

# 基于 FPGA 的远程直流电机控制系统研究

毕英建, 邱亚峰, 柳志鹏

(南京理工大学 机械工程学院, 南京 210018)

**摘要:** 针对外场试验中运动控制系统实时性与准确性问题, 提出了以 FPGA 为核心的以太网传输电机运动控制系统方案; 通过上位机完成对多个直流电机方向、速度、位置等的远程控制, 实现了软硬件相结合; 采用 YT8511/H 芯片、RGMII 接口和 Altera 官方 IP 核搭建了 FPGA 的高速以太网通信接口, 运用三段式状态机实现了 UDP 通信协议, 通过增量式光电编码器完成电机位置反馈, 以 Verilog 硬件描述语言为基础, 编写了电机驱动程序和编码器信号处理程序, 对系统的通信、输出脉冲以及编码器信号处理进行了实验和仿真; 结果表明, 系统满足远距离控制需求, 定位精度可达  $0.09^\circ$ , 有较好的移植性。

**关键词:** 以太网通信接口; UDP/IP 协议; FPGA; 直流电机; 光电编码器; 闭环控制系统

## Research on Remote DC Motor Control System Based on FPGA

BI Yingjian, QIU Yafeng, LIU Zhipeng

(School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210018, China)

**Abstract:** In order to solve the real-time and accuracy problem of motion control systems in outfield test, a motor motion control system based on Ethernet transmission by field programmable gate array (FPGA) was proposed. The upper computer is used to remotely control the multiple DC motor directions, speeds, positions, achieving the combination of software and hardware. It adopts the YT8511/H chip, RGMII interface, and Altera official IP core to build the high-speed Ethernet communication interface for FPGA. Besides, the UDP communication protocol was realized by using the three-stage machine, and the motor position feedback was completed through the incremental photoelectric encoder. Based on Verilog hardware description language, the motor driver program and encoder signal processing program were written, and the system communication, output pulse and encoder signal processing were tested and simulated. The results indicate that the system meets the requirements of remote control, with a positioning accuracy of  $0.09^\circ$ , and it has a good portability.

**Keywords:** ethernet communication interface; UDP/IP protocol; FPGA; DC motor; optical encoder; closed-loop control system

## 0 引言

随着嵌入式系统和微电子技术的不断发展, 现场可编程逻辑器件 (FPGA, field programmable gate array) 因其高速并行处理能力和可重复编程的优点在控制领域得到了越来越多的应用。传统控制方式如以 ARM、DSP 等为核心的微控制器因其硬件复杂、可移植性差、运算能力不足等缺陷逐渐难以适应远距离外场试验的电机运动控制需求<sup>[1]</sup>。相较于单片机软件控制方式出现的程序跑飞、软件延时等问题, FPGA 通过硬件电路实现控制, 增强了系统的稳定性与准确性。在电路设计上, FPGA 集成了通信、数据缓存和专用 IP 核等功能, 可有效简化设计, 缩短开发人员的设计时间<sup>[2-3]</sup>。

在当前工业自动化应用背景下, 单一的电机控制略显乏力, 更多的是需要多个多种类型电机的协同工作, 这就

对控制系统的复杂度和稳定性有一定的要求。另外远距离基于网络传输控制也越来越多地出现在电机控制系统中出现, 非现场控制场合在工程项目中逐渐增加。搭建一套远程复合电机控制系统需要一个很好的处理平台, 在满足复杂算法计算的前提下保证信息传递的准确和快速响应。以高速并行处理为优点的 FPGA 控制芯片, 不仅有丰富的接口、逻辑资源, 而且其基于硬件编程的特点方便了电路设计, 因此将 FPGA 作为远程复合电机控制系统的核心具有良好的优势和重要意义。

本文主要以外场测试用移动靶标为例, 设计了一种基于 FPGA 的远距离直流电机运动控制系统。通过上位机软件开发和 FPGA 硬件程序编写, 来实现以太网控制信号传输, 借助增量式光电编码器完成对电机的闭环反馈, 达到远程控制电机速度、位置的要求。经仿真和测试, 该控制系统运行稳定性较好, 具有较高的控制精度和响应速度,

收稿日期: 2023-06-19; 修回日期: 2023-07-23。

基金项目: 国防预研基金(1171011485)。

作者简介: 毕英建(1998-), 男, 硕士研究生。

邱亚峰(1966-), 男, 博士, 副教授。

引用格式: 毕英建, 邱亚峰, 柳志鹏. 基于 FPGA 的远程直流电机控制系统研究[J]. 计算机测量与控制, 2024, 32(6): 118-124.

