LOW POWER BZ-FAD MULTIPLIER BY USING SHIFT AND ADD ARCHITECTURE

T.Saibaba [1], D.Ritafaria [2]

[1, 2] *Asst.Professor, Dept.of ECE, CJITS, Jangaon, Andhra Pradesh, (India).*

ABSTRACT

In this paper, we proposes a new and low power architecture for synchronous ring counters which can noticeably reduce the switching activity of conventional ring counters. The latency increases in the proposed architecture is low power multiplier, this structure is also called as Bypass Zero, Feed A directly (BZ-FAD) for shift-and-add architecture. The architecture reduces the switching activity of the conventional multipliers. The modifications to the multiplier which multiplies A by B include the removal of shifting the B register, direct feeding of A to the adder, by passing the adder whenever possible, using a ring counter instead of a binary counter and removal of the partial product shift. Simulation results for 16-bit radix-2 multiplier show that the BZ-FAD architecture lowers the total switching activity up to 76% and reducing the consumption up to 30% when compared to the conventional architecture. The proposed multiplier can be used for low-power applications where the speed is not a primary design parameter. The architecture makes use of a low power ring counter proposed in this work.

Keywords: Hot Block Ring Counter, Low Power Multiplier, Low Power Ring Counter, Shift- And- Add Multiplier, Switching Activity Reduction.

I.INTRODUCTION

Multipliers are among the fundamental components of many digital systems and, hence, their power dissipation and speed are of prime concern. For portable applications where the power consumption is the most important parameter, one should reduce the power dissipation as much as possible. One of the best ways to reduce the dynamic power dissipation is to minimize the total switching activity, i.e., the total number of signal transitions of the system.

Many research efforts have been devoted to reducing the power dissipation of different multipliers. The largest contribution to the total power consumption in a multiplier is due to generation of partial product. Among multipliers, tree multipliers are used in high speed applications such as filters, but these require large area. The carry-select-adder (CSA)-based radix multipliers, which have lower area overhead, employ a greater number of active transistors for the multiplication operation and hence consume more power. Among other multipliers, shift-and-add multipliers have been used in many other applications for their simplicity and relatively small area requirement. Higher-radix multipliers are faster but consume more power since they employ wider registers, and require more silicon area due to their more complex logic.

In these work, we propose modification to the conventional architecture of the shift-and-add radix-2 multipliers to considerable reduce its power consumption. This paper is organized as follows: towards a low power shift-and-add

multiplier (sec.II), hot block ring counter (sec.III), results and discussion in section IV with section V containing summary.

II.TOWARDS A LOW POWER SHIFT-AND-ADD MULTIPLIER.

2.1 Main Sources of Switching Activity

The architecture of a conventional shift-and-add multiplier, which multiplies A by B, is shown in Figure 3.1. There are six major sources of switching activity in the multiplier. These sources, which are marked with dashed ovals in the figure, are: (a) Shifts of the B register.(b) Activity in the counter.(c) Activity in the adder.(d) Switching between "0" and A in the multiplexer.(e) Activity in the mux-select controlled by B (0).(f) Shifts of the partial product (*PP*) register. Note that the activity of the adder consists of required transitions (when B (0) is nonzero) and unnecessary transitions (when B (0) is zero).

By removing or minimizing any of these switching activity sources, one can lower the power consumption. Since some of the nodes have higher capacitance, reducing their switching will lead to more power reduction. As an example, B (0) is the selector line of the multiplexer which is connected to k gates for a k-bit multiplier. If we somehow eliminate this node, a noticeable power saving can be achieved. Next, we describe how we minimize or possibly eliminate these sources of switching activity.

