

基于 FPGA 的天文频谱仪 网络接口设计

于 威^{1,2}, 罗近涛^{1,2}, 张秀忠¹, 项 英¹

(1. 中国科学院上海天文台, 上海 200030; 2. 中国科学院研究生院, 北京 100049)

摘 要: 为了更方便地观察和分析处理天文频谱仪计算得到的频谱数据, 需要通过数据接口把数据传送到计算机上。简单介绍了如何用基于 FPGA 的嵌入式系统设计网络接口, 给出了网络接口各硬件模块的功能和软件的流程图以及测试结果。

关 键 词: 频谱仪; 网络; FPGA; 嵌入式

中图分类号: P228.4

1 引 言

谱线观测是射电天文观测的重要组成部分, 通过分子谱线观测等, 可获得天体的速度、物质结构组成等方面的重要信息。谱线观测终端的核心部分是频谱仪。随着电子技术的发展和数字信号处理方法的改进, 越来越多的射电天文终端实现了数字化, 传统无线电接收机中的模拟方法, 被软件无线电方法和数字化电路代替。上海天文台在开发 DBBC (Digital Base Band Converter, 数字基带转换器)^[1]的基础上, 开展了基于 DBBC 硬件平台的数字天文频谱仪的研究。

天文频谱仪可观测的最大谱线带宽为 64 MHz。天文频谱仪的信号流程图如图 1 所示。来自太空的射电信号被射电天线接收后, 首先通过模拟混频器转换成中频信号, 中频信号经过信号处理得到频谱数据, 再通过数据接口传送到计算机, 并存储起来以便进一步分析。信号处理部分又分为 2 部分: 基带转换单元和频谱变换及累加单元。基带转换由数字基带转换器实现, 数字基带转换器具有 AD (模数转换) 功能, 首先对天线输出的中频信号进行采样, 然后把需要的各频带信号通过频谱搬移转换成基带信号。频谱变换及累加单元把基带信号转换为频谱数据。

数据接口是数字频谱仪的重要组成部分, 它将频谱仪的处理结果传送到计算机, 以便进行图形显示等处理, 以及谱线研究等后续的深入分析。连接 FPGA (Field Programmable Gate Array) 与计算机的数据接口一般有 3 种: 串口、USB 接口、网络接口。在这 3 种接口方式中, 串口传输速率比较低, USB 接口虽然也可以达到较高的传输速率, 但是若用 FPGA 实现会非常复杂, 而且串口和 USB 接口都不能远距离传输。网络接口可以达到很高的传输速率, 操作也非常灵活, 而且可以远距离传输。因此在本文介绍的天文频谱仪中, 数据接口采用网络接口。网络接口使用基于 FPGA 的嵌入式系统实现, 该接口负责将天文频谱仪的输出结果通

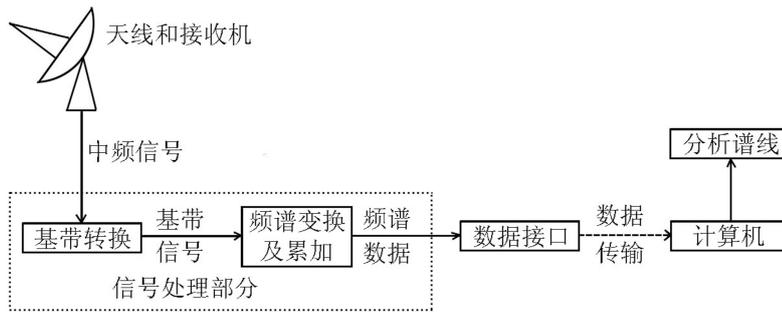


图 1 天文频谱仪系统信号流程图

过网线发送到计算机。在测试中，频谱仪每 10s 输出 16 384 个 32 位的数据，通过对计算机接收到的数据进行分析，确认数据传输正确，能得到正确的谱线，网络接口能满足使用要求。

