doi:10.11835/j.issn.1000-582X.2021.11.005

面向步进电机控制的 RISC-V 微控制器的设计与实现

唐 钏[1],刘 昱[1],曾 林[2]
(1.天津大学 微电子学院,天津 300072;2.北京智芯微电子科技有限公司,北京 100089)

摘要:为了在步进电机控制领域探索灵活的控制和设计方案,针对第五代精简指令集 RISC-V 架构的开源创新,设计开发了用于步进电机控制的 RISC-V 微控制器。在现场可编程门阵列(FPGA)中实现了处理器、存储器、总线、外设及调试接口等模块,构建了可配置的微控制器平台。通过搭建仿真调试环境以及软硬件联合测试,验证了微控制器设计的正确性。在步进电机控制系统测试中,脉冲宽度调制模块产生控制脉冲,正交编码脉冲电路检测转子位置,硬件系统正常工作并且实验的相对误差保持在千分之一量级。

关键词:RISC-V 架构;步进电机;现场可编程门阵列;微控制器;脉冲宽度调制
中图分类号:TP271 **文献标志码**:A **文章编号**:1000-582X(2021)11-031-09

Design and implementation of RISC-V microcontroller for stepper motor control

TANG Chuan[1], *LIU Yu*[1], *ZENG Lin*[2]
(1. School of Microelectronics, Tianjin University, Tianjin 300072, P. R. China;
2. Beijing Smartchip Microelectronics Technology Company Limited, Beijing 100089, P. R. China)

Abstract: To explore a flexible control and design scheme in the field of stepper motor control, a RISC-V microcontroller for stepper motor control was designed and developed, based on the open source innovation of the RISC-V (reduced instruction set computer-five) architecture in the hardware field. By integrating the processor, memory, peripherals and debugging interface modules into a single FPGA (field-programmable gate array) chip, the configurable microcontroller platform was constructed. By building simulation debugging environment and joint debugging of software and hardware, the correctness of microcontroller design was verified. In the test of stepper motor control system, pulse width modulation (PWM) module produced control pulses, quadrature coded pulse circuit (QEP) detected position of the rotor, the hardware system worked normally and the relative error of the experiment was controlled in the order of one thousandth.

Keywords: RISC-V architecture; stepper motors; field programmable gate array; microcontroller; pulse width modulation

---

收稿日期:2020-12-17
基金项目:国家电网公司总部科技项目(5700-201941501A-0-0-00)。
Supported by the Peoject of the State Grid Corporation of China(5700-201941501A-0-0-00).
作者简介:唐钏(1996—),男,硕士研究生,主要研究方向为集成电路设计及嵌入式系统开发,(E-mail) tchuan@tju.edu.cn。
通讯作者:刘昱(1976—),男,教授,博士生导师,(E-mail) liuyu@tju.edu.cn。

步进电机是一种将输入的电脉冲信号转化为相应角位移的机电元件，具有运动精确、易于控制、响应迅速等特点，广泛应用于自动控制等领域[1]。步进电机控制器一般以数字信号处理器(Digital Signal Processor，DSP)芯片和ARM架构的微控制器(MCU)为核心[2-4]，其优点在于硬件结构成熟，功能设计完备。但是，这种方案的基础软硬件平台的版权大多属于商业公司，有严格的使用限制或源码不公开以及需要版税等问题。在嵌入式软核处理器中，具有代表性的有Intel公司的NIOS Ⅱ软核和Xilinx公司的MicroBlaze软核[5-6]，在满足性能的同时，也简化了开发流程，但是都局限于自家厂商的开发套件，通用性不高，并且无法更改软核内部的结构，不利于深入研究和开发。

为了构建步进电机控制技术方案，设计了基于RISC-V指令集架构的微控制器。一方面，RISC-V开源指令集具有免费开放、架构精简、模块化、可扩展等优势[7]；另一方面，将处理器、存储器、总线以及其他IP核集成到单一的FPGA芯片上，能够减少系统的物理组件数，方便不同应用领域的差异化设计。目前，国内外关于RISC-V处理器在电机控制器中的应用相对较少[8-9]，且大部分都集中在商用领域。

