ONLINE TUTORIAL SYSTEM FOR LEARNING SEMICONDUCTOR DEVICES IN AN ENGINEERING COURSE

W.B. Tan and L. S. Tan

Department of Electrical and Computer Engineering, National University of Singapore

4 Engineering Drive 3, Singapore 117576, Republic of Singapore

ABSTRACT

An interactive, web-based tutorial system for the teaching and learning of Semiconductor Devices is currently being developed. This consists of a server which houses a set of Java Applets, each consisting of a tutorial problem. Different parameter values of the tutorial problem are generated each time a student logs on to the system. When the student submits the answer, the system provides an immediate feedback, and help if the student encounters difficulty. The server records all the parameter values and answers for each question, given by the students for analysis by the course lecturer. This paper examines the tutorial architecture and presents the responses from the students who have used it.

KEYWORDS

Interactive tutorial, web-based courses, semiconductor devices.

INTRODUCTION

The idea of an online tutorial system for learning semiconductor devices in the Department of Electrical Engineering in the National University of Singapore (NUS) first came about when we started using a program called the Jellyfish, developed in the University of Western Australia by Scott & Stone (1996, 1998).

The tutorial system consists of a series of Java applets, servlets and a student database, housed in a server. Students will log in to the website hosting the applets and attempt the questions online. While the questions that each of the students get are the same, different parameter values are generated for everyone, to reduce the likelihood of copying. The student is required to key in an answer, which is compared with that calculated by the applet, as well as some common, anticipated mistakes. Instant feedback and hints are provided.

The tutorials are mounted on the internet and students are able to attempt the questions whenever they like, and as many times as they want to. Immediately when a student submits a feedback query or an answer, the server captures the important information and logs them into individual files. These log files are updated whenever the student attempts the question again.

Because it is common for weaker students to give up attempting tutorials when they get stuck at slightly harder questions, the tutorial system disintegrates the harder questions into parts when they fail to get the correct answer after 3 attempts, to guide the students through the various steps in deriving the final answer. Besides administering the on-line tutorial, the server also collects the answers provided by the students for analysis.

TUTORIAL SYSTEM ARCHITECTURE

The tutorial system consists of the following elements:

- Tutorial webpage
- JAVA applets
- JAVA servlets
- Database
- Server program

These 5 segments of the tutorial system communicate according to the block diagram below:

Figure 1: Tutorial System Architecture

The servlets serve mainly to capture everything we need to capture from the client side. From student ID, the time he logs in and out, the parameters generated for each question, the attempted answers, feedback queries when a student has any doubts while attempting the various steps, to any other form of numerical input, the servlets are able to capture them and sort them out accordingly into a student database. We can then read or extract part of the information we need from the database into log files for analyzing problems encountered by the students.

TUTORIAL APPLETS

After trying out the FlyingFish tutorial system in the year 2000 with one batch of students, it was realized that questions that offer no hints or help were often not attempted by weaker students. Tougher questions usually require several intermediate steps to derive the final answer, and hence knowing how to start off with a problem sum is crucial to solving questions such as these. On the other hand, hints have to be subtle or else the questions become too easy to solve. With careful planning, we developed applets that break up into parts when the students fail to get the correct answer for the first 3 attempts.

