We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

1

The Real-time and Embedded Soccer Robot Control System

Ce Li[1], Takahiro Watanabe[1], Zhenyu Wu[2], Hang Li[2] and Yijie Huangfu[2]
Waseda University[1], Japan[1]
Dalian University of Technology[2], China[2]

1. Introduction

Robot Soccer becomes more popular robot competition over the last decade. It is the passion of the robot fans. There are some international soccer robot organizations who divide the competitions into several leagues, each of these leagues focus on the different technologies.
In this chapter, the rules and history of RoboCup Small Size League games will be introduced shortly. That can make the audience understand the current design style smoothly. Comparing the small robot with our human being, we can easily find that the mechanism looks like one's body, the circuit looks like one's nerve, the control logic looks like one's cerebellum, the vision system looks like one's eyes and the off-field computer which is used for decisions looks like one's cerebrum. After all, the RoboCup motto is: "By the year 2050, develop a team of fully autonomous humanoid robots that can play and win against the human world champion soccer team" (Official RoboCup Org., 2007).
Nowadays, with the development of LSI, the applications of FPGA make the circuit design more simple and convenient, especially for the soccer robot which always needs to be programmed in the field. A soft-core CPU which can be embedded in FPGA can fill a gap in the FPGA control logic and can also make the design more flexible. In this chapter, the circuit design configuration of our soccer robot which is developed based on FPGA is introduced, including real-time control system, the function of each module, the program flow, the performance and so on.
After we got a stable control system based on single CPU in the FPGA, we start to make an attempt to embed multiple CPUs in the control system. It gets an obvious advantage of high performance that two CPUs can work at the same time. Although one CPU can meet the request of global vision, multiple CPUs could pave the way for self vision systems or more complicated control logic.

2. Background

2.1 RoboCup (Official RoboCup Org., 2007)

RoboCup is an annual international competition aimed at promoting the research and development of artificial intelligence and robotic systems. The competition focuses on the development of robotics in the areas of:

- Multi-Agent robots planning and coordination
- Pattern Recognition and real time control
- Sensing Technology
- Vision Systems (both global and local cameras)
- Mechanical design and construction

The RoboCup World Championship consists of different levels:

- Soccer Simulation League
- Small Size Robot League
- Middle Size Robot League
- Standard Platform League
- Humanoid League

RoboCup is a competition domain designed to advance robotics and AI research through a friendly competition. Small Size robot soccer focuses on the problems of intelligent multi-agent cooperation and controlling in a highly dynamic environment with a hybrid centralized or distributed system.

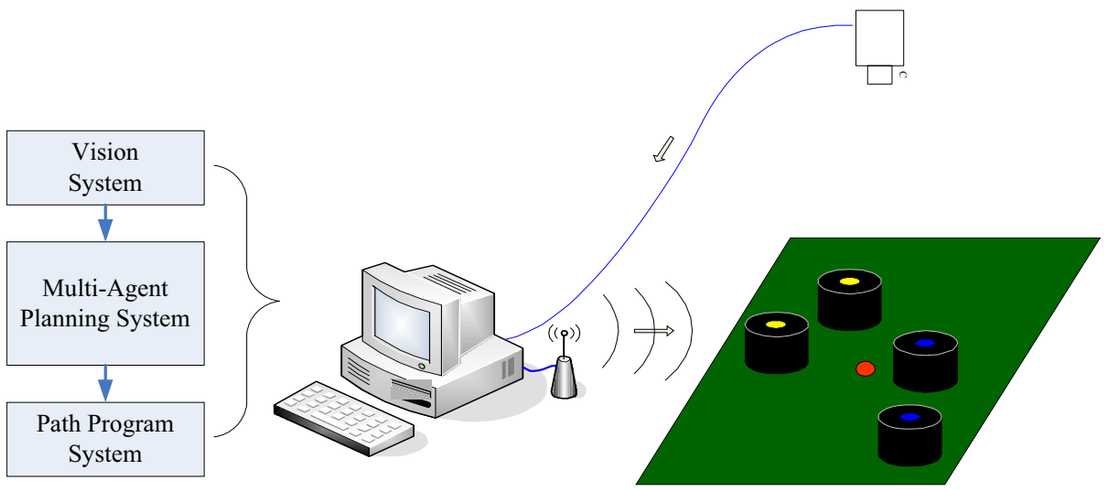

Fig. 1. Overview of the entire robot system

A Small Size robot soccer game takes place between two teams of five robots each. The environment of the game shows in Figure 1. Each robot must conform to the dimensions as specified in the rules: The robot must fit within a 180mm diameter circle and must be no higher than 15cm unless they use on-board vision. The robots play soccer (an orange golf ball) on a green carpeted field that is 6050mm long by 4050mm wide (Official RoboCup Org., 2007). For the detail rules,. For the detail rules, please refer RoboCup web site. Robots come in two ways, those with local on-board vision sensors and those with global vision. Global vision robots, by far the most common variety, use an overhead camera and off-field PC to identify and drive them to move around the field by wireless communication. The overhead camera is attached to a camera bar located 4m above the playing surface. Local vision robots have the sensing on themselves. The vision information is either processed onboard the robot or transmitted back to the off-field PC for processing. An off-field PC is used to communication referee commands and position information to the robots in the case of overhead vision. Typically the off-field PC also performs most, if not all, of the processing

required for coordination and control of the robots. Communications is wireless and Wireless communication typically uses dedicated commercial transmitter/receiver units. Building a successful team requires clever design, implementation and integration of many hardware and software sub-components that makes small size robot soccer a very interesting and challenging domain for research and education.

2.2 System Overview

The robot system is a layered set containing subsystems which perform different tasks. Figure 1 shows the flow diagram that how the system is laid out. An overview of the system is given bellow by following the flow of the information from the camera to the robots actuators (motors).

The overhead digital camera captures global images of the field by 60 fps. The vision system (software installed in off-field PC) processes these images to identify and locate the robots and the ball. The environment and state information of the field are sent to the Multi-Agent Planning System (MAPS). MAPS is the highest level planner of the robot system, arranges the task of the whole team, actually, arranges each individual robot's action and its action location. Some actions include KICK and DEFEND (Ball D., 2001).

