A Non-Binary Parallel Arithmetic Architecture

Rong Lin[1] and James L. Schwing[2]

1. Department of Computer Science, SUNY at Geneseo
   Geneseo, NY 14454 {lin@cs.geneseo.edu}
2. Department of Computer Science, Central Washington University
   Ellensburg, WA 98926 {schwing@cwu.edu}

Abstract. In this paper we present a novel parallel arithmetic architecture using an efficient non-binary logic scheme. We show that by using parallel broadcasting (or domino propagating) state signals, on short reconfigurable buses equipped with a type of switches, called GP (generate-propagate) shift switches, several arithmetic operations can be carried out efficiently. We extend a recently proposed shift switching mechanism by letting the switch array automatically generate a semaphore to indicate the end of each domino process. This reduces the complexity of the architecture and improves the performance significantly.

1 Introduction

Recently proposed shift switching mechanisms [4-9] allows modulo w ($w \geq 2$) arithmetic computations to be efficiently carried out through broadcasting a class of non-binary signals, called state signals, along short buses. The bus structure is closely related to reconfigurable bus systems (REBS, RMESH, Bit Model, RMESH, REBSIS) which have been extensively investigated (refer to [2, 8, 10]). This novel switching technique could lead to VLSI-efficient architectures for high-speed application-specific processor and arithmetic device designs, including parallel comparators, counters, multipliers, and matrix multiplication processors [4-9].

In this paper, we will review the concept of state signals , define (in general) shift switches, and introduce a specific type of shift switches, called the GP (generate and propagate) switches. We will then show that broadcasting state signals on buses equipped with GP shift switches can be used to efficiently construct fast asynchronous adders. In particular, we illustrate a CLA (carry-lookahead) reconfigurable bus-based adder featuring: 1. the combination of state signals, GP switches, and the traditional CMOS domino logic technique on short reconfigurable buses (of width 2 and 3) to achieve a fast VLSI arithmetic design; 2. asynchronous process with a semaphore produced in parallel with the desired arithmetic outputs; 3. full utilization of the inherent speed of each arithmetic opertation; 4. high performance in terms of speed and VLSI area; 5. asynchronous domino process driven by a few control bits; 6. a fast, area compact, stable design and suitable for current CMOS VLSI implementation.

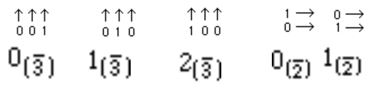
2 The shift switch logic

The shift switch logic is an arithmetic logic which manipulates small integers (as logic operands) represented by state signals by using switc hes (as logic operators) called shift switches to carry out certain basic arithmetic operations (refer to [4-9]). In this section we first introduce the state signals in general and then a GP (generate/propagate) shift switch in particular. A state signal (or S-signal) with integer value I ($0 \leq I \leq w-1$; $w \geq 2$) is represented by bit sequence b_0, b_1, ..b_{w-1} (in this paper we assume the order is either right to left or top to bottom), with the unique bit u (either 0 or 1) in the I-th position. An S-signal is said n- (p-) type denoted by $I_{(\underline{w})}$ ($I_{(\overline{w})}$) if u=0 (1). An S-signal may also be denoted by $I_{(w)}$ or simply I if only the number of bits and/or its value is of interest (Figure 1). It may be convenient to represent an S-signal just by the sequence of its bits (variables). For example, (g1, g0, p) is a 3-bit S-signal, for which g1 is

The work was supported by National Science Foundation under grant MIP-9630870.

carry generate-1 bit, g0 is carry generate-0 bit, and p is carry propagate bit. In this paper we consider only S-signals with small values, say from 0 to 3.

A degenerate state signal, I ($0 \le I \le w-1, w \ge 2$), is a state signal with the 0-th bit removed. Thus it is represented by bit sequence $b_{w-1}, b_2,..,b_1$ and its value is 0 if there is no unique bit in the sequence (i.e. all bits are identical for w>2), otherwise it is equal to the corresponding S-signal (with the 0-th bit added). A degenerate S-signal is denoted by $I_{(\underline{w})}'$ ($I_{(\overline{w})}'$) or $I_{(w)}'$, or simply I (Figure 2), and we may just call it an S-signal if no confusion arises. Again it may be convenient to represent a degenerate S-signal just by the sequence of its bits (variables; e.g. $X_{(\overline{3})}' =(A,B)$). The type of an S-signal with w>2 is implied by its representation, but for w=2 it is not, and it must be indicated explicitly (when we need to identify the unique bit). Often we use $I_{(w)}$ for an arbitrary-value w-bit S-signal.


