

基于 FPGA 的 IEEE488 接口协议实现 及驱动程序开发

刘 阳, 罗显志, 周杨罗, 李 政

(湖北大学 计算机与信息工程学院, 武汉 430062)

摘要: IEEE488 作为一种通用的并行总线, 凭借其低延时、可靠传输等优势, 广泛应用于仪器控制与测试系统中; 针对目前符合 IEEE488 接口协议的专用 GPIB 芯片如 NAT9914、uPD7210、MC68488 等都已停产, 且在售 GPIB 控制器产品类型少且价格十分昂贵, 难以满足国内市场需求的现象, 采用了一种基于 FPGA 技术来实现 IEEE488 接口协议的方案, 以典型的 GPIB 接口芯片 NAT9914 为蓝本, 使用 Verilog 语言描述 GPIB 接口功能, 调用 VISA 虚拟仪器函数库开发驱动程序与设备通信, 组建完整的自动测试系统, 并通过 SCPI 指令对仪器进行程控、数据采集等操作; 该方案有效地解决了专用 GPIB 芯片停产的问题, 同时开发成本低; 经实验测试实现了 IEEE488 接口协议功能, 产品性能稳定, 各项功能满足测试系统中的测控需求, 具有一定的工程实用价值。

关键词: IEEE488; GPIB; NAT9914; VISA; SCPI; 自动测试系统

Implementation of IEEE488 Interface Protocol and Driver Development Based on FPGA

LIU Yang, LUO Xianzhi, ZHOU Yangluo, LI Zheng

(School of Computer and Information Engineering, Hubei University, Wuhan 430062, China)

Abstract: As a general parallel bus, IEEE488 is widely used in instrument control and test systems with its advantages of low delay and reliable transmission. At present, the special GPIB chips conforming to the IEEE488 interface protocol, such as NAT9914, uPD7210 and MC68488, have been discontinued, and the products of GPIB controller on sale have few types and are very expensive, which is difficult to meet the needs of the domestic market, a scheme based on FPGA technology is adopted to realize the IEEE488 interface protocol, based on the typical GPIB interface chip NAT9914, the GPIB interface function is described in Verilog language, the visa virtual instrument function library is called to develop the driver to communicate with the equipment, a complete automatic test system is established, and the instrument is programmed and the data is collected through the SCPI instructions. The scheme effectively solves the shutdown problem of special GPIB chip, meanwhile the development cost is low. Through the experimental test, the IEEE488 interface protocol function is realized, the product performance is stable, and all functions meet the measurement and control requirements in the test system, which has certain engineering practical value.

Keywords: IEEE488; GPIB; NAT9914; VISA; SCPI; automatic test system

0 引言

在自动测试系统中, IEEE488 总线扮演着重要的角色。它是一种面向仪器设备的通用总线接口, 也被称为 GPIB 总线^[1-2]。该接口最早由 HP 公司提出, 经过几十年的发展, 已经形成了一套行业标准。且该接口能满足设备之间互相通信的功能, 实现信息共享的需求。

近年来, 国内在自动化控制、精密仪器研制等领域发展迅猛, 在医疗、军工等领域, 对测试系统的需求越来越大。GPIB 作为业界使用最广泛的测试总线, 不仅成本

低, 而且传输稳定, 在未来的一段时间内, 国内市场仍然有一定的需求量。

市面上的 GPIB 转换器基本都采用专用接口芯片来设计, 此类芯片连接微处理器, 通过读写寄存器等操作能很方便的实现 GPIB 接口的所有功能。目前, 能自主生产基于 IEEE-488 协议的专用 GPIB 接口芯片的公司仅 NI、TI、NEC、Intel 几家^[3], 但该类芯片都已停产, 可供购买的成品 GPIB 转换器如 NI-GPIB-USB-HS 价格在 3 000 元以上, 对国内庞大的用户群体来说成本过高。国内在测试接

收稿日期: 2022-02-18; 修回日期: 2022-03-21。

作者简介: 刘 阳(1996-), 男, 湖北十堰人, 硕士研究生, 主要从事嵌入式开发、自动控制等方向的研究。

通讯作者: 罗显志(1976-), 男, 湖北十堰人, 博士, 研究员, 硕士生导师, 主要从事软件无线电、实时高速信号处理、北斗卫星导航系统等方向的研究。

引用格式: 刘 阳, 罗显志, 周杨罗, 等. 基于 FPGA 的 IEEE488 接口协议实现及驱动程序开发[J]. 计算机测量与控制, 2022, 30(8): 134-141.

口领域的研究较少, 仅有电子科大等几个团队, 且市面上国产转换器占有率不高。随着 ASIC 技术的飞速发展, FPGA 在嵌入式领域得到了广泛应用, 它灵活、稳定、可移植性强, 所以能够用来替代一些专用芯片。

基于以上因素, 本文详细阐述使用 FPGA 芯片来实现 IEEE488 总线协议的方案及实现流程, 为相关开发人员提供可行性参考。

1 总体设计

基于 FPGA 的 IEEE488 协议转换器的设计主要包括硬件和软件两部分。硬件部分包括硬件电路设计与硬件控制程序设计; 软件部分主要是驱动程序开发, 最后通过用户程序与仪器通信。该方案组建一套完整的测试系统, 其总体结构^[4]如图 1 所示。

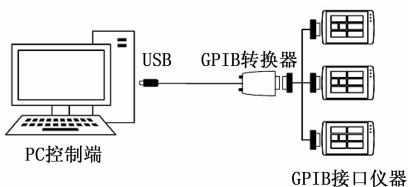


图 1 总体结构框架

该方案的核心在于硬件控制程序设计, 开发前需要对 IEEE488 总线协议做深入研究, 同时剖析 NI 的 NAT9914 的内部结构, 掌握其协议的工作原理。硬件控制程序的主要内容是模拟 GPIB 接口的十种功能, 然后通过读写控制模块控制数据准确无误的传输, 在 PC 与仪器之间搭建数据传输的桥梁。软件驱动程序连接 PC 与转换器, 实时发送程控指令并接收返回的数据。

