COMBINATIONAL LOGICAL CIRCUITS SIMULATION USING FEED FORWARD NEURAL NETWORKS

by
Remus Joldeş, Ioan Ieană, Emilian Ceuca, Corina Rotar

ABSTRACT. The combinational logic circuit (CLC) is an important chapter in the project activity of the electronic equipments. If the number of variables is greater than three the project activity with Veitch-Karnaugh diagrams become very difficult. On the other side, the feed-forward artificial neural networks have several characteristics like noise immunity, fault tolerance etc. that can be advantages for logical circuits made by these networks. In this papers we presents a new method in simulation of the combinational logic circuit with the neuronal network. There are two important advantages that we can remark: the number of transistors is reduced and the design will become simpler.

1. INTRODUCTION

The traditional calculator doesn't always manage to face the problems that require intensive calculus such as pattern recognition, robots` movement control, decisions` taking based on a large quantity of data with noise etc. As a result, other methods of information processing were approached and distributed processing is one of them. One of these non-conventional direction of information processing, that meets many of the above requirements is represented by artificial neural networks (neural calculus).

The present paper approaches the simulation, with the help of neural networks, of combinational logical circuits (CLC). Section 2 presents notions about CLC design and illustrates the difficulty of designing complex CLC. Section 3 shows an example of simulation with feed-forward artificial neural networks. In the last section some conclusions on our approach are shown.

2. COMBINATIONAL LOGICAL CIRCUITS

A combinational logic circuit (CLC) is an electronic circuit with n inputs, noted by $X_1, X_2, \ldots, X_n$, and m outputs, noted by $Z_1, Z_2, \ldots, Z_m$, for which the outputs could be expressed according to inputs using a mathematical model such as:

$$Z_1 = f_1(X_1, X_2, \ldots, X_n)$$
$$Z_2 = f_2(X_1, X_2, \ldots, X_n)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_I = f_i(X_1, X_2, \ldots, X_n)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_m = f_m(X_1, X_2, \ldots, X_n)$$

If we denote by X={ X_1, X_2, …, X_n } the set of input variables and by Z={ Z_1, Z_2, …, Z_m } the set of the output variables, then a combinational logical circuit could be easily described mathematically by the triplet CLC = (X, Z, F), in which the input-output function $F : X \rightarrow Z$ is independent of time.

In the synthesis of a circuit CLC generally one starts by classifying the functioning conditions according to the requirements imposed by a table of truth and the specification of the operation and non-operation state. The following steps are involved:

- Problem utterance; Formulation of the truth table;
- Minimization of the truth function;
- Correlated minimization of the commutation functions; Scheme analysis and hazard elimination;
- Hardware implementation of the logical functions.
-

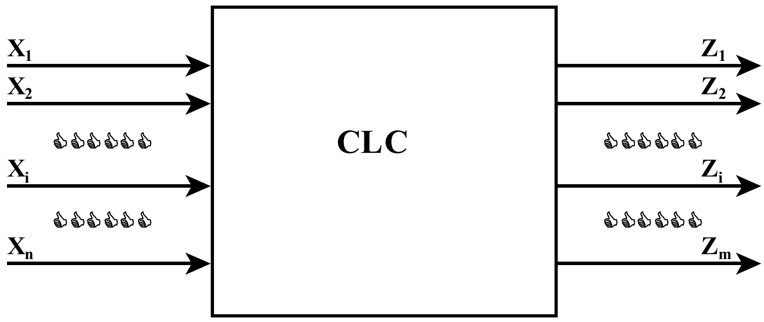
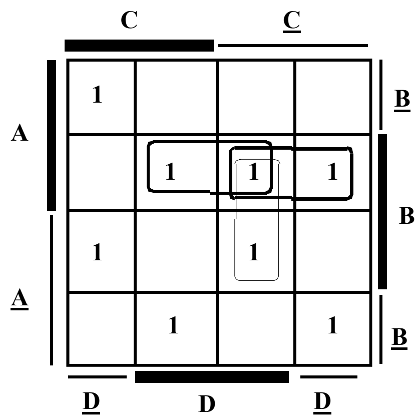


Fig. 1. CLC representation

A delicate problem in the case of the classical approach of CLC circuits` synthesis is that not always an absolute optimum rigorous scheme is achieved. Thus, in the case of synthesis of the complex functions with a high number of input variables (those with n>6), with many outputs and undetermined states, the algebraic and topological methods are very difficult to be applied.

In the following we illustrate the problem of designing CLC by an example. In figure 2 we show the process of minimisation with Veitch-Karnaugh diagram, and in figure 3 is shown the resulting CLC.



