Real-Time Reconfigurable Linear Threshold Elements and Some Applications to Neural Hardware

Snorre Aunet and Morten Hartmann

Department of Computer and Information Science
The Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
{snorre.aunet,mortehar}@idi.ntnu.no
http://caos.idi.ntnu.no/

Abstract. This paper discusses some aspects regarding the use of universal linear threshold elements implemented in a standard double-poly CMOS technology, which might be used for neural networks as well as plain, or mixed-signal, analog and digital circuits. The 2-transistor elements can have their threshold adjusted in real time, and thus the basic Boolean function, by changing the voltage on one or more of the inputs. The proposed elements allow for significant reduction in transistor count and number of interconnections. This in combination with a power supply voltage in the range of less than 100 mV up to typically 1.0 V allow for Power-Delay-product improvements typically in the range of hundreds to thousands of times compared to standard implementations in a 0.6 micron CMOS technology. This makes the circuits more similar to biological neurons than most existing CMOS implementations. Circuit examples are explored by theory, SPICE simulations and chip measurements. A way of exploiting inherit fault tolerance is briefly mentioned. Potential improvements on operational speed and chip area of linear threshold elements used for perceptual tasks are shown.

1 Introduction

Floating-gate CMOS circuits have found their use outside the digital memory use during the last decade, and an introduction to the field can be found in [1]. Floating-gate devices are used as analog memory elements, as part of capacitive-based circuits and as adaptive circuit elements [1]. Non-traditional floating-gate circuitry is also starting to find it's way into commercial use [2]. The circuits presented here are offspring from [3], which might be perceived as having their startingpoints from two basic ideas, namely the multiple-input floating-gate transistor concept [4] and UV-programmable floating-gate circuits [5], using the UV-programming method from [6].

By exploiting the inherent analog amplifying characteristics of the transistors for something more than the traditional switching function similar to the method in [7] one can reduce the number of active elements and interconnect dramatically

A.M. Tyrrell, P.C. Haddow, and J. Torresen (Eds.): ICES 2003, LNCS 2606, pp. 365–376, 2003.

[8]. Combined with operation of the transistors in weak inversion, where the currents are typically down in the sub pA to uA range [8], from experience with a standard 0.6 um CMOS process [9], there is a significant ultra low-power potential whether the building blocks are potentially used for analog, digital or mixed-signal circuits or neural network implementations.

Section 2 introduces one of several [8] real-time reconfigurable linear threshold elements and suggests ways it can be used for implementing a variety of basic Boolean functions. The ultra low power consumption potential and improved manufacturability are then briefly treated. In section 3 strategies possibly increasing fault tolerance is given.

Section 4 gives some pointers towards the potential chip area reduction, and thus production costs reduction, that might be obtained by using linear threshold elements. For some perceptual data processing the speedup of three orders of magnitude compared to a traditional computer implementation is given as an example.

2 CMOS Floating-Gate Circuits in Weak Inversion

2.1 Balancing the Circuit; Setting Switching Point and Current Levels

Our circuit elements need to have their switching point adjusted by an initial UV-programming [6], which is a post-fabrication technique that can be applied to any standard double-poly CMOS technology. The setting of the switching point ensures that the output voltage for the circuit element is at $V_{dd}/2$ for an input at exactly the same voltage level. This is done to achieve symmetric noise margins for digital zeros and ones and the best possible digital operation.

During UV-programming the chip is radiated by UV-C from a lamp. Concurrently the power supply rails and substrates get certain voltages applied that regulate charge transport to the floating gates of the device. Once the UV-exposure is ended, the charge levels on the floating-gates are defined practically indefinitely. Charge transport is done through the UV-activated conductances, which are indicated by extra circles between gate and sources in the transistor symbols in Figure 1.

One and the same UV-programmable elements might be reprogrammed for different current levels, due to for example different needs for operational speed of the basic circuit building blocks [8]. Switching current levels, I_{beq}, might be programmed to 2-3 orders of magnitude [8] due to experience with the process from [9].

