CHIP design and implementation course through a University-Industry program

Edson Lemos Horta, Orlando Del Bianco Filho and Alessandro La Neve

Department of Electrical Engineering
FEI – Faculdade de Engenharia Industrial
Av. Humberto de A. C. Branco, 3972
São Bernando do Campo – SP – CEP 09850-901

***Abstract -** This paper describes a university-industry joint program that was signed between FEI and ALTERA. This company produces EPLD's (Erasable Programmable Logical Devices), a kind of chip that enables engineers to develop a project based on Reconfigurable Logic. The paper explains briefly what Reconfigurable Logic is and which disciplines use it at FEI. After this, the disciplines chosen to use it are explained in detail. Then, it is shown how the university program supported by ALTERA works, in each one of its phases, how the laboratories at FEI were prepared to receive the material supplied by ALTERA and how they are being used now. Some results obtained with the program, such as a final term and graduate projects, are presented in the last part of the article. In the conclusions the continuity of the program is discussed for its importance for both industry and university, and some suggestions are given to improve it.*

Introduction

FEI, which is located in S.Bernardo do Campo, the most active and important industrial area in South America, is a technological center for engineering, with its main focus on industry, development and production. It has over 5,000 students, 300 faculty members, 232,000 m^2 campus area, 41,000 m^2 built area and an important Integrated Computer Center-CCI, with over 500 last generation microcomputers, IBM RISC workstations, peripheral equipment, spread all over the campus through an intranet (FCA-net); connections are implemented through fiber optics, and transfer rate is at 100Mbit rate, and communication to the outside world can be realized through Internet.

At CCI there are over 300 different software titles with network license or different sites capabilities. The titles include software for use and development, such as high level and object oriented languages, CAD/CAM/CAE packages, simulation, computer graphics, data base, communications, production and software engineering tools. Cooperation contracts and accords are held with some of the major software and hardware manufacturers, such as Microsoft, Novell, IBM, Oracle, Bay Networks, through which we can have all of the products easily on campus. In this way it possible to train on one side, students and engineers and, on the other one, test and tropicalize their products and give technical support to their clients, through development and training.

Courses at FEI are offered in Mechanical, Civil, Metallurgical, Chemical, Electrical, Production and Textile Engineering. As for Electrical Engineering three different emphasis are offered, namely Electronics, Computer and Info-Telecommunications. In the constant process of upgrading courses, especially for those which deal with technology, partnership with industry is pursued and realized; in this category there was a need for use of reconfigurable architectures, in the disciplines where digital systems are taught, and are fundamental for computer architecture and microprocessor applications.

Programmable Logic Devices

It is for this reason that the Department of Electrical Engineering decided to apply to Altera, which produces a digital component named EPLD (Erasable Programmable Logic Devices) and is one of the most important companies in the market of programmable logical devices. This component can be better understood as a device made up of a large number of combinatorial cells and a set of flip-flops. It is the result of an evolution , started with the first generation of Programmable Devices, which was intended to accommodate a lot of combinatorial gates and a few flip-flops. They were named PLA (Programmable Logic Arrays) and the main objective of such devices was to replace a large quantity of discrete SSI - Single Scale Integration devices, like AND, OR and NOT gates, opening up space in printed circuit boards. Basically, those devices offered some AND, OR and NOT cells that could be arranged according to each one's needs. The second generation presented some new devices, the FPGA - Field Programmable Gate Array, that, in principle, were big PLA's. They also presented some elementary building blocks. It is now possible to identify a third generation of programmable Devices, that Miller calls SPGA - System Programmable Gate Arrays [1]. These new devices have more than 50.000 programmable gates, which enables system-on-a-chip design and their internal architecture provides for predictable timing, which makes these chips ideal for re-use applications, and, most interesting, their current price-per-gate is quite affordable, even for low volumes.

At the beginning these components used PROM (Programmable Read Only Memory) technology, which means that they could not be erased after they had been programmed. The implications are obvious: if an error was detected, the device was no longer useful and a new device should be programmed. Later on EPROM's - Erasable Programmable Read Only Memory technology came in so that components could be erased, mainly through UV-Ultra Violet beams and chips could be re-programmed. EEPROM's - Electrically Erasable Programmed Read Only Memory became also an interesting option since they could be erased with electrical pulses, on printed boards, retaining the program when the power supply was turned off. Finally a family of new devices, based on RAM-based cells, came up and they can also be programmed on printed boards, but they retain the information only as long as a power supply is on.

