PipeSim: Pipeline-Scheduling Simulator

José M. Chaves-González, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, Juan M. Sánchez-Pérez

Department of Computer Science, University of Extremadura
Escuela Politécnica, Campus Universitario s/n, 10071, Cáceres, SPAIN
{jm, mavega, jangomez, sanperez}@unex.es

Abstract. Nowadays pipelining, pipelines and pipeline-scheduling are very important topics in the current computer architecture. All present processors use pipelining to improve their performance; in fact, the usage of the computer resources has important influences on its final performance. So, speaking in general, pipeline and pipeline-scheduling topics are very outstanding in the computer science studies and it is quite important that students can learn these in an easy and reliable way. This paper presents a project that provides both, professors and students, a tool that is useful for studying, teaching and learning how pipelines work and how they can be scheduled in an easy and widespread way. The project is called PipeSim, and features both static and dynamic pipelines.

Keywords: Pipeline, pipeline-scheduling, reservation table, collision vector, state diagram, forbidden latency list, delay insertion.

1 Introduction

What encourages us to build the simulator PipeSim were the aspects which the subject in the 4th year of computer science studies, computer architecture and engineering, offers. This subject, as occurs in many others, has a very large syllabus, so there is not enough time to look at in depth some parts of the subject. Among these parts of the syllabus are the topics about pipeline scheduling. To learn more about these topics it is necessary to solve some problems that have though a well-known methodology for solving, but the problems are neither small nor easy to work out in most of the situations.

These problems are usually handled on the blackboard by hand, and this causes that the amount and the versatility of the exercises is limited. This might be the reason why students are not confronted with many different examples and different kinds of problems in order to understand all the situations that could happen in the scheduling of a pipeline. So, the usage of a simulator that provides easy problem solving connected with a user-friendly interface for scheduling would be a progression for students. In fact, it is proved that if students are able to use tools that help them to learn and study in depth, they learn more, because the teacher not only has one more tool for teaching the most difficult parts of the syllabus, but also the students can go

on practicing on their own home and in their own speed. Actually, simulators are the best way to understand some of the most difficult concepts related to the computers design because these have many elements interacting.

Specifically, PipeSim is a full interactive application that helps the user to master the knowledge about pipeline scheduling by means of a very simple interface and a very descriptive and didactic way. The simulator has a very comprehensive help system. Using this, the students can even study the theoretical concepts of pipelining and pipeline scheduling [7], [2]. Besides, the application is completely flexible in the way of presenting the results. It is possible, for example, to see any of the stages of scheduling, whichever it is. You can either walk through the complete simulation step by step, have a look at all the details and watch concrete steps of the scheduling or just study the final results of the simulation without presenting intermediate results.

With the possibilities that the simulator provides, the students can check hand-solved problems after each completed step to obtain the final result. The ability of comparing the final scheduling results underlines the importance of simulator-supported learning in modern teaching techniques.

Besides, sincerely, we do not know any other similar teaching resource, so it is impossible for us to speak about the differences, advantages or disadvantages between different pipeline-scheduling simulators. In fact, due to the non existence of teaching resources in this specific field we felt encouraged to develop PipeSim.

2 The simulator PipeSim

Pipeline-Scheduling simulator (PipeSim) has been built with didactic purpose. So, we had to take care of both the simulator execution and the presentation of the results so that they are well interpretable by the user. With this aim, PipeSim has been carefully developed following the classical software engineering method [3]. For the development we used the Borland C++ Builder 6 [1]. Referring to hardware and software requirements, PipeSim has been designed so that it is able to run in a wide range of systems, current systems or older systems anyway. That is because students and different users could have any PC in their homes, and although PipeSim has been optimized to perform under Windows XP, it has been proved in a wide range of machines (the older PC successfully checked was a 300-MHz Pentium II, with 96 Mbytes of RAM and Windows 98 installed as operating system). Moreover, the PipeSim installation to the HD takes less than 5 Mbytes. So, we make things easy so that most of students (or any other people) can use PipeSim with no problem.

So, the simulator was developed thinking about its final usage, and this is the reason that encouraged us to build a user-friendly interface for the simulator. PipeSim is an application based on windows, very easy to use and with a well cared final appearance (buttons with descriptive icons, a lot of very clear explanations about the steps to do, complete help system, etc.) that helps users to learn and study pipeline-scheduling. Besides, the simulator controls all possibilities of error that could occur during the performance, and it presents the user many warning and error messages so that the user is informed about any irregular behaviour of the simulator.

