

Ultra-Fast Noise Immune CMOS Threshold Logic Gates

Valeriu Beiu

RN2R / Rose Research LLC, 6009 Beltline Road, Suite 100 LB24, Dallas, TX 75240, USA

Abstract—This paper details a systematic method for significantly improving the noise margins of very fast threshold gates. The method is based on adding nonlinear terms determined from the Boolean form of the threshold function to be implemented. Simulation results support our theoretical claims. Finally, two methods for drastically reducing the dissipated power of such threshold gates down to <50%, and respectively <10% are also suggested.

Index Terms—VLSI, CMOS integrated circuits, threshold logic, threshold gates, noise.

I. INTRODUCTION

THE first known manuscript on algebra—used earlier in ancient Babylon, Egypt and India—entitled *Arithmetica*, was written by Diophantus of Alexandria (c. 200-280). The term derives from the Arabic *al-jabr* (literally “reunion of broken parts,” translated as “completion”), and it is a tribute to the book of Abu Ja’far Muhammad ibn Musa (c. 780-850, known as al-Khwarizmi): *al-Kitab al-mukhtasar fi hisab al-jabr wa’l-muqabala* (*The compendious book on calculation by completion and balancing*). Algebra was thus established as that branch of mathematics in which arithmetic operations and relationships are generalized by using symbols. But it was only in 1666 that—in *De Arte Combinatoria*—Leibniz developed the binary notation, while formalizing *Boolean algebra / logic* (used in present day digital circuits) was done in 1854 in Boole’s seminal work: *An Investigation of the Laws of Thought*. Still, it was not till the 1940’s that another type of logic has emerged: *threshold logic* (TL). TL is a unified theory of logic gates which includes conventional gates as a subset [1–3]. A TL gate (TLG) performs a linearly separable function, therefore being *functionally* more powerful than conventional Boolean gates (AND/OR, NAND/NOR). TLGs have been associated with artificial intelligence—therefore closer to Boole’s original aim of *investigating the laws of thought*—as being the simplest form of artificial neurons: the perceptrons.

Formally, a TL circuit (TLC) is an acyclic graph having several input nodes, and some (at least one) output nodes. Each node (TLG) computes the *weighted sum* of its inputs x_i :

$$F(x) = F(x_1, \dots, x_\Delta) = \text{sign}(\sum_{i=1}^{\Delta} w_i x_i + \theta), \quad (1)$$

with $w_i \in \mathbb{R}$ the synaptic *weights*, $\theta \in \mathbb{R}$ known as the *threshold*, Δ being the *fan-in*, and *sign* the sign function. Because the graph is acyclic (*i.e.*, it has no feedback), TLCs can be layered. The cost functions which characterize a TLC are: (i) *depth* (*i.e.*, number of edges on the longest input-to-output path, or number of layers); and (ii) *size* (*i.e.*, number of TLGs).

When compared to Boolean logic (BL), TL has significant theoretical advantages, namely it has been proven that many complex Boolean functions (BFs) can be realized using fewer TLGs (*i.e.*, smaller *size*) and shallower / faster circuits (*i.e.*, smaller *depth*). *The more complex the BF to be implemented is, the greater the advantage of TL over BL is.*

In spite of sound mathematical results [4–12], TL has had almost no practical influence, due to the difficulties of implementing them in silicon: (i) high precision *weights* cannot easily be stored as variables which can summed [13]; (ii) the analog behaviour of the internal (summing) node leads to poor noise margins [14–20].

