Teaching Basics of Concurrent Errors Detection by Using Spreadsheet

Ilya Levin, Vadim Talis

School of Education, Tel Aviv University, Israel
Email: {ilia1, talisv}@post.tau.ac.il

Abstract-This paper examines the use of spreadsheets as a tool for learning principles of concurrent error detection. Basic concepts of the concurrent checking are presented by using specific spreadsheet templates. A matrix representation of a system of logical functions is used for this aim. The proposed spreadsheet simulation approach for teaching the subject achieves a theoretical goal of the lesson by using students' practical activities.

1 Introduction

The subject of concurrent error detection is normally considered an advanced topic, and therefore is usually not covered in any introductory digital electronic book. Though, the subject is fairly easily injected into a digital logic course. A spreadsheet "template" based methodology is proposed, which assists the students to visualize the error detection process.

The theory of concurrent error detection, being based on exact definitions and formal reasoning, represents a good example of well-formed academic discipline. At the same time, its practical component is mostly industry oriented and consequently requires non-formal understanding of its basic principles. A gap between the two approaches to teaching the topic demonstrates a current problem in the engineering education, which is an inconsistency between practical orientation of the engineering education and its academic deepness.

In the paper, we propose an approach for teaching the concurrent error detection principles by using spreadsheets. The approach is oriented on teaching theoretical basics of the discussed topic. Simultaneously, the proposed approach is based on developing and exploring a computer based learning environment that enables students to build logical circuits having a self-checking ability.

In this work, we illustrate a possibility to close the gap between the theoretical and the practical components of the discussed topic by using a simulation within the EXCEL based learning environment.

We assume that the students have basic skills in operating a spreadsheet and defining functions in its cells. Thus we can assume the students to be able to implement the following computer-based scenario and under the teacher guidance:

(1) Development of spreadsheet templates modeling a circuit;
(2) Observing the functioning of the circuit modeled by the template. The observing is affected both without a fault, and in the presence of the fault.
(3) Verification of the circuit from the point of its self-checking ability.

Our virtual lessons comprise four learning modules. Each of these modules implements the above three steps and is dedicated to a specific topic of the subject. The modules are related in such a way, that after performing a specific module a student will be motivated to perform the next one. In other words, each module solves the problem of a specific topic and formulates a new problem for the next module.

Modules that are discussed in the paper relate to the following topics:

1. Combinational circuits.
2. On-line checking circuits.
3. Error-detection coding.
4. Self-checking checker.

Some of these topics and appropriate spreadsheet templates for modeling the circuits will be described below.

2 Basic Definitions

Self-checking can be defined as the ability of a circuit to verify concurrently and automatically, whether there is any fault in the circuit's logic (without the need for externally applied test). Thus, self-checking circuits allows concurrent on-line error detection, that is detecting faults during the normal operation of the circuit.

A circuit is *self-testing* if for every fault from an assumed fault set, the circuit produces a non-code word at the output for at least one input code word.

A circuit is *fault-secure* for an assumed set of faults if, for any fault in the set, the circuit output is either a correct codeword or a non-codeword. It means that the circuit never produces an incorrect codeword for an input codeword.

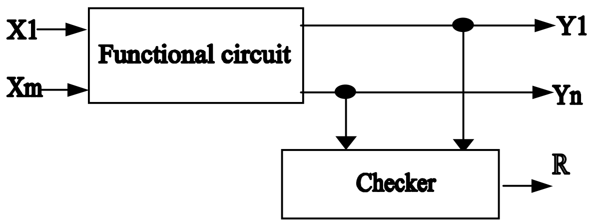

Figure 1. Concurrent checking architecture.

The circuit is *totally self-checking* if it is both self-checking and fault secure. Totally self-checking circuits are very desirable for highly reliable digital systems, since during normal operation all faults from a given set would cause a detectable, erroneous output.

A schematic diagram of a concurrent self-checking circuit is shown in Fig. 1. It consists of a functional circuit (FC) and a checker, both of which are self-checking. The function of the checker is to check validity of output codewords of the circuit.

We use a Som-of-Minterms (SOM) based checker that implements a sum of all minterms corresponding to output vectors ($Y_1,\ldots,Y_n$) of the functional circuit. When an output vector is a codeword, the corresponding minterm will be activated and consequently, the output error signal R will be equal to one. In the opposite case, when the output vector is a non-code word, no SOM checker's minterm will be activated and the error checker's signal R will be equal to 0, which means that an error is detected.

3 Logic Circuit within Spreadsheet

Spreadsheets are a useful and flexible modeling tool. The spreadsheets may be regarded as a normal calculator-type tool, but they are also universal homogeneous two-dimensional fields, which can be used for implementation of various computational and logical functions forming any functional circuits.