2 硬件电路设计

硬件电路设计主要包括电源模块、USB-UART 模块、FPGA 主控模块及外围电路、GPIB 接口模块。硬件设计框架如图 2 所示。

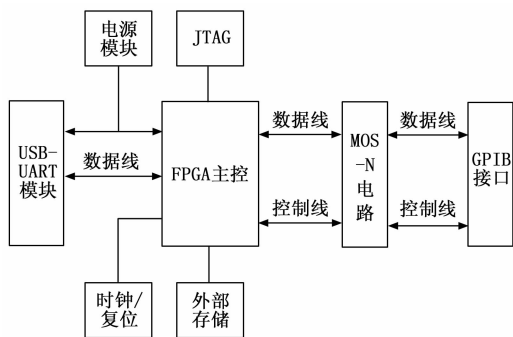


图 2 硬件设计框架

2.1 电源模块

电源模块为整个系统提供稳定电压。电源芯片使用 MC34063, 该芯片性能好, 市场应用广泛, 可实现升压和降压功能, 但其实现电路不同。本系统采用 USB 接口供

电, 其电压大致为 5.2 V 左右, 所以需设计降压电路。FPGA 芯片的不同 BANK 所需电压不同, 因此添加 LM1084 可调输出线性稳压器, 提供 3.3 V、2.5 V、1.2 V 工作电压。电源电路部分设计如图 3 所示。

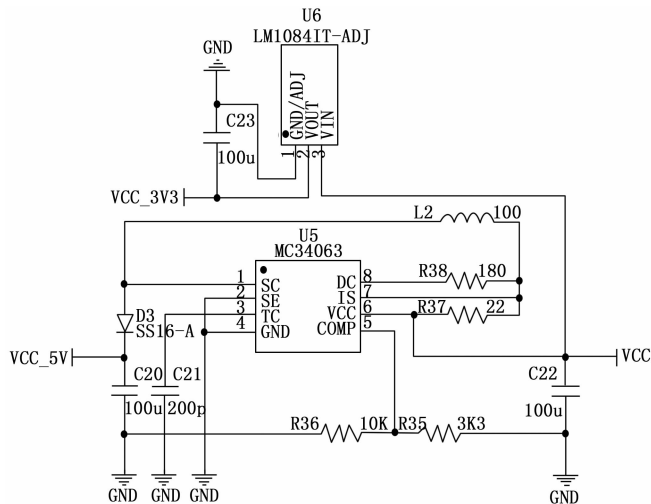


图 3 电源模块

该芯片内部有恒定的 1.25 V 基准电压, 输出电压与 R35 和 R36 的阻值有关, 其输出电压的计算公式为:

$$V_{out} = 1.25 * (1 + R_{36}/R_{35}) \quad (1)$$

由公式 (1) 推出, 为输出 5 V 电压, R36 和 R35 的阻值分别为 10 K 和 3K3。

2.2 USB-UART 模块

USB-UART 模块建立 PC 端与主控芯片的数据通道。协议转换芯片采用 CH340G-USB 转串口芯片, 该芯片外围电路简单, 性能稳定, 资料丰富。电路中将 Rx 与 Tx 引脚接入 FPGA 芯片进行数据交换, D+ 与 D- 接入 USB 接口数据引脚与 PC 端进行数据交换。该芯片工作时需外接 12 M 晶振, 同时为保证系统时钟的同步, 引出时钟输入引脚与 FPGA 芯片相连。USB-UART 模块如图 4 所示。

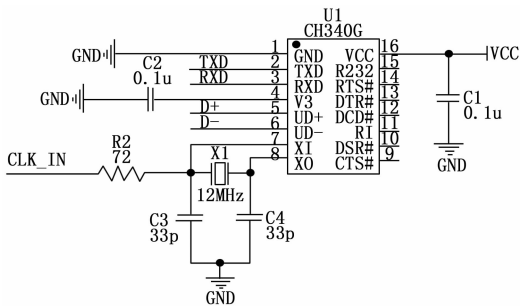


图 4 USB-UART 模块

2.3 FPGA 主控模块及 GPIB 接口模块

FPGA 主控芯片的选择参考市场两大主流 FPGA 芯片厂商, 再结合本项目的设计要求, 最终采用 Xilinx 的 Spartan-6 系列的 XC6SLX4 芯片, 该芯片的引脚数和容量能满足设计需求, 且网络资源充足, 易上手。

主控芯片的差分引脚主要与 GPIB 接口的 16 根信号线和 UART 模块的数据线连接, 以实现接口的控制和数据传输。外围电路还包括 JTAG 下载电路、SDRAM 外部存储模块以及时钟和复位电路。JTAG 下载调试电路使用 XCF01SVO20C-FPGA 配置芯片, 系统外部时钟采用 50 M 晶振。此外, 使用 K4S511632B-TC75 存储芯片设计 SDRAM 外部存储模块, 防止因 FPGA 芯片内部缓存不足造成数据丢失的现象。主控模块还外接了 LED 小灯等外围电路, 用来作为数据读写的指示灯。

传统的 GPIB 接口电路模块一般由专用 GPIB 接口芯片和总线收发器组成。本方案使用 FPGA 芯片模拟 GPIB 接口功能, 因此无需 GPIB 接口芯片和总线收发器。但 GPIB 接口工作电压是 5 V, 而 FPGA 芯片的最高电压为 3.3 V, 故在 FPGA 芯片与 GPIB 接口之间添加 MOS-N 模块实现电平转换。

