electronics

MDPI

Article

Development of Multi-Motor Servo Control System Based on Heterogeneous Embedded Platforms

Mingrui Gou, Bangji Wang * and Xilin Zhang

School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China; gmr2022201090@my.swjtu.edu.cn (M.G.); zhangxl@my.swjtu.edu.cn (X.Z.)
* Correspondence: bangjiw@swjtu.edu.cn

Abstract: Multi-motor servo systems are widely used in industrial control. However, the single-core microprocessor architecture based on the microcontroller unit (MCU) and digital signal processor (DSP) is not well suited for high-performance multi-motor servo systems due to the inherent limitations in computing performance and serial execution of code. The bus-based distributed architecture formed by interconnecting multiple unit controllers increases system communication complexity, reduces system integration, and incurs additional hardware and software costs. Field programmable gate array (FPGA) possesses the characteristics of high real-time performance, parallel processing, and modularity. A single FPGA can integrate multiple motor servo controllers. This research uses MCU + FPGA as the core to realize high-precision multi-axis real-time control, combining the powerful performance of the MCU processor and the high-speed parallelism of FPGA. The MCU serves as the central processor and facilitates data interaction with the host computer through the controller area network (CAN). After data parsing and efficient computation, MCU communicates with the FPGA through flexible static memory controller (FSMC). A motor servo controller intellectual property (IP) core is designed and packaged for easy reuse within the FPGA. A 38-axis micro direct current (DC) motor control system is constructed to test the performance of the IP core and the heterogeneous embedded platforms. The experimental results show that the designed IP core exhibits robust functionality and scalability. The system exhibits high real-time performance and reliability.

Keywords: multi-motor servo system; MCU + FPGA; heterogeneous embedded platforms; integration; IP core; CAN; FSMC; DC motor; real time

check for updates

Citation: Gou, M.; Wang, B.; Zhang, X. Development of Multi-Motor Servo Control System Based on Heterogeneous Embedded Platforms. *Electronics* **2024**, *13*, 2957. https://doi.org/10.3390/electronics13152957

Academic Editor: Raffaele Giordano

Received: 27 June 2024
Revised: 20 July 2024
Accepted: 24 July 2024
Published: 26 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid development of microelectronics technology and industrial automation has led to a significant increase in the demand for servo motors [1]. The multi-motor servo system composed of multiple servo motors can realize various complex functions according to actual needs and has a wide range of application prospects [2], such as industrial robots, robot arms, computer numerical control (CNC) machine tools, web winding systems, electric vehicles, and mechanical phased array antennas [3–8]. These devices require the collaborative work of multi-axis motors; multi-axis motors are required to operate at the same speed, or each axis motor needs to reach a specified position within a specified time, and so on.

MCUs and DSPs are frequently employed for motor control. They have rich peripherals and substantial computing power and can realize the servo control of a single motor [9–11]. However, their code is executed in serial, which limits the processor's performance and is unsuitable for occasions with high real-time and reliability requirements. A single MCU or DSP makes it challenging to meet the needs of a high-precision multi-motor servo system. The distributed architecture formed by the interconnection of multiple MCUs or DSPs will undoubtedly reduce the system integration and increase the design difficulty. At the same time, the communication between buses formed by this

interconnection is complex. It is easy to have the problem of poor synchronization between the motors of each axis [12]. FPGA chips possess the advantages of parallel processing, flexibility, and high real-time performance. Many scholars have implemented motor control algorithms through FPGA [13–16]. FPGA can perform multiple calculations in parallel in a single clock cycle to realize the acceleration of fuzzy control, neural networks, and other algorithms [17,18]. Based on this characteristic of FPGA, coupled with the modular design method, it is easy to realize the synchronous control of multi-axis motors [19]. Due to the limitation of hardware logic units, FPGAs have limited capabilities for floating point operations, trigonometric calculations, and division calculations [20–22]. In response to this problem, FPGA vendors have introduced related IP cores and softcore processors such as Nios II and MicroBlaze. Ying-Shieh Kung et al. implemented adaptive fuzzy control of permanent magnet synchronous motor using FPGA; they implemented point-to-point trajectory planning in the Nios II softcore processor and then used a hardware logic unit to implement adaptive fuzzy controller and vector control of current [23]. Fengqiu Xu et al. used intellectual property (IP core) for the efficient calculation of force and torque and a MicroBlaze processor to develop a CNC regulator to achieve stable multi-axis motion of a magnetically levitated rotary table [24]. However, considering the implementation of multi-axis motor control, this presents a considerable challenge to the logic resource requirements of FPGA. Qiang Gu et al. implemented multi-axis servo control with hardware and software synergy on Zynq SoC [25]; the PL side of the Zynq typically has fewer pins than a general FPGA, which is insufficient when faced with the need to control motors in more axes. Choosing a Zynq chip with a larger number of pins increases the cost. Separating the processor from the hardware logic can facilitate the expansion of the number of motor axes to be controlled and provide more flexibility.

Therefore, in order to realize high precision and high real-time control of a multi-motor servo system, a multi-servo motor control scheme based on the MCU and FPGA heterogeneous embedded platforms is proposed in this paper. This scheme combines the computing power of MCU with the parallel processing capability of FPGA. The design of the control system is divided into two parts by software and hardware function division. The first part is the MCU as the central processor. MCU has rich peripherals and powerful processing performance and can interact with the host computer to complete data analysis, calculation, and delivery. At the same time, the MCU acts as the monitoring center of the whole system and monitors the status of each unit. The second part is the FPGA as a co-processing unit. A motor controller IP core is designed, packaged, and multiplexed to accomplish the real-time control of multi-axis servo motors by taking advantage of the parallelism of FPGA. In addition, the real-time performance of heterogeneous embedded platforms is ensured by using the FSMC parallel bus to realize cross-clock data interaction between MCU and FPGA. The diagram of a multi-motor servo control system with the heterogeneous embedded platforms as the core is shown in Figure 1.

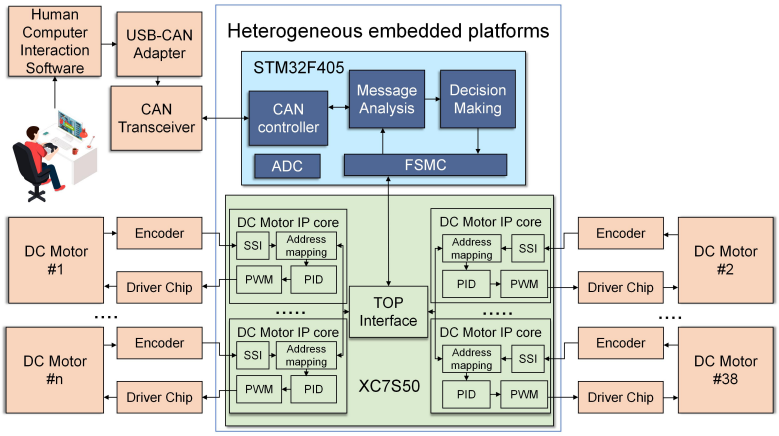

Figure 1. Multi-motor servo control system based on heterogeneous embedded platforms.

2. Background Research

2.1. Mathematical Model of Micro DC Motor

The control object of this paper is a kind of micro brush DC motor, which is simple to control and has a low cost. Most of the micro DC motors adopt a hollow cup structure, which has higher energy density, lower moment of inertia, and smaller volume size. Compared with the traditional DC motor, for the hollow cup structure of the motor rotor without an iron core, the inductance is small, the electrical time constant is small, and the higher air gap magnetic density and the more minor moment of inertia make the mechanical time constant of the motor smaller. Therefore, this kind of micro DC motor has good dynamic performance, a wide speed range, and can control the speed with high precision. It is widely used in optical fiber positioners, gyroscopes, spacecraft, and other fields [26,27].

The voltage balance equation of the armature circuit of the micromotor is shown in Equation (1):

$$u_a = k_e w + i_a R + L\frac{di_a}{dt} \tag{1}$$

where u_a is the motor armature voltage, k_e is the motor's reverse electromotive force coefficient, w is the motor's angular velocity, i_a is the motor armature current, R is the armature resistance, and L is the armature inductance.

The electromagnetic torque balance equation of the micromotor is shown in Equations (2) and (3):

$$T_e - T_L = J\frac{dw}{dt} + Bw \tag{2}$$

$$T_e = k_m I_a \tag{3}$$

where T_e is the rated torque of the motor, T_L is the load torque, J is the rotational inertia, B is the damping coefficient, w is the angular velocity of the motor, k_m is the torque coefficient, and I_a is the armature current.

2.2. Control Strategy of Micro DC Motor

Most motor servo control systems focus on three indicators, namely, position, speed, and torque. Position control will control the motor to rotate at a certain angle, and the real-time angle of the motor can be fed back through the encoder. Speed control includes the control of the motor speed and direction. The real-time speed of the motor can be obtained by the speed sensor or by differentiating the position value. The torque of the motor can measure the size of the motor load, which is proportional to the armature current, and the control of the torque is actually the control of the current. Therefore, the motor servo control system is generally a three-loop structure, including the current loop (the innermost loop), the speed loop (the middle loop), and the position loop (the outermost loop) [28].

In most common servo systems, position control has the highest priority to ensure high accuracy and high real-time performance [29,30]. In addition, in the traditional three-loop control, the speed loop may cause a time delay in the control system. In order to improve the real-time performance of position control, some scholars have adopted the double-loop control structure of the position loop and current loop [31]. The inductance of the micro DC motor with a hollow cup structure is minimal, which makes its electrical time constant very small and the response speed very fast. However, current discontinuity due to small inductance can lead to torque fluctuation [26]. Increasing the pulse width modulation (PWM) frequency can reduce the torque fluctuation [27].

Senthilnathan, A. et al. designed a brushless DC motor controller based on FPGA by adopting hysteresis current control in order to obtain stable torque fluctuation and fast response [32]. In order to ensure the high real-time performance of the multi-motor servo system and the stability of the motor operation, this paper adopts the dual-loop control structure of the position loop and current loop; the modulation frequency of the two is 10 kHz and 195.31 kHz, respectively. The position loop is used as the main loop

with the proportional–integral–derivative (PID) control algorithm. The motor's actual position is tracked in real time, and the difference between the position reference value and the position feedback value is made. The error is sent to the position loop controller and calculated by the control algorithm to obtain the duty cycle. The duty cycle is sent to the drive chip to generate two PWM waves and then transmitted to the drive circuit to drive the motor operation. The current loop adopts hysteresis current control, built into the driver chip to inhibit current interference. The maximum current through the armature is limited to prevent motor damage. The dual-loop control structure is shown in Figure 2.