0 0 1
0 1 0
1 0 0
$0_{(\overline{3})}$ $1_{(\overline{3})}$ $2_{(\overline{3})}$ $0_{(\overline{2})}$ $1_{(\overline{2})}$

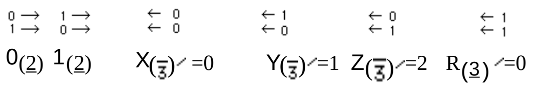

$0_{(\underline{2})}$ $1_{(\underline{2})}$ $X_{(\overline{3})}'=0$ $Y_{(\overline{3})}'=1$ $Z_{(\overline{3})}'=2$ $R_{(\underline{3})}'=0$

Figure 1. Some state signals. **Figure 2**. Degenerate state signals.

Given a function F(X, Y)=U where X and Y and U are (small) S-signals, a pass-transistor (or transmission gate) based digital filter switch which implements function F is called a shift switch with respect to function F. 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).

If a GP switch receives an additional input P/E (precharge/evaluation enable) and produces an additional output V, such that, the functional relations between inputs, P/E, X, Y, and outputs, U and V can be described by Table 1 or 2. The corresponding GP switch is called GP1 or GP2. The block symbols of GP1 and GP2 are shown in Figures 4.a and 4.b. Switch GP2 has a 3-bit input X and a 3-bit output U, while GP1 has a 2-bit input X and a 2-bit output U. The switches are assumed to work on two phases (precharge and evaluation). During the precharge phase (P/E=0), both switches have the same corresponding input/output values (all are 0, represented by appropriate S-signals). During the evaluation phase (P/E=1), both switches produce horizontal output U=GP (X, Y) denoted by S-signals $GP(X,Y)_{(\overline{3})}'$ and $GP(X,Y)_{(\overline{3})}$ respectively, but they produce different vertical outputs for GP1 V= $GP(X,Y)_{(\overline{3})}$; while for GP2, V=$Y_{(\overline{3})}$.

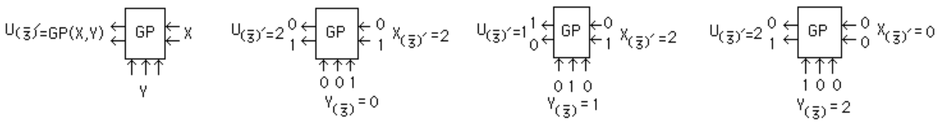
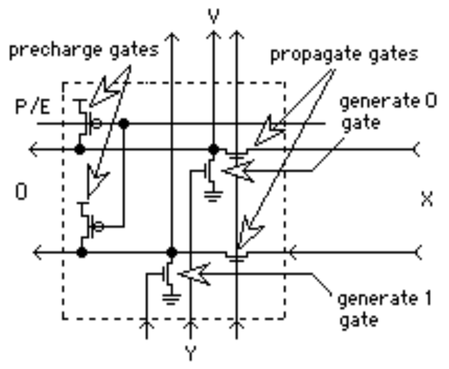


Figure 3. (a) The functionality of a GP shift switch.



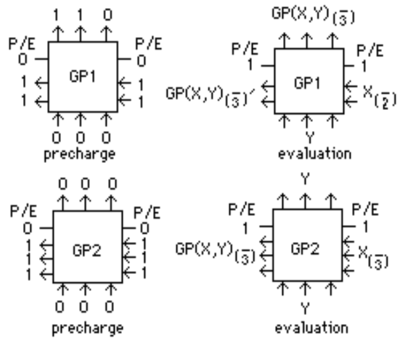


Figure 3.(b) The schematic of the GP1 switch. **Figure 4.** The GP shift switches.

