IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 6, Issue 5 (Jul. - Aug. 2013), PP 12-19
www.iosrjournals.org

A Spurious-Power Suppression technique for a Low-Power Multiplier

Kalpana P, Ch.Ramesh
E C E. Ganapathy Engineering College,Warangal, JNTUH, Andhra Pradesh, INDIA
Assoc.Prof. E C E. Ganapathy Engineering College, Warangal, JNTUH, Andhra Pradesh, INDIA

Abstract: *This paper presents the design exploration of a spurious-power suppression technique (SPST) which can dramatically reduce the power dissipation of combinational VLSI designs for multimedia/DSP purposes. The proposed SPST separates the target designs into two parts, i.e., the most significant part and least significant part (MSP and LSP), and turns off the MSP when it does not affect the computation results to save power.*

The objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip. To save significant power consumption of a VLSI design, it is a good direction to reduce its dynamic power that is the major part of total power dissipation. In this paper, we propose a high speed low-power multiplier adopting the new SPST implementing approach. This multiplier is designed by equipping the Spurious Power Suppression Technique (SPST) on a modified Booth encoder which is controlled by a detection unit using an AND gate. The modified booth encoder will reduce the number of partial products generated by a factor of 2. The SPST adder will avoid the unwanted addition and thus minimize the switching power dissipation.

Keywords-*Booth encoder; low power; spurious power suppression technique(SPST); SPST-Adder.*

Submitted Date 17 June 2013 **Accepted Date: 22 June 2013**

I. INTRODUCTION

Power dissipation is recognized as a critical parameter in modern VLSI design field. To satisfy MOORE'S law and to produce consumer electronics goods with more backup and less weight, low power VLSI design is necessary.

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, Subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them.

II. PROPOSED LOW-POWER MULTIPLIER

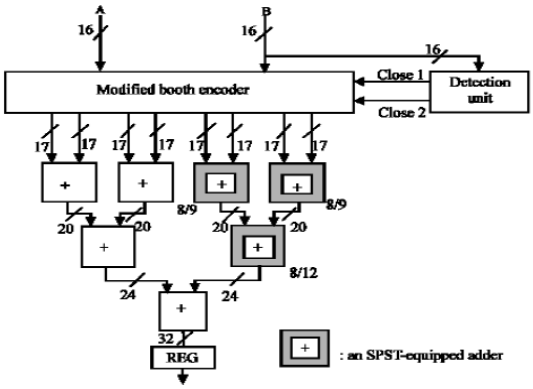

Figure 1. Proposed high performance low power equipped multiplier

The proposed SPST-equipped multiplier is illustrated in fig-1.

III. PARTIAL PRODUCT GENERATION

A. Modified Booth Encoder

In order to achieve high-speed multiplication, multiplication algorithms using parallel counters, such as the modified Booth algorithm has been proposed, and some multipliers based on the algorithms have been implemented for practical use. This type of multiplier operates much faster than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands.

B. Modified Booth's algorithm

- Insert 0 on the right side of LSB of multiplier.
- Divide the multiplier into overlapping groups of 3-bits.
- If the number of multiplier bits is odd, add an extra 1 bit on left side of MSB.
- Determine partial product scale factor from modified booth encoding table.
- Compute the Multiplicand Multiples
- Sum Partial Products.
- When new partial product is generated, each partial product is added 2 bit left shifting in regular sequence.

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is possible to reduce the number of partial products by half, by using the technique of modified Booth recoding algorithm. The basic idea is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by ±1, ±2, or 0, to obtain the same results. The advantage of this method is the halving of the number of partial products. To recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Fig-2 shows the grouping of bits from the multiplier term for use in modified booth encoding.

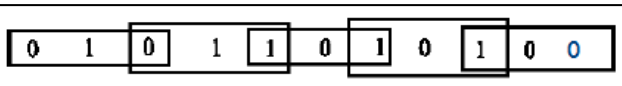

Figure 2. Grouping of bits from the multiplier term

The encoding of the multiplier Y, using the modified booth

Block	Re - coded digit	Operation on X
000	0	0 X
001	+1	+1 X
010	+1	+1 X
011	+2	+2 X
100	-2	-2 X
101	-1	-1 X
110	-1	-1 X
111	0	0 X

Table-1

algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table-1.

