Proceedings of the 9th Biennial ASME Conference on Engineering Systems Design and Analysis
ESDA 2008
July 7-9, 2008, Haifa, Israel

ESDA2008-59125

USING THRESHOLD FUNCTIONS IN TEACHING ELECTRONICS

Gabriel Shafat
gabis@afeka.ac.il
Afeka Academic College of Engineering

Binyamin Abramov
avramov@post.tau.ac.il

Ilya Levin
i.levin@ieee.org
School of Education, Tel Aviv University

ABSTRACT

Teaching of digital electronics and the teaching of analog electronics differ significantly. The methods in use today differ in two major points: the required mathematical background and the used didactic methods. The well-known gap between the *analog* and the *digital* paradigms in teaching electronics has motivated the present study. The paper introduces a novel approach for electronics course teaching. The approach uses a concept threshold functions. Threshold functions have three remarkable properties that are suitable for the purposes of teaching an electronics course. The first property is the simplicity of the functions' representation and implementation; the essence of a threshold function is understandable on the common sense level. The second property is the dual analog-digital nature of the threshold functions. The definition of a threshold function usually includes both Boolean and arithmetic portions and weaves together the two alternative domains: digital and analog. Since students are familiar with regular arithmetic functions from previous math courses, the addition of Boolean concepts is simple to grasp. The possibility to transform any threshold function from one domain to another, serves as a powerful tool for processes teaching. The third property we consider is the multiple representations possible for threshold functions. Besides the classical Boolean and arithmetic representations, a threshold function can be represented in the format of an electric/electronic circuit and also can be represented in a spatial form, by three-dimensional visualization for better understanding the functional properties of threshold functions. The paper discusses a problem-based learning with two main types of problems: synthesis and analysis problems of threshold elements. While the analysis problem is relatively simple, the problem of optimal synthesis is NP-complete, and equivalent to a well-known optimization problem that exists also in linear programming. Using the linear programming for teaching the synthesis of a threshold element is a challenging pedagogical task. The paper describes an approach for solving this task. A number of real-world problems may be formulated and efficiently solved by using the proposed threshold-based approach, for example the problems of event-driven control, fuzzy control, linear optimization, self-regulation. These problems formulate as students' assignments, and are used in the lesson. These exercises convert a lesson of electronics into an interesting, challengeable and useful educational event. Introduction of the threshold approach into the electronics curriculum enables the students to acquire much deeper understanding of electronic systems.

INTRODUCTION

Logic design of switching functions is usually based on electronics gates. A different approach for representation switching elements is using the threshold logic. The threshold element can be considered as a generalization of the conventional gates. A large class of switching functions can be realized by the threshold element through multiple combinations of weights and threshold values. Generally, the threshold element is a discrete logic element, and the threshold function realizing a threshold-element operation requires complex analysis.

The history of the study and application of the threshold logic as means of system simulation begins with the work [1], where a neuron model is described by using switching functions (Figure 1). On the architecture of the neuron model, there are four requirements in addition to the spatial summation and the threshold which give the basic characteristics to the neuron. First, the model consists of all digital elements in order to realize the circuit by VLSI. Second, the inputs should be expressed as analog value for various applications. Third, the model needs the characteristic of the temporal summation in the mutual connected network. Fourth, the value of the connection weight can be variable for learning procedure. An electronic circuit that computes a threshold function is named a *threshold gate*. A logic circuit built from threshold gates is named a *threshold circuit*.

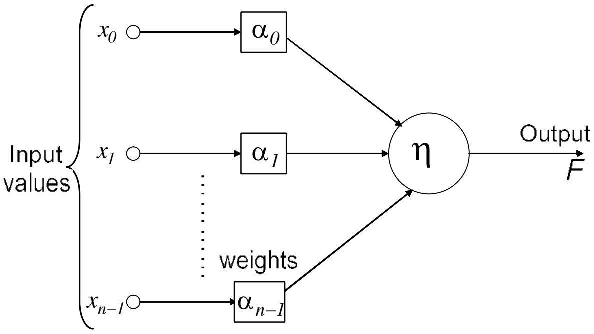

Figure 1. Neuron model

The function $F(x_0, x_1, ..., x_{n-1})$ is a *threshold function*, if it can be represented as:

$$F(x_0, x_1, ..., x_{n-1}) = Sign(\alpha_0 x_0 + \alpha_1 x_1 + ... + \alpha_{n-1} x_{n-1} - \eta);$$

$$F(x_0, x_1, ..., x_{n-1}) = Sign(\sum_{j=0}^{n-1} \alpha_j x_j - \eta);$$

Where $SignA = \begin{cases} 1 & if \quad A \geq 0 \\ 0 & if \quad A < 0 \end{cases}$