Table 1. The function of GP1 switch

precharge/ evaluation (P/E)	input		output	
	X	Y	U	V
0 (precharge)	$0_{(\underline{3})}'$ or 11	$0_{(\bar{4})}'$ 000	$0_{(\underline{3})}'$ 11	$0_{(\underline{3})}$ 110
1 (evaluation)	$X_{(\bar{3})}'$	$Y_{(\bar{3})}$	$GP(X,Y)_{(\bar{3})}'$	$GP(X,Y)_{(\bar{3})}$

Table 2. The function of GP2 switch

precharge/ evaluation (P/E)	input		output	
	X	Y	U	V
0 (precharge)	$0_{(\underline{4})}'$ or 111	$0_{(\bar{4})}'$ 000	$0_{(\underline{4})}'$ 111	$0_{(\bar{4})}'$ 000
1 (evaluation)	$X_{(\bar{3})}$	$Y_{(\bar{3})}$	$GP(X,Y)_{(\bar{3})}$	$Y_{(\bar{3})}$

3 The small shift switch adder architecture

We first define precharged carry and sum domino logic units, based on the traditional designs [1, 3, 12] and then use them as local components to construct a shift switch (7-bit) adder.

Figures 5, 6 and 7 show block symbols for carry unit 0, carry unit j and sum unit j respectively (for 1≤j≤31, note that the units will also be used for large adders). The computations for carries and sums consist of two phases: precharge and evaluation. The logic results are produced during the evaluation phase. During the precharge phase the control signal P/E is set to 0, while $\overline{P/E}$ is set to 1. It is easy to verify that (1) the carry unit 0 are precharged as follows: $g1_j$ (carry generate-1), $g0_j$ (carry generate-0), and p_j (carry propagate) are all high (Figure 5.a); the outputs of carry unit j are all low (Figure 6.a); and the outputs of all sum units are all high (Figure 7.a). When inputs are received P/E is set to 1, the evaluation phase begins. The logic units are discharged as follows: For carry units j: $g1_j$ ($g0_j$) is discharged to high if a carry of 1 (0) is generated; p_j is discharged to high if a carry-propagate is produced, i.e. a_j and b_j have different values since $g1_j = a_j \cdot b_j$; $g0_j = \overline{a_j + b_j}$; and $p_j = a_j \oplus b_j$. Note that S-signal $cb(j)_{(\bar{3})}$ =(g1, g0, p) is called the j-th carry-bit S-signal for 1≤j≤31. For carry unit 0: c_0 is discharged to high if both a_0 and b_0 are 1, and to low otherwise, while S_0 is discharged to low if a_0 and b_0 have the same value. For sum unit j (1≤j≤31): S_j is discharged to low if p_j and c_{j-1} have the same value, since $S_j = p_j \oplus c_{j-1}$ (note that c_j denotes the j-th carry-out).

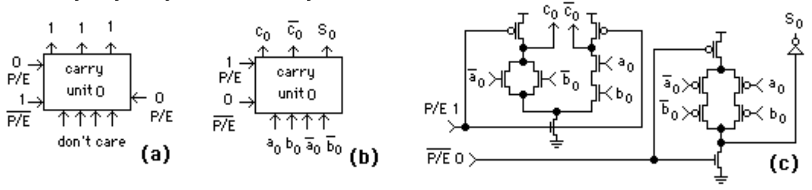


Figure 5. Carry unit 0 (a) Precharge phase; (b) Evaluation phase; (c) the schematic.

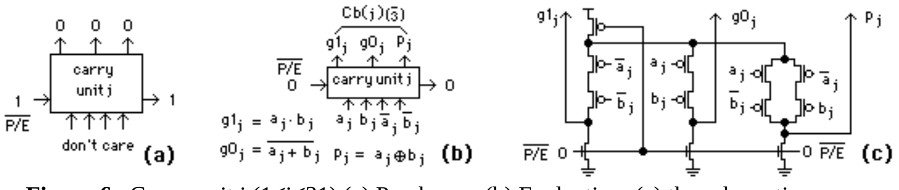
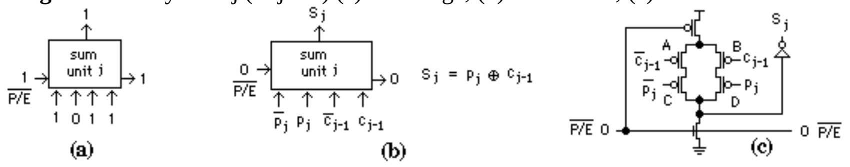


Figure 6. Carry unit j (1≤j≤31) (a) Precharge; (b) Evaluation; (c) the schematic.



sum unit j
1 0 1 1
(a)
$\bar{p}_j$ p_j $\bar{c}_{j-1}$ c_{j-1}
$S_j = p_j \oplus c_{j-1}$
(b)
A B C D
(c)

Figure 7. Sum unit j (1≤j≤31) (a) Precharge; (b) Evaluation; (c) the schematic.