为远程电机控制系统的开发提供了一定参考。

## 1 系统总体结构

电机远程控制系统结构如图 1 所示。系统运作顺序如下: 基于 UDP 网络协议由上位机向 FPGA 发送 IP 连接请求, 建立连接后上位机发送控制命令。FPGA 通过 UDP 通信模块接收控制命令数据包, 并解析控制指令传递给电机控制模块。电机控制模块按照上层要求, 结合 PID 控制反馈, 输出直流电机转动方向和脉冲宽度调制信号 (PWM, pulse width modulation)。驱动器接收 FPGA 的控制指令驱动电机做相应运动。编码器信号处理模块通过读取光电编码器实时测量电机转动输出的 A、B 相信号, 计算电机当前速度和位置信息供 PID 闭环反馈, 并通过 UDP 通信模块将运动信息回传至上位机显示。

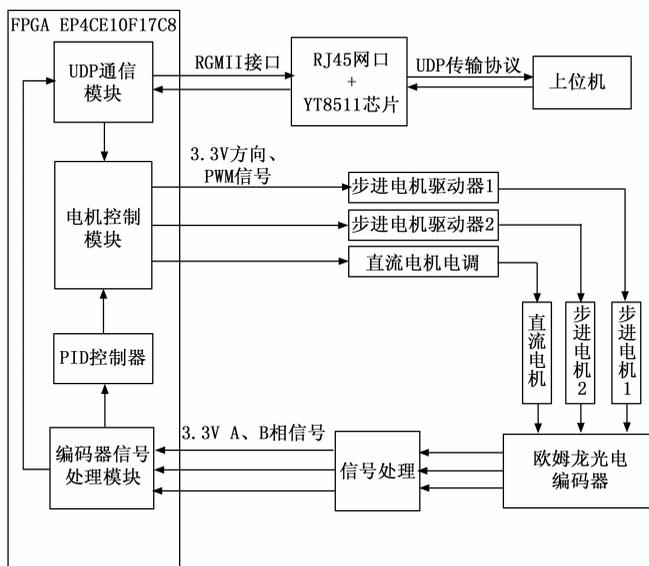


图 1 系统结构框图

为满足系统远距离控制的实时性, 减小数据传输速率带来控制上的误差, 系统采用 YT8511C/H 芯片和 RGMII 接口来搭建 FPGA 的高速以太网通道, 实现物理层传输协议。借用 Altera 官方 ADDRIO IP 核实现 RGMII 和 GMII 接口的相互转换, 以方便数据的传输和解读。使用三段式状态机硬件逻辑来完成对 UDP 通讯协议的封装和解析, 并在此基础上实现介质访问控制器 (MAC, media access control) 层数据传输的帧结构。

在电机控制上, 为适应多种电机融合使用的场合, 集成了步进电机、永磁无刷直流电机异步控制, 上位机控制命令简单且具备较丰富的扩展性。通过对编码器输出信号的上拉, 使得 FPGA 可直接读取相电平作倍频鉴相处理, 进而完成电机运动速度和位置计算。

## 2 各模块功能设计与实现

### 2.1 高速网络接口搭建

目前千兆以太网接口的设计方案通常采用“介质访问

控制器 MAC+以太网 PHY 芯片”来实现<sup>[3]</sup>。MAC 规定了数据帧的传递顺序, 包含物理地址、数据段和校验位等。端口物理层 (PHY, port physical layer) 芯片属于 OSI 模型的物理层接口, 以发送数据为例, 它接收 MAC 传来的数据并按照物理层编码规则将数据编码转换为模拟信号发送出去。MAC 与 PHY 之间遵循一定的接口规则, 常用的接口有 GMII 和 RGMII, 这两种接口都支持 1 000 Mbps 传输速率, 不同在 GMII 接口传输数据位宽为 8 位, RGMII 接口为 4 位<sup>[4-5]</sup>。网络接口的搭建包括 MAC 子层和物理层, 一般可以将 MAC 控制器模块和物理层 PHY 芯片集成在一起来实现网络传输, 也可以将两部分分开独立实现。为节省 FPGA 引脚资源, 本文采用 RGMII 接口作为 MAC 和 PHY 之间的通信接口, MAC 控制器由 FPGA 内部逻辑实现, 选用 PHY 芯片 YT8511C/H 来实现 FPGA 网络接口的物理层协议, 其结构如图 2 所示。

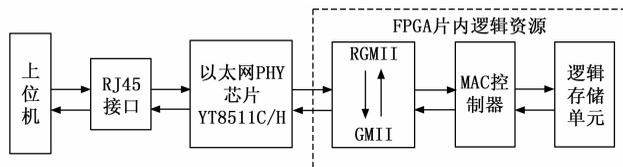


图 2 网络接口结构框图

YT8511C/H 芯片作为以太网物理层收发器, 提供半双工/全双工的 10、100、1 000 Mbps 以太网传输<sup>[6]</sup>, 包含实现物理层功能所需的所有有源电路, 以实现在 5 类无屏蔽双绞线电缆上传输和接收数据。其设计符合 RGMII 接口规范, 能够与标准以太网媒体访问控制器和交换机控制器兼容。

为方便处理传输的数据, FPGA 在接收 PHY 传递的 4 bit 数据时要先做 4 位转 8 位的操作 (发送时相反), 这样才方便后续模块对信息的解读。本文利用 Altera 官方 IP 核 ADDRIO 实现了 RGMII 与 GMII 接口的相互转换, 以接收数据为例给出 GMII 和 RGMII 接口信息如图 3 所示。

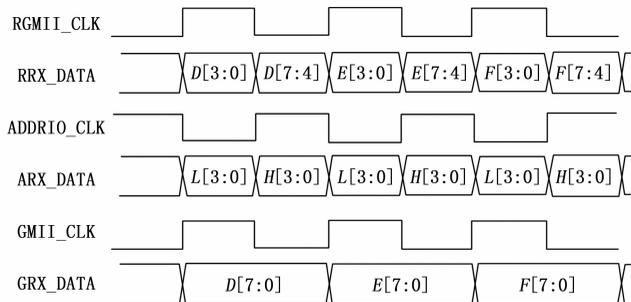


图 3 RGMII 到 GMII 接口时序

采用千兆传输速率时, RGMII 接口时钟为 125 MHz, 在时钟上升沿接收低四位下降沿接收高四位, 数据接收为双沿采样。因为 ADDRIO 的接口时序是在时钟下降沿接收低 4 位, 上升沿接收高 4 位, 再将这 8 位拼接后输出, 所以

