

A Regularly Structured Parallel Multiplier with Low-power Non-binary-logic Counter Circuits*

RONG LIN[†]

Department of Computer Science, SUNY at Geneseo, Geneseo, NY 14454

(Received 20 June 2000; In final form 3 August 2000)

A highly regular parallel multiplier architecture along with the novel low-power, high-performance CMOS implementation circuits is presented. The superiority is achieved through utilizing a unique scheme for recursive decomposition of partial product matrices and a recently proposed non-binary arithmetic logic as well as the complementary shift switch logic circuits.

The proposed 64×64 -b parallel multiplier possesses the following distinct features: (1) generating 64×8 -b partial product matrices instead of a single large one; (2) comprising only four stages of bit reductions: first, by 8×8 -b small parallel multipliers, then, by small parallel counters in each of the remaining three stages. A family of shift switch parallel counters, including non-binary $(6, 3)^*$ and complementary $(k, 2)$ for $2 \leq k \leq 8$, are proposed for the efficient bit reductions; (3) using a simple final adder.

The non-binary logic operates 4-bit state signals (representing integers ranging from 0 to 3), where no more than half of the signal bits are subject to value-change at any logic stage. This and others including minimum transistor counts, fewer inverters, and low-leakage logic structure, significantly reduce circuit power dissipation.

Keywords: Low-power high-performance VLSI design; Regularly structured parallel multiplier; Partial product matrix reduction; CMOS pass-transistor circuit; Parallel counter circuits

1. INTRODUCTION

The traditional designs of parallel (array) multipliers [1, 3–6, 17] mainly rely on the use of fast $(3, 2)$ and $(4, 2)$ parallel counter circuits for high speed. However, the traditional approaches have the following problems which hinder achieving a general VLSI high performance in the design of

larger-size (say 64×64 -b) high-speed multipliers: (1) the design irregularity inherited from the bit reduction of a large partial product matrix (even with Booth recoding); (2) the load/wire unbalance caused by the unbalanced column heights of the partial product matrices generated in many (5 to 10) reduction stages; (3) quite a large power dissipation.

*The work was supported, in part, by National Science Foundation under grant CCR-0073469.

[†]e-mail: lin@cs.geneseo.edu

In this paper we propose a highly regular parallel multiplier design based on recently proposed unique decomposition approach for partial product matrix reductions [10]. The proposed 64×64 -b parallel multiplier shows the following distinct features: (1) Distributing input bits to 64 locations using a full 4-branch tree structure, then at each location generating an 8×8 -b partial product matrix, instead of a single large one as commonly adopted by the existing designs (including those with Booth recoding). (2) Comprising only four stages of bit reductions (each corresponding to a sub-multiplication module): First, by 64 identical 8×8 -b small parallel multipliers. Second, by 16 identical arrays of (6, 2) shift switch parallel counters. And for the remaining two stages, by 4 and 1 identical arrays of the same counters. Note, a parallel counter here reduces no more than 6 input bits into 2 output bits in a column and acts as a carry-save addition unit. (3) Using a final adder significantly simpler than a traditional large final adder. The input bits of the final adder have the following simple form: one bit per column for columns 1 to k and $128-k'$ to 128, with values of k and k' in between 12 to 20 determined by detailed design; two bits per column for columns $k+1$ to $127-k'$ (refer to Fig. 5b).

All (three) inter-stage connections of the bit reduction circuits are regular and symmetrical, with the longest wire connection (between the third and the last modules) not exceeding that in traditional designs. The minimal connection delays can also be achieved by the utilization of early signals and well balanced load/wire of the regularly structured network, where each bit reduction module is associated with exactly a sub-tree of a full 4-branch input-bit tree, thus further simplifies the circuits.

Though the novel multiplier may be implemented using any existing small (say 8×8 -b) multipliers and small parallel counters (say traditional half-full adders and (4, 2) counters [4, 5]), a family of shift switch counters and variants, including non-binary 4-bit signal based (6, 3)* and complementary (k, 2), $2 \leq k \leq 8$ counters (both will be defined shortly below), are adopted to achieve low

power dissipation, while keeping high VLSI performance in speed and area. The recently proposed shift switch logic circuits [8–12] are used to perform modulo arithmetic operations, with 4-bit and 2-bit state signals as operands and small shift switch parallel counters as operators. The new approach with the novel circuits could overcome the drawbacks of the traditional designs while achieving high performance in VLSI design.

A (n, 3) or (n, 3)* non-binary shift switch counter usually adds n input bits, resulting in a sum bit of weight 1, a sum bit of weight 2, and a carry bit of weight 4. It is done by converting three binary bits into a 4-bit state signal (with a value ranging 0 to 3), then processing the state signal using an R-circuit which produces two sum bits, s_0 and s_1 , and a q-circuit which produces the carry bit q (for details refer to [9, 12], also see Fig. 7). Note that a (n, 3) receives all n input bits of weight 1, while (n, 3)* receives $n-1$ input bits of weight 1 and one input bit of weight 2. Two typical (6, 3)* parallel counters are shown in Figure 7. Each takes 5 input bits, i_1, i_2, i_3, a, b , of weight 1 and another input bit c' of weight 2, produces three output bits, s_0, s_1 and q (and, perhaps, their complements), of weights 1, 2 and 4 respectively. More precisely, a (6, 3)* parallel counter implements the following two arithmetic equations:

$$X = i_1 + i_2 + i_3; \quad (\text{A})$$

$$X + a + b + 2c' = s_0 + 2s_1 + 4q \quad (\text{B})$$

here X is a 4-bit state signal.

A (n, 2) complementary shift switch counter adds n bits of the same weight 1 through processing complementary binary bits (or 2-bit state signal, referring to Fig. 9), resulting in a sum bit and a carry bit. For example, the following binary arithmetic equation holds for (4, 2) counter of Figure 9c:

$$i_1 + i_2 + i_3 + i_4 + Cin = s + 2c + 2Cout \quad (\text{C})$$

Both types of shift switch counters may receive and produce intermediate carry bits. To reduce the same number of input bits, a complementary

counter requires significantly more intermediate carry-bits than a non-binary one. For example, the (6, 2) and (8, 2) complementary counters require 3 and 5 intermediate carry-in/carry-out pairs respectively, while each (6, 3)* requires no intermediate carry-in/carry-out bits.

The interesting low-power features of the proposed non-binary arithmetic logic include: (1) No more than half of the signal bits in the arithmetic circuits are subject to value-change at any logic stage. (2) Non-full-swing p-type 4-bit signal level restoration dissipates less power than the complementary pass transistor counterparts. The restoration is applied for major shift switch parallel counters such as (6, 3)* (refer to [8, 9, 11–13]). It is done by a circuit, called p-type restorer (refer to dotted boxes of Fig. 7), which seems slow, but actually improves the overall circuit speed because it simultaneously realizes several logic functions including converting 4-bit state signal into binary output bits of carry and sums. (3) The proposed parallel counter circuits possess another unique characteristic: 3 out of 4 signal bits, which propagate through pass transistors, are 0s (refer to Fig. 7). This could lead to significant reduction in leakage power dissipation due to less possible leakage current generated over the circuit area.