1 RISC-V微控制器设计

1.1 RISC-V指令集介绍

开源指令集，比如OpenRISC、OpenSPARC和MIPS R6等，由于指令集设计的不完备、碎片化以及运营模式等原因无法广泛持续地生存下去。美国加州大学伯克利分校的研究人员，设计RISC-V开源指令集[10-11]。RISC-V指令集具有模块化、架构精简、安全、无需向后兼容、可扩展等特点。RISC-V官方详细比较了指令集设计的衡量标准，相比其他指令集架构，在成本、简洁性、性能、架构和具体实现的分离、提升空间、程序大小、易于编程/编译/链接等方面，RISC-V均具有一定优势[12-13]。RISC-V指令集由基本的整数指令集“I”和可选的扩展指令集组成。I指令集包括RV32I、RV32E、RV64I或RV128I，数字前缀表示以位为单位的地址空间。RV32E是RV32I面向嵌入式系统设计的简化版本，其整数寄存器的数目从32个减少到16个。扩展指令集包括整数与除法指令“M”、原子操作指令“A”、单精度(32比特)浮点指令“F”、双精度(64比特)浮点指令“D”、压缩指令“C”等[14-15]。RISC-V这种模块化的设计使得在不同的场景中，可以选择配置不同的指令集组合来满足应用需求，不管是小面积、低功耗的嵌入式场景，还是高性能计算领域，RISC-V均能有一席之地。

1.2 微控制器的整体结构及流水线设计

设计的微控制器采用基于RISC-V指令集架构的32位处理器内核，存储模块包括指令存储(Instruction RAM)、数据存储(Data RAM)和用于存放内核启动代码的Boot ROM，微控制器使用AXI作为系统高速总线，通过总线协议转换桥可连接到APB子系统。AXI总线连接了SPI Slave以及JTAG调试接口，APB总线挂载了PLIC和CLINT中断接口以及UART、PWM、GPIO、SPI等常见的外设模块。微控制器的具体结构如图1所示。

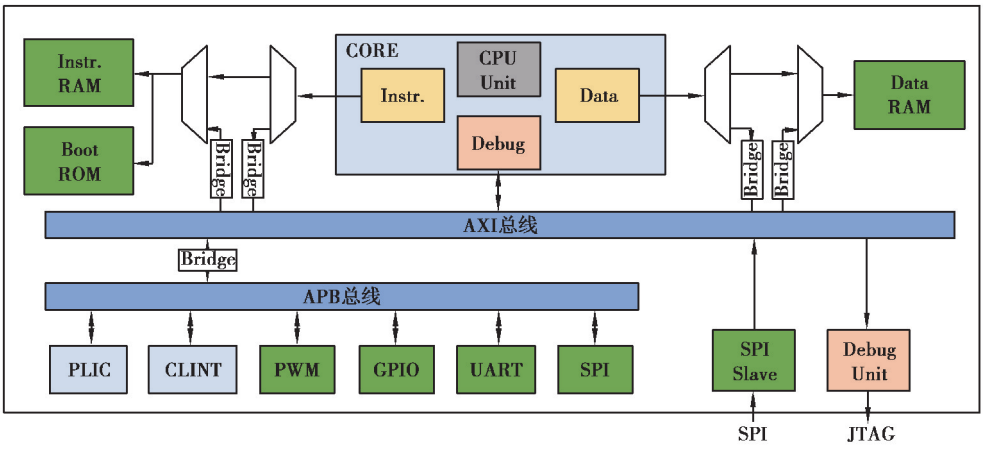

图1 微控制器结构示意图

Fig. 1 Schematic diagram of microcontroller structure

为了使微控制器的各个模块正常工作，在硬件设计中通过总线分发模块设置各个模块的地址区间，对不同模块的地址进行分配。通过在板级支持包 BSP 文件中定义与硬件模块相同的地址，在软件开发过程中，通过基地址加地址偏移量的方法配置底层寄存器，能够有效地配置系统各个模块的工作方式，设置和使用不同的外设功能。微控制器的地址映射如表 1 所示。指令存储和数据存储的大小可以设置，默认为 64 kB。在外设区域，每个普通的外设单元占 4 kB 的地址空间，并预留了足够的空间方便扩展其他需要的外设。

表 1　微控制器的地址分配

Table 1　Address assignment for the microcontroller

总线模块	地址区间	描述
DEBUG	0X0000_0000-0X0000_0FFF	调试器使用的地址
BOOT ROM	0X0000_1000-0X0000_1FFF	可存放系统启动代码
SPI Slave	0X2000_0000-0X2FFF_FFFF	片外 FLASH 映射过来的地址
APB 外设区	0X4000_0000-0X4000_FFFF	APB 外设寄存器的地址范围
指令存储区	0X8000_0000-0X8001_FFFF	用于存储指令
数据存储区	0X9000_0000-0X9001_FFFF	用于存储数据