Three alternatives for easy summing are possible: current, charge and conductance [13]. The current summing was / is used by analog implementations. The charge summing was used by *switched-capacitor neural networks* [21, 22], *capacitive TLGs* [23–27], and the *Neuro MOS transistor* (vMOS) [28–32]. These two approaches are slower than conductance-based implementations. The first conductance-based implementations were made in the mid-1950’s using resistive circuits, and were followed later by bipolar TLGs [33–35]. If a differential amplifier is used for implementing the sign (see Fig. 1), only limited speeds can be obtained. An interesting alternative was to wire together the outputs of several inverters [36–39]. Each x_i input drives an inverter. These inverters satisfy a ratio design, with the weights w_i encoded in the relative

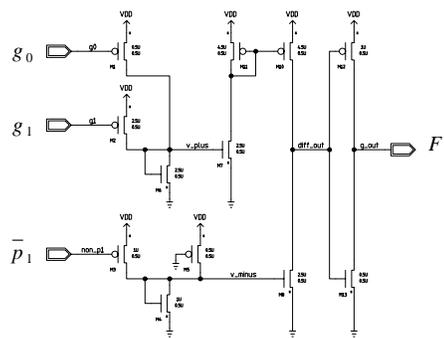


Fig. 1. A differential amplifier can be used to implement the sign function, but is a relatively slow solution.

Manuscript received November 16, 1999; revised January 24, 2000. This work was supported by Rose Research LLC, Dallas, Texas, USA.

V. Beiu (e-mail: vbeiu@rose-research.com) is on leave of absence from the “Politehnica” University of Bucharest, Computer Science Department, Spl. Independenței 313, RO-77206 Bucharest, Romania.

Readers should note that RN2R / Rose Research LLC has already filed several patent applications covering this work to the US Patent and Trademark Office.

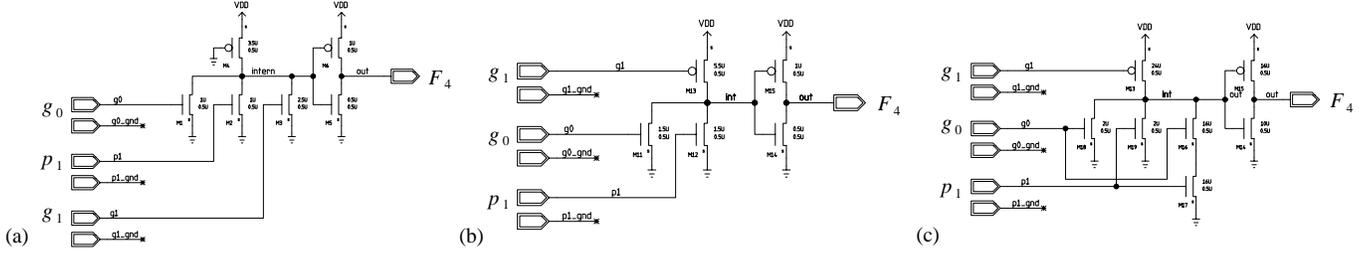


Fig. 2. Different alternatives for implementing $F_4 = \text{sign}(2g_1 + p_1 + g_0 - 1.5)$: (a) pseudo-nMOS design style; (b) conductance-based TLG; (c) conductance-based TLG with the additional noise suppression logic tree.

sizes/strengths of these inverters. The resulting TLGs are very fast, but they have poor noise margins, and are power hungry. The noise problem makes it very difficult—if not impossible—to implement large *fan-in* TLGs. These *output-wired-inverters scheme* was later rediscovered as the *ganged-CMOS-gate* [40–44], while a first enhancement [45] showed how to connect the inputs only to the nMOS transistors (pseudo-nMOS design style). This idea has been furthered by using only one transistor per input [46–50], with the same two disadvantages: poor noise margins and high power consumption. Recently [51, 52], it has been suggested to improve on the noise margins of such TLGs. The solution requires three additional reference voltages (beside V_{DD} and GND), while still being power hungry. Another solution is detailed in [53, 54].

This paper will present a different method for enhancing the noise margins of conductance-based TLGs, by adding nonlinear terms to the TL function to be implemented. These nonlinear terms can always be determined from the Boolean form of the TL function. Finally, we will suggest two methods for significantly reducing the power consumed by such TLGs.

