

Design of Dual-Core Processor Industry Robot Controller

Sun Ligong*, Zhang Jincan, Sun Xiangwen and Li Hui

School of Electrical Engineering, Henan University of Science and Technology, Luoyang, 471023, China

Abstract: The functions of the robot controller are arithmetic operations and motion control, and the common robot is the secondary robot microcontroller control system of industrial PC+ microcontroller. In this paper, two Nios II soft-core processors are designed in an FPGA chip to achieve the dual function of the robot controller with the features of programmable and being easy to implant. The requirements of robot controllers integration and miniaturization are solved by the method, and has good openness and real-time. The designs of the controller hardware and software systems are described, and the dual-core controller is used in multi-joint robot servo control. The experimental results show that the robot controller system is reasonable and feasible.

Keywords: Solid-plug conveying, Centrifugal extruder, Polymer process, Mathematics analysis.

1. INTRODUCTION

In modern industrial production process, the application field of industrial robots is increasingly expanding. The industrial robots have become an indispensable tool for factory automation, flexible manufacturing systems; computer integrated manufacturing system and other aspects. Modern robotic technology continues to combine computer technology, control technology and microelectronics technology, and it is then applied into new fields. The degree of complexity and intelligence of robots have become more and more higher. Moreover, the development of the robot controller is developing in the direction of network intellectualization [1].

In recent years, with the rapid development of microelectronics technology, the integration scale of programmable logic devices (FPGA) has become larger and larger, and its performance got more and more perfect, which provides a new way for the implementation of the robot controller. One can use specified software and a hardware description language to implement some kind of control algorithm, and then the control algorithm will be downloaded into the FPGA, finally control algorithms will be implemented by hardware. As opposed to other control systems, the method has many advantages. The SOPC (System on Programmable Chip), which is the product of further development of the FPGA, can provide adequate programmable logic resources, rich IP core resources for the robot controller, and has the advantage of low power [2].

Based on available studies, dual-core processor was designed in FPGA using SOPC resource for motion control of the robot. The experimental results show that the method is feasible.

2. DUAL-CORE CONTROLLER SYSTEM

2.1. Industrial Robot Controller

The main functions of industrial robot controller are making moving parts be in accordance with the expected trajectory and specified motion parameters to complete the appropriate action. It mainly includes two aspects: (1) computing algorithm which realizes robot trajectory planning; (2) motion control which mainly completes the control process for the position of each joint motor.

Currently, the industrial robot control system is mainly secondary robot control system of industrial PC+ microcontroller. The system of the upper machine, being mainly composed of industrial PC, is responsible for the human-computer interaction, path planning, trajectory compensation, teaching and other top task. However, lower computer is responsible for motion control of each joint, and each joint usually contains an 8 or 16-bit microcontroller, such as MCS-51 or MCS-96 and other systems. Because most of the control logic may be implemented in software, in addition to the logic, operation of the microcontroller is very well, so that their operation speed is generally very fast and the precision is high. Meanwhile, their memory capacity is so large that more complex control algorithms can be implemented. Of course, since the algorithm is implemented in software, its flexibility and adaptability are also greatly improved. For example, if one wants to modify the control requirements, the corresponding hardware circuits will not need to be changed, instead, it is only necessary to change the corresponding codes. It can be seen that the corresponding motor control circuit called controller can be modeled using software, there being a great improvement in implementing more complex control algorithms. However, this system is complex, bulky, and the information exchanging between upper and lower computer is slow, so it is difficult to implement robot of miniaturization.

*Address correspondence to this author at Mail No. #60, Electrical Engineering College, Henan university of science and technology, No. 263 Kai-Yuan ROAD, Luoyang, China; Tel: +86 13653881562; E-mail: sunligong148@126.com