After each robot has an action, the Path Program system calculates the path for the robot to achieve its action, and optimizes path for each robot.

In the robots, there is a motion system which can accelerate and decelerate the robot to the desire speed and distance by creating force limited trajectories. The motion system ensures wheel slip to a minimum.

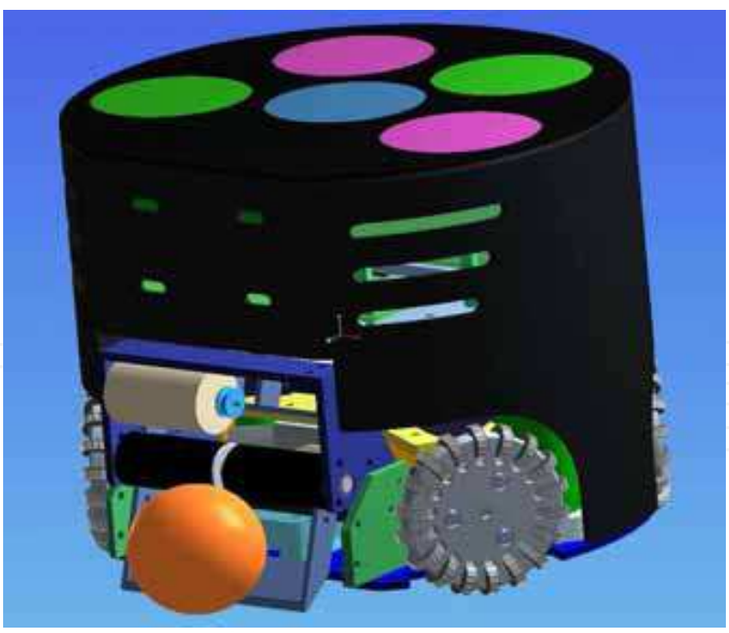

Fig. 2. The 3D assembly drawing of the small size robot

2.3 Mechanical Design

The mechanical design is consisted of an omni-directional drive system, a powerful crossbow kicker, a scoop shot kicker and a dribbler. It should be a compact and robust design. All the robots are of the same mechanical design, a mass of 4.2 kilograms each. The robots are constructed using aluminum that gives them a strong but light frame. The robots have a low centre of mass, achieved by placing the solenoid, the majority of the batteries and the motors on the chassis. It can reduce weight transfer between wheels as the robot accelerates based on the low centre of mass. Consistent weight transfer leads to less slip of the wheels across the playing surface. The whole assembly drawing is shown in Figure 2.

Omni-Directional Drive

For maximum agility, the robots have omni-directional drive. The robot omni-directional drive is implemented by using four motors each with one omni-directional wheel. The angle of the front wheels is 120 degree, because we should add ball control mechanism and kicker between the front wheels. It is 90 degrees between the back wheels. During the year 2007, we use the DC motors. But now, we start to use brushless motors to save more space and improve the performance. Figure 3 shows the 3D assembly drawing of the small size robot chassis with brushless motors.

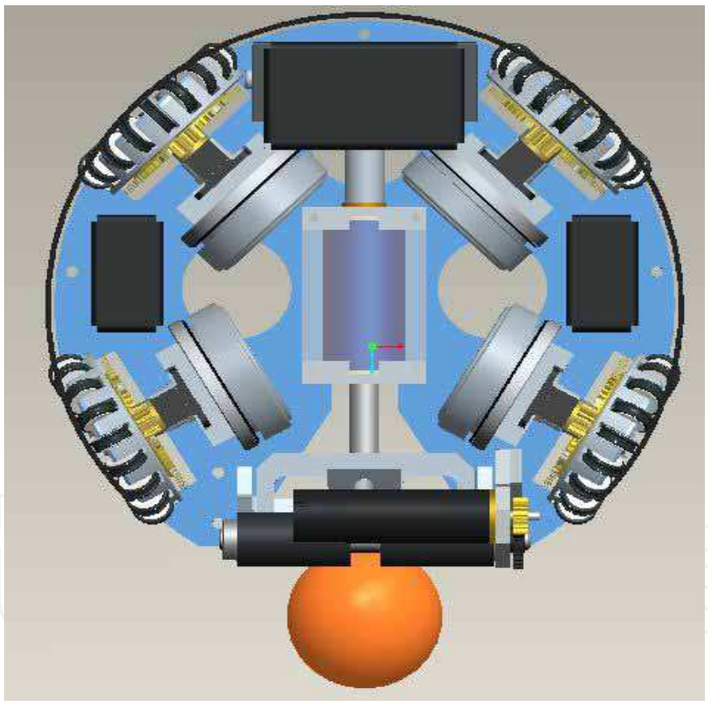

Fig. 3. The 3D assembly drawing of the small size robot chassis with brushless motors

Kicker

The robots feature a powerful kicking mechanism that is able to project the golf ball at 6.2m/s. The kicking mechanism for the Robots is a crossbow and uses a solenoid to generate the energy. The kicking mechanism, while mechanically simple, uses only one solenoid to retract the crossbow. The plate that strikes the golf ball is the same mass as the golf ball. This

gives maximum efficiency of energy transfer, this experience is mentioned in reference (Ball D., 2001). There is also another solenoid and related mechanism for scoop shot.

Ball Control Mechanism

The ball control device of Robots is a rotating rubber cylinder that applies backspin to the golf ball when touching. Here we use a 6 Volt 2224 MiniMotor to drive the shaft. A 10:1 gearbox is used between the motor and the shaft. One feature of the Robots ball control mechanism is that the cylinder is separated into 2 parts. When the ball is located in this dribbler, it has the benefit as the crossbow kicker could give a more accurate and powerful kick to a ball located in the centre of the ball control mechanism.

2.4 Electrical Control System

Comparing a robot to a human, the electrical control system of the robot would be equivalent to a nervous system of human being. In humans, actions are commanded through the nervous system, and sensed information is returned through the same system. There is no difference in the robots. Sending commands and receiving sensed information are the responsibilities of a critical human organ, the brain. The micro control system in the soccer robot is equivalent to a brain.

Fig. 4. The Soccer Robot with Electronic Control system.