其输入时钟需要按图 3 所示对 RGMII\_CLK 作反转, 以保证输出数据为正常顺序的 8 位数据。GMII 数据接口为单沿采样即在一个时钟周期内传输一个字节的信 息, 这样与网络传输最小信息量字节相同而且方便 MAC 控制器对以太网数据包的解读。

位于 OSI 七层协议中的数据链路层下层的 MAC, 它定义了数据帧在介质中传输的规则, 以太网数据必须按此规则排布才能在物理层传输<sup>[7-8]</sup>。文中 MAC 端即 FPGA 内部逻辑电路通过判断接收的初始数据是否为以太网帧头来决定一帧数据包的解读或丢弃。在发送用户数据时, MAC 端则按照标准以太网数据帧格式对数据段进行封装, 添加前导码、以太网帧头、MAC 地址等完成一帧数据包的打包后再向 GMII 接口发送数据, 经物理层编码传输给上位机。

### 2.2 UDP 通信协议实现

UDP 协议位于 OSI 参考模型中的传输层, 是一种无连接的传输协议<sup>[9]</sup>。因为通信双方不需要握手操作, 所以占用的系统资源少, 通信速率快。使 FPGA 具备 UDP 通信能力, 可以通过如下方案: FPGA + CPU 架构, CPU 解析 UDP 协议后将数据移交 FPGA 处理; FPGA + MAC + PHY 架构, PHY 作物理层传递, MAC 实现数据链路层功能, 用 FPGA 内部逻辑资源构建传输层协议<sup>[10-11]</sup>。由于使用 CPU 进行网络传输时会频繁触发中断, 传输延迟不固定, 而且在嵌入式设备上使用的 CPU 通常体积小功耗低, 处理高速以太网数据会消耗大量的计算资源, 因此本文使用 FPGA 通过三段式状态机实现对 UDP 协议的解析与封装, 其状态跳转如图 4、5 所示。

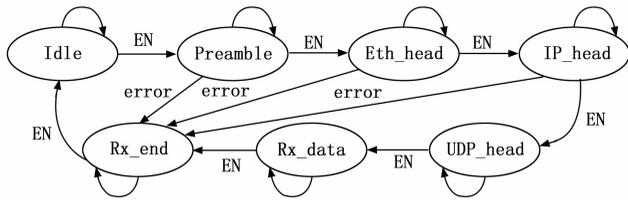


图 4 UDP 接收状态转移

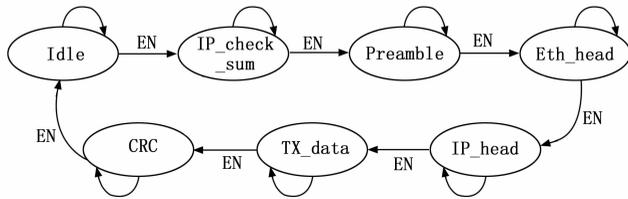


图 5 UDP 发送状态转移

UDP 通讯主要由 UDP 接收模块和 UDP 发送模块构成, 其状态跳转均遵循以太网数据报文格式, 如图 6 所示。一帧数据报包括前导码 (Preamble)、帧起始定界符 (SFD, start frame delimiter)、以太网帧头、IP 首部、UDP 首部、用户数据段和帧检验序列 (FCS, frame check sequence)。

IP 协议由 IP 首部 20 字节组成, UDP 协议由 UDP 首部 8 字节组成, 它们与用户数据段共同构成了 IP 数据报, 位于 MAC 帧中的数据段部分<sup>[12]</sup>。

前导码	帧起始定界符 SFD	以太网帧头	IP 首部	UDP 首部	用户数据	帧检验序列 FCS
-----	------------	-------	-------	--------	------	-----------

图 6 以太网数据报帧格式

UDP 接收模块完成了从初始准备接收数据到接收前导码和帧起始定界符等, 最后接收一帧数据包完成返回初始等待接收状态的动作循环。在正式接收用户数据之前, 如果判断接收信息与报文格式不符则跳转至结束状态, 表明该帧数据包出错, 等接收数据使能信号 rxdv 拉低即一帧数据包检测完后, 重新进入初始接收状态。当状态机跳转至接收用户数据时, 该模块按照时钟频率将指令信息传递给内部寄存器, 然后通过总线传到其他控制模块。

UDP 发送模块与接收模块的工作状态类似, 需要提前计算并存储 IP 首部校验和与循环冗余校验 (CRC, cyclic redundancy check), 然后根据报文顺序依次发送, 同样等待发送数据使能信号 txdv 拉低后开始下一帧数据包传输。因为以太网数据报文中数据段长度至少 46 字节, 去除固定的 IP 首部 20 字节和 UDP 首部 8 字节后, 要求用户数据至少 18 字节, 所以在发送数据时需要判断字数是否足够, 不足时作补 0 处理。

