Low-Power Multiplier Design Using a Bypassing Technique

Chua-Chin Wang · Gang-Neng Sung

Received: 14 February 2007 / Revised: 4 February 2008 / Accepted: 5 November 2008 / Published online: 4 December 2008

Abstract This paper presents a low power digital multiplier design by taking advantage of a 2-dimensional bypassing method. The proposed bypassing cells constituting the multiplier skip redundant signal transitions as well as computations when the horizontally partial product or the vertical operand is zero. Hence, it is a 2-dimensional bypassing method. Thorough post-layout simulations of a 8×8 digital multiplier using the proposed 2-dimensional bypassing method show that the power dissipation of the proposed design is reduced by more than 75% compared to prior designs. Physical measurements on silicon reveal that the proposed digital multiplier saves more than 28% even with pads' power dissipation compared to the prior works.

Keywords Low power multiplier · Bypassing · Partial product · Timing control

1 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, [8, 9, 11], 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 [3, 10]. 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, 4–6]. 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 [5] 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., [1]. What [4] proposed was close to a "bypassing" multiplier which skips the addition when the partial product of a row is zero. [6] 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

This research was partially supported by National Science Council under grant NSC96-2628-E-110-018-MY3 and National Health Research Institutes under grant NHRI-EX97-9732EI.

C.-C. Wang (✉) · G.-N. Sung
Department of Electrical Engineering,
National Sun Yat-Sen University,
Kaohsiung, 80424, Taiwan
e-mail: ccwang@ee.nsysu.edu.tw

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 0.35 μm 2P4M CMOS process. The post-layout simulations show that the power reduction compared to the prior multipliers is at least 75%.

2 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 propose to detect the bitwise nullity of the multiplicand in the vertical direction and the partial product in the horizontal direction in an

