

天基网络化通用服务平台的硬件实现^①

田 雪, 倪 明, 施华君, 凌幸华

(中国电子科技集团公司第三十二所, 上海 201808)

通讯作者: 田 雪, E-mail: 735179848@qq.com

摘 要: 针对天基系统星上处理能力不足、技术落后、无法实现多源数据在轨实时处理等难题, 本文提出一个应用于中低轨微小卫星的天基网络化通用服务平台的设计概念, 为现在的卫星技术发展提供新的硬件架构, 同时异构的体系结构既可以实现星上的独立运行, 又可以为星载计算机的数据处理提供硬件加速. 本文主要介绍天基网络化通用服务平台的硬件设计, 包括芯片选型、对外关系以及所需的存储容量和为实现接口的兼容性, 在 FPGA 上设计实现了 MIL-STD-1553B 总线的总线控制器向远程终端发送数据的相关模块.

关键词: 天基系统; 网络化通用服务; 异构架构; 接口兼容; MIL-STD-1553B

引用格式: 田雪, 倪明, 施华君, 凌幸华. 天基网络化通用服务平台的硬件实现. 计算机系统应用, 2018, 27(1): 45-51. <http://www.c-s-a.org.cn/1003-3254/6164.html>

Hardware Implementation of Space-Based Network Universal Service Platform

TIAN Xue, NI Ming, SHI Hua-Jun, LING Xing-Hua

(The 32nd Institute of China Electronics Technology Group Corporation, Shanghai 201808, China)

Abstract: For the lack of processing power and backward technology, satellites cannot achieve real-time processing of multi-source data in orbit in the space-based system, so this paper presents a design of a space-based network universal service platform for micro satellites in the medium and low orbit. This design provides a new hardware architecture for the development of current satellite technology. At the same time, this heterogeneous architecture cannot only achieve the independent operation of the satellites, but it can also provide hardware acceleration for the data processing of the onboard computers. This paper mainly introduces the hardware design of space-based network universal service platform, including chip selection, external relations, and the required storage capacity. In order to realize the compatibility of the interface, the module of the MIL-STD-1553B bus is designed and implemented on the FPGA to send data from the bus controller to the remote terminal.

Key words: space-based system; universal network service; heterogeneous architecture; interface compatibility; MIL-STD-1553B

目前, 发展天地一体化、抢占天基网络化信息竞争制高点成为各大国的又一次战略机遇. 天地一体化信息网络是以天基传输网络、地面传输网络和天基信息服务为核心构成的网络, 其中载荷技术是天基网络中研究重点之一^[1-2]. 然而, 目前我国卫星的数据处理能

力差, 卫星与卫星之间无法通信和协同工作, 针对上述需要以及为实现星上独立实时存储、处理和传输数据的能力, 提出了天基网络化通用服务平台的概念. 天基网络化通用服务平台以“标准化、模块化、软件化”设计理念为指导, 采用低成本、商业货架式产品, 可兼容

① 收稿时间: 2017-04-06; 修改时间: 2017-05-09; 采用时间: 2017-05-11; csa 在线出版时间: 2017-12-22

现役卫星平台,以板卡的形式安装在目标卫星上,具有自主运行和管理的能力,以及智能化的状态信息获取、储存、处理和分发能力,达到对空间资产的实时监控、空间资源的高效利用^[3].

1 功能与对外关系

1.1 主要功能

(1) 信息采集:具备信息采集能力,可连接卫星平台星务系统和载荷管控系统,获取平台及载荷的状态监控信息.

(2) 组网联通:具备自组网能力,可接入天基网络管理分中心,支持专题信息发送和管控信息接收.

(3) 处理与存储:具备信息处理和存储能力,支持状态监控、管理控制和任务规划等计算需求,未来根据应用需求可支持载荷管控及星务计算等能力.

(4) 管理控制与任务规划:具备对宿主星的管理控制能力,接收天基网络管理分中心的任务指令,生成任务规划信息并执行运行管控.

1.2 对外关系

如图1所示,天基网络化通用服务平台兼容现有卫星平台与载荷,具有网络、处理、存储能力,可连接载荷管控计算机和星务计算机获取状态控制信息并能与其协同处理,提供面向天基物联网应用的互联互通、智能感知、按需服务等能力.