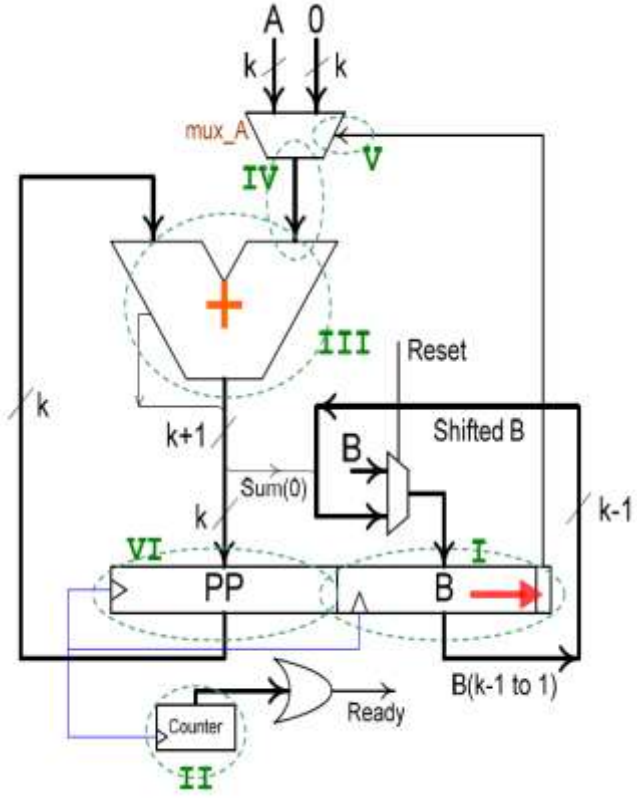


Fig 1: Architecture of Conventional Multiplier.

2.2 The Proposed low power multiplier: BZ-FAD.

A low-power structure called by pass zero, feed A directly (BZ-FAD) for shift-and-add multipliers is proposed. The architecture considerably lowers the switching activity of conventional multipliers. The modifications to the multiplier which multiplies A by B include the removal of the shifting the register, direct feeding of A to the adder, bypassing the adder whenever possible, using a ring counter instead of a binary counter and removal of the partial product shift. The architecture makes use of a low-power ring counter proposed in this work.

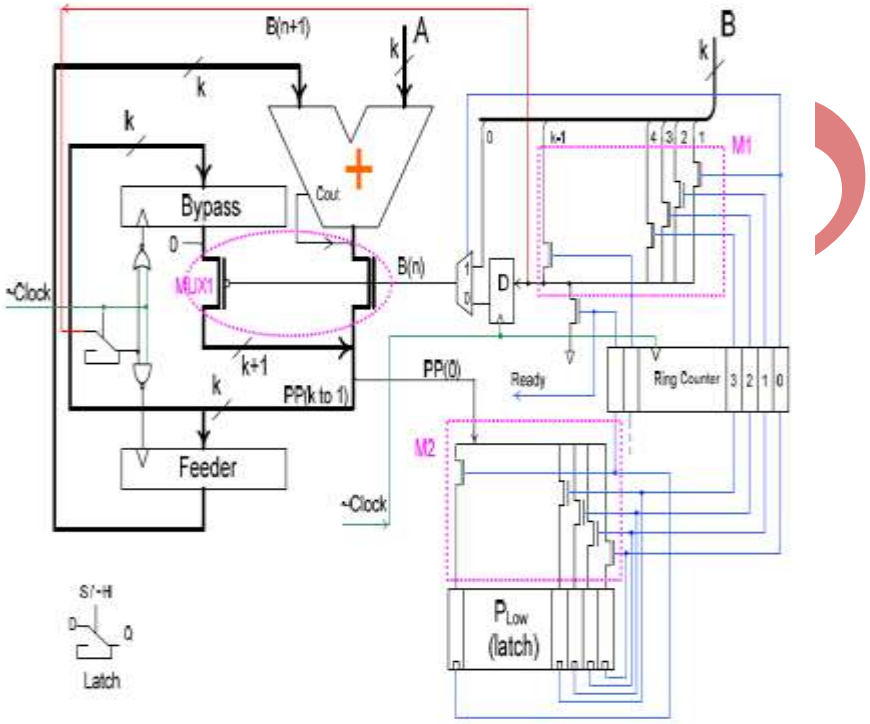


Fig 2: Architecture of BZ-FAD

2.2.1 Shift of the B Register

In the traditional architecture (see Fig.2), to generate the partial product, *B (0)* is used to decide between A and 0. If the bit is "1", A should be added to the previous partial product, where as if it is "0", no addition operation is needed to generate the partial product. Hence, in each cycle, register *B* should be shifted to the right so that its right bit appears at *B (0)*; this operation gives rise to some switching activity. To avoid this, in the proposed architecture (see Fig. 2) a multiplexer *(M1)* with one-hot encoded bus selector chooses the hot bit of *B* in each cycle.