设计的 32 位 RSIC-V 处理器具有四级流水线结构，单发射顺序执行，支持 RV32IM 整数和乘除法指令集的配置组合。处理器流水线结构如图 2 所示。

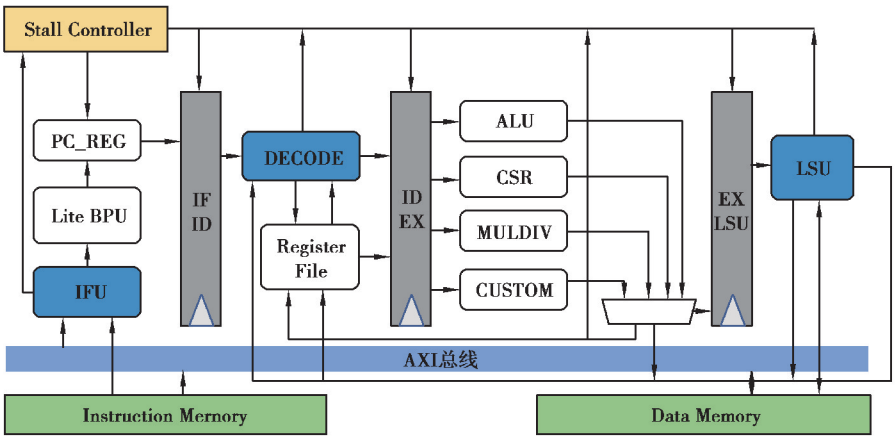

图 2　处理器流水线结构图

Fig. 2　Processor pipeline diagram

该处理器的流水线结构为“取指”、“译码”、“执行”、和“访存及写回”4 个阶段。在取指阶段，IFU 取指单元通过生成的地址控制信号访问指令存储或者外部存储，Lite BPU 采用简单的静态分支预测对指令的跳转地址进行判断，地址生成逻辑“PC_REG”产生待取指令的地址信号送入译码模块。译码模块主要是由组合逻辑电路组成，将取指阶段发送过来的指令按照 RISC-V 指令编码规则分解出各个指令字段，得到指令类型、操作数寄存器等信息，在寄存器文件模块中实现了 32 个 32 位的整数通用寄存器，其中 X0 被预留为常数 0，处理器可以同时对 2 个寄存器进行读操作和 1 个寄存器进行写操作。在执行模块中，通过对具体指令和操作数的解析后派遣给不同的运算单元模块去执行。主要有以下几种：算术逻辑运算单元 ALU 模块，整数乘除法单元 MULDIV 模块，处理异常状态寄存器的 CSR 模块以及自定义指令运算单元 CUSTOM 模块。通过运算单元复用一条数据通路的设计，减小了这部分的面积开销。访存及写回模块主要用于 Load 和 Store 指令的地址生成，对指定地址的内存进行读或写，与数据存储进行数据交互，并将不同运算单元执行的

结果写回到寄存器文件中。控制模块不属于流水线阶段,主要控制整个流水线的暂停或清除操作。

2 步进电机控制系统设计

2.1 系统硬件结构设计

文中步进电机控制系统主要由上位机和下位机两部分构成,如图3所示,上位机系统由PC机组成,可以通过USB转UART与微控制器通信,通过USB转JTAG向微控制器下载程序或者代码调试等,下位机系统以Artix-7 FPGA实现的RISC-V微控制器作为主要控制模块,分别连接电机驱动器、步进电机、编码器以及正交编码脉冲(QEP,Quadrature Encoder Pulse)检测电路等模块。这种模块化的系统结构可以降低整体的耦合度,提高系统的可移植性以及错误排查效率。

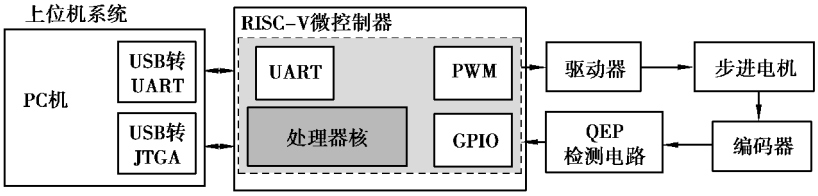

图3 系统结构设计图

Fig. 3 System structure

在系统硬件结构中,通过微控制器的PWM外设来生成相应数目和频率的脉冲,控制电机的运行速度和转动方向。驱动器主要由控制级驱动电路和功率驱动电路两部分构成,其中,控制级驱动电路主要用于分配脉冲信号给步进电机的各相绕组,功率级驱动电路用于提供高电压、大电流给步进电机。为了验证步进电机系统工作的准确性,采用了增量式光电编码器来检测步进电机转子位置。

