1999

Reconfigurable Shift Switching Parallel Comparators

R. Lin

S. Olariu
Old Dominion University

Original Publication Citation

Lin, R., & Olariu, S. (1999). Reconfigurable shift switching parallel comparators. *Vlsi Design, 9*(1), 83-90. doi: 10.1155/1999/79875

VLSI DESIGN
1999, Vol. 9, No. 1, pp. 83–90
Reprints available directly from the publisher
Photocopying permitted by license only

Reconfigurable Shift Switching Parallel Comparators

R. LIN[a] and S. OLARIU[b,*]

[a] *Department of Computer Science, SUNY at Geneseo, Geneseo, NY 14454;*
[b] *Department of Computer Science, Old Dominion University, Norfolk, Va 23529*

(Received 5 May 1997)

We present novel asynchronous VLSI comparator schemes which are based on recently proposed reconfigurable shift switch logic and the traditional (precharged) CMOS domino logic. The schemes always produce a semaphore as a by-product of the process to indicate the end of domino process, which requires no additional delay and a minimal number of additional devices. For a large percentage of inputs the computations are much faster than traditional synchronous comparators due to the full utilization of the inherent speed of the circuits. Also the schemes are simple, area compact and stable.

Keywords: Parallel architectures, reconfigurable architectures, digital signal processing, VLSI design, domino logic, computer arithmetic

1. INTRODUCTION

Recently, shift switch logic (defined by state signals and shift switches, refer to [6–9] and see below) has been proposed. It is shown that the new logic method is an efficient alternative to the traditional switching logic (*i.e.*, binary signals and logic gates) for the designs of a number of arithmetic devices including parallel counters, multipliers, and fast adders [5–10].

In this paper we propose novel VLSI asynchronous comparator schemes based on shift switch logic and traditional (precharged) CMOS domino logic [2, 4, 13, 14], which we call shift [1]switching with domino logic. The technique allows to broadcast state signals (on short reconfigurable buses [3, 7, 10] containing cascaded shift switches) in a form of a chain of pulling-down and/or pulling-up the precharged switches. It possesses the following attractive features: (1) The comparators produce two-bit output xy which represents the comparison result for two inputs a and b, such that $xy = 01$, 10, 11 indicating $a > b$, $a < b$, $a = b$ respectively, and if the result is not ready $xy = 00$. The comparators also produce (in parallel) a bit semaphore to indicate the end of each domino propagation, and the circuits can run asynchronously without high frequency clocking schemes; (2) The comparison algorithms consist of two phases, precharge and evaluation (with single

*Corresponding author. e-mail: olariu@cs.odu.edu

control signal called P/E). The precharge starts when $P/E = 0$ and is done in parallel (in less than a full adder delay). The evaluation begins when $P/E = 1$ and it consists of several domino discharging processes (2 for 32-bit); (3) It is faster than traditional designs [11–14], since the full inherent speed of the computation can be utilized; and (4) Due to that a comparator delay can be determined by individual input data instead of the worst-case input, for a large percentage of inputs, the computation is significantly faster than the worst case delay, also the comparator is not delay sensitive. Moreover, the proposed method is applicable to other asynchronous arithmetic circuits, such as fast adders [8].

The remainder of the paper is organized as follows: Section 2 defines state signals and *GP* switches. Section 3 presents a 7-bit comparator. Section 4 presents a 32-bit comparator.

2. STATE SIGNALS AND SHIFT SWITCHES

Shift switch logic is defined by state signals which represent small integers and shift switches which are basic logic units to operate the state signals (refer to [6–9]). To be self-contained we show the relevant definitions below.

DEFINITION 1 A state signal with value I ($0 \le I \le w-1$; $w \ge 2$) is represented by bit sequence $b_0, b_1, \ldots, b_{w-1}$ (here we assume the order is either right to left or top-down) with the unique bit u (either 0 or 1) in the I-th position (see Fig. 1). A state signal is said n-(p-) type denoted by $\boldsymbol{I}_{(w)}$ ($\boldsymbol{I}_{(\bar{w})}$) if $u = 0$ (1). A state signal may be denoted by $I_{(w)}$ regardless of type, or $\boldsymbol{I}$ (value only).