The SPICE simulations and preliminary tests of the multiplier component circuits have demonstrated the superiority of the new design. The delay and power comparisons are based on SPICE circuit simulation using a 0.25-micron process with a 2.5-V supply. The simulation has shown that without counting the final addition, a total delay of 4 ns for the proposed 64 × 64 multiplier can be achieved, and a significant reduction in power dissipation, compared with the traditional (3, 2)–(4, 2) based counterpart designs, can also be achieved.

2. DECOMPOSITION OF PARTIAL PRODUCT MATRICES

A novel approach of decomposing a partial product matrix, called square recursive decomposition, has

recently been proposed in [10, 9]. In this section we illustrate the application of this approach for parallel multipliers, which could lead to superior regularity and modularity in multiplier design without sacrificing VLSI circuit performance in speed and area. Figure 1a illustrates a 4 × 4-b partial product matrix which is generated by two 4-bit numbers X and Y on a matrix of AND gates. The product of X and Y is generated by adding all weighted partial product bits along the diagonal directions. Each bit of the final sum, or the product, is then indicated by a small circle, and the carry bit by a marked circle (Fig. 1b). We first show below how to use four such multipliers to compute a (virtual) product of two 8-bit numbers.

Figures 1c and 1d show four 4 × 4-b multipliers resulted from decomposing an 8 × 8-b partial product matrix, where the data from two input numbers X and Y are duplicated and sent to the 4 × 4-b matrices. The weighted bits of the four products of the four multipliers are added by (3, 2) counters in parallel to result in two numbers (note that the two numbers are not added until the final stage) as the virtual product of the 8 × 8 multiplier (Fig. 1d).

To simplify the summation as illustrated in Figure 1d, we re-position the multipliers by exchanging locations of the left-upper and left-lower multipliers, *i.e.*, C and D. With the modification, as shown in Figure 2, the circuit diagram of a virtual 8 × 8 multiplier becomes regular, symmetrical, and simpler in layout. The order of four multipliers A, B, C and D shown in Figure 2 represents a useful order that we call square order. It is also easy to verify that the process implements the right part of the following algebraic equation:

$$\sum X_i Y_j 2^{i+j} = \sum_{0 \leq m, n \leq 1} \sum_{\substack{4m \leq i \leq 3+4m \\ 4n \leq j \leq 3+4n}} X_i Y_j 2^{i+j}$$

$$\left(= \sum_{0 \leq K \leq 15} S_K 2^K + \sum_{5 \leq K \leq 12} S'_K 2^K = S + S' \right) \quad (D)$$

here X and Y are two 8-bit input numbers and $X = X_7.. X_i.. X_0$ and $Y = Y_7.. Y_j.. Y_0$. Therefore the result (unadded S plus S') is the (virtual) product

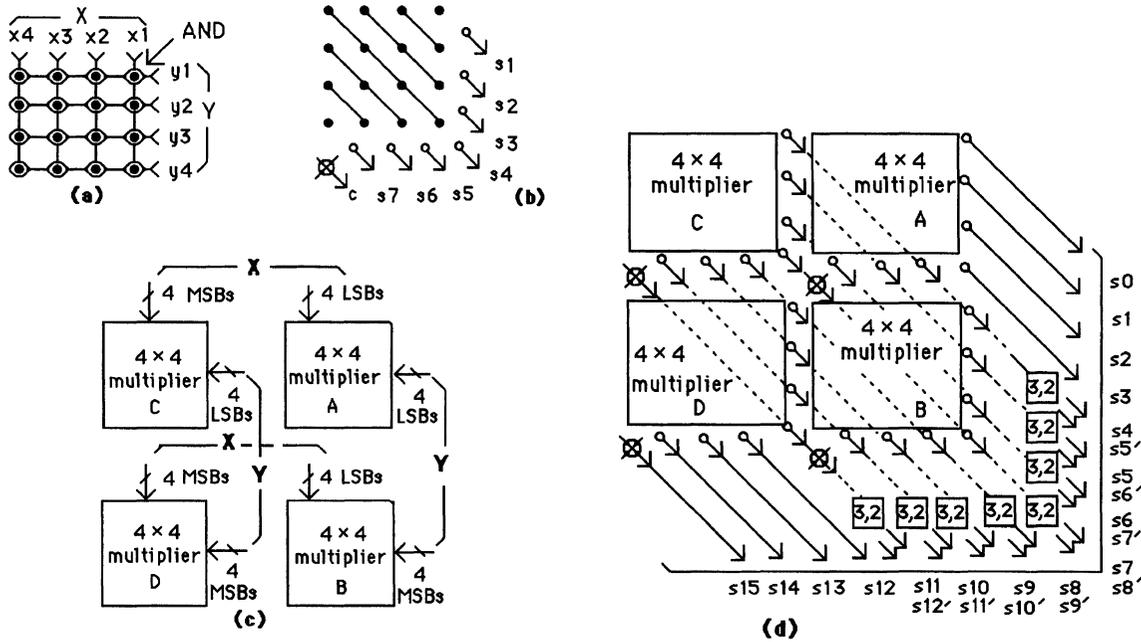


FIGURE 1 (a) The partial product matrix generated by two 4-bit numbers X and Y; (b) The partial products are added; (c, d) 8 × 8-b virtual multiplier constructed by four 4 × 4-b multipliers.

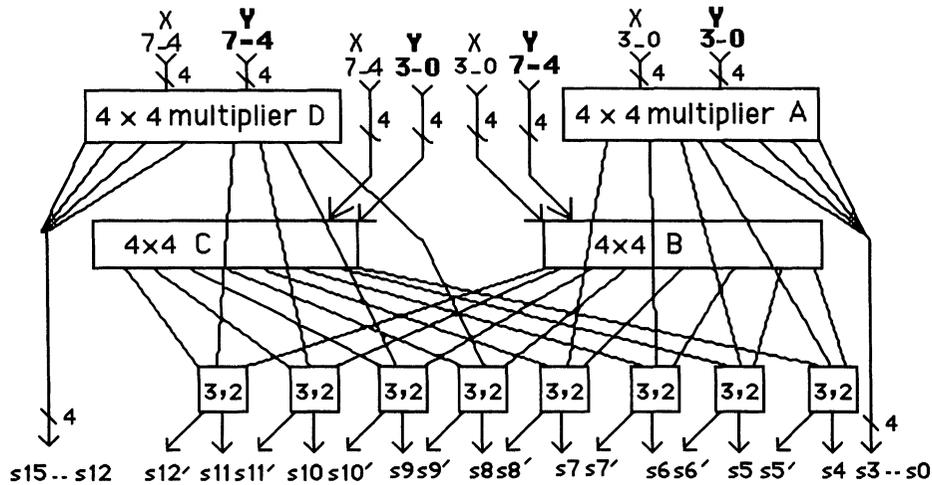


FIGURE 2 The 8 × 8-b virtual multiplier.

of X and Y (as defined by the left part of the equation).