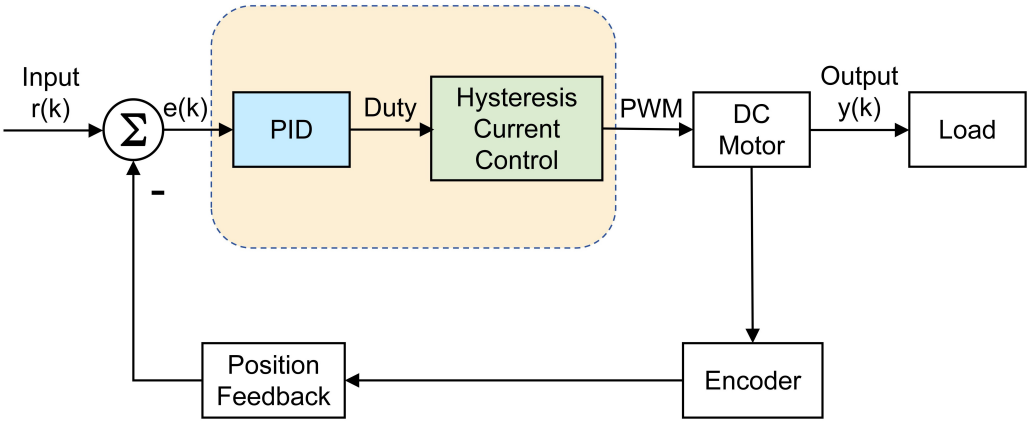

Figure 2. Double loop control structure of micro DC motor.

In Figure 2, the micro DC motor is the controlled object, and the encoder is the element that feeds back the motor's actual position. The PID calculation of deviation e(k) and hysteresis current control make the final output value y(k) quickly track the set value r(k). The expression of the classical positional PID control algorithm is shown in Equation (4):

$$u(k) = k_p e(k) + k_i T \sum_{j=0}^{k} e(j) + \frac{k_d}{T}[e(k) - e(k-1)] \quad (4)$$

where k_p, k_i, k_d represent the proportional, integral, and differential coefficients of the PID algorithm, respectively, and their values are determined by the motor and load parameters; (e(k)) and (e(k − 1)) are the deviation values of the current target position from the actual feedback position and the last deviation value, respectively; T is the control period; and $u(k)$ is the output of the PID control algorithm.

3. Implementation Process

3.1. Heterogeneous Embedded Platforms

Embedded platforms have steadily increased their computing power in recent years. The clock rate of an MCU can reach hundreds of MHz, providing faster processing performance. Rich peripherals and internal hardware modules can help the processor complete operations faster. FPGAs can achieve high-speed data processing due to their parallelism and are often used as algorithm accelerators. The multi-motor servo system involves many mathematical calculations, such as the motion trajectory planning of each axis motor and the realization of the control algorithm of each axis motor.

Considering the hardware resource requirement of multi-motor servo control, we choose STM32F405 with the 168 MHz clock rate and XC7S50 FPGA of Xilinx as the heterogeneous embedded platform. STM32F405 has rich peripherals, such as CAN controller, FSMC controller, and so on. The CAN bus is a stable industrial field bus that can realize the data interaction between the control platform and the host computer. FSMC is a parallel bus that can realize fast communication between MCU and FPGA. The XC7S50 is manufactured

at 28 nm. It has approximately 50 k LUT4 hardware logic units and 120 25-by-18 multipliers. Sufficient resources are available for the implementation of motor control algorithms.

Heterogeneity can lead to data transfer across clock domains. Since the MCU is a fast-clock domain, data processing is required to prevent data loss when receiving data at the FPGA side. The overall hardware structure is schematically shown in Figure 3.

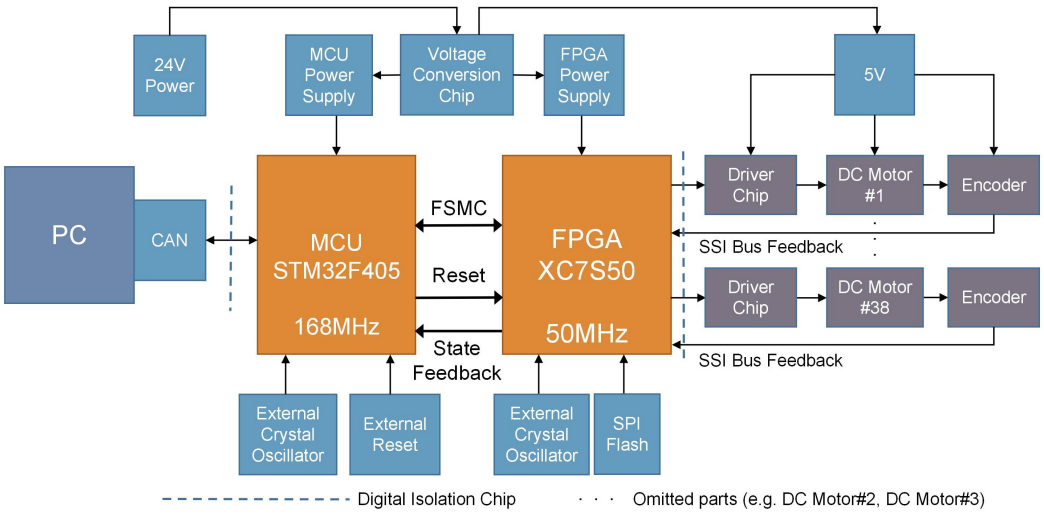

Figure 3. Hardware diagram of heterogeneous embedded platforms.

3.2. MCU Software Design

MCU is the central control core of this system. Its main task is to analyze the instructions issued by the host computer through the CAN bus and then enter the interrupt service function to achieve the corresponding function. These features include communication self-check, parameter write, parameter query, processor status query, and motor control.

The MCU first initializes peripherals and global variables after power-on. The MCU controls the power chip that supplies power to the FPGA. After the MCU initialization, the FPGA is powered on, and the program is loaded from the external Flash. After loading the FPGA program, the MCU actively sends a ready frame to the host computer and enters the loop.

When no CAN frame is received, the MCU will periodically feed the watchdog to ensure the system's normal operation. When a CAN frame is received, the MCU enters the CAN interrupt service function and performs the corresponding function according to the frame ID of the CAN frame.

The communication self-test function can detect whether the CAN communication is normal. The control parameters are stored in the Flash of MCU, and the parameter writing and parameter query functions are the assignment and query reply to these control parameters, respectively. The motor control function is the most important function of the system. When performing this function, the MCU analyzes the data field of the CAN frame, calculates each motor's rotation angle, and finally sends the information to the FPGA, which controls each motor's rotation. After waiting for a certain period, the MCU counts the rotation status of each motor and replies to the host computer. The processor status query function queries the MCU temperature and voltage and then replies to the host computer. The temperature and voltage data can be used to measure whether the system is working normally.

The workflow and function schematic of the MCU are shown in Figure 4.

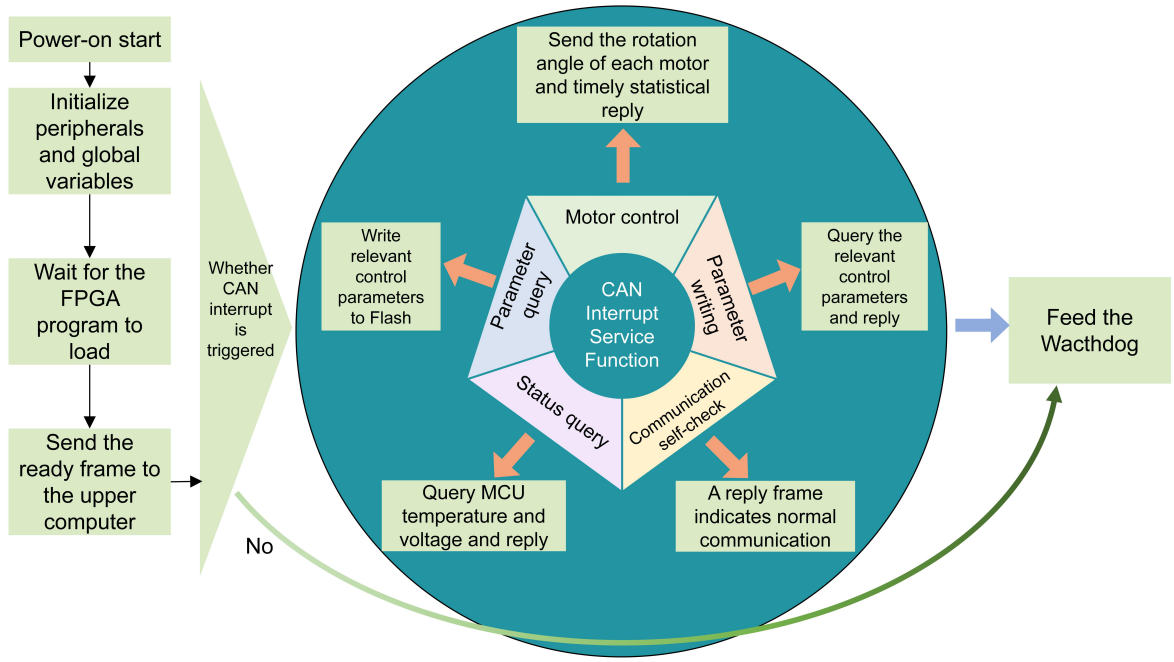

Figure 4. The workflow and function diagram of MCU.

The FSMC bus is used by the MCU to communicate with the FPGA. The address data multiplexing mode is used to save MCU pin usage, and the asynchronous mode is selected for transmission. The read-and-write sequence diagram of FSMC in multiplexing mode is shown in Figure 5.

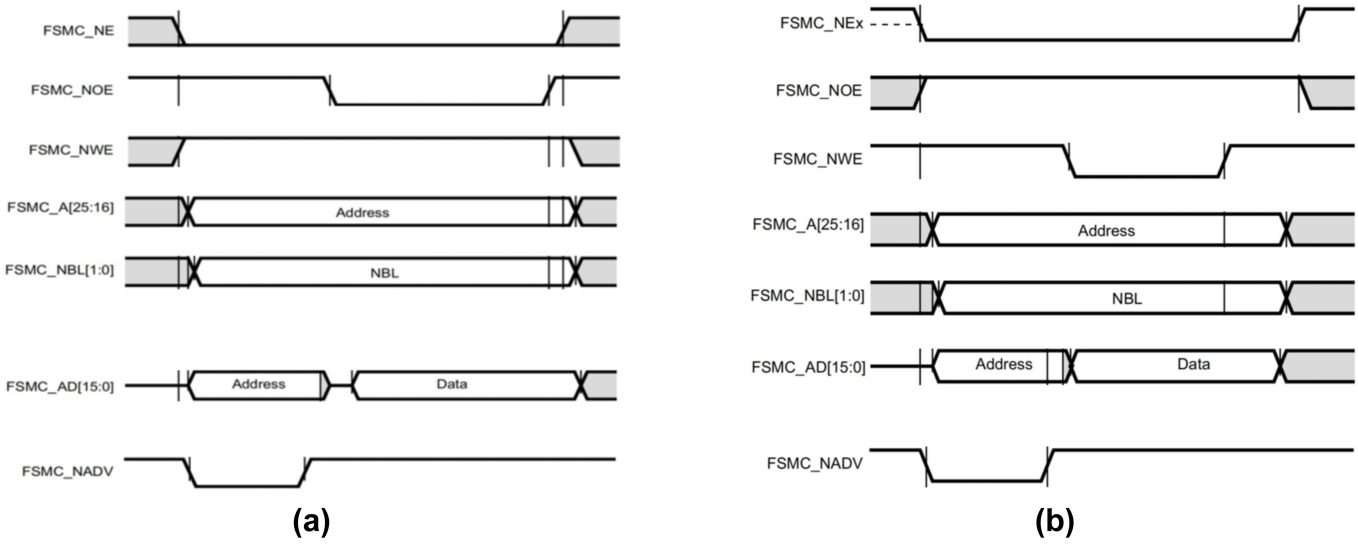

Figure 5. Asynchronous multiplexed waveforms of FSMC BUS. (**a**) read. (**b**) write.

