A Power-Aware 2-Dimensional Bypassing Multiplier Using Cell-Based Design Flow

Gang-Neng Sung, Yan-Jhih Ciou, and Chua-Chin Wang, *Senior Member, IEEE*

Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
Email: ccwang@ee.nsysu.edu.tw

***Abstract*— This paper presents a low power digital multiplier design by taking advantage of a 2-dimensional bypassing method in cell-based design flow. The proposed bypassing cells constituting the multiplier skip redundant signal transitions when the horizontally partial product or the vertically operand is zero. Thorough cell-based design flow post-layout simulations show that the power delay product of the proposed 8×8 multiplier design is reduced by more than 13.8% compared to prior designs.**

***Keywords*—low power multiplier, bypassing, CMOS, cell-based, partial product, timing control.**

I. Introduction

Low power design has become a great concern in VLSI design in recent years. Booming of battery-operated multi-media devices requires energy-efficient circuits, particularly digital multipliers which are building blocks of digital signal processors (DSP). Though many efforts have been focused on the improvement of adder and multiplier designs [1], the major trade-off of these high speed logic circuits are the high power consumption and high temperature which are not a tolerable price to pay in recent mobile technologies [2]. Besides adders, digital multipliers are the most critical arithmetic functional unit in many DSP applications, e.g., Fourier Transform, DCT, digital filtering, etc. Array and parallel multipliers are very welcomed due to their high execution speed and throughput besides its high regularity. However, the increasing capacitive wire load and operands' bit length result in very large power dissipation, [3], [4], [5], [6], [7]. Despite all of these difficulties, we still manage to reduce the power dissipation by an observation that the energy consumption of CMOS logic is proportional to the number of transitions, i.e., $P_{diss} \propto f \cdot C \cdot V^2$, where f is the frequency of switches, C is the load, and V denotes the voltage swing.

Many prior digital multipliers were aimed at transition or switch reductions to reduce power dissipation as well. A leapfrog multiplier was proposed in [7] by using a hardware bypassing approach to avoid the redundant computations by disabling the adder units whose partial product becomes zero. Another power saving approach is to skip the computation caused by the sign extension bits which are located at the left side of operands, e.g., [4]. What [6] proposed was close to a "bypassing" multiplier which skips the addition when the partial product of a row is zero. [3] revealed another power-saving strategy by grouping the operands with the same sign and then computing them separately to avoid unnecessary transitions. All of these prior methods depend on certain decision logic given that a partial product is zero to either skip or shut down adding cells in a row-based manner. In other words, all of these prior works utilized a one-dimensional bypassing approach basically. We, thus, propose a 2-dimensional bypassing approach which detects the nullity of the partial products as well as the multiplicand at the same time to determine whether the adding cells on the corresponding row and those on the corresponding column are skipped or not, respectively [9]. A 8×8 digital multiplier using the proposed 2-dimensional bypassing design is carried out by TSMC 0.18 μm 1P6M CMOS cell-based design process. The post-layout simulations show that the power delay product reduction compared to the prior multipliers is at least 13.8%.

II. 2-Dimensional Bypassing Multiplier

A basic guideline to reduce the power dissipation of a digital multiplier is to reduce its unnecessary switching activities. Hence, we proposed to detect the bitwise nullity of the multiplicand in the vertical direction and the partial product in the horizontal direction in an array multiplier to remove the unnecessary operations taken place in the corresponding adding cells.

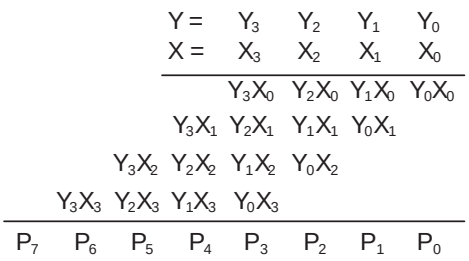
A. Prior 1-dimensional bypassing algorithm

A representative array multiplication is based upon the following equation.