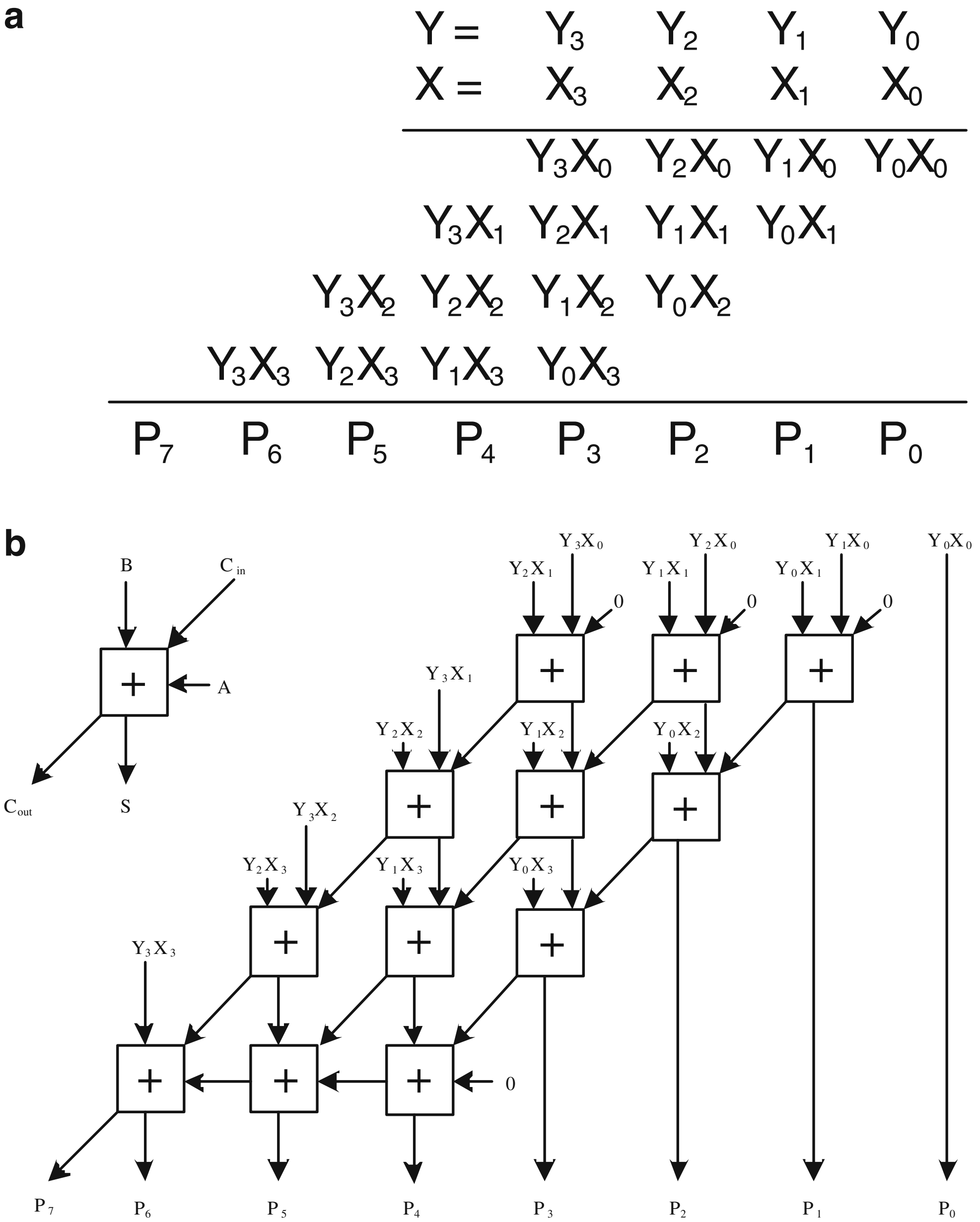


Figure 1 Generic array multiplier (**a**, **b**).

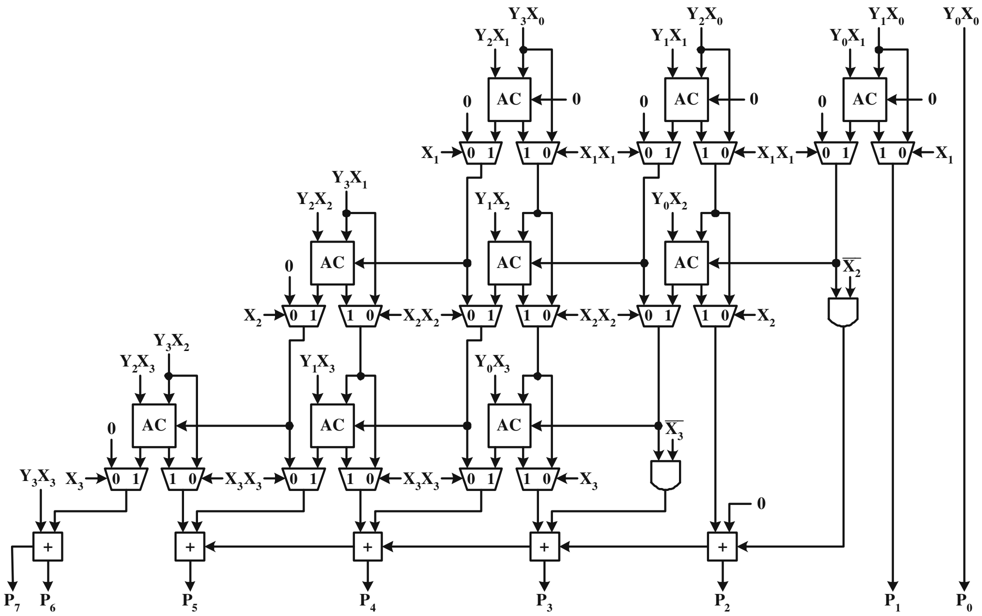


Figure 2 Prior 1-dimensional bypassing multiplier design (4×4).

array multiplier to remove the unnecessary operations taken place in the corresponding adding cells.

2.1 Prior 1-Dimensional Bypassing Algorithm

A typical array multiplication is based upon the following equation.