2 网络接口的硬件

网络接口的硬件框图如图 2 所示，网络接口分为 2 大部分：数据接收部分和数据发送部分。数据接收部分在通用计算机上用软件实现。数据发送部分，除了物理层芯片和 DDR 存储器 (双速率同步动态随机存储器) 外，全部集成在一片 FPGA 上。FPGA 选用 Xilinx 公司 virtex4 FX60 芯片，此款 FPGA 最典型的特性是集成了 2 片 32 位 RISC (精简指令集) 嵌入式硬核 CPU PowerPC 405^[2]，因此非常适合构建嵌入式系统。频谱仪的信号处理部分在另一片 FPGA 芯片 virtex4 LX160 上实现。信号处理部分和网络接口间的数据交互通过 1 个 32 位的同步 FIFO (先进先出队列) 实现。

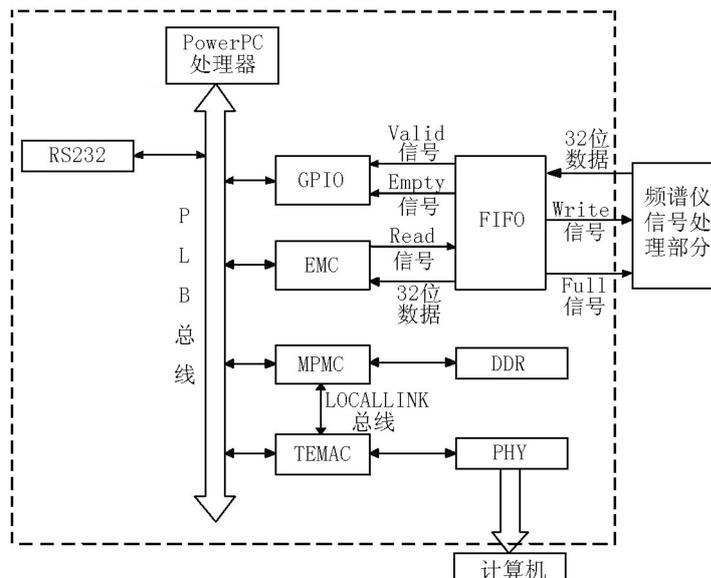


图 2 网络接口系统硬件原理图

信号处理部分输出的 32 位数据首先进入 FIFO, PowerPC 通过轮询的方式读取 FIFO 中的数据, 并把数据送入 DDR 中的数据缓存 sendbuf 中, 然后判断 sendbuf 是否已满, 如果 sendbuf 已满则把数据打包成 TCP/IP 格式通过网络发送到计算机上。整个系统在 Xilinx 公司的 ISE 和 EDK 2 个集成开发环境下完成。下面分别介绍硬件部分各模块的具体功能。

(1) 处理器总线采用 PLB 总线 (处理器本地总线), 图 2 中大部分的模块都是直接挂载在 PLB 总线上, 在构建系统时可以很方便地增减各模块。PLB 总线是一种高带宽、低延迟、高性能的处理器内部总线^[3]。PLB 总线为指令和数据提供独立的 32 位地址和 64 位数据接口。PLB 总线支持多主从设备, 每个 PLB 主机通过独立的地址总线、读数据总线和写数据总线与 PLB 总线连接, 对于每条数据总线都有 1 套复杂的传输控制和状态信号。为了允许主机通过竞争获得总线的所有权, 有一个中央判决机构, 它授予 PLB 总线上的设备不同的优先权对 PLB 总线进行访问。

(2) FIFO 采用 Xilinx 公司自带的 IP core (知识产权核), 采用异步置位, 深度为 16384, 数据位宽为 32 位, FIFO 起到数据缓存的作用。对数据进行缓存的另一种较常用方法是采用双端口 RAM (随机存取存储器), 但是如果采用双端口 RAM, 信号处理部分和网络接口间除了数据接口外还需地址接口, 会占用比较多的 FPGA 管脚, 接口所占管脚越多数据出错的概率越大。而且 FIFO 比双端口 RAM 操作更简单。因此综合考虑, 本系统采用 FIFO。在数据交互时, 如果 FIFO 不满, 信号处理部分可向 FIFO 传数据, 在 FIFO 不空的情况下 PowerPC 可读取其中的数据。

(3) GPIO (通用输入输出端口) 把 FIFO 的 2 个控制信号与 PLB 总线连接起来, 这 2 个控制信号分别为表示 FIFO 是否空的信号 Empty 和表示读取数据是否有效的信号 Valid。CPU 通过 GPIO 中的数据判断是否可以读取 FIFO 中的数据, 且读取的数据是否有效。