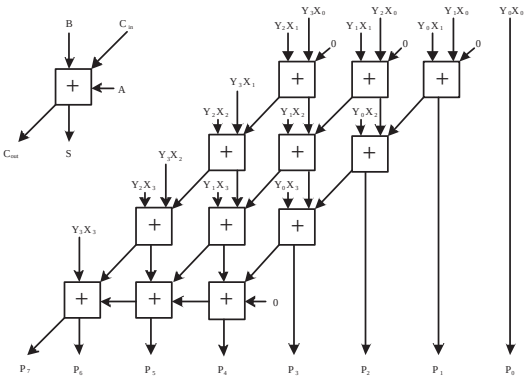
$$\begin{aligned} P &= P_{2n-1} \ldots P_1 P_0 \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (X_i \cdot Y_j) 2^{i+j} \end{aligned} \quad (1)$$

where P is the product, $X = X_{n-1} \ldots X_1 X_0$ is the multiplier, and $Y = Y_{n-1} \ldots Y_1 Y_0$ is the multiplicand. $P_k, k = 2n-1, \ldots, 0$, denote the partial products, $X_i, i = n-1, \ldots, 0$, and $Y_j, j = n-1, \ldots, 0$, are respectively the bit representations of the multiplier and the multiplicand, and n is the bit length of the operands. A typical implementation of such a multiplier

is the Braun's design shown in Fig. 1. Every adding unit consists of an AND to carry out the multiplication and an FA (full adder) to accumulate the partial product. An $n \times n$ multiplication, thus, requires a total of $n(n-1)$ FAs and n^2 AND gates.



(a)



(b)

Fig. 1. Generic array multiplier (Braun's)

A simple thought to improve the power efficiency was proposed by [7]. As soon as X_i was found to be zero, the corresponding partial product (row direction) is automatically reset and bypassed to avoid triggering those adding units in the row. Hence, two MUXs (multiplexer) are required in the adding unit to realize the bypassing operation. Meanwhile, there is a possibility that bypassing operations will result in the truncation of the carry from the corresponding previous stages. A total of $2(n-1)^2$ MUXs must be included to resolve this problem. A 4×4 multiplier example using such an implementation is shown in Fig. 2.

B. 2-dimensional bypassing design

Besides the power saving by row-based bypassing, we propose a 2-dimensional bypassing which detects the bitwise nullity of the multiplicand bits, Y_j's, in addition to the state of the multiplier, X_i's. In other words, as soon as the Y_j is found to be zero, the results from the adding units residing in the previous column are automatically passed to the corresponding adding units in the next column. However, a conflict appears

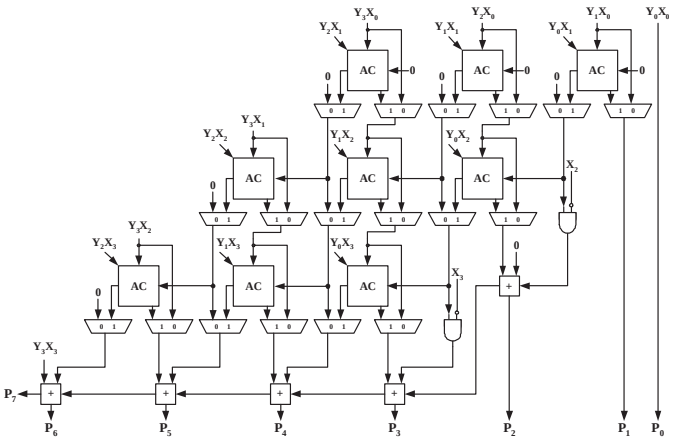


Fig. 2. Prior 1-dimensional bypassing multiplier design [7]

when one adding cell, AC_{ij}, encounters a scenario that $X_i = Y_j = 0$.

For instance, assume $i = 2, j = 1$ and $X_2 = Y_1 = 0$ in Fig. 3 which shows a 2-dimensional bypassing 4×4 multiplier design. Then, we expect the row 2 and the column 1 are bypassed if we directly apply the prior 1-dimensional bypassing method. If the carry out of the adding cell AC_{12} is "1", it should be propagated to the carry in of AC_{31} and then its carry out. However, the carry bit will be lost if AC_{31} is bypassed due to $Y_1 = 0$. Consequently, an error is occurred, since the carry out of AC_{31} will be zero. Another instance occurs conflict when $X_2 = Y_0 = 0$, the adding cell AC_{30} will miss the carry in because the column 0 is bypassed. We, thus, propose to include a bypass logic (BL) in certain adding cells.