2.2 Simple Analysis of a Basic Linear Threshold Element

The following equations can be used to describe the weak inversion currents of the PMOS and NMOS transistors [10]:

$$I_{ds,p} = I_{beq} \prod_{i=1}^{m} exp\left\{ \frac{1}{nU_t}(V_{dd}/2 - V_i)k_i \right\} \quad (1)$$

$$I_{ds,n} = I_{beq} \prod_{i=1}^{m} exp\left\{ \frac{1}{nU_t}(V_i - V_{dd}/2)k_i \right\} \quad (2)$$

Here, $k_i = C_i/C_{tot}$ is the capacitive division factor of the ith input capacitor, C_i, and C_{tot} is the total capacitance seen from the floating-gate. I_{beq} is the balanced equilibrium current, which is the drain current of the transistors when all ordinary input signals and driven nodes are equal to $V_{dd}/2$. I_{beq} is strongly dependent on the V_{dd} level. Here an equal number of capacitively weighted inputs to both the PMOS and NMOS transistor is assumed, and that the intrinsic slope factors in weak inversion, n, are equal for both devices. U_t is the thermal voltage, which is 25.8 mV at room temperature.

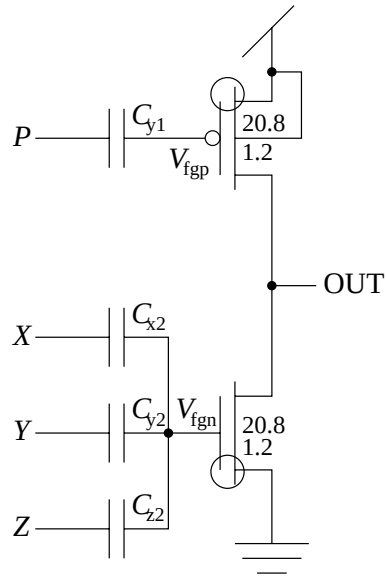

Fig. 1. A CARRY'-,NAND-,NOR and INVERT-circuit [8]. The numbers 20.8 and 1.2 refer to the width and length of the transistors in micrometers, respectively.

To illustrate some traits of the reconfigurable floating-gate circuits, the "P1N3" circuit in Figure 1 is used here. The capacitances between X,Y and Z are all designed for equal size. Also the sum of capacitances coupled to the NMOS (Figure 1) equals the capacitances connected to the PMOS. ($C_{x2} + C_{y2} + C_{z2} = C_{y1}$). The numbers of capacitively weighted inputs have been used to name the circuits, counting ordinary inputs and other inputs used for control of behavior to each two-MOSFET element. The circuit in Figure 1 gets the name P1N3 due

to this naming system. The most used inverter may be called P1N1. The circuit has previously been presented as a stand-alone circuit or building block [8]. Using the above equations yields:

$$I_{ds,p} = I_{beq} exp\left\{\frac{1}{nU_t}\left(\frac{V_{dd}}{2} - V_p\right)\right\} \tag{3}$$

$$I_{ds,n} = I_{beq} exp\left\{\frac{1}{nU_t}\left(\frac{1}{3}V_x + \frac{1}{3}V_y + \frac{1}{3}V_z - \frac{V_{dd}}{2}\right)\right\} \tag{4}$$

V_p, for example, means the voltage on the capacitively weighted input to the PMOS transistor in Figure 1. If the voltages V_p, V_x, V_y or V_z are all equal to $V_{dd}/2$ the exponentials in the above equations equal 0, and we have the equilibrium condition with $I_{ds,p}$=$I_{ds,n}$=I_{beq}. To illustrate how the circuit functions logically, a truth table is used here. As parts of the truth table, e_p and e_n are used; these are the parts of the exponentials directly dependent on input signals and the supply voltage, V_{dd}. For this particular circuit that means

$$e_p = \left(\frac{V_{dd}}{2} - V_p\right) \tag{5}$$

$$e_n = \left(\frac{1}{3}V_x + \frac{1}{3}V_y + \frac{1}{3}V_z - \frac{V_{dd}}{2}\right). \tag{6}$$

When parasitic capacitances are not accounted for, each of the inputs at nodes X,Y and Z are weighted by 1/3, a more optimistic estimate than would have resulted from including parasitics [8]. When $V_p = V_{dd}/2$ and X, Y and Z are allowed to have the binary values according to the table in Figure 2, Figure 3 results. When $e_p > e_n$ the output approaches the V_{dd} level, since the PMOS is then "strongest". In the opposite case it goes low.