3 硬件控制程序设计

硬件控制程序是设计的核心与难点。在设计之前, 需要对 IEEE488 总线协议和 GPIB 接口芯片的内部结构做深入研究。

3.1 IEEE488 总线介绍

IEEE488 总线^[5-6]自提出以来, 一直被广泛应用于测试、控制等领域。它由数据线、三根握手线、五根接口管理线以及地线组成。三根握手线配合工作, 通知总线何时传输数据, 何时接收数据, 确保数据传输准确无误。接口管理线 ATN 控制总线上传输的是数据还是指令, EOI 提示总线何时传输结束等, IFC 设置所有的设备于静止状态, REN 使能设备进行远程编程, SRQ 提醒控制器是否需要服务。在工作时, 测试系统存在 3 种角色, 即控者负责发送各种命令, 讲者发送数据, 听者接收数据。

典型的 GPIB 专用接口芯片 NAT9914 的内部电路主要分为两部分。一部分面向 GPIB 总线, 主要包括 GPIB 接口的十种功能以及各种缓冲器和译码器。另一部分面向微处理器, 通过编写驱动程序让微处理器访问 25 个可寻址寄存器, 其主要类型有数据类寄存器、中断类寄存器、地址类寄存器。NAT9914 的内部结构^[7-8]如图 5 所示。

GPIB 接口共定义了 10 种接口功能, 分别是源方和受方挂钩功能、讲者、听者、控者、器件清除功能、服务请求功能、并行轮询功能、远控/本控功能、器件触发功能。其中前五种是基本接口功能, 后 5 种是辅助接口功能。基本功能是所有带 GPIB 接口的设备必须具备的, 保证测试系统中各仪器之间能相互通信, 辅助功能让测试系统更便捷、更智能。

3.2 GPIB 接口功能的程序设计与仿真

该方案的硬件程序使用 Verilog 硬件描述语言进行开发, 软件采用 Xilinx 的 ISE。整体设计采用自顶向下的设计思想, 将整个系统拆分成若干个子功能模块, 其中包括串

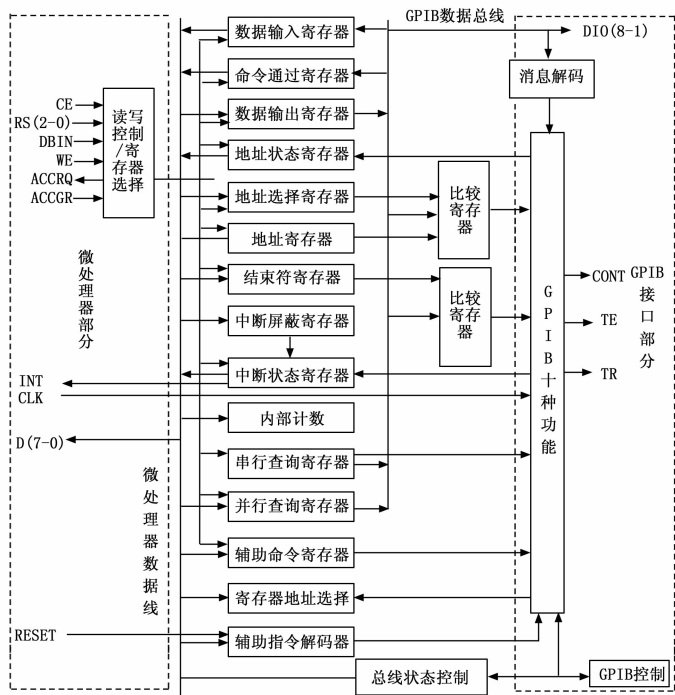


图 5 NAT9914 内部结构

口数据收发模块、数据读写控制模块、中断模块以及 GPIB 接口功能模块, 每个子功能模块又有不同的状态, 同一时刻只允许存在同一种状态, 不同的状态又通过相互联系的小状态机进行切换, 且状态的改变与之前设备所处的状态关联, 这样就构成了一个相互排斥又相互联系的状态机。通过各模块的接口功能状态机的联系, 最总形成一个顶层 GPIB 接口功能状态机。

GPIB 接口功能的实现依赖于各个功能模块的实现, 由于篇幅有限, 该章节主要对 GPIB 接口的讲者 (T)、听者 (L) 功能、FIFO 模块的设计进行详细描述, 并利用 Modelsim 软件进行仿真实验。

3.2.1 讲功能 (T) 设计

讲功能^[9-10]的主要作用是将用户发送的程控指令或者数据通过总线发送到测试系统中的仪器中, 或者将仪器自身的工作状态以及参数等信息通过数据总线传输到 PC 端。只有当系统控者寻址讲者或者设备通过 ton 指令将自身设置为讲者时, 才赋予这种功能。同一时刻, 一个测试系统中只存在一位讲者。

讲功能的状态分为 TIDS (讲者空闲态)、TADS (讲者被寻址态)、TACS (讲者活动态)、SPAS (串行活动态)、SPIS (串行空闲态)、SPMS (串行模式) 共 6 种, 按照协议功能又分为讲者数据传输和串行轮询两组事件, 分别如图 6 (a) 和图 6 (b) 所示。讲者状态图如图 5 所示。

事件 (a) 中, 在设备上电后, 讲功能同时进入 TIDS 和 SPIS 两种状态, 此时不具备数据传输能力。当设备被寻址为讲者 (即 MTA=1) 或设备通过 ton 指令设置自身为讲