DEFINITION 2 A degenerate state signal with value $\boldsymbol{I}$ ($0 \le I \le w-1$) is state signal $\boldsymbol{I}$ with the 0-th bit removed, thus represented by $b_1, b_2, \ldots, b_{W-1}$. (Fig. 2), and its value is 0 if there is no unique bit in the sequence, otherwise it is equal to the corresponding state signal (with bit 0 added). It can be seen as a special state signal, and is said n-(p-) type denoted by $\boldsymbol{I}_{(\underline{w}')}$ ($\boldsymbol{I}_{(\bar{w}')}$) if $u = 0$ (1).

DEFINITION 3 (shift switch in general) Given function $F(X, Y) = (U, V)$, or $F(X, Y) = U$, where X, Y, U and/or V are (small) state signals, a pass-transistor (or transmission gate) based digital filter which implements function F by shifting (in some way) the input state signals to obtains the outputs is called a shift switch of function F.

In this paper we consider only a specific type of shift switches called *GP* switch.

DEFINITION 4 A shift switch which implements function $GP(X, Y) = U$, such that $U = Y$ if $Y > 0$ and $U = X$ if $Y = 0$, is called *GP* (generate/propagate) shift switch. Figure 3 shows an example of such a switch.

DEFINITION 5 If a *GP* shift switch has an additional input called P/E (precharge/evaluation), such that, P/E, X, Y and U are defined by Table I (or II), it is called *GP*1 (or *GP*2) switch.

The examples of *GP*1 and *GP*2 are depicted in Figures 4(a) and 4(b). They are assumed to work in two phases (precharge and evaluation). The input and output values are represented by state

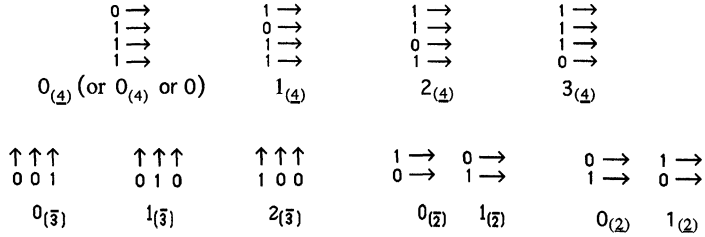


FIGURE 1 Some state signals.

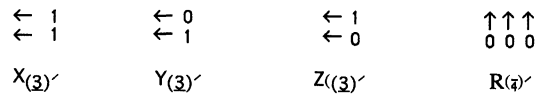


FIGURE 2 Some degenerate state signals ($X = 0$, $Y = 1$, $Z = 2$, $R = 0$).

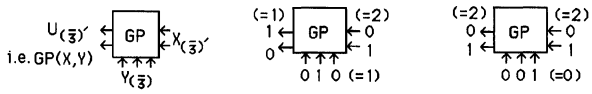


FIGURE 3 A *GP* shift switch and sample inputs and outputs.

TABLE I Shift switch *GP*1

precharge/ evaluation (P/E)	input X	input Y	output U
0 (precharge)	$0_{(3)'}$ *i.e.* 11	$0_{(4)'}$ 000	$0_{(3)'}$ 11
1 (evaluation)	$X_{(3)'}$	$Y_{(3)'}$	$GP(X, Y)_{(3)'}$

TABLE II Shift switch *GP*2

precharge/ evaluation (P/E)	input X	input Y	output U
0 (precharge)	$0_{(4)'}$ *i.e.* 111	$0_{(4)'}$ 000	$0_{(4)'}$ 111
1 (evaluation)	$X_{(3)'}$	$Y_{(3)'}$	$GP(X, Y)_{(3)'}$