V_{dd}	1	HIGH
V_{ss}	0	LOW

Fig. 2. Meanings of the 3 columns in each row are directly related.

By inspecting the truth table (Figure 3) it is clear that the value of the output depends on the number of 1's and 0's on the inputs only. Inspecting the truth table (Figure 3), just counting 1's and 0's in the input vector, makes it possible to get enough information out from a table with a simpler form, as in Figure 4.

Used this way (Figure 3, Figure 4) the circuit computes the inverted carry for a FULL-ADDER:

$$OUT = CARRY' = (XY + XZ + YZ)' \tag{7}$$

From the truth table, in Figure 3, one can see that by letting any one input be "0", the output is "0" if, and only if, both other inputs are "1". Then the circuit

X	Y	Z	e_p	e_n	OUT
0	0	0	0	-$3V_{dd}/6$	1
0	0	1	0	-$V_{dd}/6$	1
0	1	0	0	-$V_{dd}/6$	1
0	1	1	0	$V_{dd}/6$	0
1	0	0	0	-$V_{dd}/6$	1
1	0	1	0	$V_{dd}/6$	0
1	1	0	0	$V_{dd}/6$	0
1	1	1	0	$3V_{dd}/6$	0

Fig. 3. The table shows parts of the exponentials, e_p, e_n, and output values, for all possible binary values of inputs X,Y,Z when V_p=$V_{dd}/2$. "OUT" provides the CARRY' function for a FULL-ADDER.

P	number of 1's	e_p	e_n	OUT
$V_{dd}/2$	0	0	-$3V_{dd}/6$	1
$V_{dd}/2$	1	0	-$V_{dd}/6$	1
$V_{dd}/2$	2	0	$V_{dd}/6$	0
$V_{dd}/2$	3	0	$3V_{dd}/6$	0

Fig. 4. The table shows parts of the exponentials, e_p, e_n, and output values, for $V_P = V_{dd}/2$ and different numbers of "1's" on ordinary inputs X,Y,Z. Inputs can be either "0" or "1".

implements the 2-input NAND function. If any one input is 1, the output is "1" if and only if both other inputs are "0". Then the circuit works as a 2-input NOR gate. Connecting one input to Vdd or Vss and short-circuiting the other two gives an INVERTER. A 2-input inverting-structure like NAND or NOR is essentially the only function needed to implement any digital function.

If the e_p value is changed, the "threshold" for when, and if, $e_n > e_p$ changes. From the above equations it is easy to see that setting $e_p = -2V_{dd}/6$ makes it necessary to have only 1 binary input at "1" (V_{dd}) to make $e_n = -V_{dd}/6$, which is greater than e_p, and should give a low output. Changing the e_p value to $2V_{dd}/6$ makes it necessary to have all inputs X,Y,Z high, when restricted to Boolean inputs, in order to produce a low output. When perceiving the digital functionality of the linear threshold element only, it is possible to make a table like in Figure 5.

e_p	digital functionality
$-2V_{dd}/6$	NOR3, NOR2, INVERT
0	CARRY', NOR2, NAND2, INVERT
$2V_{dd}/6$	NAND3, NAND2, INVERT

Fig. 5. The table shows part of the exponential, e_p, and the Boolean functions resulting.