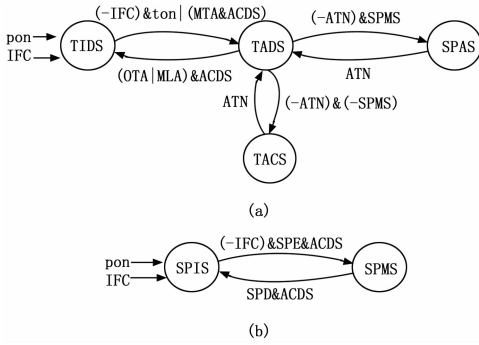


图 6 讲功能状态图

者, 且 ACDS 为活动态时, 讲功能进入被寻址态, 此时讲者准备好发送数据。ATN 信号线决定系统传输的数据类型, 当 ATN 为假时, 表示 GPIB 总线上传输的是数据, 当 SPMS 是活动态时, 设备进入 SPAS 状态; 当 SPMS 为假时, 讲者进入 TACS 状态, 此时配合 SH 功能将数据传到 GPIB 数据总线。当设备处于 SPAS 或 TACS 状态时, 若 ATN 为真, 说明系统控者发送命令, 讲功能返回寻址态。

事件 (b) 中, 若 SPE 为真, 即允许串行查询, 听者进入串行轮询模式状态, 并参与串行轮询。当系统活动控者发送 SPD 命令, 即禁用串行轮询。听者进入串行轮询空闲状态。

讲功能的 Verilog 设计包括了图 5 中两组状态, 每种状态的转换都通过分析当前的状态机来转换, 结合各状态之间相互排斥又相互联系的特征, TIDS 和 SPIS 采用并行语句执行。(b) 状态的具体实现的核心部分代码如下所示:

```

case (current_state_2)
T_STATE_2_SPIS :
if (event2_1)
current_state_2 <= T_STATE_2_SPMS;
T_STATE_2_SPMS :
if (event2_3)
current_state_2 <= T_STATE_2_SPIS;
else if (event2_2)
current_state_2 <= T_STATE_2_SPIS;
default :
current_state_2 <= T_STATE_2_SPIS;
endcase
assign event2_1 = SPE 1'b1 & ACDS 1'b1;
assign event2_2 = SPD 1'b1 & ACDS 1'b1;
assign event2_3 = IFC 1'b1;
    
```

通过编写 Testbench 仿真文件, 使用 Modelsim 仿真后波形如图 7 所示。根据分析可以看出, 讲功能的每个状态严格按照状态图进行变换, 且同一时刻只存在一种活动态。

3.2.2 听功能 (L) 设计

听功能^[11]的主要作用是接收仪器发送过来的状态信息或者主控设备的指令。同一时刻, 一个测试系统中可以存在多位听者。

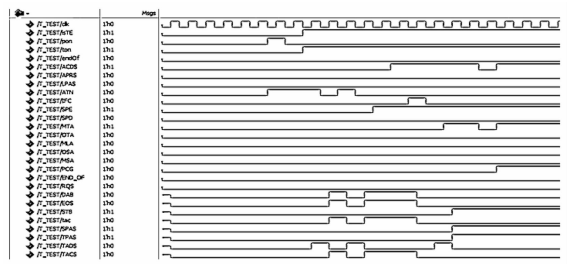


图 7 讲功能仿真波形

听功能的状态分为 LIDS (听者空闲态)、LADS (听者被寻址态)、LACS (听者活动态)、LPIS (听者主空闲态)、LPAS (听者主被寻址态), 其中后两种为扩展听者 (LE) 状态。听者状态如图 8 所示。

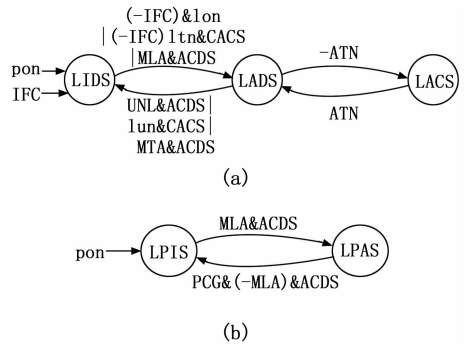


图 8 听功能及扩展听功能状态图

状态图 (a) 是听功能的状态图, 当设备上电后, 听功能同时进入 LIDS 和 LPIS 两种状态。当设备被寻址为听者 (MLA=1) 或者设备通过 lon 指令设置自身为听者, 且 ACDS 为活动态时, 听功能进入被寻址态, 此时听者准备好接收数据。若 ATN 为假, 表示 GPIB 总线上传输数据, 此时听者变为 LACS 状态, 并且配合 AH 功能从数据总线上接收数据。若 ATN 为真, 表示 GPIB 总线传输命令, 听者由活动态变为被寻址态。当控者发出 unl 命令, 宣布测试系统内所有的设备取消听者状态, 且 ACDS 为真时, 听者进入空闲态。

状态图 (b) 是扩展听功能的状态图, 在 LPIS 状态下, 听功能可以识别它的主地址, 但无法响应它的副地址。当听功能被寻址且处于接收数据状态时, 听功能进入被寻址主状态, 此时听功能可以识别并响应它的副地址。

听功能的 Modelsim 仿真波形如图 9 所示。

3.2.3 FIFO 模块设计

为提高系统传输数据的稳定性, 避免因控制端和数据端传输速率差异造成的数据丢失现象, 设计 FIFO 模块作为数据缓冲区。

FIFO 模块连接读写控制端和数据总线端。读写控制部分由 Clk、Reset_Fifo、Read、Write、Buffer_full 组成, data_in [7: 0] 和 data_out [7: 0] 连接数据总线。当转换器身份为讲者时, 从 FIFO 中取数据放入总线, 若 FIFO

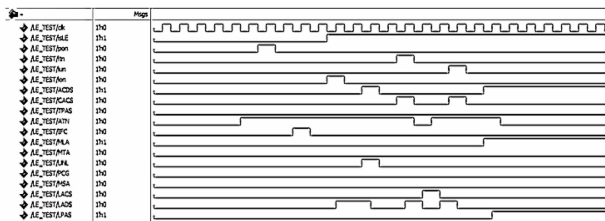


图 9 听功能仿真波形

数据为空, 则读写控制端写入准备发送的数据。当转换器身份为听者时, 将总线上的数据存入 FIFO, 若 FIFO 存在数据, 则通知读写控制端读取数据。FIFO 的 RTL 视图如图 10 (a) 所示。Modelsim 仿真波形如图 10 (b) 所示。