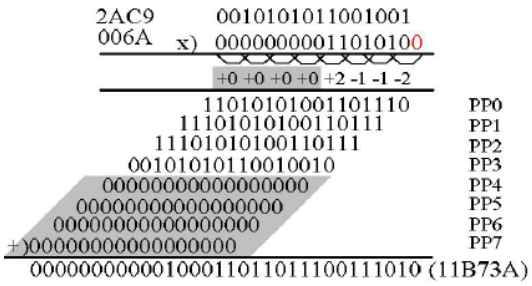

Figure 3. Illustration of multiplication using modified Booth encoding.

Fig-3 shows a computing example of Booth, multiplying two numbers, "2AC9" and "006A". The shadow denotes that the numbers in this part of Booth multiplication are all zero so that this part of the computations can be neglected. Saving those computations can significantly reduce the power consumption caused by the transient signals.

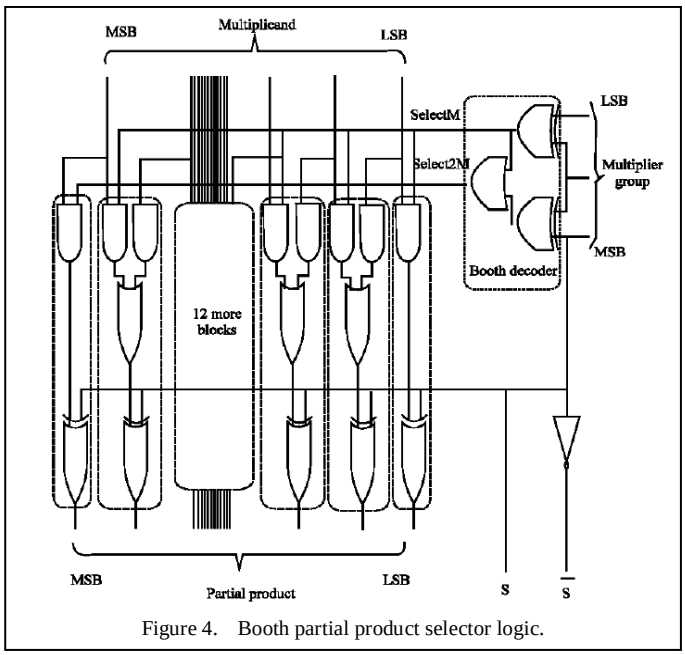

Figure 4. Booth partial product selector logic.

According to the analysis of the multiplication shown in fig-3, we propose the SPST-equipped modified-Booth encoder, which is controlled by a detection unit. The detection unit has one of the two operands as its input to decide whether the Booth encoder calculates redundant computations.

From one of the two operands, e.g., the operand A, the partial product (PP) candidate generator generates five candidates of the partial products, i.e., -2A, -A, 0, A, 2A, which are then selected according to the Booth encoding results of the other operand, i.e., the operand B. Meanwhile, the detection unit has the second one of the two operands, i.e., the operand B in this case, as its input to decide whether the Booth encoder includes redundant computations. As shown in Fig-5, the latches can, respectively, freeze the inputs of MUX-4 to MUX-7 or only those of MUX-6 to MUX-7 when the PP4 to PP7 or only the PP6 to PP7 are zeros to reduce the transition power dissipation. Such cases occur frequently in wireless multimedia-data coding like texture coding, orthogonal frequency-division multiplexing, and filter designs.

$$A_{MSP} = A[15:8];\ B_{MSP} = B[15:8] \quad (1)$$

$$A\ and = A[15].A[14].....A[8] \quad (2)$$

$$B\ and = B[15].B[14].....B[8] \quad (3)$$

$$Anor = \overline{A[15] + A[14] + ... + A[8]} \quad (4)$$

$$Bnor = \overline{A[15] + A[14] + ... + A[8]} \quad (5)$$

$$Close = (Aand + Anor).(Band + Bnor) \quad (6)$$

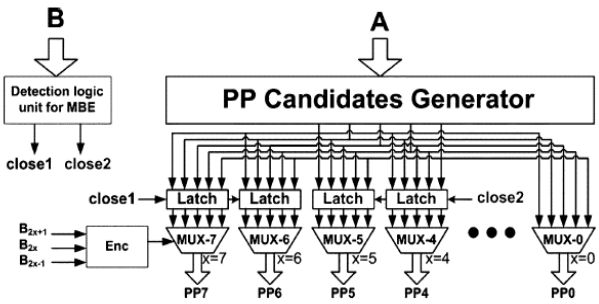

Figure 5. SPST equipped modified Booth encoder

Fig-4 shows the booth partial product generation circuit. It includes AND/OR/EX-OR logic.