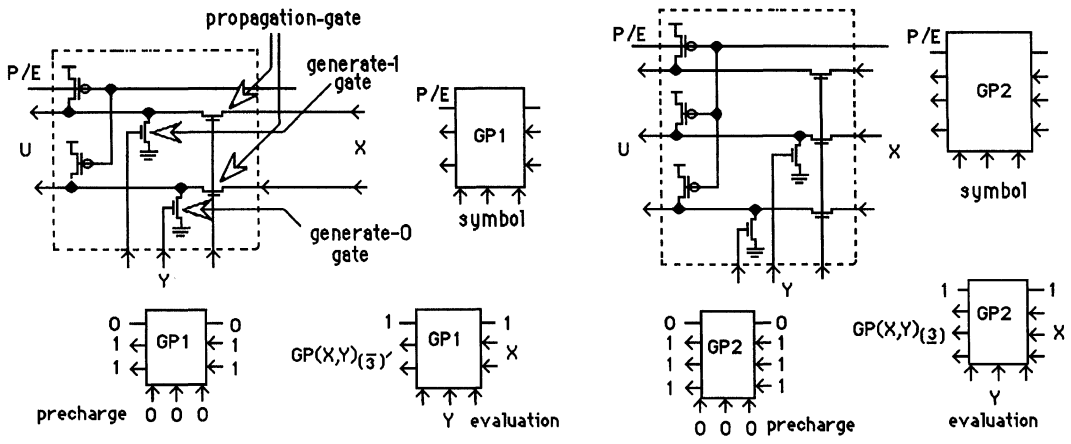
signals. During the precharge phase ($P/E = 0$), both switches have the same corresponding input/output values (all are 0) and are represented by the state signals. During the evaluation phase ($P/E = 1$), both switches produce horizontal output $U = GP(X, Y)$, but represented by state signals ($GP1$) and degenerate state signals ($GP2$) respectively.

The significance of using state signals and shift switches lies in the fact that some basic arithmetic and logic operations (such as adding two small integers, and the evaluation of logic functions like GP) can be done by shifting one state signal X in a way controlled by the other state signal Y. The operations can be efficiently performed by shift switches, essentially letting X pass through a small number of pass-transistors (or trans-gates) that are preset by Y. Also since both shifting and control signals are state signals, we can use the shift-out of a switch to control another switch, thus we may organize an efficient network of shift switches (or enhanced reconfigurable buses) for a desired arithmetic computation.

3. THE 7-BIT COMPARATOR: A PRECHARGED CMOS SCHEME

A linear array of the proposed GP switches, each associated with a CMOS domino logic unit, called *comp* unit, can implement a VLSI-efficient small-size comparator, such as a 7-bit comparator. Figures 5 and 6 show the evaluation phases of a comp unit for bit j (for $1 \le j \le 6$), and a comp unit for bit 0 respectively. The precharge and evaluation are two phases of the computation on each of these logic units. During the precharge phase the control signal P/E (precharge/evaluation, which may be received from a previous asynchronous device) is set to 0. It is easy to verify that a comp unit is precharged as follows: $g1_j$ (generate-1), $g0_j$ (generate-0), and p_j (propagate) are all low (for $1 \le j \le 6$); two output bits ($gl_0, g0_0$) of comp unit 0 are all high. When P/E is set to 1, the evaluation phase begins. The logic units are discharged as follows: For comp units $j (1 \le j \le 6)$: $g1_j$ (or $g0_j$) is discharged to high if $a_j > b_j$ (or $b_j > a_j$); p_j is discharged to high if $a_j = b_j$; note that state signal $cb(j)_{(\bar{3})} = (g1, g0, p)$ is called (the j-th) comp-bit state signal. For comp unit 0, $g1_0$ (or $g0_0$) is discharged to low if $a_0 \le b_0$ (or $b_0 \le a_0$).