2.2 PWM模块设计

为了准确控制步进电机的运动,利用PWM脉宽调制技术生成相应数目和频率的脉冲。设计中的PWM模块支持3个子模块,分别为PWM0、PWM1、PWM2。其中,PWM0是宽度8 bit的比较器,而PWM1和PWM2宽度为16 bit,工作原理和特性完全相同。每个PWM支持4个可编程的比较器(PWMCMP0-PWMCMP3),每个比较器可以产生对应的1路PWM输出和中断,文中的微控制器最多可以支持12路PWM输出通道。PWM的可配置寄存器如表2所示。

表2 PWM寄存器和功能

Table 2 PWM registers and functions

寄存器名	偏移地址	功能
PWMCFG	0X00	PWM配置寄存器
PWMCOUNT	0X08	PWM计数器计数值寄存器
PWMS	0X10	PWM计数器比较值寄存器
PWMCMP0-3	0X20-0X2C	PWM比较器寄存器

PWM模块挂载在APB总线上,以32位对齐的访问方式读写内部寄存器,通过对PWMCFG寄存器各比特域的赋值来开启或关闭特定功能,可以选择不同的PWM脉冲以及生成方式。PWMCOUNT寄存器反映的是PWM计数器的值,使用后会在每个时钟周期加1,达到预先设定的值后,计数器归0;假如没有预设值,计数到最大值即全1后溢出归0。当PWM计数器归0后,可以设置寄存器使其重新开始计数或者停止。当PWMCOUNT的值大于PWMS的值时,PWM输出信号为高电平;反之,信号为低电平。如图4所示,假设采用2 bit宽的PWMS,当PWMS计数器计数至最大值3时,溢出归0并重新开始计数。此时,PWM计数周期为4。当PWMCMP0分别配置为0、1、2、3时,PWM输出的脉冲信号波形。

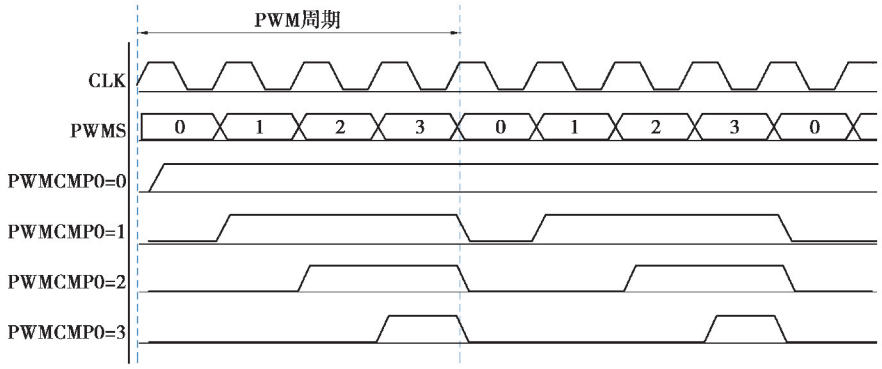

图 4　PWM 脉冲信号输出波形

Fig. 4　Output waveform of PWM pulse signal

通过设置 PWMCFG 寄存器，配置 PWM0 和 PWM1 的 2 个通道各产生不同频率和占空比的脉冲，在 PWM 的输出引脚端同时对 4 个通道用逻辑分析仪进行测量，如图 5 所示，PWM 输出端均得到了准确的频率和占空比的脉冲，PWM 功能验证正确。