Since a digital project is a specific arrangement of combinatorial cells and flip-flops, to implement a specific function, an EPLD can assumed to accommodate virtually any digital project.

Altera university program

In order to establish the necessary arrangement of the combinatorial cells and flip-flops efficiently, a software package by Altera, named MAX+PLUS II, can be used, since it implements the necessary arrangements, automatically, considering the project description supplied by the designer. MAX+PLUS II belongs to a class of software known as EDA (Electronic Design Automation) tools. With this software it is possible to start a circuit project right from the beginning, then going through its design, simulating it as to logic and timing and finally programming an EPLD device. The behavior of the device will follow exactly what was designed and simulated with MAX+PLUS II.

Altera university program intends to offer undergraduate students the opportunity to put their "hands-on" MAX+PLUS II during their academic activities. This requires that some phases be established and the progress from one phase to another implies in results and reports to be presented to and approved by Altera. By the time FEI entered the university program, there were 3 phases that should be attended. By now, as a new strategy of the company, phases 2 and 3 have been grouped in a single one. To enter the academic program, the university must describe the undergraduate courses in which MAX+PLUS II will be used, including number of students, sequence of disciplines and the hardware/software facilities available at the university that may be accessible to the students.

During the first phase, a single copy of MAX+PLUS II is supplied to the university, free of charge. This copy is intended to put professors, researchers and students in contact with this technology. At the end of phase 1, which lasts no more than 8 months, the university sends a report to Altera, describing the effective usage of MAX+PLUS II. If the report is accepted, the university enters phase 2, which is the last part of the program. In this phase Altera offers the necessary number of MAX+PLUS II copies for all PC's Pentium or Workstations available at the university laboratories. After this, the university has an 8 month period to work on and to send a new report to Altera. This is fundamental for to the maintenance of the program, and getting new software versions, besides application notes and new available software modules. At FEI a two step program was established as follows:

a) First step: only the disciplines in the computer course would be involved, that is Digital Circuits III and Computer Architecture. It was decided that all efforts would be made in these disciplines, in order to reach phase 2 successfully.

b) Second step: initially planned to begin after phase two had been reached, it would involve the "info-telecom" course, the "final term project" discipline and an "application research" in a graduate course. This second step had to be anticipated, due to the success that the available software had with both students and professors.

The Digital Circuit III Discipline

In the first part, when the discipline was offered for the first time, two experiences were given to the students, to make them familiar with MAX+PLUS II. Both of them were based on a project of a digital system, whose function was to store two "four-bit" numbers, compare them and present the greater one on an output bus. Figure 1 illustrates this system.

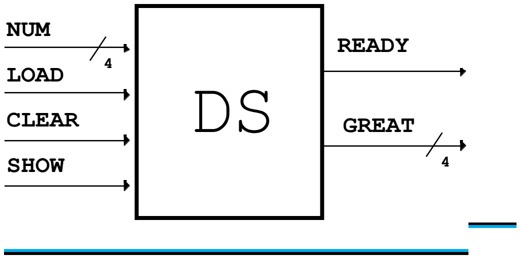

Figure 1

The Digital System (DS) worked this way :

a) The CLEAR signal was activated to set the outputs to zero;

b) The first number was presented on the NUM bus;

c) The LOAD signal was activated to store the first number;

d) The second number was presented on the NUM bus;

e) The LOAD signal was activated to store the second number;

f) The SHOW signal was activated to compare the numbers and present the greater one on the GREAT bus;

g) When the greater number was on the output bus, the READY signal would be activated.

The first experience had two parts. In the first one, the students had to project the datapath of the DS above and implement it using the graphic editor from MAX+PLUS II. In the second part they had to project the control unit of the same DS and implement it using AHDL to describe an ASM (Algorithm State Machine). There was no simulation at this time. The students compiled the datapath and the control unit in two EPLDs of a classic family. Each part of this experience was done in a three-hour class and the students worked in groups of four.

The second experience also had two parts. In the first one, the students had to use the datapath and the control unit symbols generated in the first experience to project the whole digital system. They used the graphic editor to do this task. In the second part they would have to do the simulation of the digital system but, as there were not have enough hard-locks to do this, they only drew the waveforms on a sheet. After this, the teacher showed them, with the aid of a data-show, how to simulate the digital system, using the waveform editor and the simulator of MAX+PLUS II. Each part of this experience was done in a three-hour class. This course was given to 75 students divided in three classes and the experiences above represented 50% of it.

