CMOS based β-Driven Threshold Elements with Functional Inputs[1]

V. Varshavsky*, V. Marakhovsky, I. Levin*****

* Neural Networks Technologies Ltd Israel, victor@nnt-group.com
** The University of Aizu, Japan, marak@u-aizu.ac.jp
*** Tel Aviv University, Israel, ilia1@post.tau.ac.il

I. Introduction

During the last ten years the growing interest can be noted in using threshold elements as a functional basis of artificial neural networks [1-4]. This interest to threshold elements and threshold logics, can be explained by the fact that functional capabilities of threshold elements are significantly wider in comparison with the traditionally based ones'. However, the effectiveness of using the threshold basis basically depends on the complexity of implementation of the threshold element.

In [5-7] a so-called β-driven CMOS threshold element was suggested. This element requires only one transistor per a functional input. Each input has a weight determined by the width of this transistor. It's hard to imagine the implementation to be simpler than this one.

The base for β-driven implementation was a fairly simple transformation of a regular analytic representation of the threshold function to a ratio form [7].

In the traditional representation of the threshold function

$$Y = \mathrm{Sign}\left(\sum_{j=0}^{n-1}\omega_j x_j - \eta\right),\quad \mathrm{Sign}(A) = \begin{cases}1 & \text{if } A \ge 0\\ 0 & \text{if } A < 0\end{cases}, \tag{1}$$

where ω_j - the weight of the j-th input, η - threshold, we select some arbitrary S-subset of variables, for which: $\sum_{i\in S}\omega_i = \eta$.

Then:

$$\sum_{j=0}^{n-1}\omega_j x_j - \eta = \sum_{k\notin S}\omega_k x_k - \left(\eta - \sum_{i\in S}\omega_i x_i\right) = \sum_{k\notin S}\omega_k x_k - \sum_{i\in S}\omega_i(1-x_i) = \sum_{k\notin S}\omega_k x_k - \sum_{i\in S}\omega_i \bar{x}_i \tag{2}$$

Hence:

$$Y = \mathrm{Sign}\left(\sum_{j=0}^{n-1}\omega_j x_j - \eta\right) = \mathrm{Rt}\left(\frac{\sum_{k\notin S}\omega_k x_k}{\sum_{i\in S}\omega_i \bar{x}_i}\right),\quad \text{where } \mathrm{Rt}(B) = \begin{cases}1 & \text{if } B \ge 1\\ 0 & \text{if } B < 1\end{cases}. \tag{3}$$

To avoid problems, related to the uncertainty of 0/0 type, it's enough to shift the threshold in the initial determination of a threshold function by some δ, where $0 < \delta < 1$, as follows:

$$Y = \mathrm{Sign}\left(\sum_{j=0}^{n-1}\omega_j x_j - \eta + \delta\right) = \mathrm{Rt}\left(\frac{\delta + \sum_{k\notin S}\omega_k x_k}{\sum_{i\in S}\omega_i \bar{x}_i}\right),\quad \text{where } \mathrm{Rt}(B) = \begin{cases}1 & \text{if } B > 1\\ 0 & \text{if } B < 1\end{cases}. \tag{4}$$

For example, function: $Y = \mathrm{Sign}(x_0 + x_1 + 2x_2 + 2x_3 + 4x_4 - 6)$ corresponds to the following logic expression: $Y = x_0x_1x_2x_3 \vee x_0x_1x_4 \vee x_2x_4 \vee x_3x_4$.

The following subsets of variables meet the condition (1): $\{x_0,x_1,x_2,x_3\}$, $\{x_0,x_1,x_4\}$, $\{x_2,x_4\}$, $\{x_3,x_4\}$. The ratio form may be constructed for each of these sets:

$$Y = \mathrm{Sign}(x_0 + x_1 + 2x_2 + 2x_3 + 4x_4 - 6) = \mathrm{Rt}\left(\frac{\delta + 4x_4}{\bar{x}_0 + \bar{x}_1 + 2\bar{x}_2 + 2\bar{x}_3}\right) =$$
$$= \mathrm{Rt}\left(\frac{\delta + 2x_2 + 2x_3}{\bar{x}_0 + \bar{x}_1 + 4\bar{x}_4}\right) = \mathrm{Rt}\left(\frac{\delta + x_0 + x_1 + 2x_3}{2\bar{x}_2 + 4\bar{x}_4}\right) = \mathrm{Rt}\left(\frac{\delta + x_0 + x_1 + 2x_2}{2\bar{x}_3 + 4\bar{x}_4}\right). \tag{5}$$

β-driven implementation following from the ratio form is based on changing the ratio of weight sums to the ratio of summarized conductivities of n- and p-chains of CMOS gate (Fig.1,a).