We apply the re-positioning recursively onto a larger partial product matrix as shown in Figure 3. In Figure 3a the original partial product matrix A'' , produced by two 16-b numbers X (plain) and

Y (bold), is decomposed into two levels of square sub-matrices. In Figure 3b the sub-matrices are re-positioned suitable for the constructions of four 8 × 8-b and one 16 × 16-b multipliers based on the square order approach as shown in Figure 2. The structure of input bit distributions to the

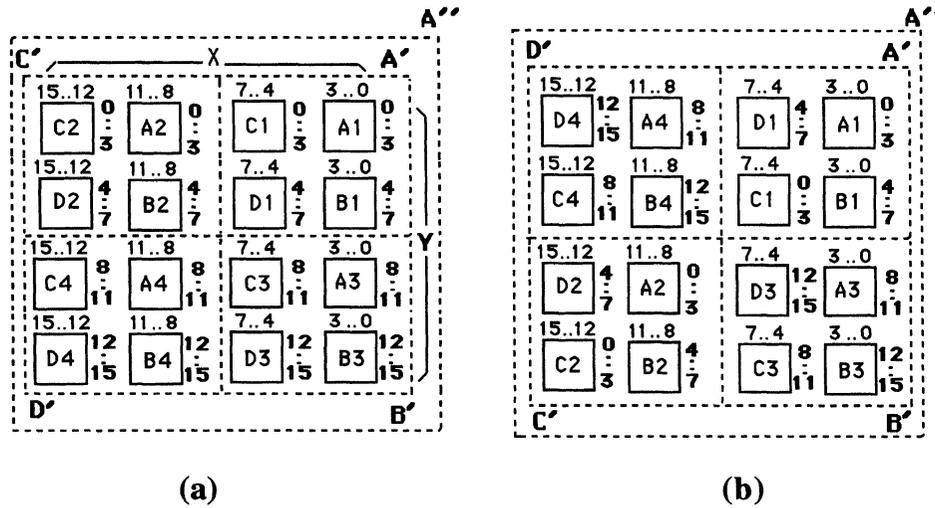


FIGURE 3 Recursive partial product matrix decomposition; (a) before re-positioning; (b) after re-positioning.

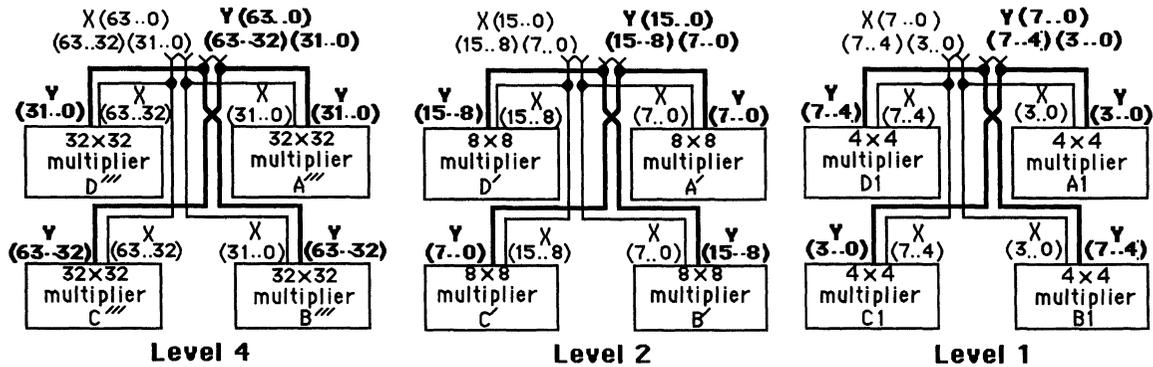


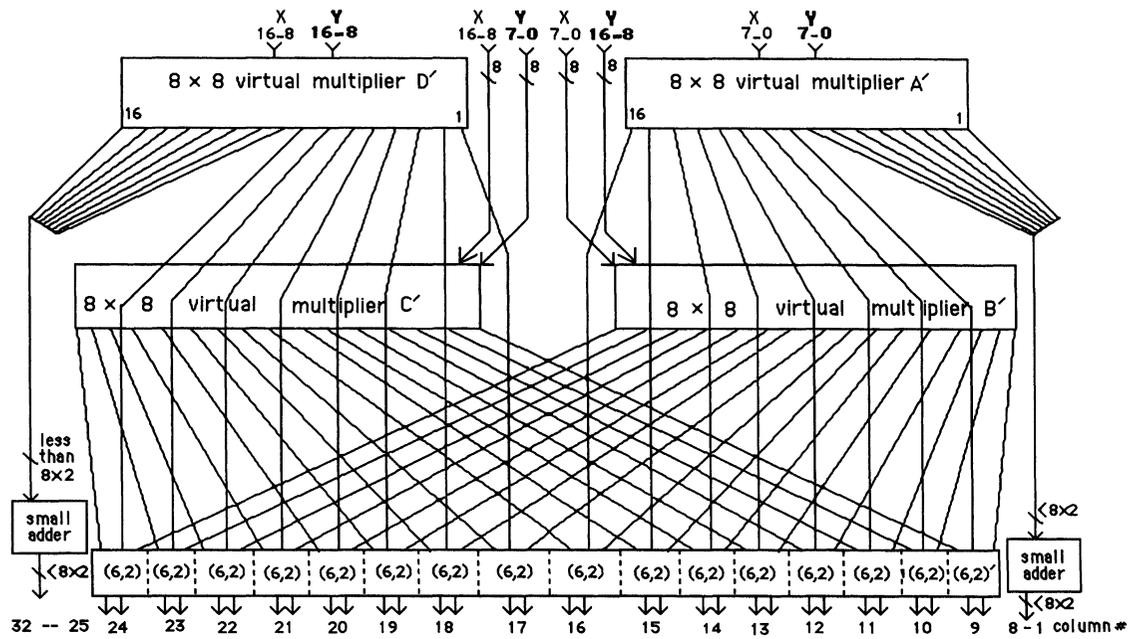
FIGURE 4 The full-4-branch complete binary tree of inputs (3 levels are shown).

sub-matrices of the decomposed partial product matrix is a full 4-branch tree of 2 levels with better load/wire balance compared to traditional approaches. Figure 4 illustrates the full 4-branch tree distribution of two 64-bit inputs X and Y to the partial product matrices in 4-levels (levels 1, 2 and 4 are shown).