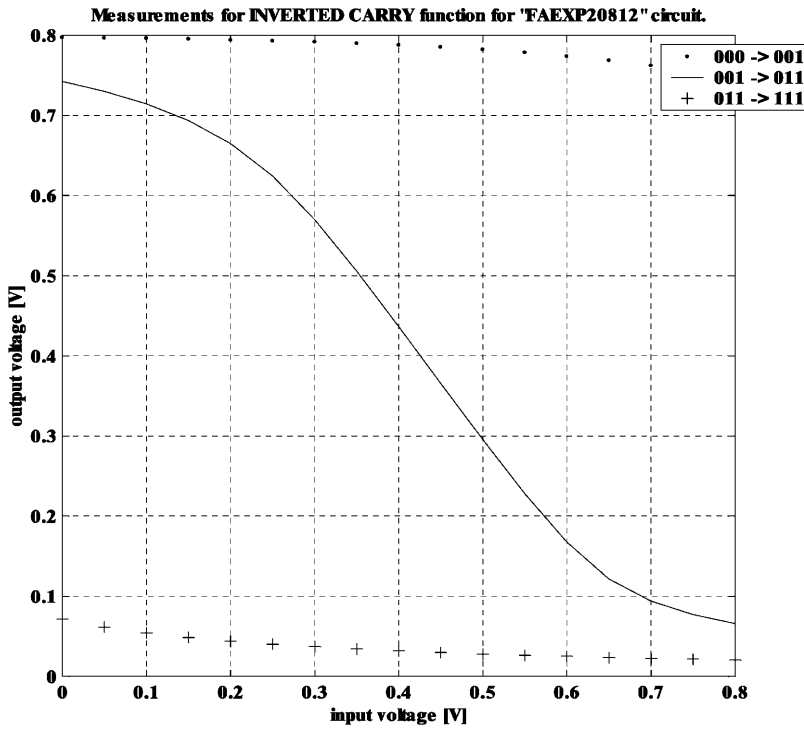

Fig. 6. Actual measurements for the P1N3 circuit, from a prototype chip, when it is producing the CARRY' of a FULL-ADDER [8].

Changing the threshold by adjusting the value of the input capacitively coupled to the PMOS can provide a real-time reconfigurable digital circuit.

2.3 Measured Functionality of P1N3 Linear Threshold Element

The inherent functionality for $e_p = 0$, demonstrated by measurements, can be seen in Figure 6. The voltage on the output goes low if, and only if, 2 or 3 of the inputs X,Y,Z goes high at the same time. Otherwise the output stays high, which in this case is $V_{dd} = 0.8$ V. The three curves (Figure 6) correpond to transitions between the four rows in Figure 4. The continous line shows the output voltage when the number of high inputs moves from 1 to 2, which is the transition between 1 and 0 on the output, for the CARRY' function of binary addition.

The functionality and performance of such circuits will depend on the implementation technology of choice. The number of inputs, their capacitive weights, if an input is coupled both to the PMOS and NMOS are among other factors that can be varied [8]. To increase the voltage gain, the size of the drawn capacitances between inputs and floating-gates should be increased [8].

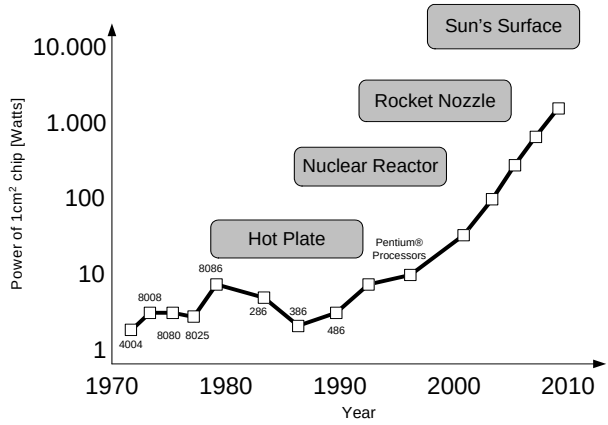

Fig. 7. If nothing is done about power consumption, power will get out of hand and Moore's law cannot continue [13].

2.4 Ultra Low Power Consumption

Dynamic power consumption depends linearly on the physical capacitance being switched [11]. Therefore using less interconnect and fewer active elements for a given function may be attractive for minimizing power consumption. Voltage reduction offers the most direct and dramatic means om minimizing energy consumption [11]. Chip measurements have demonstrated floating-gate circuitry working for supply voltages down to 93 mV [12], while typical supply voltage levels have been in the 300 - 800 mV range [8]. SPICE simulations using the "P5N5" [8] linear threshold elements as building blocks in an 8-transistor FULL-ADDER operating at a V_{dd} of 200 mV compared to a standard cell implementation in the AMS 0.6 CMOS technology [9] showed that the floating-gate implementation used 2500 times less energy, a Power-delay-Product (PDP) of 2.3 fJ while running at a clock frequency of 1 MHz [8]. Power consumption is an increasingly important issue, which is illustrated in Figure 7 based on a slide from the presentation of [13]. The PDP numbers could maybe not be undercut by any known CMOS technology of the same generation.