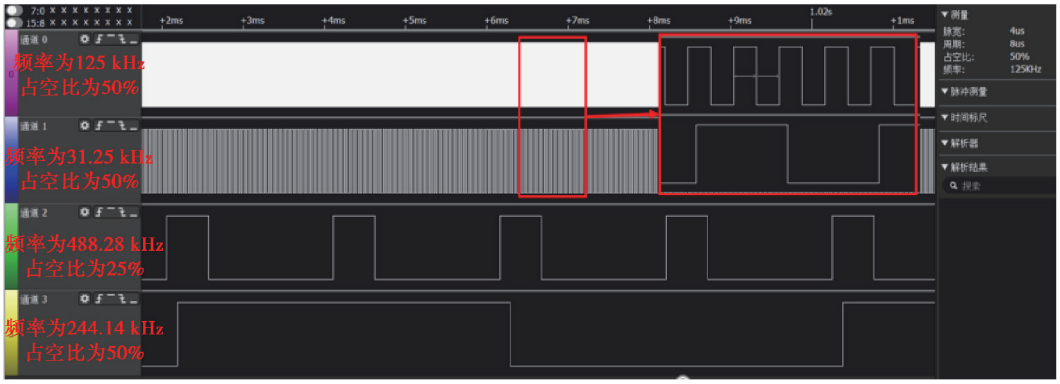

图 5　PWM 输出脉冲信号测试

Fig. 5　Test diagram of PWM output pulse signals

2.3　正交编码脉冲(QEP)检测模块设计

为了验证步进电机系统工作的准确性，通过增量式光电编码器来检测步进电机的转子位置。编码器固定在步进电机转轴上，当电机旋转时，编码器输出 2 路正交的 A、B 方波脉冲以及用于基准点定位的脉冲信号 Z。QEP 检测模块主要由载有上拉电阻及滤波电容的功能板、光耦隔离模块以及 GPIO 端口连线组成。由于该编码器是开漏输出的，需要接上拉电阻才能连接到 GPIO 端口。滤波电容用来滤除编码器输出端产生的部分杂波。光耦隔离模块用来实现模拟信号和数字信号的隔离，避免编码器的输出信号与控制模块的引脚连线产生干扰信号。QEP 模块的主要工作原理是：利用 GPIO 端口检测 A 或 B 相脉冲的上升沿，通过脉冲相位来判断电机转动方向，在上升沿进行触发并根据方向信号判定增或减计数。

3　系统仿真与测试

3.1　微控制器的 FPGA 实现和软件测试

对于设计的微控制器平台能够通过编写 C 代码方便地实现外围设备的控制功能，并且在 FPGA 中能够对各个组成模块进行修改和优化。利用 Vivado 软件完成对微控制器在 FPGA 上的实现，整个微控制器系统的资源占用情况如表 3 所示。其中，PWM 模块所占用的 LUT 数目为 356，消耗资源数较少，满足步进电机控制器的应用需求。

表3 FPGA资源占用表

Table 3 FPGA Resource occupancy

资源类型	消耗数目	可用总数	消耗率/%
LUT	13 878	20 800	66.72
LUTRAM	168	9 600	1.75
FF	9 826	41 600	23.62
BRAM	32	50	64.00
IO	65	170	38.24
MMCM	1	5	20.00

通过选取不同的嵌入式软核,将蜂鸟E203、RI5CY和Cortex-M3内核在FPGA(-1等级)中的资源消耗、最高频率以及Coremark跑分结果与文中处理器内核进行对比,内核消耗资源和性能对比如表4所示。

表4 不同软核处理器的比较

Table 4 Comparison of different soft-core processors

内核	LUTs	FF	运行频率/MHz	CoreMark/MHz
蜂鸟E203	4 105	1 807	20	2.17
RI5CY	7 095	2 581	50	2.43
Cortex-M3	15 096	5 189	35	3.32
本设计	6 239	2 287	40	2.55

相比RISC-V架构的内核即蜂鸟E203和RI5CY,CoreMark执行结果表明,文中处理器内核在单位频率的计算能力上有一定提升。相比ARM架构的Cortex-M3内核,在FPGA实现的最高运行频率提高了14.3%,电路资源消耗减小了58.7%,在使用FPGA资源不多的情况下,实现了较高的运行频率和单位频率计算能力。

系统的软件开发在Ubuntu环境下进行,开发步骤如下:首先,使用命令行运行脚本文件,通过RISC-V GNU工具链对C语言程序进行编译;然后,编写OpenOCD的底层驱动文件,使得GDB调试工具通过JTAG接口连接到RISC-V微控制器;最后,将程序下载到硬件平台中,对其实际运行情况进行测试。在UART通信、LED点灯以及中断控制等一系列功能测试后,均观察到正确的实验结果,说明微控制器的软硬件功能设计正确。板级验证成功后,生成内存配置文件(MCS)写入FLASH中,由于FLASH掉电不丢失的特性,上电即可运行MCU系统,使用时无需反复写FPGA,使得软件编程更方便。