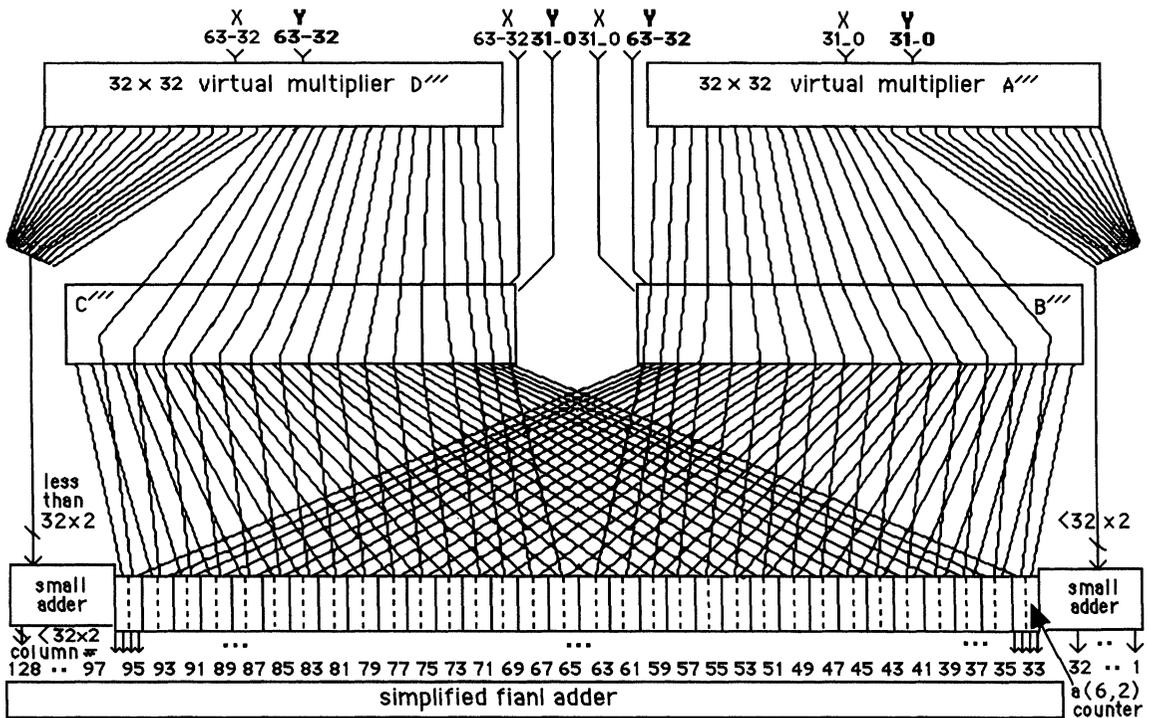
3. THE MULTIPLIER ARCHITECTURE

An overall 64×64 -b recursive multiplier architecture now can be depicted through the following

descriptions for all levels of its components: (1) Partial product generation networks. Instead of using a single large bit matrix commonly adopted by the traditional designs (64×64 -b, or about a half of size when Booth recoding [1] is applied), we generate 64 small identical 8×8 -b partial product matrices in the re-positioned form (note that not generating the 4×4 -b multipliers as exemplified in the last section). (2) The 64 identical 8×8 -b virtual multipliers. Each of them (also called a module-1) produces virtually 16-b products, or more precisely, 16 bits plus $16-k-k'$ extra bits, one per column from columns $k+1$ to $16-k'$ for k, k'



(a)



(b)

FIGURE 5 The virtual multipliers: (a) 16 × 16-b; and (b) 64 × 64-b. Note that each line represents up to 2 and 4 bits in (a) and (b) respectively, except the inputs and the outputs of the virtual multipliers where one line represents one bit.

ranging from 3 to 5 as shown in Figure 6 in the next section. (3) The 16 identical arrays. Each is called a module-2 (refer to Fig. 5a) and composed of 16 same-type parallel counters plus two (3, 2)–(4, 2) based small adders, each adding bits in about 4 columns (depending on the delay of the module) in the lower and higher 4 column positions, such as columns 6 to 8 and columns 25 to 28 of Figure 5a, where no more than 4 input bits are received in each column. The module produces virtually 32-b products, or more precisely, 32 bits plus 32-k-k' extra bits, one per column from columns k+1 to 32-k' for k, k' ranging from 6 to 10. Each counter in columns 9 to 24 receives no more than 6 bits with the inputs in a regular form, and produces two output bits (a sum and a carry) by a parallel counter. The proposed circuits of the parallel counters are illustrated in Section 4. (4) The 4 identical arrays, each called a module-3 and composed of 32 same-type parallel counters, plus again two small adders, each adding bits in about 4 columns. The module produces virtually 64-b products, or 64 bits plus 64-k-k' extra bits, one bit per column from columns k+1 to 64-k' for k, k' ranging from 6 to 10. Each counter in columns 17 to 48 receives no more than 6 bits, and produces two output bits in parallel. (5) The 64 same-type parallel counters, plus again two small adders, producing virtually 128-b products, or 128 bits plus 128-k-k' extra bits, one bit per column from columns k+1 to 128-k' for k, k' = 12 to 20. Each counter in columns 33 to 96 receives no more

than 6 bits and produces two output bits in parallel. (6) A simplified final adder. Clearly, the input bits of the final adder now have the following form which is simpler than the traditional schemes: one bit per column from columns 1 to k and 128-k' to 128, for value k, k' = 12 to 20, determined by detailed designs, and two bits per column for columns k+1 to 127-k' (refer to Fig. 5b).

**4. THE COMPONENT CIRCUITS:
8 × 8 VIRTUAL MULTIPLIER
AND PARALLEL COUNTERS**

Though any existing parallel counters such as half-full adders, (4,2) and (7,3) counters of [3–6, 16–19], may be used to implement the novel multipliers described above, in this section we propose several new CMOS shift switch circuits for the implementation, aiming at low power and high VLSI performance. We focus on two basic components of the scheme: an 8 × 8 virtual multiplier and a parallel counter which receives 6 bits and reduces them into 2 bits. In this paper, we do not involve the specific implementations of the final addition and two small adders (adding about 4-bit numbers) in each module.

We first show two schemes for an 8 × 8 virtual multiplier. Figure 6 illustrates the block diagram of the proposed 8 × 8 virtual multiplier, which reduces all partial product bits into two numbers

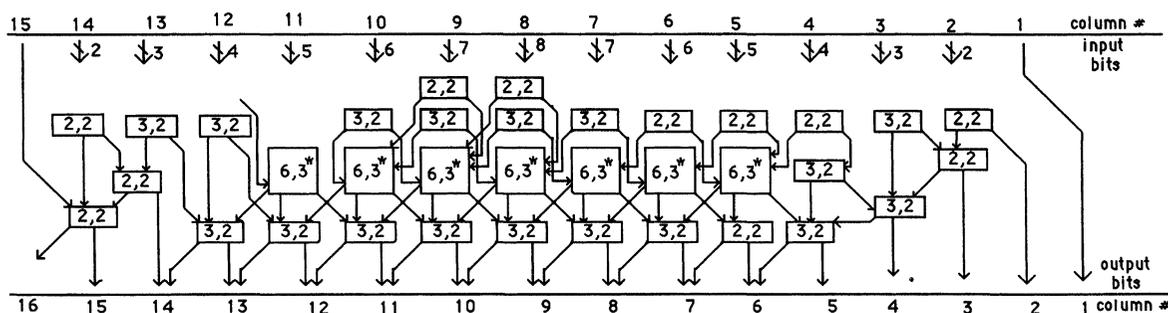


FIGURE 6 The non-binary logic implementation of the 8 × 8 virtual multiplier (critical columns).

