ASPECTS OF IMPLEMENTATION OF AN EDUCATIONAL PLATFORM FOR ROBOT BASED FPGA

Felipe Sanches Gurgel
Marcel Danilo Alves Siqueira
Marcos Banheti Rabello Vallim

Coordenação de Eletrotécnica – COELT
Centro Integrado de Pesquisa em Controle e Automação – CIPECA
Universidade Tecnológica Federal do Paraná — Campus Cornélio Procópio — UTFPR-CP
felipe-gurgel@hotmail.com, marceldanilo@gmail.com, mvallim@utfpr.edu.br

ABSTRACT

This paper describes the development of an educational robotics platform based on reconfigurable logic, with processing distributed hierarchical, flexible, modular and cost effective. The platform is designed to be applied on a project developed with students in public schools, which aims to foster the study of science and technology. The use of reconfigurable logic allows the construction of a evolutionary hardware, which is able to incrementally incorporate more sophisticated sensors and actuators in order to perform more complex processing.

1. INTRODUCTION

One of must import applications of computing is to solve complex problems of engineering and exact sciences, usually associated with factors such as, performance, response time, efficiency, fault-tolerance, among others. The solutions to these problems can be basically classified into two paradigms: Hardware solutions implemented in fixed (HF) and solutions implemented in reconfigurable hardware for Software (SW + HR) [1].

The main weakness of systems with fixed hardware is the low flexibility and sub-utilization, when they running several applications, since these systems have closed architecture and so, they don't be able to do adjustments and updates of the system.

On the other hand, systems with reconfigurable hardware by software have the functionality to adapt the needs of the system, allowing future changes, upgrades and expansions of the original design [2].

Among the technologies that emphasize flexibility and adaptability, the FPGA (Field Programmable Gate Array) is a reconfigurable hardware device that enables the replacement of part of your hardware to fit specific tasks, with a greater potential in terms of adaptability and performance [3]. This technology is suitable because can be applied to systems subject to major technological changes in a short space of time, is required demand for new tasks and performance, and the availability of new technologies for sensors and actuators [4].

This paper presents the aspects of implementation of a robotic platform for education, which is intended to be applied on a project, which aims to foster high school students from public schools to pursue the careers in science and technology. The platform is based on FPGA, and its basic requirements are open architecture, modular, user-friendly programming interface, and cost effective.

2. CONTEXT OF THE PLATFORM

The project "Ninho de Pardais" is a result of a partnership between the UTFPR-CP and the State Department of Education of the State of Paraná (SEED-PR), with financial support from the Research Projects Financing (FINEP). It is a project that involves all the bases of the fundaments of a university: research, teaching and extension.

The main goal of the project is the implementation of an experimental center for educational technologies, to foster the interest in science and technology, especially in the area of engineering, fostering the motivation of high school students from public schools in northern Paraná, in working with robotics.

The expected results are: built a center of multidisciplinary research in educational technologies; increase the demand for under degree programs of Technology and Engineering, offered by UTFPR-CP; improve the academic profile of students of Engineering and Technology programs of UTFPR-CP, attracting students from high school with excellent academic performance; provide the UTFPR-CP a framework of research in embedded systems, using educational robotics as a field of application, developing methods of teaching, using current and emerging technologies [5].

3. GENERAL CONCEPTS OF THE PLATFORM

The educational robotics platform is designed with three interdependent modules: software, structural pieces, and hardware. With the combined use of these modules, you can build different models of robots for many different types of challenges, enabling the use of the platform in mobile robotics competitions. The Figure 3a shows the block diagram of the modules of the platform.

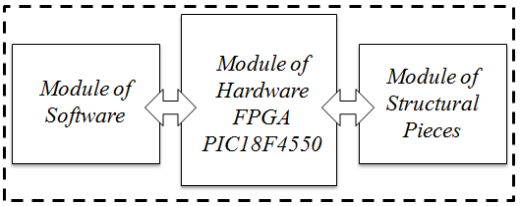

Figure 3a. Block diagram of the modules of the platform

3.1. Module of software

This module is responsible for programming the actions and movement of the s mobile robot, which is named Programming Environment. It is also responsible for the communication between the hardware (RCX) and Programming Environment, and the verification of the physical components of the robot.