A ring counter is used to select *B (n)* in the *n*th cycle. The same counter can be used for block *M2* as well. The ring counter used in the proposed multiplier is noticeably wider (32 bits versus 5 bits for a 32-bit multiplier) than the binary counter used in the conventional architecture; therefore an ordinary ring counter, if used in BZ-FAD, would raise more transitions than its binary counterpart in the conventional architecture. To minimize the switching activity of the counter, we utilize the low-power ring counter.

2.2.2 Reducing Switching Activity of the Adder

In the conventional multiplier architecture (see Fig.1), in each cycle, the current partial product is added to A (when B (0) is one) or to 0 (when B (0) is zero). This leads to unnecessary transitions in the adder when B (0) is zero. In these cases, the adder can be bypassed and the partial product should be shifted to the right by one bit. This is what is performed in the proposed architecture which eliminates unnecessary switching activities in the adder. As shown in Fig. 2, the Feeder and Bypass registers are used to bypass the adder in the cycles where B (n) is zero. In each cycle, the hot bit of the next cycle (i.e., B (n+1)) is checked. If it is 0, i.e., the adder is not needed in the next cycle, the Bypass register is clocked to store the current partial product. If B (n+1) is 1, i.e., the adder is really needed in the next cycle, the Feeder register is clocked to store the current partial product which must be fed to the adder in the next cycle.

2.2.3 Shift of the PP Register

In the conventional architecture, the partial product is shifted in each cycle giving rise to transitions. Inspecting the multiplication algorithm reveals that the multiplication may be completed by processing the most significant bits of the partial product, and hence, it is not necessary for the least significant bits of the partial product to be shifted. We take advantage of this observation in the BZ-FAD architecture. Notice that in Fig. 2 for P_{low} the lower half of the partial product, we use 'k' latches (for a k-bit multiplier). These latches are indicated by the dotted rectangle *M2* in Fig. 2. In the first cycle, the least significant bit *PP (0)* of the product becomes finalized and is stored in the right-most latch of P_{low} . The ring counter output is used to open (unlatch) the proper latch. This is achieved by connecting the S/~H line of the *n*th latch to the *n*th bit of the ring counter which is "1" in the *n*th cycle. In this way, the *n*th latch samples the value of the *n*th bit of the final product (see Fig. 2). In the subsequent cycles, the next least significant bits are finalized and stored in the proper latches. When the last bit is stored in the left-most latch, the higher and lower halves of the partial product form the final product result.

III. HOT BLOCK RING COUNTER

The power consumption of digital circuits has become a critically important parameter motivating many efforts in reducing the power dissipation of the logic blocks of digital systems. Among different blocks, ring counter is one of logic components which have several applications including control units and, multiplier and divider architectures, and the arbitration circuitry (round robin arbitration) of routers. One of the important properties of a ring counter is that its output is one-hot encoded (*i.e.*, there is always only a single '1'-valued bit in its output and all other bits are zero). This property of the ring counter makes its output wide especially as the counter size increases. As an example, consider a 5-bit binary counter which counts from 0 to 31. A ring counter with the same counting range is 32-bits wide. Hot-Block architecture lowers the switching activity of ring counters.

The first step toward a low power design is to detect signals which can be temporarily or locally shot off without affecting the circuit functionality. Therefore inspection of ring counter logic is performed.

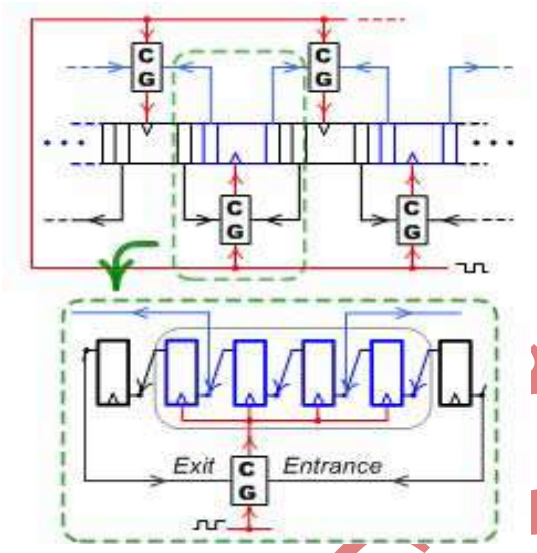


Fig 3. Hot Block Ring Counter Architecture

In order to have a low activity ring counter, partitioning of the ring counter into *b* blocks each of which is clock-gated with a special clock gator as shown in "**Fig. 3**".