IV. PROPOSED SPST

We explore five cases of 16-bit additions as shown in Fig-6. The cases of exchanging the operands A and B in additions lead to the same spurious transitions with those shown in Fig-6.

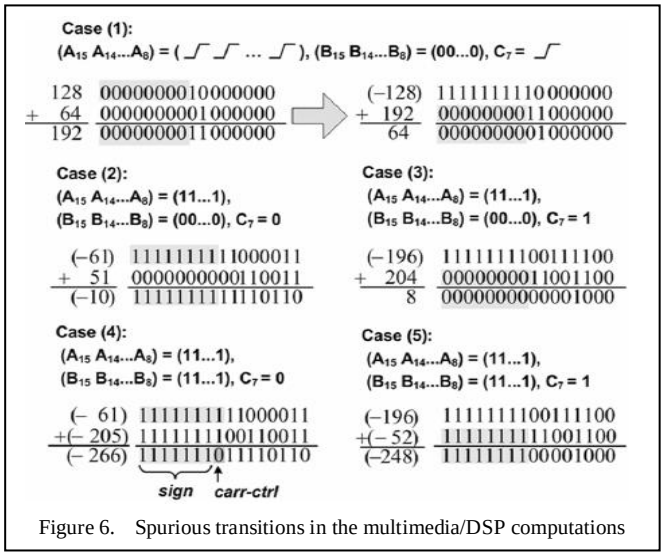

Figure 6. Spurious transitions in the multimedia/DSP computations

Hence, there is probably no other case beyond these five based on this design. The first case illustrates a transient state in which spurious transitions of carry signals occur in the MSP, although the final result of the MSP is unchanged. Meanwhile, the second and third cases describe situations involving one negative operand adding another positive operand without and with carry-in from the LSP, respectively. Moreover, the fourth and fifth cases demonstrate the addition of two negative operands without and with carry-in from the LSP, respectively.

In those cases, the results of MSP are predictable; therefore, the computations in MSP are useless and can be neglected. Eliminating those spurious computations not only can save the power consumption inside the adder/subtractor in the current stage but also can decrease the glitching noises which cause power wastage inside the arithmetic circuits in the next stage. From the analysis of Fig-6, we are motivated to propose the SPST that separates the adder/subtractor into two parts and then latches the input data of the MSP whenever they do not affect the computation results.