(S and S'). In the diagram, each of the columns from 5 to 11 where clearly the critical paths are within consists of a 4-bit state signal based non-binary parallel counter, designated as (6, 3)*, (refer to Figs. 7a and 7b), plus a couple of (3, 2) and/or (2, 2) shift switch complementary counters. The 7 input bits in column 7 are distributed as follows: 3 to the top (3, 2) counter, 4 to a (6, 3)* counter received by ports i_1 , i_2 , i_3 and a . Two remaining ports, *i.e.*, b and c' , receive output bits from a (2, 2) in column 6 and from a (3, 2) in column 8 respectively. The 8 input bits in column 8 are distributed as follows: 5 to the top (3, 2) and (2, 2) counters, 3 to a (6, 3)* counter as its i_1 , i_2 , i_3 . Each of the ports a , b and c' receives an output bit from the (2, 2) of column 8, the (3, 2) of column 7, and the (3, 2) of column 9 respectively. It is also easy to verify the input bit to port c' of the (6, 3)* counter has a weight 2 and all output bits are routed to the columns with correct weights.

Figures 7a and 7b are two typical (6, 3)* parallel counters. The sum of three input bits $i_1 i_2 i_3$ is first converted to a state signal X (represented by x_0 , x_1 , x_2 , x_3). The converter circuit is defined as the part of circuit left to dotted line L1. The state signal is then processed by R-circuit and q-circuit to yield two sum bits s_0 , s_1 and carry bit q respectively. The R circuit of Figure 7a can be roughly defined as the part right to dotted line L1 but excluding the area between dotted lines L2 and L3. For Figure 7b, it is under the dotted line L except the converter. The q-circuit of Figure 7a can be defined as the part in between dotted lines L2 and L3, while for Figure 7b it is the part above the dotted line L. It is straightforward to verify that the converter circuits, the R-circuits, and the q-circuits all together have implemented Eqs. (A) and (B) of Section 1. Here double-rail output bits s_0 and q are produced in (6, 3)* counter of Figure 7b, not the one in Figure 7a. Note that other forms of 4-bit shift switch parallel counters may be obtained through slight modification of the two proposed circuits for some other specific purposes (refer to [9–12]).

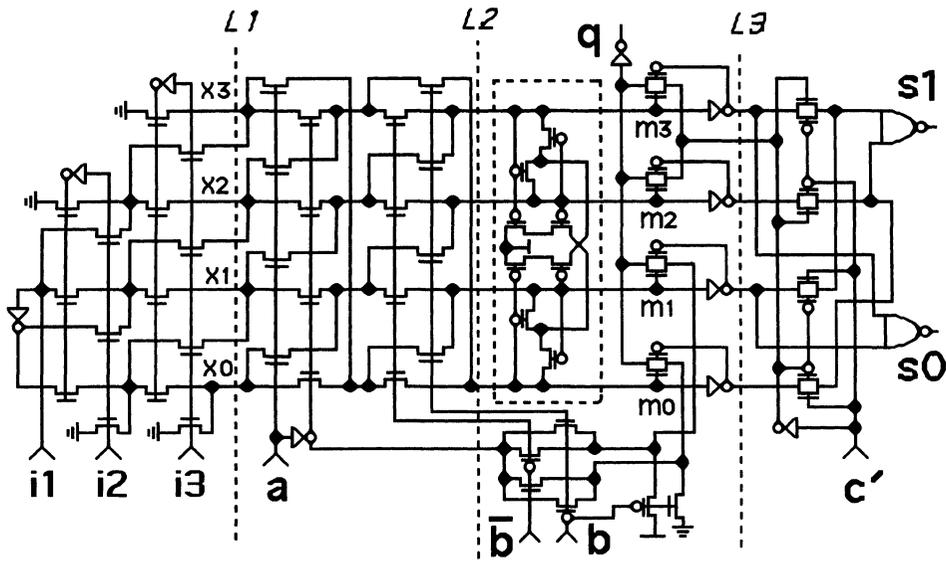
Figure 8 illustrates the critical paths and surrounding area of an alternative 8×8 multiplier design using, instead of (6, 3) non-binary parallel counters, (k, 2) complementary shift switch parallel counters and their direct variants for k in between 2 to 8. Figure 9 shows (k, 2) complementary counters for $k = 3, 4$ and 6. The (6, 2) parallel counter of Figure 9d includes the following: two complementary signal propagation paths, *i.e.*, the double-rail paths from i_1 to S and from i_4 to S; the inputs for switch controls, *i.e.*, i_2 , i_3 , i_5 and i_6 ; and the cross lines for intermediate carry bits, *i.e.*, $Cin1$, $Cin2$, $Cin3$ and $Cout1$, $Cout2$, $Cout3$. A (8, 2) counter can be obtained by modifying the (6, 2) counter and adding two (3, 2) counters in a straightforward manner.

The major circuit, for all modules 2, 3 and 4, is the parallel counter which reduces 6 (or fewer) bits to 2 bits. Clearly, the (6, 2) counter of Figure 9d and its variants can be directly used for the purpose. The following equation holds for the (6, 2) parallel counter:

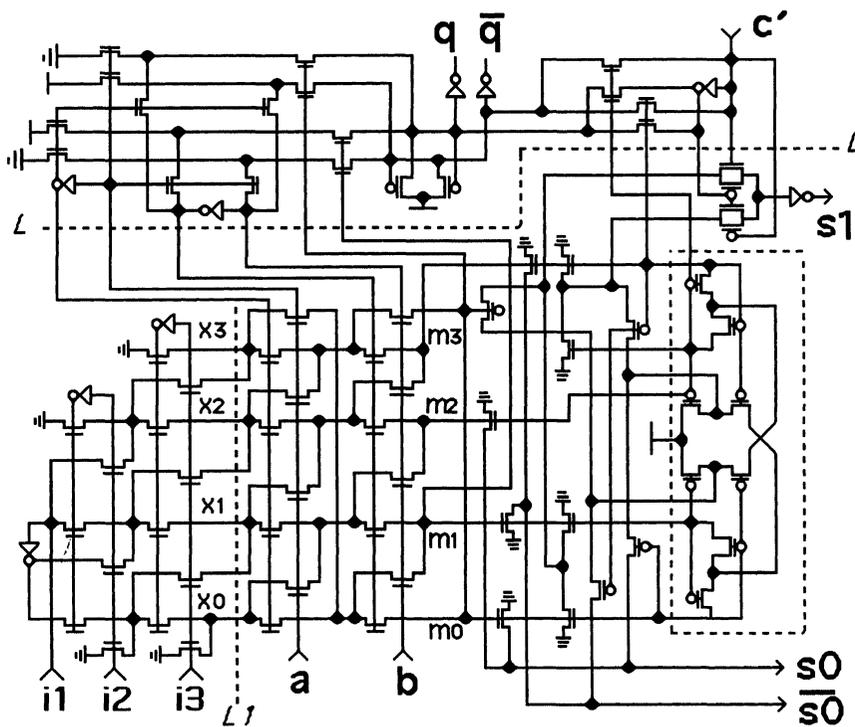
$$i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + Cin1 + Cin2 + Cin3 \\ = s + 2c + 2Cout1 + 2Cout2 + 2Cout3 \quad (E)$$

An alternative scheme is to use a non-binary parallel counter (6, 3)* of Figure 7, plus a complementary (3, 2) counter of Figure 9a or its variants in each column to reduce 6 input bits into 2.