α_j - weight of input (variable) x_j, η -threshold.

The interest in using threshold gates (functions) instead of standard logical AND, NOT and OR gates relies on the fact that threshold elements are more powerful than Boolean (AND, OR, NOT) gates and as a consequence, the size of the circuits that can be constructed to compute the desired functions can be smaller. The next figure (Figure 2) presents the NOR gate implemented by threshold element.

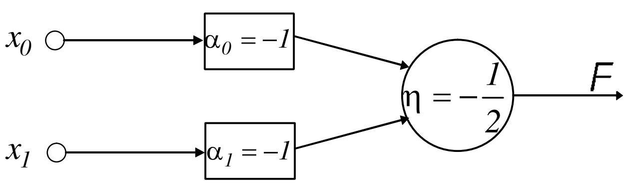

Figure 2. A two input NOR gate

Since the NOR gate is functionally complete, any logic function can be realized by the threshold elements only. In the given example, when $\alpha_0 = \alpha_1 = 1$ and $\eta = \frac{1}{2}$, the function is $f = x_0 \vee x_1$ (OR), for same weights and threshold value of $\eta = 1\frac{1}{2}$, the function is $f = x_0 x_1$ (AND). When $\alpha_0 = \alpha_1 = -1$ and $\eta = -1\frac{1}{2}$, the function is $f = \overline{x_0 x_1}$ (NAND).

A small example (Figure 3) is presented in this section to motivate the need for our threshold network synthesis methodology. Consider the Boolean network (Figure 3a) which has seven gates and five levels (including the inverter). If we simply replace each gate with a threshold gate, the resulting network (Figure 3b) will contain seven threshold gates and five levels.

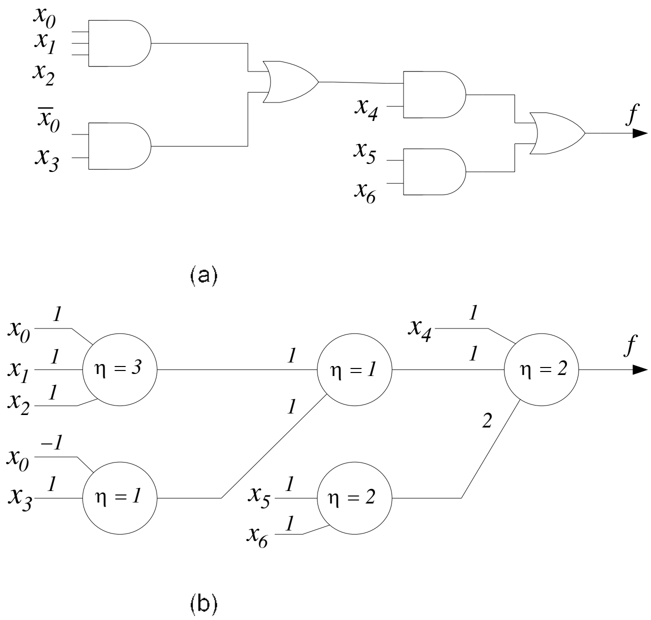

Figure 3. The Boolean and threshold networks

Since the number of possible combinations of weights and threshold is large, many switching functions can be realized by only one threshold element. Research in threshold logic synthesis was done mostly in the 1960s [2, 3]. Nowadays, different implementations of circuits by threshold gates are available, and several theoretical results have been obtained [4] together with different applications [5]. If a function is realizable by a single threshold element then by an appropriate selection of negations of input variables, may be realized by a threshold element whose weight has any desired signs. For example, let a threshold function $f(x_0, x_1, \ldots, x_{n-1})$ be positive in variable x_i and the weight-threshold vector $A = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}, \eta\}$. Since f is positive in variable x_i, there exists a set of values $v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n-1}$, for inputs $x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n-1}$ such that

$f(v_0, \ldots, v_{i-1}, 1, v_{i+1}, \ldots, v_{n-1}) = 1$ and

$f(v_0, \ldots, v_{i-1}, 0, v_{i+1}, \ldots, v_{n-1}) = 0$.