(4) EMC (外部存储器控制器) 用来对 FIFO 进行扩展, 其数据输入端口与 FIFO 的数据输出端口相连, 其信号线 Read 与 FIFO 的读数据使能信号线相连。EMC 的功能是帮助 CPU 将 FIFO 中的数据移到数据缓存 sendbuf 中。当 FIFO 中数据不空, 即 Empty 信号为低电平时, CPU 给 EMC 发出指令读取 FIFO 中的数据, EMC 接到 CPU 的读数据指令后, Read 信号变成高电平, FIFO 中的数据被读出, 并送入 EMC 的数据输入端口, EMC 再把数据通过数据输出端口送入 sendbuf, 供 CPU 处理。

(5) PowerPC 把从 EMC 收到的数据, 通过 MPMC (多端口内存控制器) 送入 DDR 内存, 并将数据按照 TCP/IP 格式打包、解包, 当数据打包完成后, PowerPC 控制 MPMC, 使数据通过 LOCALINK 总线进入 TEMAC (三态媒体接入层控制器), 并最终通过网口发送出去。

(6) MPMC 在本系统中用到 4 个端口, 其中 2 个端口分别接 PowerPC 的数据端处理器本地总线 (DSPLB) 和指令端处理器本地总线 (ISPLB), 另外 2 个端口 1 个挂接到 PLB 总线上, 1 个接到 LOCALINK 总线上与 TEMAC 连在一起。

(7) DDR 内存由 2 片 Infineon HYB25D256160BT-7 芯片组成, 共有 64×10^6 字节的存储空间, 每片 16 位位宽, 两片组成 32 位位宽。

(8) PHY 芯片 (物理层芯片) 采用的是 marvell 公司的 88E1111 芯片。

(9) TEMAC 模块调用 Xilinx 公司的硬 IP core, 其自带 2 个 LOCALINK 接口, 通过 LOCALINK 总线与 MPMC 连接在一起。通过 LOCALINK 总线传输数据, 与通过 PLB

总线传输数据相比,可以大大简化外围设备的逻辑,节省 CPU 资源,而且能大大的提高 IP 层到 MAC 层的数据传输速率,非常方便数据直接从 DDR 进入 TEMAC。另外,TEMAC 还具有计算传输层校验和的功能和损坏数据帧自动过滤的功能。计算校验和的功能把原来 CPU 所做的部分工作卸载到了硬件上,这样不必等待 CPU 花费很多指令周期去计算校验和(TCP 校验和的计算是对 1 个 TCP 数据包中所有数据进行累加求和,会浪费比较多的 CPU 指令周期)。同样,帧过滤的功能也分担了 CPU 的工作,使得 CPU 有更多的空闲时间去做别的事情。

(10) LOCALLINK 接口是专门针对 Xilinx 公司 IP core 的一种接口标准,它是一种点对点、高性能、采用同步方式的数据传输接口。LOCALLINK 接口的一端是发送端,另一端是接收端,因此如果 2 个模块间要进行双工通信,就需要 2 组 LOCALLINK。本系统中 MPMC 和 TEMAC 间正是用了 2 组 LOCALLINK,本系统中 LOCALLINK 接口数据位宽为 32 位,时钟频率为 100 MHz,因此数据传输能力非常强。LOCALLINK 包括 1 组在传输数据时起同步作用的控制信号,具体说明如表 1 所示。LOCALLINK 把要传输的数据打包成帧,每一帧数据包括帧头、有效负载数据和帧尾,有效负载数据是要传送的实际数据,帧头、帧尾则包括工作模式和控制信息。帧头和帧尾的大小根据两端的收发情况而不同。本系统中,MPMC 向 TEMAC 发送数据时,帧头为 8 个 32 位的数据,帧尾为 1 个 32 位的数据,此时波形图如图 3 所示。TEMAC 向 MPMC 发送数据时,帧头为 1 个 32 位的数据,帧尾为 8 个 32 位的数据^[4],此时波形图如图 4 所示。