### 2.3 电机控制模块

无刷直流电机的控制依赖输出脉冲信号的占空比, 电调会根据不同占空比分配大小不同的正反电压到电机的线圈上来驱动转子作正反方向不同速度地旋转。步进电机的控制则依赖输出脉冲信号的个数, 与占空比无关。驱动器会根据脉冲信号的个数和方向信号的高低电平来驱动步进电机作不同方向的对应步数运动。由此可见直流电机控制实际是控制脉冲信号变化, 又称 PWM 调制<sup>[13]</sup>。本文主要以外场测试用移动靶标为例, 步进电机作位置闭环控制, 用来调节靶标的方位和俯仰角度; 无刷直流电机作速度闭环控制, 用于驱动靶标移动。

上位机发送的控制指令如图 7 所示。该数据段为固定 14 字节, 每字节对应一个字符的 8 位 ASCII 码。首字节为不同电机控制切换, 根据后四位的十六进制值可划分 15 种控制, 异步控制就是根据该字节的值分别对步进电机和无刷直流电机的控制信号进行读取和赋值; 第二字节为步进电机步进或连续运动方式判断, 用逻辑 0 和 1 区分; 第三、四字节表示两步步进电机的运动使能信号, 用逻辑 0、1 区分; 第五、六字节表示两步步进电机运动方向, 用逻辑 0、1 区分; 第七、八字节为步进电机速度控制, 值从 0 到 99, 可对应 100 个不同的速度值; 第九、十字节为步进距离控制, 可对应 100 个不同的步进距离; 第十一字节为直流电机加减速运动使能, 值为逻辑 0 或 1; 第十二、十三字节可表示 100 个不同直流电机脉宽控制信号标志; 第十四字节

为编码器位置计数归 0 控制。上位机发送数据段“10100003061040”为例, FPGA 实际接收的数据为 16 进制的“3130313030303033303631303430”, 通过读取第一字节的模式值为 1 可知该数据段为步进电机控制, 则将第 2 至第 10 字节的值赋值给对应的控制信号而之后的字节不作赋值, 这样 1 号步进电机会按照 0 模式作方向为 1 的第 3 速度运动, 若为步进模式则运动一个第 6 位置步长。

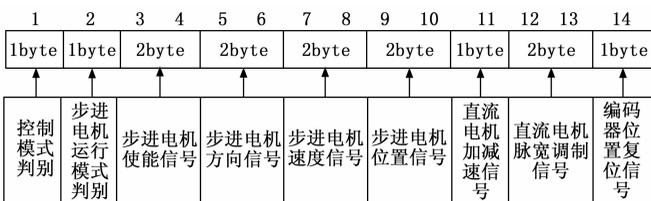


图 7 14 byte 用户数据段

因为步进电机 PWM 信号不用考虑占空比问题, 所以程序中固定以 50% 占空比的脉冲信号作为控制信号。脉冲信号频率与步进电机转速正相关, 其关系如下式:

$$f = v/\alpha = n \cdot v/60 \quad (1)$$

式中,  $f$  为脉冲频率 (Hz);  $n$  为步进电机取东西细分数;  $v$  为电机转速 (r/min)。

因 FPGA 系统时钟为 50 MHz, 所以脉冲信号的半周期计数器的值如下:

$$cnt = \frac{15}{n \cdot v} \times 10^8 \quad (2)$$

每当计数器计到  $cnt$  时对脉冲信号做一次反转就得到了速度  $v$  对应的频率为  $f$  的 PWM 信号。步进运动控制则是在此 PWM 信号的基础上添加步进距离对应的时间使能来生成一段而非持续的脉冲信号。按当前速度计算步进距离  $\beta$  对应的时间  $T$ , 使能信号在维持了  $T$  时间的高电平后拉低, PWM 信号则在使能信号为逻辑 1 时生成逻辑 0 时结束, 完成一段步进距离运动。

直流电机控制信号为持续性脉冲, 与占空比有关而与频率无关, 所以人为规定一个脉冲频率, 一般脉冲周期大于 2 ms。通过改变脉冲信号中高电平的持续时间及占空比, 进而控制电极间的电压来控制电机转子作不同速度转动。

### 2.4 反馈控制模块

步进电机位置闭环控制依靠编码器计数反馈, 将测量计算结果与设定步长比较后作二次输出脉冲调整。直流电机速度闭环控制由增量式 PID 算法根据反馈的速度误差做脉宽调整来实现。

传统 PID 算法虽然结构简单但当系统特性因外界影响而发生改变时则控制精度略显不足。基于增量式 PID 算法加入模糊控制, 可以有效改善系统的响应性能并拥有更好的稳定性和鲁棒性<sup>[14-16]</sup>。通过对传统 PID 算法做离散优化可得增量式 PID 公式:

$$u(k) = u(k-1) + K_1 e(k) + K_2 e(k-1) + K_3 e(k-2) \quad (3)$$

$u(k)$  为第  $k$  个采样周期输出的控制变量,  $e(k)$  为第  $k$  个采样周期输入的误差值,  $K_1 K_2 K_3$  为系统系数它们与系统的比例、积分和微分系数  $K_p K_i K_d$  有关。模糊 PID 控制则是在原有模型基础上基于速度误差和误差变化率输出比例、积分和微分系数的变化量作用到 PID 算法中来实时优化比例、积分和微分系数<sup>[17-18]</sup>。