3.2. Module of structural pieces

This module is responsible for the pieces that make up the physical structure of the robot, such as flat bars, bars for two and three planes, connectors, gears, angle, base, wheels and axles.

3.3. Module of Hardware

This module is responsible for the control of actions and tasks of mobile robot. It is implemented by hierarchical architecture of different levels of processing and control. From that, we have better distribution of tasks, including parallel processing. The parallelism shows a great advantage in these types of projects since it allows the construction of complex models closer to real models [6]. Figure 3b shows the block diagram to illustrate the architecture used. Thus, the different modules of software and hardware can be seen structured in independent blocks.

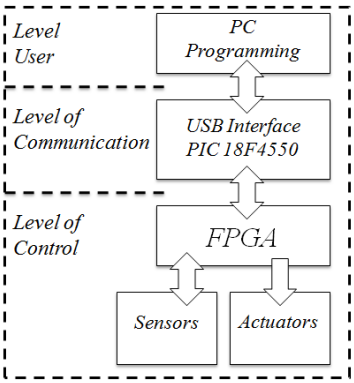

Figure 3b. The block diagram of implemented architecture

3.1.1. User level

This level is used to provide the parameters of operation to level control over the level of communication. At this level it is possible to manage and program the robot for performing various tasks according to the user needs, and the challenge presented.

3.1.2. Communication Level

This level performs the communication between the user and the control level. The data transmission between the user and the communication level is done through a USB interface using the PIC18F4550 microcontroller. The USB bus is managed by a central device, called the host device. The protocol used in the USB system is a HID class (Human Device Interface) [7]. There is a need for microcontroller storing a file descriptor, at least the information VID (Vendor ID) and PID (Product ID). This file is acquired in the Programming Environment MikroC, through Descriptor Generator.

The data transmission between the communication level and the control level is performed through an interface based on protocol UART (Universal Asynchronous Receiver Transmitter) implemented in FPGA and configured the PIC microcontroller. UART cores are commonly used as peripheral circuits for serial data transmission.

The implementation of the UART interface in the FPGA was performed by the hardware description language, VHDL (VHSIC Hardware Description Language) and is equipped with a transmitter TX and a receiver RX.

The start of data transmission happens when the start bit is detected, followed by the date and occasionally the bit parity bit, used for error detection. Data transmission is interrupted when the stop bit is detected. Figure 3c shows an example of transmitting a packet of 8 bits.

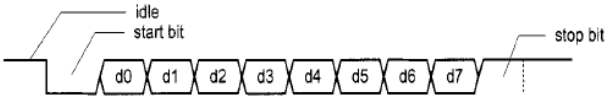

Figure 3c. Data transmission example

The organization of the receiver circuit consists of a UART receiver, a baud rate generator and a circuit interface. The core system, based on algorithms implemented in reconfigurable logic, receives data from the read port of the interface circuit based on a FIFO buffer (first-in-first-out). A FIFO buffer is an “elastic” storage between two subsystems. It has two control signals, *wr* and *rd*, for write and read operations. When *wr* is asserted, the input data is written into the buffer. The read operation is somewhat misleading. The head of the FIFO buffer is normally always available and thus can be read at any time. The *rd* signal actually acts like a “remove” signal. When it is asserted, the first item of the FIFO buffer is removed and the next item becomes available. FIFO buffer is a critical component in many applications and the optimized implementation can be quite complex. In this case, the FIFO buffer provides more buffering space and further reduces the chance of data overruns [8].

The UART receiver is responsible for writing the FIFO buffer for the main system can read it later. The

interface circuit performs two functions: provides a mechanism to signal the availability of a new word; provides an area of intermediate memory between the receiver and the main system.

The organization of the interface circuit of the transmitter circuit is similar to the receiver. The only difference is that the main system writes the FIFO buffer to the UART transmitter can read it later.

Combining the receiver with the transmitter, you can build a complete UART core. Figure 3d shows the block diagram of the UART core implemented on FPGA.