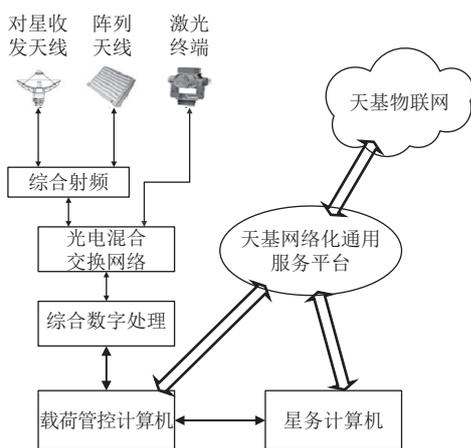


图1 对外关系

2 初步方案设计

2.1 硬件架构

硬件架构如图2所示,服务平台主要由存储模块、通用处理模块、网络通信模块、接口匹配模块组

成,针对接口兼容需求,提供基于软件定义的方法设计扩展接口.

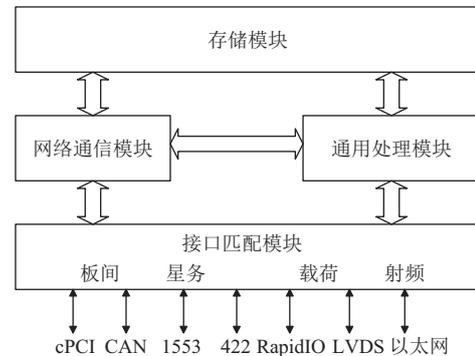


图2 硬件架构

通用服务平台硬件模块:

(1) 网络通信模块:提供卫星平台、空间站等天基资源的互联、互通.

(2) 通用处理模块:基于低功耗的通用处理器进行设计,提供高性能的处理、计算能力.

(3) 存储模块:基于Flash存储阵列进行设计,具备单粒子翻转检测控制能力.

(4) 接口匹配模块:根据天基电子系统接口需求,灵活扩展通信接口.

为实现所需功能,需要外接Flash和DDR3器件,以及在芯片的可编程逻辑(PL)部分设计所需的接口总线,最终设计成一个3U的VPX模块,基本的硬件模块原理图如图3所示.

2.2 芯片选型

天基网络化通用服务平台是面向应用、技术密集、高度集成化的一个硬件模块,选用低电压、高主频、低功耗、微封装的ARM和FPGA才能够满足本系统功能和性能指标需求.基于高集成度通用芯片设计,完成微小卫星中低轨应用.因此,在该硬件设计中,芯片选用Xilinx的ZYNQ-7000系列开发板进行搭建和实现的,ZYNQ-7000分离了处理器和FPGA系统,搭载的双核ARM A9处理器同时结合了工程上最新的28nm工艺制程的可编程逻辑器件,使其相较之前的FPGA,占用空间更小,静态功耗降低多达50%,系统性能提升多达50%,容量提升2倍.同时,采用ARM和FPGA结合的方式,既具备了ARM的操作系统可嵌入性,又具备了FPGA的接口可拓展性.因此,相较于单FPGA结构,ARM+FPGA的异构结构具有更高的

性能. 这样的结构也为在星上发展异构加速提供了基础, 例如异构多核 SoC 实现对 SAR 雷达信号的处理^[4]. ZYNQ-7000 芯片内部模块如图 4 所示.

与传统 CPU+FPGA 的双芯片结构不同, ZYNQ-7000 是一个以处理器为中心的器件, 它将处理器和可编程逻辑集成到一个芯片内, 每款器件都是基于处理器的系统, 能够通过可访问的可编程逻辑重设时即可

启动操作系统^[5]. 系统架构师和嵌入式软件开发人员可以通过串行 (基于 ARM 处理器) 和并行 (基于 FPGA) 处理相结合的方式, 满足各种日趋复杂的高性能应用需求, 同时可以利用其高度集成的优势大大降低成本和功耗, 并缩小产品尺寸. 这样异构的结构使得该平台在实现计算、存储以及通信功能后易于实现为载荷计算提供协同处理的能力.