3.2 步进电机控制系统测试与分析

在步进电机控制系统的性能测试实验中,驱动对象是1台步距角为1.8°、扭矩为1.2 N·m、额定电流为2.5 A、转动惯量为300 g·cm^{2}的两相四线混合式步进电机,型号57BYG250B。该步进电机主要应用于低速工业控制领域,转速不超过1 200 r/min。驱动器使用的是东芝公司TB67S109A芯片,是一种配备PWM斩波器的两相步进电机驱动器,外围元件较少,提供高达4 A的峰值电流和DC40V的最大工作电压。编码器选用的是HN38-06-N型号的增量型旋转编码器,线数为600,即每转一次产生600个脉冲,DC7-30V电压供电。实验原理如图6所示,在实验中驱动器的供电电压为24 V,编码器供电电压为12 V。驱动器采用的是共阳极接法,将PUL^+、DIR^+、EN^+与FPGA开发板的3.3 V相连,EN^-、DIR^-与GPIO相连,通过控制GPIO电平使能电机或者控制其正反转,PUL^-与PWM脉冲输出端相连,将输出脉冲传递给驱动器。

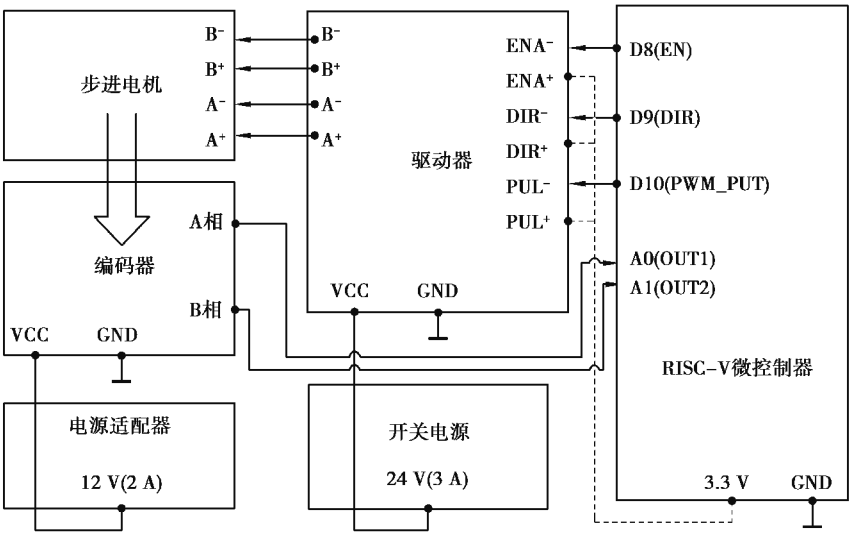

图 6　实验原理图

Fig. 6　Experimental schematic diagram

在系统测试中，先对每个组件进行单独测试，检查无误后，用杜邦线连接系统的各个模块，连线完毕后如图 7 所示。系统测试的程序流程为：硬件初始化后，配置外部中断和计时器中断，从串口终端发送控制命令，通过是否使能、正反转、产生脉冲频率以及运行时间的配置，设置 PWM 相关的寄存器，PWM 产生相应的输出脉冲到步进电机驱动器。然后开启计时器，GPIO 端口检测编码器输出的 A 或 B 相脉冲上升沿信号，根据方向判断逻辑得到电机转动方向，并根据上升沿信号进行计数。当计时到设定值时，串口发送编码器 A 或 B 相的累计脉冲总数，程序结束。系统测试程序的主要流程如图 8 所示。其中，定时部分是通过 CLINT 实现的，在计时器中断开启后，根据定时时间来设置 mtimecmp 比较寄存器，mtime 计时器一直默认计数，当其值大于或等于 mtimecmp 的值后，步进电机停止运动，程序结束。