In the second part of the course, the students had to use MAX+PLUS II to project four digital systems. The first and second projects were done as exercises during class and the third and fourth ones were done as final projects, with a final report each. During classes the student were instructed how to use the Graphic Editor (projecting data-path of a digital system, generating symbols for the data-path and projecting the entire digital system using these two symbols), the Text Editor (projecting an generating the symbol of the control unit, using the AHDL) and the Waveform Editor (simulating the digital system).

The first project was a system that had to load a four bit number, after pressing the LOAD key, and calculate its double each time the DOUBLE key was pressed. The students got the data-path and a description of the control unit through a Petri Net Graph. They had to describe the data-path in a schematic file, the Petri net in a text file, compile these two files, generate two symbols for each one, join these symbols in a second schematic and compile this schematic, that's the complete digital system. Figure 2 illustrates this system.

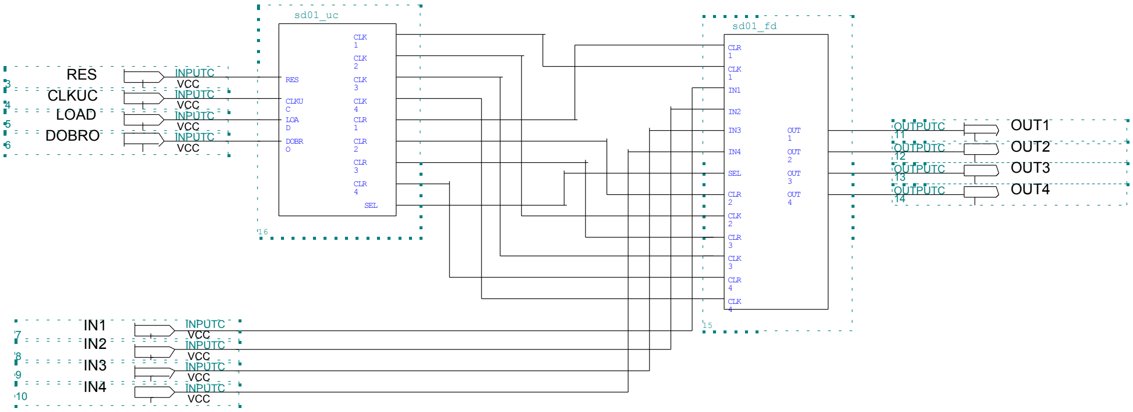

Figure 2

The second project was a system that had to load four "two digits" numbers using four keys (N1, N2, N3 and N4). After this, when one of the keys M1, M2, M3 or M4 was pressed, the corresponding loaded number had to be shown on the output. Figure 3 illustrates this system.

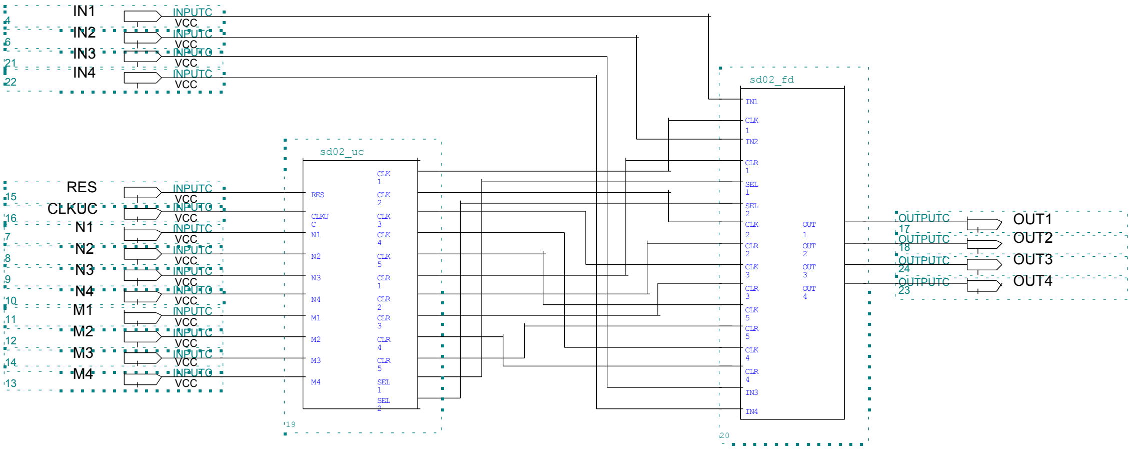

Figure 3

The third project was a system that had to load three numbers of four digits, pressing the LOAD key for each one. After this, pressing the CALCULA key, the system had to calculate a number, according to the formula bellow, show the result in the output lines and set the output PRONTO to “one”.

