A Low-Power 2-Dimensional Bypassing Multiplier Using 0.35 um CMOS Technology

Chua-Chin Wang†, *Senior Member, IEEE*, Gang-Neng Sung
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 8×8 digital multiplier design by taking advantage of a 2-dimensional bypassing method. The proposed bypassing cells constituting the multiplier skip redundant signal transitions when the horizontally partial product or the vertically operand is zero. Hence, it is a 2-dimensional bypassing architecture. Thorough post-layout simulations show that the power dissipation of the proposed design is reduced by more than 75% compared to the prior design with obscure cost of delay and area. A physical implementation of the proposed design using a standard 0.35 μm 2P4M CMOS process is also presented to justify the functionality as well as the low power performance of the 2-dimensional bypassing method.**

Keywords : low power multiplier, bypassing, CMOS, partial product, timing control

I. Introduction

Booming of battery-operated multimedia 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, [7], to challenge the GHz operations, the major trade-off of these GHz logic circuits is the high power consumption which is not a tolerable price to pay in recent mobile technologies [4]. Besides adders, digital multipliers are the most critical arithmetic functional unit in many DSP applications, e.g., Fourier Transform, DCT, filtering, etc. Array and parallel multipliers are very welcomed due to their high execution speed and throughput. However, the increasing capacitive wire load and operands' bit length result in very large power dissipation, [1], [2], [3], [5], [6]. 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., $P_{\text{diss}} \propto f \cdot C \cdot V^2$, where C is the load, V denotes the voltage swing, and f is the frequency of switches.

Many prior digital multipliers were aimed at transition or switch reductions to reduce power dissipation. A leapfrog multiplier was proposed in [6] 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 th sign extension bits which are located at the left side of operands, e.g., [2]. What [5] proposed was close to a "bypassing" multiplier which skips the addition when the partial product of a row is zero. [1] 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. A 8×8 digital multiplier using the proposed 2-dimensional bypassing design is carried out by TSMC 0.18 μm 1P6M CMOS process. The post-layout simulations show that the power reduction compared to the prior multipliers is at least 75%.

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 typical array multiplication is based upon the following equation.

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

where P, $X = X_{n-1} \dots X_1 X_0$, $Y = Y_{n-1} \dots Y_1 Y_0$ are the product, the multiplier and the multiplicand, respectively. $P_k, k = 2n-1, \dots, 0$ denote the partial products, $X_i, i = n-1, \dots, 0$ and $Y_j, j = n-1, \dots, 0$ 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 which is given in

Proceedings of the 2006 Emerging VLSI Technologies and Architectures (ISVLSI'06)
0-7695-2533-4/06 $20.00 © 2006 IEEE

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 simple thought to improve the power efficiency was proposed by [6]. 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 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 second row and the second column are bypassed if we directly apply the prior 1-dimensional bypassing method. If the carry out of the adding cell AC_{02} is "1", it should be propagated to the carry in of AC_{21} and then its carry out. However, the carry bit will be lost if AC_{21} is bypassed due to $Y_1 = 0$. Consequently, an error is occurred, since the carry out of AC_{21} will be zero. We, thus, propose to include a bypass logic in certain adding cells.

C. Adding cell with bypass logic

According to the illustrative example, a simple rule is introduced. If $X_i = Y_j = 0$ and the carry out of $AC_{(i-2)(j+1)}$ is 1, then adding cell, AC_{ij} can not be bypassed. Hence, an adding cell with the bypass logic is proposed in Fig. 4. Notably, all of the 3 tri-state buffers as well as the two MUXs are gated by the output signals of the embedded bypass logic. Notably, an adding cell with the bypass logic is represented with a gray box in Fig. 3 and the other figures in this work.

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_{21} is the only unit with the necessity of a bypass logic. If $n = 5$ and the identical array structure is used, then $AC_{21}, AC_{22}, AC_{31}, AC_{32}$ 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 $n-2 \geq i \geq 2$ 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. 5. 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, $\forall n > 3$.

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_{22} is activated, then the following adding cell, AC_{32}, must be activated to ensure a carry (=1) is propagated correctly from the carry in of AC_{22} to the carry out of AC_{32} 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) 0.35 2P4M CMOS process was adopted to carry out the proposed design. Fig. 6 and Fig. 7, respectively, shows the layouts of a normal adding cell and an adding cell with bypass logic. The area penalty is 33% increase for a single adding cell. The power dissipation of the proposed 8×8 multiplier using 2-dimensional bypassing method has been simulated at all PVT corners (process transistor models, power supply voltage variations, and temperatures). The critical path delay is 13.0 ns. The simulations are carried out by HSPICE Monte Carlo method with sweep = 30. The outcome is tabulated in Table I.