As shown in figure 1, the simulator has a very simple main window. The upper part of the window is occupied by a main menu, and in the rest of the window there are nine buttons, grouped into three rows, according to their category. In the upper row are the system options to *insert a new reservation table (RT) -static or dynamic-*. In the central row there are options to *load a RT* in the simulator. This operation can be made by using a file or by synthesis operations. And in the lower row are the more useful operations of the application located (Show the RT loaded in the system; show the hardware diagram of the pipeline obtained from the loaded RT; and show the state diagram obtained in the pipeline-scheduling, which is the most important part of the scheduling, because it represents the occupation of the pipeline for each clock cycle).

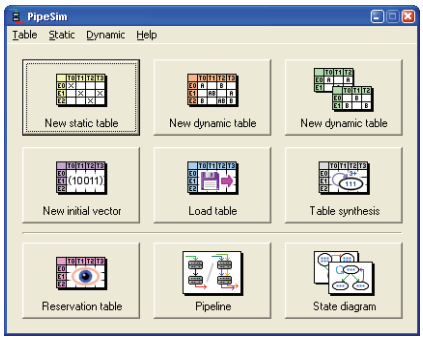

Fig. 1. PipeSim main window.

The buttons placed in the main window have not the ability to control the entire functions that can be performed by the simulator. To access all the functions in the simulator you have to use the main menu, which has four options, which are explained in the following lines:

Table menu: The load and save operations for static and dynamic reservation tables are managed using this menu, besides there are options to show the loaded table in PipeSim and to quit the simulator.

Static menu: Through this menu is possible to handle operations that can be made over a static reservation table, that is, static pipeline-scheduling. These operations are: To draw the static pipeline (hardware elements) which represents the reservation table (RT) loaded in the simulator, to get the forbidden latency list from this RT, to calculate the initial collision vector using the forbidden list, to generate and draw the state diagram obtained from the initial vector, to make a study of simple cycles (and their properties) found in the state diagram, to insert delays in a RT that have not an optimal throughput and to synthesize a new RT taking an arbitrary cycle. If we omit this last operation, the others are the sequence of operations that are necessary to solve a complete problem about static pipeline-scheduling. Nevertheless, it is not necessary to go through each option, because the simulator has the capacity of going to a concrete option, running automatically the previous stages in the scheduling.

Dynamic menu: The user can manage functions that can be made over a dynamic reservation table using this menu. Here are the options to do the entire dynamic pipeline-scheduling, step by step or jumping to a concrete stage of the scheduling. The dynamic scheduling is more complicated than the static, because the pipeline

performs several operations at once. PipeSim works with dynamic pipelines of 2 operations at once.

Help menu: Help options, language configuration and about option are accessible from this menu. PipeSim help is divided in a complete *user manual* where is explained the simulator usage and in a *theoretical explanation* of the concepts which support the simulator behaviour. Besides, it is possible to change the language of the simulator in any moment of the simulation from English to Spanish and vice versa. The complete simulator works in both languages, including help files, installation system, warning and error messages, etc.

3 Principal contributions of PipeSim

We explained in the introduction that the main aim of PipeSim is to help students to understand and to study the concepts and the method of pipeline scheduling. The simulator gives them the opportunity to improve their theoretical knowledge and, over all, to get the practical dexterity in the part of Computer Architecture referred to pipeline scheduling. But PipeSim can be also an application for the analysis and the detailed study of big or very complex pipeline scheduling, which is an important issue in pipelining and parallelism fields. PipeSim has the following characteristics:

- It allows the study of both static and dynamic pipelines, so this makes the simulator more complex and useful.
- The user can insert data in the simulator in different ways. Users can fill in the static or dynamic reservation tables by hand, they can load files with the information of the tables, they can insert a collision vector to start the simulation or even they can synthesize a new table from a cycle of the state diagram. The state diagram represents the pipeline occupation, so the user can choose an optimal cycle to create the new table. Therefore, the simulator gives the user the possibility to start the simulation from different input data, and this provides a widespread usage of the simulator to anybody who uses it, and if we look at specifically in students, this means that they can do a bigger amount and more sorts of exercises, which is very good for their learning.
- It is possible to save reservation tables in specific files. This information can be loaded automatically in the simulator later. Moreover, PipeSim is able to save some results, for example, BMP images of the state diagrams or pipeline draws (these pictures can be used later for different purposes: printing, insertion in documents, professor exams…).
- The simulator offers a complete multi-language configuration, in English and in Spanish, offering the possibility to change the language in any moment of its execution, from the installation to any other time.
- The application can calculate the forbidden list of a static pipeline or the forbidden lists of a dynamic pipeline.
- It can calculate the collision vector (or dynamic collision matrices) generated from the forbidden list (or lists -in the dynamic case-).
- PipeSim can represent the hardware elements that symbolize the pipeline loaded in the simulator. The picture presents the pipeline stages and the data path followed in

each clock cycle. The simulator can draw both static (fig. 2) and dynamic (fig. 3) pipelines of any reservation table up to 100 stages. It can represent different high-level hardware components such as: logic of the stages, latches, multiplexers and demultiplexers.