Hot Block architecture there is many flip-flops, the output of which does not go to any clock gator. This point noticeably reduces the total switching activity of the ring counter. A ring counter has a very interesting property of whose advantage is taken in design of Hot Block architecture**:** a '1' is moving through the cells of the ring counter, and at each moment all cells of the ring counter except one is zero(0). This property helps in removing the OR gate from the clock gator of "**Fig. 4**". The cited property tells us that in the partitioned ring counter of "**Fig.3**" there is always only one block the flip-flops of which should be clocked (except in some occasions where the '1' leaves a block and enters another); this block is called the Hot Block (hence the architecture name).

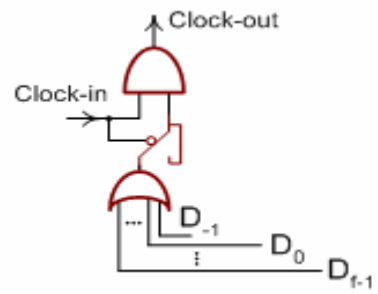


Fig 4: The multiple bit clock gator logic.

Although reduces the total switching activity of the ring counter, the multiple bit clock gator of "Fig. 4" is problematic in terms of area overhead and switching activity. The OR gate of the logic widens as the block becomes wider and poses fan-in problems; as an example for 8-bit blocks the OR gate has 9 inputs which should be divided to 3 OR gates with fewer inputs.

The area overhead is dependent on the block size such that as the block size increases the area overhead decreases. However, the larger the block size is, the higher the power consumption .The critical path of the Hot Block architecture is the same as that of the conventional architecture except that the clock signal in the Hot Block passes through a NAND gate.

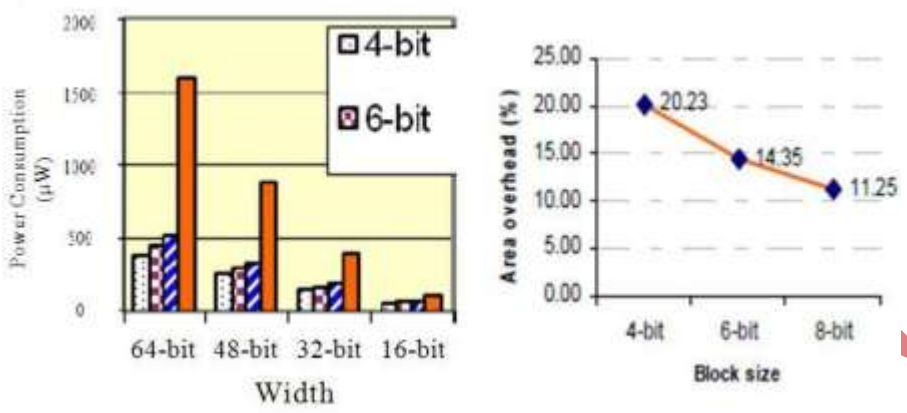


Fig 5. (a) Power consumption of the conventional ring counter versus that of the Hot-Block ring counter with different block sizes - (b) The area overhead for different block sizes.

Therefore, the edge difference between the *~Clock-IN* and *Clock-OUT*, which is independent of the counter width, is equal to the delay of a NAND gate. This technology dependent delay is very short (e.g. about 10ps for the TSMC 0.13µm CMOS).

IV. RESULTS AND DISCUSSIONS

In brief, from the six sources of activity in the multiplier, we have eliminated the shift of the B register, reduced the activities of the right input of the adder, and lowered the activities on the multiplexer select line. In addition, we have minimized the activities in the adder, the activities in the counter, and the shifts in the PP (partial product) register.

Component	BZ-FAD	Conventional	Reduction (%)
Low order partial product	6,564 (latch)	82,208 (Register B)	92.02 %
Adder	46,301	74,870	38.16 %
Multiplexer	56,722	10,013 (mux_A)	-82.35 %
Counter	20,965	22,937	8.60 %