N1>N2 => OUT = N1+N3
N1<N2 => OUT = N2+N3
N1=N2 => OUT = N2+N2

Figure 4 illustrates this system.

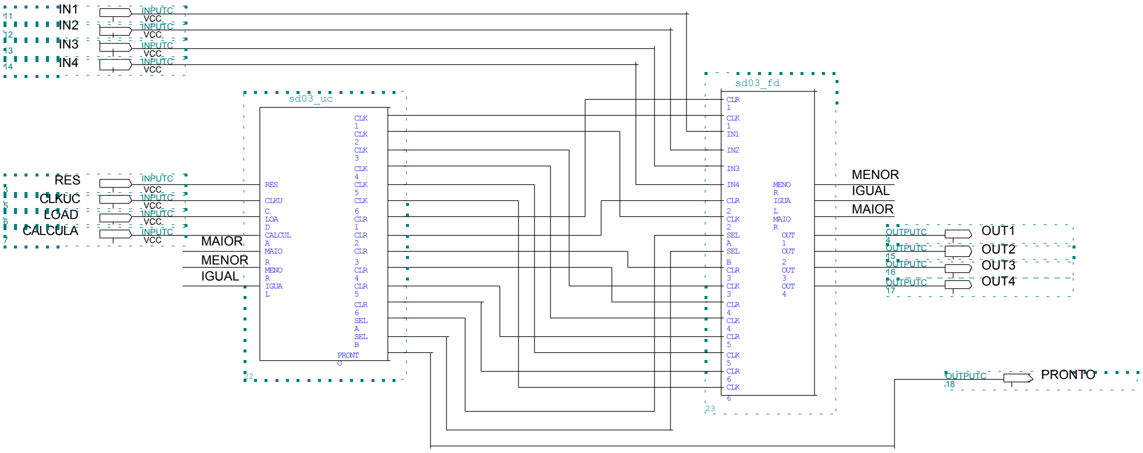

Figure 4

The fourth project was a system that had to load three numbers of four digits, pressing the LOAD key for each one. After this, pressing the CALCULA key, the system had to calculate a number, according to the formula bellow, show the result in the output lines and set the PRONTO output to “one”.

IF N3 = 0 :
OUT = N1 + N2

IF N3 <> 0 :
N1>N2 => OUT = N1+N3
N1<N2 => OUT = N2+N3
N1=N2 => OUT = N2+N2
Figure 5 illustrates this system.

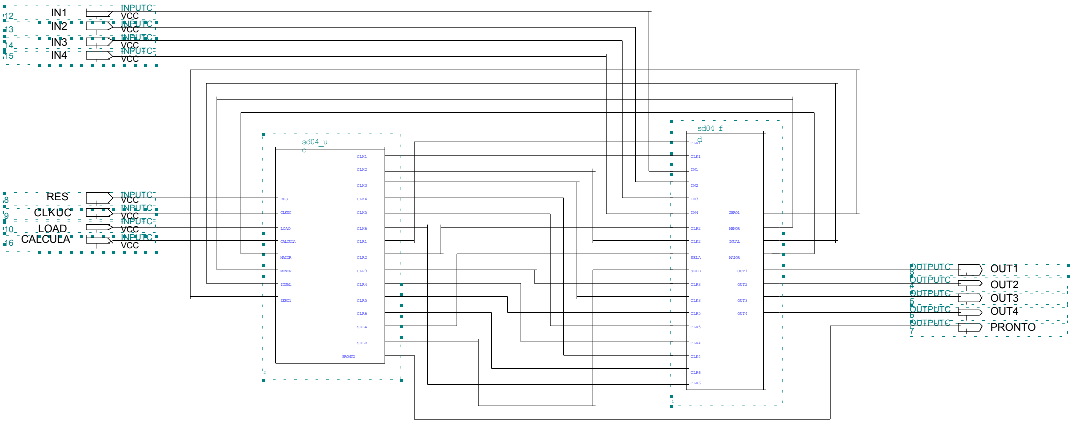

Figure 5

The first and the second projects were given to all the students (four classes of 20 students each). They had to design and compile the systems individually. The third one was given to the first class and the fourth one, to the other classes. They worked in groups of four students and had to compile, simulate and test the system on a proto-board, with EP7096 devices.