Figure 7 shows the 7-bit comparator scheme. It consists of six $GP1$ switches connected with seven comp units (including comp unit 0). When signal P/E is set to 0 ($\overline{P/E}$ is set to 1), all $GP1$ switches and comp units are precharged as described above. The precharge phase is as the follows: each propagation gate in $GP1$ is off and will be kept off if the corresponding comp unit bit evaluation signal $cb(j)_{(\bar{3})} = (g1_j, g0_j, p_j) \ne (0, 0, 1) (1 \le j \le 6)$. This ensures a stable discharge during the evaluation phase, *i.e.*, the domino discharging can



(a) precharge (b) evaluation

FIGURE 4 Precharged CMOS schematics of $GP1$ and $GP2$. Note that P/E is the enable signal with values 0 and 1 for precharge and evaluation respectively.

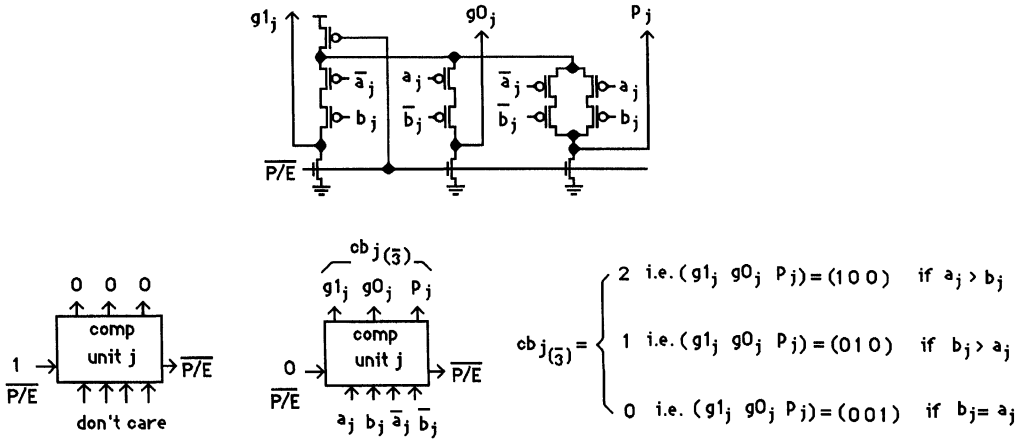
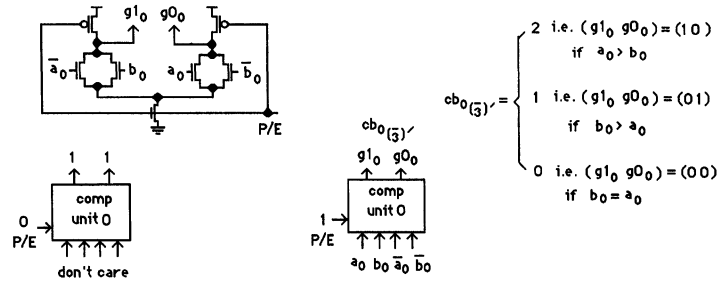


FIGURE 5 Comp unit $j(1 \leq j \leq 6)$.



(a) precharge (b) evaluation

FIGURE 6 Comp unit 0.

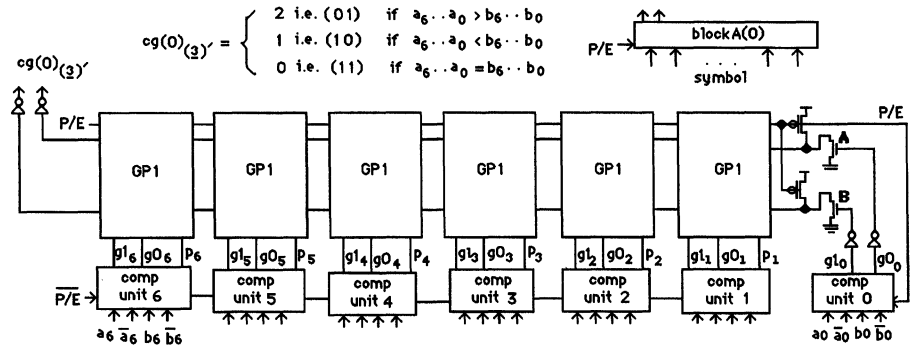


FIGURE 7 BlockA(0), a 7-bit comparator (evaluation phase). (a) precharge phase (note: the values of inputs are "don't care"); (b) evaluation phase.

correctly propagate from the comp units to *GP*1 switches and then generate comp group output $cg(0)_{(3)'}$ (or $cg0$) for blockA(0).