2.5 Matching and Manufacturability

This technology has some very attractive features, but an important problem that remains is that the UV-programmable circuits have not been demonstrated for circuits containg more than between 10 and 30 transistors transistors yet, at least from the open literature. Hopefully this number can be increased significantly, due to a proposed design method in [8], using as few different building blocks as possible. Since "The Achilles Heel of analog is that every transistor is different" [14] matching [15] of both transistors and passive elements should become the best possible if a larger system could be built from identical building blocks instead of dedicated circuitry for each basic function. Good matching

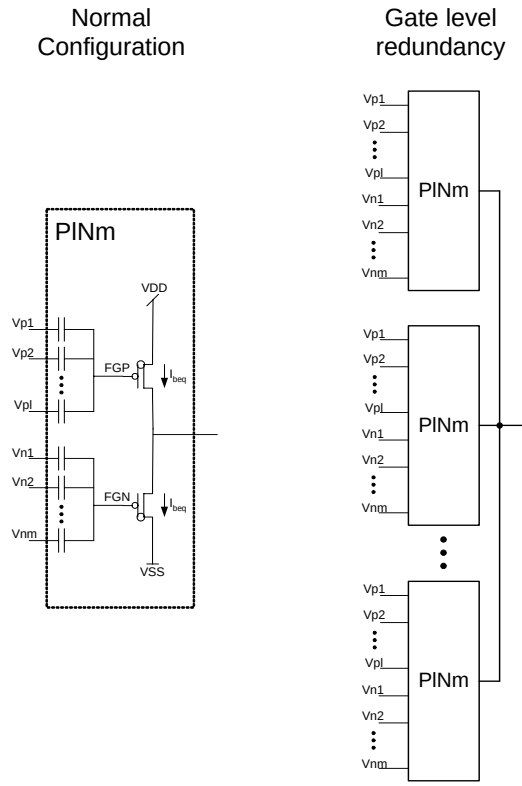

Fig. 8. Fault tolerance improvement scheme.

means a good relative accuracy between identically constructed circuit elements. Certain chip layout techniques [15] improve matching, as well as increasing the size of the drawn circuitry.

3 Increasing Faul Tolerance for Floating-Gate Linear Threshold Elements in Weak Inversion

One way to increase fault tolerance is to introduce redundancy on the gate level by letting, for an example, three identical gates (or more) with similar inputs drive the same output, as is illustrated in Figure 8. Then if one out of the arbitrarely number of redundant gates fails to take part in producing the correct output, the others could still maintain proper functionality. The steep dependency on output current and output resistance of the transistors from the gate voltage, in weak inversion compared to the classical area of operation, might make the principle useful. Letting more than one FGUVMOS circuit element share a common output node has been implemented earlier [7], [16], but for different purposes.

4 Using Linear Threshold Elements for Speeding up Perceptual Processing

The circuit elements herein are linear threshold elements, which are basic processing units in certain neural networks [17]. A classic model of a neuron is a linear threshold device, which computes a linear combination of the inputs, compares the value with a threshold, and outputs +1 or (-1) if the value is larger than the threshold [17].

Real neurons are found in the biological nervous systems. Human brains are by far superior to computers in solving hard problems such as combinatorial optimization and image and speech processing, although their basic building blocks are several orders of magnitude slower, which have boosted interest in the field of artificial neural networks [18], [19].

While neural networks have found wide application in many areas, the limitations and behavior of such networks are far from being understood [20].

Despite the power of digital computers they are not clever enough in the sense of perceptual and "intelligent" processing like seeing an object in the visual field, recognizing what it is, and taking proper action in real time. For biological systems, including humans, in general those are generally effortless tasks [21]. Such tasks are extremely difficult even for state-of-the-art computers. According to [21] the performance gap could never be narrowed by just increasing the clock frequencies of MPU's, integration densities of memories and further sophistication of software programs.

What is sometimes denoted "Intelligent data processing" tasks of today are presented as software programs which is reduced to a series of simple binary operations executed on MPU chips. However, in our brains the algorithms and the hardware are inseparably merged in the system [21]. To carry out the intelligent data processing directly in hardware, "neuron MOS transistors" are exploited in [21]. These transistors are multiple-input floating-gate devices where the floating-gate potential is determined as a weighted sum of multiple input signals via capacitance coupling, and thereby controls the current in the channel. The neuron MOS transistors are so named due to their similarity to the McCulloch-Pitts model of a neuron [21]. An association processor architecture utilizing these principles has been verified to about three orders of magnitude higher performance as compared to typical SISC processors of the time [21].