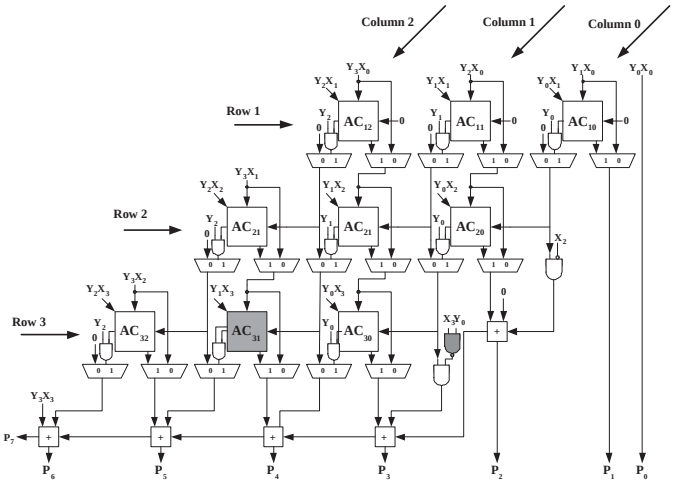


Fig. 3. Proposed 2-dimensional bypassing multiplier (4×4)

C. Adding cell with bypass logic

According to the illustrative example, a simple rule is introduced : **If and only if X_i is not equal to "0" and the carry in is "1", then the adding cell, AC_{ij}, can not**

be bypassed. Hence, an adding cell with the bypass logic is proposed in Fig. 4. It is also represented by a gray box in Fig. 3. Others AC_{ij}s are shown in Fig. 5.

In order to save more bypass logic area, the adding cell AC_{30} can be further simplified such that a single NAND gate (gray one shown in Fig. 3) is used to replace the adding cell with bypass logic. It still provides a correct operation when $X_2 = Y_0 = 0$ and carry out of AC_{11} is "1".

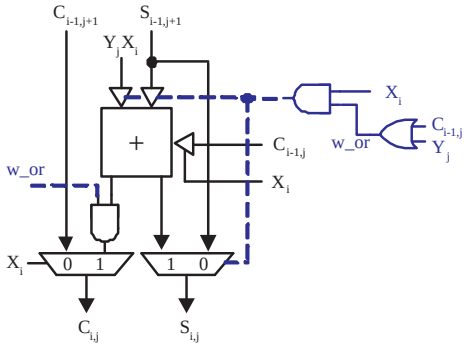


Fig. 4. Adding cell with bypass logic

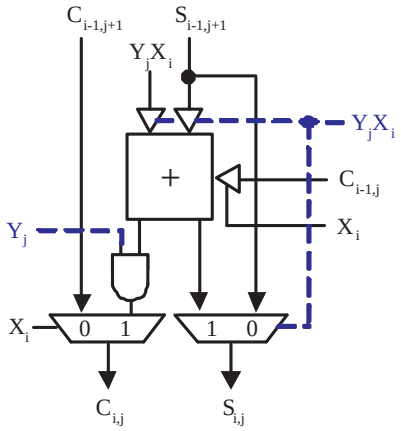


Fig. 5. Adding cell without bypass logic

D. Domino effect in large multipliers

It is obvious that not every adding cell needs the bypass logic. For instance, those adding cells in charge of the calculation of LSBs of X and Y. It will be very area-efficient if we can identify which adding cells require the bypass logic to produce a correct multiplication result. Given $n = 4$, it can be easily concluded that AC_{31} is the only unit with the necessity of a bypass logic. If $n = 5$ and the identical array structure is used, then $AC_{31}, AC_{32}, AC_{41}, AC_{42}$ need the bypass logic to attain correct results. By a similar induction, for any $n \times n$ multipliers, where $n \geq 4$, all of the adding cells, AC_{ij}, where $n-1 \geq i \geq 3$ and $n-3 \geq j \geq 1$, must contain the bypass logic to execute the correct multiplication. In other words, when $n = 4$, there is only one adding cell which must contain the bypass logic. If $n = 5$, then the 5×5 multiplier has a total of $(5-3) \times (5-3) = 4$ adding cells with bypass logic. If $n = 8$, a total of $(8-3) \times (8-3) = 25$ adding cells with bypass logic are required, as shown in Fig. 6. In short, the number of the required adding cells with bypass logic is as follows.

$$\begin{aligned} 1 &= (4-3) \times (4-3), \quad n = 4 \\ 4 &= (5-3) \times (5-3), \quad n = 5 \\ 9 &= (6-3) \times (6-3), \quad n = 6 \\ \vdots &= \vdots \end{aligned}$$

Therefore, the following rule is concluded.