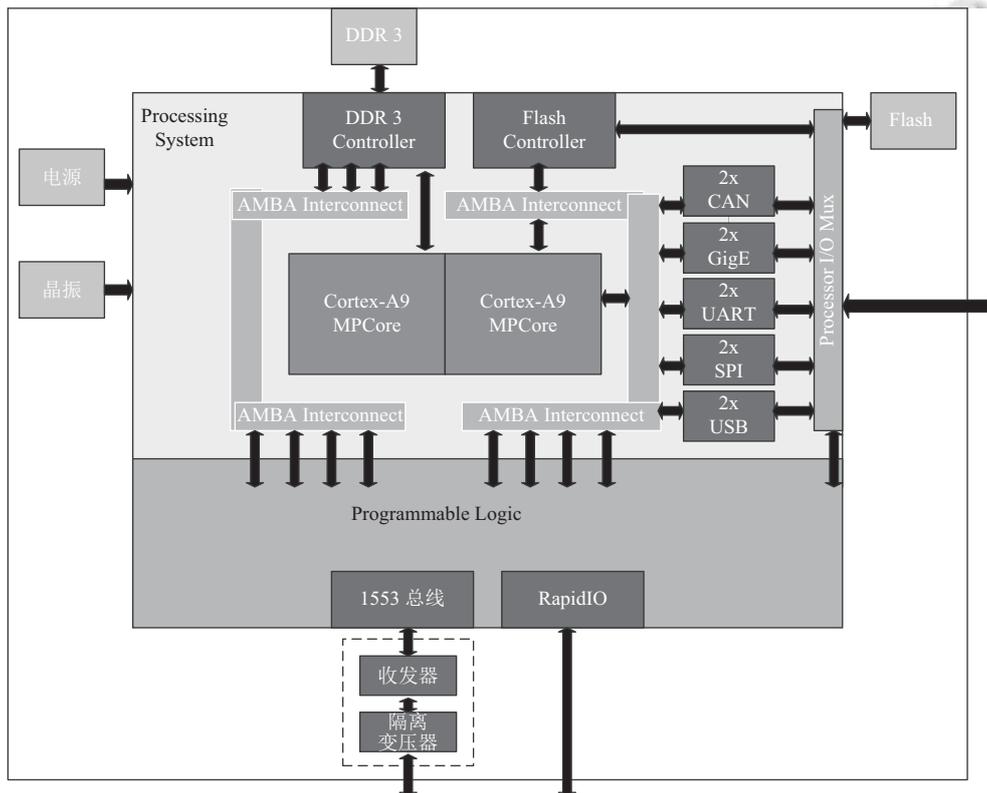


图 3 硬件模块原理图

3 存储模块

(1) Flash 存储器, 在没有电流供应的条件下也能够长久地保存数据, 其存储特性相当于硬盘, 这项特性使得 Flash 成为各类便携式数字设备的存储介质^[6].

Zynq-7000 中具有静态存储控制器可以作为 NAND Flash 控制器. NAND Flash 接口模式如下:

- 1) ONFI 标准 1.0.
- 2) 可连接高达 1 GB 的设备.
- 3) 单一片选有 8/16-bit 的 IO 宽度.
- 4) 具有 16-word 读写数据的 FIFO.
- 5) 具有 8-word 命令的 FIFO.
- 6) 可编程 IO 循环时间.

7) 具有软件协助的 1-bit ECC.

8) 具有异步存储操作模式^[7].

(2) DDR3 存储器, 属于 SDRAM, 具有 8 bit 的预取设计, 采用点对点的拓扑架构, 以减轻地址/命令与控制总线的负担; 采用 100 nm 一下的生产工艺, 将工作电压降至 1.5 V, 增加异步重置与 ZQ 校准功能^[8].

Zynq-7000 中具有动态存储控制器可以作为 DDR3 控制器. 多协议的 DDR 存储控制器可以被配置成提供 16-bit 或 32-bit 宽度的通道连接到一个单一等级配置的 8-bit、16-bit 或 32-bit DRAM 存储器的 1 GB 地址空间. ECC 在 16-bit 总线模式下被支持. 该 DDR 存储控制器是多端口且使得处理系统 (PS) 和可编程逻辑

辑 (PL) 通过共享通道访问一个共同的内存, 因此 DDR 控制器具有四个 AXI 从端口来实现该目的。

- 1) 一个 64-bit 端口通过二级高速缓存控制器用于 ARM CPU.
- 2) 两个 64-bit 端口用于 PL.
- 3) 一个 64-bit AXI 端口通过中央互联被所有其他

AXI 主端口共享^[7].