Fig. 8 is the diephoto of the proposed 8×8 multiplier using 2-dimensional bypassing. The core of the chip is merely 363×208 μm^2. The physical measurement on silicon by Agilent 16702B Logic Analysis System is given in Fig. 9. A comparison of the proposed design with several prior 8×8 multiplier designs is summarized in Table II. It is obvious that the proposed design possesses the edge of low power.

IV. Conclusion

We have proposed a low power digital multiplier design by taking advantage of a 2-dimensional dynamic bypassing method. The post-layout simulations by HSPICE Monte Carlo method justify the advantage of the proposed design in terms of power dissipation. By a small area penalty, we gain more than 75% power saving compared to the prior designs. Physical implementation and measurement of the proposed design using a standard 0.35 μm 2P4M CMOS process also justify the functionality as well as the low power performance of the 2-dimensional bypassing method.

Acknowledgment

The authors would like to thank National Science Council (NSC), since this research was partially supported by NSC under grant 92-2220-E-110-001 and 92-2220-E-110-004.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] C.-C. Wang, Y.-L. Tseng, P.-M. Lee, R.-C. Lee, and C.-J. Huang, "A 1.25 GHz 32-bit tree-structured carry lookahead adder using modified ANT logic," *IEEE Trans. on Circuits and Systems – I Fundamental Theory and Applications,* vol. 50, no. 9, pp. 1208-1216, Sep. 2003.

SS model	TT model	FF model
10.719 mW	12.376 mW	14.980 mW

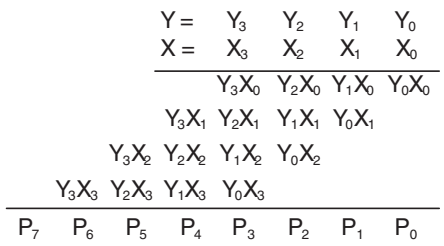
TABLE I

Power dissipation of the 8×8 multiplier using 2-dimensional bypassing @ 77 MHz data rate

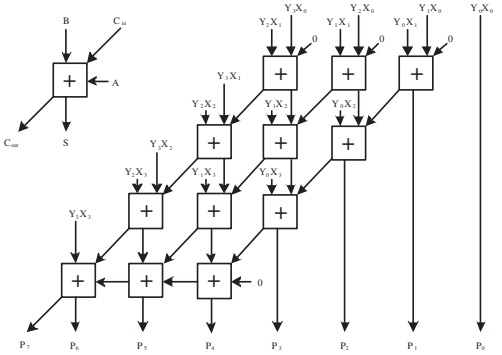
Braun's	100 mW
[6]	62 mW
[1]	67 mW
ours	47.91 mW (with pads power)

TABLE II

Comparison with the prior 8×8 multiplier designs



(a)



(b)

Fig. 1. Generic array multiplier

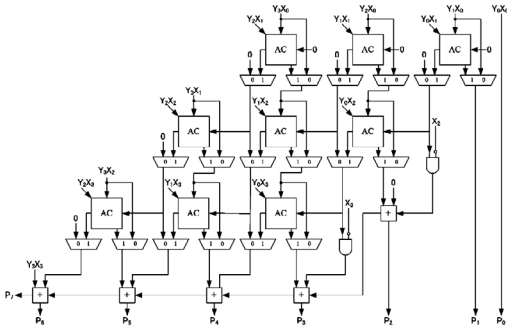


Fig. 2. Prior 1-dimensional bypassing multiplier design

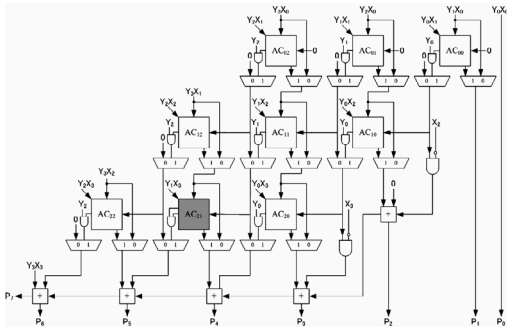


Fig. 3. Proposed 2-dimensional bypassing multiplier (4×4)