In Figure 5, FSMC_NE is the slice selection signal, FSMC_NOE is the read enable signal, FSMC_NWE is the write enable signal, FSMC_A[25:16] is the high 10 bits of the address line, and FSMC_NBL[1:0] is the byte selection signal. FSMC_AD[15:0] is the multiplex bus for address data, and FSMC_NADV is the address latch signal.

3.3. Design of Micro-DC Motor Control IP Core in FPGA

A single micro DC motor control IP core is designed according to the top–down design idea. It includes the FSMC interface module, PID algorithm module, synchronous serial interface (SSI) module, PWM module, and so on. By enabling the gated clock in the FSMC

interface module, other modules can be effectively controlled, which is more conducive to the internal timing constraints of FPGA. Finally, the IP cores are instantiated and reused at the top layer to construct a micro-DC motor multi-axis servo control co-processing unit based on FPGA. The FPGA design framework of single motor control IP core is shown in Figure 6, and the working flow chart of multi-motor control is shown in Figure 7. The TOP-Interface accesses individual cores through different register address spaces.

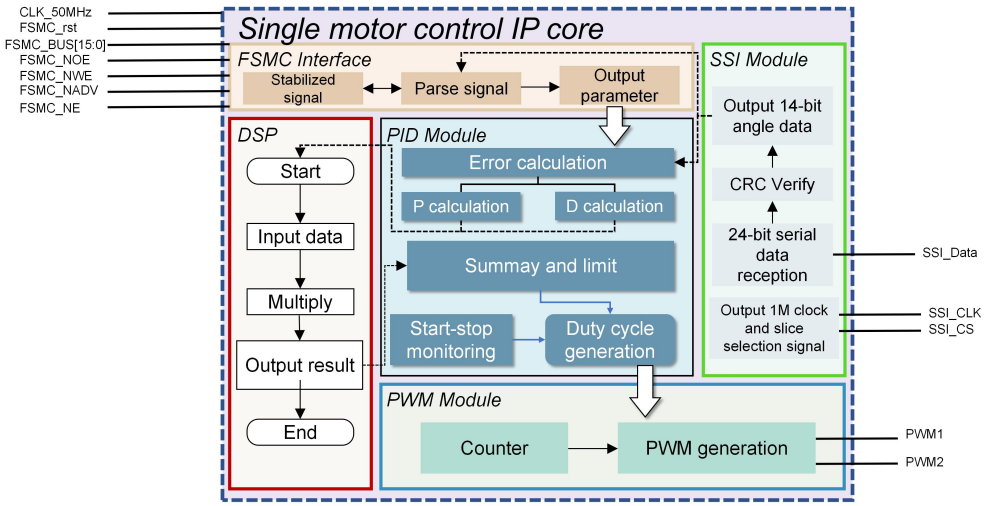

Figure 6. FPGA design framework of single motor control IP core.

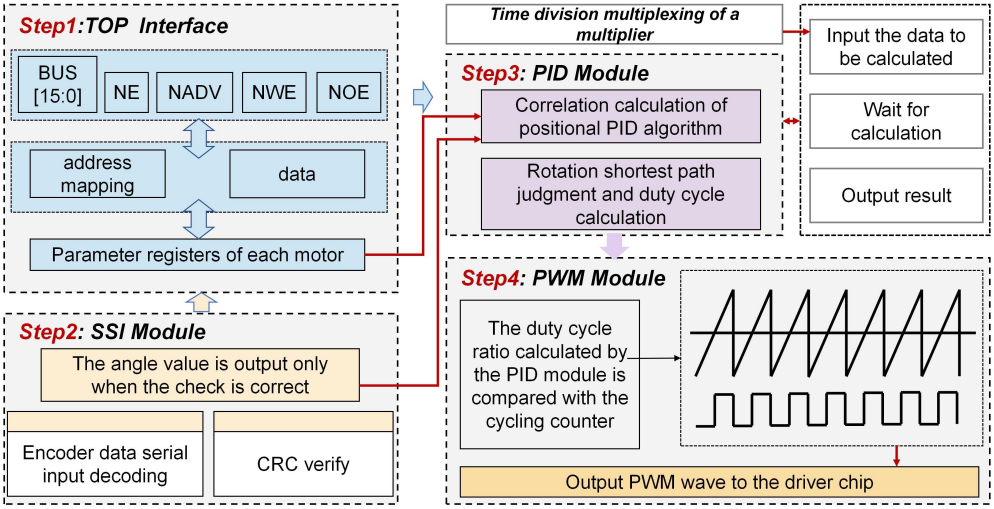

Figure 7. The workflow diagram of IP core.

3.3.1. Fsmc Interface Module Design

The data interaction between MCU and FPGA is an asynchronous transfer across the clock domain. The clock rate of the FPGA is lower than that of the MCU, so data processing must be performed on the FPGA side to prevent metastability. Specifically, the 16-bit bus is delayed by two beats, and then the address latch signal and read/write enable are edge triggered to prevent burrs. The input of the interface module is the relevant signal of FSMC, and the output is some control registers and parameter registers. For the MCU, each register has its unique address so that the motor control of each axis can be realized according to need. Table 1 gives the address correspondence of the registers.

Table 1. Registers address mapping table.

Register Name	Offset	Bit Wide	Read/Write	Explanation
SSI_reg	0x1	16	read	Feedback from encoder
FPWM_reg	0x2	16	write	Number of fixed duty
P_reg	0x3	16	write	Register of P paramter
D_reg	0x4	16	write	Register of D paramter
Pos-set_reg	0x5	16	write	Reference of rotation angle
Control_reg	0x6	16	write	Bit0:reset Bit1:enable of closed loop mode Bit2:brake Bit3:enable of fixed-duty mode Bit4:direction of fixed-duty mode

The ILA acquisition signal graph presented in Figure 8 shows the FSMC interface module's actual waveforms. A write operation and a read operation are shown here. The value 30,000 is written to the D parameter register at address 4, and this will assign the D parameter a value of 30,000. The encoder feedback register SSI_reg at address 1 is read, and the encoder feedback value read is 1855, indicating that the motor is currently in the encoder reading 1855 position. The stability of the asynchronous transmission is achieved by delaying the signals including FSMC_NE, FSMC_NOE, FSMC_NWE, FSMC_NADV, and FSMC_BUS by two beats. The waveform of the interface module corresponds to the timing diagram shown in Figure 5. After the transmission is completed, the chip selection signal FSMC_NE pulls up.

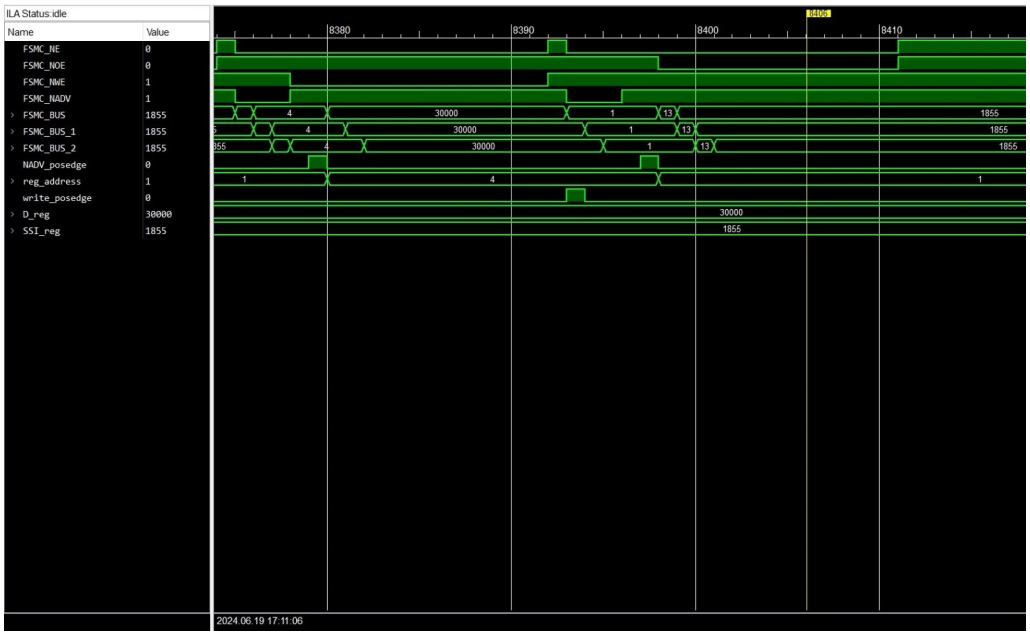

Figure 8. The FSMC interface waveform captured by the embedded logic analyzer (ILA).

3.3.2. PID Module and Multiplicative Time Division Multiplexing Design

This design uses the positional digital PID algorithm. According to Equation (4), the positional digital PID algorithm is prone to integral saturation when the error between the current and target positions is relatively large. The nonlinear PD control can realize faster and more accurate positioning of the servo control mechanism [33]. In this design, PD control is used to avoid integral saturation, and the output saturation limit is carried out to prevent the duty ratio passed to the PWM module from being too large.

According to Equation (4), there are two multiplication calculations. When the FPGA hardware implements a multi-axis motor control algorithm, more multipliers are needed. The frequency of the position loop is 10 kHz, which has sufficient time for pipeline computation. In order to save hardware resources, the multiplier is used in a time-division multiplexing way. Each motor IP core only needs to use one multiplier to complete the PID algorithm calculation.

The specific operation flow of the multiplier time-division multiplexing is shown in Figure 9. A multiplication enable is generated before each multiplication calculation, and the corresponding multiplier and multiplicand are simultaneously inputted to the FPGA hardware multiplier. The multiplier calculation requires a certain clock cycle, and the delay is set to ensure the completion of the multiplication calculation. After the calculation, the multiplier returns the result, and the enable is turned off to wait for the next multiplication operation.