根据模糊控制和增量式 PID 算法原理, 将控制模块划分为 PID 输出和模糊控制两大模块, 其系统如图 8 所示。PID 输出模块主要完成系统系数计算并赋值、速度误差计算和寄存以及系统输出量计算和寄存的任务, 模糊控制模块通过对输入的速度误差和误差变化率进行模糊化处理变为模糊值后, 根据存入 ROM 中的模糊控制规则进行模糊推理, 然后将推理结果解模糊化得到比例、积分和微分系数的增量作用到 PID 输出模块的计算中。

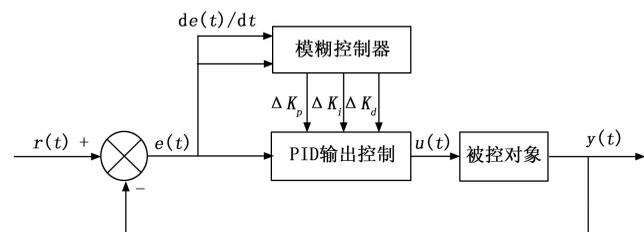


图 8 模糊 PID 控制框图

模糊控制采用 Mamdani 型模糊控制器, 根据系统需要将输入的变量划分为 NB、NM、NS、ZO、PS、PM、PB 共 7 个语言变量, 误差  $e$  的论域为  $[-3, 3]$ , 误差变化率  $ec$  的论域为  $[-12, 12]$ ,  $\Delta K_p$  的论域为  $[200, 1\ 000]$ ,  $\Delta K_i$  的论域为  $[100, 700]$ ,  $\Delta K_d$  的论域为  $[10, 100]$ , 隶属度函数设置为三角形。基于以下理论和实际经验建立模糊推理规则。

- 1) 当系统误差  $e$  较大, 无论  $e$  增大或减小,  $K_p$  取值尽可能大,  $K_d$  尽可能小, 同时为减小系统超调,  $K_i$  取值尽可能小。
- 2) 当  $e$  处于中间值, 为降低系统出现的超调量,  $K_p$  取值尽可能小。  $K_i K_d$  取适中值保证系统响应性能,  $K_d$  取值对系统影响较大。
- 3) 当  $e$  较小, 为保证系统稳态性能,  $K_p K_i$  取值应较大。同时为避免系统出现较大振荡, 当  $ec$  较小时,  $K_d$  取较大值。

采用“if  $a$  is  $A$  and  $b$  is  $B$  then  $c$  is  $C$ ”形式, 通过多次对比确立控制规则如表 1 所示。将设计好的规则表保存为 Quartus 可读的 .mif 文件并存储到 ROM 中。按模糊规则推理得到的模糊量采用重心法进行清晰化处理计算出清晰值如下式:

$$v = \frac{\sum_{j=1}^r v_j \mu_j(e, ec)}{\sum_{j=1}^r \mu_j(e, ec)} \quad (4)$$

表 1 模糊控制规则表

$K_p/K_i/K_d$		$e$						
		$PB$	$PM$	$PS$	$ZO$	$NS$	$NM$	$NB$
$ec$	$PB$	$NB/PB/PB$	$NM/PB/PB$	$NM/PB/ZO$	$NM/PM/ZO$	$NS/PS/ZO$	$NS/ZO/ZO$	$ZO/ZO/PS$
	$PM$	$NB/PB/PS$	$NM/PB/PS$	$NM/PM/ZO$	$NM/PM/NS$	$NS/PS/NS$	$ZO/ZO/NS$	$ZO/ZO/NM$
	$PS$	$NM/PM/PS$	$NS/PM/PS$	$NS/PS/ZO$	$NS/PS/NS$	$ZO/ZO/NS$	$PS/NS/NM$	$PS/NS/NB$
	$ZO$	$NM/PM/PM$	$NM/PS/PS$	$NS/PS/ZO$	$ZO/ZO/NS$	$PS/NS/NM$	$PS/NS/NM$	$PM/NM/NB$
	$NS$	$NM/PS/PM$	$NS/PS/PS$	$ZO/ZO/ZO$	$PS/NS/NS$	$PM/NS/NM$	$PM/NM/NB$	$PM/NM/NB$
	$NM$	$ZO/ZO/PM$	$ZO/ZO/NS$	$PS/NS/ZO$	$PM/NM/NS$	$PM/NM/NS$	$PB/NB/NB$	$PB/NB/NS$
	$NB$	$ZO/ZO/PB$	$PS/ZO/PB$	$PS/NM/ZO$	$PM/NB/ZO$	$PM/NB/ZO$	$PB/NB/PS$	$PB/NB/PS$

式中,  $\mu_j(e, ec)$  为隶属度函数值,  $r$  为模糊规则数目。将计算的 3 个控制系数的增量清晰值作用到 PID 控制器中, 对比例、积分、微分系数的调整方式如式 (5) 所示:

$$\begin{aligned} K_p &= K'_p + \Delta K_p \\ K_i &= K'_i + \Delta K_i \\ K_d &= K'_d + \Delta K_d \end{aligned} \quad (5)$$