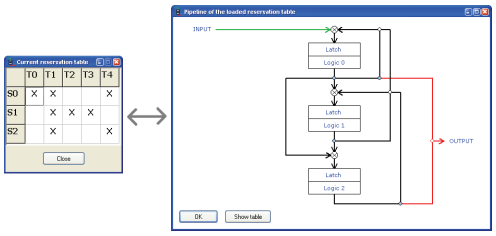

Fig. 2. Static reservation table and associated pipeline drawn by PipeSim.

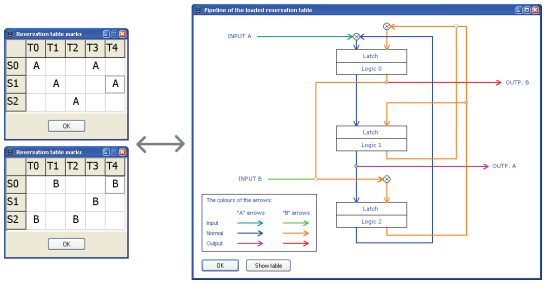

Fig. 3. Dynamic reservation tables and associated pipeline drawn by PipeSim.

- The simulator obtains and represents the static or dynamic state diagram, from the collision vector or collision matrices of a loaded reservation table. This diagram represents the different states of a pipeline for a requested period of time. As it is shown in figures 4 and 5, the diagrams are build using different colours (as it also occurs in the pipeline pictures shown above) and a clear state distribution, to make it easy for reading. The algorithm we have developed for the drawing of *any* state diagram created from arbitrary data inserted by the user is one of the most complex parts of PipeSim because it can draw diagrams containing up to 128 states.
- Therefore, the state diagram can become really complex. It is then very useful to see the diagram as a table (the state diagram is a graph that can be represented by its matrix). This possibility is also offered by PipeSim.

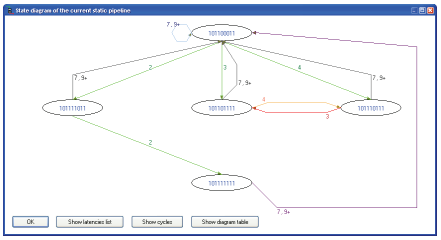

Fig. 4. Static state diagram.

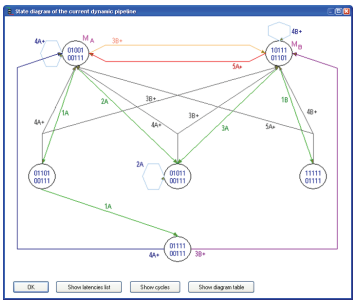

Fig. 5. Dynamic state diagram.

- The simulator allows an intensive study of the simple cycles in the diagram state (for static and dynamic pipelines). This allows knowing when a pipeline is optimal for the operations loaded in it. Besides, the simulator allows changing the data distribution in the pipeline so that the throughput becomes optimal if it is necessary. Therefore, PipeSim offers a complete performance study of the pipeline, and gives the user the entire detailed information about it (as shown in figure 6).

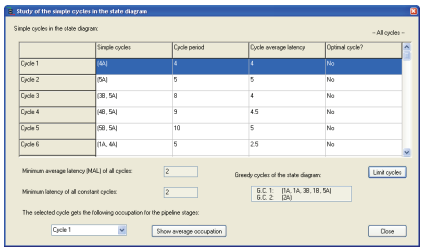

Fig. 6. Simple cycle study in a dynamic pipeline.

- PipeSim does a study of the pipeline throughput improvement by means of delay insertion (and it is even possible to keep the data dependencies in the pipeline stages). Figure 7 shows some screens of the process and the final result for a small example. We have achieved that PipeSim increases the pipeline throughput in an efficient way. The simulator inserts the minimum amount of delays to get the optimal performance.
- The simulator can synthesize new reservation tables from different input data, showing the details of the new table creation.

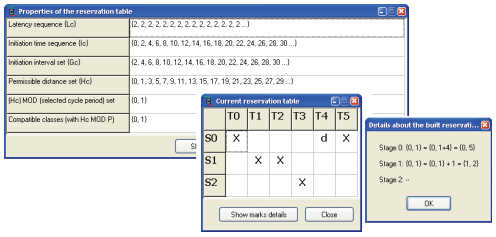

Fig. 7. Example of delay insertion in a static reservation table.