Acting as a metaphorical brain, the micro control system must process received information and generate the appropriate response. The off-field artificial intelligence (AI) computer does most of the required brainwork to make the robots play a recognizable game of soccer, but the on board brain translates the AI's decisions into robotic actions and does the

required thought-processing which is needed to maintain these actions. Encoded commands are received from the AI computer via a wireless module.
By decoding these commands, the microcontroller system determines whether to kick, dribble or move. Onboard sensor feedback indicates if the robot should carry out a kick command. Adequate microcontrollers are necessary for quick and reliable processing of these inputs and outputs.
Microcontrollers are microprocessors with a variety of features and functionality built into one chip, allowing for their use as a single solution for control applications. The operation of a microcontroller revolves around the core Central Processing Unit (CPU), which runs programs from internal memory to carry out a task. Such a task may be as simple as performing mathematical calculations, as is done by an ordinary CPU of a personal computer. On the other hand, the task may be more complex, involving one or many of the microcontroller's hardware features including: communications ports, input/output (I/O) ports, analog-to-digital converters (A/D), timers/counters, and specialized pulse width modulation (PWM) outputs. With access to hardware ports, the CPU can interface with external devices to control the robot, gather input from sensors, and communicate with off-field PC by wireless. The control of these ports, handled by the program running on the CPU, allows for a great deal of flexibility. Inputs and outputs can be timed to occur in specific sequences, or even based on the occurrence of another input or output. A major drawback of microcontrollers is that, there are usually constraints to design and implement a system in a reasonable size for integration in compact systems, because so much processing power can be provided due to the need to fit the CPU and all of the hardware features onto the same chip.

3. Previous Control System

3.1 Control Part

The first real-time and embedded control system of our robots was designed in competitions of year 2006 by us (Zhenyu W. et al., 2007). It was remarkably reliable, and had run without big failure in 2007 China RoboCup Competitions as well as periods of testing and numerous demonstrations.
The robots' main CPU board uses TI TMS320F2812, a 32 bit DSP processor, as a CPU. With the high performance static CMOS technology, it works at a frequency of 150 MHz. This processor executes the low level motor control loop. The major benefit of using this processor is its Event Managers. The two Event Managers have 16 channels Pulse Width Modulation (PWM) generation pins that can be configured independently for a variety of tasks. Except for the kicker mechanism, the motors have encoders for fast and immediate local feedback.
A DSP and FPGA based digital control system has already been developed for the control system where FPGA gathers the data and DSP computes with it in 2006. Figure 5 describes the frame of the previous control system. This hardware architecture takes the advantage of the higher computation load of DSP and the rapid process.
In this Figure, there are two broken line frames, one is DSP board, and the other one is FPGA board. On the DSP board, DSP processor communicates with wireless module by serial interface. If the wireless module gets data from the off-field PC, it generates an interrupt to the DSP processor. When DSP is interrupted, it resets the PID parameter, then,

starts a new PID control loop. After a new speed is calculated, DSP processor drivers the motors by PWM signal.
On the FPGA board, there are FPGA, RAM, flash memory, kicker control circuit, LED, etc. FPGA is used to connect with these peripherals and has the features below:

- save and fetch data in RAM and flash
- decode the signals from the 512 lines motor speed encoders
- display the system states by LED
- control the kicker
- gather the information of the acceleration

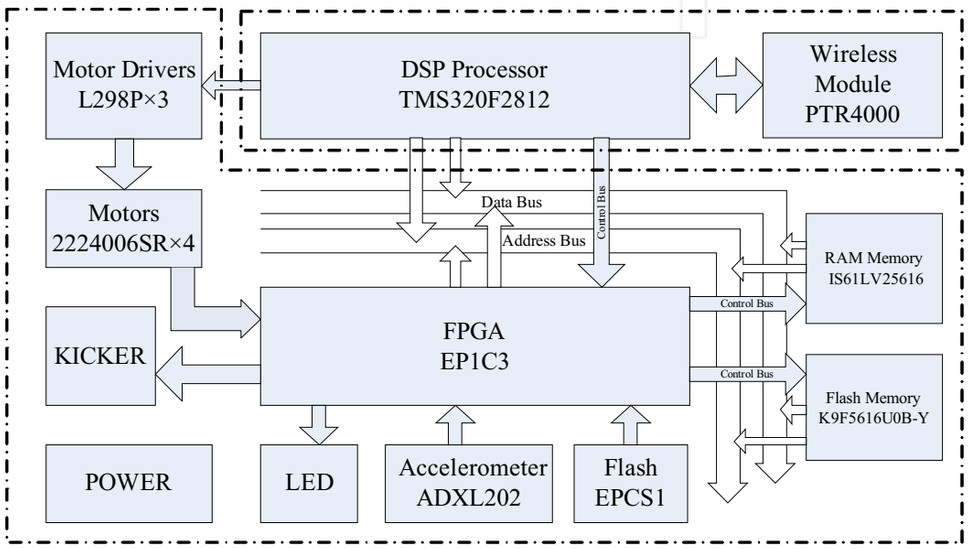

Fig. 5. The DSP and FPGA based digital control system (Zhenyu W. et al., 2007)

The innovative idea of the previous control system is that DSP controls each peripheral by writing and reading data of the registers at specific addresses in FPGA. The rest of tasks are done by FPGA except wireless communication and motor PWM control.

3.1.1 DSP

The TMS320F2812 devices, member of the TMS320C28x DSP generation, is highly integrated, high-performance solutions for demanding control applications. The C28x DSP generation is the newest member of the TMS320C2000 DSP platform. Additionally, the C28x is a very efficient C/C++ engine, hence enabling users to develop not only their system control software in a high-level language, but also enables math algorithms to be developed using C/C++. The C28x is as efficient in DSP math tasks as it is in system control tasks that typically are handled by microcontroller devices. The 32 x 32 bit MAC capabilities of the C28x and its 64-bit processing capabilities, enable the C28x to efficiently handle higher numerical resolution problems that would otherwise demand a more expensive floating-point processor solution. Add to this the fast interrupt response with automatic context save of critical registers, resulting in a device that is capable of servicing many asynchronous events with minimal latency. Special store conditional operations further improve performance (TI Corp., 2004).