为满足指标要求即存储能力不小于 2 GB, 选用 MT41K512M16 器件 2 片, 这是因为 MT41K512M16 为 16-bit DDR3, 有行数 64 K、列数 1 K 和 8 个 bank, 所以一个 MT41K512M16 的内存容量为 512 M*2 Byte=1 GB, 故为达到 2 GB 的容量要求, 需要有 2 个 MT41K512M16.

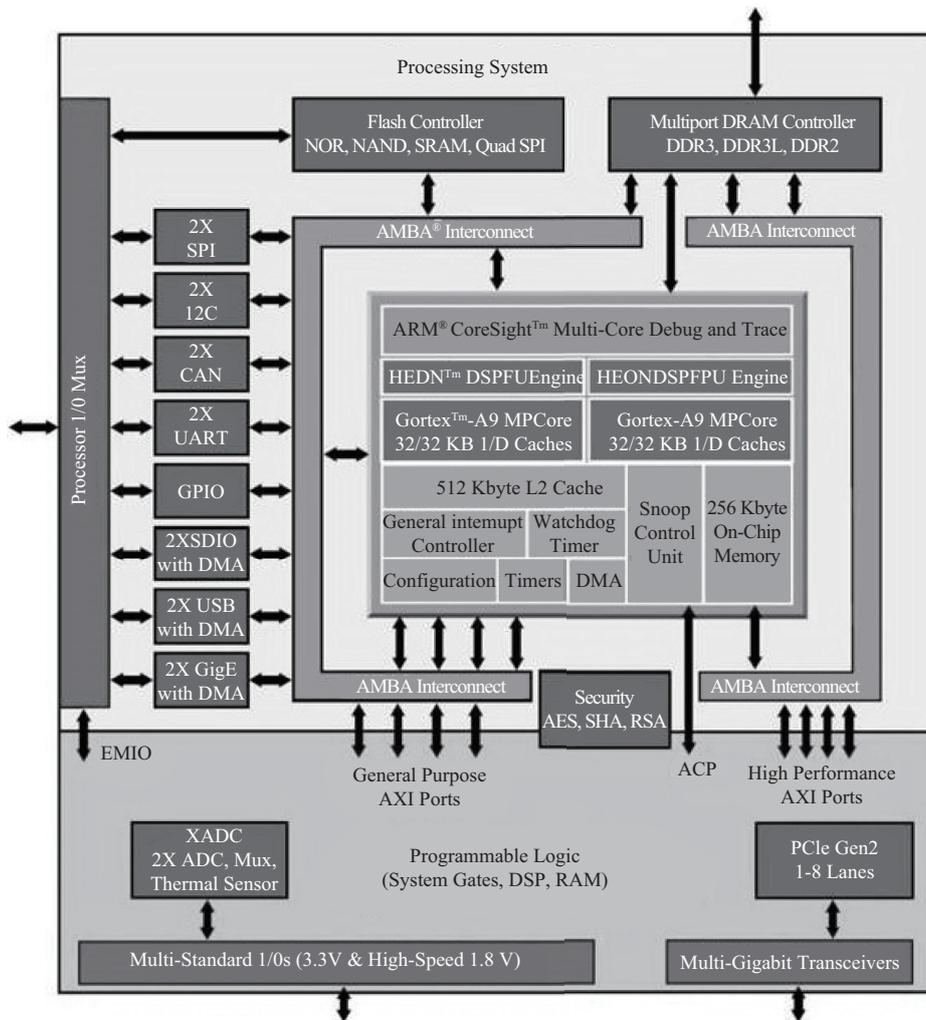


图 4 芯片内部模块图

4 接口匹配模块

SoC 芯片采用的是 Xilinx 的 Zynq-7000 内部集成了接口匹配模块所需的 CAN 总线、PCI 总线、LVDS、RS-422 以及千兆以太网. 由于要实现接口的兼容, 实现通用性, 所以根据软件的需求, 还要实现 1553B 以及 RapidIO 总线接口, 下面介绍 MIL-STD-1553B 总线在 FPGA 上的设计实现, 主要实现总线控制器 (BT) 到远程终端 (RT) 的数据传输。