Hence,

$\alpha_0 v_0 + \ldots + \alpha_{i-1} v_{i-1} + \alpha_i + \alpha_{i+1} v_{i+1} + \ldots + \alpha_{n-1} v_{n-1} \geq \eta$

and $\alpha_0 v_0 + \ldots + \alpha_{i-1} v_{i-1} + \alpha_{i+1} v_{i+1} + \ldots + \alpha_{n-1} v_{n-1} < \eta$,

$\alpha_i > 0$. The weight α_i associated with a threshold function which is positive in variable x_i, is positive. The weights associated with a threshold function which is positive in all of its variables, are all positive. Another aspect of threshold functions is the possibility for geometric representation by the hypercube model. Equation $\sum_{j=0}^{n-1}\alpha_j x_j - \eta = 0$ is the equation of a hyper-plane in the n-dimensional space. For $n = 3$, we have the equation $\alpha_0 x_0 + \alpha_1 x_1 + \alpha_2 x_2 - \eta = 0$ describing the plane in three-dimensional space. The following figure (Figure 4) is a threshold function described by the equation $D(Y,P) = \frac{\alpha_0 x_0 + \alpha_1 x_1 + \alpha_2 x_2 - \eta}{\sqrt{\alpha_0^2 + \alpha_1^2 + \alpha_2^2}}$. Where $\alpha_0 = \alpha_1 = \alpha_2 = 1$, in this case, we have $D(Y,P) = \frac{x_0 + x_1 + x_2 - 1}{\sqrt{3}}$.

The value of the threshold function in the point Y (cube vertex) is determined by the sign of the distance from this point to the hyper-plane determined by the weights of the variables and by the threshold. For all of the cube vertices that are located above or within the hyper-plane, the value of threshold function is equal to 1.

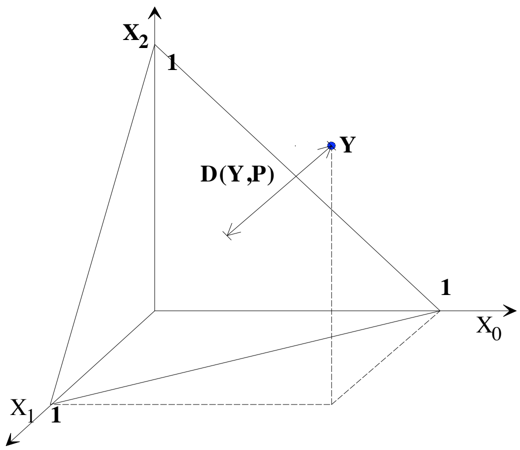

Figure 4. Geometrical representation of threshold function

For all of the cube vertices that are located below the hyper-plane, the value of threshold function is equal to 0.

The ability to show of geometric representation of the threshold functions is of great pedagogical value and is the concentration of this work. Threshold functions also enable us to select the implementation method: analog or digital, and the transition from one element base to another element base capability inherent in the basis of a threshold function, allowing us to view them as a universal platform for the teaching of digital and analog electronics through a common area of knowledge. In general terms, the analog model of threshold element is presented in Figure 5, where the resistors $R_\eta, R_0, \ldots, R_{n-1}$ values are compared to the threshold level and inputs weight respectively.