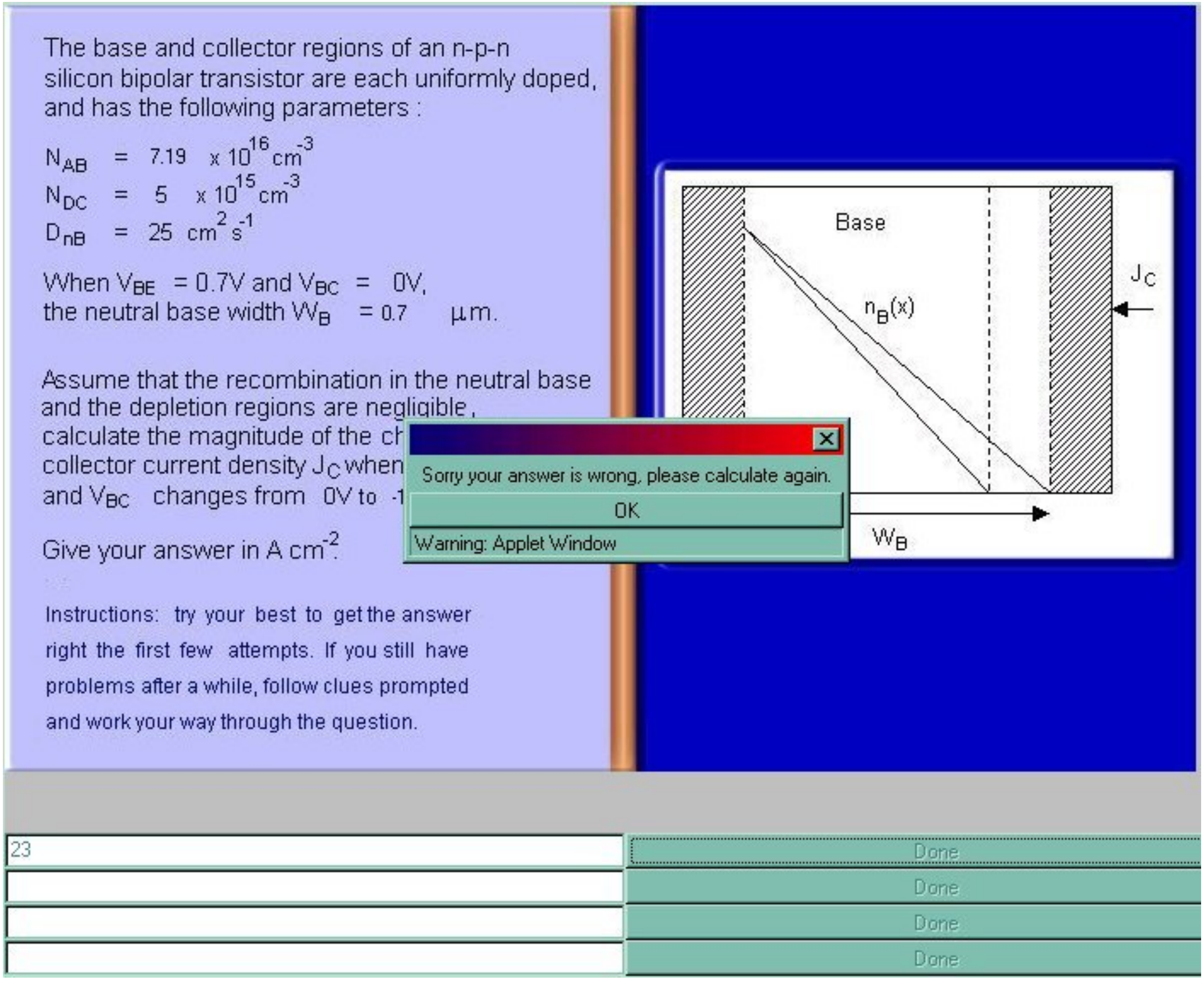

Figure 2: Question with feedback for wrong answer

Our previous trials with tutorial applets showed that such a system is not very useful if the feedback dialogue simply tells us whether we got the answer correct or wrong. This type of information can only allow teachers to tabulate the marks for individual students and for grading them at the end of the semester, without really knowing what went wrong for those who got the answers wrong.