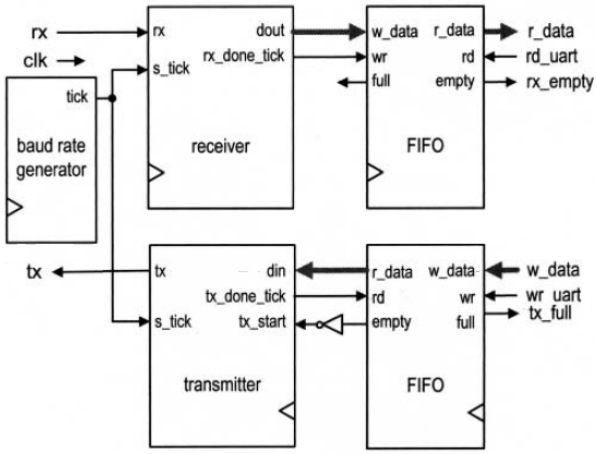

Figure 3d. The block diagram of complete core of implemented UART

The data received by the core UART is stored in a RAM (Random Access Memory) implemented in FPGA. Communication between the main system and RAM is implemented by a set of routines implemented in reconfigurable logic.

3.1.3. Control level

The tasks associated with this level are performed by algorithms implemented in reconfigurable logic. Through these, you get the communication between sensors and control actuators, given the need for adequate processing of information obtained from the external environment. Using the FPGA, the processing is done in parallel, allowing the processing time requirements of sensors and actuators are addressed in more significant. Figure 3e shows the block diagram of the level of control.

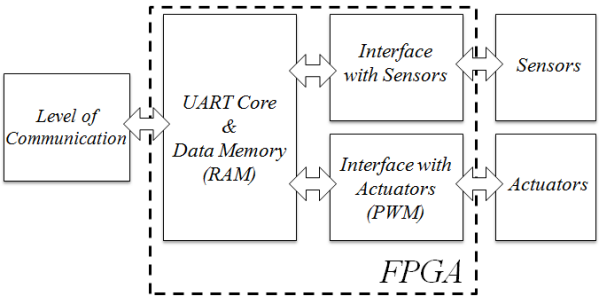

Figure 3e. The diagram block of the control level

The reading of the sensors is performed in parallel, and the information obtained in the external environment are shared with the memory data and control PWM (Pulse Width Modulation). Thus, the time to read the sensors and processing is minimized. As result, the local control works more efficiently, by example, when driving motors.

The control level is based on algorithms of logic circuits described in VHDL. The control actuators are based on PWM technique implemented in FPGA through hardware description language, VHDL.

The PWM has been implemented using sequential logic circuits. Figure 3f shows the block diagram of the circuit logic applied.

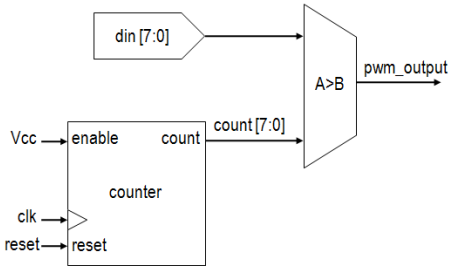

Figure 3f. Implemented PWM Device in VHDL

For development of the comparator, 8 bits, is applied a Boolean function, where A> B if, and only if:

$$Z(An-1,Bn-1)+Q(An-1,Bn-1)Z(An-2,Bn-2)+$$
$$+Q(An-1,Bn-1)Q(An-2,Bn-2)Z(An-3,Bn-3)+...+$$
$$+Q(An-1,Bn-1)Q(An-2,Bn-2)...Z(An-8,Bn-8)=1$$

Knowing that:

$$Q=\overline{A}\overline{B}+AB$$

$$Z=A\overline{B}$$

It gets the following Boolean equation:

$$(A7\overline{B}7)+[(\overline{A}7\overline{B}7)+(A7B7)(A6\overline{B}6)]+$$
$$+[(\overline{A}7\overline{B}7+A7B7)(\overline{A}6\overline{B}6+A6B6)(A5\overline{B}5)]+$$
$$+...+[(\overline{A}7\overline{B}7+A7B7)...(A0\overline{B}0)]=1$$

Thus, applying the concept of obtaining circuits by Boolean expressions, it was a logical circuit later described in VHDL. Figure 3g presents the simulation of the circuit designed using the binary word **"1000000"** in the duty cycle, while Figure 3h shows the simulation of the circuit using the binary word **"10111111**" in the duty cycle. Both simulations were performed using the changed environment ***ALTERA*** Quartus II.

