LEARNING THE RELATIONSHIP BETWEEN COMPUTER ARCHITECTURE AND TECHNOLOGY BY RECONFIGURING

Domingo Benitez

University of Las Palmas G.C.
Departamento de Informática y Sistemas
35017 Las Palmas G.C.(Spain)
E-mail: dbenitez@dis.ulpgc.es

ABSTRACT

This paper proposes an approach for teaching Computer Organization and Architecture which is based on building knowledge from the bottom up. Students should design three processors with increased complexity and measure their performances. These processor designs are assigned during a sequence of three 15-week courses and are implemented using a low-cost FPGA-based reconfigurable platform developed at University of Las Palmas G.C. Emphasis is placed on comparing the relation of computer performance with hardware requirements to what has been experimented during the recent history of computers. Our experience shows that students understand better the architecture-technology relation and gain a sense of accomplishment on the computer design when given the opportunity to use real hardware. It is demonstrated here that the learning curve can be modelled as an exponential function of time.

1 INTRODUCTION

Learning by doing is a teaching methodology that is used at many universities ([Hen-96], [HOYD-96]). Rapid prototyping has been applied in this methodology since the equipments required are available to schools at prices comparable to existing instructional laboratories. We try to teach the relationship between computer architecture and technology by using FPGA devices. Our approach is based on the following two experimental rules.

Rule 1: Microprocessor performance has grown at an annual rate about 54% [PH-98]. Modeling the variation of computer performance with time, it can be expressed as:

$$P(t) = 0{,}65\, e^{0{,}43\, t} \tag{1}$$

where P is performance, t is time measured in years, and we suppose that P(1)=1.

Rule 2: Moore's law says that the number of transistors on a processor doubles approximately every 18 months [Yu-96]. This is equivalent to say that the number of transistors grows at an annual rate about 59%. The following formula models this temporal rate,

$$X(t) = 0{,}63\, e^{0{,}46\, t} \tag{2}$$

where X is the number of transistors, t is time measured in years, and we suppose that X(1)=1.

Combining the previous two laws,

$$P = X^{0{,}93} \tag{3}$$

Expression (3) can model the average relation between processor performance and hardware complexity experimented during the recent history of computers. This function is a main point in our education methodology. Students experiment with a function similar to (3) while learning computer architecture concepts.

Computer Organization and Architecture education in the Computer Engineer curriculum at University of Las Palmas G.C. is divided into three semester-based courses taken in sequence at the beginning of three consecutive years. Students have laboratory assignments for each of these courses in which they design a processor with increased hardware complexity and computer performance (see fig. 1). These processors are grouped into a family called CEREPRO.

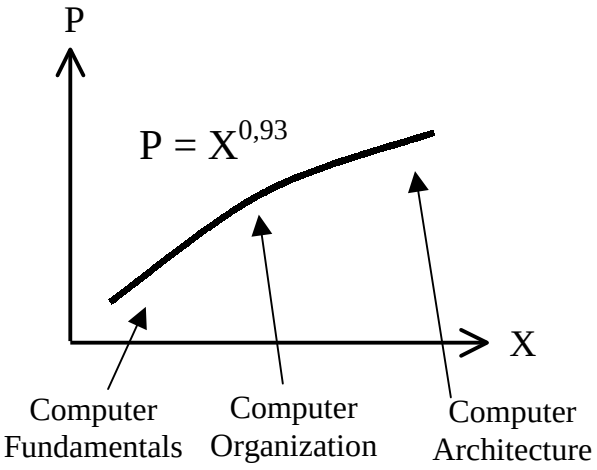

Figure 1: Relationship between computer performance and number of transistors in the recent history of computers in correspondence with the sequence of Computer Engineering courses.

This paper describes in section 2 general characteristics of the family of processors and the low-cost reconfigurable board used in their implementations. Section 3 describes the sequence of undergraduate courses where processors are designed, and finally some conclusions are exposed in section 4.