3.1.2 FPGA

The Field Programmable Gate Array (FPGA) plays an important role in the sub-system on the robots. In the motion control system, the FPGA provides the interface between the motor encoder and the motion control DSP. It takes the two motor encoder signals to decode the speed of each motor. It can also be used to control the peripherals by setting the registers at the special address. During the design period of the previous control system, we compared a number of FPGAs in the MAX 7000 family and Cyclone family to choose the one that satisfied our requirements.

One major factor we considered is in-system programmability since the FPGAs are able to be programmed on board. The capacity of MAX 7000S family chip is so little, and when we select a suitable type, the price is very high. While most of the FPGAs in the EP1C3 family meet our needs, our final board design uses this family FPGA. EP1C3 devices are in-system programmable via an industry standard 10-pin Joint Test Action Group (JTAG) interface. With a large capacity of LEs, it leaves us room for future additions. For our actual implementation and functional testing, we chose the EP1C3T114C6 type FPGA. The number of available LEs determines how complicated our circuit can be. The EP1C3T114C6 has 2,910 LEs. As a point of reference, our final design utilizes 27% of all the usable resource. This covers miscellaneous logic to support enable functionality. The maximum number user I/O pins for the EP1C3T114C6 is 144 and we used 129 in our final design. These numbers are all on the order of nanoseconds and easily met our requirements. Our final circuit could run at speed of 45.7MHz.

3.2 Wireless module

Fast and compatible wireless communication module is required for the heavy task such as increasing the speed of transmission and reducing the latency of the entire system. In addition, the system must exhibit a high level of interference rejection and low error rate.

In order that a robot works successfully, it must be able to receive up-to-date game data rapidly and consistently. To meet these requirements, the following properties are necessary for a communications system:

- High transmission speed
- Low latency
- Interference rejection
- Low error rate

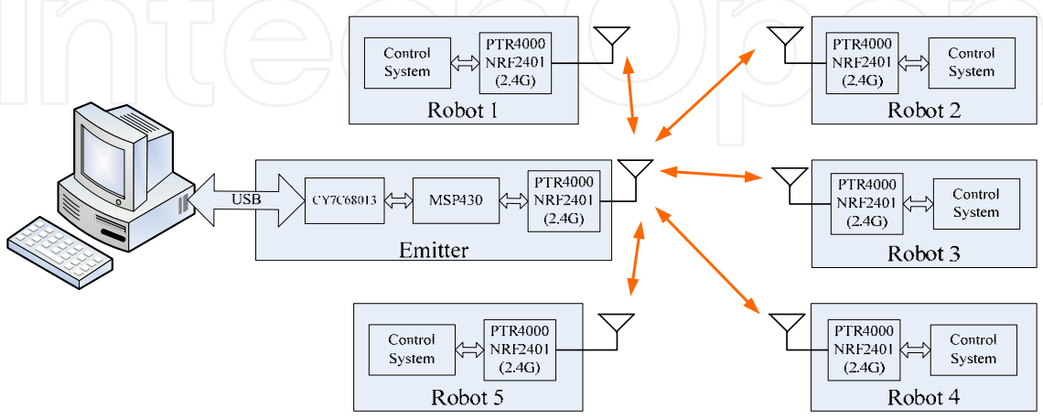

Fig. 6. Full Duplex System

Figure 6 shows the communicated method between robots and the off-field PC. The off-field PC sends motion instructions to the robot by a USB emitter, and also can get the states and information of them by the wireless communication.
For fast and steady communication, we selected PTR4000 wireless module. The central chip of PTR4000 is nRF2401 which is a single-chip radio transceiver for the world wide 2.4-2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator and a modulator. Output power and frequency channels are easily programmable by use of the 3-wire serial bus. Current consumption is very low, only 10.5mA at an output power of -5dBm and 18mA in receive mode.

3.3 Motor control

The reactive control loop is initiated every millisecond by the Event managers (EVA and EVB). Once robot receives its desired velocities, it can calculate the wheel velocities and then sends and then sends them the PWM signals to the motors. Figure 7 illustrates the control model that is implemented in the reactive control loop with motors. In determining the output of the system the control loop calculates the proportional and integral errors of the wheel velocities. The velocity proportional errors for each of the wheels are calculated.

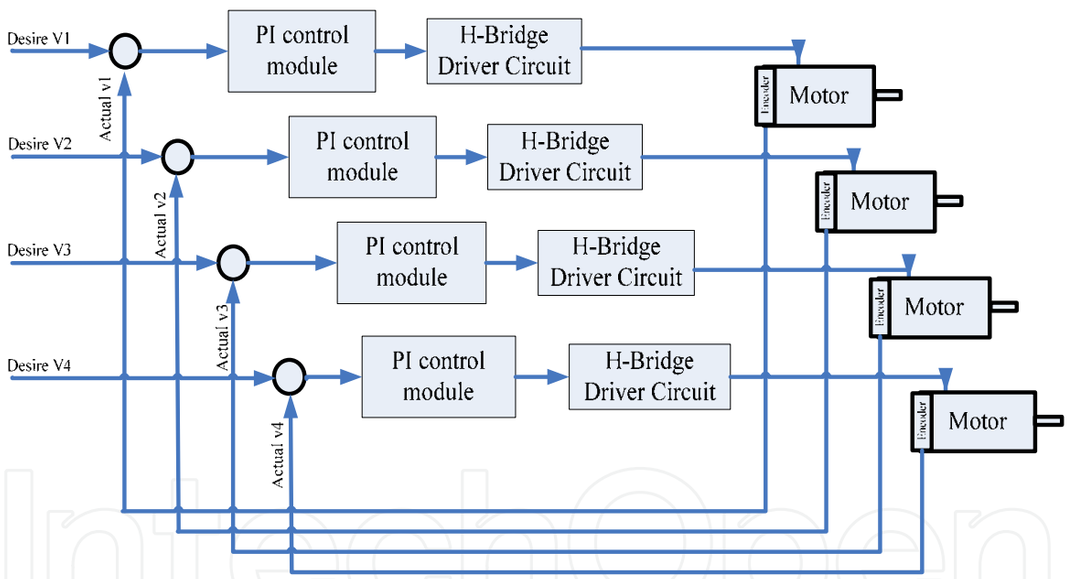

Fig. 7. The control module implemented in our robots in 2006

3.4 Evaluation

In the year 2006, the control system of the robot was a modular design that is fully capable of performing its required tasks. But the design is mainly constrained by some related factors:

- Long design period
- hard modification

There are many chips and many supply voltages, that will cause wire congestion and large area. Many interfaces between the two boards, it is easy to make mistakes. So, we explore a new method to simplify the design.