表 1 LOCALLINK 各信号描述

信号名称	方向	功能简介
CLK	输入	同步所需的时钟
RST	输入	当此信号为高电平时,数据传输无效
DATA	源端到目的端	数据通过这个数据总线传输
SRC_RDY_N	源端到目的端	源端准备就绪,等待传输数据
DST_RDY_N	目的端到源端	目的端准备就绪,等待接收数据
SOF_N	源端到目的端	每帧数据的起始标志
EOF_N	源端到目的端	每帧数据的结束标志
SOF_N	源端到目的端	有效负载数据传输开始标志
EOP_N	源端到目的端	有效负载数据传输结束标志
REM	源端到目的端	EOF 有效时,数据帧的最后一个字节在 DATA 中的位置

3 网络接口的软件

网络接口的软件分为 2 部分,嵌入式系统下的软件和计算机上的软件。嵌入式系统下软件的功能是读取频谱仪中的数据并把数据通过网络发送出去,计算机上软件的功能是通过网络接收数据并把数据存成文件。本系统采用客户端/服务器模式,这是网络传输最常用的模式,嵌入式系统作为客户端发送数据,计算机作为服务器接收数据。

为了保证传输数据的可靠性,本系统采用了 TCP/IP 协议。计算机上的软件基于 LINUX 操作系统, TCP/IP 协议通过调用 LINUX 内核 TCP/IP 协议栈提供的标准 API 函