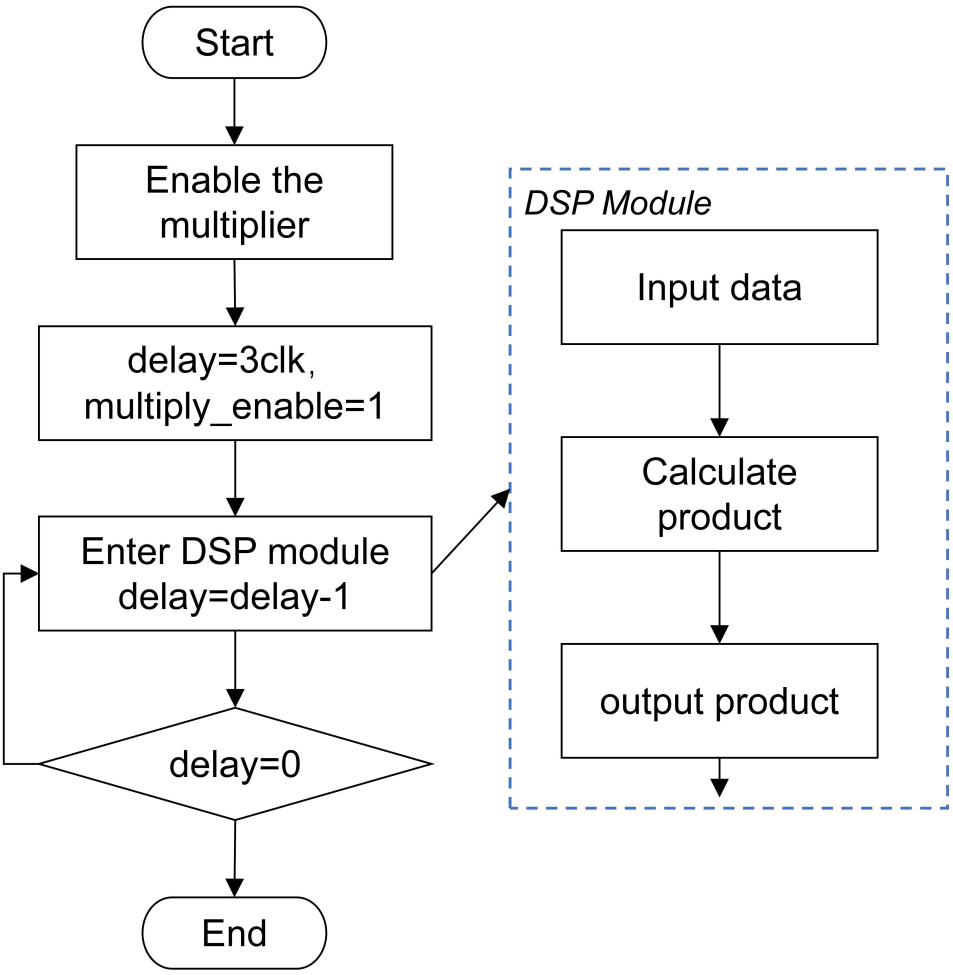

Figure 9. Flow chart of time division multiplexing of the multiplier.

The absence of the integral calculation may lead to static errors. When the motor rotates at a small angle or approaches the target position, the error is small, and the output duty ratio may be small, resulting in insufficient motor starting torque or early stopping. In order to improve the positioning accuracy of the motor servo control system, the motor start–stop monitoring mode is designed. Whether the motor is in place is determined by recording the error value for four consecutive times. If all the values are 0, the output duty cycle is 0, and at the same time, the motor does not rotate. If all four error values are equal and not 0, the duty cycle ratio is amplified several times so that the motor has

enough starting torque. In order to ensure that the motor can rotate in the shortest path and improve the response time, the rotation direction judgment is performed.

The block diagram of the structure of the positional digital PID module is shown in Figure 10. The input set angle and target angle are both 14-bit unsigned numbers, and the error generated by their computation is a signed number with a bit width of 15 bits, which is negative when the highest bit is 1 and positive when the highest bit is 0. Since the data transmission width of FSMC is 16 bits, the bit widths of the P and D parameters are set to 16-bit unsigned numbers. The entire module is controlled by the enable transmitted by the FSMC interface module, and the calculation is performed when the enable is detected. When the enable is not detected, the registers will all be cleared to avoid power consumption from bit flipping. The entire position loop algorithm is implemented by a finite state machine with a frequency of 10 kHz. The state-hopping schematic of the finite state machine is shown in Figure 11.

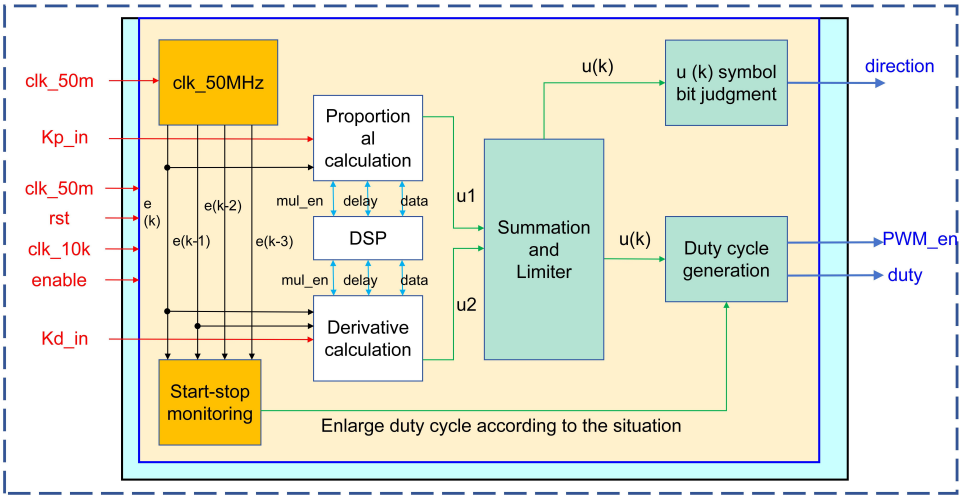

Figure 10. The structure block diagram of positional digital PID algorithm module.

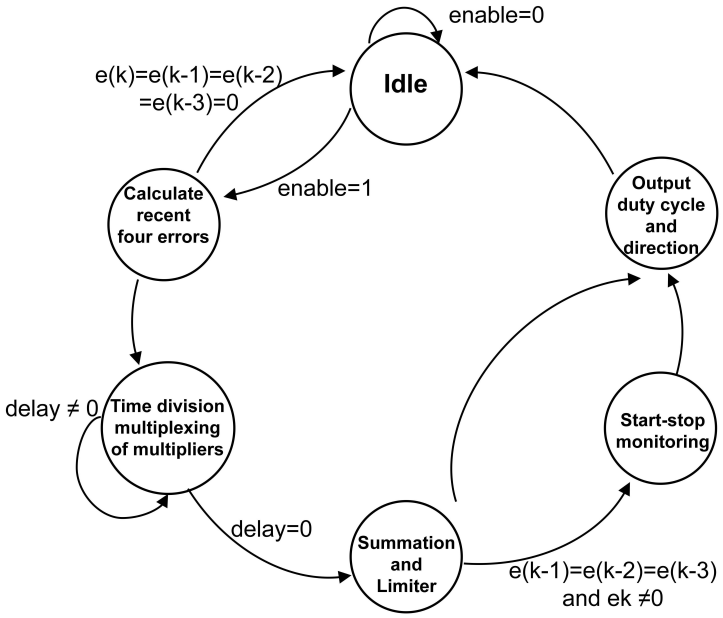

Figure 11. State machine diagram of positional digital PID algorithm module.

3.3.3. SSI Module Design and Cyclic Redundancy Check (CRC) Formula Method

In this system, the MT6701 absolute magnetic encoder is used to collect the position information of the motor. Compared with the incremental encoder, the absolute encoder can eliminate the zeroing operation and is more convenient for the motor position servo control. The communication protocol of the absolute magnetic encoder used is the SSI protocol, which contains two input signals, clock and chip select, and one data output signal. The data output of the magnetic encoder is 24-bit serial data, consisting of 14-bit absolute angle data, 4-bit magnetic field status data, and 6-bit CRC check code generated from the first 18 bits of data. The angle data have 14 bits and can reach an accuracy of 0.022°, guaranteeing precise control of the position servo. The absolute angle of 0–360° is calculated according to Equation (5):

$$\theta = \frac{D}{16384} \cdot 360^{\circ} \tag{5}$$

where D corresponds to the 14-bit angle data of the encoder.

Bit errors may occur during the transmission process due to electromagnetic interference and other factors. At the FPGA receiving side, we will carry out a CRC verification comparison to ensure the correctness of the angle. The generation process of CRC is often a modular 2 algorithm for binary polynomials. A corresponding generating polynomial G is generated for a given data M [34]. The generated polynomial is shown in Equation (6):

$$G(x) = \sum_{0}^{g-1} g_i x^i \tag{6}$$

where g corresponds to the highest degree of the polynomial, g_i is the coefficient, g_0 takes the value 1, and other coefficients can be 0 or 1. The polynomial can be represented by a linear feedback shift register (LFSR). The schematic diagram of Galois LFSR is shown in Figure 12.

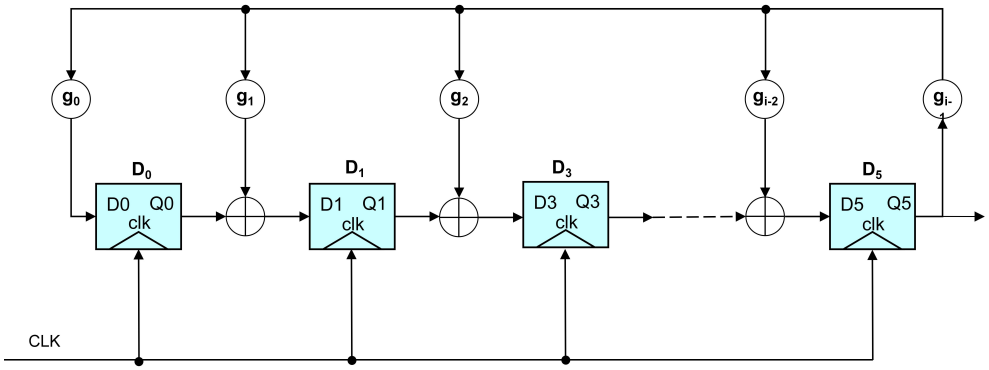

Figure 12. Galois LFSR.

The LFSR consists of D flip-flops (which can also be seen as registers), XOR gates, and feedback branches. The individual coefficients from g_i to g_0 determine whether this feedback branch exists, with a 1 indicating it exists and a 0 indicating that it does not. The register needs to be assigned an initial value where all the initial values are 0. Figure 13 is a depiction of serial data, with each bit of data requiring one clock cycle. According to the idea of parallel computing, the original data are input in parallel into the branch with coefficient 1 to increase the rate. The data are input from the input side by bit from high to low. According to the data provided by the chip manual, the CRC-generated polynomial adopted by MT6701 is $X^6 + X + 1$, and the corresponding polynomial is 1000011. The digital circuit model of its modulo 2 operation is shown in Figure 13.