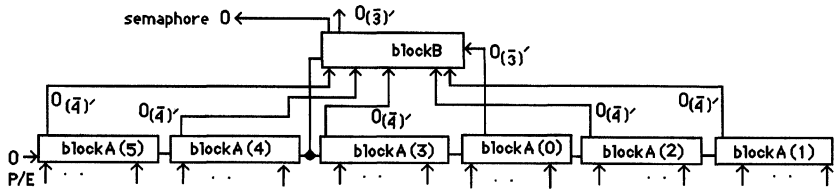
The evaluation phase is as follows: First, the discharge of each comp unit j generates evaluation output signal $cb(j)_{(\bar{3})} = (g1_j, g0_j, p_j)$ (see Fig. 5) which then discharges the corresponding switch *GP*1 if $cb(j)_{(\bar{3})} \neq 0$, otherwise it turns on the corresponding propagation gates of the *GP*1 for propagation of the comparison result of lower bits. The bus (with *GP*1s) is then partitioned (by the propagation gates), and the worst case of comp evaluation signal propagation (or discharging) will start from gate A and/or B passing all 6 *GP*1s to produce $cg0$. The worst case delay of the comparator is T_c (time to discharge a comp unit) + $7T_{GP}$ (time to discharge seven cascaded pass-transistors including A or B) + T_{inv} (an inverter delay, low to high). The best case delay of the comparator is $T_c + T_{GP} + T_{inv}$.

4. THE 32-BIT SCHEME: SHIFT SWITCH WITH DOMINO LOGIC

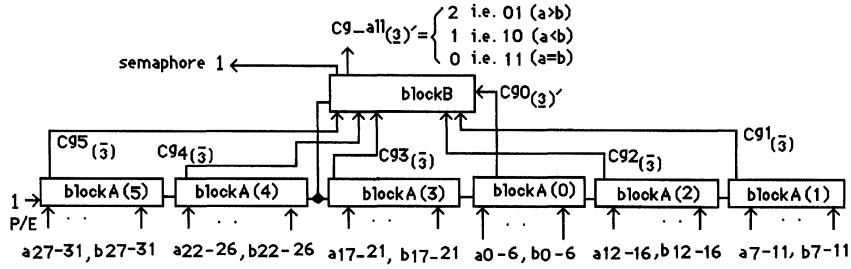
A 32-bit comparator can be constructed by several *GP* switch blocks and organized in two levels (see Fig. 8). The first level consists of six blocks:- blockA(0), *i.e.*, the 7-bit comparator and blockA(i) (for $1 \leq i \leq 5$), The second level is a single block termed blockB.

Figure 9 shows blockA(i) (for $1 \leq i \leq 5$). It contains five cascaded *GP*2 shift switches and five comp units. It produces state signal $cg(i) = (gg1, gg0, pg)$ which represents the group's comparison result, *i.e.*, values 2, 1, 0 representing the relationship between the corresponding bit segments of a and b for $>$, $<$ and $=$, and is called (the i-th) comp-group output (state) signal. In other words, $gg1 = 1$ (or $gg0 = 1$, or $pg = 1$) if the group's bit segment of a is greater than (or less than, or equal to) the bit segment of b. The worst case and the best case delays of the first level are dominated by the longest blockA(0), *i.e.*, $T_c + 7T_{GP} + T_{inv}$ and $T_c + T_{GP} + T_{inv}$ respectively. Figure 10 shows the second level block, *i.e.*, blockB. It is similar to blockA(0), except the follows:

(1) it has five cascaded *GP*1 switches and its five vertical inputs are comp-group output signals from blockA(i) (for $1 \leq i \leq 5$), and its horizontal input (for evaluation) is the degenerate comp-group output $cg0_{(3)'}$ of blockA(0);
(2) its two outputs are a 2-bit degenerate comp all-groups output signal $cg\_all_{(3)'})$ and a semaphore. The 2-bit signal $cg\_all_{(3)'}$ which is the final comparison result, has a value either 0, or 1, or 2, *i.e.*, 11, 10 or 01, which represents $a = b$, $a < b$ and $a > b$ respectively. If $cg\_all_{(3)'}$ is 00 the semaphore is 0, meaning the result is not ready, otherwise the semaphore is 1, meaning the result is ready;



(a) precharge phase (note: the values of inputs are "don't care")



(b) evaluation phase

FIGURE 8 The 32-bit comparator.

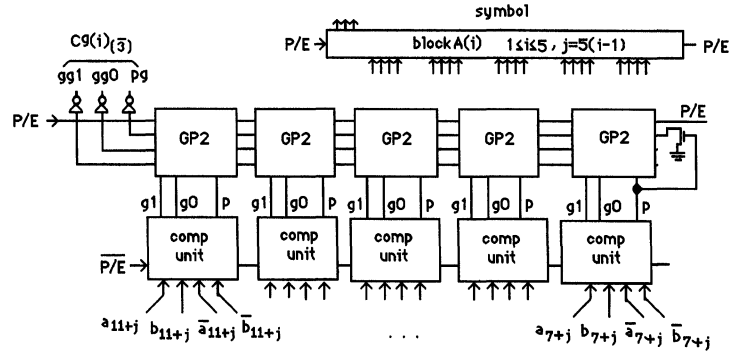


FIGURE 9 BlockA(i) (for $1 \leq i \leq 5$; evaluation phase is shown).

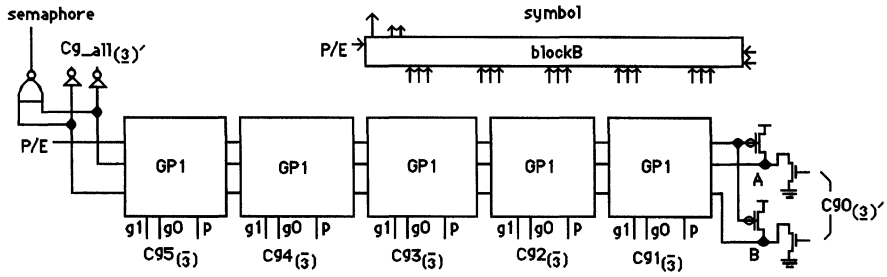


FIGURE 10 BlockB (evaluation phase is shown).

The output bits and the semaphore are produced at about the same time. This is the unique feature of our scheme. More significantly, the best case delay of the comparator can be specified as 3 times of a cascaded pass-transistors delay plus two times of an inverter delay (or $3\,T_{GP} + 2T_{\text{inv}}$), the

TABLE III Area Comparisons of comparators

Comparators	device count (number of transistors)			
	level 1	level 2	total area	area per output bit
asynchronous	662	42	704	352
synchronous	417	19	436	436

worst and the average case delays of the comparator can be specified as $13T_{Gp} + 2T_{\text{inv}}$ and $6T_{GP} + 2T_{\text{inv}}$ respectively. This implies that the asynchronous scheme can be adapted to significantly reduce average case delay of a comparator (by automatically choosing single or double instruction cycles for the computation). This will lead to an expected 60% faster for 50% inputs, or 30% faster for other 50% inputs (by a simulation program).