4. New System

4.1 System on a Chip

System on a Chip or System On Chip (SoC or SOC) is an idea of integrating all components of a computer or other electronic system into a single integrated circuit (chip). It may contain digital signals, analog signals, mixed-signal, and radio-frequency function on one chip. A typical application is in the area of embedded systems (Wikipedia, 2009).

With the planned improvements and designs for new control system, the previous system was no longer sufficient, thus we have to select and design a new system. This process of designing the onboard brain for the new robots involved many steps. It was important to understand the workings of the previous system in order to make educated decisions about improving upon the old design. In addition, it was important to be in contact with the other team members who were concurrently designing and developing electrical components that directly interact with the DSP. It is necessary that the interface between those components and different processors should be the easily to change. This information was the basis upon which we selected the candidate for the new microcontroller selection and also the succeeding evaluation process.

For the new control system, we decide to use SoC as the kernel of the control system. When considering the new processor, Nios II processor, a soft core processor based on FPGA, enters the field of our vision.

4.2 Nios II Processor

4.2.1 Nios II Processor

The Nios II processor is a general-purpose RISC processor core, provided by Altera Corp. Its features are (Altera Corp. 2006):

- Full 32-bit instruction set, data path, and address space
- 32 general-purpose registers
- 32 external interrupt sources
- Single-instruction 32 × 32 multiply and divide producing a 32-bit result
- Dedicated instructions for computing 64-bit and 128-bit products of multiplication
- Floating-point instructions for single-precision floating-point operations
- Single-instruction barrel shifter
- Access to a variety of on-chip peripherals, and interfaces to off-chip memories and peripherals
- Software development environment based on the GNU C/C++ tool chain and Eclipse IDE

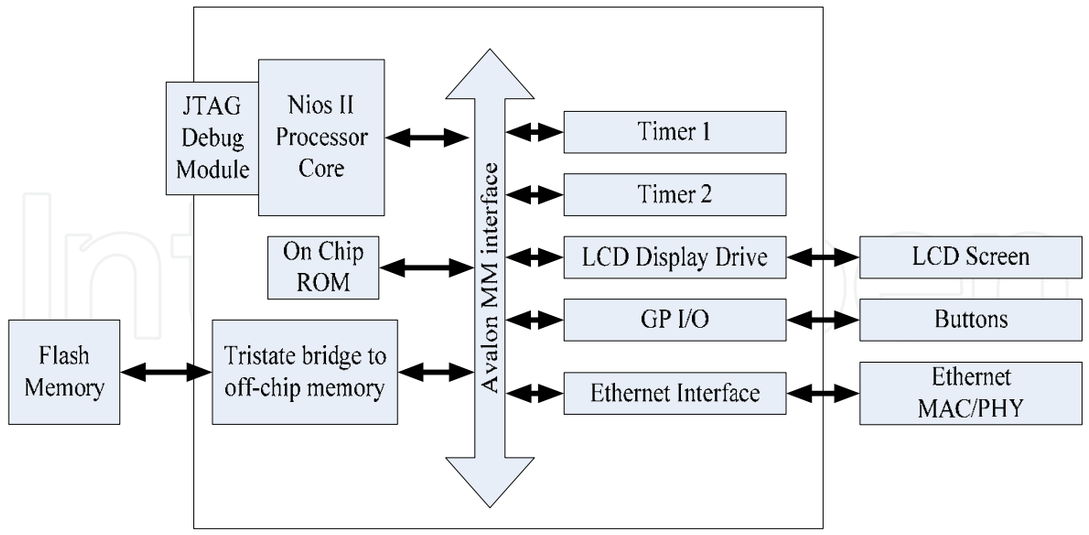

Fig. 8. Example of a Nios II Processor System (Altera Corp., 2006)

A Nios II processor system is the system that we can generate it with one or more Nios II processors, on-chip ROM, RAM, GPIO, Timer and so on. It can add or delete the peripherals and regenerate the system in minutes. Figure 8 is just an example of this system.
If the prototype system adequately meets design requirements using an Altera-provided reference design, the reference design can be copied and used as-is in the final hardware platform. Otherwise, we can customize the Nios II processor system until it meets cost or performance requirements.

4.2.2 Advantage of Nios II Processor

This section introduces Nios II concepts deeply relating to our design. For more details, refer (Altera Corp., 2006).

Configurable Soft-Core Processor

The Nios II processor is one of configurable soft-core processors provided by Altera Corp., as opposed to a fixed, off-the-shelf microcontroller. "Configurable" means that features can be added or removed on a system-by-system basis to meet performance or price goals. "Soft-core" means the CPU core is offered in "soft" design form (i.e., not fixed in silicon), and can be targeted to any FPGA. The users can configure the Nios II processor and peripherals to meet their specifications, and then program the system into an Altera FPGA, and also they can use readymade Nios II system designs. If these designs meet the system requirements, there is no need to configure the design further. In addition, software designers can use the Nios II instruction set simulator to begin writing and debugging Nios II applications before the final hardware configuration is determined.

Flexible Peripheral Set & Address Map

A flexible peripheral set is one of the most notable features of Nios II processor systems. Because of the soft-core nature of the Nios II processor, designers can easily build the Nios II processor systems with the exact peripheral set required for the target applications.
A corollary of flexible peripherals is a flexible address map. Software constructs are provided to access memory and peripherals generically, independently of address location.

Therefore, the flexible peripheral set and address map does not affect application developers.

Automated System Generation

Altera's SOPC Builder design tool is used to configure processor features and to generate a hardware design that can be programmed into an FPGA. The SOPC Builder graphical user interface (GUI) enables us to configure Nios II processor systems with any number of peripherals and memory interfaces. SOPC Builder can also import a designer's HDL design files, providing an easy mechanism to integrate custom logic into a Nios II processor system. After system generation, the design can be programmed into a board, and software can be debugged executing on the board.