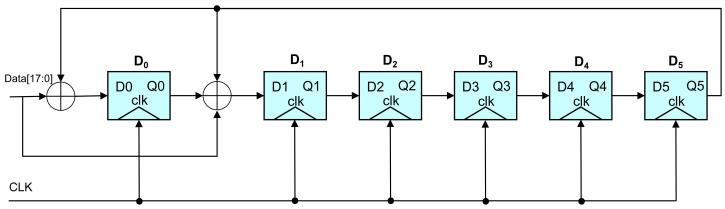

Figure 13. Parallel processing computation LFSR corresponding to $X^6 + X + 1$ polynomials.

After all the data are entered and the shift calculation is completed, a six-digit CRC check code is obtained. The initial value of CRC[5:0] is 0. According to the XOR algorithm, the result of any XOR calculation with 0 is the original value. If the input 18-bit data are d, the corresponding CRC code is simplified as Equation (7):

$$\begin{aligned}
CRC[0] &= d[17] \oplus d[16] \oplus d[15] \oplus d[12] \oplus d[10] \oplus d[6] \oplus d[5] \oplus d[0] \\
CRC[1] &= d[15] \oplus d[13] \oplus d[12] \oplus d[11] \oplus d[10] \oplus d[7] \oplus d[5] \oplus d[1] \oplus d[0] \\
CRC[2] &= d[16] \oplus d[14] \oplus d[13] \oplus d[12] \oplus d[11] \oplus d[8] \oplus d[6] \oplus d[2] \oplus d[1] \\
CRC[3] &= d[17] \oplus d[15] \oplus d[14] \oplus d[13] \oplus d[12] \oplus d[9] \oplus d[7] \oplus d[3] \oplus d[2] \\
CRC[4] &= d[16] \oplus d[15] \oplus d[14] \oplus d[13] \oplus d[10] \oplus d[8] \oplus d[4] \oplus d[3] \\
CRC[5] &= d[17] \oplus d[16] \oplus d[15] \oplus d[14] \oplus d[11] \oplus d[9] \oplus d[5] \oplus d[4]
\end{aligned} \tag{7}$$

The formula method can perform the calculation quickly, avoiding the consumption of hardware storage resources by the look-up table (LUT) method. The CRC codes calculated by Equation (7) are compared with the received CRC codes. If the CRC codes are equal, the transmission is considered normal. In the case of an incorrect transmission, the angle register on the FPGA side will not be updated. However, it will retain the correct value from the previous cycle until the CRC checksum is correct for the next angle transmission cycle. The frequency of the motor angle update needs to be faster than the control cycle of the position loop in order to reduce the power consumption caused by clock reversal. The input clock of the encoder is set to 1 MHz, and the angle update frequency is finally 20 kHz. The design diagram of the SSI module is shown in Figure 14.

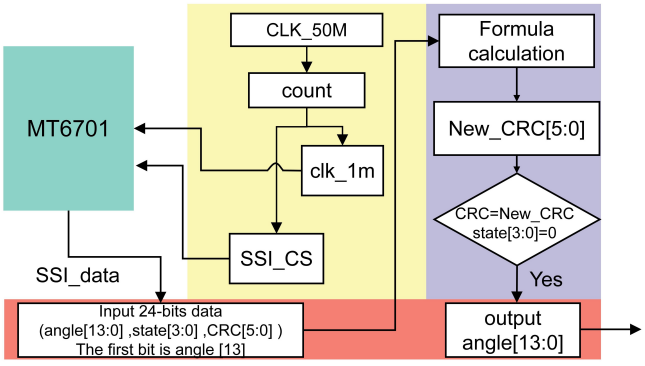

Figure 14. The design diagram of the SSI module.

3.3.4. PWM Module Design

The function of the PWM module is to adjust the pulse signal width in real time according to the duty cycle value calculated by the PID algorithm module. The PWM module outputs two PWM signals directly connected to the general-purpose H-bridge motor driver chip DRV8848. Under current hysteresis control, the micro-DC motor with a hollow cup structure will have current discontinuity and torque fluctuation. Increasing the frequency of the pulse signal can improve this problem, combined with the maximum output PWM signal frequency of the driver chip; the PWM modulation frequency is set to 195.31 kHz.

The clock of the IP core is 50 MHz; a full 8-bit counter which counts cyclically between 0 and 255 is used. When the duty cycle ratio calculated by the PID module is greater than the count value, the PWM signal is high, and vice versa. Then, the PWM1 and PWM2 are determined according to the direction signal transmitted from the PID module, and finally, these two signals are output to the motor driver chip. The higher the duty cycle, the faster the motor will rotate.

4. Experiment and Analysis of Experimental Results

Based on the software scheme of the heterogeneous embedded platform proposed in the previous section and the design of the motor control IP core, we constructed a multi-motor servo control system and performed experimental tests.

Section 4.1 introduces the constructed experimental platform. Section 4.2 shows the FPGA resource consumption of the designed motor control IP core. Section 4.3 shows the results of system function and performance testing, including the servo control performance of the single-axis motor and the synchronization and environmental adaptability of multi-axis motor control. The response time of the commands is also tested, which has an impact on the real-time performance of the system.

4.1. Construction of Experimental Platform

The experimental object selected in this experiment is a micro DC motor with a magnetic ring at the back end of the motor. The motor encoder is placed behind the magnetic ring, and the encoder interface is distributed on a foldable flexible board. The foldable flexible board is connected to the driver board through a connector. The driver board adopts the DRV8848, which can control two micro DC motors. The driver board contains the CAN bus communication circuit and the chip's download and burn interface.

The core board is mainly composed of MCU, FPGA, and an external Flash chip. The Flash chip is used to store the FPGA code and configure the FPGA each time the system is powered on and initialized. The core board is connected to the drive board through three high-speed connectors to ensure signal transmission. The power supply of the entire system is guaranteed by the power supply board. The power supply of the system is 24 V, which is converted to 5 V through the power conversion circuit of the power board to supply power to the subsequent core board and drive board.

We designed the overall scheme in the early stage, completed the selection of components, and then the circuit board was designed and realized by engineers in our laboratory. Circuit board welding was completed by the relevant manufacturers. We completed the assembly and construction of the system. The FPGA and MCU codes were compiled and burned by the Vivado 2019.1 tool and Keil uVersion5 software, respectively. The overall system is shown in Figures 15 and 16.

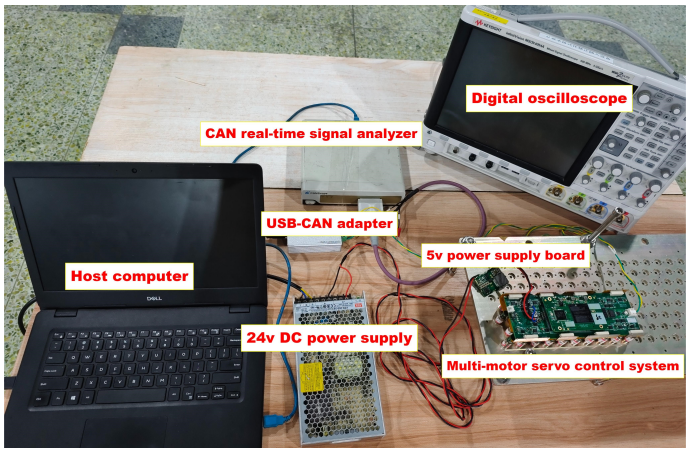

Figure 15. Diagram of the experimental platform.

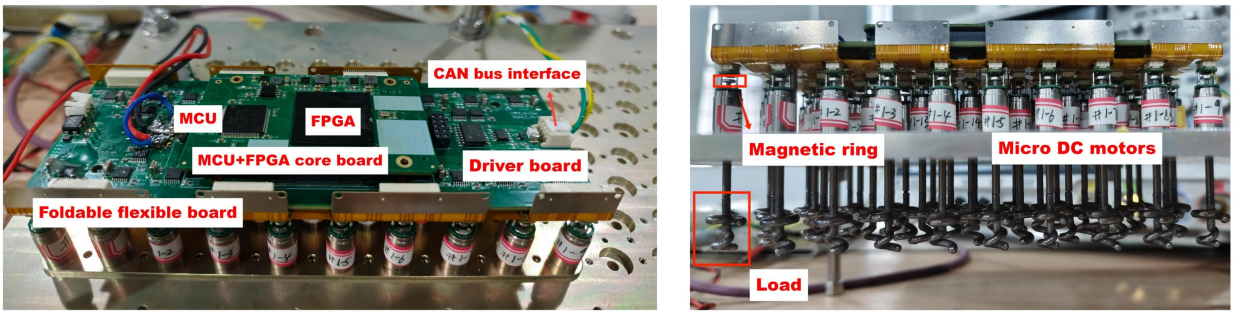

Figure 16. Diagram of the experimental platform. (**a**) Diagram of the heterogeneous embedded platform. (**b**) Diagram of the micro DC motors.

4.2. Resource Consumption

The project was compiled by Vivado 2019.1 tool and the relevant data were obtained. Tables 2 and 3 show the resource consumption for single-axis motor control and 38-axis motor control, respectively. According to the data in Table 2, the single-axis motor control consumed 453 LUTs, 16 LUTRAMs, 632 triggers, and 1 DSP. By multiplexing a single IP core, more motors can be controlled. According to the data in Table 3, the 38-axis motor control consumed 15,562 LUTs, 32 LUTRAMs, 18,823 triggers, and 38 DSPs. The designed motor control IP core is portable, and a larger capacity FPGA can be selected to realize the control of more axes of motors as required.

Table 2. Table of logical resources consumed for single-axis motor control.

Resource	Estimation	Total Available	Utilization (%)
LUT	453	32,600	1.39
LUTRAM	16	9600	0.17
FF	632	65,200	0.97
DSP	1	120	0.83
IO	27	250	10.80
BUFG	1	32	3.13

Table 3. Table of logical resources consumed for 38-axis motor control.

Resource	Estimation	Total Available	Utilization (%)
LUT	15,562	32,600	47.74
LUTRAM	32	9600	0.33
FF	18,823	65,200	28.87
DSP	38	120	31.67
IO	212	250	84.80
BUFG	1	32	3.13

The number of pins limits the number of controllable motors. Five pins are required per axis for motor control, and twenty pins are required for the FSMC bus, in addition to a clock pin and a reset pin. This means that the XC7S50 FPGA we used can control up to 45-axis micro DC motors synchronously. We present the compilation results for controlling 75-axis motors in the XC7S75-FGGA676 FPGA, as shown in Table 4. The 75-axis motor control consumed 31,673 LUTs, 32 LUTRAMs, 24,607 triggers, and 75 DSPs of the XC7S75-FGGA676 FPGA.

Table 4. Table of logical resources consumed for 75-axis motor control in XC7S75-FGGA676 FPGA.

