Physical Demonstration of Polymorphic Self-checking Circuits

Richard Ruzicka
Faculty of Information Technology
Brno University of Technology
Bozetechova 2, 612 66 Brno, Czech Republic
E-mail: ruzicka@fit.vutbr.cz

Lukas Sekanina
Faculty of Information Technology
Brno University of Technology
Bozetechova 2, 612 66 Brno, Czech Republic
E-mail: sekanina@fit.vutbr.cz

Roman Prokop
Faculty of Electrical Engineering and Communication
Brno University of Technology
Udolni 53, Czech Republic
E-mail: prokop@feec.vutbr.cz

Abstract

Polymorphic gates can be considered as a new reconfigurable technology capable of integrating logic functions with sensing in a single compact structure. Polymorphic gates whose logic function can be controlled by the level of the power supply voltage (Vdd) represent a special class of polymorphic gates. A new polymorphic NAND/NOR gate controlled by Vdd is presented. This gate was fabricated and utilized in a self-checking polymorphic adder. This paper presents an experimental evaluation of this novel implementation.

1 Introduction

Advances in CMOS technology have enabled the design and development of smart adaptive systems on silicon. These systems can perform sensing, computing and actuating on a single silicon die [17]. The adaptation can be achieved when a part of the design is configurable. In some applications, the adaptation process is performed only once with the aim to mitigate fabrication imperfections such as fluctuation of frequencies, faulty memory cells etc. In more complex applications, the adaptation process is executed continually in order to adapt the chip to a dynamic environment [4].

Polymorphic gates introduced by Stoica et al. [15] represent a new reconfigurable technology capable of integrating the logic function with sensing in a single compact structure. Logic function of the polymorphic gates can be controlled by an external environment in form of temperature, power supply voltage, control signals etc. [15, 16, 14, 19]. For example, the polymorphic AND/OR gate performs the AND function in its first mode (when the first environment is present, for example, temperature is 27 °C) and the OR function in its second mode (when the second environment is present, for example, temperature is 125 °C).

Various potential applications of polymorphic gates were analyzed using simulators, including:

- multifunctional adaptive circuits (i.e. a single circuit structure can perform two or more logic functions according to the external environment [13, 12, 5]),
- reduction of test vector volume (i.e. when some gates of the original circuit are replaced by polymorphic gates, it is possible to reduce test vector volume when the mode of polymorphic gates is changed before test is applied [13]),
- multifunctional counters (polymorphic gates enable a very efficient implementation of flip-flops and next-state logic [21]) and
- self-testing circuits [20, 10, 11] (see Section 3).

A six-transistor polymorphic gate was fabricated which operates as NOR when Vdd = 1.8 V and as NAND when Vdd = 3.3 V, i.e. the logic function is controlled by power supply voltage (Vdd) [14]. Note that a HP 0.5 micron CMOS technology was used for fabrication. This gate exhibits a very unconventional structure; in particular, it does not follow the well-accepted rules for transistor-level design of CMOS gates. As the original paper [14] does not provide detailed characterization of this gate, we do not completely

know its electric properties, especially the power consumption, maximum operating frequency, logic gain etc. No circuits utilizing this gate were described in open literature.

The aim of this paper is to demonstrate a physical implementation and evaluation of more complex polymorphic circuits than a single gate. In the first part of this paper, a new polymorphic NAND/NOR gate controlled by Vdd is described and characterized. This gate was fabricated in AMIS CMOS 0.7 micron technology. Proposed gate exhibits almost standard properties w.r.t. ordinary digital gates developed in the same technology. In addition to some differences in electrical properties, it significantly differs from Stoica's NAND/NOR gate [14] in the fundamental concept of required behavior. Stoica's NAND/NOR gate operates with identical logic levels for logic signal 1 in both modes, i.e. the logic 1 is represented using 1.8 V independently whether Vdd = 1.8 V or Vdd = 3.3 V (note that the logic 0 is always at 0 V). On the other hand, proposed gate moves logic levels with changing Vdd. When Vdd = 3.3V, the logic 1 is represented by 3.3V. When Vdd = 5V, the logic 1 is considered as 5 V. This property implies a different class of applications for proposed gate. The application class assumes that all circuit components (polymorphic as well as ordinary gates) are always connected to the same power supply unit. Polymorphic gates do change logic functions with changing Vdd. Ordinary gates must be able to perform correctly for Vdd = 3–5 V as well as for the logic 1 represented as 3–5 V.