II. NOISE IMMUNE THRESHOLD GATES

All the simulations have been performed in $0.5\mu\text{m}$ CMOS at either 3.3V or 1.5V. We start with a simple example, which appears when computing group carry-generate from the carry-generate and carry-propagate signals in a carry look-ahead type of adder (see [55, 56] for details):

$$F_4 = g_i \vee (p_i \wedge g_{i-1}) = \text{sign}(2g_i + p_i + g_{i-1} - 1.5). \quad (2)$$

Fig. 1 shows an implementation using a differential amplifier having a delay of 250ps (on 20fF). Three faster implemen-

tations use an inverter to implement the sign. A pseudo-nMOS style TLG can be seen in Fig. 2(a). It achieves 65ps on the same 20fF load. The voltage gap (*i.e.*, the difference between “the lowest 1” and “the highest 0”) on the internal node is 1.4V (see *int* in Fig. 3(a)). This implementation dissipates DC-power in seven out of all the eight possible input combinations.

Because g_i and p_i represent the “generate” and respectively “propagate” signals from a classical adder, they are not independent signals. In particular, if $g_i = a_i \wedge b_i$ and $p_i = a_i \vee b_i$, we can use g_i to drive the pMOS, and the internal node will never float. Fig. 2(b) shows this conductance-based TLG, which has a delay of 65ps (on the same 20fF load), and a voltage gap of 1.37V. It dissipates less power, as only three input combinations (out of all the eight possible input combinations) form a current path from V_{DD} to GND.

Finally, a noise immune version of the TLG is detailed in Fig. 2(c). It has the same structure like the TLG from Fig. 2(b) with an additional path between the internal node and GND. The BF $f_{NSL} = p_i \wedge g_{i-1}$ represents the noise suppression logic (NSL) stack. This makes the functioning nonlinear, namely:

$$F_4 = \text{sign}[(2+\alpha)g_i + p_i + g_{i-1} + \alpha f_{NSL} - (3 + \alpha)/2]. \quad (3)$$

The nonlinearity introduced by f_{NSL} increases the voltage gap from 1.37V to 2.2V (see *int* in Fig. 3(b)). With a proper resizing of the transistors, the delay is 70ps (on 80fF). For comparison, Fig. 3(c) plots the voltages on the internal and output nodes for the TLG with and without the NSL.

For testing the noise (and power) rejection ratio of these designs, additional simulations have been performed. The supply voltage V_{DD} was lowered to 1.5V, with a 0.4V swing (*i.e.*, $V_{DD} = 1.1 \dots 1.5$ V). Random spikes of 0.4V amplitude were also

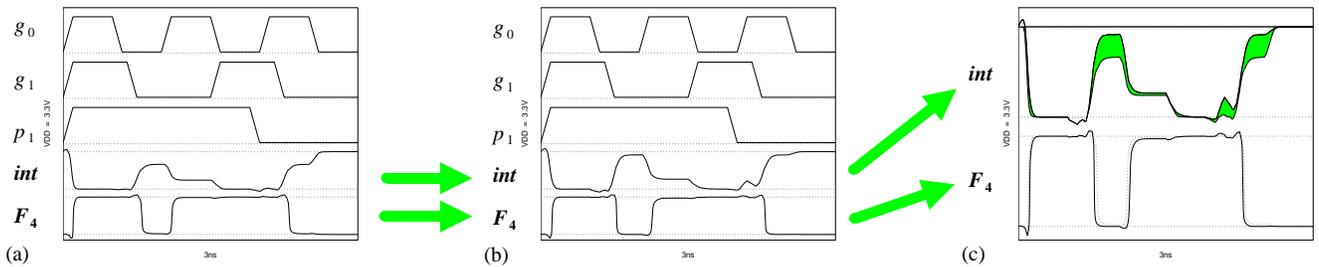


Fig. 3. Simulation results for: (a) the TLG without NSL (Fig. 2(b)); (b) the TLG with NSL (Fig. 2(c)); (c) detail showing the internal and the output nodes of the TLG with NSL (continuous line) and without NSL (dotted line).