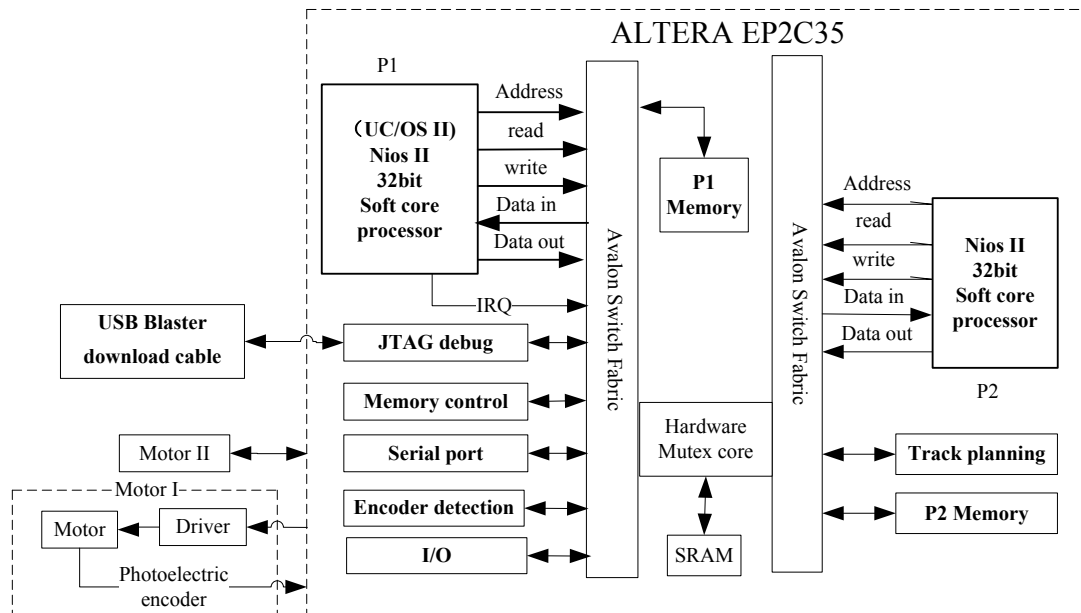


Fig. (1). The block diagram of the control system.

2.2. Design of Dual-Core Industrial Robot Control System

Taking into account the dual function of the robot controller, the programmable and easy-to-implant characteristics of FPGA, two Nios II soft-core processors were designed in an FPGA, and μ C/OS-II embedded system was transplanted into one Nios II soft-core, to complete motion control requiring high real-time and the tasks of switching control, in turn to control the rotation of joints motor. The tasks of the other Nios II soft-core processor is to do trajectory planning with a large amount of computation, solving the inverse kinematics equations, location interpolation calculation and other pre-processing of movement instructions, in addition to obtaining real-time position feedback returning from encoder, transmitting the input and output data and the position command. Dual CPU structure consists of dual Nios II, and its function corresponds to the upper and lower machines of conventional robot controller. The block diagram of hardware module is shown in Fig. (1).

In the system, the compensation control for the stepping motor, includes the control of its speed, acceleration and other aspects: the relationship between the rate of the pulse frequency change and the amount of acceleration change, the relationship between the level and speed of pulse frequency, the relationship between the number and the displacement amount of the pulse, and so on. There are two input ways in system: data sampling and pulse sequence input, where software and hardware can implement the function of the data-sampling position control. In operation, the corresponding calculation on speed and increments were done by the software, and it was finally completed by the hardware. For example, interpolation command data was obtained in the first-step position servo interrupt, that is, the axis in the first step should act as a displacement increment movement; while the actual displacement of the axis was got according to the position feedback sampled [3].

Motion controller not only requires the system to complete the coarse interpolation and fine interpolation during operation, but also to complete the I/O management and feedback pulse count. In this process, due to the large computation amount of coarse and fine interpolation, it is not easy to be completed by use of the SCM system. Therefore, a dual Nios II soft-core FPGA architecture is adopted in the system to achieve the purpose of real-time controlling and high-speed pulse output. Since both soft-core CPU are in an FPGA, there is no signal transmission delay which can realize seamless connection. The dual CPUs implement man-machine interaction and algorithm, respectively. CPU1 responsible for motion control was to control each control volume in the use of a high real-time μ COS II operating system; CPU2 responsible for algorithm does neither assume other additional tasks nor resets operation system.

