International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 1, Issue 4 (December 2014), PP.77-84

VLSI Design of High Performance Complex Multiplier

D.Padma Shri[1],D Subba Rao[2],G Vijay Goud[3]
[1, 2, 3] Assistant Professor,Ece, Department,Hitscoe

Abstract: Complex Multiplication Is Of Immense Importance In Digital Signal Processing (DSP) And Image Processing (IP). To Implement The Hardware Module Of Discrete Fourier Transformation (DFT), Discrete Cosine Transformation (DCT), Discrete Sine Transformation (DST) And Modem Broadband Communications; Large Numbers Of Complex Multipliers Are Required. Complex Number Multiplication Is Performed Using Four Real Number Multiplications And Two Additions/ Subtractions. In Real Number Processing, Carry Needs To Be Propagated From The Least Significant Bit (LSB) To The Most Significant Bit (MSB) When Binary Partial Products Are Added. Therefore, The Addition And Subtraction After Binary Multiplications Limit The Overall Speed. Vedic Mathematics Is The Ancient Methodology Of Indian Mathematics Which Has A Unique Technique Of Calculations Based On 16 Sutras (Formulae). A High Speed Complex Multiplier Design (ASIC) Using Vedic Mathematics Is Presented In This Paper. The Idea For Designing The Multiplier And Adder, Subtractor Unit Is Adopted From Ancient Indian Mathematics "Vedas". On Account Of Those Formulas, The Partial Products And Sums Are Generated In One Step Which Reduces The Carry Propagation From LSB To MSB..

Index-Terms: ASIC, VLSI, VHDL, FPGA. LSB, MSB

I. INTRODUCTION

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) and inner product are among some of the frequently used Computation- Intensive Arithmetic Functions(CIAF) currently implemented in many

Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications**.** This work presents different multiplier architectures. Multiplier based on Vedic Mathematics is one of the fast and low power multiplier**.**

Basics of Multiplier:

Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form a result (product). In elementary school, students learn to multiply by placing the multiplicand on top of the multiplier. The multiplicand is then multiplied by each digit of the multiplier beginning with the rightmost, Least Significant Digit (LSD). Intermediate results (partial products) are placed one atop the other, offset by one digit to align digits of the same weight. The final product is determined by summation of all the partial-products. Although most people think of multiplication only in base 10, this technique applies equally to any base, including binary. Figure-2.1 shows the data flow for the basic multiplication technique just described. Each black dot represents a single digit.

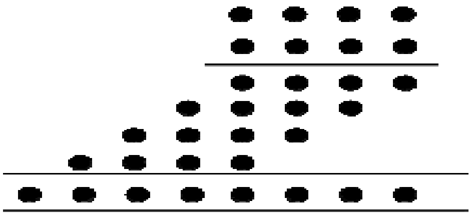


Figure-1: basic Multiplication

Here, we assume that MSB represent the sign of the digit. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers. Based upon the above procedure, we can deduce an algorithm for any kind of multiplication which is shown in figure 1.2. We can check at the initial stage also that whether the product will be positive or negative or after getting the whole result, MSB of the results tells the sign of the product.

Block Diagram

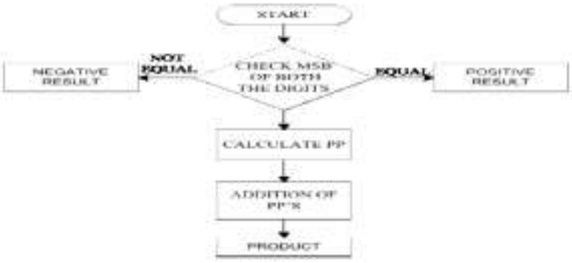
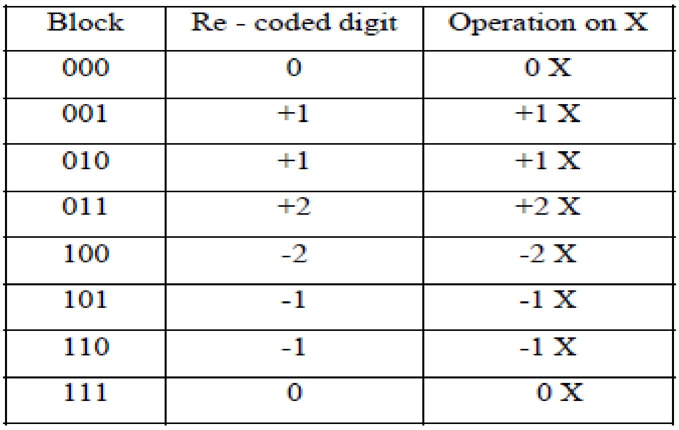