The goal of the second part of this paper is to demonstrate the behavior and properties of a self-checking polymorphic adder which was physically implemented using proposed polymorphic gate and some ordinary gates. The most significant property of the adder is that it is able to detect a fault when the mode of polymorphic gates is switched. A fault is indicated by oscillations at the C_{out} output of the adder. This concept was introduced and simulated in our previous work [10, 11]. This paper moves the approach from simulations to reality.

The rest of this paper is organized as follows. Section 2 introduces a new NAND/NOR gate controlled by Vdd. In Section 3, the concept of polymorphic self-checking circuits is presented. Section 3.3 presents the results of experimental evaluation of proposed self-checking adder. Conclusions are given in Section 4.

2 Proposed NAND/NOR gate controlled by Vdd

In Figure 2, a newly proposed polymorphic NAND/NOR gate is shown. When Vdd = 3.3V, the gate exhibits the NOR function and when Vdd = 5V, the gate exhibits the NAND function. Figure 1 shows behavior of the gate for two different levels of Vdd.

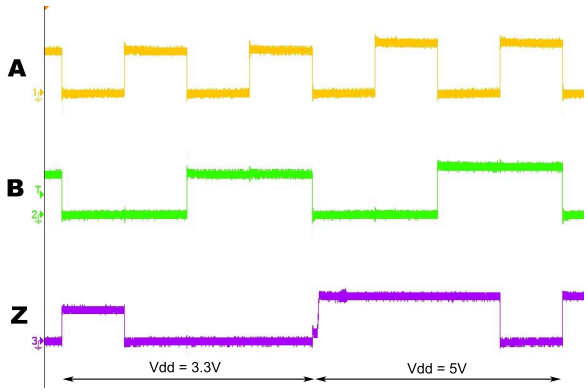

Figure 1. Proposed gate behavior (measured at 5 kHz)

First two transistors on inputs A and B serve as an inverter and guarantees the correct output logic levels for the 00 and 11 input combinations. These combinations cause the same output value for both modes of the gate. The difference in the gate behavior comes when A and B inputs differ. Then, the output level is 0 for the NOR function and 1 for the NAND function. In the NOR mode, the output level 0 is achieved by opened MN transistor (or MN_INV transistor respectively). As transistor MP_MIR1 is closed, Vdd is not sufficient to open MP_MIR and M2 transistors. In the NAND mode, the output level 1 is ensured by opened MP_MIR1 transistor. This transistor is much larger than all MN transistors and thus overrides their 0s. In Figure 3, the output voltage is shown for different Vdds and input combination "10". In the range of Vdd = 0–3.8 V, the gate exhibits the NOR function (logic 0 at the output). For Vdd higher than 3.9 V, the gate exhibits the NAND function – the output is at logic 1 (the output voltage is close to Vdd). It can be seen that there is a very small hysteresis.

The gate was fabricated in CMOS 0.7 micron technology (area 55.8 x 68.2 μm – not optimized in this design). Measured characteristics show that the logic function is changed around Vdd = 3.8 V. It is stable for Vdd below 3.7 V and over 3.9 V. In the neighborhood of Vdd = 3.9 V the output of the gate may oscillate. This is because Vdd can vary due to varying power consumption during switching of transistors used in the gate. However, the function instability range is adequately small and reduced by hysteresis. The absolute maximum ratings of the gate are inferred from used technology and shown in Table 1.

Table 2 summarizes DC electrical characteristics of the gate. Because the gate is polymorphic and exhibits two functions according to Vdd, these characteristics were measured for both levels of Vdd. It can be seen that the input threshold voltage moves from approx. 1V in the NOR mode (Vdd = 3.3 V) to approx. 1.45 V in the NAND mode (Vdd =