3. SOFTWARE SYSTEM OF DUAL-CORE ROBOT CONTROLLER

3.1. Real-Time Operation System

The robot controller is a multi-tasking real-time controller for parallel execution. In the software implementation, the modules were divided using the structure composed of management layer, execution layer, motion control layer, and the communication between the modules is made according to the standard interface to replace and add new modules easily. In real-time operating system available, μ C/OS-II is an open-source, pollable, cured, scalable, preemptive real-time multitasking core, providing task scheduling, task management, time management and inter-task communication, etc., and it continues to upgrade and add new features. Therefore, μ C/OS-II was chosen to act as real-time operation system. The most of μ C/OS-II code is written in C language, but during doing some operations related to processor, it also needs to adopt assembly lan-

guage. For example, the read and write of register are only to be achieved through the store and load instructions of assembly language. For $\mu\text{C}/\text{OS-II}$ to work well, the related processors need to meet certain requirements [4], such as, C compiler should be able to generate reentrant code processor; the processor supporting interrupt which can generate timer interrupt should exist, data hardware stack can be stored in processor, and in the processor, there are instructions which can read out and store other registers and stack pointer into the stack.

3.2. Communication Between Nios \square Multiple Processors

Altera Nios II processor is a universal soft-core micro-processor which can be implemented in specific FPGA and support 32-bit RISC. On the aspect of development, the method of parameterization interface bus is adopted in Altera Corporation, that is, either its capability of multi-master and slave arbitration is very friendly, or its system resource sharing capabilities is very powerful. There are a variety of shared communication solutions between dual-core processor, each of which has different characteristics, advantages and disadvantages. The dual-core query communication of exclusive hardware and shared storage was adopted in this paper.

When the system is designed using the development tools SOPC Builder provided by Altera [5], one can use Mutex hard core components of the system. The core is integrated into the SOPC builder development software, and data access is made and sharing of resources is coordinated through the Avalon interface. In the environment of multiprocessor, each processor has the right to use Mutex core, but they do not have the right to use it simultaneously. There are hardware-based atomic measurements and test-and-set (set) operations in the core.

In the Nios II HAL, the functions of Mutex core access have been provided by Altera, which are open, lock, unlock, judge Mutex, respectively. When the Mutex core is used, each CPU must obtain the ownership of Mutex core to make the appropriate read and write operations. When the corresponding operations are completed, CPUs controlling Mutex core must release the right of using the Mutex core, in order to allow other CPUs to get the appropriate permissions, and in turn to prevent Mutex core from being used exclusively by a CPU, thus to avoid the event of using exclusively shared resources.

4. THE IMPLEMENTATION OF DUAL-CORE ROBOT CONTROLLER

In the paper, the EP2C35F484 FPGA provided by Altera Corporation was adopted to act as the carrier of the dual-core architecture robot controller. The chip is a cost-effective FPGA chip supporting the Nios II embedded microprocessor core. The EP2C35F484 chip consists of 33216LEs, 35 multiplication modules, 475 available I/O pins, 483,840 RAM bits, four phase-locked loops, 1B-20KB on-chip storage, scalable 4 GB off-chip memory, 16-bit or 32-bit configurable function possessed by Nios II embedded microprocessor.

Therefore, the chip is used to develop a multi-axis servo robot motion controller in the paper.

When the system was built in SOPC Builder, the above modules or components provided in the library were chosen. SOPC Builder can import or provide interface which could reach user-defined logic blocks. When SOPC Builder system was used in conjunction with user-defined logic, there are following features: PIO connection, system module instantiation, reaching the bus interface of external logic and releasing local SOPC Builder component. The SOPC Builder provides library components (modules) for downloading, including Excalibur embedded processor, NIOS processor, CPU, UART, timer, PIO, Avalon tristate bridge, multiple memory interfaces and OS/RTOS core and so on.

The robot controller hardware platform deployed by Quartus II and SOPC Builder software provided by Altera Corporation has obvious advantages. One can freely add and remove the components in the design, it being only need to deploy the corresponding output pin when the design will be completed. Engineers can quickly and easily complete hardware design, to reduce greatly the cost and the corresponding development time. In addition, all the design work is done in a single chip, thus the volume of the controller is much smaller than the traditional robot control system.

5. KINEMATICS ALGORITHM OF CONTROLLER

The operation of the robot is achieved through operating hand end effector. The movement of end effector is critical to the effect of the robot control. The robot kinematics is the subject of studying the robot movement as a function of time with respect to the fixed reference frame, regardless of the forces and moments of the movements. That is, the space displacement of the robot should be analyzed as a function of time, to particularly study the relationship between the joint variable space, robot end effector position and attitude.