Figure-2 Signed Multiplication Algorithm

2.5 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.

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 radix-4 Booth recoding. 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 Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping.



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

Table1: Encoding Scheme

Vedic Multiplication Algorithms History Of Vedic Mathematics

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus. His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swahiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. Obviously these formulae are not to be found in present text of Atharva Veda because these formulae were constructed by Swamiji himself. Vedic mathematics is not only a mathematical wonder but also it is logical. That‟s why VM has such a degree of eminence which cannot be disapproved. Due these phenomenal characteristic, VM has already crossed the boundaries of India and has become a leading topic of research abroad. VM deals with several basic as well as complex mathematical operations. Especially, methods of basic arithmetic are extremely simple and powerful.

The word „Vedic‟ is derived from the word „veda‟ which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

Vedic Multiplication

The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. Vedic multiplication based on some algorithms, some are discussed below:

Urdhva Tiryakbhyam sutra

The multiplier is based on an algorithm Urdhva Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means “Vertically and crosswise”. It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Triyakbhyam explained in fig 2.1. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power

multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed.

Multiplication of two decimal numbers- 325*738

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Fig.2.2. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero. To make the methodology morclear, an alternate illustration is given with the help of line diagrams in figure where the dots represent bit „0‟ or „1‟.

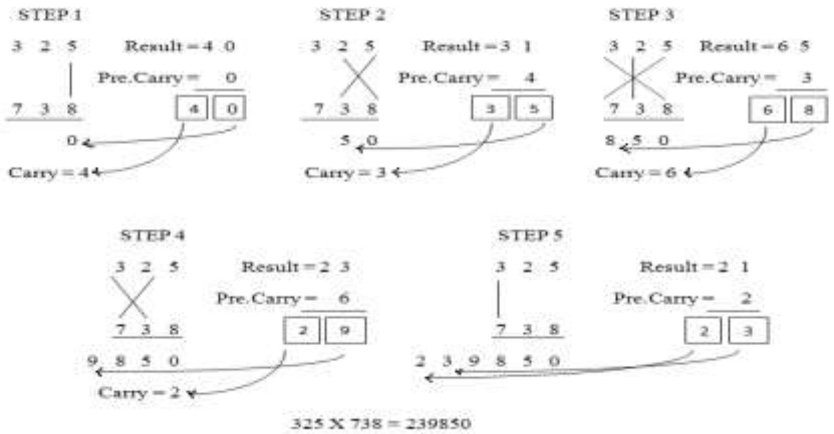


Figure-3: Multiplication of two decimal numbers by Urdhva Tiryakbhyam.