A 4-bit state signal as shown in Figure 7 represents a decoded form of a binary number with an integer value between 0 to 3. In Figure 7, the initial 4-bit state signal X formed by bits x_0 , x_1 , x_2 , x_3 of the (6, 3)* counter has a value equal to $i_1 + i_2 + i_3$, note that the unique bit 1 of X is a level-swing signal and will be restored later. It then propagates cyclically along the horizontal direction. The intermediate state signal M formed by $m_0 m_1 m_2 m_3$ (note that the unique bit voltage level of M has been restored by the p-type restoration circuit inside each dotted box) is utilized in producing the output bits, including q , s_0 and s_1 .



(a)



(b)

FIGURE 7 Two non-binary logic $(6, 3)^*$ parallel counter circuits with CMOS pass-transistors. Note that each dotted box shows a p-type 4-bit state signal level restorer; $X = (x_0, x_1, x_2, x_3)$ and $M = (m_0, m_1, m_2, m_3)$ are two state signals. (a) parallel counter $(6, 3)^*a$; (b) parallel counter $(6, 3)^*b$ (with differential outputs).

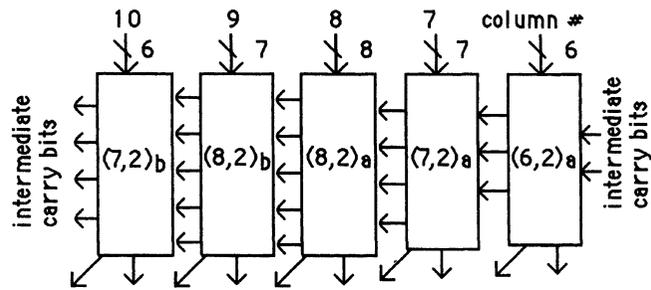


FIGURE 8 The complementary shift switch counter based 8×8 virtual multiplier (the critical columns are shown).

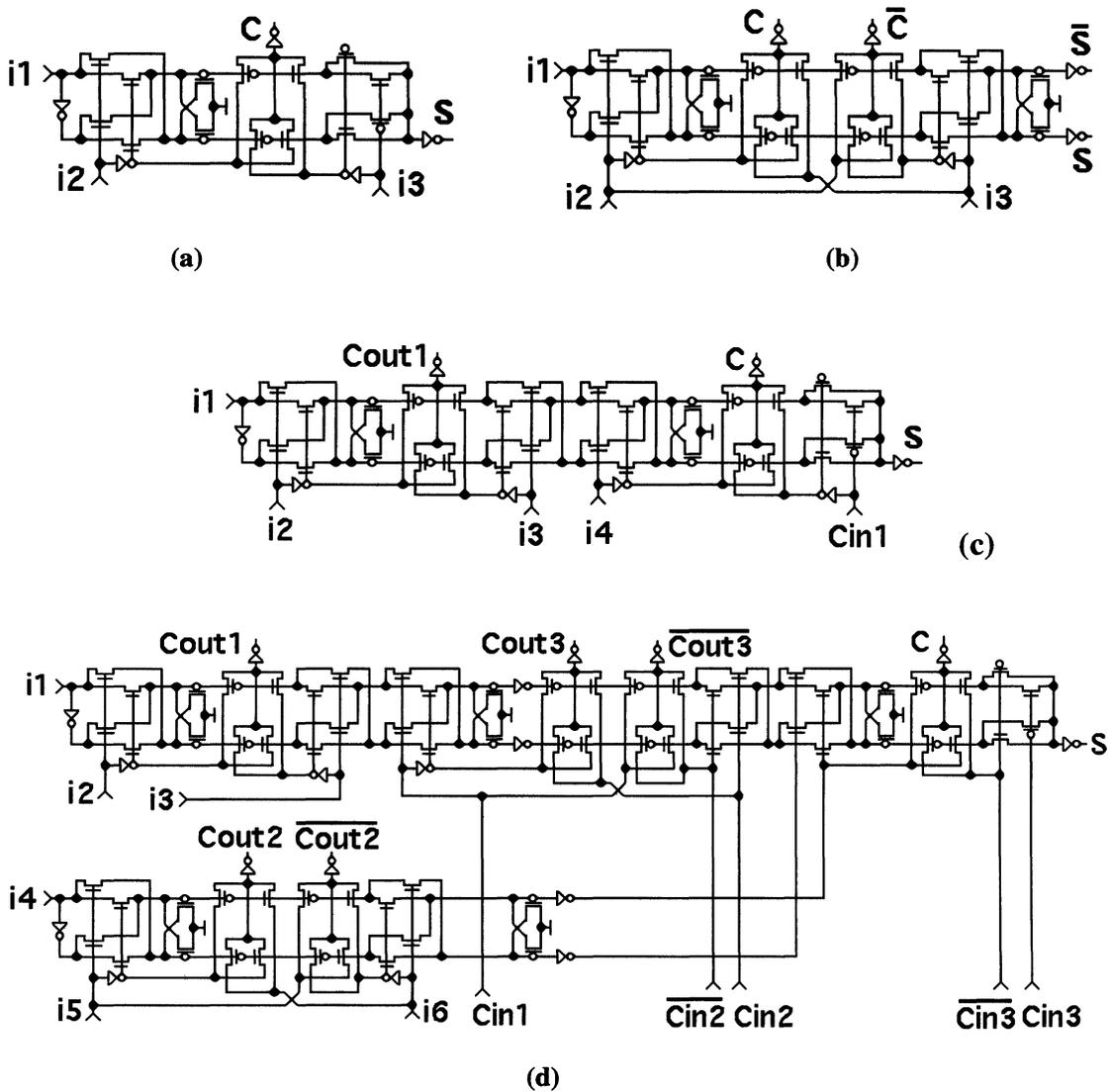


FIGURE 9 The complementary shift switch counters: (a) the (3,2) counter with a transistor count of 24; (b) the differential (3,2) counter with a transistor count of 30; (c) the tiny (4,2) with a transistor count of 44; and (d) the (6,2) parallel counter.

5. THE LOGICALLY LOW-POWER NATURES OF THE NON-BINARY ARITHMETIC CIRCUITS

In this section we characterize the low power natures of the proposed non-binary arithmetic circuits. Since the logical superiority of the circuits for low power dissipation may be best captured by the typical (6, 3)* parallel counter illustrated in Figure 7b, we redraw the circuit in Figure 10

focusing on illustrations of power dissipation activities occurred along signal paths.