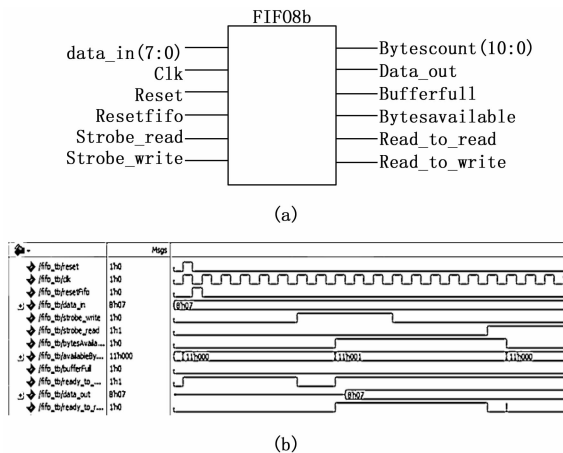


图 10 FIFO RTL 视图与仿真波形

3.3 内部寄存器设计

NAT9914 芯片的工作主要通过配置内部寄存器来完成。设计中通过读写控制模块对各个寄存器进行配置, 从而控制接口的数据传输。

设计的寄存器主要有地址寄存器、 GPIB 总线寄存器、 GPIB 状态寄存器、接口功能寄存器、 GPIB 控者寄存器、读写控制寄存器, 中断控制寄存器。其中, 中断寄存器 ISR0 与听、讲功能相关, 其结构如表 1 所示。

表 1 ISR0 寄存器

	7	6	5	4	3	2	1	0
ISR0	INT0	INT1	BI	BO	END	SPAS	RLC	MAC

ISR0 可读寄存器的 BI、BO、END 位分别代表数据接收位、数据发送位、数据传输结束位。当设备为系统活动控者或者讲者时, BO 位置 1, 此时可以向系统内其他设备发送命令或数据; 当设备为听者时, 当数据接收寄存器接收到数据时, BI 位置 1; 当接收到结束符 0x0A 时, END 位置 1, 结束本次数据传输。

3.4 GPIB 数据传输

数据传输是硬件控制程序的核心部分, 也是该协议转换器要实现的最主要功能。

在各个子功能模块完成的基础上, 构建从 UART 模块到 GPIB 接口模块的数据通道。数据的传输主要依靠设备的听、讲功能以及读写控制模块, 还有 FIFO 存储模块。听讲功能控制数据传输的方向, FIFO 模块控制数据的存取。读写控制模块直接配置状态寄存器, 进而配置 GPIB 的接口功能, 并配合 FIFO 模块完成数据的存取。在程序中设计了 GPIB_Read 和 GPIB_Write 两个资源文件, 用来实现 GPIB 的读写操作。在 GPIB_Read 模块中, 输入输出端口名分为 GPIB 接口、外部接口、FIFO 模块 3 部分。GPIB 接口部分用来判断当时是否处于听活跃状态, 外部接口模块负责产生数据中断等信号, FIFO 模块则提示是否存在数据以及数据的存取。GPIB_Write 模块的结构与 GPIB_Read 模块基本类似。读写的逻辑结构如图 11 所示。

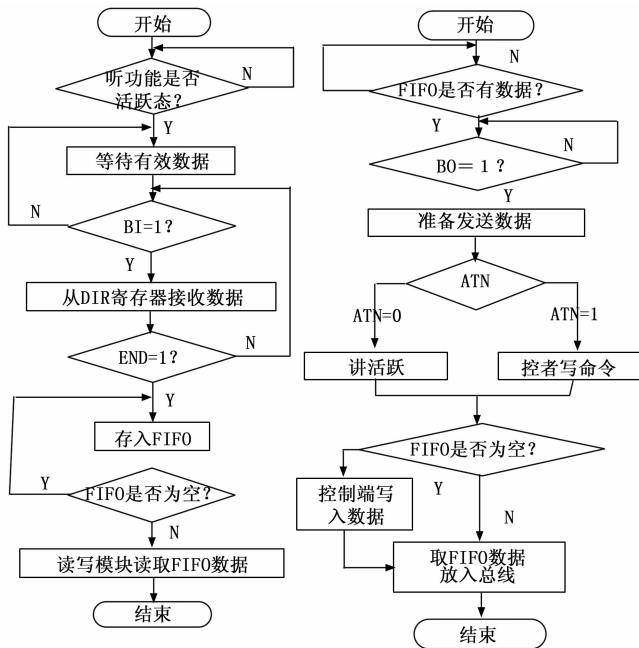


图 11 GPIB 读/写流程图

4 软件驱动程序设计

4.1 VISA 简介

VISA (virtual instrument software architecture), 即虚拟仪器软件结构^[12-14]。VISA 是一种通用的 I/O 标准, 提供了一套完整的、易操作的函数库, 存在于计算机系统内, 通过软件层与仪器建立连接。

传统的仪器, 不同的接口需要开发不同的驱动程序来供用户使用, 这将耗费大量的人力物力来开发与维护。VISA 独立于接口、语言和操作系统, 采用统一的操作函数, 无论是串口、GPIB、LAN 接口, 都使用 VISA 函数来操作, 体现了 VISA 的硬件无关性。统一的核心函数与其它无法合并的、与仪器类型相关的功能函数一起构成了自底向上的 I/O 接口模型, 实现不同类型的仪器的互操作性与兼容性, 这也是 VISA 的最大优点之一。