2 CEREPRO PROJECT

CEREPRO is an acronym made up from "CEntral REduced PROcessing unit". This is the name of an educational project based on the development of processors using programmable devices. This processor set is grouped into a family called CEREPRO which is formed at this moment by six processors. Each processor is differentiated by its instruction set architecture which can be classified by the type of internal storage. The major choices are an accumulator or a set of registers. Table 1 gives a brief description of the processors.

Table 1. Characteristics of the CEREPRO processors.

CEREPRO's processor	Architecture	Number of instructions/ Instruction bits	Addressing modes	Data/ Address bus	Number of registers/ bits	Pipelined data path	Normalized Performance	Normalized number of gates
CEREPRO-0	Accum	12/{8,16}	Dir, Imm	8/8	1/8	No	1,00	1,00
CEREPRO-1	Registers	20/32	Reg,Imm,Dis	16/16	8/16	No	8,57	3,37
CEREPRO-2	Registers	10/{8,16}	Dir, Reg	8/8	8/8	No	0,9	1,16
CEREPRO-3	Accum	23/{8,16}	Dir, Ind	8/12	1/8	No	3,39	4,11
CEREPRO-4	Registers	13/16	Reg,Imm,Dis	16/16	16/16	Yes	32,62	5,51
CEREPRO-5	Registers	13/16	Reg,Imm,Dis	16/16	16/16	Yes	19,47	5,12

Accum: Accumulator, Dir.: Direct, Ind.: Indirect, Imm.: Immediate, Reg.: Register, Dis.: Displacement.

Each processor has been synthesized onto the same programmable device from Altera using MAX+plus II software [Alt-96]. The results of the synthesis process are depicted in figure 2 supposing that the number of transistors (X) is obtained by multiplying the number of gates by a constant factor. Computer performance (P) is measured by running a multiplying algorithm with the same input data and calculating the inverse of CPU time. Performing power regression calculations, it can be obtained the following formula for modeling computer performance in CEREPRO family (see fig. 2):

$$P' = 0{,}64\ X^{1{,}81} \tag{4}$$

This is equivalent to say that the computer performance growth would increase at an annual rate of 130% if Moore's law is considered to be the temporal model of hardware complexity.

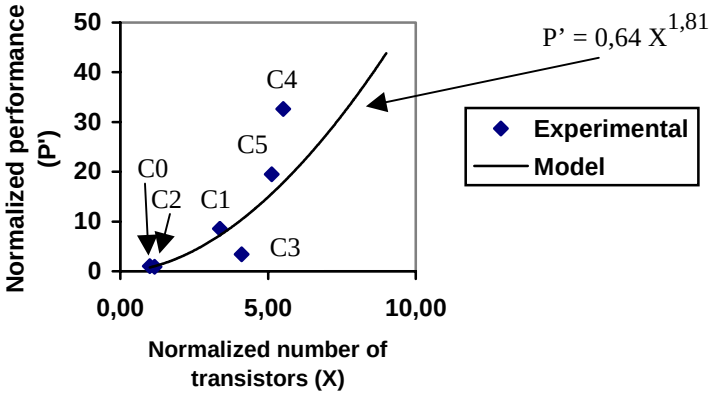

Figure 2: Relationship between computer performance (P') and number of transistors (X) found out in CEREPRO processors.

We have developed a low-cost prototyping platform which allows the design and implementation of all CEREPRO processors (see fig. 3). It is constituted by a programmable device from Altera (EPF8820AGC192-4), two 2KB 4-port memory modules from IDT (IDT7052S35G), 64 KB EPROM (two modules of 32Kx8, 27C256-20), and 64 KB high speed SRAM (two modules of 32Kx8, TC55328-15). This board has 3 I/O ports: 1 full-duplex serial port with programmable transmission rate, 1 serial download port, and a 40-bit configurable bidirectional port.