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

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

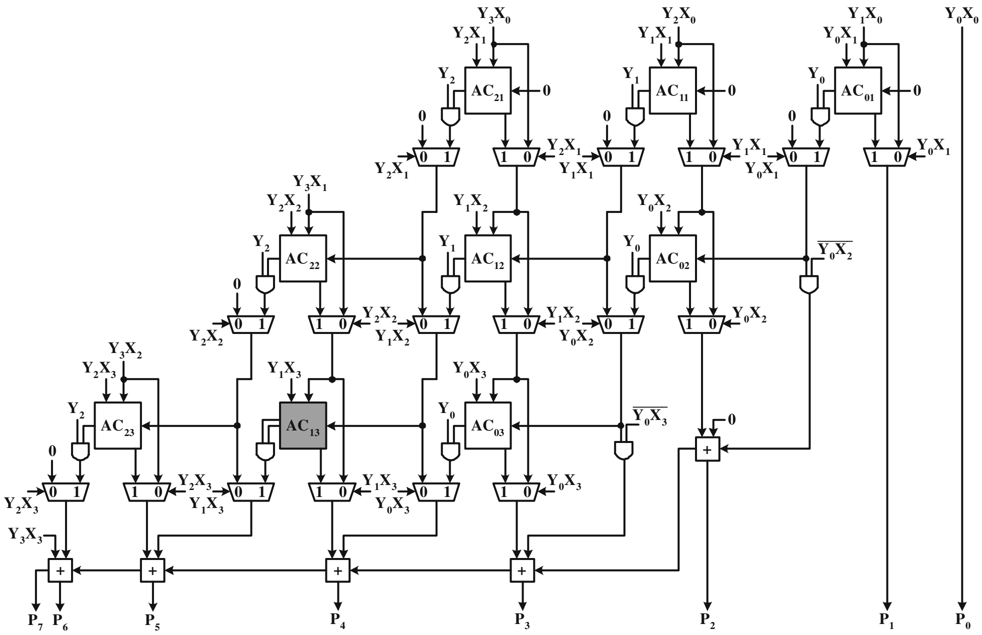


Figure 3 Proposed 2-dimensional bypassing multiplier (4×4).

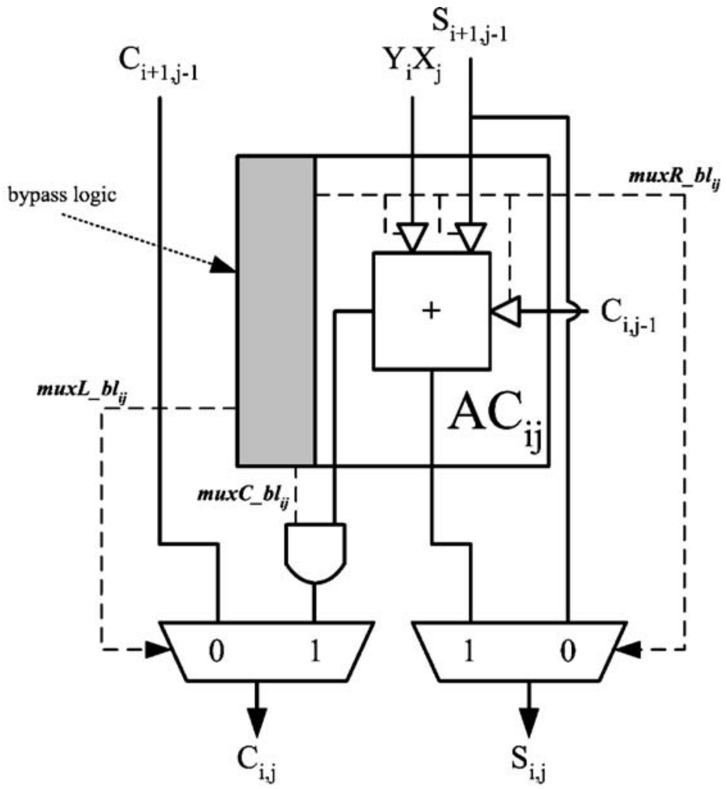


Figure 4 Schematic of the bypass logic.

$$muxR\_bl_{ij} = X_j \cdot Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j \,|_{i=1,j=3}$$

$$muxR\_bl_{ij} = X_j \cdot Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j + \overline{X_{j-2}} \cdot \overline{Y_i} \cdot C_{i,j-2} \,|_{n-3\geq i\geq 2, n-1\geq j\geq 4}$$

$$muxC\_bl_{ij} = Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j \,|_{i=1,j=3}$$

$$muxC\_bl_{ij} = Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i,j-2} \,|_{n-3\geq i\geq 2, n-1\geq j\geq 4}$$

$$muxL\_bl_{ij} = X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j \,|_{i=1,j=3}$$

$$muxL\_bl_{ij} = X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i,j-2} \,|_{n-3\geq i\geq 2, n-1\geq j\geq 4}$$