MIL-STD-1553B 总线: 美国军用标准, 是现代航天电子系统广泛应用的通信总线, 其数据传输速率可达到 1 Mbps, 抗干扰能力强, 实时性好, 主要用于电子系统内部的时分、指令/应答和多路传输. 通过 1553B 总线可连接多达 32 个子系统 (RT 终端), 完成各子系统的通信和数据交换^[9]. 通常对于 1553B 总线接口的设计一般基于 DDC 公司的 61580 协议芯片完成, 但只能完成协议处理部分, 应用时还需要外围的存储器和

微处理器等辅助芯片,导致成本高,功耗大,占用PCB面积大,故为了实现项目目标的小型化和低功耗,我们采用Zynq-7000中可编程逻辑(PL)部分实现整个接口核心的设计。

总线接口包括隔离变压器、模拟收发器、曼彻斯特编解码器和协议处理逻辑三大模块。其中,隔离变压器和模拟收发器完成FPGA输出信号与总线信号之间的电平转换,可由专用转换芯片完成,而曼彻斯特编解码器和1553B协议处理逻辑是接口的主要组成部分,完成数据编解码和协议处理,通过Zynq-7000中可编程逻辑(PL)部分实现^[10]。

1553B总线信号进入接口板后,首先通过隔离变压器进行电平转换,使之匹配后面收发器的工作电压。收发器再把电平转化后1553B差分信号转化为可识别的TTL电平^[11],之后进入Zynq-7000中可编程逻辑(PL)部分完成相应处理。原理图如图5所示。

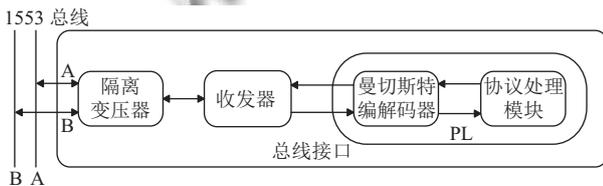


图5 1553B原理图

4.1 曼彻斯特编解码模块设计

4.1.1 1553B总线数据格式

1553B总线上的数据是以消息的形式调制成曼彻斯特II码在进行传输的,每条消息的长度最长为32个字,所有的字分为三类:命令字,数据字和状态字。字的长度为20位,每个字的前3位是该字的同步字头,之后的16位是有效信息位,最后的1位是奇偶校验位,采用奇校验。其中,只有有效信息位和奇偶校验位在总线上是曼彻斯特码的形式传输的,传输码率位1MHz,即传输一位用时1 μ s。同步字头占3位,其中命令字和状态字是1.5 μ s的高电平和1.5 μ s的低电平,数据字是1.5 μ s的低电平和1.5 μ s的高电平^[12]。

4.1.2 编码模块设计

编码模块主要是对待发送的消息字进行曼彻斯特编码,之后再对其并串转换然后发送。由于在曼彻斯特编码中,每位数据中间都有一个跳变沿,逻辑‘1’编码表示为先高后低,逻辑‘0’编码表示为先低后高,所以在发送数据时需要将每位数据分成两位。为了满足

1553B总线速率为1Mb/s的要求,要使编码后的每一位数据占0.5 μ s,因此编码模块需要使用2MHz时钟来完成数据的编码与发送。其工作过程为:

(1) 同步字头编码,若待发送的消息字为命令字或状态字,则同步字头编码为“111000”,若为数据字,则编码为“000111”;

(2) 对16位数据位和奇偶校验位进行编码,将逻辑‘1’编码为“10”,逻辑‘0’编码为“01”;

(3) 最后将并行的消息字进行串行转换,由低到高进行输出。

关键Verilog代码如下:

```
// Determine the sync pattern based on word type.
always @(posedge enc_clk or negedge rst_n) begin
    if (!rst_n) sync_bits <= 6'b000_000;
    else if (tx_csw) sync_bits <= 6'b111_000;
    else if (tx_dw) sync_bits <= 6'b000_111;
    else sync_bits <= sync_bits;
end

// Generate Manchester encoded data for combined
sync pattern,
// data word and parity.
assign enc_data = {sync_bits, data_reg[0], ~data_reg[0],
data_reg[1], ~data_reg[1], ....., data_reg[16], ~data_reg[16],
1'b0};
// Serialize the encoded data
always @(posedge enc_clk or negedge rst_n) begin
    .....
    else if (cnt_en || cnt_en_reg) begin
        tx_dval <= 1'b1;
        tx_data <= enc_data[busy_cnt];
    end
    .....
end
```