Figure8 shows a 7-bit adder, called block A(0). It consists of six GP1 switches connected with seven carry units and six sum units. When signal P/E is set to 0, all GP1 switches, carry units and sum units are precharged as described above.

Next let us discuss the evaluation phase: First, the discharge of each carry unit j generates a carry-bit S-signal $cb(j)_{(\bar{3})}=(g1_j, g0_j, p_j)$, which then discharges the corresponding GP1 switch immediately if $cb(j)_{(\bar{3})}\neq 0$, otherwise it will open the propagation gates of the GP1 switch for the carry propagation. The bus (with the switches) is then partitioned (by the propagation gates). In the worst case carry propagation (discharging) will start from port A or B passing through all 6 pass transistors on the bus. Next, the top outputs of each GP1 (except the last one) go to gate the sum units. Finally the discharge occurs on each sum unit in parallel and simply produces each sum bit of the adder. The worst case delay of the adder is T_c (discharge a carry unit) + $7T_{GP}$ (tdischarge all seven cascaded pass-transistors on the bus) + T_s (discharge a sum unit).

4 The larger shift switch adder architecture

A larger adder can be constructed by several blocks that contain GP switches organized in three levels. We now illustrate a 32-bit adder (refer to Figure 9). The first level consists of six blocks: blockA(0), the 7-bit adder (Figure 8), blockA(i) (for $1\leq i\leq 4$) (Figures 10) and blockA(5) (Figure 11). The second level is a single block, block B (Figure 12). The third level consists of five blocks: blockC(i) (for $1\leq i\leq 4$, Figure 13) and blockC(5) (Figure 14).

BlockA(i) (for $1\leq i\leq 4$,) contains five cascaded GP2s and carry units. The switches directly send the input carry-bit S-signals $cb(7+j)_{(\bar{3})}=(g1_j, g0_j, p_j)$ (for j=5(i-1)) to the sum units and produce an output S-signal at the left end of the bus. In other words, in the vertical direction the bus simply produces carry-bit S-signals for each bit; in horizontal direction, it produces signal cg(i) =(gg1, gg0, gp) which is called (the i-th) group-carry S-signal, and gp=1 if all p_j of the group are 1, otherwise gp=0 and gg1=1 (gg0=1) if the sum of the group's two bit segments definitely generates a carry 1 (0). BlockA(5) produces only carry-bit signal for each bit.

The worst case delay of the first level (except blockA(0)) can be specified as T_c (discharge a carry unit) + $7T_{GP}$(discharge seven cascaded pass-transistors) + T_{inv} (inverter delay, low to high), i.e. $T_c + 7T_{GP} + T_{inv}$. The best case delay can specified as: $T_c + T_{GP} + T_{inv}$.

The second level bus is block B. It is similar to the cascaded (four) switches of blockA(0), except as follows: (1) its four vertical inputs are group-carry S-signals from blockA(i) (for $1\leq i\leq 4$) and its horizontal input is a 2-bit S-signal of actual carry C_6; (2) its four outputs are 2-bit S-signals for C_{11}, C_{16}, C_{21}, C_{26}; (3) it also produces a semaphore to indicate whether the discharge on the bus (i.e. on all switches) is completed (when semaphore becomes 0) or not. This is a unique feature provided by this scheme. More significantly, for the best case (or average case) of the adder inputs the domino discharge involves only no more than about 3 cascaded pass-transistors. This implies that an asynchronous scheme can be adapted to reduce average case delay of an adder.

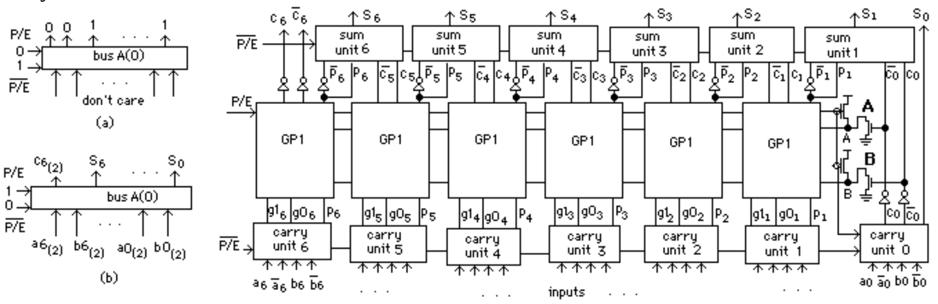


Figure 8. Block A(0), a 7-bit adder, (a) Precharge phase (b) Evaluation phase.