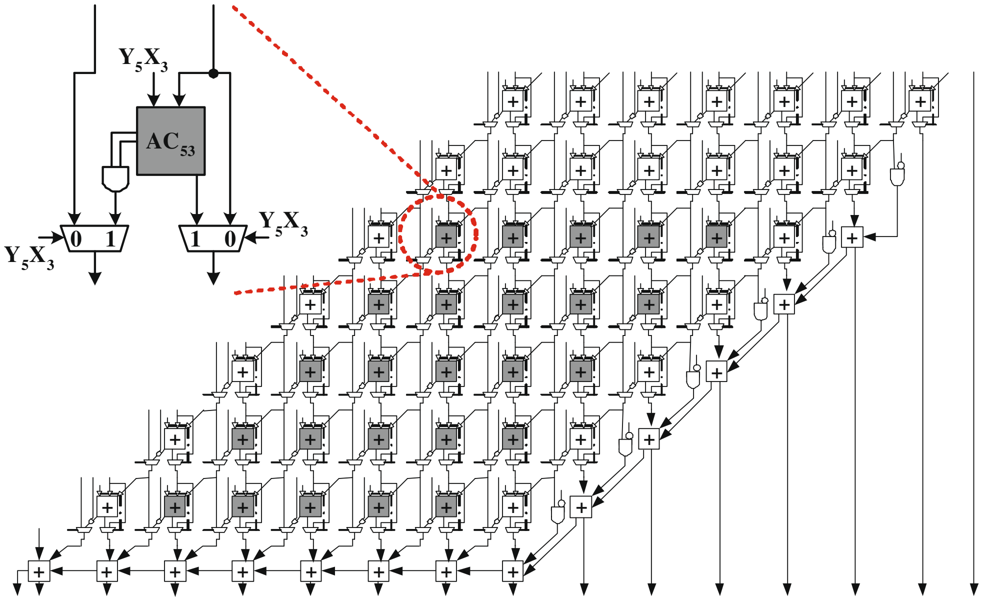


Figure 5 A 8×8 multiplier using 2-dimensional bypassing.

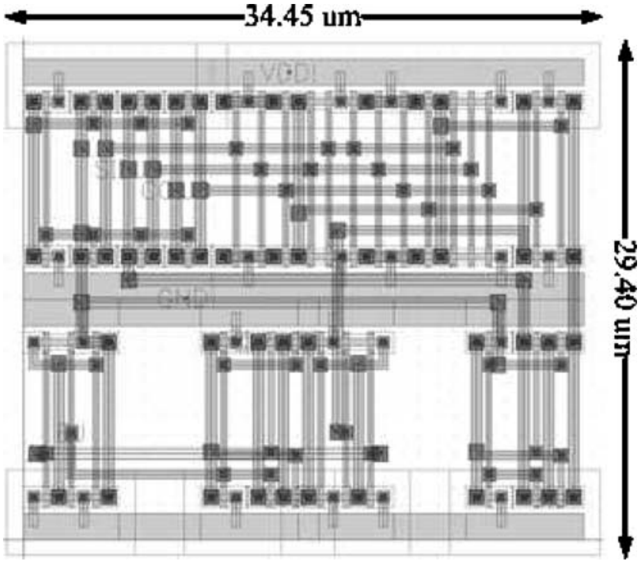


Figure 6 Layout of a normal adding cell.

A simple thought to improve the power efficiency was proposed by [5]. As soon as X_j 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, where the control signal of each MUX in the same row is X_j.

2.2 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_i's, in addition to the state of the multiplier, X_j's. Notably, the input control signal of each MUX is one summand of each adding cell, namely the bitwise product of Y_i and X_j. In other words, as soon as the Y_i 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 $X_j = Y_i = 0$.

For instance, assume $j = 2, i = 1$ 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_{12} is "1", it should be propagated to the carry in of AC_{13} and then its carry out. However, the carry bit will be lost if AC_{13} is bypassed due to $Y_1 = 0$. Consequently, an error is occurred, since the carry out of AC_{13} will be zero. We, thus, propose to include a bypass logic in certain adding cells.