In the kinematics problems of robots, including the expression of kinematics equations, forward and inverse solution of kinematics, the coordinate position of i with respect to the $i-1$ and homogeneous transformation matrix of direction T and T may be obtained by means of Denavit-Hartenberg method.

Table 1 shows geometric parameters of each joint of the industrial robot SCARA and its structural dimensions are shown in Fig. (2).

According to the mathematical derivation of the kinematic system [6], homogeneous transformation formula is given by:

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Substituting the link parameters of each joint, A_1, A_2, A_3, A_4 can be obtained. Multiplying each link transform matrix,

Table 1. Joint Parameters of Robot.

Link	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	l_1	d_1	θ_1
2	180°	l_2	0	θ_2
3	0	0	d_3	0
4	0	0	0	θ_4

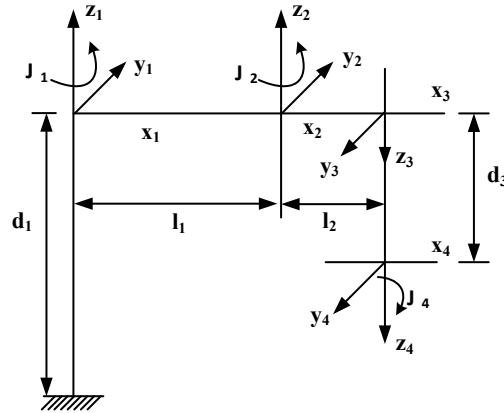


Fig. (2). D-H coordinates of robots.

the position equation of the robot end effector can be obtained as:

$$T_{04} = \begin{bmatrix} \cos(\theta_1 + \theta_2 - \theta_4) & \sin(\theta_1 + \theta_2 + \theta_4) & 0 & l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2 - \theta_4) & -\cos(\theta_1 + \theta_2 - \theta_4) & 0 & l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & -1 & d_1 + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Therefore, the rate equation can be obtained as follows:

$$\dot{T}_{04} = \begin{bmatrix} \dot{n}_x & \dot{o}_x & \dot{a}_x & \dot{p}_x \\ \dot{n}_y & \dot{o}_y & \dot{a}_y & \dot{p}_y \\ \dot{n}_z & \dot{o}_z & \dot{a}_z & \dot{p}_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

Where, T_{04} and \dot{T}_{04} are velocity vector matrix and position vector matrix of the arm end gripper in the base coordinate system, respectively; $p = (p_x, p_y, p_z)^T$ is the position

vector of the end effector; $a = (a_x, a_y, a_z)^T$ is proximity vector; $o = (o_x, o_y, o_z)^T$ is the direction vector; $n = (n_x, n_y, n_z)^T = a \times o$ is the normal vector.

Left multiply $A^{-1}_1, A^{-1}_2, A^{-1}_3, A^{-1}_4$ for T_{04} . Then compare the corresponding elements of both ends of the equation. Finally the inverse solution of each pose can be obtained including joint variable value and speed. The inverse solution of each joint variable is shown as follows:

$$\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2} \quad \cos \theta_2 = \frac{1}{2l_1 l_2} (r^2 - l_1^2 - l_2^2) \quad (4)$$

$$\theta_2 = \arctg\left(\frac{\sin \theta_2}{\cos \theta_2}\right) \quad \theta_1 = \arctg\left(\frac{\sin \theta_1}{\cos \theta_1}\right) \quad (5)$$

$$\sin \theta_1 = \frac{1}{r^2} [(l_1 + l_2 \cos \theta_2) p_y - l_2 \sin \theta_2 p_x] \quad (6)$$

$$\cos \theta_1 = \frac{1}{r^2} [(l_1 + l_2 \cos \theta_2) p_y + l_2 \sin \theta_2 p_x] \quad (7)$$

$$d_3 = d_1 - p_z \quad (8)$$

$$\cos \theta_4 = o_x \sin(\theta_1 + \theta_2) + o_y \cos(\theta_1 + \theta_2)$$

$$\sin \theta_4 = -[o_x \sin(\theta_1 + \theta_2) + o_y \cos(\theta_1 + \theta_2)] \quad (9)$$