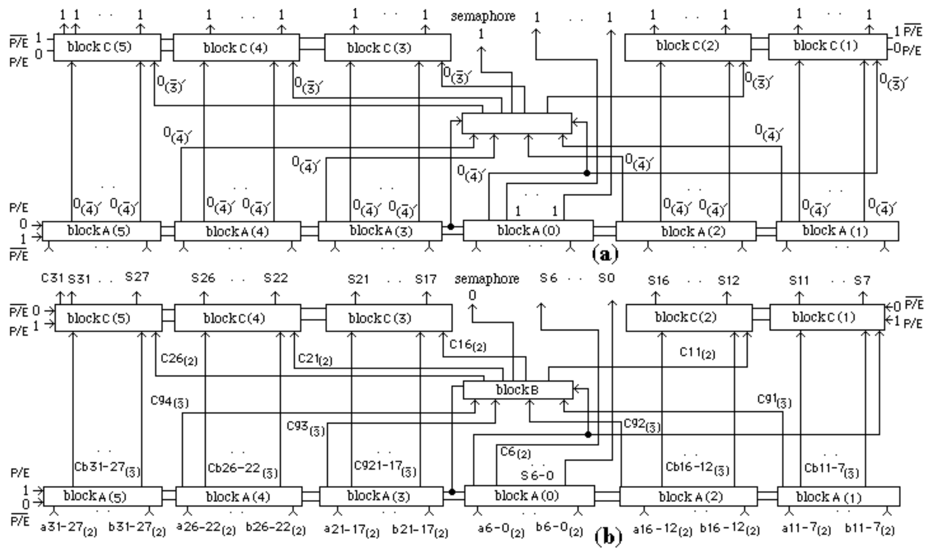


Figure 9. The 32-bit asynchronous adder scheme: (a) precharge; (b) evaluation.

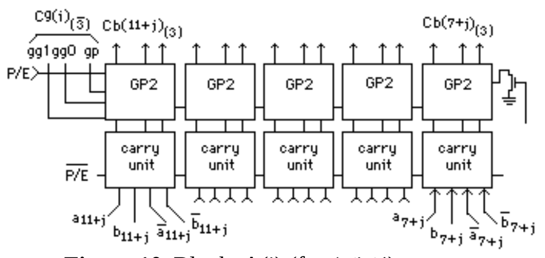


Figure 10. Block A(i) (for 1≤i≤4).

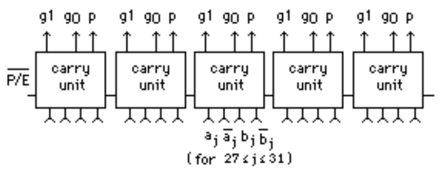


Figure 11. Block A(5).

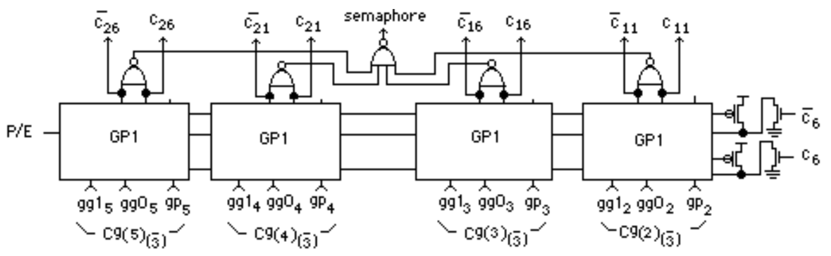


Figure 12. Block B.

The worst case delay of the second level (excluding the generation of semaphore) can be specified as: $5T_{GP}$ (time to discharge five cascaded pass-transistors) + T_{inv} (an inverter delay, low to high). The best case delay can be specified as: $T_{GP} + T_{inv}$.