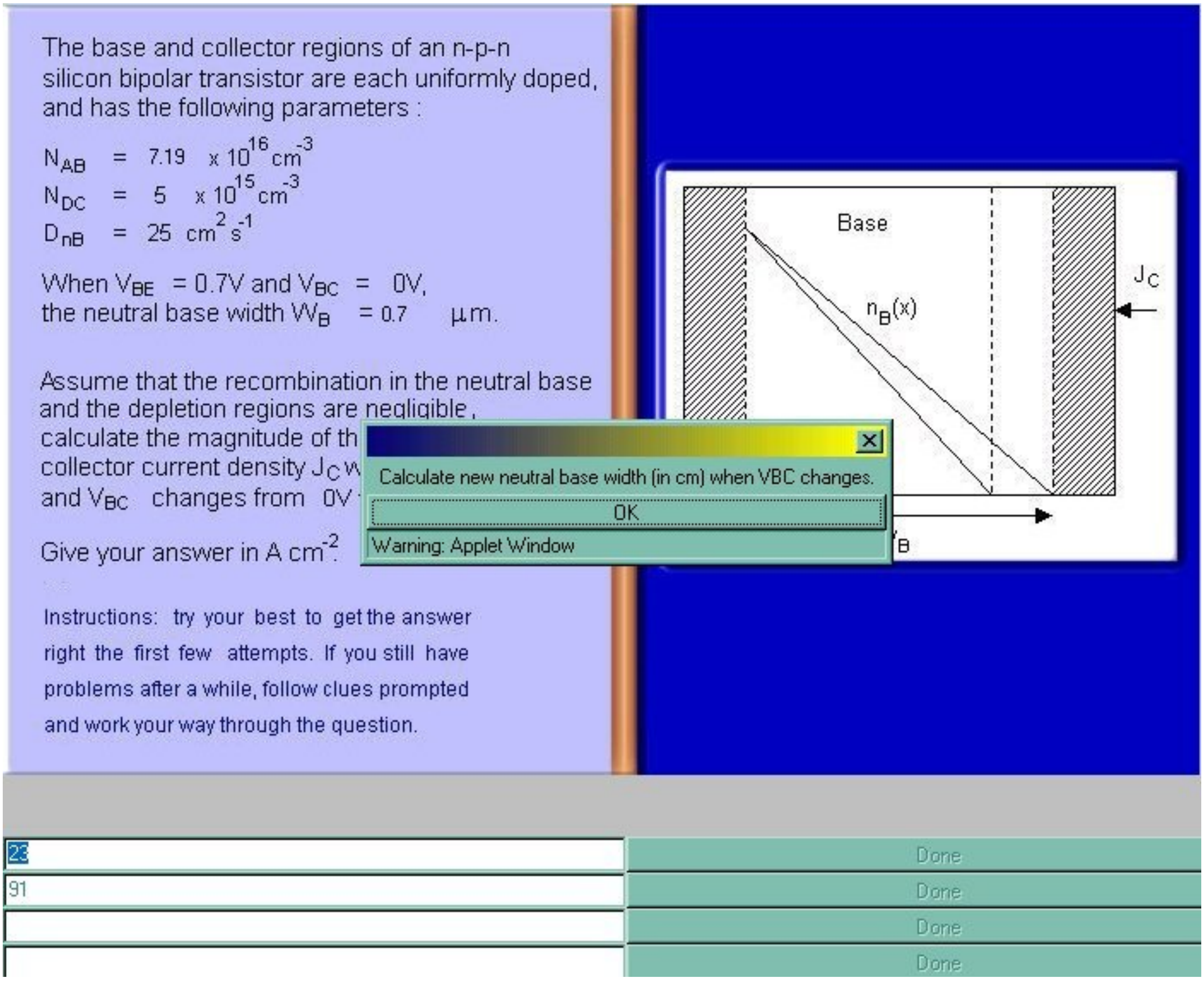

Figure 3: Dialogue box guiding the student through the first step

In the above figure, the student got the answer wrong 3 times consecutively, and the tutorial question starts to break up into the first step required in deriving the answer. With no hints to the formula he needs to derive the answer to the first step, the tutorial is trying to guide him through the steps to the final answer, but not spoon feeding him with the necessary formula and constants.

ANALYSIS

At the end of the semester when all the students have completed their tutorials, the log files showing all the data that we captured, were analyzed. A typical log file for a tutorial question is shown in Figure 4 below.

```
----------------------------------------
Wed Oct 31 09:26:35 CST 2001
From: 137.132.134.108 : 137.132.134.108
UserID = eng90078
User Name = Low Kar Seng
E-mail: null
URL: null
Comment: null
tox: 18.0
na: 2.5E16
area: 4.0E-7
correctCmax: 0.07673466666666665
answerCmax: 0.0767
correctCmin: 0.016619074660248784
answerCmin:
----------------------------------------
Wed Oct 31 09:28:13 CST 2001
From: 137.132.134.108 : 137.132.134.108
UserID = eng90078
User Name = Low Kar Seng
E-mail: null
URL: null
Comment: null
tox: 18.0
na: 2.5E16
area: 4.0E-7
correctCmax: 0.07673466666666665
answerCmax: 0.0767
correctCmin: 0.016619074660248784
answerCmin: 0.0767
----------------------------------------
```

Figure 4: Student input captured in a log file

The log files provide some important information for analyzing the way students attempt tutorials, such as the time taken to complete a question, and the number of attempts they made. While the correct answers given by the students allow lecturers to award marks to grade them, it is more often the wrong answers that provide clues to why students cannot do some questions. Just as in the above example, this student did not seem to know the difference between Cmax and Cmin, and apparently gave the same answer for both entries. Hence, it might appear to be a problem in understanding the principles behind this topic, rather than the calculations involved. Some other wrong answers shed light to a particular pattern of errors which could be avoided by the students. A lecturer can then trace back the steps, by using the parameters captured, to attempt to derive the wrong answer given, so as to find out why the student went wrong.

CONCLUSION

The challenge in this area of research boils down to the possibilities in coming up with stimulating questions to sustain the interest in students. So far, we have attempted creating single answer questions, double answer questions, drag and drop applets as well as some interactive animation programs. With these, we are looking into the development of tutorials that can break themselves up into parts to help weaker students and applets that allow students to do graphical interactive questions, such as drawing a graph online. Another useful exercise is to allow students to carry out semiconductor fabrication steps online and to visualize the various steps through the animations. However, how these programs react towards the students, depend on how the lecturers want them to do. The objective is to give some guidance to weak students while trying not to let the questions appear too easy for the strong students. Ultimately, lecturers and educators using this program play an important role in predicting the mistakes students might make while answering them, and design questions to test their abilities.

References

Scott, N.W. & Stone, B.J., (2000). *FlyingFish v1.39 Documentation*, The University of Western Australia, Perth, Australia.

Scott, N.W. & Stone, B.J., (1996). "An Integrated Computer Environment for Collaborative Learning in Engineering Dynamics", *Proc. of HERDSA'96*, The University of Western Australia, 8-12 July 1996, 755-764.

Scott, N.W. & Stone, B.J., (1998). "A flexible web-based tutorial system for Engineering, Maths and Science subjects", *Global J. of Engineering Education*, **2(1)**, 7-16.