### 2.5 编码器信号处理模块

增量式光电编码器 E6B2-CWZ3E 与电机轴相连, 将电机的转动角度位移量转化成电信号传递给 FPGA。为满足 FPGA 的 IO 引脚电平要求, 需将编码器输出的 A、B、Z 相信号高电平拉高至 3.3 V, 其接口电路如图 9 所示。将编码器的 0 V 与 FPGA 板载的地端相连, 并在输出信号线与地线之间接入相应倍数的放大电阻  $R_3$ , 在电阻的高电势端引出信号端子接入 FPGA 引脚。

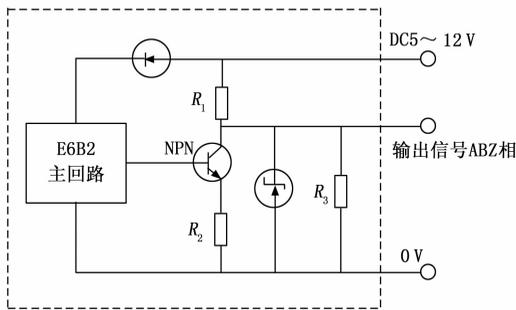


图 9 编码器信号输出电路

编码器的分辨率决定其能识别的最小位移量, 通过倍频的方法可提高编码器识别位移的精确度与准确性<sup>[19]</sup>。4 倍频原理即是在 A 相或 B 相信号的一个周期内, 读取 A、B 相信号的 4 次跳变沿, 并在跳变沿处生成新的脉冲信号  $P$ 。为实现四倍频先对 A、B 相信号延迟打拍得到  $A_1B_1$ , 然后使用 3 次异或逻辑计算得到。公式如下:

$$P = (A \oplus B_1) \oplus (A_1 \oplus B) \quad (6)$$

电机的转动方向通过检测 A 相上升沿时 B 相的电平状态来判断<sup>[20-21]</sup>。规定在 A 相信号上升沿时 B 相为高电平, 方向为反方向 0; 在 A 相信号上升沿时 B 相为低电平, 方向为正方向 1。系统采用的光电编码器的线数为 1 000, 四倍频后编码器转动一圈产生 4 000 个脉冲, 转动识别精度为

0.09°。用于计量四倍频信号脉冲个数的可逆计数器设为 32 位, 计数范围从 -2 147 483 648 ~ +2 147 483 647, 可测量转数最大为 536 870 r, 满足运动距离需求。在四倍频信号的基础上, 通过计量固定脉冲数即固定角度所用的时间来求得转速。

## 3 实验验证与仿真分析

### 3.1 以太网通信验证

基于 Visual Studio 建立上位机通讯界面, 使用 Socket 网络编程实现与 FPGA 的 UDP 通信, 通过以太网与 FPGA 建立连接的上位机界面如图 10 所示。在建立连接之前, 要将电脑的 IP 地址更改为与 FPGA 开发板 IP 地址相同的网关下, 来保证 IP 数据包能正常接收发送。借助第三方网络监控工具 Wireshark 对通信过程实时抓取, 并利用板载的 6 位数码管来显示接收到的控制命令前六位, 以证明通信可靠。



图 10 上位机通讯界面

打开上位机界面, 用网线连接主机和开发板, 点击界面上的连接按键, 显示初始化完成, 之后点击步进电机 1 方位调节栏下的左键, 则在信息反馈栏中显示发送的命令数据为“101010030100000”共 15 字节, 最后一字节为冗余字节不在控制命令范围内。利用 Wireshark 抓取的数据包如图 11 所示, 可以清晰地看到该数据包的目的 IP 地址为本机 IP 地址“192.168.1.102”, 目的 IP 地址即为 FPGA 开发板地址“192.168.1.10”, 发送的数据为字符串对应的 ASCII

码值, 以十六进制显示在 Wireshark 界面的下方数据栏中为“313031303130303330313030303030”。由于板载数显位数有限, 所以在 FPGA 中将控制命令解读后显示前六位数码管上为“101010”, 说明上位机与 FPGA 通信成功, FPGA 接收信息正常。

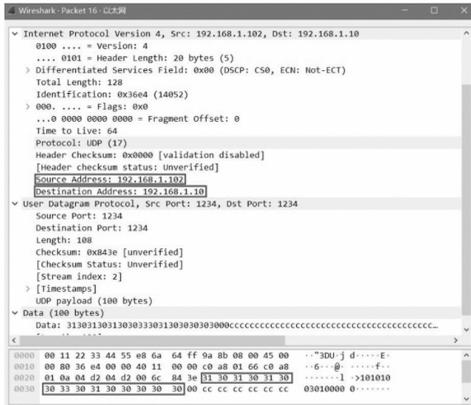


图 11 以太网通讯抓包

### 3.2 控制信号输出仿真

以无刷直流电机梯形加减速控制信号为例, 本文按照等时间间隔对速度作递加和递减处理, 即在人为规定的时间内每经过一段时间间隔便对输出信号的高电平占空比作调整。实际控制中高电平持续时间的取值范围在 1~2 ms, 高电平为 1.5 ms 时电机无动力输出是停止状态, 从 1.5 ms 增加至 2 ms 和减小至 1 ms 的过程是速度正向和反向递增至最大的运动过程, 速度每 20 ms 变化一次, 2 s 内完成加减速动作。