Note that if a synchronous comparator is preferred we can modify the first level of GP switches to implement only $g1$ and p (*i.e.*, removing $g0$ circuit), thus eight transistors per input bit may be reduced. However, such a synchronous comparator can provide only one output bit indicating either $a > b$ or $a \leq b$ (or alternatively $a \geq b$ or a $a < b$). In order to provide all three comparison results two such comparators may be required.

We can count the number of devices (in terms of pass-transistors) for both asynchronous and synchronous comparators as listed in Table III. It is clear that the synchronous schemes have about 30% savings in area, however, asynchronous schemes are more compact for area per output bit.

Acknowledgements

The work was supported, in part, by National Science Foundation under grants MIP-9630870 and CCR-9522093 and ONR grant N00014-95-1-0779.

References

[1] Hwang, H. (1979). *Computer Arithmetic. Principles, Architectures and Designs*, New York: Wiley.

[2] Hwang, I. S. and Fischer, A. L. (1989). Ultrafast compact 32-bit CMOS adders in multi-output domino logic, *IEEE J. Solid-State Circ.*, **24**(2), 358–369.

[3] Jang, J., Park, H. and Prasanna, V. K. (1994). A bit model of the reconfigurable mesh, *Proc. of the Workshop on Reconfigurable Architectures, the 8th International Parallel Processing Symposium*, Cancun, Maxico.

[4] Krambeck, R. H., Lee, C. M. and Law, H. S. (1982). High-Speed Compact Circuits with CMOS, *IEEE Journal of Solid-State Circuits*, **SC**-17(3).

[5] Lin, R., Shift switching with domino logic: asynchronous VLSI comparator schemes, to appear in *Proc. of The 10th Internation Conference on VLSI Design*, January, 1997, Hyderabad, India.

[6] Lin, R. Shift switching and novel arithmetic schemes, In: *Proc. of 29th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1995. pp. 580–585.

[7] Lin, R., A fast 64×64 multiplier without a complete matrix reduction on short shift switching buses, In: *Proc. of 1995 International Conference on High Performance Computing*, Dec. 1995, New Delhi, India.

[8] Lin, R. (1996). A fast reconfigurable-bus-based 32-bit CLA adder scheme with domino logic, *Proc. of the Workshop on Reconfigurable Architectures, the 10th International Parallel Processing Symposium*, Hawaii.

[9] Lin, R. and Olariu, S. (1996). A Reconfigurable Network Architecture For Parallel Prefix Counting, *Proc. of the Workshop on Reconfigurable Architectures, the 10th International Parallel Processing Symposium*, Hawaii.

[10] Lin, R. and Olariu, S. (1995). Reconfigurable buses with shift switching–concepts and applications, In: *IEEE Transactions on Parallel and Distributed Systems*, **6**(1), 93–102.

[11] LSI logic 1.0 micron cell-based products data book, LSI logic corporation, Milpitas, Califoria, 1991.

[12] Motorola CMOS Logic Data book, Motorola Inc., 1991.

[13] Mukherjee, A. (1986). *Introduction to nMOS & CMOS VLSI System Design*, Prentice-Hall, NJ.

[14] Weste, N. and Eshraghian, K. (1993). *PRINCIPLES OF CMOS VLSI DESIGN*, Addison-Wesley Pub. Company.

Authors Biographies

Stephan Olariu received the M.Sc. and Ph.D. degrees in computer science from McGill University, Montreal in 1983 and 1986, respectively. In 1986 he joined the Computer Science Department at Old Dominion University.

Dr. Olariu has published extensively in various journals and conference proceedings. His research interests include parallel algorithms, parallel algorithms, image processing and machine vision, computational graph theory, computational geometry, and mobile computing.

Rong Lin received the B.S. in mathematics from Peking university (China), M.S. in Computer Science from Beijing Polytechnical University (China) and Ph.D. in computer science from Old

Dominion University in 1989. He is currently an associate professor with the Department of Computer Science of SUNY at Geneseo. His current research interests include VLSI arithmetic, parallel architectures and algorithms, digital signal processing, and reconfigurable architectures.