VISA 的结构共分为五层, 资源管理层主要主要负责资源的寻址、创建与删除等功能的实现。I/O 资源层提供了底层 I/O 函数, 对硬件接口进行读写等操作, 该资源与传统的 I/O 函数相似。仪器资源层主要针对不同仪器的特定功能。用户资源层是可变层, 根据用户自己定义资源。应用程序层提供用户与仪器交互的窗口。

4.2 VISA 驱动程序开发

本设计采用 NI 公司提供的 VISA 库, 在开发之前需安装 NI-VISA。驱动程序的一般设计流程如图 12 所示。

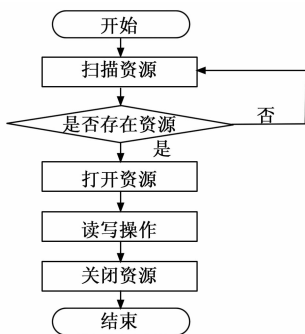


图 12 驱动开发流程

驱动程序具体实现流程^[15-18]如下:

1) 扫描资源: 首先调用 viOpenDefaultRM () 函数来扫描系统中的接口和设备资源, 此后才能调用其他 VISA 函数。当用户需要查找特定的设备时, 可以调用 viFindRsrc () 函数。

2) 打开资源: 通过 viOpen () 函数建立仪器之间的会话通道, 其参数根据接口类型确定。如建立与主地址为 2 的 GPIB 接口的会话, 则参数设置为 GPIB0.: 2.: INSTR。

3) 读写操作: 公用的 VISA 底层 I/O 函数有 viRead ()、viWrite () 等; GPIB 接口特殊 I/O 函数有的 viReadSTB ()、viClear () 等, 每个函数实现一种接口功能。本设计调用公用 I/O 函数与建立会话通道的接口进行通信。

4) 关闭资源: 调用 viClose () 函数关闭已建立的会话通道, 并清除资源。

在 VC++6.0 环境下测试 VISA 驱动程序, 将 VISA.H 和 VISATYPE.H 文件拷贝到工程中, 使用 C 语言调用相关库函数。核心代码部分如下所示:

```

int main(void){
    ViSession defaultRM;
    ..... //定义变量
    status=viOpenDefaultRM (&defaultRM);
    status=viOpen (defaultRM,"ASRL4.::INSTR",VI_NULL,VI_NULL,&instr);
    status = viSetAttribute (instr,VI_ATTR_TMO_VALUE,5000);
    ..... //参数配置
    status=viWrite (instr," * IDN? \n", (ViUInt32) strlen (" * IDN? \n"),&writeCount);
  
```

```

status = viRead (instr,buffer,100,&retCount);
if (status! <VI_SUCCESS){
    printf ("\nData read: %s\n",retCount,buffer);
}
status = viClose (instr);
status = viClose (defaultRM);
  
```

将协议转换器和 Tek1002 示波器连接, 运行该程序生成的 .EXE 可执行文件, 运行测试结果如图 13 所示。

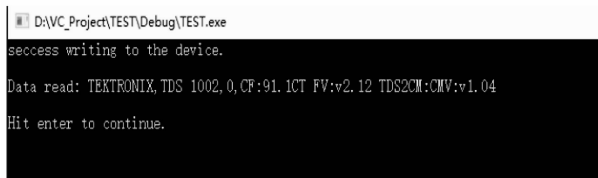


图 13 VC++下 VISA 测试结果

4.3 驱动应用层开发

驱动应用层开发平台为 Visual Studio 2017, 采用 C# 面向对象高级程序设计语言, 它运行在 .NET Framework 框架上。 .NET Framework 包括 3 个应用程序模型, 设计采用 Windows Forms 模型, 它属于 .NET Framework 中的 GUI 类库, 该模型采用控件式编程, 开发难度较低。

NI-VISA 提供了基于 .NET 的 VISA 函数库, 在项目中添加 NI 的 NationalInstruments.Visa 和 Ivi.Visa 函数库可直接调用 VISA 函数。

应用层开发主要分两部分, 一部分为资源搜索界面, 另一部分为基本读写操作界面。资源搜索界面通过 ResourceManager.Find (" (ASRL | GPIB | TCPIP | USB"? * INSTR)) 函数来搜索系统中所有类型的设备, 并将值返回 ListBox 列表框中。选中资源, 点击 OK 按钮时, 执行 ResourceManager.Open 函数与设备建立连接。操作界面主要由 Button 和 TextBox 两大控件组成, 读写等按钮执行基本 IO 操作, 在控件中调用 RawIO.BeginWrite () 和 RawIO.BeginRead () 函数, 当点击事件发生时, 则执行读写函数, 并将返回值发送到 TextBox 文本框中, 此外还添加了 Trigger 和 Read STB 等功能型按钮, 通过调用接口功能函数对设备进行操控。

最终设计的用户操作界面如图 14 所示。

5 SCPI 指令测试

5.1 SCPI 简介

可编程仪器标准命令, 简称 SCPI^[19-20]。它是一种控制仪器的语言规范, 规定了一种标准的语言结构来控制仪器, 采用 ASCII 字符串的形式与设备之间进行数据的交互。

早期的仪器设备, 由于没有统一的程控命令标准, 不同的仪器在软件上很难兼容。SCPI 的出现, 给整个行业提供了一套可参考的标准, 完美解决了不同仪器的兼容性问题。且 SCPI 采用助剂符的方式, 把复杂的指令缩写成简洁且容易记住的指令, 便于用户使用。



