

doi: 10.7690/bgzdh.2021.02.013

# 基于 FPGA 的 SRIO 端点设计与实现

陈刚<sup>1</sup>, 康林<sup>2</sup>, 陈航<sup>1</sup>, 李坤贺<sup>1</sup>

(1. 中国兵器装备集团自动化研究所有限公司特种计算机事业部, 四川 绵阳 621000;  
2. 陆军装备部驻重庆地区代表局驻广元地区军代室, 四川 广元 628017)

**摘要:** 为提高芯片间及板间互连的带宽、灵活性和可靠性, 提出一种基于 FPGA (field programmable gate array) 的 SRIO (serial rapid IO) 端点的设计方法。介绍 RapidIO 的应用, 对 SRIO IP 核及其参数设置进行分析, 结合 Xilinx 提供的官方例程, 编写用户逻辑, 完成 FPGA 与 DSP 的高速通信, 测试结果表明: 该设计具有较高的带宽, 有一定的参考价值。

**关键词:** 嵌入式系统; 赛灵思; 高速通信  
**中图分类号:** TP336 **文献标志码:** A

## Design and Implementation of SRIO Endpoint Based on FPGA

Chen Gang<sup>1</sup>, Kang Lin<sup>2</sup>, Chen Hang<sup>1</sup>, Li Kunhe<sup>1</sup>

(1. Department of Special Computer, Automation Research Institute Co., Ltd. of China South Industries Group Corporation, Mianyang 621000, China; 2. Army Representative Office in Guangyuan, Army Representative Bureau of Army Equipment Department in Chongqing, Guangyuan 628017, China)

**Abstract:** In order to improve the bandwidth, flexibility and reliability of inter chip and inter board interconnection, this paper proposes a design method of SRIO (serial rapid IO) endpoint based on field programmable gate array (FPGA). Introduces the application of RapidIO, analyzes the SRIO IP core and its parameter setting, compiles the user logic with the official example provided by Xilinx, and realizes the high-speed communication between FPGA and DSP. The test results show that the design has a high bandwidth and a certain reference value.

**Keywords:** embedded system; Xilinx; high-speed communication

## 0 引言

随着高性能嵌入式系统的飞速发展, 芯片间及板间互连对带宽、成本、灵活性及可靠性的要求越来越高。传统的互连方式如 PCI、千兆以太网等已经不能满足上述要求。RapidIO 互连进行设计 (RapidIO 支持并行和串行 2 种模式。笔者主要针对串行 RapidIO, 即 SRIO), 有效解决了上述问题, 不但具有高带宽、低延迟、管脚少等特点, 而且对软件透明度很高, 通过纠错机制和点对点架构排除单点故障, 满足可靠性需求<sup>[1]</sup>。

笔者基于 Xilinx Kintex7 系列 FPGA 的 SRIO<sup>[2]</sup> 接口的设计和实现方法, 针对 Xilinx 提供的官方例程提出使用方法, 并在此基础上, 在 4 通道 (lane), 每个通道速度 5 Gbaud, 参考时钟为 125 MHz 的模式下与 DSP 进行了通信测试, 以 FPGA 作为发起者, 通过流写事务 SWRITE (streaming write), 完成了 FPGA 到 DSP 大数据量传输的测试与验证, 达到了较高的带宽, 可满足大多数应用要求。

## 1 RapidIO 简介

RapidIO 是为满足和未来高性能嵌入式系统需求而设计的一种开放式互连技术标准, 主要应用于嵌入式系统内部互连, 支持芯片到芯片、板到板间的通信, 可作为嵌入式设备的背板连接。RapidIO 协议由逻辑层 (logical layer)、传输层 (transport layer) 和物理层 (physical layer) 构成。逻辑层定义了所有协议和包格式。传输层为数据包从一个终端到另一个终端通道信息。物理层描述了设备之间的接口协议<sup>[3]</sup>, 例如报文传输机制、流量控制、电气特性及底层错误管理等。

从 2001 年 3 月至今, RapidIO 行业协会先后发布多个规范, 如表 1 所示。

表 1 RapidIO 协议发布历史

发布时间	版本	发布时间	版本
2001 年 3 月	RapidIO1.1 规范	2007 年 6 月	RapidIO2.0 规范
2002 年 6 月	RapidIO1.2 规范	2009 年 8 月	RapidIO2.1 规范
2005 年 2 月	RapidIO1.3 规范	2011 年 5 月	RapidIO2.2 规范