4.1.3 解码模块设计

解码模块主要是对接收到的数据进行曼彻斯特解码,之后在对其进行串并转换。解码模块接收数据首先收到同步头,解码模块采用8MHz的时钟,故在检测连续收到12个‘0’和12个‘1’或12个‘1’和12个‘0’时,就表示已经收到了有效的同步字头,可以进行之后的数据接收。

对于同步字头之后的数据接收解码,由于解码时

钟采用 8 MHz 且总线上数据传输速率为 1 Mb/s, 故对接受到的数据每 8 位采一次样. 由于接收到的每个数据占 8 位, 所以如果采样在前 4 位, 则解码后数据值就为本次采样数据; 如果采样在后 4 位, 则解码后数据值为本次采样数据取反. 在对接收到的数据完成解码后, 将数据并行输出.

关键 Verilog 代码如下:

```
always @(posedge dec_clk or negedge rst_n) begin
    if (!rst_n)
        dword_int <= 17'h0000;
    else if (data_sample && cnt_enb)
        dword_int <= {dword_int[1:16], ~data_sftreg[2]};
    else if (!cnt_enb) dword_int <= 17'h0000;
    else dword_int <= dword_int;
end
always @(posedge dec_clk or negedge rst_n) begin
    if (!rst_n) begin
        .....
    end
    else if (cnt == 'd131) begin
        rx_dword <= dword_int[0:15];
        rx_dval <= 1'b1;
        rx_perr <= ((^dword_int[0:15]) != dword_int[16]);
        rx_csw <= sync_csw_reg;
        rx_dw <= sync_dw_reg;
    end
    .....
end
```

4.2 BC 发送模块设计

在 BC 模式下, 每一个 BC 消息块中的第一个字都是 BC 控制字, 它不在总线上传输, 而是包含所有 BC 发送数据所需要的信息. 控制字之后就是 BC 所要发送的命令字给要接收的 RT, 然后是 BC 要发送的数据字, 之后 BC 等待 RT 返回的状态字.

在这里我使用了状态机进行 BC 发送模块任务的编写, 工作流程如下文.

(1) 在空闲状态下, 当 RT 接收开始的信号有效, 进入一个延时状态, 为了之后加入检错重传提供方便.

(2) 之后状态机进入命令字发送状态, 这里将命令字数据赋给曼彻斯特编码模块进行编码和发送, 判断发送结束标志, 结束则进入数据字发送状态, 未结束则

重新跳转到该状态继续发送命令字直到结束.

(3) 在数据字发送模块下, 同样将要发送的数据字赋给曼彻斯特编码模块进行编码和发送, 这里要判断发送指定数据字的个数是否完成, 若完成则进入等待状态字的状态, 未完成则重新进入该状态继续发送下一个指定的数据字直到完成.

(4) 数据字发送完成后开始等待远程终端返回的状态字, 当检测到有数据返回时, 开始进行曼彻斯特解码, 最后得到状态字.

该模块使用状态机进行代码编写, 状态转换图如图 6 所示.

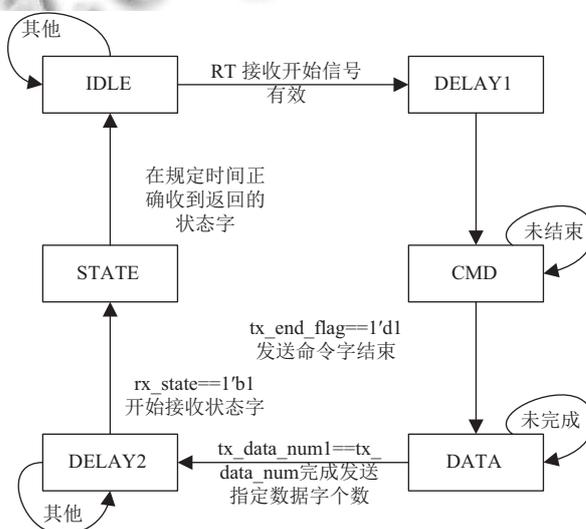


图 6 状态转换图

4.3 设计仿真

根据上述设计方案, 编写了相应模块的 Verilog 代码, 仿真实现 BC 向 RT 发送数据的功能, 一共发送 6 个数据字, 采用 ModelSim 仿真软件, 仿真结果如图 7 所示.