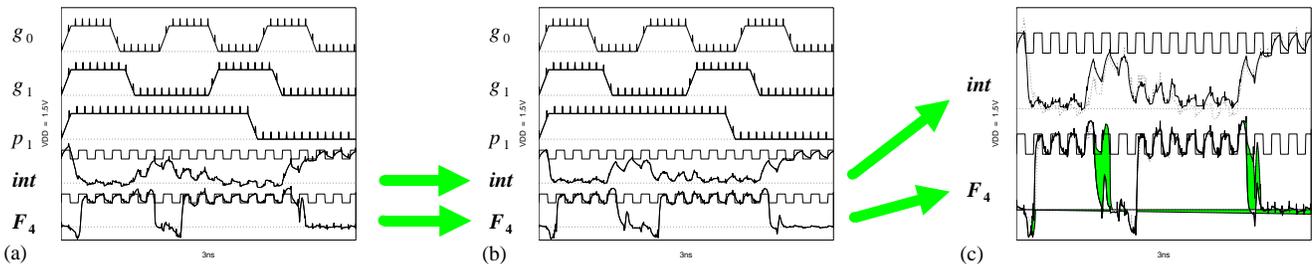


Fig. 4. Simulation for $V_{DD}=1.1\dots1.5V$: (a) the TLG without NSL; (b) the TLG with NSL; (c) detail showing the internal and the output nodes for the TG with NSL (continuous line) and without NSL (dotted line).

added to all the inputs. No other modifications have been made. The results can be seen in Fig. 4. The TLG without the NSL produces two errors (see Fig. 4(a)). The noise enhanced TLG works correctly (see Fig 4(b)). For clarity, both the internal and the output results are overlapped in Fig. 4(c).

In general, any linearly separable function can be represented in ratio form [51, 52]:

$$F(x) = \text{sign} \left[\frac{\sum_{i \in S} w_i x_i}{\sum_{i \in S} w_i \bar{x}_i} - 1 \right]. \quad (4)$$

where $\sum_{i \in S} w_i = \theta$ (S being a subset of indexes). The solution we propose for enhancing the noise margins is to add nonlinear terms. These terms form a reduced BF (f_{NSL}) which can easily be determined by subtracting from the original function F the minterm implemented by the pMOS stack:

$$f_{NSL} = F \setminus (\wedge_{i \in S} x_i), \quad (5)$$

while a mirrored solution is obtained by subtracting the minterm implemented by the nMOS stack. This reduced BF is used to implement the NSL stack, and also for resizing the transistors (see (3) where the sizing is modified with respect to α).

All the solutions presented are more or less power hungry. That is why—for eliminating the second major disadvantage of TG—several solutions have been suggested [57–59]. A proprietary method using a self-timed power-down mechanism was developed and can be seen in Fig. 5. It uses two additional transistors to isolate the gate from V_{DD} and GND. Each of these transistors is driven by a very simple control logic which has

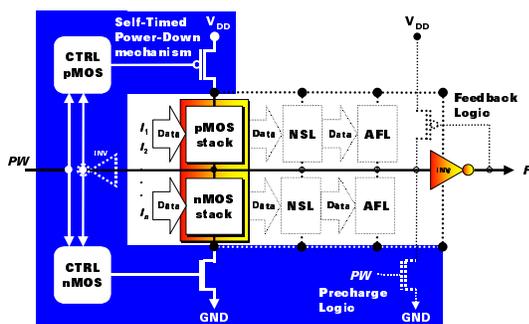


Fig. 5. The self-timed power-down mechanisms uses two additional transistors to isolate the gate from V_{DD} and GND.

as inputs a combination of the incoming data, the output (of the gate) and an asynchronous external signal pw . For keeping the power consumption low, pass transistors are used to implement these control logic blocks. Additional features are: the fact that the internal node can be precharged, and the use of a keeper.

A first solution uses only the additional pMOS transistor (which separates the gate from V_{DD}). The pMOS control logic for the function F_4 implements $g_i \wedge \overline{pw}$, while the internal node is reset by an nMOS transistor connected to GND. This reduces the power consumption to less than 50%. A second solution reduces the DC power to about 10%, making the gate useful even for low-power applications. It uses both the pMOS and the nMOS additional transistors. The pMOS control logic implements $\overline{pw} \vee (p \overline{w} \wedge F)$, while the nMOS control logic implements $(pw \vee pw^c) \wedge (pw \vee F)$.