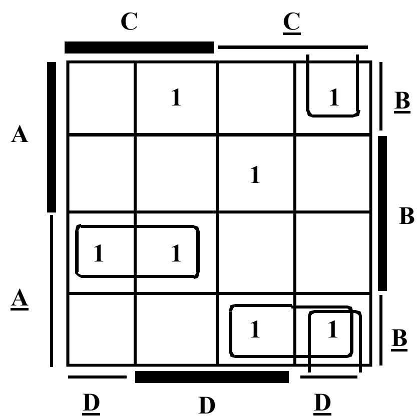
Nr.	A	B	C	B	L	K
0	0	0	0	0	1	1
1	0	0	0	1	0	1
2	0	0	1	0	0	0
3	0	0	1	1	1	0
4	0	1	0	0	0	0
5	0	1	0	1	1	0
6	0	1	1	0	1	1
7	0	1	1	1	0	1
8	1	0	0	0	0	1
9	1	0	0	1	0	0
10	1	0	1	0	1	0
11	1	0	1	1	0	1
12	1	1	0	0	1	0
13	1	1	0	1	1	1
14	1	1	1	0	0	0
15	1	1	1	1	1	0

$L=\underline{A}.\underline{B}.\underline{C}.\underline{D}+\underline{A}.\underline{B}.C.D+\underline{A}.B.\underline{C}.D+\underline{A}.B.C.\underline{D}+A.\underline{B}.C.\underline{D}+$

$+A.B.\underline{C}.\underline{D}+A.B.\underline{C}.D+A.B.C.D$

$L_{min}=A.\underline{B}.C.\underline{D}+\underline{A}.B.C.\underline{D}+\underline{A}.\underline{B}.C.D+\underline{A}.\underline{B}.\underline{C}.\underline{D}+A.B.D+$

$+A.B.\underline{C}+B.\underline{C}.D$



Legend:

$\underline{A}$ = NON (A)

A.B = A AND B

A+B = A OR B

$K=\underline{A}.\underline{B}.\underline{C}.D+A.\underline{B}.C.D+\underline{A}.B.C.D+A.B.\underline{C}.D+\underline{A}.\underline{B}.\underline{C}.D+$

$+A.\underline{B}.\underline{C}.\underline{D}+\underline{A}.\underline{B}.\underline{C}.\underline{D}$

$K_{min}=\underline{A}.B.C+\underline{A}.\underline{B}.\underline{C}+A.\underline{B}.C.D+\underline{B}.\underline{C}.\underline{D}+A.B.\underline{C}.D$

Fig. 2. The synthesis of the CLC using Veitch-Karnaugh diagram

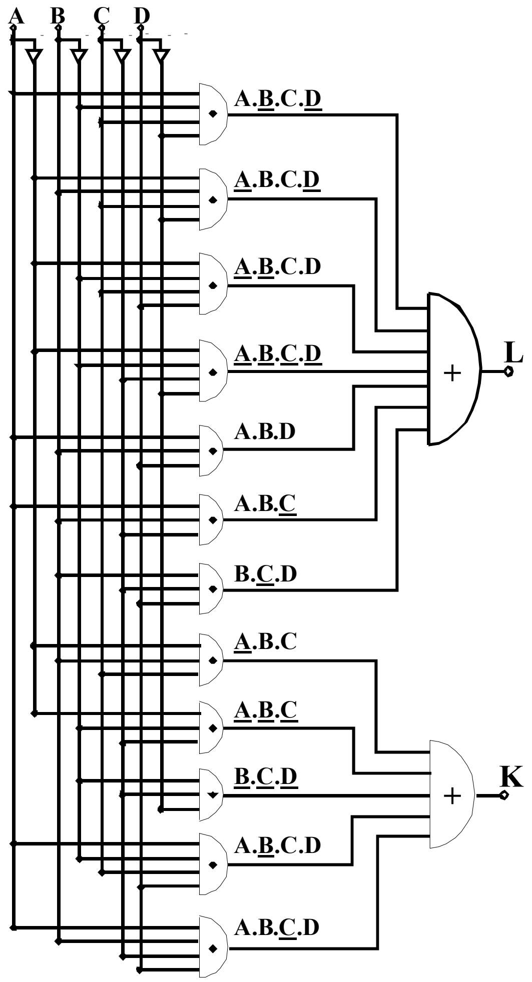


Fig.3. The resulting CLC

3. CLC SIMULATION WITH THE HELP OF FEED-FORWARD NEURAL NETWORKS

This paper proposes a special treatment of these CLC's by means of the utilization of neural networks. The solution was suggested by the functional similitude existing between a CLC and a neural network with n inputs and m outputs. In fact, a neural network of three layers was used, the first layer having n (4) input neurons and the third m (2) output neurons (fig. 4). For the neural network the set of input and output data is given by the truth table in figure 2.