收稿日期: 2020-09-21; 修回日期: 2020-10-24

作者简介: 陈刚 (1984—), 男, 黑龙江人, 学士, 工程师, 从事数据采集、存储与数字信号处理研究。E-mail: chengang-32@163.com。

RapidIO 1.x 标准支持的信号速率为 1.25 Gbaud、2.5 Gbaud 和 3.125 Gbaud。RapidIO 2.x 标准在兼容 RapidIO 1.x 的基础上，增加了支持 5.0 Gbaud 和 6.25 Gbaud 的传输速率。包是 RapidIO 传送事务的基本单元。通过请求响应包的方式来完成 RapidIO 操作。包内包含确保把事务可靠传送到目标器件的位字段。发起器件产生一个请求事务包。该事务包将被发送至目标器件。目标器件在接收到请求事务包后产生一个响应事务包并且返回到发起器件以完成本次操作<sup>[4]</sup>。

## 2 基于 FPGA 的设计与实现

### 2.1 IP 核简介、参数设置及注意事项

Xilinx 提供的 SRIO 端点 IP 核 (SRIO Gen2 Endpoint) 符合 RapidIO2.2 规范，支持 1x、2x、4x 操作，每个通道 (lane) 支持 1.25、2.5、3.125、5.0 以及 6.25 Gbaud 的速率。图 1 是 IP 核的结构框图，从中可以看出，IP 核主要包含 2 个部分：1) SRIO 封装模块，包含物理层、逻辑及传输层和一个缓存设计；2) 时钟、复位及配置访问的参考设计。

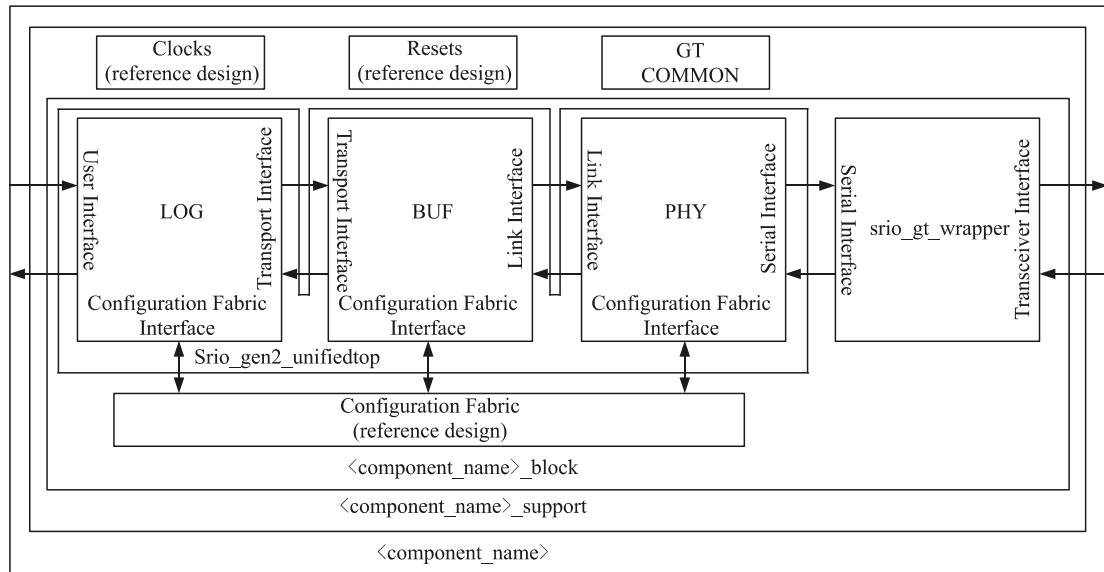


图 1 SRIO IP 核功能框图

在逻辑层的框图中可以看到一个用户接口 (user interface)，通过编写控制逻辑，用户可实现 FPGA 内部功能模块与 SRIO 接口之间的高速数据交换。笔者的设计采用以下设置：

链路宽度选择“4”，链路速率选择“5.0”，参考时钟选择“125”。其他选项选择默认值。如图 2 所示，在 Vivado 下，打开 SRIO IP 核即可进行相关设置。