III. CONCLUSIONS

The paper has detailed solutions for enhancing conductance-based TLGs. They require: (i) the addition of an NSL stack (easily determined from the Boolean form of the TL function to be implemented); (ii) the addition of a self-timed power-down mechanism; (iii) resizing the transistors. These can drastically increase the noise immunity, and reduce the dissipated power, while conserving the high-speed characteristics of conductance-based TLGs.

REFERENCES

- [1] R.C. Minnik, "Linear-input logic," *IRE Trans. Electr. Comp.*, vol. 10, pp. 6–16, 1961.
- [2] S. Hu, *Threshold Logic*, Berkeley, CA: Univ. of California Press, 1965.
- [3] S. Muroga, *Threshold Logic and Its Applications*, New York: Wiley, 1971.
- [4] E.I. Neciporuk, "The synthesis of networks from threshold elements," *Soviet Mathematics—Doklady*, vol. 5, pp. 163–166, 1964 [English transl., *Automation Express*, vol. 7, pp. 27–32 and pp. 35–39, 1964].
- [5] N.P. Red'kin, "Synthesis of threshold circuits for certain classes of Boolean functions," *Kibernetika*, vol. 5, pp. 6–9, 1970 [English transl., *Cybernetics*, vol. 6, pp. 540–544, 1973].
- [6] O.B. Lupanov, "The synthesis of circuits from threshold elements," *Problemy Kibernetiki*, vol. 20, pp. 109–140, 1973.
- [7] K.-Y. Siu, V.P. Roychowdhury, and T. Kailath, "Depth-size tradeoffs for neural computations," *IEEE Trans. Comp.*, vol. 40, pp. 1402–1412, 1991.