As addressed in [2], the four sources of power dissipation in digital CMOS circuits can be summarized as: (1) $P_{switching}$, the switching component of power; (2) $P_{short-circuit}$, due to the direct-path short circuit current which arises when both the nMOS and pMOS transistors are simultaneously active, conducting current directly from the supply to the ground; (3) $P_{leakage}$, primarily

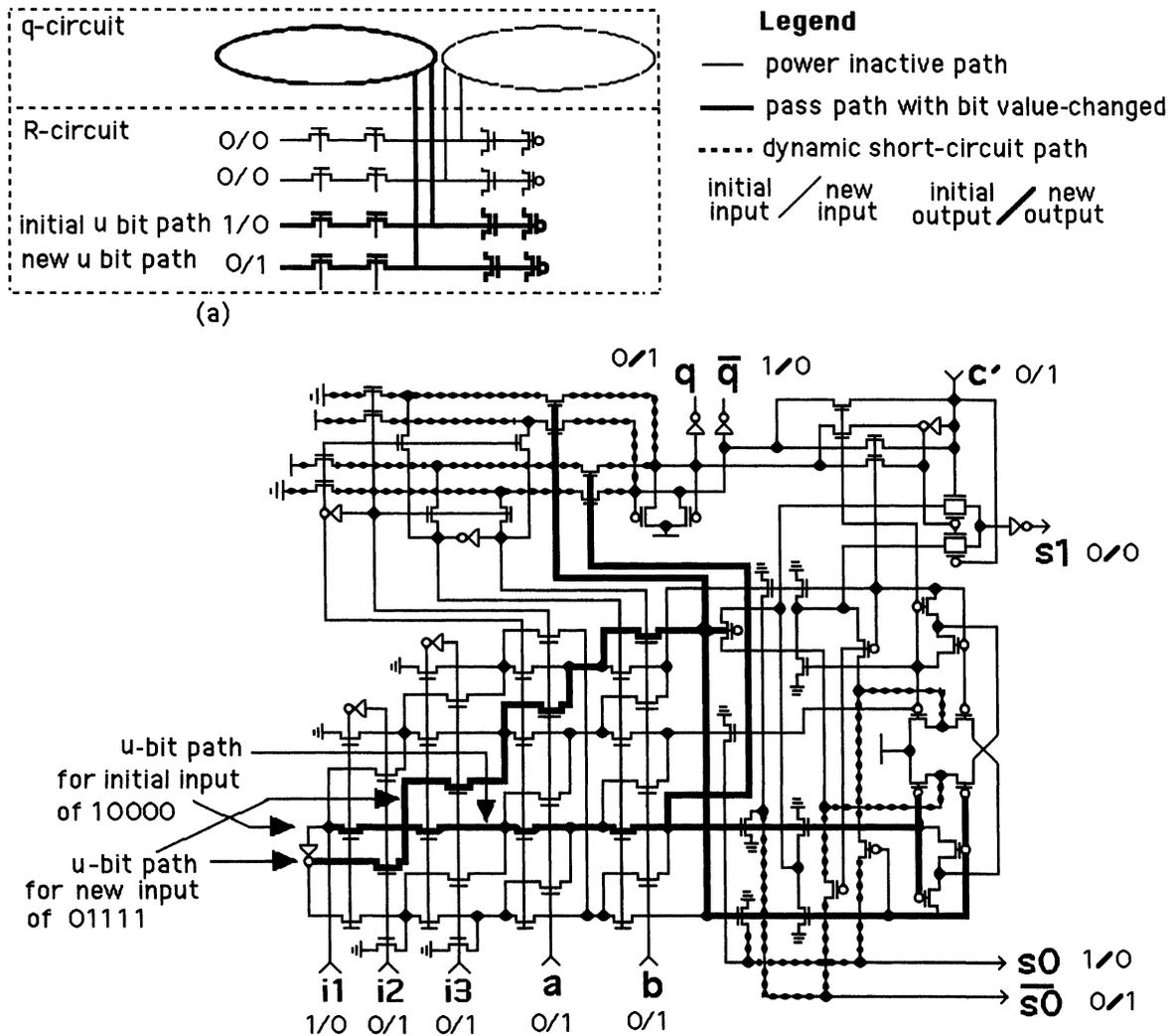


FIGURE 10 Power-consuming activity of a shift switch logic circuit: (a) an abstract illustration; (b) the (6, 3)* parallel counter of Figure 7b.

determined by fabrication technology considerations (but there is some room for reducing $P_{leakage}$ with circuit styles, see below), and finally, (4) P_{static} , arising from circuits that have a constant source of current between the power supply and the ground.

Referring to Figure 10, four low-power natures of the (6,3)* circuit now can be described as follows: The first comes from the fact that the logic transitions of the circuit are significantly related to the propagation of 4-bit state signals (as X and M in Fig. 7), where no more than half (or 2 out of 4) of the signal bits are subject to value-change at any logic stage. One of the worst cases of inputs of Figure 7b is shown in Figure 10. The initial value of the 6-bit input is given as 100000, the input is then changed to 011111. The bold lines indicate state signal u(unique) bit paths. The bold-dotted lines indicate short-circuit paths, excluding those within inverters (note that the dynamic current in these paths is weaker than that in a standard inverter). It is easy to see that near a half amount of the all transistors (except input/output inverters) does not have a power-consuming activity caused by state signal propagation, though a state signal may change its values at every step during the propagation. The charge/discharge transitions do not occur along the unbold lines. In contrast, a binary gate based circuit does not hold the property, where all transistors may take power-consuming transition during a computation.

The second low-power nature of the circuit, including lower signal level swings as well as higher ratio of nMOS to pMOS transistors, is directly inherent from the non-full swing pass transistor logic which is utilized in the implementation of the circuit.

The third low-power nature comes again from the shift switch logic itself, which allows almost all pMOS transistors being in minimal size (*i.e.*, with a size about the same as a minimum nMOS). Regular-size pMOS transistors are required only for the inverters which receiving input bits i_1 , i_2 and i_3 , to guarantee high speed state signal propagation along the sequence of four shift

“BARs”. The reason for that the pMOS in other inverters and in restoration circuits could be minimized is related to how restorations work in the circuit and the partially sequential nature of the signal propagation. And this has been verified by circuit-level simulation. The total VLSI area thus is minimized and the total switching capacitance is reduced, which also reduces the total leakage current.

The last but important low-power nature comes from the fact that three out of four signal bit-paths propagate 0 bits, only one path propagates 1 or level-high signal bit. We know that leakage current occurs only in the area occupied by level-high signal bits. In our approach approximately a quarter of the total signal passing area of the circuit is with level 1 signal bits in worst case, against about a half of that as for a binary logic circuit. This unique feature implies that the new circuit style can lead to a smaller $P_{leakage}$, compared with other CMOS circuit styles.