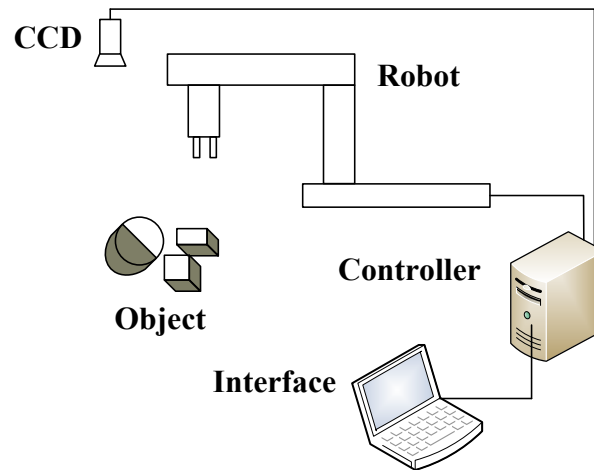


Fig. (3). Schematic of system hardware.

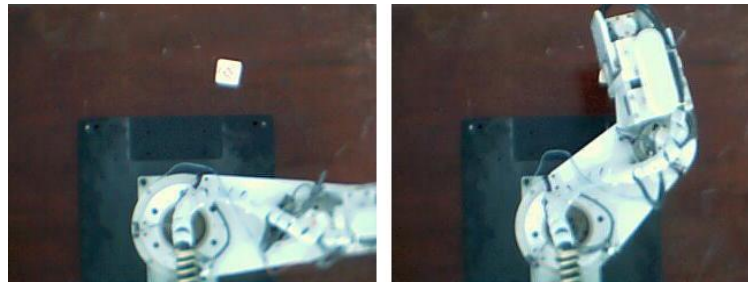


Fig. (4). Robot capturing the square target.

$$\theta_4 = \arctg\left(\frac{\sin\theta_4}{\cos\theta_4}\right) \quad (10)$$

Where, $r_2 = p_x^2 + p_y^2$.

The above algorithm was implemented in a dedicated processor of the robot controller.

6. EXPERIMENT

The industrial robot controller of dual-core architecture designed in this paper is being applied to multi-joint robot servo control to grab the target. Satisfactory results were achieved. The experimental hardware system consists of GRBA4PAH industrial robots, industrial robot controller of dual-core architecture, miniature CCD camera, target object and Human Machine Interface. The experimental hardware system is shown in Fig. (3). The experiment was done indoors, targets being some objects of various shapes of the size of 70mm×70mm, while the distance between CCD camera and the platform is 1.7 meters.

The experimental procedure is as follows: putting the target object on the experimental platform, obtaining targets images by the camera, determining the target position information by the coordinate transformation, planning controller trajectory by the robot, sending control commands to control the action of robot's each joint and capturing the target object. The capturing process is shown in Fig. (4).

CONCLUSION

The experimental results show that the robot controller system of dual-core architecture designed in the paper is reasonable and feasible, and it is able to meet real-time requirements, moreover, its overall performance is well. The dual function of complex calculation and motion control in a single FPGA was achieved in the system. There is a high degree of system integration and a large number of I/O interfaces were set aside for general-purpose connector which provided a lot of room for expansion. Its openness is well. Due to the advantages of independence and online programming for SPOC system, it is easy to make robots small and integrative and space is also reserved for extending robot functions.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGMENTS

This work was financially supported by the Natural Science Foundation of Henan Educational Committee (Grand No. 2006510004).

REFERENCES

- [1] Speck A, and Gruhler G. "Object-oriented robot control Framework", in International Conference on Robotics and Automation, pp. 1663, 1998.

- [2] Shao X Y, and Sun D. "A FPGA-based motion control IC design", *Industrial Technology*, vol. 10, pp. 131, 2005.
- [3] Hong K S, and Choi K H, etc. "A PC-based open robot control system", *Robotics and Computer Integrated Manufacturing*, vol. 17, pp. 355, 2001.
- [4] He J L. "Study and Application of Robot Controller Based on System on Programmable Chip", *Doctoral Dissertation of Shanghai Jiaotong University, Shanghai, ON, China, 2004.*
- [5] G. Q. Wang. *Technology and application of the SOPC*, CA: Beijing, 2006, pp.41.
- [6] Cheng D. "Research on design movement and dynamics of SCARA robot", *Master Thesis of Hefei University of Technology, Hefei, ON, China, 2008.*

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Ligong *et al.*; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the (<https://creativecommons.org/licenses/by/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.