为缩短仿真时间并直观看到输出脉宽变化效果是否与设计相符, 取脉冲基准周期 800 ns, 高电平持续时间从 300 ns 递增至 400 ns 再递减, 速度变化时间间隔为 1.6  $\mu$ s, 加减速时间设为 8  $\mu$ s。通过 Modelsim 软件仿真得到的结果如图 12 所示。从图中标注可以看到前两个脉冲信号高电平持续时间从 380 ns 增加至 400 ns, 此时处在脉宽递增状态, 后两个脉冲信号高电平从 400 ns 递减至 380 ns, 处于脉宽递减状态, 每次速度变化对应脉宽高电平持续时间增加或减少 20 ns, 变化 5 次完成加减速过程。可知程序速度变化效果满足实际需求。

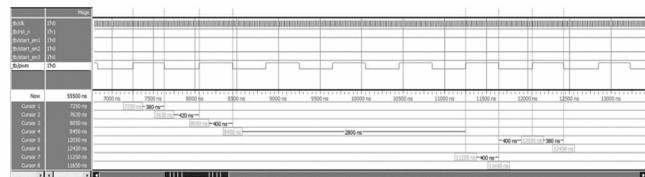


图 12 输出脉冲信号仿真

### 3.3 编码器信号处理仿真

编码器实际传递的 A、B 相信号的频率是有变化的并不是一成不变, 此处主要对四倍频信号的生成、位置累计、速度和方向信号的输出做程序上的逻辑验证。在仿真代码

中给定固定频率为 2.5 MHz 的 A、B 相信号且开始时 A 相超前 B 相四分之一周期, 一段时间后改为 B 相超前 A 相四分之一周期, 再运行一段时间后结束。测试四倍频信号、方向信号和可逆计数器的逻辑正确性, 其仿真结果如图 13 所示。从图中可见在 A、B 相的每一次跳变沿处都对生成成了一个阶跃信号即四倍频信号 cnt\_clk, 频率为 10 MHz。前半段显示 A 相超前 B 相四分之一周期, A 相上升沿时 B 相为低电平, 此时方向信号 direction 显示为高电平 1, 每经过一个四倍频信号上升沿, 可逆计数器 position 的值累加 1; 后半段 B 相超前 A 相四分之一周期, A 相上升沿时 B 相为高电平, 方向信号 direction 变为低电平 0, 可逆计数器 position 的值开始递减。速度信号 speed 的计算结果为 900 000°/s, 与实际结果相同。可知仿真结果与预期一致, 验证了编码器信号处理模块的正确性。

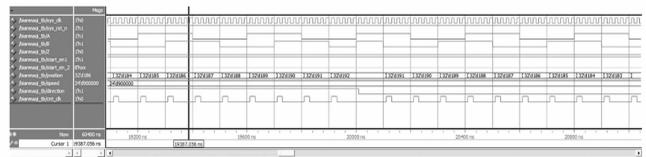


图 13 编码器信号处理仿真

### 3.4 系统运行调试

FPGA 通过网线与电脑相连, 步进电机控制信号输出线接到步进电机方向、使能、脉冲和公共端端口上, 直流电机控制信号线与电调上的脉冲、公共地端口连接, 编码器输出信号线接到 FPGA 对应的 IO 口上。接通电源后运行上位机程序进行调试。

通过上位机与 FPGA 建立连接后发送步进电机控制指令测试位置环。设定步进电机驱动器细分为 4 000, 电机转动速度为 18°/s, 转动一圈后停止即步进电机运行步长值为 4 000。上位机发送的数据为“10100003100000”, 步进电机在 FPGA 的控制下能正常运转并停止。为更直接看到闭环反馈效果, 采用 Modelsim 与 Quartus 联合仿真测试, 系统运行效果如图 14 所示。使能信号 EN1 为初始电机运行脉冲监控信号, 当使能信号 EN1 由逻辑 1 变为逻辑 0 则表示第一阶段脉冲输出完成, 与此同时反馈模块会根据实际测量结果对控制脉冲作第二阶段输出。pulse\_bc 脉冲补偿信号在第一阶段脉冲输出完成后给出了补偿脉冲数为 11, 此时补偿使能信号 EN2 由逻辑 0 变为逻辑 1 并持续到 11 个脉冲补偿完毕后变为低电平, 结束运行。通过将位置信息回传上位机, 对系统控制精度作十组测量, 结果如表 2 所示。测量误差最大为 0.27°最小为 0.09°, 电机定位精度较高且稳定, 能够满足靶标位姿控制要求。

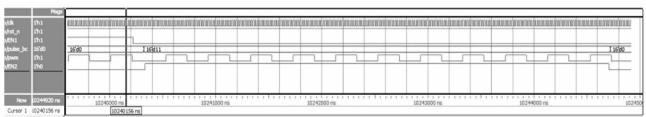


图 14 步进电机位置闭环测试

表 2 360°位置统计 (°)

序号	1	2	3	4	5
测量结果	360.27	360.18	360.09	359.82	359.91
序号	6	7	8	9	10
测量结果	360.18	360.18	360.09	359.82	360.27

#### 4 结束语

本文对基于以太网传输的电机运动控制系统做了一定研究,搭建了 FPGA 与上位机之间的高速以太网通信接口,提供了一种可扩展性丰富的上位机命令格式,并实现了多个电机的异步控制和闭环反馈。基于 Visual Studio 创建了系统人机交互界面,利用 Modelsim 对 FPGA 内部的脉冲输出和编码器信号处理模块进行了验证仿真。仿真和实验结果表明系统响应速度快,具有很好的远距离运动控制精度和处理复杂算法的能力。另外在本系统的基础上,可完成对多种直流电机的远程控制,具有较好的移植性和参考性。

#### 参考文献:

[1] 邱靖超,刘新妹,殷俊龄. 基于 FPGA 的步进电机多轴运动控制系统设计 [D]. 太原:中北大学,2021.