Resource	Estimation	Total Available	Utilization (%)
LUT	31,673	48,000	65.99
LUTRAM	32	17,600	0.18
FF	24,607	96,000	25.63
DSP	75	140	53.57
IO	397	400	99.25
BUFG	1	32	3.13

Below, we compare the implementation of multi-axis motor control in the MCU + FPGA heterogeneous embedded platform and SoC FPGA based on the IP core designed in this paper, as shown in Table 5. We especially chose XC7Z020 SoC FPGA mentioned in [25] to compare with XC7S50 FPGA used in this paper. Although the number of pins of the two is different, the scheme adopted in this paper is more advantageous in terms of the maximum number of controllable motors. After that, we compared the XC7S75 FPGA with the XC7Z020 SoC FPGA, which has the same number of pins, and it is clear that the XC7Z020 SoC FPGA is too expensive. By comparing the maximum number of controllable motors with the chip price, we can find that the heterogeneous embedded platform of MCU + FPGA is more cost effective in the face of motor control requirements up to tens of axes.

Table 5. Comparison of between MCU + FPGA and SoC FPGA.

Platform	MCU + FPGA (XC7S50)	SoC FPGA XC7Z020 [25]	MCU + FPGA (XC7S75)	SoC FPGA XC7Z100
Package of FPGA	FGGA-484	CLG-484	FGGA-676	FFG-1156
Total available pins	250	200 (PL side)	400	400 (PL side)
Total available logic cells	52,160	85,120	76,800	444,000
Maximum number of controllable motors (use the IP core designed in this paper)	45	40	75	80
Commercial price (USD) *	13 + 88 = 101	194	13 + 149 = 162	4044

* Information obtained from: http://www.xilinx.com and https://www.mouser.cn/ (accessed on 25 June 2024).

4.3. System Function Verification and Performance Analysis

After receiving the command from the host computer, the multi-motor servo system will execute the calculation and start the rotation of the multi-axis motors until each axis motor reaches the target position. In this regard, we focused on testing the response time and accuracy of a single motor rotating at different angles, the consistency of multiple motors rotating at the same angle, the response time of commands, and the environmental adaptability of the system. It should be noted that in the relevant test of motor rotation, we used the MCU timer to collect the real-time position of each motor at a period of 1 ms (in the multi-axis test, due to the seriality of data transmission, the motors rotate and collect data sequentially, and then start the next motor's rotation after the previous one is completed), and then we used the debug function of Keil uVersion5 software to read out the relevant data and collate them. According to the waveform diagram shown in Figure 8, the time for FSMC bus to complete a read operation is 380 ns, which can be ignored compared with the 1 ms acquisition cycle (380 ns/1 ms = 0.038%). As for the real-time test, we used a CAN real-time signal analyzer to complete it.

4.3.1. Single Axis Servo Control Performance Experiment

First, the performance of a single motor rotating at different angles is tested. With the target position as a gradient of 10°, the motor forward and reverse tests are carried out, respectively. The results show that the response time of the micro DC motor from the start

of rotation to completion is less than 28 ms, and the trajectory remains relatively smooth without significant overshoot. The response times of the motors for forward and reverse rotations indicate that there is very little difference in the forward and reverse performance of the motors. The steady-state error is always less than 1.5°, which belongs to the normal fluctuation caused by the algorithm modulation and also shows that the position control is accurate. The results of the motor performing forward and reverse rotation tests are shown in Figures 17 and 18.

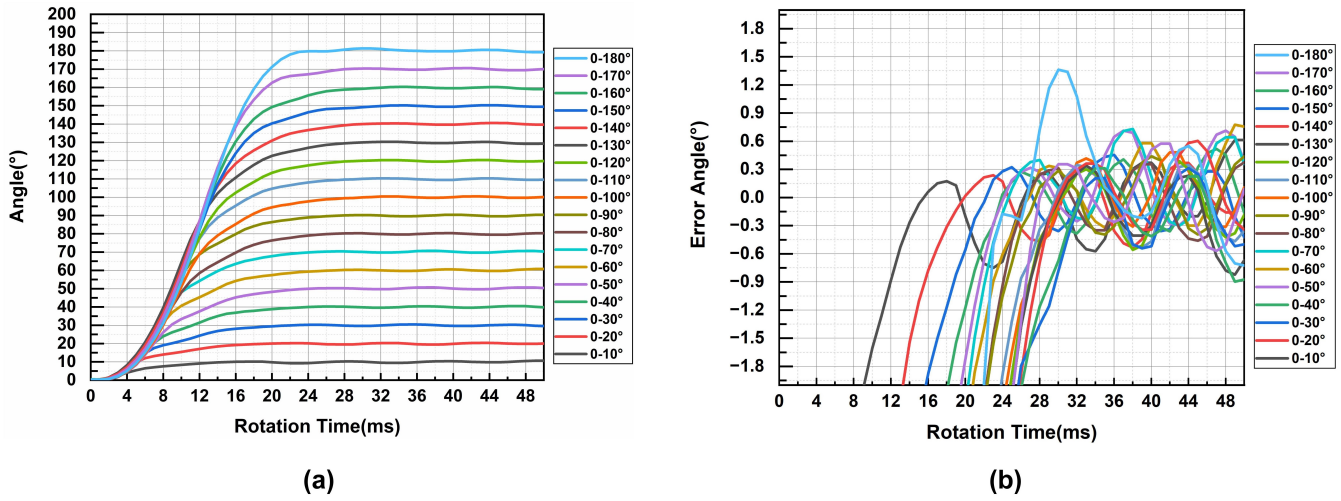

Figure 17. Test results of different angles of motor forward rotation. (**a**) Curve of rotation trajectory. (**b**) The corresponding error angle.

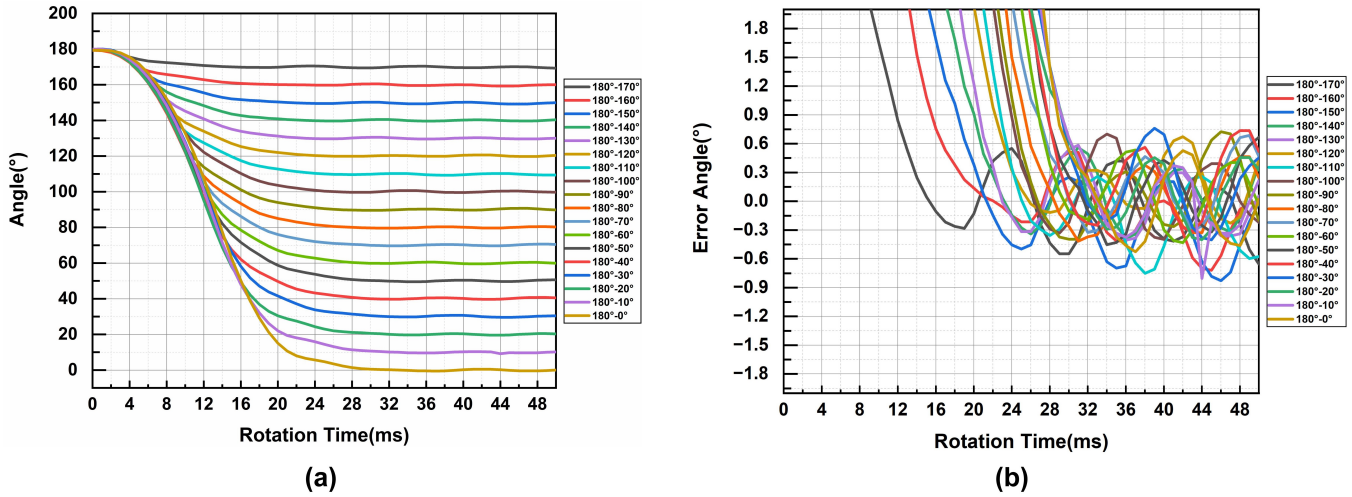

Figure 18. Test results of different angles of motor reverse rotation. (**a**) Curve of rotation trajectory. (**b**) The corresponding error angle.

The response performance of the motor to position mutation is also tested as shown in Figure 19. Five sets of experimental tests are carried out on the same motor. First, the motor is controlled to 45°, and then the motor is ordered to 135° at 30 ms. In the process of position mutation, the micro DC motor still has a fast response and high rotation accuracy, and the steady-state error does not exceed 1°.

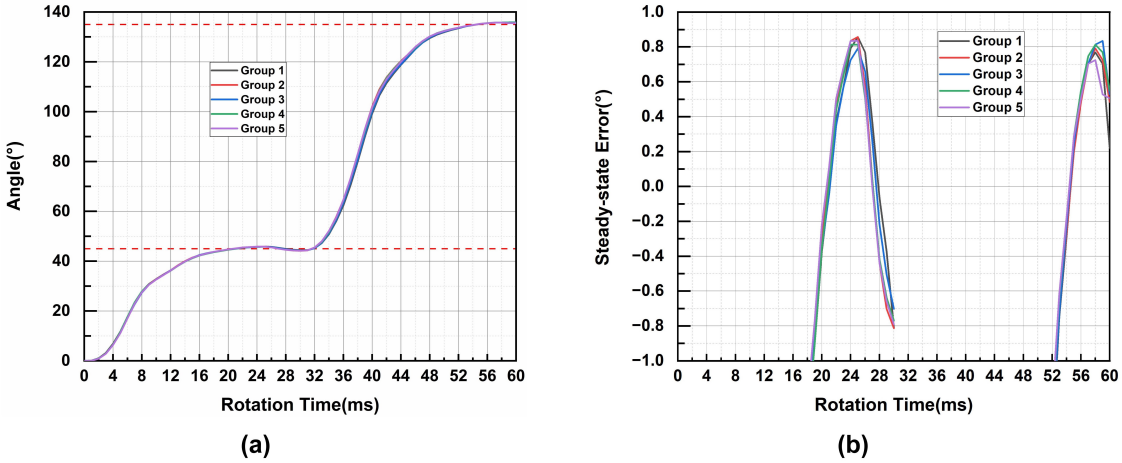

Figure 19. Position mutation rotation test results. (**a**) Curve of rotation trajectory. (**b**) The corresponding error angle.

4.3.2. Multi-Axis Servo Control Performance Experiment

For the target positions for the experiments of 38-axis motor rotation, 45°, 90°, 135°, and 180 ° were selected. The average value of the running trajectory of the 38-axis motors was used as the reference position curve, and the process error of each axis motor and the steady state error after reaching the target position were analyzed. And they are presented in Table 6.

Table 6. The summary of the maximum process error and the maximum steady-state error.

Rotation Angle (°)	Maximum Process Error (°)	Maximum Steady-State Error (°)
45	2.2	1.1
90	4.67	1
135	8.75	1.21
180	9	1

The process control of the motors in each axis shows a significant deviation and increases with the rotation angle, which is due to the otherness between the motors and the nonlinearity of the algorithms used. The dual-loop control structure focuses mainly on the fast control of position, so it is not sufficient for speed control. The steady-state errors are all less than 1.5°, indicating the stability of the position control. The results are shown in Figures 20–22.