Now we also compare two implementation approaches proposed in the previous section, *i.e.*, Figure 7 based non-binary and Figure 9 based binary implementations (both are shift switch circuits). The non-binary logic counters require fewer inverters than the binary complementary counters in reducing same number of bits, therefore results in smaller $P_{short-circuit}$. It can be verified that the ratio of required inverters between these two approaches is about 3 vs. 5. The preliminary simulation results reveal that the selection of the above implementation schemes may lead to the trade-off among speed, VLSI area, and power dissipation of the circuits. In general, complementary (k,2) counter based schemes require slightly fewer transistors (about 5%) and may be slightly faster (5% to 10% by circuit simulation), while (6,3)* based schemes require smaller VLSI areas (due to the possible use of more smaller transistors [9, 11, 13]) and possess several advantages for low power dissipation as described above. Tables I and II show the circuit simulation results for the critical paths of the 8×8 multipliers and the related parallel counters respectively.

TABLE I Comparisons of 8×8 virtual multipliers (for the critical paths)

	A	B	C
Transistor count	152	142	144
Area EMTC	182	190	192
Power dissipation	0.47	0.60	0.74
Delay	1.28	1.32	1.55

Note: (1) All columns A, B and C represent the results for the critical paths of the 8×8 multipliers. Column A is for the path using non-binary-logic (6, 3)* of Figure 7b, column B is for the path using complementary (k, 2) parallel counter of Figure 9, and column C is for the path using (4, 2) parallel counter of [5]. (2) EMTC stands for Equivalent Minimum Transistor Count with nMOS=1, pMOS=3, minimum pMOS=1. (3) Worst case instantaneous power dissipation (in *mW-ns*) are used for the power comparisons. (4) The delay (in *ns*) is for the worst case delay amongst all inputs to all outputs.

TABLE II Comparisons of circuits for reducing 6 bits into 2 bits

	A	B	C
Transistor count	110	98	104
Area EMTC	125	138	145
Power dissipation	0.32	0.48	0.57
Delay	1.02	0.92	1.15

Note: Column A is for the non-binary-logic (6, 3)* based design; column B is for the complementary (6, 2) based design, and column C is for the traditional (4, 2)-(3, 2) of [5] based design; also refer to note of Table I for detailed comparison descriptions.

6. CONCLUDING REMARKS

A highly regular parallel multiplier design has been presented. The approach has minimized the common irregularity occurred in existing designs and simplified the overall logic scheme and wiring structures. The superiority in low power dissipation may be achieved through the use of large amount of identical low power, high performance 4-bit (non-binary) and 2-bit (complementary) state signal based shift switch parallel counter circuits, as well as repeatable modules (for several levels of sub-multipliers). SPICE circuit simulations have demonstrated the advantages of the overall multiplier architecture and the new component circuits. The proposed schemes can be easily extended for multipliers larger than that of 64×64 -b. Both the novel multiplier architecture and the proposed component circuits can also be applied independently for specific arithmetic unit designs.

References

- [1] Booth, A. D. (1951). A signed binary multiplication technique, *Quart. J. Mech. Appl. Math.*, **4**.
- [2] Chandrakasan, A. P. and Brodersen, R. W. (1995). *Low Power Digital CMOS Design*, Kluwer Academic Publishers.
- [3] Dadda, L. (1976). On parallel digital multipliers, *Alta Freq.*, **45**.
- [4] Goto, G., Sato, T., Nakajima, M. and Sukemura, T. (1992). A 54×54 -b regularly structured tree multiplier, *IEEE J. of Solid-state Circuits*, **27**(9).
- [5] Goto, G., Inoue, A., Ohe, R., Kashwakura, S., Mitarai, S., Tsuru, T. and Izawa, T. (1997). A 4.1-ns compact 54×54 -b multiplier utilizing sign-select Booth encoders, *IEEE J. of Solid-state Circuits*, **32**(11).
- [6] Kai Hwang (1979). *Computer Arithmetic*, New York: John Wiley.
- [7] Law, C. F., Rofail, S. S. and Yeo, K. S. (1999). A lower-power 16×16 -b parallel multiplier utilizing pass-transistor Logic, *IEEE J. of Solid-state Circuits*, **34**(10).
- [8] Lin, R. (2000). A Regularly Structured Parallel Multiplier With Non-Binary-Logic Counter Circuits, In: *Proc. of the Second IEEE Asia-Pacific Conference on ASICs*, Cheju Island, Korea.
- [9] Lin, R. (2000). Parallel VLSI Shift Switch Logic Devices (*US Patent 96125379*). 1999; Reconfigurable inner product processor architecture (*US Patent pending*); (2000) and A Family of High Performance Multipliers and Matrix Multipliers (*US Patent pending, No. R-1265-125*).
- [10] Lin, R. (2000). Reconfigurable Parallel Inner Product Processor Architectures, to appear in *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*.
- [11] Lin, R. (2000). Parallel Multiplier Designs Utilizing A Non-Binary Logic Scheme, In: *Proc. of Euromicro'00, Workshop on Digital System Design*, IEEE Press, Maastricht, the Netherlands.
- [12] Lin, R. (2000). Low-Power High-Performance Non-Binary CMOS Arithmetic Circuits, to appear In: *Proc. of 2000 IEEE Workshop on SIGNAL Processing Systems (SiPS)*, Lafayette, Louisiana.
- [13] Lin, R. (1992). Reconfigurable buses with shift switching-VLSI radix sort, *Proc. of International Conference on Parallel Processing (ICPP)*, Chicago, **III**, 2-9.
- [14] Lin, R., Kerr, K. E. and Botha, A. S. (1999). A novel approach for CMOS parallel counter design, In: *Proc. Euromicro'99, Workshop on Digital System Design*, Milan, Italy, 112-119.
- [15] Lin, R. and Olariu, S. (1995). Reconfigurable buses with shift switching-architectures and applications, In: *IEEE Transactions on Parallel and Distributed Systems*, **6**(1), 93-102.
- [16] Pasternak, J. H., Shubat, A. S. and Salama, C. A. T. (1987). "CMOS differential pass-transistor logic design", *IEEE J. of Solid-state Circuits*, SC-22, pp. 216-222.
- [17] Swartzlander, E. E. Jr., *Computer Arithmetic* Vol. 1. 2. (IEEE CSP, CA, 1990).
- [18] Yano, K., Yamanaka, T., Nishida, T., Saito, M., Shimohigashi, K. and Shimizu, A., A 3.8-ns CMOS 16×16 multiplier using complementary pass-transistor logic, *IEEE J. of Solid-state Circuits*, **25**(2), April, 1990.
- [19] Yano, K., Sasaki, Y., Rikino, K. and Seki, K. (1996). Top-down pass-transistor logic design, *IEEE J. of SSC*, **31**(6).

Author's Biography

Rong Lin received the BS degree in mathematics from Peking University, Beijing China, the MS degree in computer science from Beijing Polytechnic University, Beijing China, and Ph.D. degree in computer science from Old Dominion University, Norfolk, Virginia, in 1989. He Joined the faculty

of State University of New York at Geneseo in September, 1989, where he now is a professor and the department chair of computer science. His current research interests include parallel architectures, VLSI arithmetic circuits, run-time-reconfigurable digital circuits, and parallel algorithm designs.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