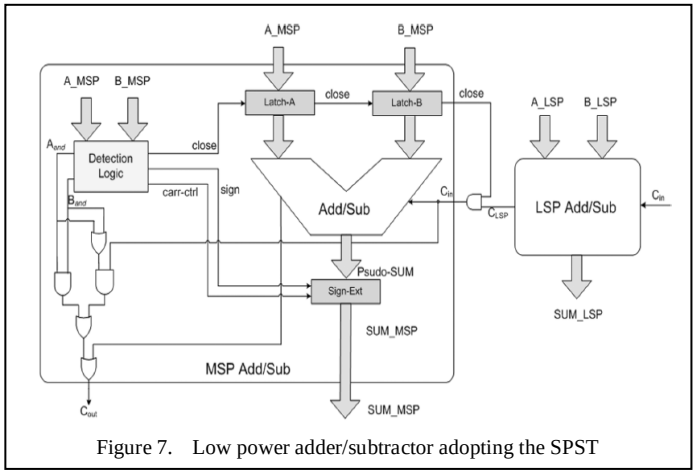

Figure 7. Low power adder/subtractor adopting the SPST

$$
\begin{aligned}
carr-ctrl = {} & \overline{C_{\mathrm{LSP}}} \times \overline{A_{\mathrm{and}}} \times A_{\mathrm{nor}} \times B_{\mathrm{and}} \times \overline{B_{\mathrm{nor}}} \\
& + \overline{C_{\mathrm{LSP}}} \times A_{\mathrm{and}} \times \overline{A_{\mathrm{nor}}} \times \overline{B_{\mathrm{and}}} \times B_{\mathrm{nor}} \\
& + C_{\mathrm{LSP}} \times \overline{A_{\mathrm{and}}} \times A_{\mathrm{nor}} \times \overline{B_{\mathrm{and}}} \times B_{\mathrm{nor}} \\
& + C_{\mathrm{LSP}} \times A_{\mathrm{and}} \times \overline{A_{\mathrm{nor}}} \times B_{\mathrm{and}} \times \overline{B_{\mathrm{nor}}} \\
= {} & \overline{C_{\mathrm{LSP}}} \times (\overline{A_{\mathrm{and}}} \times B_{\mathrm{and}} + A_{\mathrm{and}} \times \overline{B_{\mathrm{and}}}) \\
& \times (A_{\mathrm{and}} \times B_{\mathrm{and}} + A_{\mathrm{and}} \times B_{\mathrm{nor}} + A_{\mathrm{nor}} \times B_{\mathrm{and}} \\
& + A_{\mathrm{nor}} \times B_{\mathrm{nor}}) + C_{\mathrm{LSP}} \\
& \times (A_{\mathrm{and}} \times B_{\mathrm{and}} + \overline{A_{\mathrm{and}}} \times \overline{B_{\mathrm{and}}}) \\
& \times (A_{\mathrm{and}} \times B_{\mathrm{and}} + A_{\mathrm{and}} \times B_{\mathrm{nor}} + A_{\mathrm{nor}} \times B_{\mathrm{and}} \\
& + A_{\mathrm{nor}} \times B_{\mathrm{nor}})(C_{\mathrm{LSP}} \oplus A_{\mathrm{and}} \oplus B_{\mathrm{and}}) \\
& \times (A_{\mathrm{and}} + A_{\mathrm{nor}}) \times (B_{\mathrm{and}} + B_{\mathrm{nor}}) \\
= {} & (C_{\mathrm{LSP}} \oplus A_{\mathrm{and}} \oplus B_{\mathrm{and}}) \times (A_{\mathrm{and}} + A_{\mathrm{nor}}) \\
& \times (B_{\mathrm{and}} + B_{\mathrm{nor}}) \qquad (7)
\end{aligned}
$$

*A. **SPST equipped Adder/Subtractor***

A 16-bit adder/subtractor design based on the proposed SPST is shown in the Fig-7. In this, the 16-bit adder/subtractor is divided into MSP and LSP at the place between the 8th bit and the 9th bit. The adder subtractor is divided into two parts, the most significant part (MSP) and the least significant part (LSP). The MSP of the original adder/subtractor is modified to include detection logic circuits, data controlling circuits, sign extension circuits, logics for calculating carry in and carry out signals. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain the

same as usual, while the MSP is negligible, the input data of the MSP become zeros to avoid switching power consumption.

To know whether the MSP affects the computation results or not, we need a detection logic unit to detect the effective ranges of the inputs. The Boolean logical equations shown below express the behavioral principles of the detection logic unit in the MSP circuits of the SPST-based adder/subtractor:

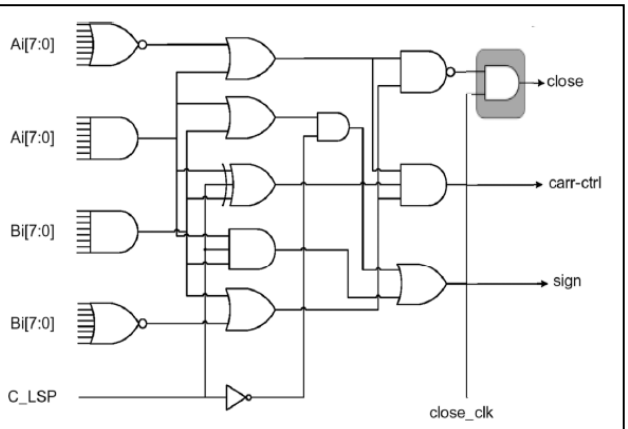

Figure 8. Detection logic circuits using an AND gate

$$
\begin{aligned}
sign = {} & \overline{C_{LSP}} \times \left(\overline{A_{and}} \times A_{nor} \times B_{and} \times \overline{B_{nor}} + A_{and} \right. \\
& \left. \times \overline{A_{nor}} \times \overline{B_{and}} \times B_{nor} + A_{and} \times \overline{A_{nor}} \right. \\
& \left. \times B_{and} \times \overline{B_{nor}} \right) \\
& + C_{LSP} \times A_{and} \times \overline{A_{nor}} \times B_{and} \times \overline{B_{nor}} \\
= {} & \overline{C_{LSP}} \times \left(\overline{A_{and}} \times B_{and} + A_{and} \right) \\
& + C_{LSP} \times A_{and} \times B_{and} \\
= {} & \overline{C_{LSP}} \times \left(A_{and} + B_{and} \right) + C_{LSP} \times A_{and} \times B_{and} \qquad (8)
\end{aligned}
$$

where A[m] and B[n] respectively denote the mth bit of the operands A and the nth bit of the operand B, and AMSP and BMSP respectively denote the MSP parts, i.e. the 9th bit to the 16th bit, of the operands A and B. When the bits in AMSP and/or those in BMSP are all ones, the value of Aand and/or that of Band respectively become one, while the bits in AMSP and/or those in BMSP are all zeros, the value of Anor, and/or that of Bnor respectively turn into one. Being one of the three outputs of the detection logic unit, close denotes whether the MSP circuits can be neglected or not. When the two input operand can be classified into one of the five classes as shown in fig-6, the value of close becomes zero, which indicates that the MSP circuits can be closed to save power dissipation. This design intends to close the MSP circuits by feeding zero inputs into them, which may freeze the switching activities in the MSP circuits to avoid dynamic power consumption. The ways to compensate for the sign bits of the computing results are also shown in case 4 in Fig-6. In equation (7) and (8), CLSP denotes the carry propagated from the LSP circuits.

From the derived Boolean equations (1) to (8), the detection logic unit of the SPST is designed as shown in fig-8, which can determine whether the input data of MSP should be latched or not. Moreover, we add three 1-bit to control the assertion of the close, sign, and Carr-ctrl signals in order to further decrease the glitch signals occurred in the cascaded circuits.

Based on Figs -7 and 8, the timing issue of the SPST is analyzed as follows.

1) *When the detection-logic unit turns off the MSP:* At this moment, the outputs of the MSP are directly compensated by the SE unit; therefore, the time saved from skipping the computations in the MSP circuits shall cancel out the delay caused by the detection-logic unit.
2) *When the detection-logic unit turns on the MSP:* The MSP circuits must wait for the notification of the detection-logic unit to turn on the data latches to let the data in. Hence, the delay caused by the detection-

logic unit will contribute to the delay of the whole combinational circuitry, i.e., the 16-bit adder/subtractor in this design example.

3) *When the detection-logic unit remains its decision:* No matter whether the last decision is turning on or turning off the MSP, the delay of the detection logic is negligible because the path of the combinational circuitry (i.e., the 16-bit adder/subtractor in this design example) remains the same.

V. SIMULATION RESULTS AND DISCUSSIONS

In this project we are examining the performance of the proposed high speed low power multiplier. This multiplier can be implemented using Verilog-HDL coding. In order to get the power report and delay report we are synthesizing this multiplier using Xilinx. Simulation results correspond to the multiplier are given in fig-10 and 11.