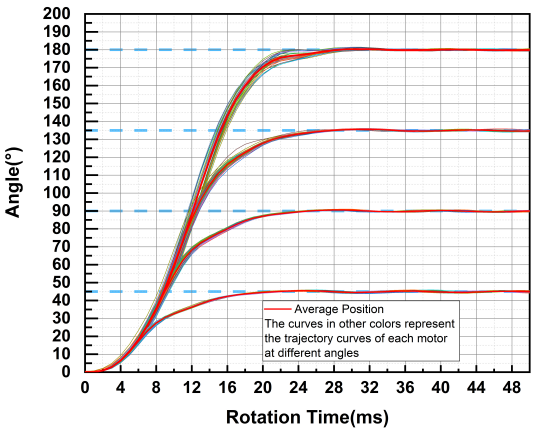

Figure 20. Trajectory curves for four angles of rotation of 38-axis motors.

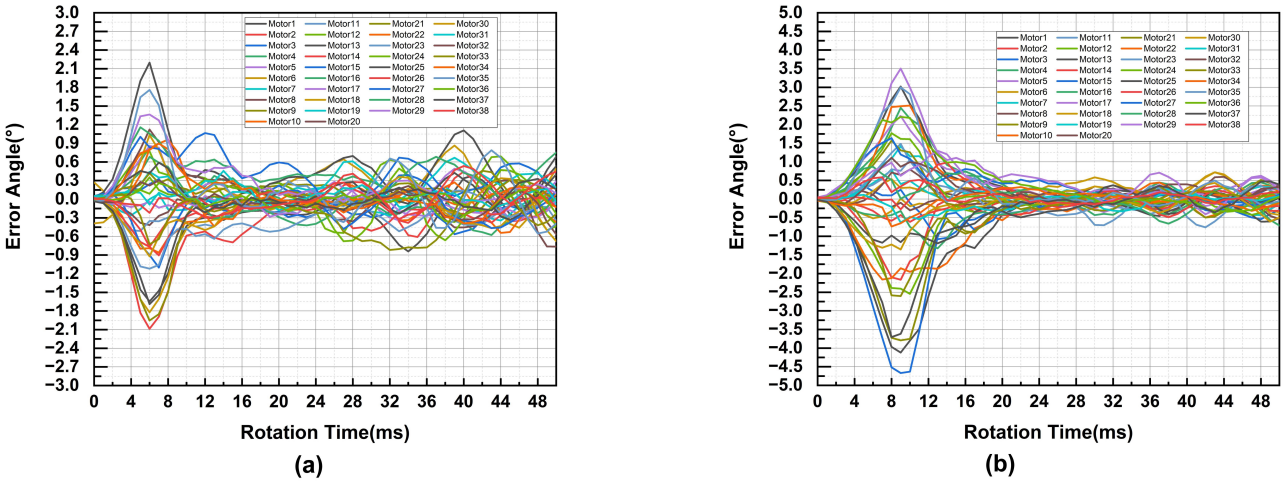

Figure 21. Error angle curves for 45° and 90° of rotation of the 38-axis motor.(**a**) Error angle at 45° rotation. (**b**) Error angle at 90° rotation.

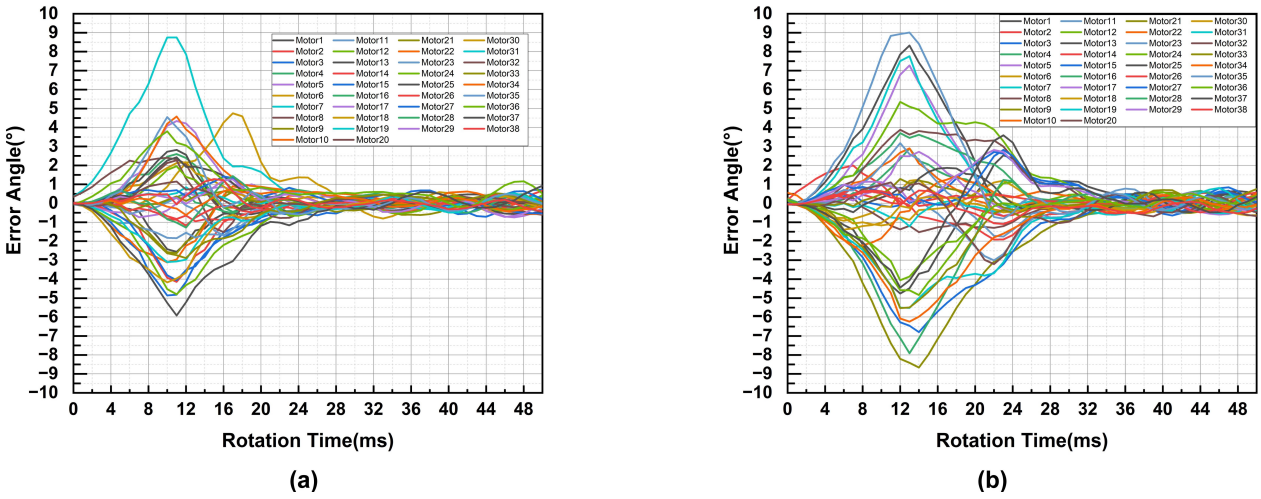

Figure 22. Error angle curves for 135° and 180° of rotation of the 38-axis motor. (**a**) Error angle at 135° rotation. (**b**) Error angle at 180° rotation.

4.3.3. System Environment Adaptability Test

To further validate the developed system's industrial applicability, robustness tests were carried out to evaluate its performance at high and low temperatures. We used a high- and low-temperature test chamber, which is shown the Figure 23, and we communicated with the system through a hole in the side of the chamber. With temperatures of 85 °C and −20 °C, a 180° position control command was issued to the system, and the rotation result of the 38-axis motor was obtained. The running track at ordinary temperature was taken as the reference position curve, and the performance variation of the system at high and low temperatures was analyzed. At high and low temperatures, the motor's performance decreased slightly, but it was still able to reach 180° within 30 ms. The maximum process deviation angles were 4.65° at 85 °C and 4.17° at −20 °C. The specific results are shown in Figure 24.

Figure 23. The high- and low-temperature test chamber.

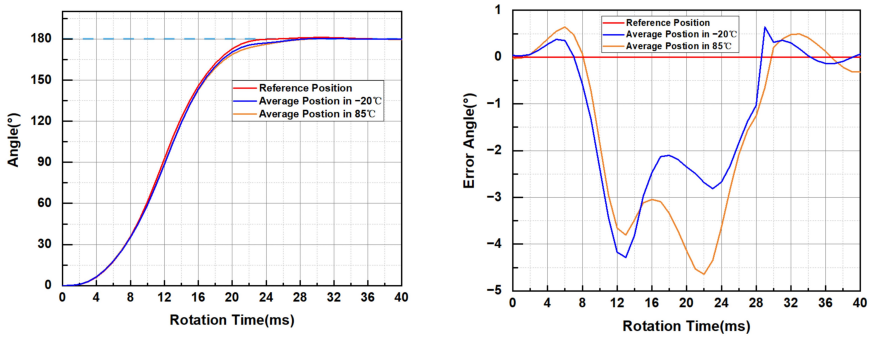

Figure 24. Motor rotation curve at high and low temperatures. (**a**) Motor rotation curve. (**b**) Error angle curve.

4.3.4. Command Response Time Test

The real-time performance of a multi-motor servo control system depends on the fast parsing and computation of commands and the fast response of the motors. The overall data routing is shown in the Figure 25, where the motor rotation time was shown in the previous subsections.

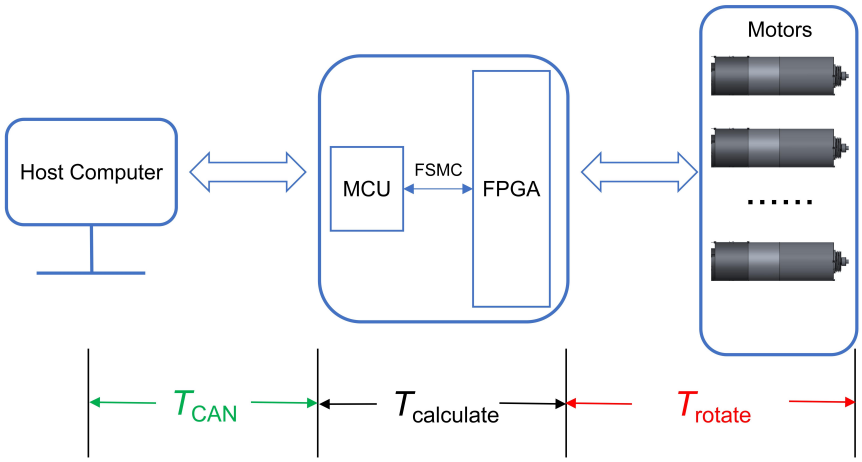

Figure 25. Data routing of the system.

For the communication self-test function, the status reply function, and the parameter query function, 20 repetitions of the experiments were conducted using a CAN real-time signal analyzer to test the specific performance of the system's CAN communication, respectively. The results show that there is a slight jitter in the response time of the commands for different functions. The average response time for the communication

self-test is 193.2 µs, for the status reply, it is 291.3 µs, and for the parameter query, it is 237.2 µs.

The motor control function is the most important, and the MCU will reply to the host computer after 30 ms (waiting for the motor rotation to be completed and counted) after receiving this command. Therefore, starting with the system receiving the CAN frame signal and ending with the No. 38 motor receiving the PWM pulse, the calculation time of the heterogeneous embedded platform is tested by the oscilloscope. The results show that the average time of the motor control function response is 594.8 µs. The maximum value is 662.8 µs, and the minimum value is 500 µs. The time fluctuation is the difference caused by the computation time based on the CAN frame resolution angle. Two boxplots are plotted based on the test results as shown in Figure 26.

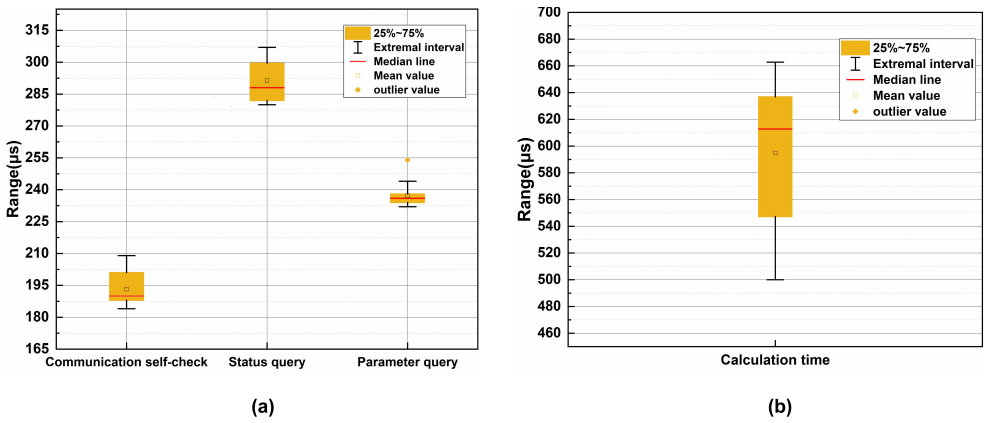

Figure 26. Boxplot of experimental data. (**a**) Response time of three functions. (**b**) Calculation time.

