

# Hardware Procedures vs. Simulation Procedures: What Do Computer Science Engineering Students Think About Them?

E.J. GONZÁLEZ, A. HAMILTON, L. MORENO, R.L. MARICHAL, V. MUÑOZ  
Dept. de Física Fundamental y Experimental, Electrónica y Sistemas  
Universidad de La Laguna  
38207 S/C de Tenerife , Spain

**Abstract:** - A well-planned design of procedures is highly important in Engineering Courses. Several researchers have found many reasons for choosing a combination of simulation and hardware procedures for this design. However, what do Computer Science Engineering students think about these procedures? The aim of this paper is analysing some advantages and disadvantages that Engineering students find when they are asked about hardware or simulation procedures.

**Key-Words:** - System Architecture, Simulation Tools, Computer Engineering, Hardware and Software Procedures

## 1 Introduction

Several researchers [1] have found many reasons for selecting a combination of simulation/software and hardware procedures when they design an Computer Science Engineering Course. Simulation/software experiences, with tools like Octave [2], SPIM [3] or WinDLX [4], are often used because they are easier and cheaper, and teachers do not need to supervise the work of the students as closely as in hardware instruction. In contrast, hardware experimental procedures take a long time, and may be tedious and too difficult when they are not appropriately designed. Nevertheless, it is well known that there are many areas that cannot be shown through simulation. These practical aspects can only be learned through carefully chosen hardware experimentation.

On the other hand, there are some procedures that join software to hardware. For example, programming a device such as a programmable peripheral interface (PPI), an interrupt controller (PIC) or a microprocessor. These procedures are more complete but sometimes they are not possible to implement in some subjects.

The aim of this paper is analysing some advantages and disadvantages that Engineering students find when they are asked about hardware or simulation/software procedures. For this purpose, fifty Computer Science Engineering students in University of La Laguna were asked to answer some questions about hardware, software or hardware-software procedures. The authors think that it is important to know what students think about these procedures. With this information, some mistakes could be avoided when procedures courses are designed. For example, procedures that students do not find interesting at all or that take too much time. Moreover, the authors have observed that students are

usually prejudiced against some procedures, especially about hardware implementations, making some wrong assumptions. These assumptions could be removed in an easier way with these answers.

The questions proposed to the students were the following:

- 1) What kind of procedures do you *prefer*? Why?
- 2) What kind of procedures do you find *more useful*? Why?
- 3) Advantages and disadvantages of hardware procedures
- 4) Advantages and disadvantages of software procedures
- 5) Advantages and disadvantages of hardware-software procedures

In the remainder of this paper, the authors will analyse the answers to these questions in sections 2 to 6, while conclusions are shown in section 7.

## 2 What kind of procedures do Computer Science Engineering students prefer?

In respect to this question, the answers that the authors have obtained are the following:

Hardware.....	6 %
Software.....	12 %
Hardware-Software.....	78 %
None.....	4 %

As can be seen, hardware-software procedures are clearly the most favourite ones. On the other hand, only the 6 per cent of the students answer that hardware procedures are their favourite ones. The

authors have observed that students show a lot of prejudices against hardware procedures.

When students are asked about the reasons, the great majority of them answer that the hardware-software procedures are the most complete ones, with the advantages of both hardware and software procedures (these advantages are detailed in next sections). The students give other answers such as these procedures are more useful (in the next section, it is shown that the most of the students believe that) and more powerful for the “real life”.

In contrast, the 12 per cent think that software procedures are more interesting because students can be more creative. Moreover, they answer that these procedures are less problematic than the hardware ones. Once more, prejudices against these last procedures are shown. A significative fact is that none of the students was able to give any reason in favour to the hardware procedures.

However, these conclusions are complemented in the next section.

### 3. What kind of procedures do Computer Science Engineering students find more useful?

In respect to this question, the answers that the authors have obtained are the following:

Hardware.....	24 %
Software.....	8 %
Hardware-Software.....	62 %
None.....	6 %

Once more, hardware-software procedures are clearly the procedures that the students give more value, although only the 62 per cent of the students find these procedures more useful, in contrast to the 78 per cent in respect to the first question. The reasons are very similar. They are considered more complete, more useful and more powerful. They are, in their opinion, the kind of problem they will find in the “real life” and allow them to acquire experience from important instrumentation devices as logic analysers or oscilloscopes. Moreover, they gain experience on debugging strategies in both fields: hardware and software.