4.2.3 Avalon-MM interface

The Avalon Memory-Mapped (Avalon-MM) interface specification provides with a basis for describing the address-based read/write interface found on master and slave peripherals, such as microprocessors, memory, UART, timer, etc(Altera Corp., 2006).

The Avalon-MM interface defines:

- A set of signal types
- The behavior of these signals
- The types of transfers supported by these signals

For example, the Avalon-MM interface can be used to describe a traditional peripheral interface, such as SRAM, that supports only simple, fixed-cycle read/write transfers.

4.3 Nios II Multi-Processor Systems

Multiprocessing is a generic term for the use of two or more CPUs within a single computer system. It also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them. The CPUs are called multiprocessors. There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how multiprocessors are defined (multiple cores on one chip, multiple chips in one package, multiple packages in one system unit, etc.). (Wikipedia, 2009).

Multiprocessing sometimes refers to the execution of multiple concurrent software processes in a system as opposed to a single process at any one instant. However, the term multiprogramming is more appropriate to describe this concept, which is implemented mostly in software, whereas multiprocessing is more appropriate to describe the use of multiple hardware processors. A system can be both multiprocessing and multiprogramming, only one of the two, or neither of the two (Wikipedia, 2009).

Multiprocessor systems possess the benefit of increased performance, but nearly always at the price of significantly increased system complexity. For this reason, the using of multiprocessor systems has historically been limited to workstation and high-end PC computing using a complex method of load-sharing often referred to as symmetric multi processing (SMP). While the overhead of SMP is typically too high for most embedded systems, the idea of using multiple processors to perform different tasks and functions on different processors in embedded applications (asymmetrical) is gaining popularity (Altera Corp., 2007).

Multiple Nios II processors are able to efficiently share system resources using the multimaster friendly slave-side arbitration capabilities of the Avalon bus fabric. Many processors can be controlled to a system as by SOPC Builder.
To aid in the prevention of multiple processors interfering with each other, a hardware mutex core is included in the Nios II Embedded Design Suite (EDS). The hardware mutex core allows different processors to claim ownership of a shared resource for a period of time. Software debug on multiprocessor systems is performed using the Nios II IDE.

4.4 Structure of a new system

After we had selected the Nios II multiprocessor, we constructed the structure of the control system. First, we enumerated the old function of the control system, for example, motor control, speed sampling, wireless communication, kicker, LED display, flash data access and so on. The new system hardware architecture of year 2007 is shown in Figure 9.
There are many methods to separate the tasks and peripheral equipments of the control system for Multi-Processing (MP). Here we select one method which consists of two parts: a motor control part and a peripheral control part. The kernel of each part is a Nios II processor. One is used for the PID control of the motors. So that, motors have the real-time control that makes them respond quickly. The other one implements other functions of the control system, for example, wireless communication, states displaying, kicker controlling and accelerate sampling.
In the control part, using the H-bridge circuit, processor 1 controls the motors with the PID method. Each motor has a decoder, which can provide rotor position or speed information.

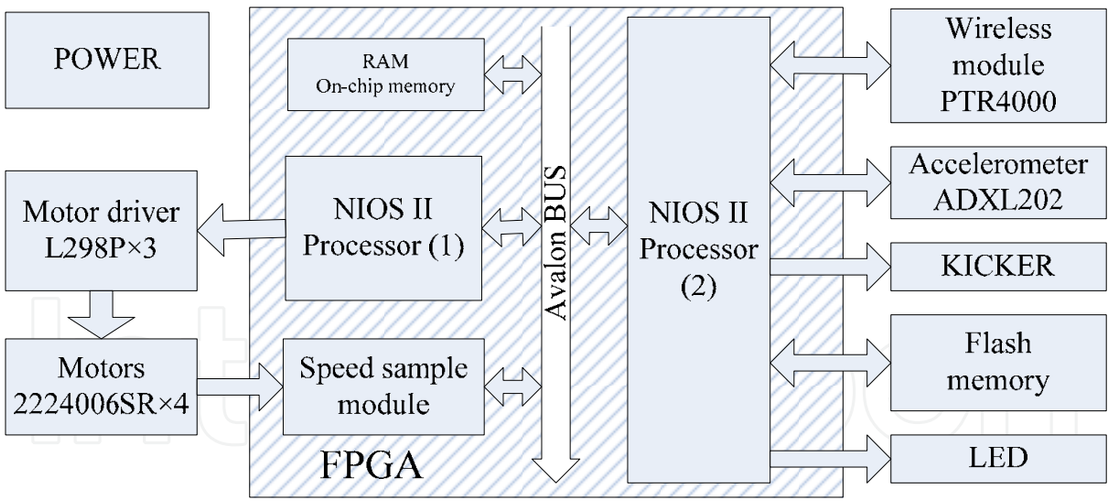

Fig. 9. The Hardware architecture of a new system

Processor 1 can read this information from speed sample module via Avalon bus. Then it compares these values with the desired value in the RAM, and outputs control signals to the motor device.
Processor 2 communicates with the off-field PC by wireless module, samples the acceleration by ADXL202, and controls the kicker and LED. It gathers the information from

PC and writes the data into the internal RAM. Then processor 1 fetches the data which is the desired value set by each control period.

The resolving of multiprocessor
Multiprocessor environments can use the mutex core with Avalon interface to coordinate accesses to a shared resource. The mutex core provides a protocol to ensure mutually exclusive ownership of a shared resource.
The mutex core provides a hardware-based atomic test-and-set operation, allowing software in a multiprocessor environment to determine which processor owns the mutex. The mutex core can be used in conjunction with shared memory to implement additional interprocessor coordination features, such as mailboxes and software mutexes.

4.5 Wireless module
The hardware of the wireless module is not changed for this new system. But because the kernel of the control system has been changed to Nios II processors, we should rewrite the wireless control model by verilog in FPGA. The interface which should accord with the avalon bus is shown in Figure 10. As mentioned in the section 3.2, the communication method is full duplex, so we should code the wireless module with the function of transmitting and receiving.