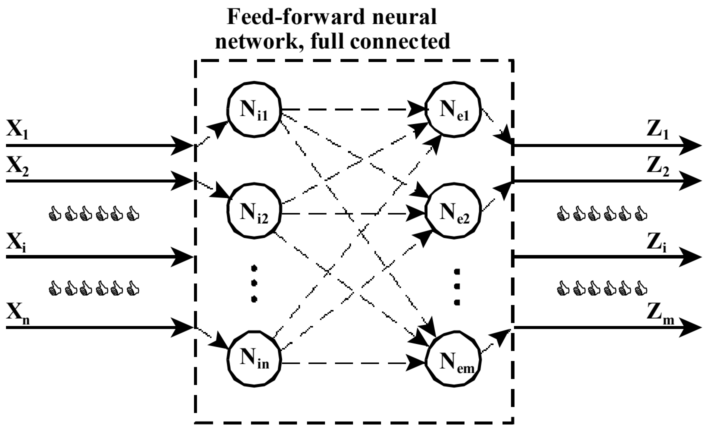


Fig. 4. Neural simulation of CLC

The simulations have shown that, for a network in configuration 4:6:2, the I/O dependence of the CLC can be already achieved. If we note that for this network are necessary 12 neurons (including the passive input neurons), it results an equivalent of 12 gates, in opposition with the CLC in figure 3 which uses 18 gates of several kinds. It is obvious that, in hardware implementation, a neural approach for realizing CLC's functions can be competitive. In figure 5 one can see the real output of the "neural" CLC and the desired output, for every combination of inputs in truth table.

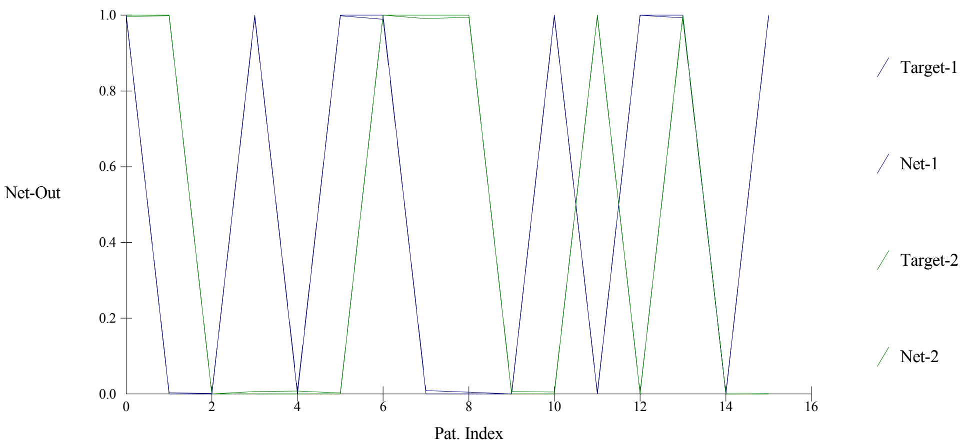


Fig. 5. Outputs of trained neural network

4. CONCLUSIONS

As a conclusion, the methodology of combinational circuits simulation, with the help of neural networks, requires the following steps:

Step 1. Formulation of the truth table for the chosen combinational circuit;

Step 2. Selection of a neural network with three layers which has n neurons on the first layer, a number of neurons in the hidden layer that will be find and m neurons in the output layer;

Step 3. Training the neural network using the truth table;

Step 4. Verification of the correct functioning of the network;
Step 5. Network hardware implementation.

REFERENCES

[1]. Dumitra∏ Adriana: **Proiectarea re⇔elelor neuronale artificiale**, Casa editorial| Odeon, Bucure∏ti, 1997.

[2]. Dumitrescu D., H. Costin – **Neural networks. Theory and applications**. Ed. Teora, Bucharest, 1996.

[3]. Joldeş R. – **Neural Networks. Basic notions. Implementations.**, Seria Didactica, “1 Decembrie 1918”University, Alba Iulia, 1995.

[4]. Ştefan G. M., I. V. Drăghici, T. Mureşan, E. Barbu – **Digital integrate circuits**, Ed. Didactică şi Pedagogică, Bucharest, 1983.

Authors:
Remus JOLDEŞ, rjoldes@uab.ro
Ioan ILEANĂ, iileana@lmm.uab.ro
Emilian CEUCA, eceuca@lmm.uab.ro
Corina ROTAR, crotar@lmm.uab.ro
“1 Decembrie 1918” University of Alba Iulia