- [8] I. Parberry, *Circuit Complexity and Neural Networks*, Cambridge: MIT Press, 1994.
- [9] V. Beiu, "Digital integrated circuit implementations," in *Handbook of Neural Computation*, E. Fiesler, and R. Beale, Eds., New York: Inst. of Physics (IoP), 1996, Chapter E1.4.
- [10] V. Beiu, "Optimal VLSI implementation of neural networks," in *Neural Networks and Their Applications*, J.G. Taylor, Ed., Chichester, UK: John Wiley, 1996, pp. 255–276.
- [11] V. Beiu, "On the circuit and VLSI complexity of threshold gate COMPARISON," *Neurocomputing*, vol. 19, pp. 77–98, 1998.
- [12] V. Beiu, and H.E. Makaruk, "Deeper sparser nets can be optimal," *Neural Proc. Lett.*, vol. 8, pp. 201–210, 1998.
- [13] A.H. Kramer, "Array-based analog computation; principles, advantages and limitations," *Proc. Microelectronics for Neural Networks* (Feb. 12–14, Lausanne, Switzerland), pp. 68–79, 1996.
- [14] J. Myhill, and W.H. Kautz, "On the size of weights required for linear-input switching functions," *IRE Trans. Electr. Comp.*, vol. 10, pp. 288–290, 1961.
- [15] J.S. Denker, and B.S. Wittner, "Network generality, training required, and precision required," in *Neural Information Processing Systems*, D.Z. Anderson, Ed., New York: AIPress, pp. 219–222, 1988.
- [16] D. Hammerstrom, "The connectivity analysis of simple association—or how many connections do you need," in *Neural Information Processing Systems*, D.Z. Anderson, Ed., New York: AIPress, pp. 338–347, 1988.
- [17] J.L. Holt, and J.-N. Hwang, "Finite precision error analysis of neural network hardware implementations," *IEEE Trans. Comp.*, vol. 42, pp. 281–290, 1993.
- [18] A.H. Khan, and E.L. Hines, "Integer-weight neural networks," *Electr. Lett.*, vol. 30, pp. 1237–1238, 1994.
- [19] J. Wray, and G.G.R. Green, "Neural networks, approximation theory, and finite precision computation," *Neural Networks*, vol. 8, pp. 31–37, 1995.
- [20] M. Stevenson, and S. Huq, "On the capability of threshold adalines with limited-precision weights," *Neural Computation*, vol. 8, pp. 1603–1610, 1996.
- [21] Y.P. Tsividis, and D. Anastassiou, "Switched-capacitor neural networks," *Electr. Lett.*, vol. 23, pp. 958–959, 1987.
- [22] Y.P. Tsividis, "Switched neural networks," *US 04873661*, Oct. 10, 1989.
- [23] U. Çilingiroglu, "Capacitive synapses for microelectronic neural networks," *Proc. Int. Symp. Circ. & Sys.* (May 1–3, New Orleans, LA, USA), pp. 2982–2985, 1990.
- [24] U. Çilingiroglu, "A purely capacitive synaptic matrix for fixed-weight neural networks," *IEEE Trans. Circ. & Sys.*, vol. 38, pp. 210–217, 1991.
- [25] H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, and U. Çilingiroglu, "A capacitive threshold-logic gate," *IEEE J. Solid-State Circ.*, vol. 31, pp. 1141–1150, 1996.
- [26] A. Stokman, *Implementation of Threshold Logic*, MSc thesis TR 1-68340-28(1998)01, Delft Univ. of Technology [<http://dutepp0.et.tudelft.nl/~sorin/ms/sander/Main.ps.gz>], Jan. 23, 1998.
- [27] A. Stokman, S. Coşofana, and S. Vassiliadis, "A versatile threshold logic gate," *Proc. Int. Semicond. Conf.* (Oct. 6–9, Sinaia, România), pp. 163–166, 1998.
- [28] T. Shibata, and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE J. Solid-State Circ.*, vol. 39, pp. 1444–1455, 1992.
- [29] T. Shibata, and T. Ohmi, "Neuron MOS binary-logic integrated circuits—Part I: Design fundamentals and soft-hardware-logic circuit implementation," *IEEE Trans. Electr. Dev.*, vol. 40, pp. 570–576, 1993.
- [30] T. Shibata, and T. Ohmi, "Neuron MOS binary-logic integrated circuits—Part II: Simplifying techniques of circuit configuration and their practical applications," *IEEE Trans. Electr. Dev.*, vol. 40, pp. 974–979, 1993.
- [31] V. Bohossian, P. Hasler, and J. Bruck, "Programmable neural logic," *Proc. Int. Conf. Innovative Systems in Silicon* (Oct. 9–10, Austin, Texas, USA), pp. 13–21, 1997.
- [32] V. Bohossian, *Neural Logic: Theory and Implementation*, PhD thesis, California Institute of Technology [http://paradise.caltech.edu/papers/vincent_thesis.ps], July 17, 1998.