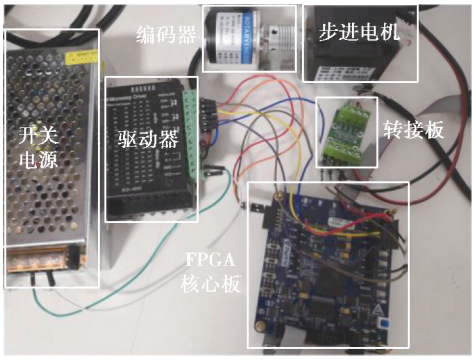

图 7　系统结构组成图

Fig. 7　System connection diagram

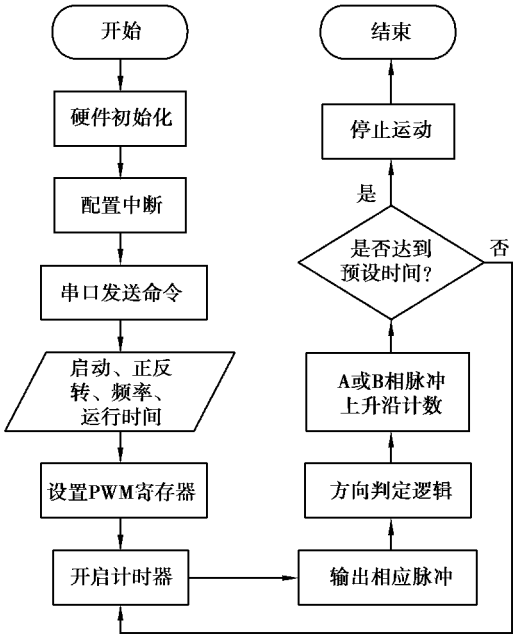

图 8　系统测试程序流程图

Fig. 8　Flow chart of system test program

在实验中，检查系统的各个模块并且连线无误后通电，打开串口终端并设置波特率，将编译后的主程序下载到微控制器中进行测试。PWM 模块所在时钟域的频率为 32 MHz，选择 PWM0 的通道 0 作为脉冲产

生通道，设置时钟分频系数为 2 使其产生 31.25 kHz 频率的输出脉冲。在 1/32 细分的测试实验中，对编码器输出的 A、B 两相脉冲用示波器多次测量得到，两相脉冲输出频率范围在 2.89～2.95 kHz 之间，编码器频率的理论值应为 2.93 kHz。该误差是信号经过各级设备时的传输误差以及编码器存在的固有误差，在允许范围以内。

在 1/32 细分模式下，分别设置运行时间为 1 s、10 s、1 min，对应的理论脉冲数为 2 929、29 296、175 781，3 组实验实际对应的脉冲计数结果如图 9 所示。实际结果和理论结果进行误差计算，如表 5 所示，实验的相对误差控制在千分之一量级以下。更改细分模式为 1/16，此时驱动器输出的频率较大，步进电机产生啸叫声且无法工作，所以更改 PWM 分频系数为 3，即 PWM 输出频率为 15.625 kHz，在不同运行时间的实验测试中，实验误差和 1/32 细分实验近似相同。

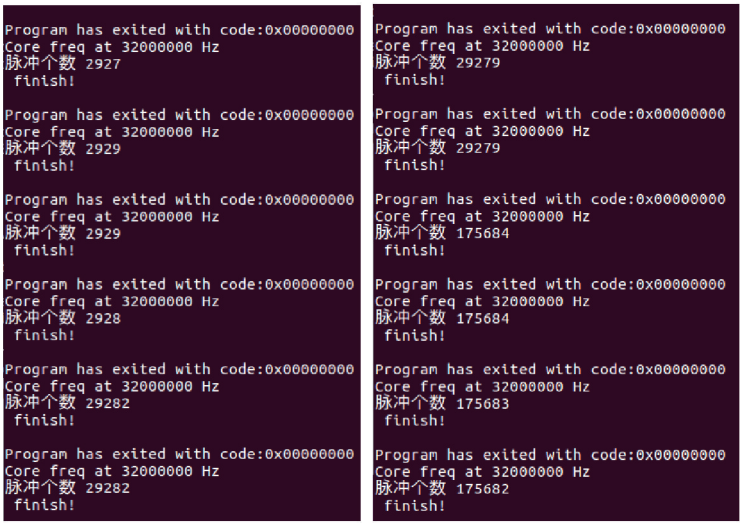

图 9　编码器脉冲计数示意图

Fig. 9　Schematic diagram of encoder pulse count

表 5　实验数据表

Table 5　Experimental data

实验	细分数	设置运行时间/s	实际计数值	理论值	误差/%
1	32	1	2 927	2 929	0.068 3
2	32	1	2 928	2 929	0.034 1
3	32	1	2 929	2 929	0
4	32	10	29 282	29 296	0.047 8
5	32	10	29 282	29 296	0.047 8
6	32	10	29 279	29 296	0.058 0
7	32	60	175 684	175 781	0.051 8
8	32	60	175 683	175 781	0.055 8
9	32	60	175 682	175 781	0.056 3