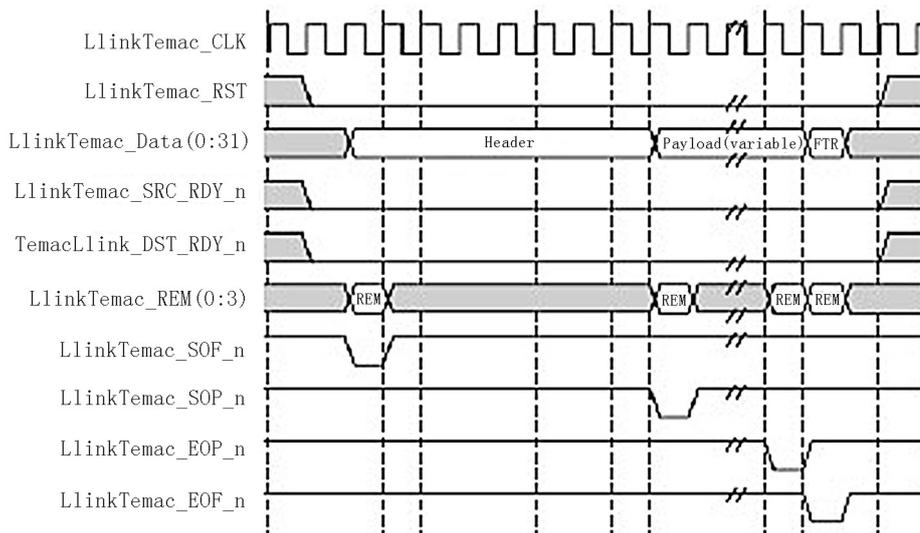


图 3 MPMC 向 TEMAC 传输数据 LOCALLINK 波形图

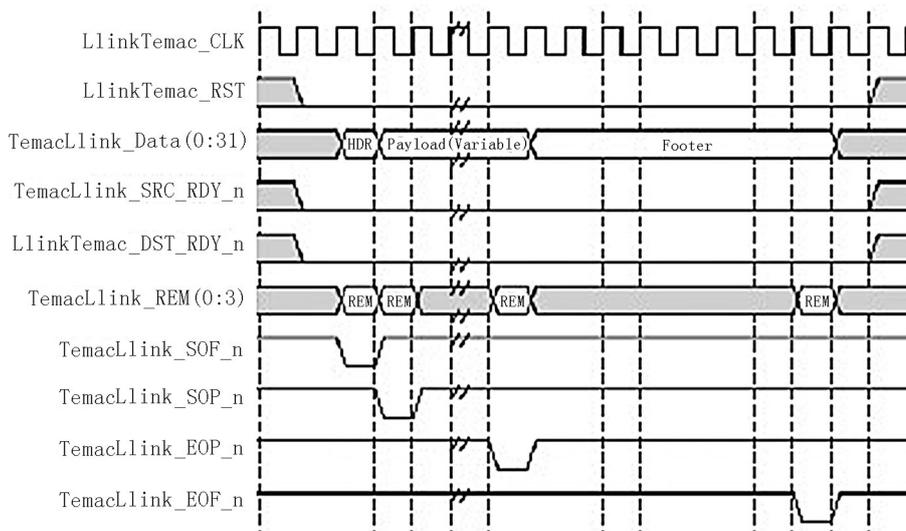


图 4 TEMAC 向 MPMC 传输数据 LOCALLINK 波形图

数实现。嵌入式系统下使用 TCP/IP 协议一般有 2 种方式：(1) 移植嵌入式操作系统，如 LINUX, vxworks 等；(2) 只移植 TCP/IP 协议栈。移植嵌入式操作系统的优点是，嵌入式 CPU 可以多线程工作，在进行网络工作的同时还可以进行其他工作，但这些操作系统大部分售价较高，且移植非常复杂。只移植 TCP/IP 协议栈较为简单，且开源的协议栈比较多，但是 CPU 只能单线程工作。对于本接口要实现的功能，单线程可满足要求，因此综合考虑，本接口采用只移植 TCP/IP 协议栈的方式。本接口移植的协议栈是 Treck 公司的 TCP/IP 协议栈。此 TCP/IP 协议栈是 Treck 公司专门为嵌入式系统设计的 TCP/IP 协议栈，可以很容易地移植到 PowerPC 处理器上^[5]。

数据发送端 (嵌入式系统) 的软件流程图如图 5 所示，具体如下：

(1) 对 TEMAC 和 Treck 协议栈进行初始化工作, 配置 TEMAC 所需参数, 调用 Treck 库, 并建立二者间的接口 (TEMAC 为二层模块, Treck 实现三层四层协议, 二者间需要传输数据的接口)。

(2) 构建一个用来接收 FIFO 中数据的缓冲区 sendbuf, 大小设为 2048 字节。

(3) 基于 Treck 库生成网络通信所需的 socket 套接字, 然后用 CONNECT 函数与已经启动的服务器建立 TCP 连接。

(4) 连接成功后, 对 EMC 操作读取 FIFO 中的数据, 并把读到的数据放入缓冲区 sendbuf, 直到 sendbuf 已满。

(5) 发送 sendbuf 中的数据, 并等待服务器回送 ACK 确认。

(6) 发送成功后, 判断是否还有数据要发送, 如果有, 回到操作 (4), 否则跳到操作 (7)。

(7) 关闭连接。

数据接收端计算机软件流程图如图 6 所示, 具体如下:

(1) 生成网络通信所需的 socket 套接字。

(2) 建立接收数据的缓冲区 recvbuf (2048 字节), 并建立新文件 text.txt 用来存储接收到的数据。

(3) 调用 Bind 函数把生成的 socket 描述符与端口捆绑在一起。

(4) 调用 Listen 函数, 监听是否有客户端的服务请求。

(5) 如果有合法的客户端服务请求到来, 则调用 Accept 函数, 通过三次握手协议与客户端建立起 TCP 连接。

(6) TCP 连接成功后, 调用 Recv 函数接收数据并将其存于 recvbuf 中, 回送 ACK 给客户端, 表明已正确接收到数据。

(7) 如果收到的是客户端发送的有结束标志位的数据包, 则跳转到操作 (9), 否则跳转到操作 (8)。

(8) 把 recvbuf 中的数据保存到 text.txt 文件中, 并跳转到操作 (6)。

(9) 与客户端进行四次握手关闭连接。

4 系统性能测试

FPGA 中的嵌入式系统布局布线后所占用的 FPGA 资源如表 2 所示。

表 2 嵌入式系统占用的 FPGA 资源

资源名称	资源总数目	占用的数目	占用百分比 /%
BUFG	32	9	28
BUFGCTRL	32	1	3
DCM_ADV	23	2	16
EMAC	2	1	50
RAMB16	232	31	13
Slice	25 280	6 285	24
SLICEM	12 640	165	1