It is important to remark that although students are not highly interested in hardware procedures, they consider them very useful (the 24 per cent) for their future work as Computer Science Engineer. They think that these procedures are simpler and the students do not need to know any computer language. Once more, the students consider that working with hardware allow them to acquire experience from

instrumentation devices. In contrast, only the 8 per cent of the students think that the software procedures are the most useful.

## 4. Advantages and Disadvantages of Hardware Procedures

In next sections the authors will analyse the advantages and disadvantages that the students find with respect to these procedures. With regard to the hardware procedures, although the students find many advantages, they point a lot of disadvantages. Most of these thoughts show the prejudices against hardware.

### 4.1 Advantages

Although the students have several prejudices against them, they find the following advantages:

- *Acquiring experience from instrumentation devices*
- *Acquiring experience from design and debugging techniques*
- *More useful for “real life”*
- *Students can learn more concepts*
- *These procedures usually require less time*
- *Easier to implement*

The authors have observed that students are highly interested in devices, such as logic analysers, which they are not used to. It is an important point for those professors that wish to improve their hardware procedures. The students think that using these devices is as important as the implementation itself. So, the authors recommend including more and different instrumentation devices in these procedures. Moreover, many of them are reasonably priced [5]. This way, students pay more attention to the procedures that they implement, controlling more aspects of them. For example, in the case of a logic analyser, they can access simultaneously to several signals and time relations. With devices like them, they acquire experience from debugging techniques. The students appreciate them because in the “real life” this experience will help them to solve a lot of problems.

The hardware procedures usually tend to be easier to implement and require less time because the time that students spend in laboratory is limited. There are many reasons for this limitation. There are many students or there are not enough devices. This limitation is an apparent advantage for the students, as can be seen in the answers to this question, because they do not have to spend a lot of time implementing procedures. However, the authors

consider it a great disadvantage, because making the procedures too easy produce three negative effects:

- Students could think that hardware problems in “real life” are very easy to resolve.
- Students do not assimilate the concepts properly.
- Students are less interested in these procedures.

#### 4.2 Disadvantages

In this section, the students show clearly many of the prejudices against this kind of procedures. The list of disadvantages that they find is the following:

- *They may be tedious*
- *Very difficult to debug*
- *Possible damaged devices*
- *Students spend too time in things that do not teach anything*
- *Students cannot be creative*
- *Too sensible to non-evident effects*
- *All these procedures are very similar*
- *Difficult to update*
- *Limited Time*
- *It is possible implement them without understanding them.*

Many of these disadvantages are originated by the difficult work of debugging a hardware system. Software systems become sometimes complex, but they allow the students to delimit the code zones that contain bugs in a easier way. On the other hand, this delimitation is highly difficult to make in hardware procedures. Debugging all the components may be tedious and some of them could be damaged or bad soldered, for example. Moreover, these procedures usually imply limited time for their implementation. All these factors increase the number of these prejudices. However, the authors consider these situations highly pedagogical because the students are forced to manage problems that they will find in their future work. They recommend remarking these points to the students when procedures are explained.

Other group of disadvantages lies in the update of the components. Nowadays computers are faster and faster, but including this fact in these procedures is highly difficult. For example, the components such as a resistance or a diode are hardly to update. With regard to this point, the authors propose two recommendations:

- Removing all the possible damaged components. Students usually get angry about the procedures when they find a damaged component. They do not understand that these situations are not unusual. The authors recommend one step more: removing all the components whose aspect is doubtful.

- If updating the components is not possible, professors should update debugging devices, including for example, logic analysers or digital oscilloscopes.

Finally, students complain about the lack of creativeness in these procedures. The authors suggest giving the students more time to design their implementation properly.

#### 4.3 Suggested Improvements

Finally, students were asked to suggest some improvements. The results are the following:

- *Studying a system that has been already implemented*
- *Designs more detailed*
- *More time for the design and the implementation*
- *More documentation*
- *Demo procedures*
- *Integrating these procedures in a more complex project*

Once more, students emphasise the time as factor to improve. They wish more time for design and implementation. Moreover, they wish to substitute these procedures for the study of systems yet implemented. The authors consider that creating a balance between both techniques is appropriate if it is possible. This way, students can study more concepts in a practical way.

Other suggested improvement lies in procuring more details about the designs to implement, apart from the data sheets of the components. This is an apparent contradiction to the option of increasing the creativeness. The authors think that Computer Science Engineering students should tend to base their implementations only in these data sheets, because they will find themselves in these situations in their future work.