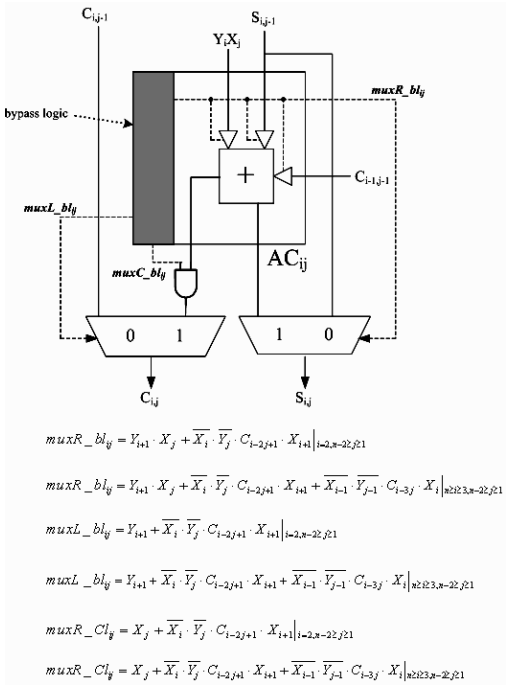


Fig. 4. Scehmatic of the bypass logic

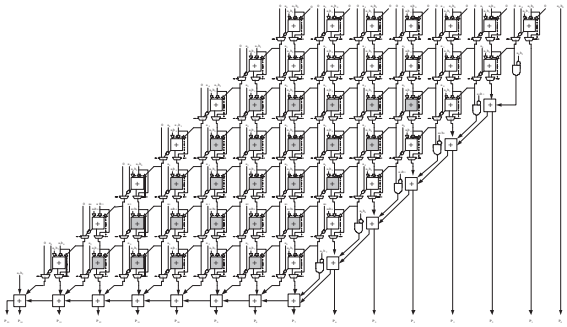


Fig. 5. A 8×8 multiplier using 2-dimensional bypassing

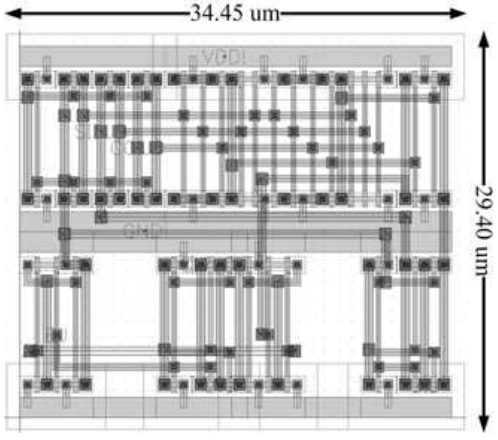


Fig. 6. Layout of a normal adding cell

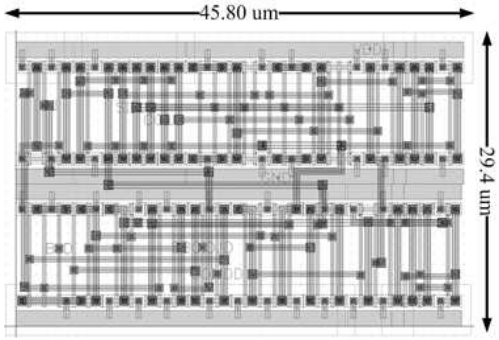


Fig. 7. Layout of the adding cell with bypass logic

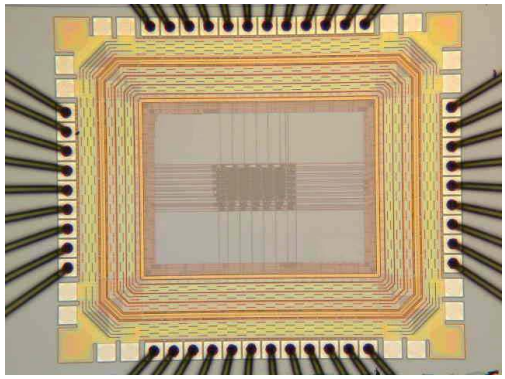


Fig. 8. Die photo of the 8 × 8 multiplier using 2-dimensional bypassing

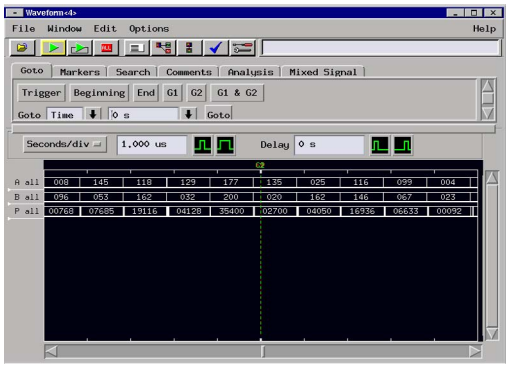


Fig. 9. Measurement of the low-power 8 × 8 multiplier on silicon