[1] This research was supported by the Ministry of Industry and Trade of Israel, Magneton Agency (file N 27995)

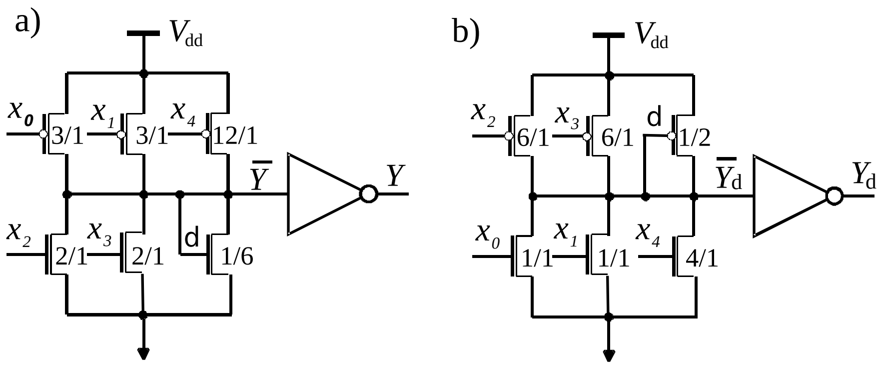

Fig. 1 CMOS implementations of the β-driven threshold element in the case of charge carrier mobility ratio 1/3 of n- and p-transistors: a) for function (5), b) for dual function.

Functioning of the circuit in Fig.1,a is illustrated by Table 1. In the table, every cell corresponding to one combination of input variable values is divided to 3 sub-cells. The upper left sub-cell contains the number of single conductivities of p-transistors, switched on, for the given values set of input variables. The upper right sub-cell contains the analogous number for n-transistors. The lower sub-cell contains the output function value (the inverter output).

Table 1. Functional table for the circuit shown in Fig.1,a.

$x_0x_1x_2x_3x_4$	000		100		010		110		001		101		011		111	
00	6	d	6	2+d	6	2+d	6	4+d	2	d	2	2+d	2	2+d	2	4+d
	0		**0**		**0**		**0**		**0**		**1**		**1**		**1**	
10	5	d	5	2+d	5	2+d	5	4+d	1	d	1	2+d	1	2+d	1	4+d
	0		**0**		**0**		**0**		**0**		**1**		**1**		**1**	
01	5	d	5	2+d	5	2+d	5	4+d	1	d	1	2+d	1	2+d	1	4+d
	0		**0**		**0**		**0**		**0**		**1**		**1**		**1**	
11	4	d	4	2+d	4	2+d	4	4+d	0	d	0	2+d	0	2+d	0	4+d
	0		**0**		**0**		**1**		**1**		**1**		**1**		**1**	

A threshold element realizes the threshold function, which is monotonic one. However, there are monotonic functions that are not threshold. Among monotonic functions of four (and more) variables there are some, which are not threshold.

We propose a method to expend functional capabilities of β-driven elements right up to the capability to implement an arbitrary monotonic function by introducing so-called functional inputs. We present CMOS implementations the β-driven threshold element with functional inputs.

2. Threshold elements with functional inputs

The circuit, shown in Fig.2 a, implements the following symmetrical monotonic function

$$Y_1 = x_0x_1 \quad x_0x_2 \quad x_0x_3 \quad x_0x_4 \quad x_1x_2 \quad x_1x_3 \quad x_1x_4 \quad x_2x_3 \quad x_2x_4 \quad x_3x_4 = \\ = \mathrm{Sign}(x_0 + x_1 + x_2 + x_3 + x_4 - 2) \qquad (6)$$

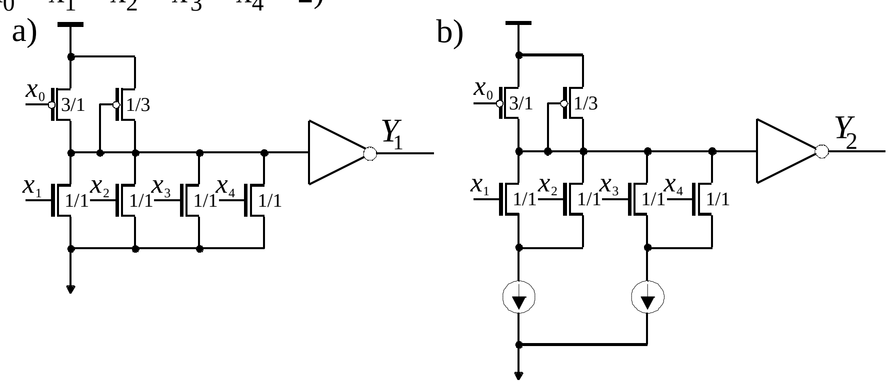

Fig. 2 Implementations of functions: a) for the function (6); b) for the function (7).