图 14 用户程序界面

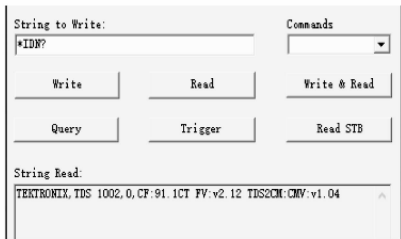
5.2 SCPI 命令测试

测试采用对比实验, 实验设备使用 Tek1002 示波器。通过对基本指令以及示波器的数据采集等指令的测试, 评估其性能。

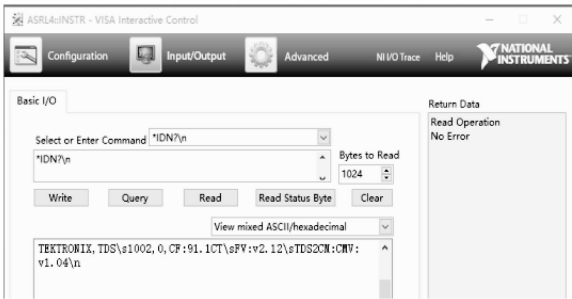
5.2.1 公用指令测试

SCPI 公用指令共 39 条, 是所有带 GPIB 接口的设备必须具备的。

公用指令测试使用自主研发的 GPIB 上位机和转换器与 NI 控制面板和 NI-GPIB-USB 作比较, 都发送 *IDN? 指令, 查看其返回结果是否一致。测试结果如图 15 所示。其中, 图 (a) 为自主研发设备返回结果, 图 (b) 为 NI 设备返回结果。



(a)



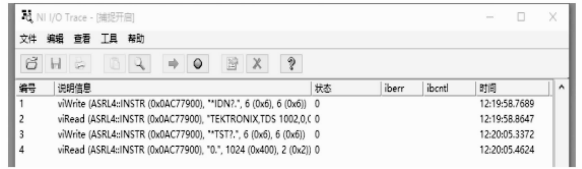
(b)

图 15 公用命令测试结果

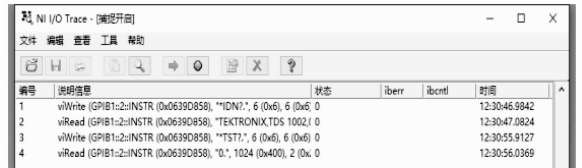
5.2.2 性能测试

性能测试软件采用 NI 公司的 I/O Trace, 该软件能捕获各函数的响应时间, 通过在同一软件下对比自研 GPIB 转换器和 NI-GPIB-USB 转换器来评估该设计的设备性能。测

试结果如图 16 所示。图 (a) 自主研发设备响应时间, 图 (b) 为 NI 设备响应时间。



(a)



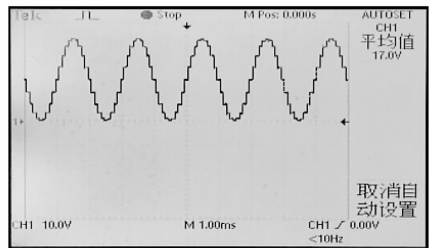
(b)

图 16 指令响应时间

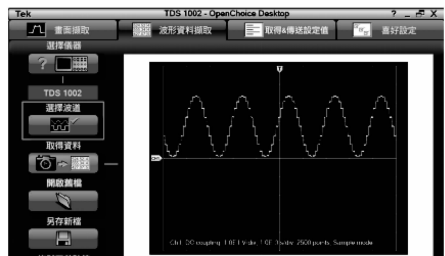
从图 16 中可以看出, 自研设备在响应 *IDN? 和 *TST? 两个指令的时间分别为 958 ms 和 1 252 ms, NI 的响应时间为 982 ms 和 1 242 ms。经对比, 自研 GPIB 转换器性能优异, 完全能满足实际应用需求。

5.3 示波器数据采集测试

GPIB 接口的重要功能是采集仪器的数据。数据采集实验通过 STM32F103 开发板的 DA 模块输出波形, 使用 Tek1002 示波器测量波形, 将该设计的 GPIB 卡连接示波器, 通过 Tek 公司的 OpenChoiceDesktop 上位机软件连接示波器, 首先发送: SELECT: CH1 选择示波器通道 1, 然后发送 CH1: PROBE? 等查询指令查询示波器的各种参数, 最后发送: CURVE? 返回曲线数据, 并反馈到 Tek 公司的 OpenChoiceDesktop 上位机软件上。测试结果如图 17 所示。



(a)



(b)

图 17 数据采集测试结果

6 结束语

本文设计了一款以 FPGA 芯片为主控, 通过硬件描述语言实现 IEEE488 接口的所有功能的协议转换器。文中介绍了转换器硬件设计、控制程序设计, 还详细介绍了基于 VISA 的驱动程序设计流程。该设计为解决当前市场 GPIB 专用芯片停产、转换器价格昂贵等问题提供了一套完整的替换方案。经测试实验表明, 该转换器性能稳定、体积小, 完全满足日常测试所需, 为从事相关方面研究的开发人员提供了可行性的参考案例。

参考文献:

- [1] 陈长龄, 田书林, 师奕兵. 自动测试及接口技术 [M]. 机械工业出版社, 2005.
- [2] 孙晓云. 接口与通信技术原理与应用 [M]. 北京: 中国电力出版社, 2007.
- [3] 陈 敏. 智能仪器接口的设计与实现 [D]. 武汉: 华中师范大学, 2012.
- [4] 靳 取, 方 晖, 柏业超. 基于 GPIB 的数字电路自动测试系统研究 [J]. 南京大学学报 (自然科学), 2016, 52 (6): 1133 - 1138.
- [5] 朱兴邦, 李 兵. IEEE 488 与 SCPI 解释 [J]. 仪器仪表标准化与计量, 2007 (1): 15 - 16, 26.
- [6] 高玉栋. 基于 FPGA 的 GPIB 接口 IP 核的研究与设计 [D]. 镇江: 江苏大学, 2007.
- [7] National Instruments Corporation. NAT9914 Reference Manual [Z]. America, National Instruments Corporation, 1995.
- [8] 王术群, 师奕兵, 田书林. GPIB 接口的 FPGA 实现 [J]. 电测与仪表, 2009, 46 (1): 69 - 72, 76.
- [2] 石章松, 龚文斌, 吴中红, 等. 基于天基信息的海上远程精确打击技术现状及发展 [J]. 海军工程大学学报 (综合版), 2020, 17 (3): 26 - 30.
- [3] 洪 俊, 张辉武, 沈振华, 等. 远程精确打击对卫星信息时效的需求研究 [J]. 指挥控制与仿真, 2017, 39 (5): 9 - 11.
- [4] 王勇平. 空间信息支援作战 [M]. 北京: 国防大学出版社, 2014.
- [5] 孙亚楠, 钟选明, 王俐云, 等. 天基信息支持远程精确打击作战及其体系建设的需求 [J]. 战术导弹技术, 2018 (5): 13 - 18.
- [6] 熊 威, 车梦虎. 海洋大气波导环境下反舰导弹作战效能评估 [J]. 计算机测量与控制, 2019, 27 (11): 252 - 254.
- [7] 洪 俊, 张辉武, 沈振华. 远程精确打击对卫星信息时效的需求研究 [J]. 指挥控制与仿真, 2017, 39 (5): 9 - 11.
- [8] 彭 耿, 李 涛, 刘 磊. 海上远程精确打击对卫星侦察信息的时效需求研究 [J]. 中国电子科学研究院学报, 2020, 7: 685 - 690.
- [9] 汪新刚. 反舰导弹目标捕获概率模型分析 [J]. 战术导弹控制技术, 2006, 7 (3): 104 - 106.
- [10] 沈培志, 张邦钰, 聂奇刚. 反舰导弹落点区离散化的目标捕捉概率模型 [J]. 战术导弹技术, 2015, 2: 90 - 95.
- [11] 侯学隆, 姜青山. 海上机动目标均匀概略航向下的散布规律研究 [J]. 火力与指挥控制, 2018, 43 (5): 49 - 53.

子设计应用, 2004 (10): 77 - 79.

- [9] 陈立超. 基于 FPGA 的 GPIB 控制器的 IP 核设计 [D]. 成都: 西南交通大学, 2008.
- [10] 朱怀宇. 基于 IEEE-488.2 协议的 FPGA 设计 [D]. 成都: 电子科技大学, 2003.
- [11] 王澜锦. GPIB-ENET 和 GPIB-UART 转换的研究与设计 [D]. 天津: 天津工业大学, 2008.
- [12] 王学伟, 童永霞, 孟 辉. VISA 函数库 I/O 资源层 RS-232 接口的扩充与实现 [J]. 电测与仪表, 2009, 46 (1): 69 - 72, 76.
- [13] 吴 敏, 赵 波. 虚拟仪器技术及仪器控制实例 [J]. 电子质量, 2012 (8): 10 - 12.
- [14] 祖俊婕. 高速高精度数据转换器测试验证系统设计与实现 [D]. 南京: 东南大学, 2017.
- [15] 尹洪涛, 付 平, 郭论平. LXI-GPIB 控制器及 VISA 库设计 [J]. 计算机测量与控制, 2013, 21 (5): 1263 - 1265.
- [16] 王 旭. 基于 FPGA 和 DSP 的通用多功能 PXI 板卡设计与实现 [D]. 北京: 中国科学院大学 (工程管理与信息技术学院), 2016.
- [17] 许倩文. LXI-VXI 零槽控制器 VISA 库的设计与实现 [D]. 武汉: 华中科技大学, 2016.
- [18] 童永霞. VISA 标准事件机制研究及 DLL 扩展的实现 [D]. 北京: 北京化工大学, 2009.
- [19] 秦 凡, 韦 高. 基于 VISA 库及 SCPI 命令的仪器程控测量 [J]. 现代电子技术, 2011, 34 (11): 118 - 120, 123.
- [20] 奚全生. 可编程仪器标准命令 (SCPI) 介绍 [J]. 计算机自动测量与控制, 1994 (1): 22 - 31.
- [12] 张 武, 刘勇志. 潜射反舰导弹目标指示信息延迟对捕捉概率的影响 [J]. 四川兵工学报, 2008, 29 (4): 9 - 11.
- [13] 李 建, 范 辉, 刘 铁. 目标指示信息延迟对反舰导弹末制导雷达参数装订的影响分析 [J]. 弹箭与制导学报, 2013, 33 (1): 176 - 178.
- [14] 程晓雪. 对海中远程精确打击体系 [J]. 指挥信息系统与技术, 2015, 6 (1): 16 - 21.
- [15] 卢代军, 夏学知, 张子鹤, 等. 目标信息的时效性分析 [J]. 火力与指挥控制, 2007, 32 (1): 38 - 41.
- [16] 高 源. 基于空间信息系统的时间敏感目标打击 [J]. 指挥信息系统与技术, 2014, 5 (4): 10 - 13.
- [17] 郭 亮, 吴美希, 王 峰, 等. 数据中心算力评估: 现状与机遇 [J]. 信息通信技术与政策, 2021 (2): 79 - 86.
- [18] NVIDIA. 适用于服务器的 TESLA 数据中心 GPU [EB/OL]. (2021 - 11 - 01) [2021 - 11 - 01]. <https://www.nvidia.cn/data-center/tesla/>.
- [19] 张紫龙, 黄 晨, 施自胜, 等. 基于干涉仪测向的电子侦察卫星单星定位精度分析 [J]. 火力与指挥控制, 2016, 41 (1): 67 - 72.
- [20] 蒲文其. 三星时差频差无源定位与测速算法研究 [D]. 长沙: 国防科学技术大学, 2016.
- [21] 汪 松, 肖 倩, 杨德运. 遥感卫星几何定位精度影响因素分析 [J]. 信息通信, 2019 (9): 4 - 5.