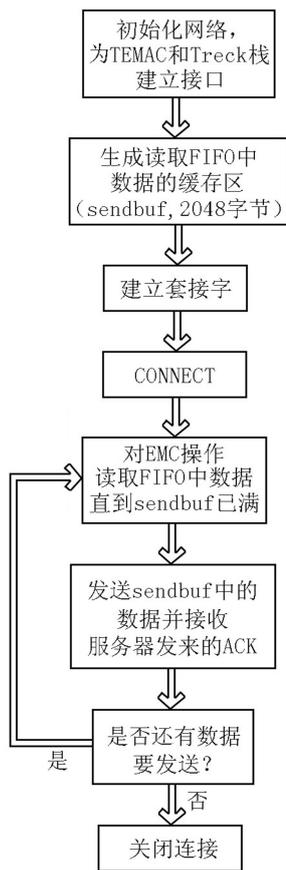


图 5 嵌入式系统的软件流程

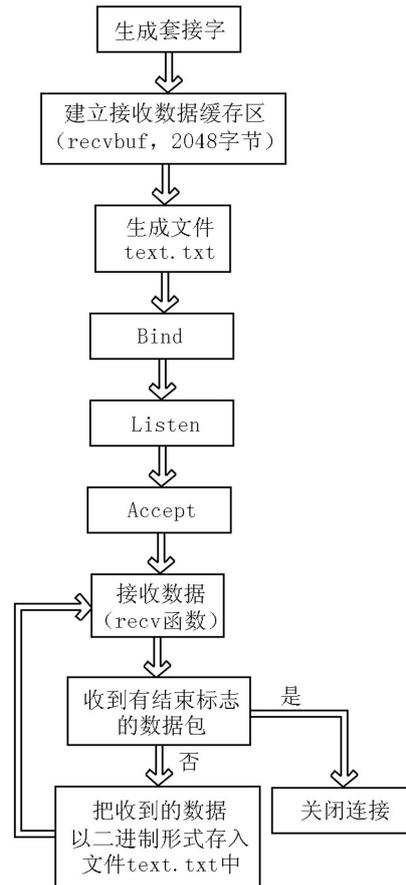


图 6 计算机端的软件流程

FPGA 资源占用情况的主要指标是 Slice 和 RAMB16。从表 2 可以看出, 本系统所占资源并不多, 是一个小型系统, 可以很容易地移植到低端的 FPGA 上, 也可以很容易嵌入到其他系统当中。

频谱仪中 FFT (快速傅里叶变换) 的长度为 16 384, 积分时间为 10s, 因此频谱仪信号处理部分每 10s 传出 16 384 个 32 位的数据 (16 384 为 1024 的整数倍)。为了使频谱仪传出的每批数据的数据量正好是 TCP 数据包中数据量的整数倍, 在实际设计中, 把本接口 TCP 数据包所携带的数据量设定为 1024 字节, 再加上 TCP 报头 32 字节, IP 报头 20 字节, 因此本系统的 MTU 设定为 1076 字节。经过测试, 在 sendbuf 设定为 2048 字节的情况下, 网络接口的传输速率完全可以满足实用要求。当 sendbuf 增大时网络速率仍可增大。通过 IPtraf 测试网络速率最大可达 120 Mbps。

为了验证网络接口数据传输的正确性, 在发送端循环发送 16 384 个 32 位累加的数据 (数据从 0 累加到 16 383 为一个周期), 计算机收到的数据如图 7 所示 (图中显示了 50 个周期的循环数据), 图中横坐标是点数, 纵坐标是数据大小, 由图可以看出接收到的数据完全正确。

在整个频谱仪系统的验证实验中, FFT 的输入为基带转换部分输出的 24 MHz 的复信号, 信号时钟为 128 MHz。计算机收到的数据的频谱图如图 8 所示。图中横坐标是数据的点