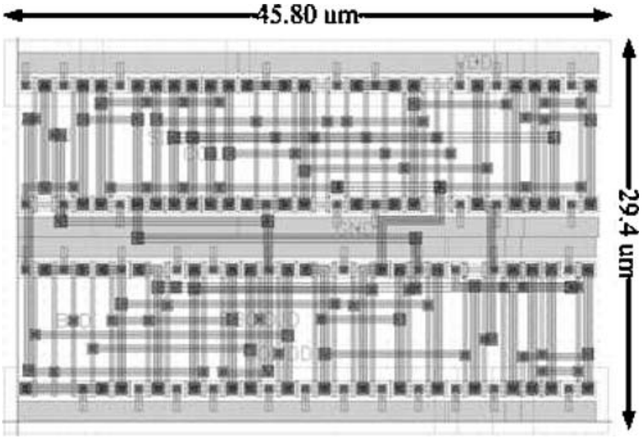


Figure 7 Layout of the adding cell with bypass logic.

Table 1 Power dissipation of the 8 × 8 multiplier using 2-dimensional bypassing @ 77 MHz data rate.

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

2.3 Adding Cell with Bypass Logic

According to the illustrative example, a simple rule is introduced. If $X_{j-1} = Y_i = 0$ and the carry out of $AC_{(i+1)(j-2)}$ is "1" and $X_j = 1$, then adding cell, AC_{ij}

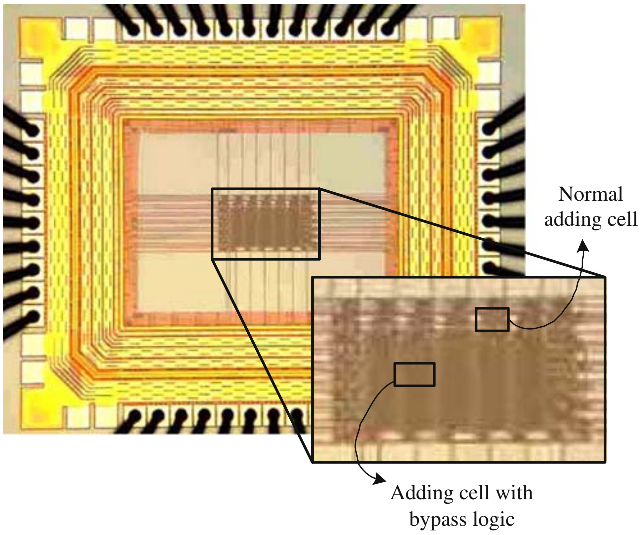
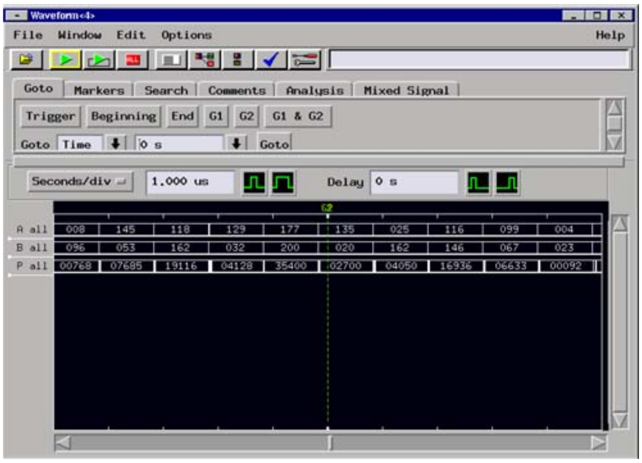


Figure 8 Die photo of the 8 × 8 multiplier using 2-dimensional bypassing.



```
Goto  Time  0 s  Goto
Seconds/div  1.000 us    Delay  0 s
A all  008   145   118   129   177   135   025   116   099   004
B all  096   053   162   032   200   020   162   146   067   023
P all  00768 07685 19116 04128 35400 02700 04050 16936 06633 00092
```

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

can not be bypassed. Hence, an adding cell with the bypass logic is proposed in Fig. 4. The logic equations in Fig. 4 are analyzed as follows.

Let's use an example to derive the control signal, $muxR\_bl_{13}$. If $X_3 = 1$ and $Y_1 = 0$, then suppose the AC_{13} should be bypass the input signal. However, when $X_2 = 0$ and the carry out of AC_{21} is "1", then AC_{13} should be turned on to avoid the carry out of AC_{21} missing. Therefore, the control signal of right hand side MUX, $muxR\_bl_{13}$, should be set to "0". In shorts, if $X_2 = 0$, $Y_1 = 0$, $X_3 = 1$, and when the carry out of AC_{21} is "1" then $muxR\_bl_{13}$ should be set to "1". By the same derivation, we can get the logic equation: $muxR\_bl_{ij} = X_j \cdot Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$.

