zynq 输出接口的驱动能力,硬件设计上采用了 TI 公司的 74lvc164245 数字驱动芯片,不但提高了接口驱动能力,同时对外起到隔离保护作用。接口芯片的作用则是完成信号的差分 and 单端之间相互转换的功能。

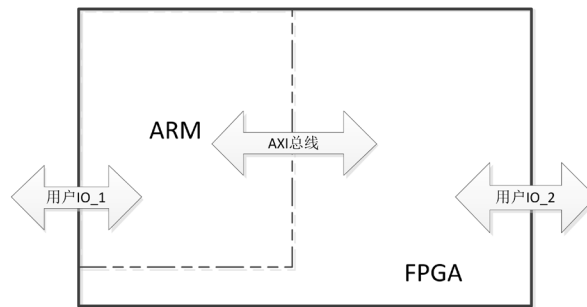


Figure 3. Core of Zynq
图 3. Zynq 内核架构示意图

3.2. ARM 部分设计实现

ARM 部分的编译环境采用了 Xilinx SDK[®]。设计中, ARM 作为一个中间单元, 充分发挥其在通信上的灵活性, 调用其中的网口驱动子模块, 来完成与上位机的通信[5] [6] [7]。为提高通信速率, 网口通信协议采用 UDP 格式。由于 UDP 协议本身存在信息传输上的不可靠性, 所以在用户层的信息帧格式中, 添加了帧流水号和帧尾的和校验。

为使得系统兼顾专用性和通用性, 在设计上, 将 422 分为两部分实现。一部分 422 驱动内核采用 ARM 实现, 另一部分则采用 FPGA 实现。工作中, ARM 接收计算机通过网口发送来的指令帧和数据帧, 若为指令帧, 则解析出指令帧中的指令码, 根据指令码的不同来完成相应的工作, 这其中包括: 串口选择、串口配置(波特率, 校验方式配置)等; 若为数据帧, 则分两种状况: 标准波特率 422 数据帧, 先在 ARM 中完成 422 的协议实现, 然后经由 AXI 总线传输给 FPGA 来最终输出(由于系统中, Zynq 的 422 对外接口均是从其 FPGA 部分的硬件管脚引出); 如果是非标准 422 数据帧, 则 ARM 不做处理, 直接打包通过 AXI 总线传输给 FPGA, 由下级的 FPGA 负责协议实现和驱动输出。

3.3. FPGA 部分设计实现

FPGA 作为 ARM 的下游设备, 负责高速和通用 422 信号的最终对外交互, 编译环境选用 Xilinx 公司的 Vivado[®]。

标准波特率 422 信号驱动协议层已由 ARM 部分完成, 串行数据通过 AXI 总线传输给 FPGA。在 FPGA 端, 只需在图形界面 Block Design 中, 调用 AXI 总线 IP 模块 axi_uart16550, 如图 4 所示, 便可将该串行指令通过 FPGA 的硬件端口 uart_rtl 对外进行数据交互[7]。IP 核的个数根据 422 端口需要可配置多个。这一类串口的特点是: 协议层由 ARM 完成, 灵活性强, 可通过网口动态重配。

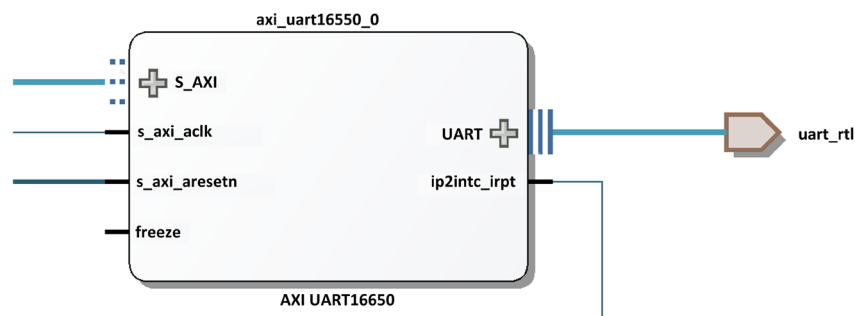


Figure 4. Core of axi_uart16550 IP
图 4. axi_uart16550 IP 核接口图

由于 ARM 受到内部寄存器配置及工作时钟的限制, 所以无法驱动和处理一些非标准波特率的串行信号。而 FPGA 因其具有很好的实时性和灵活的时钟配置, 使得在一些特殊以及较高波特率的工程应用中, 就需要由其来完成底层协议工作。系统中, 首先, 由 ARM 判断解码出来的数据, 如果属于非标准波特率数据, 则直接打包, 通过 AXI 总线发送给 FPGA 进行处理。FPGA 在收到总线上的数据后, 先进行解码, 解析出需要串行化输出的数据(数据位宽一般为 8bit), 然后, 按照设定的波特率将数据串行化输出到芯片引脚。同样, 在接收外部输入的串行数据时, FPGA 采用特定的波特率时钟对串行数据进行并行化处理, 转换成 8bit 位宽的数据, 再进行打包, 通过 AXI 总线传输给 ARM。FPGA 的底层驱动程序中, 有一个模块专门负责生成串口波特率时钟, 其基本原理是采用时钟计数的方式, 根据工程需求, 灵活配置波特率[8] [9]。