The Computer Architecture discipline

The discipline is divided in two parts, that is:

a) the first one is "theoretical", and discusses the internal organization of a computer and the techniques used to improve performance in a "standard" machine. This includes the utilization of interrupts, characterization, advantage, costs (in terms of growing in circuitry) and penalties for the usage of DMA (Direct Memory Access) and CACHE sub-systems; some new machines like RISC's (Reduced Instruction Set Computers), Superscalar and VLIW (Very Long Instruction Word) are also studied in this part of the course;

b) the second part, which is done in laboratory, requires that the students participate in all of the phases that projects generally go through, to get the final circuit.

There are about 40 students per semester with 2 regular lab classes with 20 students each, and they are divided in groups of 2. The objective of each group is to design and simulate some circuits that are proposed to them. The circuit description has some parameters, based on the students' registration number, so, in practice, each project is different from one another. With this in mind, 2 themes were proposed to each group. The themes were handed out with a textual description of the circuit, like" the objective of theme number 1 is to get a divide by 7 circuit". After the students understood the goal, they had to create a block diagram for the circuit. With the circuit described in terms of blocks, the required inputs, outputs and the internal connections among the blocks, it was possible to get all the benefits offered by MAX+PLUS II. The sequence of steps is as follows:

1) the circuit must be described in terms of its elementary components: logical elements (AND, OR, NOT gates, decoders and so on and also with the memory elements (flip-flop, register, shift registers etc.). This activity can be developed with MAX+PLUS II in a conventional graphical representation or by the textual behavior of the circuit. MAX+PLUS II accepts textual description with the VHDL (Very high speed integrated circuit Hardware Description Language) language [3], a standard circuit description language or with the AHDL (Altera Hardware Description Language), that can be seen as a simpler, although efficient, circuit description language created by Altera [2].
2) a compiler is initially invoked to check for the correct the circuit description, which means absence of short circuits or floating outputs. After the circuit description is checked, the compiler chooses the minimum number of components, in which the circuit can be fitted. This activity can be avoided if the designer specifies which particular device should be adopted in the project. In real life situations, these option enables to choose some components normally used by the company. The use of a limited set of devices reduces the number of component types that must be bought, and minimizes the stock items that must be managed, among others benefits. The compiler also prepares the programming file, which contains the cell descriptions, and the interconnections that will be programmed inside the selected component.
3) The designer can now create a file that contains the possible expected time behavior of the circuit input signals; a simulator program, within MAX+PLUS II, combines this input behavior file with the

programming file generated by the compiler and generates the sequence of output signals, according to the project. The designer can then compare the expected output signals with the ones obtained by the simulator. If there are no differences, the project is correct. With some commands issued to MAX+PLUS II it is possible to program a component that contains the desired circuit.

Figure 6 shows the steps covered by MAX+PLUS II during the development of a project. Continuous lines show the regular flow of a project; the traced lines represent the baptism of the project to the MAX+PLUS II software; the feedback lines indicate the eventual corrections that must be made until the correct circuit is obtained.

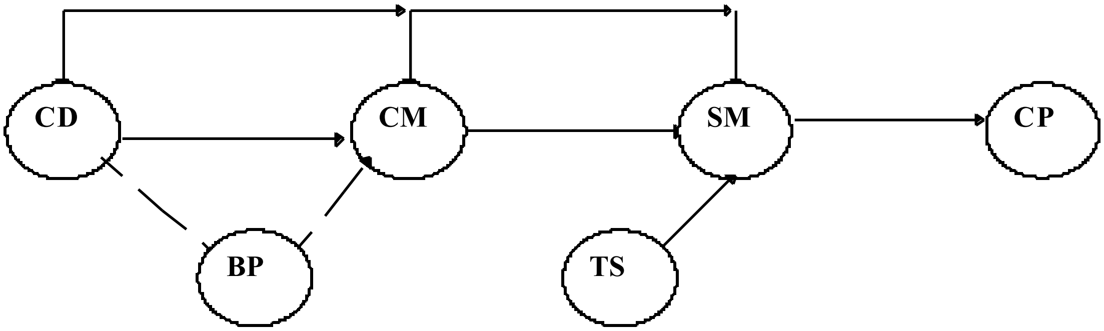

Figure 6: Flow of a project with MAX+PLUS II software

CD: Circuit Description (graphic or text)
CM: Compiler
BP: Baptism of the project
SM: Simulator
TS: Time behavior of the input signals
CP: Circuit Programming

