A Curriculum for the Professional Branch in High School

Paulo Leão

Departamento de Informática, Universidade do Minho
4710 Braga, PORTUGAL
leaopaulo@clix.pt

Abstract. Course curricula for the professional branch in High Schools have not been really updated since 1989. A proposal for a renewed Computer System Architecture curriculum is here presented and discussed, based on literature search and on other countries' experiences. It focuses the course core and presents a new approach to the traditional Portuguese high school teaching: the use of low-level language simulators, replacing real CPU instruction sets. The result is a gain in cost, time, flexibility and adequacy of the teaching method to new technologies and opportunities. From a broader perspective, some additional transferable skills are also mentioned, since they help to prepare the young generation to face the external labour environment.

1 Introduction

Since the beginning of the electronic devices that technology never ceases to evolve driven by the needs of society and the new discoveries.

Therefore, school programs - especially those in such a rapid changing environment and status - should accompany such changes. Not only at the technological level, but also in terms of society integration level, assuring that schools respond to the needs of the society, meaning that it responds to the needs of business environment (job availability, flexibility necessary in job functions, specialisation in a certain area but without loosing perspective of the whole picture).

An updated curricular program must be in line with the society of the 21st century and its entire technological environment: 3rd generation mobile phones, interactive TV, digital cameras, automobile control systems, just to name a few "intelligent" devices that are used in modern life.

"A Curricula for the Professional Branch in High School" is a proposal for Portuguese Secondary Professional Schools and will be the subject of this work.

2 Methodology

To elaborate a training program, the first item to define is the output profile of the students. So in the case of the Professional Course of Management Computer Engineering – level III, the students may have as job output the following:

- Participate or be responsible for the choice of computer equipment for the company they work in;
- Develop software (Visual Basic, SQL, C, C++);
- Know how to work with networks;
- Manage and maintain the hardware and software of the company; installing Windows, Office, antivirus software, business applications (such as Infologia) and upgrading the existing software;
- Sell Computer products

The goal is that students, throughout the course, acquire knowledge related to accounting and company organisation and know how to build some software applications if necessary, as well as install software, network environments and install some basic hardware components (like memory upgrades ,etc). And at the same time, they should acquire core knowledge at computer architecture level that will allow them to build up on this knowledge to progress in this area easing up their way in the business world.

Students will have to adapt and update their knowledge throughout their lives, being alert to the changes in market, having therefore flexibility in their abilities.

3 Examples

Examples of school programs of others countries around the globe regarding this area, should help boost the speed of upgrade of these curricular programs.

3.1 Joint Task Force on Computing Curricula Model

In the USA, a Joint Task Force[1] on "Model Curricula for Computing" was created to undertake a major review of curriculum guidelines for undergraduate programs in computing revising the 1991 Computing Curricula[2]. The mission of the task force was expressed as follows:

> *"To review the Joint ACM and IEEE/CS Computing Curricula 1991 and develop a revised and enhanced version that addresses developments in computing technologies in the past decade and will sustain through the next decade."*

3.2 Curriculum for K-12 Computer Science

ACM also produced a Model Curriculum for K-12 Computer Science, which Final Report was produced on October 22, 2003[3]. This report proposes a model curriculum that can be used either in the USA or in any other country around the world.

According to the above mentioned Final report of October 2003, concerning Model Curriculum for K-12 Computer Science, a survey conducted in 2002 by ACM in the USA[4] reached the conclusion that on the international level, the development of K-12 computer science is making more progresses than in the USA, as follows:

- In Israel, a secondary school computer science curriculum was approved by the Ministry of Higher Education and implemented in 1998. It blends conceptual and applied topics, and is offered in the 10th, 11th, and 12th grades. All students in the 10th grade are required to take a 1/2-year course in the foundations of computer science. This is followed by 1-1/2 or 2-1/2 years of electives taught at the 11th and 12th grades. These electives have a particularly heavy emphasis on the foundations of algorithms.
- In Canada, a comprehensive curriculum was recently implemented for all secondary schools in Ontario. It provides two alternative tracks, one emphasising computer

[1] In the USA a Task Force was created to work in the area of Computer Engineering in the end of 1998 joining the Computer Society of the Institute for Electrical and Electronics Engineers (IEEE-CS) and the Association for Computing Machinery (ACM).

[2] A world wide web was created to support the exchange of experiences and ideas : http://www.eng.auburn.edu/ece/CCCE. A draft report was produced on October 3, 2003 and it can be accessed through the above mentioned URL.

[3] http://www.acm.org/education/k12.

[4] Available at http://www.acm.org/education/k12/research.

science and the other emphasising computer engineering. All courses balance foundational knowledge with skills acquisition, and they prescribe outcomes at each level. At the 9th grade level, a full-year "integrated technologies" course is available to all students. This is followed by three parallel 3-year tracks –one in computer and information science and two in computer engineering.

- In many other parts of the world, including Europe, Russia, Asia, South Africa, New Zealand and Australia, computer science is being established in the K-12 curriculum.