If $X_j = 0$, then we should let the control signal of left hand side MUX, $muxL\_bl_{ij}$, set to "0" to execute the bypass condition. Similarly, if $X_{j-1} = 0$, $Y_i = 0$, $X_j = 1$, and the carry out of $AC_{(i+1)(j-2)} =$ "1", then we cannot let the bypass condition execut. Hence, the control signal $muxL\_bl_{ij}$ should be set to "1". Therefore, we can get the logic equation: $muxL\_bl_{ij} = X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$.

The control signal $muxC\_bl_{ij}$ is used to pass the carry out signal of AC_{ij} quickly. If $Y_i = 0$ and the carry out of $AC_{(i+1)(j-2)}$ is "0", then the carry out signal of AC_{ij} should be "0". Thus, set the control signal $muxC\_bl_{ij}$ is set to "0" right away. Hence, we can get the logic equation: $muxC\_bl_{ij} = Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$.

If the proposed design is expended to be an $n \times n$ multiplier, every adding cell, AC_{ij}, with bypass logic should consider the state of the carry out of it's predecessor cell. The logic equation in the bypass logic is summarized as follows.

$$muxR\_bl_{ij} = X_j \cdot Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$$

$$muxR\_bl_{ij} = X_j \cdot Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j + \overline{X_{j-2}} \cdot \overline{Y_i} \cdot C_{i,j-2}|_{n-3 \geq i \geq 2, n-1 \geq j \geq 4}$$

$$muxL\_bl_{ij} = X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$$

$$muxL\_bl_{ij} = X_j + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j \cdot \overline{Y_i} \cdot C_{i,j-2}|_{n-3 \geq i \geq 2, n-1 \geq j \geq 4}$$

$$muxC\_bl_{ij} = Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j|_{i=1,j=3}$$

$$muxC\_bl_{ij} = Y_i + \overline{X_{j-1}} \cdot \overline{Y_i} \cdot C_{i+1,j-2} \cdot X_j \cdot \overline{Y_i} \cdot C_{i,j-2}|_{n-3 \geq i \geq 2, n-1 \geq j \geq 4}$$

Notably, all of the 3 tri-state buffers as well as the two MUXs are gated by the output signals of the 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.

2.4 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_{13} is the only unit with the necessity of a bypass logic. If $n = 5$ and the identical array structure is used, then AC_{13}, AC_{23}, AC_{14}, AC_{24} 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 $n - 3 \geq i \geq 1$ and $n - 1 \geq j \geq 3$, 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

Table 2 Comparison with the prior 8 × 8 multiplier designs (PDP = power delay product).

Design	Power (mW)	Delay (ns)	PDP ($10^{-10} J$)	Process (μm)	Area (μm^2)
Braun's	69	N/A	N/A	0.35	33,286
[5]	64	N/A	N/A	0.35	48,991
[6]	67	N/A	N/A	0.35	N/A
[7]	18.06	16.00	2.8896	0.35	111,997
Ours	16.3132	14.28	2.3295	0.35	75,504

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.

$$1 = (4-3)\times(4-3), \quad n = 4$$

$$4 = (5-3)\times(5-3), \quad n = 5$$

$$9 = (6-3)\times(6-3), \quad n = 6$$

$$\vdots = \vdots$$

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.

3 Simulation and Implementation

TSMC (Taiwan Semiconductor Manufacturing Company) 0.35 2P4M CMOS process was adopted to carry out the proposed design. Figures 6 and 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 1.

Figure 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. The testing vectors are generated by the pattern generator of Agilent 16702B using a PRSG (pseuo-random number sequence generator) with a 77 MHz data rate. A comparison of the proposed design with several prior 8 × 8 multiplier designs is summarized in Table 2. It is obvious that the proposed design possesses the edge of low power.

4 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 some 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.

Acknowledgement The authors would like to express their deepest gratefulness to CIC of NSC 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.