To which extent the linear threshold elements briefly described in this paper could switch places with the neuron MOS transistors in [21] could be researched further.

5 Using Linear Threshold Elements for Decreasing Chip Area

Assuming that each threshold gate can be built at a cost comparable to that of traditional AND, OR, NOT, termed AON logic, neural networks can be much more powerful than traditional logic circuits [17].

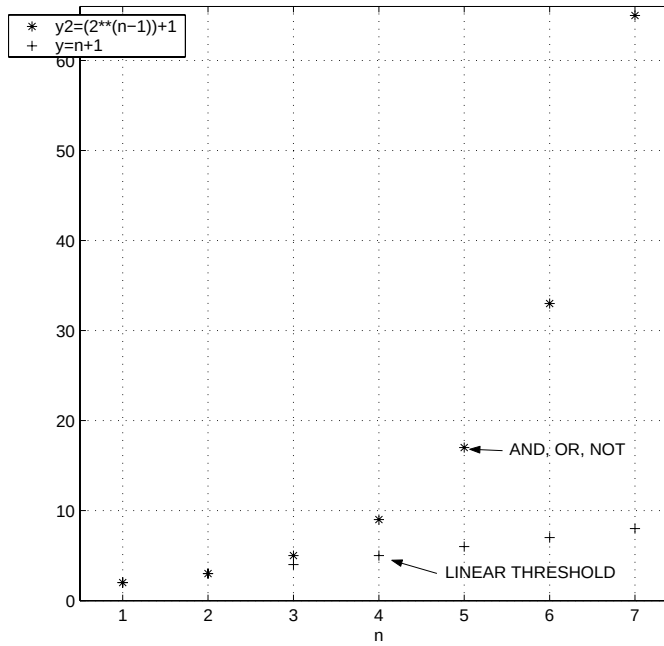

Fig. 9. Number of gates necessary to implement certain functions, as a function of number of bits, n. Number of gates using traditional AND-OR-NOT logic increases exponentially, while the increase can be linear when linear threshold elements are used.

In CMOS the production costs of a chip has a strong dependency on the chip area. Since the area of basic linear threshold elements, like P5N5 [8] and P1N3, are in the same order as normal gates, the costs might be comparable on the gate level.

For some functions, like XOR, the number of elements in a traditional AON circuit will grow exponentially with the number of bits in the input, while when implemented using linear threshold elements the number of gates are linear in the number of input bits [18]. Generally, a depth-2, AON circuit computing XOR of n bits requires at least $2^{n-1} + 1$ gates. A Linear Threshold circuit needs only n+1 gates. Figure 9 illustrates these relationships for n=1, $\cdots$, 7.

Another example of potential use of threshold logic [17] is: Whereas any logic circuit of polynomial size (in n) that computes the product of two n-bit numbers requires unbounded delay, such computations can be done in a neural network with "constant" delay. The product of two n-bit numbers and sorting of n n-bit numbers can be computed by a polynomial-size neural network using only 4 and 5 unit delays, respectively. Unit delay is equal to a "depth" of one for an artificial neural network [20].

Symmetric Boolean functions depend only on the sum of input values, and since the parity function is symmetric it can be computed in two layers of a

neural network whereas it takes unbounded delay to compute parity in a logic circuit [17].

For many years the topic of linear threshold logic has been approached in two different ways: theory on computational circuit complexity on one hand, and hardware implementation on the other. There has been very little interaction between the two approaches, as was stated in [18].

6 Conclusion

One type of real-time reconfigurable linear threshold element implemented in floating-gate UV-programmable CMOS technology has been presented, including measurements from a prototype CMOS chip demonstrating functionality. Power-Delay-Product numbers are among the lowest reported, and there are an interesting potential for implementing certain functions using less area then by traditional logic.

To determine the true computational power, due to number of inputs and capacitive weights for an example, further research should be done.

A simple proposal for how to make fault-tolerant circuitry using our linear threshold elements has been proposed.

An example on how similar CMOS circuitry could be used in speeding up a certain perceptual signal processing task about 3 orders of magnitude was briefly mentioned.

Due to the above mentioned aspects the true potential of this immature technology should be explored further.