3.3 Other Studies for Education in Computer Technology

PECTOPAH: Promoting Education in Computer Technology using an Open-ended Pedagogically Adaptable Hierarchy

This study reinforces the idea that it is essential to understand Computer Systems Architecture to understand Computer Science. Still overall, there is a bias to teach Information and Communications Technology and to neglect Computer Systems Architecture.

There is even an interesting comparison between CSA and ICT teaching and Russian Teaching: Teaching ICT without teaching CSA is like teaching Russian without teaching the Cyrillic alphabet.

Therefore, the proposal that is presented in this work follows these ideas.

Also due to a misconception of the effect of technological change, there is a tendency to use inappropriate didactic tools.

Three tools were developed in order to provide together a progressive hierarchy of teaching aids that can be used at different levels of teaching, providing students with an incremental toolbox.

- Primary and Secondary Education –"How Computers Really Work" This tool consists of a pilot interactive CD-ROM for teaching CAS in primary and secondary schools. The largest area of the CD is CA area dealing with the CPU, memory and data.
- Introductory Undergraduate Level – "The Postroom Computer". The Postroom Computer (PC) is an extended emulation of the LMC model[5], in which the focus is made on flexibility and generality. It is designed to introduce aspects of CSA and low level programming in an incremental way.
- Advanced Undergraduate/Postgraduate – "Update Plans". Update Plans is a formalism for the description of abstract machines and algorithms.

4 Proposal

4.1 Proposal of the Joint Task Force on Computing Curricula Model

The proposal made by the Joint Task Force on Computing Curricula on October 3, 2003 for the scope of this work was the following:

[5] LMC: Little Man Computer model was created by Stuart Madnick and John Donovan in 1965 to teach low level programming and computer architecture to undergraduate level.

CE-CAO. Computer Architecture and Organisation (70 hours)
CE-CAO0. History and overview of computer architecture (1)
CE-CAO1. Fundamentals of computer architecture (12)
CE-CAO2. Computer arithmetic (3)
CE-CAO3. Memory system organisation and architecture (9)
CE-CAO4. Interfacing and communication (12)
CE-CAO5. Device subsystems (6)
CE-CAO6. Processor systems design (12)
CE-CAO7. Organisation of the CPU (12)
CE-CAO8. Performance (3)
CE-CAO9. Performance enhancements (elective)
CE-CAO10 Crosscutting Issues (elective)

CE-DIG. Digital Logic (50 hours)
CE-DIG0. History and overview of digital logic (1)
CE-DIG1. Switching theory (6)
CE-DIG2. Combinational logic circuits (4)
CE-DIG3. Modular design of combinational circuits (6)
CE-DIG4. Memory elements (3)
CE-DIG5. Sequential logic circuits (12)
CE-DIG6. Register Transfer Logic (6)
CE-DIG7. Digital Systems Design (12)

4.2 A Proposal for Computer System Organisation in the "Informática de Gestão" course

The course curricula for the professional branch in High Schools have not been updated since its beginning in 1989, except for the introduction of minor changes within the existing guidelines.

There is a major gap in the actual curricula as it gives more emphasis to the electronic and logic areas, that no longer respond to the needs of professional branch high schools' students in "Informática de Gestão". For these students the most important issue is to understand how the high level languages are processed and transformed, rather than the particular aspects of what is, for example, inside an integrated circuit.

It is important to remind that the basic knowledge on computer hardware (processor, memory and data) and software, peripherals (namely, screens, modems, graphics cards), history of computing and network systems were previously covered in the Computing Applications course.

The proposal here described for this subject contains 129 hours[6] and has the modules presented below. The duration of these modules was based in the experience that the school has in teaching these subjects, as well as the indications of the international schools' studies referred above.

CE-CAO Computer Architecture and Organisation (129 hours)

CA-01 Computer Architecture (10)
CA-01a History and overview of computer architecture (1)
CA-01b Fundamentals of computer architecture (9)

[6] According to the Portuguese Law nº 1112 of September 12,1995

Learning Objectives:

1. To know the main marks of the computer history.
2. To know the architecture of a typical computer architecture and its main components.

CA-02 Number Systems (21)
CA-02a Binary, Decimal and hexadecimal system (2)
CA-02b Conversions between the various systems (5)
CA-02c Sum, subtraction and multiplication of two or more binary numbers (5)
CA-02d Sum, subtraction and multiplication of two or more numbers in the various bases (5)
CA-02e Signed and twos-complement representations (4)

Learning Objectives:

1. To explain to the student the reasons to have different formats to represent numerical data.
2. To convert numerical data from one format to another, namely in binary and hexadecimal systems
3. To solve arithmetic problems having numbers in different bases, other than decimal.
4. To represent relative numbers in complements codes
5. To perform arithmetic problems (sum and subtraction) with relative numbers.

CA-03 Memory system I/O devices (23)
CA-03a Memory system organisation and architecture (8)
CA-03b Interfacing and communication (10)
CA-03c Device subsystems (5)

Learning Objectives:

1. To understand how the memory is organised.
2. To understand the different levels of memory (RAM, Cache, disk, etc).
3. To understand what are Input/Output devices.