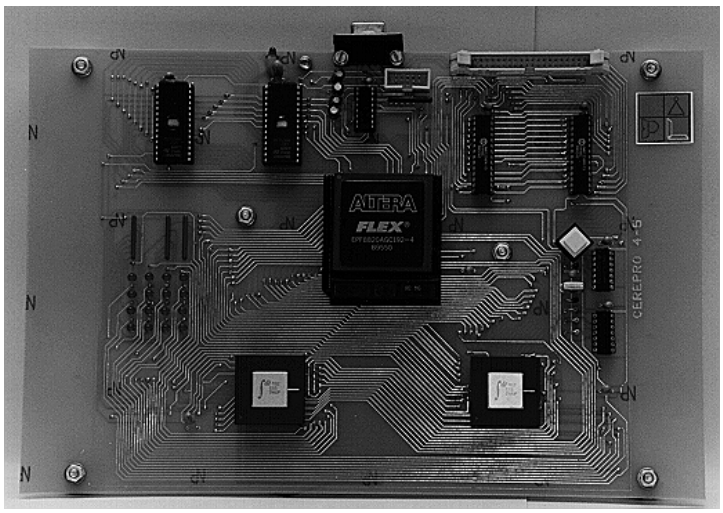

Figure 3: The reconfigurable prototyping platform.

Our reconfigurable prototyping platform is not intended to be used only by one course on processor or computer design in comparison with the purpose of other educational boards [VKSJ-96]. It is used by several courses and therefore, its cost can be considered by restricted budgets. The cost of each board is about 400$.

Some professors have found that around 10.000 gates in a RISC data path is a reasonable target instead of higher number of gates contained in other general-purpose reconfigurable development systems [HOYD-96]. Our board is based on a 8.000 gates programmable device and our experience shows that this is a good number of gates for a wide range of educational objectives.

3 UNDERGRADUATE COURSES FOR COMPUTER ARCHITECTURE EDUCATION

Rapid prototyping design laboratory can demonstrate and unify many of the ideas taught in numerous undergraduate classes. This type of laboratory is ideal for students to learn the basic processor design skills as has been shown in the literature ([HOYD-96], [VKSJ-96]). We use this type of laboratory for students to put into practice computer architecture concepts.

A sequence of three 15-week courses on Computer Architecture are required in the Computer Engineer curriculum: Computer Fundamentals, Computer Organization, and Computer Architecture. These courses use a main book which are [MK-97], [PH-98], and [HP-96] respectively. Each course has a major assignment in which a processor from CEREPRO family is designed and implemented using our reconfigurable board.

Computer Fundamentals is required 5 hours, one quarter. Students are already familiar with digital design, Field Programmable Gate Arrays, and modeling and simulation of simple digital logic circuits from a prerequisite course called Digital Systems. Lectures are held 3 hours a week, and 2 hours of laboratory work per week are required. Computer Fundamentals covers state machine design and an introduction to computer organization. The final design example is a very simple computer based on CEREPRO-0 [Ben-98]. This is a 8-bit processor with instruction set architecture based on an accumulator (see table 1). Students develop a schematic with simulation using a modern digital CAD tool, synthesize onto the FPGA which is contained in the prototyping board, and develop a benchmark program that multiplies two 8-bit numbers. Metrics for the design include gate count and total execution time.

Computer Organization is required 4 hours, one quarter. The goals of this course are to introduce students to computer arithmetic, RISC-like processor design, memory hierarchy, and I/O. At the end of Computer Organization, students should have a basic understanding of computer operations from the high level language programming level to the gate level implementation of the computer system. This course has a major assignment in which CEREPRO-1 is designed and implemented using our reconfigurable board. CEREPRO-1 is a 16-bit processor with a load-store architecture and multi-cycle data-path. Its organization is a little bit more complicated than CEREPRO-0's (see table 1). Students complete a machine language program which multiplies two numbers. Finally, they compare the gate count and total execution time with the results obtained in the synthesis process of CEREPRO-0. So, the complexity of the processor design is increased as student experience grows (see figs. 1 and 2). VHDL language has been introduced into the laboratory work, and we have experienced that this high level language allows processor prototyping to be relatively rapid.