References

1. P. Hasler, T. S. Lande *Overview of Floating-Gate Devices, Circuits and Systems*, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, January 2001.
2. *http://www.impinj.com* September, 2002.
3. S. Aunet, Y. Berg, T. Sæther *A New 2-MOSFET Universal Floating-Gate Element for Reconfigurable Digital Logic* Proceedings of the 19th IEEE Norchip Conference, Kista, Sweden, 12-13 November 2001, pp. 240-245.
4. T. Shibata, T. Ohmi *An Intelligent MOS Transistor Featuring Gate-Level Weighted Sum and Threshold Operations*, Technical Digest, International Electron Devices Meeting, 1991.
5. T. S. Lande, D. T. Wisland, T. Sæther, Y. Berg *FLOGIC - Floating-Gate Logic for Low-Power Operation* Proceedings of the 3rd IEEE International Conference on Electronics, Circuits and Systems, 1996, pp 1041-1044.
6. Y. Berg, T. S. Lande, S. Næss *Low-Voltage Floating-Gate Current Mirrors* Proceedings of the the Tenth Annual IEEE International ASIC Conference and Exhibit (ASIC), Portland, OR, USA, 7-10 Sept. 1997, pp 21-24.
7. Y. Berg, D. T. Wisland, T. S. Lande *Ultra Low-Voltage/Low-Power Digital Floating-Gate Circuits* IEEE Transactions on Circuits and Systems II, analog and digital signal processing, Vol. 46, Issue 7, pp. 930-936, July 1999.

8. S. Aunet *Real-time reconfigurable devices implemented in UV-light programmable floating-gate CMOS* Dissertation for the degree of doktor ingeniør, Norwegian University of Science and Technology, ISBN 82-471-5447-1, 2002.
9. Austria Mikro Systeme International AG *0.6 um CMOS CUP Process Parameters* Document no. 9933011, Rev. B, Oct. 1998.
10. Y. Berg, D. T. Wisland, T. S. Lande *Ultra Low-Voltage/Low-Power Digital Floating-Gate Circuits* IEEE Transactions on Circuits and Systems II, analog and digital signal processing, Vol. 46, Issue 7, pp. 930-936, July 1999.
11. J. Rabaey, M. Pedram, P. Landman *Low Power Design Methodologies* in J. Rabaey, M. Pedram (editors), Low Power Design Methodologies, Kluwer Academic Publishers, 1997.
12. S. Aunet, Y. Berg, O. Tjore, Ø. Næss, T. Sæther *Four-MOSFET Floating-Gate UV-Programmable Elements for Multifunction Binary Logic* Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, Volume 3,2001, pp 141-144.
13. P. P. Gelsinger *Microprocessors for the New Millennium: Challenges, Opportunities, and New Frontiers* Digest of technical papers, IEEE International Solid-State Circuits Conference, 2001, pp. 22-25.
14. G. Gilder, B. Swanson *Seattle Sunburst* Gilder Technology Report, 2002.
15. K. R. Laker, W. M. C. Sansen *Design of analog integrated circuits and systems* McGraw-Hill International Editions, ISBN 0-07-113458-1, 1994.
16. R. Bahr *A Design of Linear Four Quadrant Analog Multipliers Using Floating-Gate Transistors* thesis for the cand. scient. degree, University of Oslo, Faculty of Mathematics and Natural Sciences, Department of informatics, May 2001.
17. K. Y. Siu, J. Bruck *Neural Computation of Arithmetic Functions* Proceedings of the IEEE, No. 10, pp. 1669-1675, October 1990.
18. V. Bohossian *Neural Logic: Theory and Implementation* Dissertation for the Ph.D. degree, California Institute of Technology, 1998.
19. D. Hammerstrom *Computational Neurobiology Meets Semiconductor Engineering* Proceedings of the 30th IEEE International Symposium on Multiple-Valued Logic, 2000, pp. 3-12.
20. K. Y. Siu, J. Bruck, T. Kailath, T. Hofmeister *Depth Efficient Neural Networks for Division and Related Problems* IEEE Transactions on information theory, Vol. 39, No. 3, pp. 946-956, May 1993.
21. T. Shibata *Intelligent VLSI Systems Based on a Psychological Brain Model* Proceedings of the 2000 IEEE International Symposium on Intelligent Signal Processing and Systems, pp 323-332, Hawaii, U.S.A., November 5-8, 2000.