Let us split the n-channel part of the circuit to two channels and incorporate a current stabilizer, thus fixing the current equal to i_0 in the channels (Fig. 2b). The function (6) will be changed as follows:

$$Y_2 = \mathrm{Sign}(x_0 + z_1 + z_2 - 2) = x_0z_1 \quad x_0z_2 \quad z_1z_2, \qquad (7)$$

where $z_1 = x_1 \vee x_2$ and $z_2 = x_3 \vee x_4$, hence:

$$Y_2 = x_0x_1 \vee x_0x_2 \vee x_0x_3 \vee x_0x_4 \vee x_1x_3 \vee x_1x_4 \vee x_2x_3 \vee x_2x_4. \tag{8}$$

The most important thing here is that the function Y_2 (7) is monotonic but not threshold.

It can be seen from the above examples, that, if a particular subset of n-transistors controlled by $x_j \in R_k$ variables is selected, and the total current is limited by unit value i0 of a current stabilizer via this subset, the whole of this variable subset will operate in the output function as a single variable: $z_k = \bigvee_{x_j \in R_k} x_j$.

On the other hand, if a particular subset of p-transistors controlled by $x_i \in R_m$ variables, is selected, and the total current via this subset of transistors is limited by unit value i_0 of a current stabilizer, the whole of this variable subset of these variables operates in the output function as a single variable: $z_m = \overline{\bigvee_{x_i \in R_m} \bar{x}_i} = \mathop{\&}_{x_i \in R_m} x_i$.

For CMOS β-driven threshold elements we'll define as a functional input the subcircuit (Fig. 2) consisting of:

a) transistors, connected in parallel and controlled by the variables of a particular subset,
b) such a current stabilizer, connected in series, that the current via this subcircuit doesn't depend on the number of open transistors.

Let us consider a particular monotonic function F(X):

$$F(X) = \bigvee_{j=0}^{k-1} c_j = \mathop{\&}_{i=0}^{m-1} d_i, \tag{9}$$

where c_j – the set of products in the minimum sum of products form of the function *F(X)*, d_j –the set of sums in the minimum product of sums form of the function *F(X)*, c_j and d_j are realized by *p-* and *n-* functional inputs respectively.
Implementation of F(X) by a single threshold element, having functional inputs, can be expressed as follows:

$$F(X) = Rt\left(\frac{\sum_{j=0}^{k-1} \bar{c}_j + \delta}{k}\right) = Rt\left(\frac{m}{\sum_{i=0}^{m-1} d_i + \delta}\right). \tag{10}$$

The above expression gives the positive answer to the central question about a possibility to increase functional capacity of the β-driven threshold element up to the capacity of implementing any monotonic function.
Note, that the positive feature of the offered circuit is that its implementability depends on the number of products in the corresponding minimum form only, but not on the rank of the product (sum).

Conclusion

We described a CMOS implementation of a β-driven threshold element. Using such an element as a functional basis of artificial neurons has a number of advantages in comparison with the traditional functional basis. In this paper we have focused on extension of functional capability of the threshold element.
The main contribution of the paper is introducing functional inputs into the β-driven threshold element to allow increasing its functional capability up to the capability of realizing every monotonic Boolean function.
We presented an implementation of the β-driven threshold element, having the proposed functional inputs, by a CMOS circuit. The proposed solutions were simulated. The SPICE simulation results demonstrate the high efficiency of the solutions and consequently motivate further investigations of the artificial neurons based on β-driven elements.

References

[1] T. Shibata, T. Ohmi, "Neuron MOS Binary-logic Integrated Circuits: Part 1, Design Fundamentals and Soft-hardware Logic Circuit Implementation", *IEEE Trans. Electron Devices*, Vol.40, No.5, 1993, pp. 974-979.

[2] T. Ohmi, T. Shibata, K. Kotani, "Four-Terminal Device Concept for Intelligence Soft Computing on Silicon Integrated Circuits". *Proc. of IIZUKA'96*, 1996, pp. 49-59.

[3] S.M. Fakhraie, K.C. Smith, *VLSI-Compatible Implementations for Artificial Neural Networks*. Kluwer, Boston-Dordrecht-London, 1997.

[4] Montalvo, R. Gyurcsik and J. Paulos, "Toward a General-Purpose Analog VLSI Neural Network with On-Chip Learning", *IEEE Transactions on Neural Networks*, Vol.8, No.2, March 1997, pp.413-423.

[5] V. Varshavsky, "Beta-Driven Threshold Elements", *Proceedings of the 8-th Great Lakes Symposium on VLSI*, IEEE Computer Society, Feb. 19-21, 1998, pp.52-58.

[6] V. Varshavsky, "Simple CMOS Learnable Threshold Element", *International ICSC/IFAC Symposium on Neural Computation*, Vienna, Austria, Sept.23-25, 1998.

[7] V. Varshavsky, "CMOS Artificial Neuron on the Base of Beta-Driven Threshold Element", *IEEE International Conference on Systems, Man and Cybernetics*, San Diego, CA, October 11-14, 1998, pp.1857-1861.