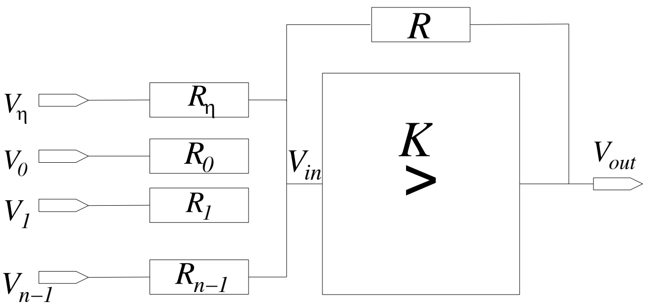

Figure 5. Generalized analog representation of threshold element

This paper considers the above thresholds function's properties from the point of teaching electronics.

SYSTEM MODELING BY THRESHOLD FUNCTION

The threshold functions are used as means for systems modeling at a high level of abstraction. In the following example the model is independent from implementation. We consider the problem of designing of two numbers (X and Y of n bits) adder (Figure 6), where to simplify the schemes $n = 2$.

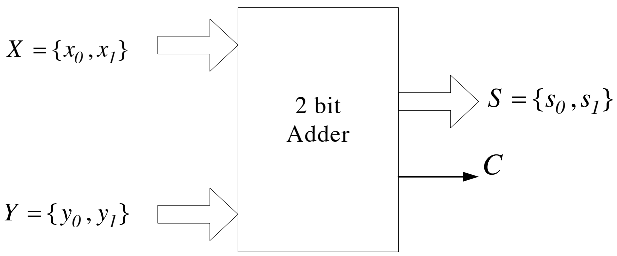

Figure 6. Two bit adder

For simplification let examines the representation of function C (carry out), the remaining bits are presented in similar manner as threshold network [6]. In the traditional design methodology, based on truth tables and minimization of the Boolean functions (for example, through K-map), the function is described by the following expression as SOP:

$$C(x_1, x_0, y_1, y_0) = \sum(7, 10, 11, 13, 14, 15);$$
$$C(x_1, x_0, y_1, y_0) = x_0 y_1 y_0 \vee x_1 x_0 y_0 \vee x_1 y_1;$$

The C function may be represented alternatively as a threshold function. The weight of the input variables and the threshold can be defined formally and heuristically. In this example, we can conclude that $C = 1$ when the result of the summing greater or equal than 4 (decimal value), i.e. the threshold is 4. Furthermore, the analysis of the possible combinations of input variables that amount to 4 or more, will lend the following weights to the input variables: $\alpha_{x_1} = \alpha_{y_1} = 2; \alpha_{x_0} = \alpha_{y_0} = 1$.

Finally we will have the following expression:

$$C(x_1, x_0, y_1, y_0) = Sign\left(2x_1 + x_0 + 2y_1 + y_0 - 4\right)$$

The above expression describes the behavior of the system, regardless of its implementation. Transition from the mathematical model of the threshold function to its digital implementation (synthesis problem) is simple and formal. The following K-map based example (Figure 7) illustrates the typical problem of logic synthesis. Cells of the K-map include the value of the sums of input variable weights. Cells where the sum is more or equal to the threshold comprise the on-set of the Boolean function.

y_0y_1 \ x_0x_1	0	1	3	2
0	0	1	3	2
1	1	2	4	3
3	3	4	6	5
2	2	3	5	4

Figure 7. K-map presentation of threshold function

As a result we obtain the following expression:

$$C(x_1, x_0, y_1, y_0) = x_0 y_1 y_0 \vee x_1 x_0 y_0 \vee x_1 y_1;$$

Not only the synthesis problem may solved by using the above technique. Let us consider an analytical problem that may be also solved. For example, implement the threshold network (Figure 8) by a single threshold function.

The functions of every layer are represented by the following expressions:

$$z = x_1 + x_2 + x_3 - 3$$

$$y_1 = z + x_4 + x_5 - 1$$

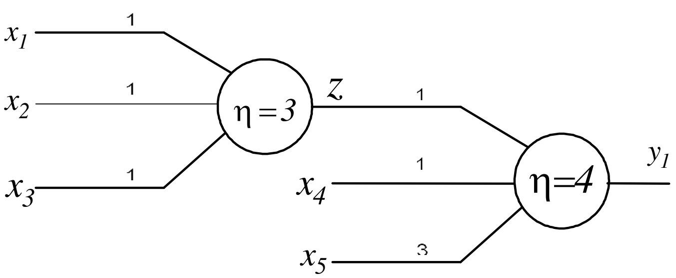

Figure 8. The network of threshold elements

From these equations it is very simple to obtain following Boolean expressions:

$$z = x_1 x_2 x_3$$

$$y_1 = \sum \begin{pmatrix} 7,8,9,10,11,12,13,14,15,16,17, \\ 18,19,20,21,22,23,24,25,26,27, \\ 28,29,30,31 \end{pmatrix}$$