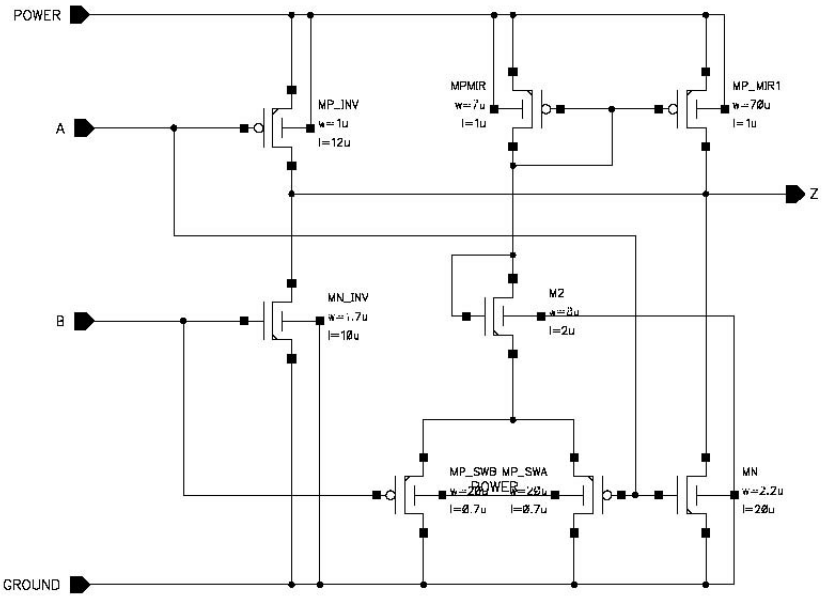

Figure 2. Proposed polymorphic NOR(when Vdd = 3.3 V)/NAND (when Vdd = 5 V) gate

Table 1. Absolute maximum ratings of the proposed gate

	min.	max.
Vdd	-0.5 V	7 V
input/output voltage	-0.5 V	Vdd + 0.5 V
operating temp.	0°C	+70°C
junction temp.	-40°C	+125°C

5 V). Also output logic levels move with Vdd (see Table 2).

Table 3 summarizes measured dynamic characteristics of the gate. Delays are depending on the gate mode. Especially in the NAND mode, the delays of rising and falling edge differ by an order of magnitude. This difference is caused by different physical sizes of used transistors. Note that all measurements were performed with ESD protection circuits used on the inputs and output of the gate. The maximum operational frequency was measured as a frequency of a ring oscillator containing one polymorphic gate.

3 Polymorphic self-checking adders

Various applications employ self-checking circuits capable of on-line detecting transient and permanent faults, especially in systems where dependability and data integrity are important. In self-checking circuits, the functional logic

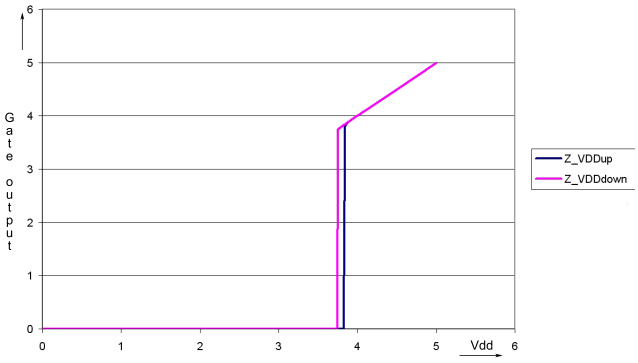

Figure 3. Output voltage for input = ”10” and different Vdd

Table 2. DC electrical characteristics of the proposed gate

	Vdd = 3.3 V	Vdd = 5 V
V_{iL}	max. 0.79 V	max. 1.09 V
V_{iH}	min. 1.37 V	min. 1.78 V
V_{oL}	0.0 V	0.0 V
V_{oH}	3.29 V	4.97 V
f_{max}	38.6 MHz	56.2 MHz

Table 3. Dynamic characteristics of the proposed gate

	Vdd = 3.3 V	Vdd = 5 V
A to Z t_{pLH}	64 ns	2.8 ns
A to Z t_{pHL}	56 ns	24.4 ns
B to Z t_{pLH}	81 ns	3.0 ns
B to Z t_{pHL}	71 ns	26.4 ns

provides the output encoded with an error detecting code and the checker determines if the output is a codeword. The checker itself traditionally provides a two-rail signal in which an error is indicated by both rails being at the same value. Self-checking circuits are conventionally constructed by adding checking logic around an unmodified original output of the circuit [9, 1, 8]. An innovative solution was proposed by Garvie who utilized the evolutionary algorithm to design small self-checking circuits which merge the original function with the checker logic [3]. He reported better results than conventional approaches if the total area of the circuit is measured (less than duplication overhead). However, conventional as well as evolutionary approaches require special signals which indicate that a fault is present in the circuit. When more complex circuits are composed of these elementary self-checking circuits, these special signals have to be interconnected and aggregated to obtain the global information about the fault.