Table 1: Comparison of Transition Counts in BZ-FAD with Conventional Multiplier

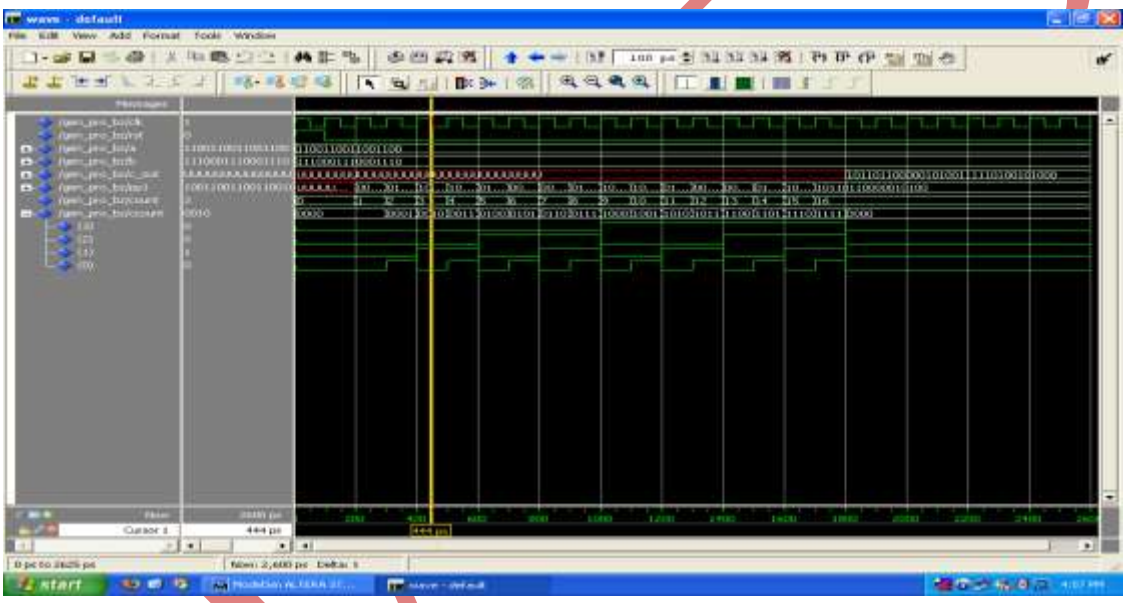
The proposed architecture, however, introduces new sources of activities. These include the activities of a new multiplexer which has the same size as that of the multiplexer of the conventional architecture. Note that the higher part of the partial product in both architectures has the same activity.

V.CONCLUSION & FUTURE SCOPE

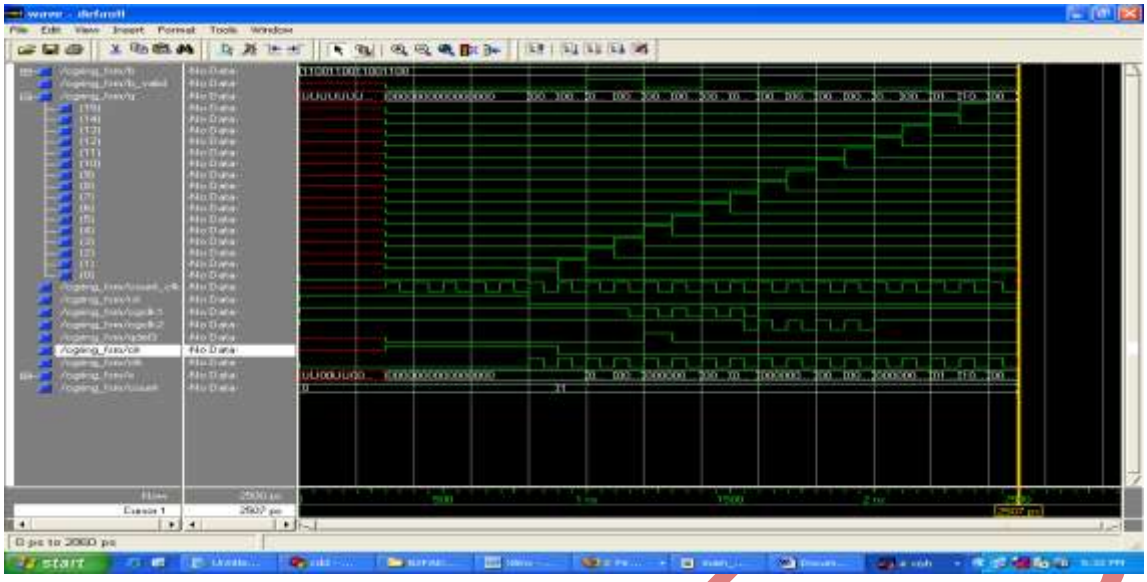
BZ-FAD multiplier can optimize two design constraints simultaneously. The use of hot block ring 75% for a 64-bit counter. The implementation of this multiplier in the design of digital systems can yield good results and for applications where small area and high speed are important concerns, BZ-FAD is an excellent choice.