As each project has a specific name, it is possible to treat these elements as stand-alone sub-systems to be used by bigger projects. All sub-systems are stored in the MAX+PLUS II library of functions, which can be accessed at any time. All projects can be individually compiled, and simulated, before being incorporated to the library. The themes proposed to the students of Computer Architecture are, in fact, associated with computer sub-systems like simple ALU, 32 bits BCD adder/subtractor, event driven interrupt generator, data path control, memory addressing circuit, interrupt generator with priority and so on. The course is organized in 4 classes (2 hours each) dedicated to discuss the proposed themes and to review the flow of a project using MAX+PLUS II and the rest of the course, about 8 classes of 2 hours each, are free to the students so that they can develop the proposed themes. Besides the activities scheduled in class, the students can use of MAX+PLUS II in the Computing Center, which is open to work and for support from 7:00 AM to 10:00 PM. A minimum of 8 hours per week is reserved for each group.

Conclusions

A very interesting characteristic of the program is that Altera supplies the Student Version of MAX+PLUS II for universities already engaged in the program. This version of MAX+PLUS II works without a hardkey protection element. The students can install this software package in their private micros and work out of the classes. Although this is a limited version of MAX+PLUS II (all functions are available only for the MAX EPM 7128 and FLEX EPF 10K20 devices), more than a third of the students install this package in their own micros. In general, the majority of groups conclude the proposed themes completely and only some of the groups faced problems in simulations. As described earlier a positive response from our students was observed.

In the Digital Circuits III discipline, with standard experiments to be implemented during classes, most of the students could learn how to use MAX+PLUS II and all experiments were concluded. The difficulties within this discipline were related to the wrong configuration of software drivers.

In the Computer Architecture discipline, most of the students installed the student version of MAX+PLUS II in their private micros. The classes were used to discuss possible alternatives to implement the desired circuits with the professor, and about 75% of the teams implemented their circuits. A great and welcome surprise was that MAX+PLUS II was used by other disciplines, which were not initially supposed to be involved along phases 1 and 2 of Altera academic program.

In the Final Term Project discipline one of the teams decided to use Altera components in the 8 bit Programmable Data Acquisition ISA compatible board. The project used a FLEX EPF 8282ALC84-4 device, which should be programmed via the ISA bus or via the serial interface available in the PC. A set of buttons and LED's would simulate the input and output data from the outside world. The way the FLEX should operate was determined by software. The data read from the buttons were presented on the PC video. The necessary response from the algorithm was presented on the PC video and with LED's, simulating a command of a specific control device. This project was successfully concluded by the students.

Finally it is worth while mentioning the work developed by one of our graduate students, who has been using Altera since phase 1 was initiated [6]. His main objective was to perform data compression/decompression of high resolution Digital Video in real time, using mathematical functions implemented in hardware. The interest for Altera reconfigurable devices is due to the fact the parameters used to compress/decompress the signal can be changed at any time according to the complexity of the signal that has to be compressed. For this project it was very useful the support offered by Altera, and in particular the application notes that deal with the FIR (Finite Impulse Response).

A very interesting and important aspect of this program is the continuous ALTERA support, by which it is possible to have application notes, specific software modules and even component samples for the prototypes, in a very short time and at no expenses.

This joint program has enriched our Computer Engineering course a great deal, putting students both in contact with high-tech devices and recent state-of-the-art technology in this field, and also allowing them to create and implement their own dedicated chips. This is a great advantage for their professional career, and for industry as well, since they receive well-trained engineers who will help them to be competitive and profitable.

Bibliography

[1] Miller, W., "Designing with System Programmable Gate Array", *Electronic Engineering*, January 97, pp.59-62.

[2] ALTERA MAX+PLUS II AHDL Manual, 1996.

[3] Mazor, S. & Langstraat, P., "A guide to VHDL", second edition, Synopsys, Inc, 1993.

[4] Del Bianco, O.Filho, Sanchez, P.L.P. "Reconfigurabilidade: Alternativa para Manutenção de Compatibilidade entre Gerações Diferentes de Máquinas VLIW". WAR'97, SBAC-PAD'97 Campos do Jordão, SP, 8/10/97.

[5] Horta, E.L., Kofugi, M. "RECATS: a Reconfigurable ATM Witch Proposal". WAR'97, SBAC-PAD'97 Campos do Jordão, SP, 8/10/97.

[6] Melo, A.de, La Neve, A. "Processador Reprogramável para Compressão e Descompressão PRCO, em Tempo Real de Imagens de Vídeo Digital". WAR'97, SBAC-PAD'97 Campos do Jordão, SP, 8/10/97.