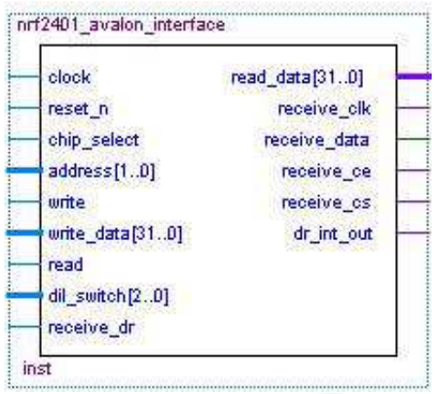

Fig. 10. The block diagram of wireless module.

4.6 Motor control module
The control loop implemented in the robot team had proven itself robust and reliable both in testing and competitions in 2006. The control system was coded by C of DSP processor. Essentially, the motor control module provides the following interface between the processor and the motors. The microcontroller sends two signals to the motor control module (a PWM and direction). These two signals get transformed to voltages applied to the DC motor terminals in our previous robot. In the reverse direction, the motor encoder sends two signals to the motor control module (channel A and B).
The motor control module is newly coded in Verilog because we select Nios II as our processor. The advantages of Verilog include a more optimized design. Also since the code is similar to C, it is easier to maintain. Most significantly, the code is portable between different FPGA families.

The motor control module consists two parts, one is speed sampling part and the other one is PWM part. Speed sampling part, as its name, is used for sampling the motor's real-time speed. Figure 11 shows the speed sampling principle circuit. The output of this circuit is the motor's digital speed, the high bit of which is the direction of the motor.

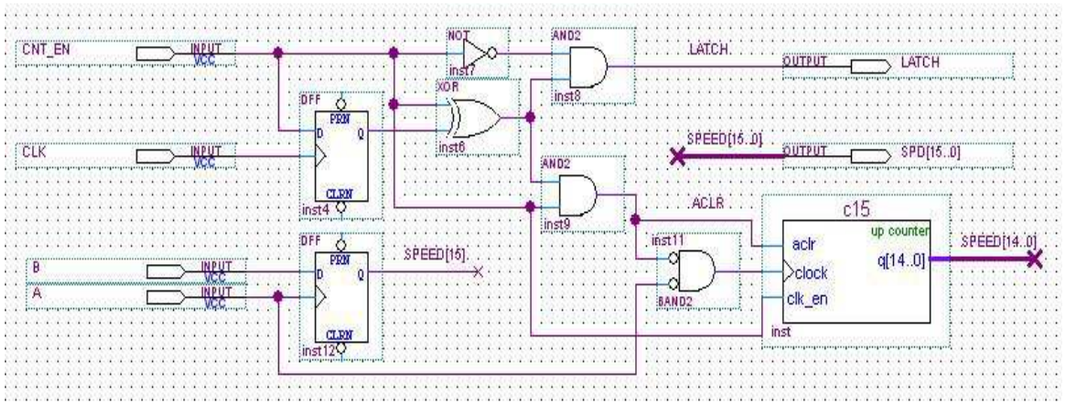

Fig. 11. The principle of speed sampling circuit

When testing the motor control module, we do the simulation with the input signals shown in Figure 12 which is the speed sampling module's simulation result. Output latch is used for latching the SPD signals (speed data) to the data bus, when Nios II processor needs to do the PI control.

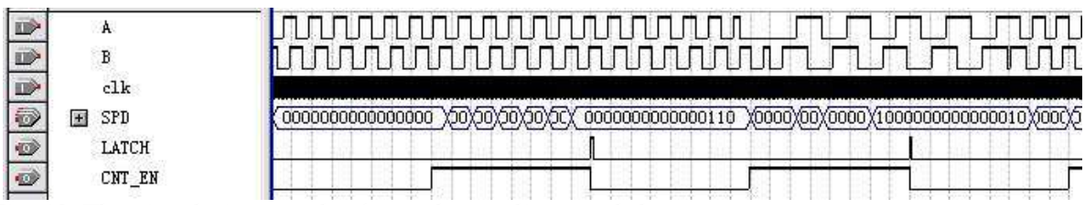

Fig. 12. The simulation result of speed sampling module

The other part of the motor control module is PWM part. The waveform shows in Figure 13. The PMU control design should be packed with Avalon MM interface. With different input number of duty_cycle, the waveform of pwm_out is changed easily. This combination of PWM part and speed sampling part is just for one motor, in our robot control system, 4 modules are mounted.

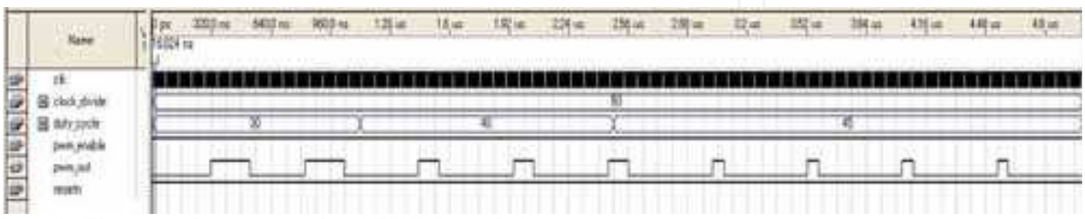

Fig. 13. The simulation result of PWM module.

5. Experiment

Before we start to do the experiment, we should know what we could do, and what the result that we want to get is. We pay more attention to the system design, and test whether the system can work with the module of motor control and wireless.