## 5. Advantages and Disadvantages of Software Procedures

Once the hardware procedures have been analysed, in this section we will analyse the point of view of the students about the software procedures.

### 5.1 Advantages

The advantages that the students find in respect to software procedures are the following:

- *Acquiring experience with programming languages*
- *If they go well once, they always go well*

- In “real life”, they will find more software bugs than hardware ones.
- They can be implemented at home, without limited time
- Students can be more creative

In general, students show more interest in respect to these procedures,. The first reason is that they find the software more “visual” and allows them to be more creative. However, the authors have the opinion of that these aspects are not the most interesting ones. They allow students to learn about algorithms, optimisation techniques and software resources such as the use of interruptions. It is recommended remarking these aspects when a new procedure is designed.

Once more prejudices against hardware appear in these answers. Two of the five reasons are given in contrast to hardware procedures: unlimited time for implementation, usually at home and that if they go well once, they always go well. In general, the authors recommend dedicating some time to try to relieve these beliefs.

### 5.2 Disadvantages

In contrast, this is the list of disadvantages that the students have found.

- *Students have to dominate too programming languages.*
- *It is possible implementing them without understanding them.*
- *Software should be free software*

The main disadvantage the students have found is that they are usually forced to study a new programming language for an only procedure. In this sense, the authors suggest limiting the software procedures to two programming languages, as far as possible. Concretely, one high-level and one low-level language. The authors recommend especially C and Assembler. This way, students can dominate these languages and spend more time implementing their procedures.

Using free software in these procedures is considered an interesting suggestion also. This way, students can implement their procedures in an easier way at their home, instead of a laboratory. Nowadays there are free packages about a wide number of fields such as circuit designs or Computer Architecture.

### 5.3 Suggested Improvements

Apart from the suggestion about free software, these are the improvements that the students suggest.

- *Several small procedures instead of a complex one*
- *Analysing more the instructions that are going to be used*

The authors consider that these improvements are very interesting. Several small procedures instead of a complex one allow professors to cover more topics and students to understand better these procedures. On the other hand, an analysis of the instructions allows the students to focus on the theoretical aspects of the procedures.

## 6. Advantages and Disadvantages of Hardware-Software Procedures

Finally, the authors analyse briefly the answers about hardware-software procedures.

### 6.1 Advantages

In spite of these procedures are the favourite of the students they only find one advantage:

- *They are more complete, with the advantages of both hardware and software procedures*

This explanation implies that software allows hardware to be more creative and more “visual”.

### 6.2 Disadvantages

In respect to these procedures, the main disadvantage that the students find is the following:

- *Sometimes they cannot be debugged properly*

In respect to this disadvantage, the authors think that including modern debugging devices is a good alternative, as it is pointed above.

## 7. Conclusions and Future Work

The aim of this paper is analysing some advantages and disadvantages that Computer Science Engineering students find when they are asked about hardware or software/simulation procedures. For this purpose, fifty Engineering students in University of La Laguna were asked to answer some questions about hardware, software or hardware-software procedures. The students consider these last procedures as their favourites (78 %) and the most useful (62 %). It is important to remark that although students are not interested in hardware procedures, they consider them very useful (the 24 per cent) for their future work as Computer Science Engineer.

The authors have observed that the students usually show many prejudices against hardware. They consider these procedures quite tedious and difficult to debug. Starting from the answers of the students, some suggestions are made for improving the procedures, such as including modern debugging devices (logic analysers, for example).

In respect to future work, the authors wish to repeat this study in other courses such as Computer Science ones, comparing the obtained data.

### **Acknowledgements**

The authors of this paper wish to thank the remain of the professors of the Grupo de Computadoras y Control of the University of La Laguna, especially Dr. Graciliano Nicolás Marichal Plasencia and Dr Jose Luis Sánchez de La Rosa, for their collaboration in this work

### *References*

- [1] Moreno L., González E.J., Piñeiro J.D., Estévez J.I., Merino J.J., Sigut J.F., Aguilar R.M. ‘Experiments on VME and ISA buses for a computer architecture laboratory’. International Journal of Electrical Engineering Education. 38/2. pp. 95-102. April 2001.
- [2] Octave has been developed by John W. Eaton
- [3] SPIM has been developed by J.R.Larus (Computer Science Department. University of Winsconsin-Madison).
- [4] WinDLX has been developed by H. Grünbacher (Vienna University of Technology).
- [5] HP 1600-A Series 50/100-Mhz State/500 MHz Timing Logic Analyzers User’s Reference (Hewlett Packard, Colorado Springs, 1994)