References

1. Choi, J., Jeon, J., & Choi, K. (2000). Power minimization of functional units by partially guarded computation. In *2000 International symposium on low power electronics and design (ISLPED'00)* pp. 131–136, July.
2. Di, J., Yuan, J. S., & Hagedorn, M. (2002). Energy-aware multiplier design in multi-rail encoding logic. In *The 2002 45th Midwest symposium on circuits and systems (MWSCAS-2002)* (Vol. 2, pp. 294–297), Aug.
3. Hwang, W., Gristede, G. D., Sanda, P. N., Wang, S. Y., & Heidel, D. F. (1999). Implementation of a self-resetting CMOS 64-bit parallel adder with enhanced testability. *IEEE Journal of Solid-State Circuits, 34*(8), 1108–1117.
4. Hong, S., Kim, S., Papaefthymiou, M. C., & Stark, W. E. (1999). Low power parallel multiplier design for DSP applications through coefficient optimization. In *1999 twelfth annual IEEE international ASIC/SOC conference* pp. 286–290, Sep.
5. Ohban, J., Moshnyaga, V. G., & Inoue, K. (2002). Multiplier energy reduction through bypassing of partial products. In *2002 Asia-Pacific conference on circuits and systems (APCCAS '02)* (Vol. 2, pp. 13–17), Oct.
6. Ahn, T., & Choi, K. (1997). Dynamic operand interchange for low power. *Electronics Letters, 33*(25), 2118–2120.
7. Shen, N.-Y., & Chen, O. T.-C. (2002). Low-power multipliers by minimizing switching activities of partial products. In *IEEE international symposium on circuits and systems, 2002 (ISCAS 2002)* (Vol. 4, pp. IV-93-IV-96), May.
8. Wang, C.-C., Huang, C.-J., & Tsai, K.-C. (2000). A 1.0 GHz 0.6-μm 8-bit carry lookahead adder using PLA-styled all-N-transistor logic. *IEEE Transactions on Circuits and Systems, Part II: Analog and Digital Signal Processing, 47*(2), 133–135, Feb.

9. Kang, J.-Y., & Gaudiot, J.-L. (2006) A simple high-speed multiplier design. *IEEE Transactions on Computers, 55*(10), 1253–1258.
10. Wang, C.-C., Tseng, Y.-L., Lee, P.-M., Lee, R.-C., & Huang, C.-J. (2003). A 1.25 GHz 32-bit tree-structured carry lookahead adder using modified ANT logic. *IEEE Transactions on Circuits and Systems—I Fundamental Theory and Applications, 50*(9), 1208–1216.
11. Islam, F. F., & Tamaru, K. (1995). High speed merged array multiplication. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Technology, 10*(1), 41–52.

Chua-Chin Wang was born in Taiwan in 1962. He received the B.S. degree in electrical engineering from National Taiwan University in 1984, and the M.S. and Ph.D. degree in electrical engineering from State University of New York in Stony Brook in 1988 and 1992, respectively. He then joined the Department of Electrical Engineering, National Sun Yat-Sen University, Taiwan, and became a full professor since 1998. His recent research interests include mixed-signal circuit design, low-power and high-speed circuit design, communication interfacing circuitry, and bio-chips. He founded SOC group in Department of Electrical Engineering, National Sun Yat-Sen University during 2005. He is currently also serving as the Director of Engineering Technology Research & Promotion Center (ETRPC), National Sun Yat-Sen University. He is the Chair of IEEE Circuits and Systems Society (CASS), Tainan Chapter. He is also the founding Chair of IEEE Solid-State Circuits Society (SSCS), Tainan Chapter, and the founding Councilor of IEEE NSYSU Student Branch. He is also a member of the IEEE CASS Multimedia Systems & Applications (MSA), VLSI Systems and Applications (VSA), Nanoelectronics and Giga-scale Systems (NG), and Biomedical Circuits and Systems (BioCAS) Technical Committees. He is a senior member of IEEE since 2004. Currently he is also serving as the Associate Editor of International Journal of VLSI Design, and IEICE Trans. on Electronics. He is also a Guest Editor of International Journal of Electrical Engineering, and Journal of Signal Processing Systems. He is IEEE CASS Nanoelectronics and Giga-scale Systems (NG) Technical Committee Chair term from 2008 to 2009. He was the General Chair of 2007 VLSI/CAD Symposium.

Gang-Neng Sung was born in Taiwan in 1981. He received the B.S. degree in Department of Computer and Communication Engineering in National Kaohsiung First University of Science and Technology in 2004, and the M.S. degree in Department of Electrical Engineering in National Sun Yat-Sen University in 2006. He is currently working toward the Ph.D. in the Department of Electrical Engineering National Sun Yat-Sen University, Kaohsiung, Taiwan. His recent research interests include VLSI design and low power design.