VI. SIMULATION RESULTS

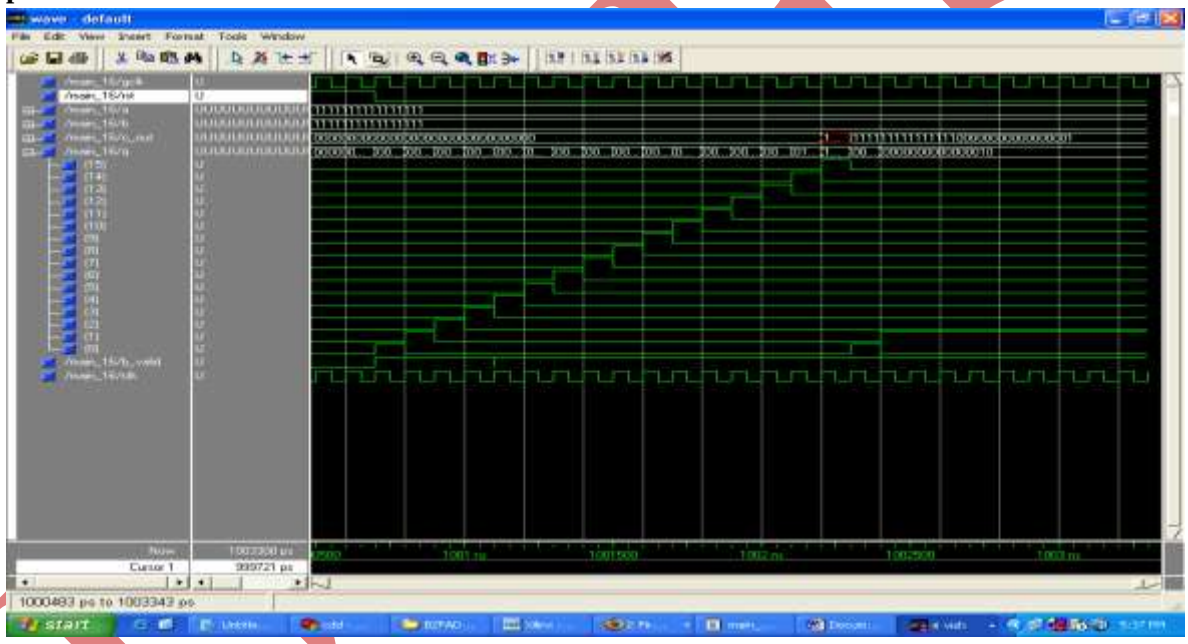
6.1 Existing BZ-FAD Simulation Results



6.2 CG plus Ring Counter Simulation Results



6.3 Proposed BZ-FAD Simulation Results



VII. REFFERENCES

[1] M.Mottaghi-Dastjerdi, A.Afzali-Kusha, and M.Pedram "*IEEE Transactions on very large scale integration (vlsi) Systems"*, Vol.17, No.2, February 2009"

[2] A. Chandrakasan and R. Brodersen, "Low-power CMOS digital design,"IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473–484, Apr.1992.

[3] N.-Y. Shen and O. T.-C. Chen, "*Low-power multipliers by minimizing switching activities of partial products*," in Proc. IEEE Int. *Symp. Circuits* Syst., May 2002, vol. 4, pp. 93–96.

[4] O. T. Chen, S. Wang, and Y.-W. Wu, "*Minimization of switching activities of partial products for designing low-power multipliers*," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 3, pp. 418–433, Jun. 2003.

[5] B. Parhami, *"Computer Arithmetic Algorithms and Hardware Designs"*, 1st ed. Oxford, U.K.: Oxford Univ. Press, 2000.

[6] V. P. Nelson, H. T. Nagle, B. D. Carroll, and J. I. David, *"Digital Logic Circuit Analysis & Design"*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[7] K.-H. Chen and Y.-S. Chu, "*A low-power multiplier with the spurious power suppression technique*," IEEE Trans. Very Large Scale Integr.