5 结论与展望

本文提出了一个天基网络化通用服务平台的设计概念, 主要介绍了其硬件的设计方法, 同时为了实现对接口的兼容性, 介绍 1553B 总线控制器在 Zynq-7000 开发板中的 PL 部分的设计, 实现了曼彻斯特编解码, 并完成其中总线控制器向远程终端发送数据的相关模块基本功能, 但对于天基网络化通用服务平台的最终实现, 还有很长一条路要走. 为满足平台接口的兼容性, 下一步要完善 1553B 总线的其他模式功能以及 RapidIO

在 FPGA 中的实现,完成平台的硬件设计.天基网络化通用服务平台的提出,对未来我国实现天基信息港以

及天地一体化的航天互联网构想奠定了重要的基础以及关键技术的突破^[13,14].

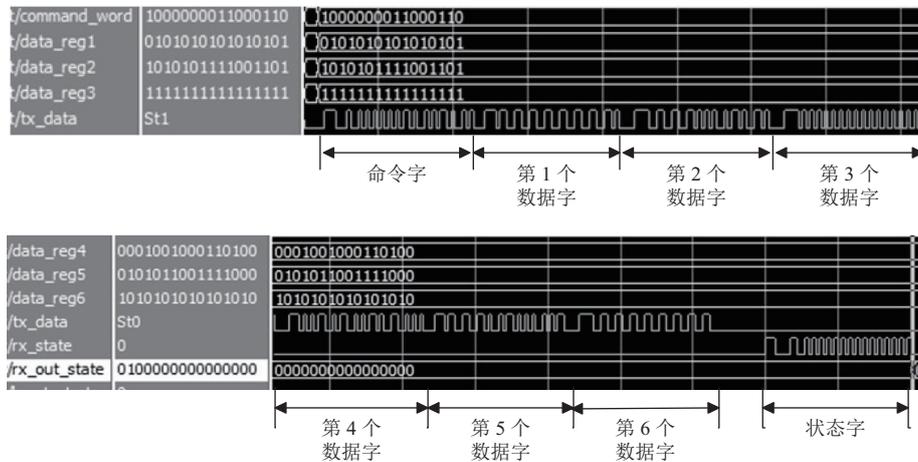


图7 仿真结果图

参考文献

- 孙晨华. 天基传输网络和天地一体化信息网络发展现状与问题思考. 无线电工程, 2017, 47(1): 1-6.
- 张杰, 郁小松. 天地一体化网络中卫星通信港控制架构与路由技术研究. 无线电通信技术, 2017, 43(2): 1-5.
- 李智勇, 卞国良. 天基综合信息网的设想及其与国防信息网的关系. 2000 军事电子信息学术会议论文集(上册). 呼和浩特, 中国. 2000.
- 宋立国, 胡承秀, 亓洪亮. 面向 SAR 雷达信号处理的异构多核 SoC 研究. 电子技术应用, 2017, 43(1): 50-52, 56.
- 佚名. Zynq-7000 可扩展处理平台. 世界电子元件, 2011, (6): 43-46.
- 佚名. 微型存储技术. 个人电脑, 2004, 10(2): 212-216.
- Crockett LH, Elliot RA, Enderwitz MA, et al. The Zynq book: Embedded processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 all programmable SoC. Strathclyde Academic Media, 2014.
- 史晓蓉. 基于 DDR3 数据的信号完整性分析[硕士学位论文]. 西安: 西安电子科技大学, 2013.
- 马萌. 航空专用数据总线技术研究. 数字技术与应用, 2013, (10): 61-63.
- 洋朋军, 朱浩文. 基于 FPGA 的 1553B 总线接口设计与验证. 现代电子技术, 2015, 38(3): 26-30.
- 黄伟, 吴国安, 汤清华, 等. 基于 FPGA 的 1553B 总线接口板的设计. 2006 中国科协年会. 北京, 中国. 2006.
- 李宪强. FPGA 项目开发实战讲解. 北京: 电子工业出版社, 2015.
- 沈荣骏. 我国天地一体化航天互联网构想. 中国工程科学, 2006, 8(10): 19-30. [doi:10.3969/j.issn.1009-1742.2006.10.004]
- 闵士权. 我国天基综合信息网构想. 航天器工程, 2013, 22(5): 1-14.