- [33] J.D. Heightley, "Threshold logic gate," *US 03597626*, Aug. 3, 1971.
- [34] H.M. Martin, "Threshold logic for integrated full adder and the like," *US 03609329*, Sep. 28, 1971.
- [35] R.P. Foester, "Threshold and majority gate elements and logical arrangements thereof," *US 03644923*, Feb. 22, 1972.
- [36] J.B. Lerch, "Threshold gate circuits employing field-effect transistors," *US 03715603*, Feb. 6, 1973.
- [37] J.S. Britton, "CMOS comparator," *US 04031511*, Jun. 21, 1977.
- [38] R. Zuleeg, and J.K. Notthoff, "Symmetrical input NOR/NAND gate circuit," *US 04038563*, Jul. 26, 1977.
- [39] M.G. Johnson, "A symmetric CMOS NOR gate for high-speed applications," *IEEE J. Solid-State Circ.*, vol. 23, pp. 1233–1236, 1988.
- [40] K.J. Schultz, and K.C. Smith, "A CMOS binary adder using quaternary ganged-logic internal node," *Proc. Intl. Symp. Multiple-Valued Logic* (May 29–31, Guangzhou, PRC), pp. 356–359, 1989.
- [41] K.J. Schultz, R.J. Francis, and K.C. Smith, "Ganged CMOS: Trading standby power for speed," *IEEE J. Solid-State Circ.*, vol. 25, pp. 870–873, 1990.
- [42] H. Chung, and S. Kim, "MOS multi-layer neural network and its design method," *US 05293458*, Mar. 8, 1994.
- [43] Ch.L. Lee, and Ch.-W. Jeh, "CMOS threshold gates and networks for order statistics filtering," *IEEE Trans. Circ. & Sys.*, vol. 41, pp. 453–456, 1994.
- [44] N.-E. Belabbès, A. Guterman, Y. Savaria, and M.R. Dagenais, "Ratioed voter circuit for testing and fault-tolerance in VLSI processing arrays," *IEEE Trans. Circ. & Sys.*, vol. 43, pp. 143–152, 1996 ["Ratioed voter circuit for testing and fault-tolerance," *Proc. Intl. Symp. Circ. & Sys.* (May 10–13, San Diego, CA, USA), pp. 1125–1128, 1992].
- [45] S. Goodwin-Johansson, "Circuit to perform variable threshold logic," *US 04896059*, Jan. 23, 1990.
- [46] H. Jeong, "Neural network implementation of a binary adder," *US 05016211*, May 14, 1991.
- [47] H. Jeong, "Adaptive associative memory comprising synapses of CMOS transistors," *US 5034918*, Jul. 23, 1991.
- [48] H. Chung, and S. Pack, "Floating point adder circuit using neural network," *US 05086405*, Feb. 4, 1992.
- [49] H. Jeong, "Digital multiplier employing CMOS transistors," *US 05095457*, Mar. 10, 1992.
- [50] H. Chung, and K. Lee, "Programmable multilayer neural network," *US 05448682*, Sep. 5, 1995.
- [51] V. Varshavsky, "Beta-Driven Threshold Elements," in *Proc. Great Lakes Symp. on VLSI'98* (Feb. 19–21, Lafayette, Louisiana), pp. 52–58, 1998.
- [52] V. Varshavsky, and V. Marakhovskiy, "Beta-CMOS Artificial Neuron and Implementability Limits," in *Engineering Applications of Bio-Inspired Artificial Neural Networks*, Vol. II, J. Mira, and J.V. Sánchez-Andrés, Eds., Berlin: Springer, 1999, pp. 117–128.
- [53] R. Thewes, S. Prange, E. Wohlrab, and W. Weber, "Circuit for comparing two electrical quantities provided by a first neuron MOS field effect transistor and a reference source," *US 05990709*, Nov. 23, 1999.
- [54] S. Prange, R. Thewes, E. Wohlrab, and W. Weber, "Circuit arrangement for realizing logic elements that can be represented by threshold value equations," *US 05991789*, Nov. 23, 1999.
- [55] V. Beiu, and J.G. Taylor, "On the circuit complexity of sigmoid feed-forward neural networks," *Neural Networks*, vol. 9, pp. 1155–1171, 1996.
- [56] V. Beiu, "Neural addition and Fibonacci numbers," in *Engineering Applications of Bio-Inspired Artificial Neural Networks*, Vol. II, J. Mira, and J.V. Sánchez-Andrés, Eds., Berlin: Springer, 1999, pp. 198–207.
- [57] P.D. Kartschoke, and N.J. Rohrer, "Techniques for reduced power and increased speed in dynamic- and ratio-logic circuits," *Proc. Midwest Symp. Circ. & Sys.* (August 18–21, Ames, Iowa, USA), 1996.
- [58] P.D. Kartschoke, and N.J. Rohrer, "Low power pre-discharged ratio logic," *US 05572150*, Nov. 5, 1996.
- [59] P.D. Kartschoke, N.J. Rohrer, and T. Sulzbach, "Self-timed low power ratio-logic system having an input sensing circuit," *US 05867038*, Feb. 2, 1999.