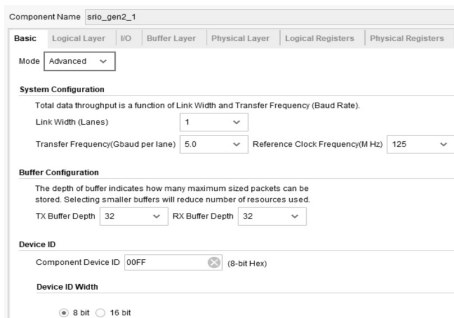


图 2 参数设置

Xilinx 提供 SRIO IP 核为软核，软核通常是用 HDL 文本形式提交给用户，经过 RTL 级设计优化

和功能验证，但其中不含有任何具体的物理信息。在工程综合、实现等过程中要加入合理的约束，可妥善处理跨时钟域问题，尤其在通道速率较高、器件资源紧张等情况下更要多加注意，以确保 IP 核能高效、稳定地工作<sup>[5]</sup>。

### 2.2 官方例程

在 SRIO IP 生成之后，可以直接在工程界面中生成官方例程。例程界面如图 3 所示。该例程为初次使用 IP 核的用户提供了极大的便利，经过简单设置即可完成 2 个 SRIO 端点间通信的仿真。通过仿真可以方便地观察到原始模块 (primary，模拟用户方端点) 向镜像模块 (mirror，模拟与原始模块连接的端点) 发送的 SWRITE、NWRITER、DOORBELL、MESSAGE 等多种类型的事务以及接收到的响应事务等。官方例程中除了包含 SRIO IP 核外，还有 2 个比较重要的功能模块：srio\_request\_gen.v 和 srio\_response\_gen.v。前者在端点作为发起者时产生请求事务，同时追踪响应事务，对接收到的响应事

务与预期值进行比较。后者在端点作为目标设备时针对接收到的请求，产生并发送对应的响应事务。

```

u_srio - srio_example_top_srio_gen2_0 (srio_example_top_sr
srio_gen2_0_inst - srio_gen2_0 (srio_gen2_0_xci)
srio_ireq_report_inst - srio_report (srio_report.v)
srio_request_gen_inst - srio_request_gen_srio_gen2_0 (s
srio_iresp_report_inst - srio_report (srio_report.v)
srio_tresp_report_inst - srio_report (srio_report.v)
srio_response_gen_inst - srio_response_gen_srio_gen2_0
srio_treq_report_inst - srio_report (srio_report.v)
srio_quick_start_inst - srio_quick_start_srio_gen2_0 (s
srio_statistics_inst - srio_statistics_srio_gen2_0 (sri
    
```

图 3 官方例程界面

srio\_request\_gen.v 还具有一个用户接口，接口定义如表 2 所示。在 FPGA 中，借助 VIO<sup>[6]</sup>工具，用户可以根据实际需求更改该用户接口信号，如目的地址、格式类型、数据量大小等，验证所要发送的事务是否正确，在实际应用中，则需要编写功能模块，控制该接口完成所需操作。

表 2 srio\_request\_gen.v 模块用户接口信号定义

接口信号名称	含义
user_addr	请求事务的目的地址
user_ftype	格式类型
user_ttype	事务类型
user_size	数据量大小
user_data	所要发送的数据
go	启动发送命令

### 2.3 大数据量传输的设计与测试

在基于 SRIO 的系统中，大数据量传输主要通过 SWIRTE 事务来实现<sup>[7]</sup>。笔者在官方例程的基础上，增加若干功能模块，编写了相应的接口逻辑，完成了功能设计与测试如图 4 所示。



图 4 数据传输与测试

主要功能模块说明如下：

**测试数据 (testData)：**产生测试数据，并将数据写入 FIFO (first input first output) 中，数据最终通过 SWRITE 事务发送至 DSP (digital signal processor)。

**缓存 (FIFO)：**存储测试数据，并在数据量达到设定的门限后通知下个模块读取数据。

**接口转换 (FIFO2SRIO)：**该模块是本设计的核心部分，将 FIFO 接口时序转换为 SRIO IP 核的用户接口时序，主要的功能包括读取 FIFO 数据、接收 DSP 下发的参数、控制用户接口产生 SWRITE、DOORBELL 事务等。该模块首先会监测 FIFO 内的数据量，当数据量达到设定的门限后，控制逻辑将