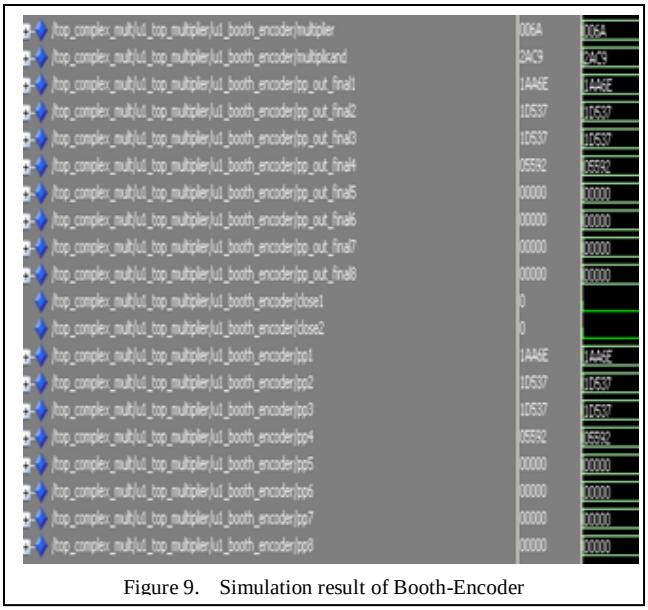

Figure 9. Simulation result of Booth-Encoder

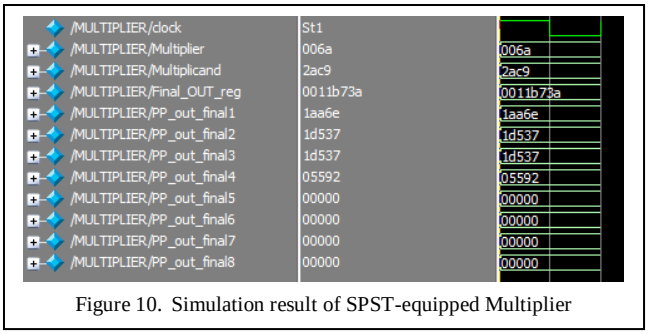

Figure 10. Simulation result of SPST-equipped Multiplier

VI. CONCLUSION:

This work presents the designing of a 16x16 multiplier with low-power technique called SPST. A Modified Booth Encoder circuit is used for this Multiplier architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count.

The presented low-power technique called SPST and the theoretical analysis of the SPST are fully discussed. The proposed SPST can obviously decrease the switching (or dynamic) power dissipation, which comprises a significant portion of the whole power dissipation in integrated circuits. And simulation results for Booth Encoder and SPST-adder equipped Multiplier are shown.

REFERENCES

[1] K. H. Chen, K. C. Chao, J. I. Guo, J. S. Wang, and Y. S. Chu, "Design exploration of a spurious power suppression technique (SPST) and its applications," in Proc. IEEE Asian Solid-State Circuits Conf., Hsinchu, Taiwan, Nov. 2005, pp. 341–344.

[2] A. Bellaouar and M. I. Elmasry, "Low-Power Digital VLSI Design" Circuits and Systems. Norwell, MA: Kluwer, 1995.

[3] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," Proc. IEEE, vol. 83, no. 4, pp. 498–523, Apr. 1995.

[4] K. K. Parhi, "Approaches to low-power implementations of DSP systems," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 48, no. 10, pp. 1214–1224, Oct. 2001.

[5] J. Choi, J. Jeon, and K. Choi, "Power minimization of functional units by partially guarded computation," in Proc. IEEE Int. Symp. Low Power Electron. Des., 2000, pp. 131–136.
[6] O. Chen, R. Sheen, and S. Wang, "A low-power adder operating on effective dynamic data ranges," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 4, pp. 435–453, Aug. 2002.
[7] O. 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.
[8] S. Henzler, G. Georgakos, J. Berthold, and D. Schmitt-Landsiedel, "Fast power-efficient circuit-block switch-off scheme," Electron. Lett., vol. 40, no. 2, pp. 103–104, Jan. 2004.
[9] L. Benini, G. D. Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Glitch power minimization by selective gate freezing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 8, no. 3, pp. 287–298, Jun. 2000.
[10] S. Henzler, G. Georgakos, J. Berthold, and D. Schmitt-Landsiedel, "Fast power-efficient circuit-block switch-off scheme," Electron. Lett., vol. 40, no. 2, pp. 103–104, Jan. 2004.
[11] Z. Huang and M. D. Ercegovac, "High-performance low-power left-toright array multiplier design," IEEE Trans. Comput., vol. 54, no. 3, pp. 272–283, Mar. 2005.
[12] M. C.Wen, S. J. Wang, and Y. N. Lin, "Low-power parallel multiplier with column bypassing," Electron. Lett., vol. 41, no. 12, pp. 581–583, May 2005.
[13] J. S.Wang, C. N. Kuo, and T. H. Yang, "Low-power fixed-width array multipliers," in Proc. IEEE Symp. Low Power Electron. Des., Aug. 9–11, 2004, pp. 307–312.
[14] W. C. Yeh and C. W. Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, no. 7, pp. 692–701, Jul. 2000.