And eventually we can get the expression:

$$y_1 = x_1 x_2 x_3 + x_4 + x_5$$

To determinate the threshold function it is necessary to nominate the weight of input variables and the threshold. There are at most 2^ℓ distinct cubes for a logic function of ℓ variables and this leads to 2^ℓ inequalities, which represent the constraints [2]. Define new weights relations between all variables:

$$\left.\begin{matrix} \alpha_1 + \alpha_2 < \alpha_4 \\ \alpha_3 + \alpha_2 < \alpha_4 \\ \alpha_3 + \alpha_1 < \alpha_4 \end{matrix}\right\} \left.\begin{matrix} \alpha_1 + \alpha_2 < \alpha_5 \\ \alpha_3 + \alpha_2 < \alpha_5 \\ \alpha_3 + \alpha_1 < \alpha_5 \end{matrix}\right\}$$

It follows those minimal possible weights and threshold:

$$\alpha_1 = \alpha_2 = \alpha_3 = 1; \alpha_4 = \alpha_5 = 3$$

Using this method the final function we get is:

$$y_1\left(x_1, x_2, x_3, x_4, x_5\right) = sign\left(x_1 + x_2 + x_3 + 3x_4 + 3x_5 - 3\right)$$

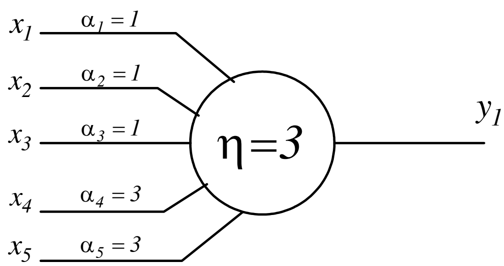

Figure 9. Incorporated threshold function

Figure 9 presents the scheme of the above threshold function that is equivalent to the network of threshold functions shown in Figure 8. We have demonstrated properties of threshold functions that enable to use them for electronics system modeling. We also demonstrated solutions to problems encountered in the use of threshold functions such as analysis and synthesis problems.

ANALOG AND DIGITAL PARADIGM

The definition of a threshold function usually includes both Boolean and arithmetic portions and weaves together the two alternative domains: digital and analog [7]. Since students are familiar with regular arithmetic functions from previous math courses, the addition of Boolean concepts is simple to grasp. The possibility to transform any threshold function from one domain to another, serves as a powerful tool for processes teaching. At has been shown and demonstrate that a problem represented by a threshold function can be represented and implemented by digital logic gates or by analog amplifiers and resistors. Analyzing the function to be implemented by comparing gauge able performance, frequency response etc. enables us to decide which implementation is preferable digital or analog. The digital implementation of the carry function for 2-bit adder is shown in Figure 10, where the Boolean function obtained is implemented on the basis of AND/OR gates.

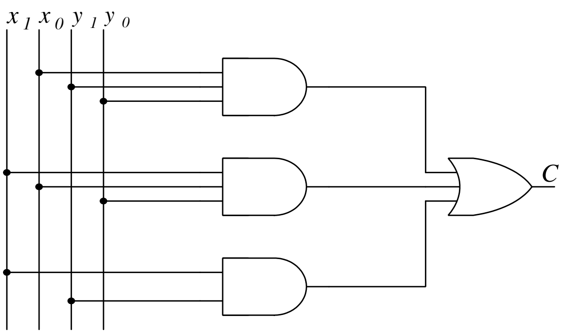

Figure 10. The AND/OR implementation of Carry function

Analog circuits are of great importance in electronic system design since the world is fundamentally analog in nature. While the amount of digital design activity far outpaces that of analog design, most digital systems require analog modules for interfacing to the external world. Techniques for analog circuit design automation began appearing about two decades ago. Due to the growth of digital electronic, the design of analog electronic circuits has played a minor role during the last two decades. Presently, however, analog circuit design is rapidly becoming an important factor again, particularly in the area of mixed-signal design. The analog application for the carry function of 2 bit adder is shown in the following circuit (Figure 11). This circuit was also given as a design problem to junior electronics students.