The paper [1] aims at studying the use of spreadsheets for simulation of a system of logical functions in a matrix form usually called Programmable Logic Arrays (PLA) or a Decision Table. The main idea of such a simulation is based of a two-layers spreadsheet template. The first *programmable* layer relates to a system of logical functions in its matrix form, while the second *functional* layer implements the universal PLA matrix. Students are able to implement any system of logical functions by reprogramming the first layer of the spreadsheet template. Moreover, they are able to run the simulation of the circuit and analyze its functioning "on-line".

Logical capacity of spreadsheets has been mentioned in [2] and used for teaching principles of the error detection for digital courses.

The present study also uses the logical capacity of spreadsheets as a basis for developing systems of logical functions. The simplicity of both defining and redefining logic systems is utilized in our study for modeling the functional circuits (Fig. 1).

4 Concurrent Checking Architecture within Spreadsheet

Using the above methods, we construct an EXCEL template that simulates functioning of the self-checking functional circuit implementing a certain system of logic functions represented in a matrix form. We assume that students are able to create a spreadsheet model of the logic system.

We then propose to construct the checker as a Sum-of-Minterms checker [3]. The SOM checker implements a logical function and, consequently, can be implemented by the matrix model.

The programmable layer of the spreadsheet model for the self-checking architecture is shown in Tab. 1.

Table 1. Programmable matrix of Spreadsheet template for Self-checking scheme

x_1	x_2	y_1	y_2	y_3	
0	0	0	0	1	
0	1	1	1	0	
1	0	0	0	0	
1	1	1	0	1	
		y_1	y_2	y_3	R
		0	0	1	1
		1	1	0	1
		0	0	0	1
		1	0	1	1

In this table: x_1 and x_2 are inputs of the functional circuit (FC), y_1, y_2 and y_3 are its outputs, R is an error signal.

The upper part of the table is a spreadsheet template of the FC matrix. The lower part of the table is the matrix of the SOM checker. The last column of the checker's template represents an error function.

5 Example of a lesson for studying self-testing property

Students study basics of the error detection, (which are unidirectional errors and an unordered coding) by developing a spreadsheet model of a functional circuit implementing a system of logic functions and by constructing the SOM checker for this system (Tab. 1).

After developing the model, students analyze the template. They introduce various faults and analyze results.

Students "inject" a fault into the spreadsheet model of the functional circuit and observe/analyze the consequence of the fault injection using the R output of the circuit. After that, they add checking bits to the output vector of the FC and analyze behavior of the architecture in the new improved situation. The aim of adding the checking bits is to define limitations of the developed template.

Students come to the conclusion that the proposed model enables detecting all errors that lead to the occurrence of a non-code output word. However, some faults may lead to appearance of a code vector, which means that this fault is undetectable. Example: if due to a fault, the 3-th and 4-th product terms (see the 3-th and 4-th lines of Tab. 1) are equal to one; the resulting output vector will be equal to codeword 101. In this case, the error signal is absent despite the presence of the fault. The students have to analyze the reason of this phenomenon, and will be asked to suggest their own solution of this problem. In other words, they will have to propose a method of achieving the self-testing property of the system. One well-known solutions of the problem is to complement outputs of the FC by checking bits representing any unordered code [4]. The teacher may suggest this elegant solution after explaining the nature of the problem. The students may complete the template with the unordered code as shown in Tab. 2.

Table 2. Spreadsheet Template for Unordered Coded Circuit

x_1	x_2	y_1	y_2	y_3	Checking	bits	
0	0	0	0	1	0	1	
0	1	1	1	0	0	0	
1	0	0	0	0	1	0	
1	1	1	0	1	0	0	
		y_1	y_2	y_3	Checking	bits	R
		0	0	1	0	1	1
		1	1	0	0	0	1
		0	0	0	1	0	1
		1	0	1	0	0	1

Upon building such a template, the students can verify the self-testing property of the functional circuit and the checker.

6 Conclusions

We have presented a method for teaching the subject of concurrent error detection to electrical engineering students in the framework of a digital design course.

Students are taught to build spreadsheet templates for modeling digital circuits, which allow visualizing the error detection process. The proposed method can be extended for teaching other concurrent checking concepts and methods.

References

1. Levin I., "Matrix Model of Logical Simulator within Spreadsheet", International Journal of Electrical Engineering Education, Vol.30, N3, pp.216-223, 1993.
2. Anneberg L., Error Detection and Correction Templates for Digital Courses. IEEE Transaction on Education, Vol. 42, No. 2, May 1999.
3. Levin I., Karpovsky M. "On-line Self-Checking of Microprogram Control Units", 4-th International On-line Testing Conference, Capri, Compendium of papers, pp.153-159, 1998.
4. Lala P., "Self-checking and Fault-Tolerant Digital Design", Morgan Kaufmann Publishers, San Francisco / San Diego / New-York/ Boston/ London/ Sydney/ Tokyo, 2000.