Theorem 1 : A total of $(n-3)^2$ adding cells with bypass logic are required to constitute a 2-dimensional bypassing multiplier, $\forall n > 3$.

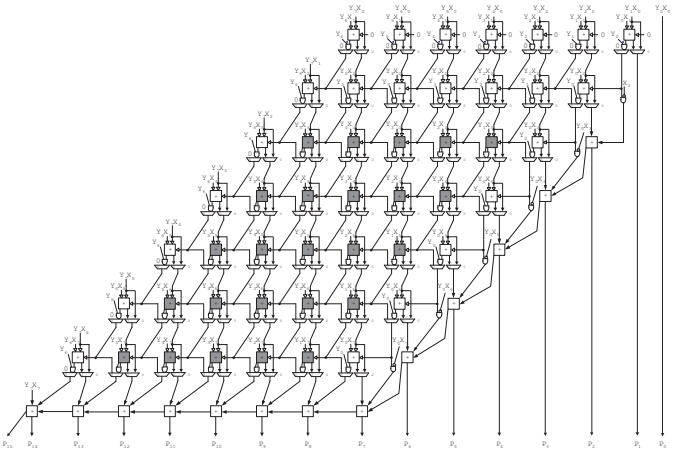


Fig. 6. A 8×8 multiplier using 2-dimensional bypassing

Notably, if an earlier adding cell with the bypass logic is set to be activated, all of the following adding cells in the same column must be activated, too. Otherwise, a carry generated in the earlier adding cell will be lost in the bypassing chain. For instance, if the adding cell AC_{32} is activated, then the following adding cell, AC_{42}, must be activated to ensure a carry (=1) is propagated correctly from the carry in of AC_{32} to the carry out of AC_{42} and even further. Namely, it is a "domino" effect of activation of adding cells in the same column.

III. Simulation and Implementation

TSMC (Taiwan Semiconductor Manufacturing Company) 0.18 μm 1P6M CMOS cell-based design flow process was adopted to carry out the proposed design. Fig. 7 is the layout of the proposed 8×8 multiplier using 2-dimensional bypassing. The core of the chip are our design and column-bypassing design proposed by [8]. Therefore, we can compare the power dissipation between these two designs using the same technology.

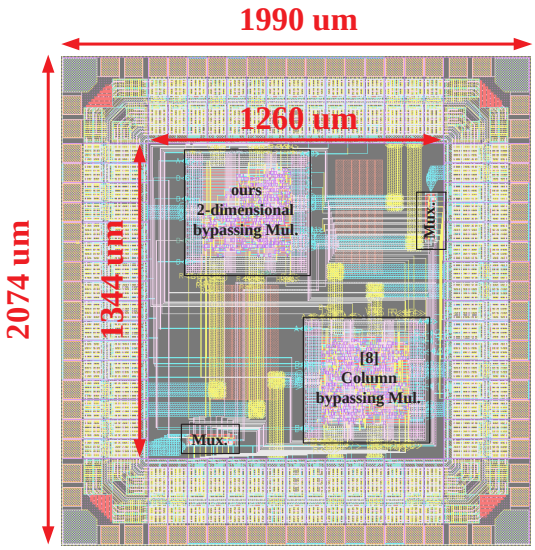


Fig. 7. Layout of the 8×8 multiplier using 2-dimensional bypassing

Fig. 8 shows area comparison of several designs. After using the Design Compiler software of Synopsys to synthesize these designs, the proposed design reduces a great portion of area. When the operand length n increases, the ratio of area reduction is increasedly more drastically. Table I shows the power delay product (PDP) comparison with these prior designs. Table II shows the comparison with our prior work [9]. Notably, our work in [9] was implemented by full-custom flow such that it lacks flexibility. Besides, the transistor count of one single bypass logic is reduced from 28 to 12. If the proposed design is also designed by full-custom method, the power dissipation will be much less than 7.00 mW.