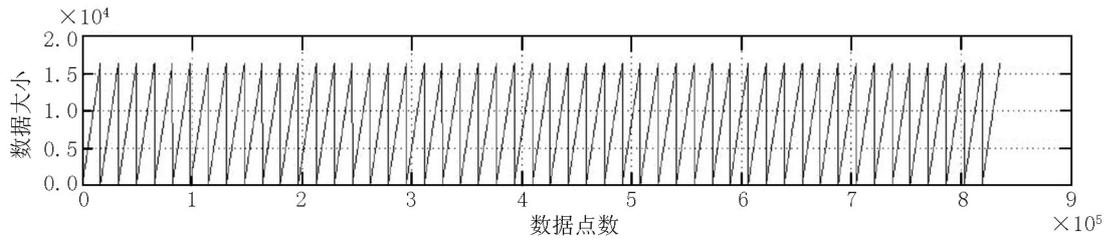


图7 接收到的累加数据

数, 纵坐标是数据的幅度, 其中 2 个明显的频点分布在第 3072 点和第 13312 点。用公式 (1) 可以把横坐标由点数变换为实际的频率值。

$$x \times 128 / 16384 = y, \quad (1)$$

其中 x 为图 8 中的点数, y 为实际的频率值 (单位为 MHz)。当 x 为 3072 时 y 为 24 (MHz), 当 x 为 13312 时 y 为 104 (MHz), 24 (MHz) 正是采样前模拟信号的频率值, 24 (MHz) 和 104 (MHz) 关于采样率的一半 (即 64 MHz) 对称, 因此频谱图完全正确。

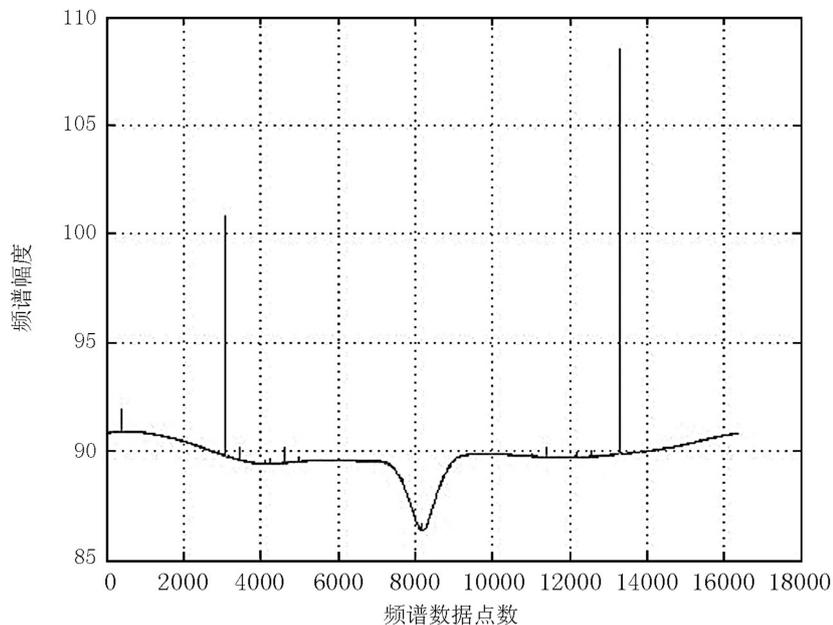


图8 24 MHz 正弦信号的频谱图

5 结束语

本接口实现了天文频谱仪与计算机之间的数据传输, 通过测试证明数据传输完全正确, 能够满足使用要求。而且本接口是一个相对独立的系统, 并不局限于传输天文频谱仪的数据, 任何基于 FPGA 的设备, 如果要把数据送入计算机, 都可以很容易把本接口嵌入其中。

参考文献:

- [1] 项英. 博士论文. 上海: 中国科学院上海天文台, 2005: 13
- [2] 田耘, 徐文波. Xilinx FPGA 开发使用教程. 北京: 清华大学出版社, 2008
- [3] Xilinx Processor Local Bus(PLB)v4.6(v1.02a) datasheet
- [4] Xilinx XPS LL TEMAC(v1.01a) datasheet
- [5] 李健, 张秀忠, 项英. 中国科学院上海天文台年刊, 2007, 28: 128

The Design of Network Interface of Astronomical Spectrum Analyzer Based on FPGA

YU Wei^{1,2}, LUO Jin-tao^{1,2}, ZHANG Xiu-zhong¹, XIANG Ying¹

(1. *Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030;*
2. *Graduate School of Chinese Academy of Sciences, Beijing 100049*)

Abstract: In order to observe and analyze the data of the astronomical spectrum analyzer, we need to transmit the data to the computer by data interface. This paper introduces how to design the network interface by using embedded system based on FPGA, and describes hardware components of the system, the software flow and the test results of the system.

Key words: spectrum analyzer; network; FPGA; embedded system