3.1 Proposed adder

In our recent work, we proposed a 1-bit adder which exhibits self-checking properties; however, which does not utilize any additional signals to indicate a fault [10, 11]. Figure 4 shows that the sum calculation is quite standard. A conventional 1-bit full adder uses two operands, A and B, and input carry, C_{in}. It generates the sum $S = A \oplus B \oplus C_{in}$. On the other hand, the carry out output of proposed adder is calculated unconventionally using the three NAND/NOR gates that are connected to the circuit inputs as well as to inverted sum, $\overline{S}$. Independently of the level of Vdd, this logic network always generates a correct carry-out signal when there is no fault present in the circuit. The NAND/NOR gates can be switched simultaneously between NAND and NOR function by changing Vdd. If the inputs were not changed and the control signal were switched at the frequency kf (k is a positive integer and f is an operational frequency of the circuit) then the circuit outputs would remain steady, which is a normal operation of the circuit. Figure 4 also shows that n-bit ripple-carry adder can be composed when n 1-bit adders are used.

However, when a stuck-at-fault is present within the adder then C_{out} output will oscillate between 0 and 1 at the same frequency as Vdd oscillates. In addition to its primary function, this output works as the indicator of a stuck-at-fault in the circuit. Vdd is switched whenever the circuit should be tested. The test can be performed either before the system is put into operation or in some time slots devoted to testing of the system or on-line.

The motivation for this approach reflects the fact that some of future computing architectures will probably contain a large massively parallel array of locally interacting computing elements in which the issue of efficient wiring will be very important. The existence of some extra signals or even global signals will be problematic [2].

The proposed adder costs 42 transistors (considering that the XOR as well as NAND/NOR gate requires 8 transistors and inverter costs 2 transistors), i.e. the overhead is 75% in comparison with a conventional optimized transistor-level implementation of a 1-bit adder which costs 24 transistors [18]. The overhead is slightly higher than conventionally designed self-checking adders [7, 3] (for example, the self-checking adder reported in [6] consists of 36 transistors); however, the proposed adder does not require any additional wires. It is assumed that standard gates (the XORs and inverters) operate correctly for the both modes of polymorphic gates.

3.2 Fault detection capabilities

Table 4 shows the fault coverage for all possible test vectors applied on the polymorphic adder. Test vectors are indexed 0–7 which corresponds with the circuit inputs ordered as (Cin, B, A). Symbol "x" means that a corresponding test vector is able to induce oscillations at the carry-out output for the particular stuck-at-fault. We can observe that the stuck-at-fault can not be detected only when injected to gate g6. The reason is that this gate is connected directly to the primary output of the adder. It is easy to derive from Table 4 that by applying test vectors $M_{min} = \{1, 2, 3, 5\}$ or $\{1, 2, 5, 6\}$ or $\{2, 4, 5, 6\}$, all single faults can be detected. In other words, at least four test vectors have to be applied in order to initiate oscillations at the carry-out output when a single fault is present in the adder. The probability of fault detection is 0.325 when only a single randomly generated test vector is applied.

3.3 Physical implementation

In order to verify polymorphic self-checking properties, proposed adder was created in a breadboard using two XOR gates (2/4 of 4030 used), three NAND/NOR gates (see Section 2) and inverter (also an XOR gate from 4030 was used as the inverter). Again, NAND/NOR gates are equipped

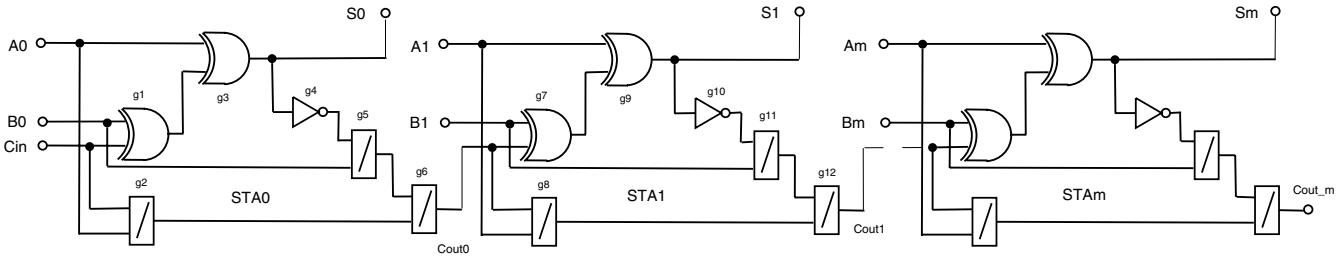

Figure 4. Proposed 1-bit self-checking adder and its utilization in the ripple carry adder