对于步进电机运行时间长的失步情况，可采取步校验设计，将图 8 中的“停止运动”替换为“编码器脉冲计数小于理论值，则补偿相差的输入脉冲数”，比如，在运行时间为 1 min 的实验中，对其补偿 98 个脉冲，即可实现位移的精确控制。经过测试，该控制系统能够准确地控制步进电机的运行步数，满足步进电机低速控制场景的性能要求。

4　结　论

1)针对步进电机控制系统的应用要求，采用开源的 RISC-V 指令集设计 32 位的四级流水线处理器核，通过 AXI 和 APB 片上总线实现对外部设备的控制。

2)在单片 FPGA 中实现了 RISC-V 软核处理器、总线、存储器、外设和调试接口等模块组成的微控制器，通过板级测试验证了微控制器设计的正确性。与其他嵌入式软核相比，文中设计的内核在使用 FPGA 资源不多的情况下，实现了较高的运行频率和单位频率计算能力。

3)设计了 PWM 和 QEP 等模块，结合 FPGA 核心板、驱动器、PC 机等搭建了步进电机控制系统。通过软硬件仿真和实验测试表明，系统的相对误差控制在千分之一量级，通过步校验设计控制效果更加精确。

参考文献：

[1] Acarnley P. Stepping Motors: a guide to theory and practice [M]. London: Institution of Engineering and Technology, 2002.

[2] Zhang L L, Liu L, Shen J, et al. Research on stepper motor motion control based on MCU[C]// 2017 Chinese Automation Congress (CAC). October 20-22, 2017, Jinan, China. IEEE, 2017: 3122-3125.

[3] Le K M, Van Hoang H, Jeon J W. An advanced closed-loop control to improve the performance of hybrid stepper motors[J]. IEEE Transactions on Power Electronics, 2017, 32(9): 7244-7255.

[4] 荣盘祥，张亚慧，张欢欢，等. 基于 DSP 的运动控制卡的研究与开发[J]. 电机与控制学报，2011，15(3)：35-39.
Rong P X, Zhang Y H, Zhang H H, et al. Research and development of multiple-axis motion control board based on DSP [J]. Electric Machines and Control, 2011, 15(3): 35-39.(in Chinese)

[5] 孙恺，王田苗，魏洪兴，等. 嵌入式 CPU 软核综述[J]. 计算机工程，2006，32(7)：6-9.
Sun K, Wang T M, Wei H X, et al. Summary of embedded CPU soft-core[J]. Computer Engineering, 2006, 32(7): 6-9.(in Chinese)

[6] Tong J G, Anderson I D L, Khalid M A S. Soft-core processors for embedded systems[C]// 2006 International Conference on Microelectronics. December 16-19, 2006, Dhahran, Saudi Arabia. IEEE, 2006: 170-173.

[7] Asanovic K, Avizienis R, Bachrach J, et al. The rocket chip generator[J]. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, 2016.

[8] Proeger J. Welterster FOC-Motorcontroller mit RISC-V als single-chip-lsung[J]. Elektronikpraxis, 2019(2):36-37.

[9] Fernández-Mesa B, Andrade L, Pérrot F. Electronic system level design of heterogeneous systems: a motor speed control system case study[C]// 2019 17th IEEE International New Circuits and Systems Conference (NEWCAS). June 23-26, 2019, Munich, Germany. IEEE, 2019: 1-4.

[10] Parulkar I, Wood A, Hoe J C, et al. OpenSPARC: An open platform for hardware reliability experimentation[C]. Fourth Workshop on Silicon Errors in Logic-System Effects (SELSE). Citeseer, 2008:1-6.

[11] Tandon J. The openrisc processor: open hardware and linux[J]. Linux Journal, 2011, 2011(212):6.

[12] Patterson D, Waterman A. The RISC-V Reader: an open architecture Atlas[M]. Strawberry Canyon, 2017.

[13] Höller R, Haselberger D, Ballek D, et al. Open-source RISC-V processor IP cores for FPGAs—overview and evaluation[C]// 2019 8th Mediterranean Conference on Embedded Computing (MECO). June 10-14, 2019, Budva, Montenegro. IEEE, 2019: 1-6.

[14] Waterman A S. Design of the RISC-V instruction set architecture[D]. UC Berkeley, 2016.

[15] Waterman A, Lee Y, Patterson D A, et al. The RISC-V instruction set manual. volume 1: user-level ISA, version 2.0 [R]. Defense Technical Information Center, 2014.

（编辑　陈移峰）