4. 实验验证与分析

针对标准和非标准波特率的端口, 采用了两种不同的通信回路进行实验验证。

4.1. 标准波特率通信实验验证

该类端口的验证采用了 USB 转 422 数据转接器来进行辅助验证。具体连接关系如图 5 所示。



Figure 5. Connection for standard baud rate test
图 5. 标准波特率验证实验接线图

实验中, 用网线将串口服务器和计算机相连接, 然后将一路 422 端口波特率配置为 115,200 bps, 再通过 USB 转 422 数据转接器连接至计算机。计算机端采用网络调试助手发送数据, 串口调试助手接收返回的 422 数据, 测试结果如图 6 所示。图中, 左侧的网口发送数据和右侧串口接收数据一致。

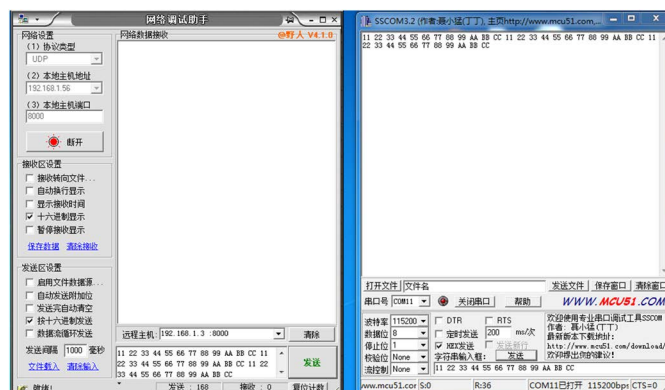


Figure 6. Transmitted data of Ethernet and received data of 422 serial port
图 6. 以太网发送和 422 串口接收数据对比图

4.2. 非标准波特率通信实验验证

由于采用的非标准波特率，所以无法接至计算机端采用串口调试助手进行观测，所以实验中采用路由器自闭环的方式进行验证。具体连接关系如图 7 所示。



Figure 7. Connection for non-standard baud rate test
图 7. 非标准波特率验证实验接线图

实验中，用网线将串口服务器和计算机相连接，然后将通信接口 1 的一路 422 和通信接口 2 的一路 422 波特率配置为 2 Mbps，并在硬件接口处闭环短接(通信接口 2 的 422 线路进行了收发交叉)。计算机端采用网络调试助手发送和接收数据，测试结果如图 8 所示，图中，下方的发送数据和上方的接收数据一致。



Figure 8. Transmitted data and received data of Ethernet
图 8. 以太网发送和接收数据对比图

4.3. 实验结果分析

上述实验结果表明，该串口服务器可实现标准和非标准波特率的发送和接收，且数据传输稳定。

5. 结论

相比于市场上的大多数采用 USB 转串口的服务器，该服务器采用了以太网，实现了长距离的串行数

据传输；同时，相比于现有的只采用 FPGA 或微处理器实现的串口服务器，多功能串口服务器在设计过程中，充分利用了 Zynq 中 ARM 内核和 FPGA 内核各自的优势，既实现了串口波特率的动态可重配，又实现了非标准波特率的串口通信。

参考文献

- [1] 胥飞燕, 郑华容, 周宦银, 等. 基于 FPGA 的多串口控制器的设计与实现[J]. 物联网技术, 2017, 7(10): 31-32.
- [2] 李伟. 基于 FPGA 的多串口通信设计与实现[J]. 数字技术与应用, 2015(2): 174-175.
- [3] 李辉, 张梦瑶, 张春晖. 基于 MicroBlaze 的串口通信设计[J]. 计算机与网络, 2016, 42(16): 61-63.
- [4] Crockett, L.H., Elliot, R.A., Enderwitz, M.A., *et al.* (2016) The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC. Strathclyde Academic Media, Strathclyde, Scotland, UK.
- [5] 倪继利. Linux 内核分析及编程[M]. 北京: 电子工业出版社, 2005.
- [6] Hollabaugh, C. (2002) Embedded Linux: Hardware, Software, and Interfacing. Pearson Education, New York.
- [7] 何宾. Xilinx FPGA 权威设计指南[M]. 北京: 电子工业出版社, 2015.
- [8] 陈发堂, 主父文刚, 胡利伟. 基于 ARM 的多 FPGA 串口命令选择配置及应用[J]. 自动化仪表, 2016, 37(10): 89-92.
- [9] 王春媚. 多串口并行通信数据传输系统设计[J]. 现代电子技术, 2017, 40(4): 86-90.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2163-3983, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: hjwc@hanspub.org