Table 4. Fault coverage of proposed polymorphic self-checking adder

M_i /vector	0	1	2	3	4	5	6	7	stuck-at
M_1	.	.	x	x	x	x	.	.	0
M_2	.	x	.	.	x	.	.	.	0
M_3	.	x	x	.	x	.	.	.	0
M_4	.	.	.	x	.	x	x	.	0
M_5	x	.	x	.	.	.	.	.	0
M_6	.	.	.	.	.	.	.	.	0
M_1	.	x	.	.	.	.	x	.	1
M_2	.	.	.	x	.	.	x	.	1
M_3	.	.	.	x	.	x	x	.	1
M_4	.	x	x	.	x	.	.	.	1
M_5	.	.	.	.	.	x	.	x	1
M_6	.	.	.	.	.	.	.	.	1

with ESD protection. The power supply unit is synchronized with input signals. The level of Vdd is switched between 3.3V and 5 V at frequency $f_{cin}/2$ where f_{cin} is the frequency of C_{in} signal (f_{cin} = 2.4 kHz in experiments). The following properties of the adder were measured:

- It was verified that the adder computes S and C_{out} correctly for all possible input combinations and both levels of Vdd.
- When Vdd is switched, S and C_{out} can oscillate if the change of Vdd is not steep enough.
- Maximum operational frequency of the adder corresponds with operational frequency of the polymorphic gate (see section 2).
- It was verified that the adder works according to Table 4 when a stuck-at-fault is present in the circuit. Figure 5 shows one particular example of this behavior. A stuck-at-zero was inserted at the output of gate g4. We can observe a different response pattern at C_{out} for Vdd = 3.3 V and Vdd = 5 V which indicates the presence of a fault.

3.4 Note on a ripple carry self-checking adder

Consider a 2-bit self-checking adder (Fig 4). When a fault is present in STA1 then the fault is indicated by oscillations at C_{out1} as explained for the 1-bit full adder in the previous paragraph. In order to detect at C_{out1} a fault which is present in STA0 then C_{out0} has to propagate through STA1. This can only be achieved by setting $A_1 \neq B_1$. Similarly to previous section, we can observe that the stuck-at-fault can not be detected when injected to gates 6 or 12. Only four test vectors are needed to detect all single faults at remaining gates and thus to initiate the oscillations at C_{out1}. This observation also holds for n-bit ripple carry adder.

4 Conclusion

The idea of polymorphic self-checking was initially presented in [20]; however, the authors composed their adder of 14 NAND/NOR polymorphic gates and allowed oscillations on both outputs of the adder. No physical demonstration of the concept was published. In comparison to [20], the advantage of proposed solution is that: (1) faults are indicated at C_{out} only, (2) faults can propagate through all stages of the n-bit ripple-carry adder to be detected at the primary C_{out} of the whole n-bit adder and (3) the number of transistors was significantly reduced.

Although we used a relatively obsolete technology to fabricate polymorphic gates and the experiments were performed on a breadboard, the idea that the self-checking property can be achieved using polymorphic gates was clearly demonstrated. Further optimization of the proposed NAND/NOR gate and a compact implementation of the adder on a single chip are planned for our future research.

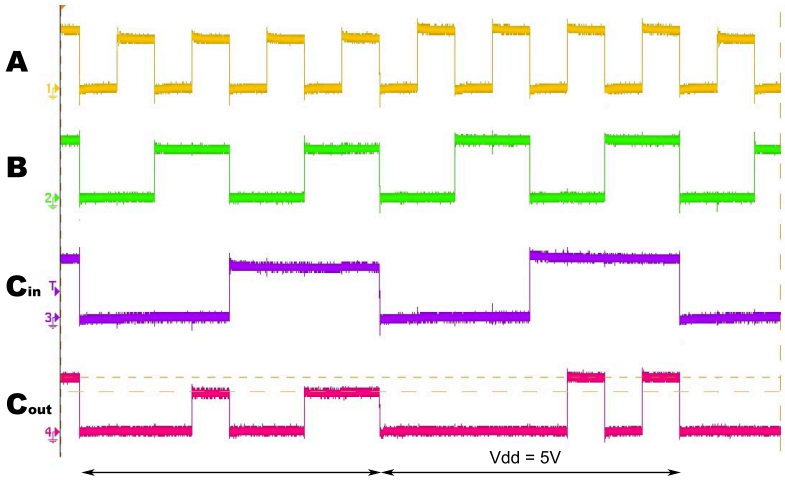

Figure 5. Different values at C_{out} (when Vdd is switched) indicate a stuck-at-fault at output of g4

Acknowledgements

This work was partially supported by the Grant Agency of the Czech Republic under contract No. 102/06/0599 *Methods of polymorphic digital circuit design* and the Research Plan No. MSM 0021630528 – *Security-Oriented Research in Information Technology*.