- The application has a complete help system, where not only the simulator usage is explained, but also the theoretical concepts of pipelining and scheduling.
- To make the simulator distribution and installation easier, PipeSim has an automatic installation assistant.
- Furthermore, we have to say that PipeSim can work with little (learning exercises), but also with huge problems (pipelines up to 100 stages, state diagrams up to 128 states). So, the simulator has the flexibility for operating with all kind of users, beginners (who want a lot of explanations and details) and experts (who want to get a specific result or see a concrete part of a more complex simulation).
- Finally, we have to emphasize that we had a lot of care not only in the development of the PipeSim behaviour, but also in its *interface*, because we took into account that the simulator has a learning aim, so a clear and nice appearance make easy the student learning, which is the main purpose of PipeSim.

4 Educational application

PipeSim has begun to be used this year successfully in the subject "Computer Architecture and Engineering" among the students of 4th year in Computer Science. In fact, this is not the first simulator used in this subject ([4] and [5]). Because of the success obtained with other simulators we were encouraged to develop and to create PipeSim.

The simulator is being used in the following way: The professor presents it in class, explaining how can PipeSim be obtained (download from the web site [6]), how does it work, what is it used for, and solving some practical problems with it. Using PipeSim, the professor can solve more problems, and these take him less time than if he does the problems by hand (blackboard). So, the concepts and methodology of pipeline scheduling is clearer and deeper in the lessons than without the simulator, because it is studied more in less time. After the teacher explanation, the student can download and use PipeSim at home. He can do all the exercises he needs, and he can study using the simulator and its complete help system. PipeSim has a set of typical reservation tables saved in files, but the professor suggests some interesting exercises that the students can solve by hand and check using the simulator.

To check the success of PipeSim, a survey has been made among some students that used the simulator. We asked about how good are the different parts in the simulator for them, if it is useful, the management of the simulator is easy, the theoretical concepts are well explained, it is well-organized, the interface is clear, the final conclusion they have… The anonymous survey was done among 83 students and the results obtained were very satisfactory for us. In the following picture (fig. 8), some of the questions done and the answers obtained are shown.

In the graphic is shown a sample for the results obtained in 6 of the questions done in the survey to the students. All questions about the simulator have 5 level of satisfaction, from *bad* (the lowest) to *great* (the highest). As it can be observed, the success of PipeSim among the students has been quite high, so we can conclude saying that the use of PipeSim together with professor explanations in class is a successful method for teaching pipeline scheduling.

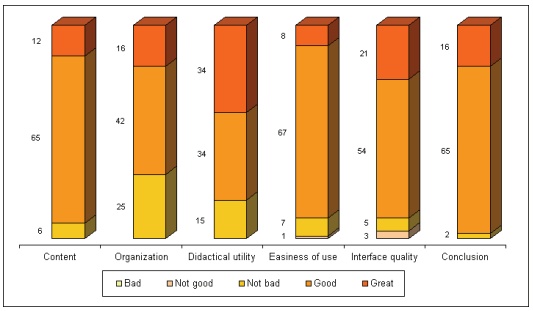

Fig. 8. Some results of the survey done to the students.

5 Conclusions

A very complete and versatile simulator about pipeline scheduling has been developed. In fact, not all functionalities of the simulator are explained in this paper due to space reasons. Although PipeSim could be used in other fields (research), it was developed for students, to teach them pipeline scheduling. As students are inexperienced in the beginning, we took care of explaining the methodology step by step, showing all the details in the clearest way, trying to give all available help. So, although PipeSim is used for the professor to solve problems in class, it is a tool thought mainly for students. Each student has his/her own speed in learning pipeline scheduling. Moreover, it is possible to go further in the topic and simulate real problems, because PipeSim has enough power to do it. Therefore, the simulator is useful to learn how the pipeline scheduling is done, learning and understanding it gradually. But it is useful too for studying more complex cases about scheduling in real pipelines and maybe for doing some research over it.

Finally, we have to say that the simulator can be downloaded, free of charge, for teaching and research purposes from [6].

References

1. Charte, Francisco. Programación con C++ Builder 5. Anaya Multimedia, 2000.
2. Hwang, Kay. Advanced Computer Architecture: Parallelism, Scalability, Programmability. McGraw-Hill, 1993.
3. Pressman, Roger S. Software Engineering: A Practitioner's Approach. McGraw-Hill, 2005.
4. Vega, Miguel A., Sánchez, Juan M., Gómez, Juan A. An Educational Tool for Testing Caches on Symmetric Multiprocessors. Microprocessors and Microsystems, vol. 25, no. 4, pp. 187-194, June 2001.
5. Vega, Miguel A., Burgueño, J. Carlos, Sánchez, Juan M., Gómez, Juan A. Web Simulator for the Teaching of Instruction Dynamic Scheduling. 6th International Symposium on Computers in Education, Cáceres, Spain, pp. 1-8, November 2004.
6. Vega, Miguel A., Chaves, José M. PipeSim. http://arco.unex.es/pipesim, 2006.
7. Zargham, Mehdi R. Computer Architecture: Single and Parallel Systems. Prentice-Hall, 1996.