CA-04 Organisation and Internal Structure of a CPU (32)
CA-04a Von Neumann Computer model (2)
CA-04b Processor systems design (5)
CA-04c Organisation of the CPU (5)
CA-04d CPU function: instruction fetch, decode and execution (15)
CA-04e Evolution of computers centred in the evolution of the CPU architecture: from the 8085 onwards (5)

Learning Objectives:

1. To explain the organisation of the Von Neumann machine and its main functional units.
2. To explain how an instruction is executed in a Von Neumann machine.
3. To understand the CPU function in the machine.
4. To understand how the components of a computer interconnect and interact with each other.
5. History of the evolution of the CPU architecture starting with the 8085 until the latest version available in the market.

6. To understand the CPU organisation and basic instructions
7. To be able to differentiate the processors based on the records and instructions.

CA-05 Program Execution in a Computer (24)
CA-05a Levels of abstraction: procedures, functions and iterations as abstraction mechanisms (8)
CA-05b Conversion of an abstraction level to the CPU commands: modules in programming language (8)
CA-05c Usage of a computer systems simulator: discrete-event simulation, continuous simulation, verification and validation of simulation models (8)

Learning Objectives:
1. To understand what are levels of abstraction.
2. To understand the abstraction mechanisms (procedures, functions and iterations)
3. To be able to do small programs that will have action on registers and flags.
4. To be able to create cycles (jump; jz; jnz ; call; ret; leave; etc)
5. To understand how to use a computer systems simulator

CA-06 Performance (12)
CA-06a Compilation (4)
CA-06a Disassemble (4)
CA-06a Optimisation (4)

Learning Objectives:
1. To understand the process of compilation
2. To know the options of compilation
3. To understand how to optimise
4. To understand how the language is transformed. The student has to understand the above mentioned once he/she works with HLL.

CA-07 Future Vision of Computing (8)
CA-07a High performance computing (8)

Learning Objectives:
1. To know the potentiality of parallel systems. To have a simplistic vision of it.
2. Future trends

5 Conclusions

Despite differences in emphasis and content, there are certain common elements that should be expected of any computer engineering program: the core knowledge. This knowledge should be expected to be present in all programs. Still, it should be as short as possible, allowing more time for the specialisation in a chosen area, while maintaining a "broader picture".

The actual curricula has five modules that focus mainly in the electronic and logic areas, namely Boolean algebra, flip-flops, combinational circuits, sequential logic circuits and the study of 16-bits microprocessors, that do not respond to the needs of the "Informática de Gestão" course.

So, though important in other areas of teaching, the specific and deep knowledge of the above mentioned electronic and logic areas should be taken out of this course curriculum as proposed above.

It is very important that the Computer Systems Organisation is present in this curriculum, so students can understand how a computer works, even though they do not come from an electronic background. Having the knowledge of how the computer works, they will be better prepared to face and solve future problems when working with high level languages and applications. Students will be more open-minded in computer systems organisation and will be more able to progress in the area they are working.

One of the constrains of the majority of schools is the lack of financial resources to offer students the latest technology to perform "hands-on" work. Therefore, this issue has also to be taken into account when preparing the course.

Simulators are very interesting teaching aid tools and can be a response for these difficulties. These simulators are freely available on the internet and may help overcome the problem of limited resources[7].

Also providing regular access to well-equipped computer laboratories and networks is mandatory to achieve a successful result, since laboratory work is an important area of the program.

From a higher-level perspective, however, there are also transferable skills that one can reasonably expect from all courses, especially those in the professional branch, as they aim to prepare students for the labour environment, as follows:

- *Communication Skills* – students should be able to communicate their work in appropriate formats (written, oral, graphical) and to critically evaluate materials presented by others in those formats.
- *Teamwork Skills* – students should be able to work in a team, building up each other potential to achieve a final and unique goal.

References

[1] Osborne, H., Crossley, S, Mencak, J., Yurcik, W., PECTOPAH: Promoting Education in Computer Technology using an Open-Ended Pedagogically Adaptable Hierarchy, published in the Proceedings of the Workshop on Computer Architecture Education, 2002

[2] Tucker, A., Deck, F., Jones, J., McCowan D., Stephenson, C., Verno, A., A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Education Task Force Curriculum Committee, October 2003

[3] Wolfe, G.S., Yurcik, W., Osborne, H., Holliday, M. A.: Teaching Computer Organisation/Architecture With Limited Resources using Simulators, Proceedings of the 33rd Technical Symposium of Computer Science Education, ACM Press, Feb/March 2002

[4] The Joint Task Force on Computing Curricula, IEEE Computer Society, Computing Curricula 2001, Preliminary Draft, October 2003

[5] The Computer Architecture and Assembly Language Education Homepage, www.sosresearch.org/caale

[6] Computing Curricula 2001 Computer Science: Final Report, December 15,2001

[7] Please refer to the work "Teaching Computer Organisation/Architecture with Limited Resources Using Simulators presented by the National Science Foundation under grant DUE-9850534