References

[1] M. Diaz, P. Azéma, and J.-M. Ayache. Unified design of self-checking and fail-safe combinational circuits and sequential machines. *IEEE Trans. Computers*, 28(3):276–281, 1979.

[2] M. Frank. *Reversibility for Efficient Computing*. PhD thesis, Massachusetts Institute of Technology, 1999.

[3] M. Garvie. *Reliable Electronics through Artificial Evolution*. PhD thesis, University of Sussex, 2005.

[4] T. Higuchi, Y. Liu, and X. Yao. *Evolvable Hardware*. Springer, 2006.

[5] W. Luo, Z. Zhang, and X. Wang. Designing polymorphic circuits with polymorphic gates: a general design approach. *IET Circuits, Devices & Systems*, 1(6):470–476, 2007.

[6] D. Marienfeld, V. Ocheretnij, M. Gössel, and E. S. Sogomonyan. Partially duplicated code-disjoint carry-skip adder. In *Proc. of the 17th IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 78–86. IEEE, 2002.

[7] V. Ocheretnij, D. Marienfeld, E. S. Sogomonyan, and M. Gössel. Self-checking code-disjoint carry-select adder with low area overhead by use of add1-circuits. In *10th IEEE International On-Line Testing Symposium*, pages 31–36. IEEE Computer Society, 2004.

[8] S. J. Piestrak. Feasibility study of designing tsc sequential circuits with 100% fault coverage. In *17th IEEE Int. Symposium on Defect and Fault-Tolerance in VLSI Systems*, pages 354–364. IEEE Computer Society, 2002.

[9] D. K. Pradhan. *Fault-Tolerant Computer System Design*. Prentice Hall, 1996.

[10] L. Sekanina. Design and Analysis of a New Self-Testing Adder Which Utilizes Polymorphic Gates. In *Proc. of the 10th IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop DDECS 2007*, pages 246–246, Krakow, Poland, 2007. IEEE Computer Society.

[11] L. Sekanina. Evolution of Polymorphic Self-Checking Circuits. In *Proc. of the 7th Conf. on Evolvable Systems: From Biology to Hardware*, number 4684 in LNCS, pages 186–197, Wuhan, China, 2007. Springer.

[12] L. Sekanina, L. Starecek, Z. Gajda, and Z. Kotasek. Evolution of multifunctional combinational modules controlled by the power supply voltage. In *Proc. of the 1st NASA/ESA Conference on Adaptive Hardware and Systems*, pages 186–193. IEEE Computer Society, 2006.

[13] L. Sekanina, L. Starecek, Z. Kotasek, and Z. Gajda. Polymorphic Gates in Design and Test of Digital Circuits. *Int. J. Unconventional Computing – accepted*, 4:18, 2008.

[14] A. Stoica, R. Zebulum, X. Guo, D. Keymeulen, I. Ferguson, and V. Duong. Taking Evolutionary Circuit Design From Experimentation to Implementation: Some Useful Techniques and a Silicon Demonstration. *IEE Proc.-Comp. Digit. Tech.*, 151(4):295–300, 2004.

[15] A. Stoica, R. S. Zebulum, and D. Keymeulen. Polymorphic electronics. In *Proc. of Evolvable Systems: From Biology to Hardware Conference*, volume 2210 of *LNCS*, pages 291–302. Springer, 2001.

[16] A. Stoica, R. S. Zebulum, D. Keymeulen, and J. Lohn. On polymorphic circuits and their design using evolutionary algorithms. In *Proc. of IASTED International Conference on Applied Informatics AI2002*, Insbruck, Austria, 2002.

[17] M. Valle. *Smart Adaptive Systems on Silicon*. Kluwer, 2004.

[18] N. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective (3rd edition)*. Addison Wesley, 2004.

[19] R. S. Zebulum and A. Stoica. Four-Function Logic Gate Controlled by Analog Voltage. *NASA Tech Briefs*, 30(3):8, 2006.

[20] R. S. Zebulum and A. Stoica. Multifunctional Logic Gates for Built-In Self-Testing. *NASA Tech Briefs*, 30(3):10, 2006.

[21] R. S. Zebulum and A. Stoica. Ripple Counters Controlled by Analog Voltage. *NASA Tech Briefs*, 30(3):2, 2006.