The third level bus is blockC(i) (for 1≤i≤5). It is similar to the cascaded (four) switches with sum units in blockA(0), except that during evaluation phase the actual group carry-in (horizontal input) is available now and is represented by S-signal (C_{6+j}, $\overline{C}_{6+j}$) for j=5(i-1) and 1≤i≤4. The worst case delay of the third level can be specified as: $6T_{GP}$ (discharging six cascaded pass-transistors) + T_S (discharging a sum unit), i.e. $6T_{GP}+T_S$. The best case of the first level delay +

the best case second level delay + the worst case third level delay is.($T_C + T_{GP} + T_{inv}$)+ ($T_{GP} + T_{inv}$) + ($6T_{GP} + T_S$.) = $T_C + 8T_{GP} + 2T_{inv} + T_S$. By contrast, if the adder is a synchronous adder, the adder delay is estimated as: $T_C + 18T_{GP} + 2T_{inv} + T_S$. This indicates the adder will run an expected 50% faster for best case inputs, or 30% faster for average case inputs.

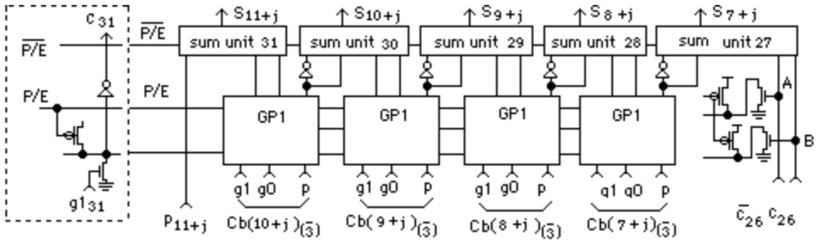


Figure 13. Block C(i) for 1≤i≤5 (note that the dotted box is attached only for i=5).

5 Concluding remarks

A novel non-binary arithmetic architecture using the combination of state signals [4-10], GP switches and traditional CMOS domino logic techniques [1, 3] on short reconfigurable buses has been presented. The design produces a semaphore as a by-product to indicate the end of domino discharge process, thus it is an asynchronous domino logic scheme. The significance of the design is that for a large percentage of input cases (about 80%), the computation in the first two of a total three stages can be done much faster than a traditional synchronous adder, which can lead to a computation about 30% faster on average by a program simulation. The scheme can be easily extended for larger adders, for example 64-bit or 128-bit adders. It can also be applied to construct array multipliers and other related arithmetic designs.

References

1. I. S. Hwang, and A. L.. Fischer, Ultrafast compact 32-bit CMOS adders in multi-output domino logic, IEEE J. Solid-State Circ., 24, No 2 April 1989, pp. 358-369.
2. J. Jang, H. Park and V. K. Prasanna, A bit model of the reconfigurable mesh, *Proc. of the Workshop on Reconfigurable Architectures*, the 8th IPPS, Cancun, Maxico, 1994.
3. R. H. Krambeck, C. M. Lee, and H.S. Law, High-Speed Compact Circuits with CMOS, *IEEE Journal of Solid-State Circuits,* Vol SC-17, No. 3, June, 1982.
4. R. Lin, Shift switching and novel arithmetic schemes, in *Proc. of 29th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1995.
5. R. Lin, A Reconfigurable Low-power High performance Matrix Multiplier Design, *Proc. of 1th Intl. Symp. on Quality of Electronic Design,* San Jose, California. March 2000.
6. R. Lin, Parallel VLSI Shift Switch Logic Devices, US Patent, Serial No. 09/022,248, 1999.
7. R. Lin and S. Olariu, Reconfigurable shift switching parallel comparators, in *Intl. Journal of VLSI Design*, March, 1999.
8. R. Lin and S. Olariu, Efficient VLSI architecture for Columnsort, *IEEE Transactions on Very Large Scale Integration (VLSI) systems,* VOL. 7, No. 1. March, 1999.
9. R. Lin, and S. Olariu, Reconfigurable buses with shift switching --concept and architectures, in *IEEE Trans. on Parallel And Distributed System.* January 1995.
10. R. Lin, K. Nakano, S. Olariu, M.C. Pintoti, J.L. Schwing, A.Y. Zomaya Scalable hardware-algorithms for binary prefix sums, in Proc. of 6th International Workshop on Reconfigurable Architecture(RAW),1999.
11. R. Miller, V. K. P. Kumar, D. Reisis, and Q. F. Stout, Parallel Computations on Reconfigurable Meshes, *IEEE Transactions on Computers*, 42 (1993), 678--692.
12. N. Weste and K. Eshraghian, *PRINCIPLES OF CMOS VLSI DESIGN, A systems Perspective* (Second Edition), A,ddison-Wesley Publishing Company, 1993.