[2] 张龙,叶松,张宝国. 基于 FPGA 的伺服电机转速控制系统研究 [J]. 计算机测量与控制,2019,27(11):66-69.

[3] 贺政,彭端,龙晓斌. 基于 FPGA 的多轴运动控制系统的研究 [D]. 广州:广东工业大学,2020.

[4] 孔德伟,袁国顺,刘小强. 基于 FPGA 的万兆以太网链路的设计与实现 [J]. 微电子学与计算机,2019,36(12):21-25.

[5] 王文,郑建生. 基于 FPGA 的 TCP/IP 网络通信系统的设计与实现 [J]. 现代电子技术,2018,41(8):5-9.

[6] 李杨,苏和平,张丹,等. 基于 FPGA+ARM 多路千兆以太网通信接口设计 [J]. 现代电子技术,2022,45(15):25-29.

[7] 朱保琨,刘广文. 千兆以太网通信端口 FPGA 设计与实现 [J]. 计算机工程与设计,2016,37(9):2292-2298.

[8] 江明旭. 基于 FPGA 的实时以太网通信模块设计 [J]. 单片机与嵌入式系统应用,2022,22(10):62-66.

[9] 潘忠英. 以太网与 CAN 总线通信的 FPGA 设计与实现 [J]. 计算机时代,2022,361(7):44-48.

[10] 赵世超,左金印,魏骁,等. 基于 FPGA 的万兆以太网 UDP 协议通信接口设计 [J]. 电子技术应用,2022,48(10):113-117.

[11] 李杨,苏和平,张丹,等. 基于 FPGA+ARM 多路千兆以太网通信接口设计 [J]. 现代电子技术,2022,45(15):25-29.

[12] 史振国,李德和,尹虎. 基于 FPGA 的高效传输速率的千兆以太网通信系统设计 [J]. 中国新通信,2016,18(14):4-5.

[13] 陈祖霖,沈英,吴靖. 结合 STM32 和 FPGA 的步进电机运动控制系统设计 [J]. 福州大学学报(自然科学版),2020,48(1):27-33.

[14] 顾文斌,杨生胜,王贤良,等. 基于模糊 RBF 神经网络的无刷直流电机 PID 控制 [J]. 计算机技术与发展,2022,32(8):15-19.

[15] 杨昕红,刘长文. 基于 Matlab 的直流无刷电机模糊 PID 控制设计 [J]. 仪表技术与传感器,2019,442(11):105-108.

[16] 奚方园,黄周松,李燕,等. 基于 FPGA 直流电机控制的改进增量式 PID 算法 [J]. 湘南学院学报,2022,43(5):24-29.

[17] 顾祖成,耿小江,王永娟,等. 模糊自适应 PID 控制在无刷直流电机调速系统中的应用 [J]. 机械设计与制造工程,2020,49(1):39-41.

[18] 杨兴旺,胡黄水. 基于 FPGA 的直流电机转速控制研究与设计 [D]. 长春:长春工业大学,2020.

[19] 孙小平. 基于 FPGA 的增量式光电编码器接口电路设计 [J]. 自动化应用,2017(12):6-8.

[20] 张圣超,余洪波. 基于 FPGA 的增量式编码器接口电路的设计 [J]. 电气技术,2014,170(1):38-41.

[21] 张驰. 基于 FPGA 的光电编码器接口设计 [J]. 电子设计工程,2014,22(21):162-164.

(上接第 84 页)

[17] 王家亮,李树华,张海涛. 基于贝叶斯估计与区域划分遍历的四轴飞行器避障路径规划算法 [J]. 计算机应用,2021,41(2):384-389.

[18] 陈志梅,李敏,邵雪卷,等. 基于改进 RRT 算法的桥式起重机械避障路径规划 [J]. 系统仿真学报,2021,33(8):1832-1838.

[19] 陈海,李志刚,冯加果. 基于深度置信神经网络的半潜式平台浮体运动模型和响应预测研究 [J]. 船舶力学,2021,25(5):586-597.

[20] 卞鸿巍,祝中磊,王荣颖,等. 基于  $a&\omega$  六自由度运动模型的船舶运动参数发生器 [J]. 系统工程与电子技术,2022,44(8):2628-2634.

[21] 林亨,方华建,吴冬雁. 基于动态安全距离的 CACC 车辆混合交通流模型 [J]. 北京交通大学学报,2022,46(6):36-42.

[22] 张三川,马喙. 基于轨迹加权预测的主动避障安全距离模型及算法 [J]. 郑州大学学报(工学版),2022,43(3):104-110.

[23] 韩杰祥,张哲,徐可寒,等. 电力电子变压器公共冗余结构及容错控制策略 [J]. 电力自动化设备,2022,42(4):99-106.

[24] 丁亮,王兵,马玺越. 基于压电激励作动器的大型复杂舱壁结构振动主动控制试验研究 [J]. 振动与冲击,2022,41(17):131-137.