Design	4×4 @ 125 MHz	8×8 @ 45 MHz	16×16 @ 21 MHz	32×32 @ 9 MHz
Braun	11.67 (1)	19.98 (1)	49 (1)	133.64 (1)
[7]	11.29 (0.97)	21.42 (1.07)	51.61 (1.05)	161.32 (1.21)
[8]	12.08 (1.03)	21.77 (1.09)	51.86 (1.06)	139.94 (1.05)
ours	10.59 (0.91)	18.46 (0.92)	36.59 (0.75)	89.2 (0.67)

TABLE I

PDP IN DIFFERENT DESIGNS USING CELL-BASED DESIGN FLOW

	design method	flexibility	power dissipation	transistor in one adding cell
[9]	full-custom	No	7.00 mW	28
this work	cell-based	Yes	18.46 mW	12

TABLE II

COMPARISON WITH OUR PRIOR DESIGN

IV. CONCLUSION

This paper proposed a low power digital multiplier design with 2-dimensional dynamic bypassing method by taking advantage of cell-based design flow which can extend any length of the operator and easily to integrate with other digital blocks. We justify the advantage in terms of power delay product after the software optimization. In this work, we gain more than 13.8% power saving compared to the prior designs.

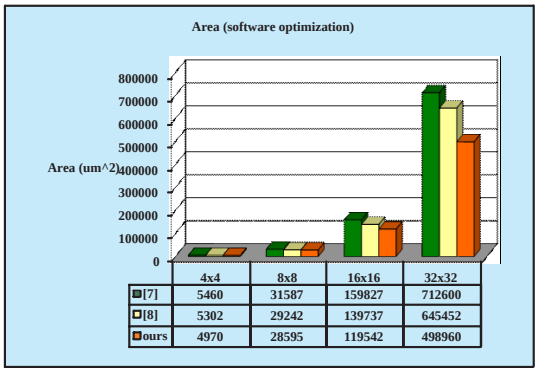


Fig. 8. Area comparison in different designs after software optimization

ACKNOWLEDGMENT

The authors would like to express their deepest gratefulness to CIC (Chip Implementation Center) of NAPL (National Applied Research Laboratories), Taiwan, for their thoughtful chip fabrication service. The authors also like to thank "Aim for Top University Plan" project of NSYSU and MOE, Taiwan, for partially supporting this investigation. And this research was partially supported by National Science Council under grant 96-2628-E-110-018-MY3 and National Health Research Institutes under grant NHRI-EX97-9732EI.

REFERENCES

[1] C.-C. Wang, C.-J. Huang, and K.-C. Tsai, "A 1.0 GHz 0.6-μm 8-bit carry lookahead adder using PLA-styled all-N-transistor logic," *IEEE Trans. of Circuits and Systems, Part II : Analog and Digital Signal Processing*, vol. 47, no. 2, pp. 133-135, Feb. 2000.

[2] W. Hwang, G. D. Gristede, P. N. Sanda, S. Y. Wang, and D. F. Heidel, "Implementation of a self-resetting CMOS 64-bit parallel adder with enhanced testability," *IEEE J. Solid-State Circuits*, vol. 34, no. 8, pp. 1108-1117, Aug. 1999.

[3] T. Ahn, and K. Choi, "Dynamic operand interchange for low power," *Electronics Letters*, vol. 33, no. 25, pp. 2118-2120, Dec. 1997.

[4] J. Choi, J. Jeon, and K. Choi, "Power minimization of functional units by partially guarded computation," *2000 International Symposium on Low Power Electronics and Design (ISLPED'00)*, pp. 131-136, July 2000.

[5] J. Di, J. S. Yuan, and M. Hagedorn, "Energy-aware multiplier design in multi-rail encoding logic," *The 2002 45th Midwest Symposium on Circuits and Systems (MWSCAS-2002)*, vol. 2, pp. 294-297, Aug. 2002.

[6] S. Hong, S. Kim, M. C. Papaefthymiou, and W. E. Stark, "Low power parallel multiplier design for DSP applications through coefficient optimization," *1999 Twelfth Annual IEEE International ASIC/SOC Conference*, pp. 286-290, Sep. 1999.

[7] J. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduction through bypassing of partial products," *2002 Asia-Pacific Conference on Circuits and Systems (APCCAS '02)*, vol. 2, pp. 13-17, Oct. 2002.

[8] W.-C. Wen, S.-J. Wang, and Y.-N. Lin, "Low power parallel multiplier with column bypassing," *Proc. 2005 IEEE Inter. Symp. on Circuits and Systems (ISCAS '05)*, vol. 2, pp. 1638-1641, May 2005.

[9] C.-C. Wang, and G.-N. Sung, "A low-power 2-dimensional bypassing multiplier using 0.35 um CMOS technology," *2006 IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI '06)*, vol. 0, pp. 405-410, Mar. 2006.