启动多次 256Byte 的 SWRITE 事务，完成全部数据的传输后，控制逻辑再发送 DOORBELL 事务，通知处理器读取数据，上述过程反复进行即可完成大数据量数据的传输功能。流程如图 5 所示。

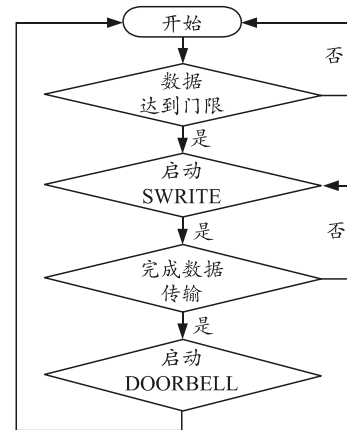


图 5 FIFO2SRIO 模块工作流程

测试流程分以下步骤：

- 1) DSP 对 SRIO 接口进行初始化；
- 2) DSP 向接口转换模块 (FIFO2SRIO) 下发目的地址、数据量大小等参数，启动测试，测试数据模块开始向缓存填入数据；
- 3) 数据传输完成后，DSP 读取数据，并根据整个流程的消耗的时间和传输数据量大小计算带宽。

在笔者的设计中，采用 Xilinx Kintex7 系列的 FPGA，SRIO 为 2.2 版本，4x 模式，链路速度为 5.0 Gbaud，理论速度为  $4 \times 5.0 \text{ Gbaud} / 8 = 2.5 \text{ GB/s}$ ，考虑到 8 B/10 B 编码效率为 80%，则  $2.5 \text{ GB/s} \times 80\% = 2 \text{ GB/s}$ 。经过以上流程测试，FPGA 发送 64 kB 字节所需时间约为 43.83 ms， $64 \text{ kB} / 43.83 \text{ ms} \approx 1.43 \text{ GB/s}$ ，达到传输理论值的 72%<sup>[8]</sup>。在实际应用中，上述传输过程会重复进行，随着传输次数的增加，传输带宽也会进一步提高，如连续传输 10 万次 64 kB 数据后，传输带宽可达 1.87 GB/s，更加接近理论带宽。

### 3 结束语

笔者基于 SRIO IP 核，在参考例程中各模块的作用后，增加若干用于测试大数据量传输的功能模块。经测试，带宽达到了 1.87 GB/s，突破了并行传输系统的速度瓶颈，满足了嵌入式系统的高速数据传输的需求。

### 参考文献：

[1] 杨涛. RapidIO 互联技术在下一代无线基站中的应用 [D]. 西安: 西安电子科技大学, 2008.

[2] Serial RapidIO Gen2 Endpoint v4.0 Product Guide PG007 [DB]. 2015-06-24.  
 [3] 胡欣, 魏龙, 陈怡君. 基于 FPGA 的雷达后端电路设计与实现[J]. 兵工自动化, 2019, 38(9): 39-43.  
 [4] 刘云鹏. 基于 FPGA 的 RapidIO 总线接口设计、验证与实现[D]. 西安: 西安电子科技大学, 2013.  
 [5] Vivado Design Suite User Guide Using Constraints UG903 (v2016.2)[DB]. 2016-08-08.

[6] Vivado Design Suite Tutorial Programming and Debugging UG936(v2015.1) [DB]. 2015-05-18.  
 [7] Xilinx Answer 50166 LogiCORE IP Serial RapidIO Gen2 Debugging and Packet Analysis Guide [DB]. 2013-03-02.  
 [8] 姜宏旭, 刘亭杉, 李辉勇, 等. FPGA+DSP 异构视频处理系统中基于 SRIO 的数据高效传输方法[J]. 计算机学报, 2010, 38(6): 1119-1130.