Computer Architecture requires 2 hours of lecture and 2 hours of laboratory work per week. This course focuses on advanced uniprocessors, including microarchitecture and instruction level parallelism. A major assignment is required in which students develop a processor called CEREPRO-4 with pipelined datapath as described in [PH-98] and [HP-96]. It is the most complex of CEREPRO family and implements a subset of the DLX machine with 13 instructions (see table 1). As in earlier courses, students measure gate count and execution time and compare these results with those obtained previously.

These three courses are taken in the first semester of three consecutive academic years at the University of Las Palmas G.C. So, the hardware complexity of processor designs assigned in the respective laboratory work increases as:

$$X(t) = 0{,}48\, e^{0{,}85\, t} \tag{5}$$

which is equivalent to a factor "2,34*x*" per year (134%/yr). On the other hand, the performance of successive processors grows as

$$P(t) = 0{,}2\, e^{1{,}74\, t} \tag{6}$$

which is equivalent to "5,7*x*" per year (470%/yr).

Current technology can put 2'5 millions of gates on a single chip, and one of our teaching goals is for future computer engineers to manage this number of gates and even more. Using expression (5) and starting from a 2000 gates design in Computer Fundamentals, students could manage 2'5 millions of gates designs after 9 years. This is a very long term educational objective. Nevertheless, if the processor complexity in educational projects could be increased every semester than every year, students would be able to manage real processor designs after 5 years approximately.

4 CONCLUSIONS

Students learn computer architecture rules in theoretical classes and then, some of these rules are put into practice by reconfiguring real prototyping hardware. With our method, we have experienced an increase in the learning phase of students that is higher than the increase in growth of computer performance and complexity in the recent history of computers. The learning curve of computer architecture concepts and related technology aspects may be qualitatively modeled as an exponential function of time. If students begin a computer engineering career with no computer architecture knowledge, they can manage current processor complexity after approximately 5 years supposing a exponential complexity curve $X(t) = 0{,}48\, e^{0{,}85\, t}$ and starting with a 2000 gates design. This is the reason why future processors may be designed by computer engineers that are taking their careers now. I think this approach works because it builds knowledge from the bottom up and students enjoy enormously when designing computers.

ACKNOWLEGMENTS

The author would like to thank the following students who have contirubuted to CEREPRO project: Francisco Peña, Maximo Gonzalez, Pablo Ramírez, Juan J. Gutierrez, Juan Jiménez and Pedro Suárez. Software donation from Altera Corporation is thankfully acknowledged.

REFERENCES

[Alt-96] Altera; Data Book; Altera, 1996.

[Ben-98] D. Benítez, F. J. Peña; Processor Design using Programmable Logic Devices Oriented to Computer Technology Education; Información Tecnológica, Vol.9, No.9, 1998.

[HOYD-96] J.O.Hamblen, H.Owen, S.Yalamanchili, B.Dao, Using rapid prototyping in Computer Architecture Design Laboratories, 2nd Annual Workshop on Computer Architecture Education, Computer Architecture Technical Committee Newsletter, June 1996, pp.44-52.

[Hen-96] J.L.Hennessy, The Computer Architecture Curriculum at Stanford: Challenges and Approaches, 2nd Annual Workshop on Computer Architecture Education, Computer Architecture Technical Committee Newsletter, June 1996, pp.56.

[HP-96] J.L.Hennessy, D.Patersson; Computer Architecture. A Quantitative Approach (second edition); Morgan Kaufmann Publishers, 1996.

[MK97] M. Morris Mano, Charles R. Krime; Logic and Computer Design Fundamentals; Prentice-Hall International Inc., 1997.

[PH-98] D.Patersson, J.L.Hennessy; Computer Organization and Design (second edition); Morgan Kaufmann Publishers, 1998.

[VKSJ-96] A.Varma, L.Kalampoukas, D.Stiliadis, Q.Jacobson, CPU design kit: An instructional prototyping platform for teaching processor design, 2nd Annual Workshop on Computer Architecture Education, Computer Architecture Technical Committee Newsletter, June 1996, pp.23-26.

[Yu-96] A. Yu; “The Future of Microprocessors”; IEEE Micro, Vol.16, No.6, pp.46-53, 1996.