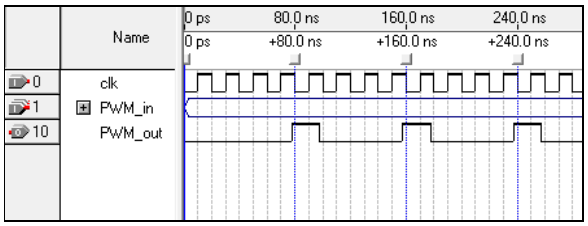

Figure 3g. PWM simulation with binary word "1000000"

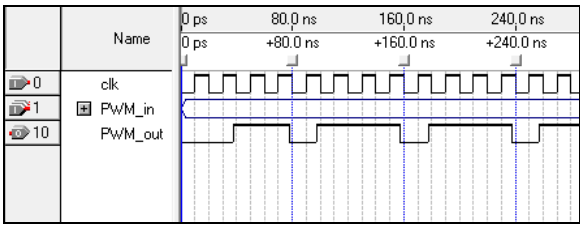

Figure 3h. PWM simulation with binary word "10111111"

Figure 3i shows the implementation of the architecture developed using two boards of Development: ***ALTERA*** DE2 board with Cyclone II FPGA and, ***EXSTO*** NEO201 board with ***MICROCHIP*** PIC18F4550.

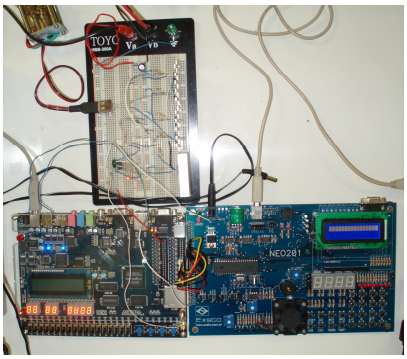

Figure 3i. Prototype of implemented architecture

4. CONCLUSION

In this paper was presented the aspects of implementation of an educational robotics platform for use in a research project extension education, aimed to foster motivation on high school students to pursue careers in technology end science.

The main goal of the platform is to provide a flexible, open architecture, modular, user-friendly programming interface, and cost effective, means of learning in the education robotics for the public schools. The general concept of the platform is to implement a hierarchical architecture, which was appropriate, making possible the modernization of the final design and construction of prototypes complex, due to the possibility of processing in parallel, supported by FPGAs. Moreover, these prototypes can be built using simple devices, significantly reducing the final cost of the project.

5. ACKNOWLEDGMENTS

This work was supported by the Financiadora de Estudos e Projetos – FINEP (Research Projects Financing), subordinated to the Ministry of Science and Technology (MCT), under grant Number 4971/2006-FINEP. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the MCT and the FINEP.

It also has the support of the Foundation for Support of Education, Research and Scientific and Technological Development of UTFPR (FUNTEF) and Araucaria Foundation for Support to Scientific and Technological Development of Paraná.

6. REFERENCES

[1] L.A. Casillo, "Projeto e Implementação em FPGA de um Processador com Conjunto de Instruções Reconfigurável utilizando VHDL," *Dissertação de Mestrado*, Universidade Federal do Rio Grande do Norte, Natal, 2005.

[2] T. Miyazaki, "Reconfigurable Systems: A Survey," *Proceedings of IEEE Design Automation Conference*, ASP-DAC '98, pp. 447-452, 1998.

[3] L. Rocha, H.S. Lopes and C.R.E. Lima, "Um Ambiente para Apoio ao Aprendizado de Automação e Controle e sua Aplicabilidade no Desenvolvimento de Competências e Habilidades," *XXXV Congresso Brasileiro de Educação em Engenharia*, Curitiba, 2007.

[4] F.S. Gurgel and M.B.R. Vallim, A Robotic Platform Based on FPGA for Educational Applications, *VIII Microelectronic Students Forum*, Rio Grande do Sul, 2008.

[5] M.B.R. Vallim, "Projeto Ninho dos Pardais: Atraindo Estudantes para Carreiras na Área Tecnológica," *Workshop Robocontrol'08*, Bauru, Brasil, 2008.

[6] S. Kilts, *Advanced FPGA Design: Architecture, Implementation and Optimization*, New Jersey, 2007.

[7] D. Ibrahim, *Advanced PIC Microcontroller Projects in C: from USB to RTOS With the PIC18F Series*, Elsevier, Oxford, 2008.

[8] P.C. Pong, *FPGA Prototyping by VHDL Examples*, Wiley-Interscience, New Jersey, 2008.