5. Conclusions

In conclusion, we propose a multi-motor servo control scheme based on heterogeneous embedded platforms. The fast and accurate control of 38-axis servo motors is realized through the IP core design and the coordinated work of hardware and software. The experimental results demonstrate the real-time and industrial applicability of the system, including the following. (a) The designed IP core is well scalable to realize the position servo control of micro DC motors. (b) The servo control of multi-axis motors is independent of each other and is fast and stable, which can complete the rapid analysis and real-time response to the commands of the host computer. (c) The system is reliable and performs well in high- and low-temperature environments.

In this research, we developed a physical platform with MCU + FPGA as the core. It has the scalability and flexibility to increase the number of axes of the motor being controlled as required. The MCU + FPGA heterogeneous embedded platform solution proposed by us has cost-effective performance in the face of motor control of tens of axes. The XC7S50 FPGA we used can control up to 45-axis micro DC motors synchronously by reusing the designed motor IP core in this paper, and a XC7S75 FPGA with 400 pins can control up to 75-axis motors synchronously. Moreover, the designed IP core has low consumption of logical resources, which provides the possibility for further optimization of the motor control algorithm.

Author Contributions: Conceptualization, M.G. and B.W.; methodology, B.W.; software, M.G.; validation, M.G. and X.Z.; formal analysis, B.W.; investigation, M.G.; resources, M.G.; data curation, M.G. and X.Z.; writing—original draft preparation, M.G.; writing—review and editing, M.G. and B.W.; visualization, M.G.; supervision, B.W.; project administration, M.G. and B.W.; funding acquisition, B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2682024GF013.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The DC motor's parameters are shown in Table A1.

Table A1. DC motor's parameters.

Parameter	Magnitude
Armature resistance	12.5 Ω
Armature inductance	168 μH
Back-EMF constant	4.39 mV · s/rad
Torque coefficient	4.39 mN · m/A
Moment of inertia	0.2605 g · cm^2
Viscous friction coefficient	1.549×10^{-6} N · ms/rad
Nominal voltage	6 V

References

1. Fazdi, M.F.; Hsueh, P.W. Parameters identification of a permanent magnet dc motor: A review. *Electronics* **2023**, *12*, 2559. [CrossRef]
2. Gao, Y.; Wu, Y.; Guo, J.; Chen, Q. Adaptive control of four motor servo systems based on characteristic model and gradient projection estimator. *IEEE Access* **2020**, *8*, 192239–192247. [CrossRef]
3. Bae, S.Y.; Koh, J.S.; Jung, G.P. A miniature flapping mechanism using an origami-based spherical six-bar pattern. *Appl. Sci.* **2021**, *11*, 1515. [CrossRef]
4. Gu, Q.; Tian, J.; Yang, B.; Liu, M.; Gu, B.; Yin, Z.; Yin, L.; Zheng, W. A novel architecture of a six degrees of freedom parallel platform. *Electronics* **2023**, *12*, 1774. [CrossRef]
5. Yu, D.Y.; Ding, Z.; Tian, X.Q.; Huang, X.Y. Local corner smoothing algorithm for screw motor high-precision machining. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 2117–2127. [CrossRef]
6. Chu, X.; Nian, X.; Sun, M.; Wang, H.; Xiong, H. Robust decentralized fault estimation for loss of actuator effectiveness of multi-motor web-winding system. *Int. J. Control Autom. Syst.* **2019**, *17*, 691–704. [CrossRef]
7. Shu, X.; Guo, Y.; Yang, W.; Wei, K.; Zhu, Y.; Zou, H. A detailed reliability study of the motor system in pure electric vans by the approach of fault tree analysis. *IEEE Access* **2019**, *8*, 5295–5307. [CrossRef]
8. Bosma, S.; Van Rooijen, N.; Alonso-Delpino, M.; Spirito, M.; Llombart, N. First demonstration of dynamic high-gain beam steering with a scanning lens phased array. *IEEE J. Microwaves* **2022**, *2*, 419–428. [CrossRef]
9. Ortega-García, L.E.; Rodriguez-Sotelo, D.; Nuñez-Perez, J.C.; Sandoval-Ibarra, Y.; Perez-Pinal, F.J. DSP-HIL comparison between IM drive control strategies. *Electronics* **2021**, *10*, 921. [CrossRef]
10. Mohammed, S.A.Q.; Alsofyani, I.M.; Lee, K.B. Improved adaptive CCS-MPCC for distorted model parameters mitigation of IPMSM drives. *IEEE Trans. Ind. Electron.* **2023**, *71*, 3483–3493. [CrossRef]
11. Tran, D.M.; Kim, K.; Choi, J.Y. CLB-Based Development of BiSS-C Interface Master for Motor Encoders. *Electronics* **2023**, *12*, 886. [CrossRef]
12. Feng, X.; Zhuo, Q.; Liu, X.; Qian, Y.; Li, Y. Development of multi-motor synchronous control system based on network-on-chip. *Proc. Inst. Mech. Eng. Part J. Syst. Control Eng.* **2020**, *234*, 1000–1010. [CrossRef]
13. Horvat, R.; Jezernik, K.; Čurkovič, M. An event-driven approach to the current control of a BLDC motor using FPGA. *IEEE Trans. Ind. Electron.* **2013**, *61*, 3719–3726. [CrossRef]
14. Salehifar, M.; Arashloo, R.S.; Eguilaz, M.M.; Sala, V. FPGA based robust open transistor fault diagnosis and fault tolerant sliding mode control of five-phase PM motor drives. *J. Power Electron.* **2015**, *15*, 131–145. [CrossRef]
15. Schenke, M.; Haucke-Korber, B.; Wallscheid, O. Finite-set direct torque control via edge computing-assisted safe reinforcement learning for a permanent magnet synchronous motor. *IEEE Trans. Power Electron.* **2023**, *38*, 13741–13756. [CrossRef]
16. Wang, B.; Jiao, J.; Xue, Z. Implementation of continuous control set model predictive control method for PMSM on FPGA. *IEEE Access* **2023**, *11*, 12414–12425. [CrossRef]
17. Krim, S.; Gdaim, S.; Mtibaa, A.; Mimouni, M.F. Design and implementation of direct torque control based on an intelligent technique of induction motor on FPGA. *J. Electr. Eng. Technol.* **2015**, *10*, 1527–1539. [CrossRef]
18. Zhang, Z.; Wang, G.; Wang, Z.; Liu, Q.; Wang, K. Neural network based Q-MRAS method for speed estimation of linear induction motor. *Measurement* **2022**, *205*, 112203. [CrossRef]
19. Stanković, M.R.; Manojlović, S.M.; Simić, S.M.; Mitrović, S.T.; Naumović, M.B. FPGA system-level based design of multi-axis ADRC controller. *Mechatronics* **2016**, *40*, 146–155. [CrossRef]

20. Villalba, J.; Hormigo, J. High-radix formats for enhancing floating-point fpga implementations. *Circuits Syst. Signal Process.* **2022**, *41*, 1683–1703. [CrossRef]
21. Fang, C.; Mao, K.; Fang, S.; Zhao, Z.; Hua, B.; Liu, T.; Zhu, Q. Cordic-based general multiple fading generator for wireless channel digital twin. *Sensors* **2023**, *23*, 2712. [CrossRef] [PubMed]
22. Kusaka, T.; Tanaka, T. Fast and accurate approximation methods for trigonometric and arctangent calculations for low-performance computers. *Electronics* **2022**, *11*, 2285. [CrossRef]
23. Kung, Y.S.; Huang, C.C.; Tsai, M.H. FPGA realization of an adaptive fuzzy controller for PMLSM drive. *IEEE Trans. Ind. Electron.* **2009**, *56*, 2923–2932. [CrossRef]
24. Xu, F.; Peng, R.; Zhang, K.; Xu, X. 6-DoF magnetically levitated rotary table with dynamic numerical force and torque regulator. *IEEE/ASME Trans. Mechatron.* **2021**, *27*, 1345–1356. [CrossRef]
25. Gu, Q.; Li, Y.; Niu, P.; Sun, J. Optimized design of soc-based control system for multi-axis drive. *Elektron. Elektrotechnika* **2014**, *20*, 15–22. [CrossRef]
26. Sun, J.; Ren, J.; Le, Y.; Wang, H. Analysis of air-gap magnetic field and structure optimization design of hollow-cup motor. *Aerospace* **2022**, *9*, 549. [CrossRef]
27. Guo, S.; Yang, Y.; Yin, Y.; Tang, Y.; Tian, Y.; Zhai, C. The Application of Permanent Magnet Synchronous Motor with Small Electrical Time Constant in Fiber Positioner. *Res. Astron. Astrophys.* **2024**, *24*, 015017. [CrossRef]
28. Ni, Q.; Yang, M.; Odhano, S.A.; Tang, M.; Zanchetta, P.; Liu, X.; Xu, D. A new position and speed estimation scheme for position control of PMSM drives using low-resolution position sensors. *IEEE Trans. Ind. Appl.* **2019**, *55*, 3747–3758. [CrossRef]
29. Sun, Y.; Yang, M.; Wang, B.; Chen, Y.; Xu, D. Precise position control based on resonant controller and second-order sliding mode observer for PMSM-driven feed servo system. *IEEE Trans. Transp. Electrif.* **2022**, *9*, 196–209. [CrossRef]
30. Molitor, D.A.; Arne, V.; Spies, D.; Hoppe, F.; Groche, P. Task space control of ram poses of multipoint Servo Presses. *J. Process. Control* **2023**, *129*, 103057. [CrossRef]
31. Pan, J.; Zou, Y.; Cao, G.; Cheung, N.C.; Zhang, B. High-precision dual-loop position control of an asymmetric bilateral linear hybrid switched reluctance motor. *IEEE Trans. Magn.* **2015**, *51*, 1–5. [CrossRef]
32. Senthilnathan, A.; Palanivel, P. A new approach for commutation torque ripple reduction of FPGA based brushless DC motor with outgoing phase current control. *Microprocess. Microsyst.* **2020**, *75*, 103043. [CrossRef]
33. Zheng, C.; Su, Y.; Mercorelli, P. A simple nonlinear PD control for faster and high-precision positioning of servomechanisms with actuator saturation. *Mech. Syst. Signal Process.* **2019**, *121*, 215–226. [CrossRef]
34. Tran, D.; Aslam, S.; Gorius, N.; Nehmetallah, G. Parallel computation of CRC-code on an FPGA platform for high data throughput. *Electronics* **2021**, *10*, 866. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.