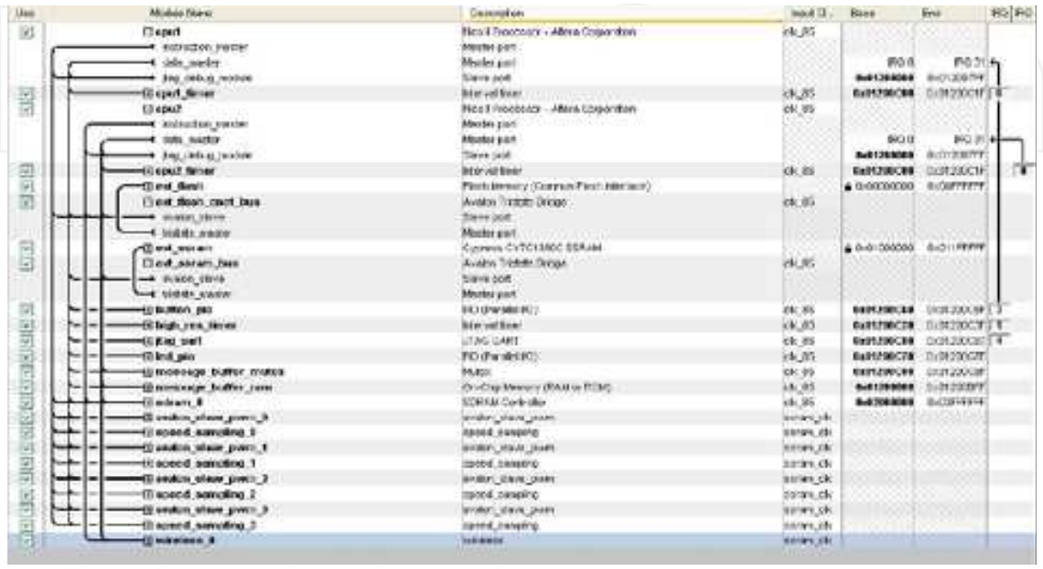

Fig. 14. The Architecture of our robot control system

The FPGA chip which contains two CPUs is EP2C8. EP2C8 is one chip of Altera Cyclone II family. With the advanced architectural features of Cyclone II FPGAs, the enhanced performance of Nios II embedded multiple processors becomes clearly. For the detailed steps of multiple processors generating, please refer (Altera Corp., 2008).

We generate the Nios II MP by SOPC Builder. In this system, as it shows, we add one timer for each Nios II CPU. One interval timer is also added. For the whole system, there are some other contents which would be used in the control system, such as LED_pio, JTAG_UART, message_buffer_mutex. The architecture of our MP robot control system is shown in Figure 14.

After the modules generation, the analysis and synthesis results of each module are shown as Table 1.

Module name	**Total Logic elements**	**Total registers**	**Total PLLs**
Nios CPUs related	5837	2985	1
Speed related module	426	265	0
Wireless module	519	431	0
others	1216	417	0
Total	7884	3998	1

Table 1. the analysis and synthesis results of the robot control system.

6. Conclusion

To build a reliable, robust robot control system which would improve upon the previous design, we approach the problem with a vastly different perspective which contains the MP. In some cases, it improves upon the design tremendously. But if we can not find a best arrangement of the task and cooperation for two processors, that is, it will not turn out to be a splendiferous control system for the robot.

During the design, there are many troubles we have met. The most important one is how to use the previous excellent circuit. If we achieve this, we could change our system as fast as possible.

The electrical design is slim this year. There are still areas which can be improved even more, but by and large, we are proud of the brevity and simplicity of the real-time and embedded soccer robot control system. We have read references for its feasibility and reliability, and started to embed an OS for the MP of Nios II processors.

7. Reference

Robert M. (2001). Control System Design for the RoboRoos, pp. 41~46.

Wikipedia. (2009). http://en.wikipedia.org/wiki/Multiprocessing

Wikipedia. (2009). http://en.wikipedia.org/wiki/System_on_a_Chip

Altera Corp. (2007). Creating Multiprocessor Nios II Systems Tutorial, pp. 5-11.

Altera Corp. (2006). Nios II Processor Reference Handbook, pp. 17-19.

Official Robocup Org. (2007). http://small-size.informatik.uni-bremen.de/

Official Robocup Org. (2007). http://www.robocup.org/

Ball D. (2001). Intelligence System for the 2001 RoboRoos Team. Brisbane: Univ. of Queensland

Dingle P. et al. (2004). 2002 Cornell robocup documentation. New York: Cornell Univ.

Zhenyu W.; Ce L. & Lin F. (2006). A Multi Micro-Motor Control System Based on DSP and FPGA. Small & Special Electrical Machines, vol. 35, No.1, Jan. 2007. pp 30-32, 1004-7018

TI Corp. (2004). TMS320R2811/2 Digital Signal Processors Data Manual, pp 4-7

IntechOpen

Robot Soccer
Edited by Vladan Papi

ISBN 978-953-307-036-0
Hard cover, 348 pages
Publisher InTech
Published online 01, January, 2010
Published in print edition January, 2010

The idea of using soccer game for promoting science and technology of artificial intelligence and robotics was presented in the early 90s of the last century. Researchers in many different scientific fields all over the world recognized this idea as an inspiring challenge. Robot soccer research is interdisciplinary, complex, demanding but most of all, fun and motivational. Obtained knowledge and results of research can easily be transferred and applied to numerous applications and projects dealing with relating fields such as robotics, electronics, mechanical engineering, artificial intelligence, etc. As a consequence, we are witnesses of rapid advancement in this field with numerous robot soccer competitions and a vast number of teams and team members. The best illustration is numbers from the RoboCup 2009 world championship held in Graz, Austria which gathered around 2300 participants in over 400 teams from 44 nations. Attendance numbers at various robot soccer events show that interest in robot soccer goes beyond the academic and R&D community. Several experts have been invited to present state of the art in this growing area. It was impossible to cover all aspects of the research in detail but through the chapters of this book, various topics were elaborated. Among them are hardware architecture and controllers, software design, sensor and information fusion, reasoning and control, development of more robust and intelligent robot soccer strategies, AI-based paradigms, robot communication and simulations as well as some other issues such as educational aspect. Some strict partition of chapter in this book hasn't been done because areas of research are overlapping and interweaving. However, it can be said that chapters at the beginning are more system-oriented with wider scope of presented research while later chapters generally deal with some more particular aspects of robot soccer.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ce Li, Takahiro Watanabe, Zhenyu Wu, Hang Li and Yijie Huangfu (2010). The Real-Time and Embedded Soccer Robot Control System, Robot Soccer, Vladan Papi (Ed.), ISBN: 978-953-307-036-0, InTech, Available from: http://www.intechopen.com/books/robot-soccer/the-real-time-and-embedded-soccer-robot-control-system

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen

IntechOpen