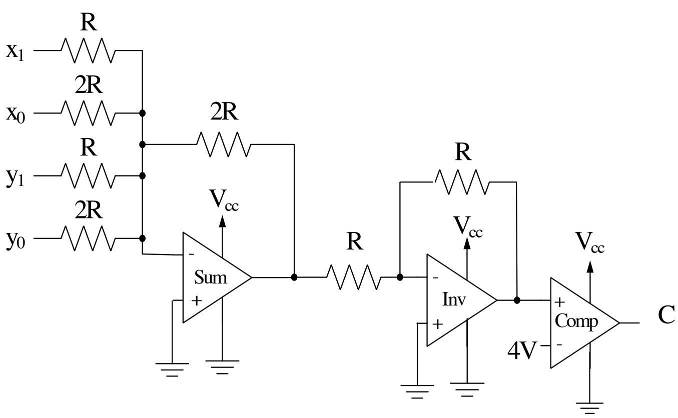

Figure 11. The analog implementation of Carry function

CASE STUDY

The topic "Threshold function for electronics systems" deals with the electronic systems modeling on the high level of abstraction. Representing of electronics systems using threshold functions was teaches in a class of 40 undergraduate's electrical engineering students. The topic was a part of "Introduction to logic design" course, was teaches during four hours. This part of the course concludes with a practical task, that solved by using of threshold functions. About 30% of students preferred the analog implementation of given problem and the rest had solved the problem digitally.

At the beginning of the course students interviewed for preference in implementing electronic systems. The same test is invited them to the end of the threshold functions topic. The test results show unambiguous trend of deepening understanding surveyed subject. Also, there has been a decrease in the number of students prefer to use the classic methods of electronics circuits synthesis. This is because part of students' choice the threshold functions as a new tool for synthesis.

CONCLUSION

This paper, demonstrates a possibility to teach modelling and synthesis of electronics systems by using the concept of threshold functions. It shifts the focus of the course from the implementation level to modelling level. The paper gives the motivation to teach both analog and digital electronics by using the same pedagogical approach. We presented relevant examples of lectures and assignments of introductory logic course. The versatile nature of threshold circuits allows "what-if" analysis that can be used to strengthen students understanding of electronics. The authors have found the threshold based teaching to be efficient and enhancing the student's pacification. Introduction of the threshold approach into the electronics curriculum enables the students to acquire much deeper understanding of electronic systems. The preliminary results of integration of the threshold logic approach in the academic curricula indicate convergence between analog and digital paradigms in teaching electronics.

REFERENCES

[1] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, vol 7, pp. 115 – 133, 1943.

[2] M.L Dertouzos, Threshold Logic: A Synthesis Approach. Cambridge, MA: The M.I.T. Press, 1965.

[3] S. Muroga, Threshold Logic and its Applications, Wiley, New York, 1971.

[4] K.Y. Siu, V.P. Roychowdhury, and T. Kailath. Depth-Size Tradeoffs for Neural Computation IEEE Transactions on Computers, pp. 1402-1412, 1991.

[5] V. Beiu, J.M. Quintana, J.M. and M.J. Avedillo. LSI implementations of threshold logic - A comprehensive survey, IEEE Trans. Neural Networks, vol 14, pp. 1217-1243, 2000.

[6] S. Vassiliadis, S. Cotofana, K. Bertels, "2-1 Addition and Related Arithmetic operations with Threshold Logic", IEEE Transactions on Computers, 45, 1996, vol 9, pp.1062-1067.

[7] V. Varshavsky "β-Driven Threshold Elements", Proceedings 8th Great Lakes Symposium on VLSI, Lafayette, Lo, USA, IEEE Computer Society, pp 52-58, 1998.