\*\*\*\*\*

(上接第 44 页)

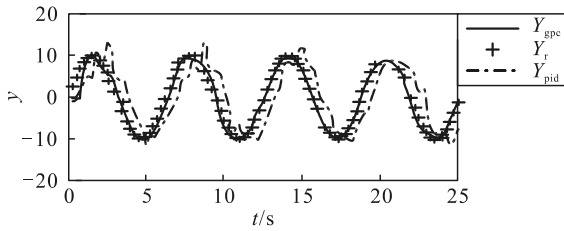


图 3 正弦信号的跟踪输出

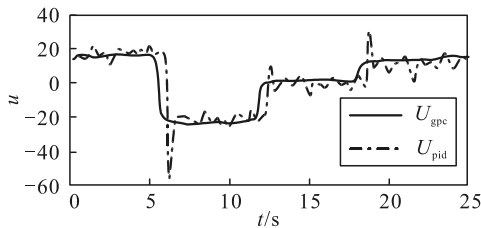


图 4 组合阶跃信号的跟踪控制量

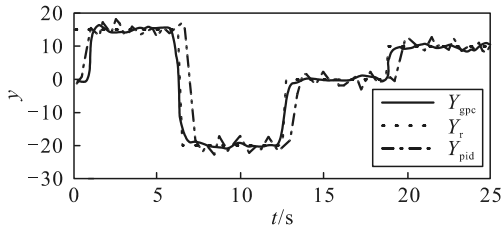


图 5 组合阶跃信号的跟踪输出

从图 2—5 中可以看出:滑模预测控制器可以有效地跟踪目标信号,并提前预测未来的输出,从而实现了消除时滞的作用。将预测控制的思想引入到滑模变结构中,给出了滑模预测控制器的设计方法,并对其参考轨迹进行了改进,通过仿真验证了该方案可以有效减弱抖动对系统的不利影响,对系统参数变化和阻力扰动具有很强的鲁棒性。并且,该方法还可以有效地削减噪声所带来的误差,保证跟踪目标的准确性。同时,与常规 PID 控制器相比,滑模预测控制基本没有超调量,响应迅速,具有很好的鲁棒性。

### 5 结束语

笔者针对纯时滞系统的快速跟踪问题,设计了滑模预测控制器。通过应用广义预测控制理论来预

测未来输出,消除时滞造成的误差,使得滑模控制方法无需变换即可直接应用在时滞系统中。通过与常规 PID 控制器的仿真结果对比可知:该方法不仅能克服时滞,而且能抑制系统的超调量,提升系统的响应速度;对存在噪声和干扰的情况,还可以有效地减少误差,保证整个系统的鲁棒性,即证明了滑模预测方法在纯时滞系统中的有效性和优越性。同时,该方法结构简单,无需大量在线运算,在保证一定存储空间的前提下,控制性能良好,具有一定的推广和研究价值。

### 参考文献:

[1] BI Q, CAI W J, LEE E L, et al. Robust Identification of First-order Plus Dead-time Model from Step Response[J]. Control Engineering Practice (S0967-0661), 1999, 7(1): 71-77.  
 [2] 李超, 黄德先, 金以慧. 利用阶跃响应求解一阶加纯滞后系统参数区间的鲁棒辨识算法[J]. 化工自动化及仪表, 2003, 30(5): 25-27.  
 [3] UTKIN V I. Variable structure systems with sliding modes[J]. IEEE Trans on AC, 1977, 22(2): 212-222.  
 [4] 高为炳. 变结构控制理论基础[M]. 北京: 中国科学技术出版社, 1990: 85-88.  
 [5] 郑峰, 程勉, 高为炳. 控制存在时滞的系统的变结构控制[J]. 控制与决策, 1993, 8(2): 95-99.  
 [6] FIAGBEDZI Y A, PEARSON A E. A multistage reduction technique for feedback stabilizing distributed time-lag systems[J]. Automatica, 1987, 23(3): 311-326.  
 [7] 郑尚德. 变结构预测控制[J]. 信息与控制, 1993, 22(6): 347-352.  
 [8] 符建国, 曾庆军, 谢成祥. 基于预测模型变结构控制交流伺服系统, 电机与控制学报, 2003, 7(2): 143-146.  
 [9] 王伟. 广义预测控制理论及其应用[M]. 北京: 科学出版社, 1998: 25-28.  
 [10] GAO W B, WANG Y F, ABDOLLAH H. Discrete-time variable structure control systems[J]. IEEE Transaction on Industrial Elections, 1995, 42(2): 117-122.  
 [11] 杨朗. 冲压发动机推力控制系统研究[D]. 哈尔滨: 哈尔滨工业大学 2010: 12-13.