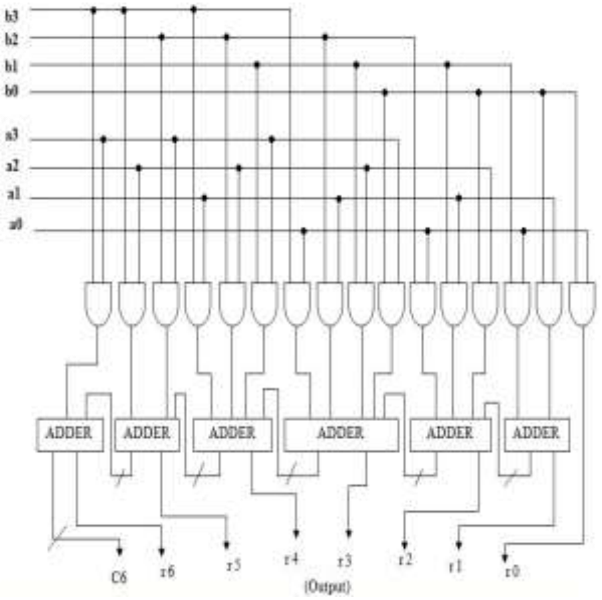


Figure 4: Hardware architecture of the Urdhva tiryakbhyam multiplier

Nikhilam Sutra

Nikhilam Sutra literally means "all from 9 and last from 10". Although it is applicable to all cases of multiplication, it is more efficient when the numbers involved are large. Since it finds out the compliment of the large number from its nearest base to perform the multiplication operation on it, larger is the original number, lesser the complexity of the multiplication. We first illustrate this Sutra by considering the multiplication of two decimal numbers (96 * 93) where the chosen base is 100 which is nearest to and greater than both these two numbers.

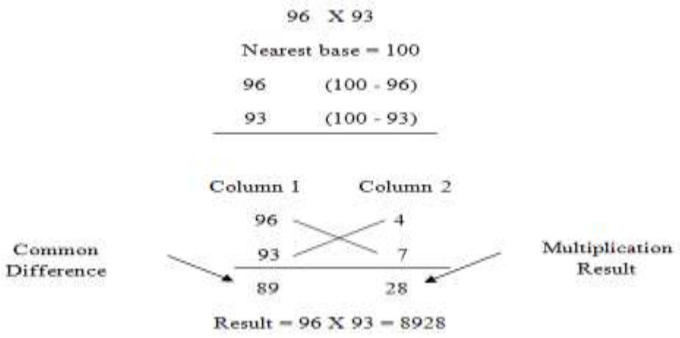


Figure-5: Multiplication Using Nikhilam Sutra

Compressor

A compressor adder is a logical circuit which is used to improve the computational speed of the addition of 4 or more bits at a time. Compressors can efficiently replace the combination of several half adders and full adders, thereby enabling high speed performance of the processor which incorporates the same. The compressor adder used in this paper is a 4:2 compressor adder. A lot of research in the past has been carried out on the same. This has been elaborated below. A comparison of the 4:2 compressor with an equivalent circuit, using full adders and half adders has also been given below.

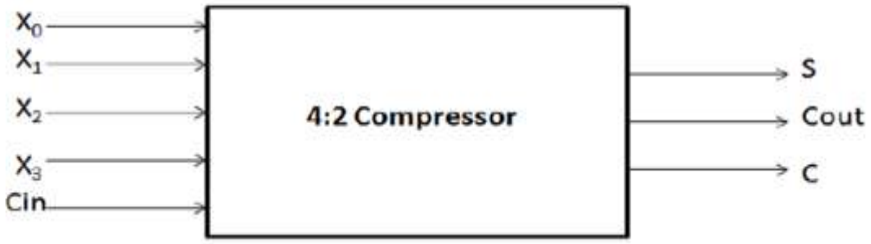


Figure-6: Block box of a 4:2 compressor adder

A 4:2 compressor as shown in fig.3. is capable of adding 4 bits and one carry, in turn producing a 3 bit output. The internal architecture of the same has been show in Figure

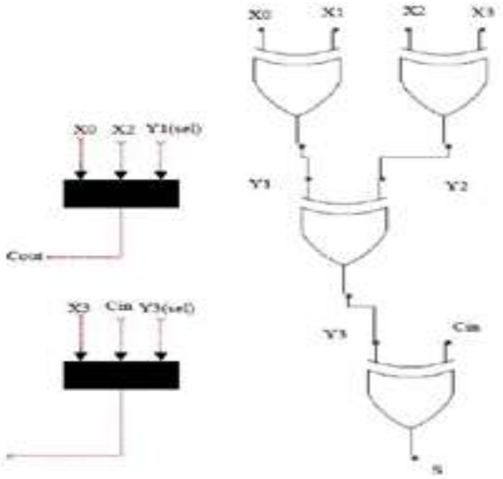


Figure-7: Gate level diagram of 4:2 Compressor

It can be clearly seen, the critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders. For the sake of comparison, the equivalent circuit to add 5 bits has also been shown in Figure

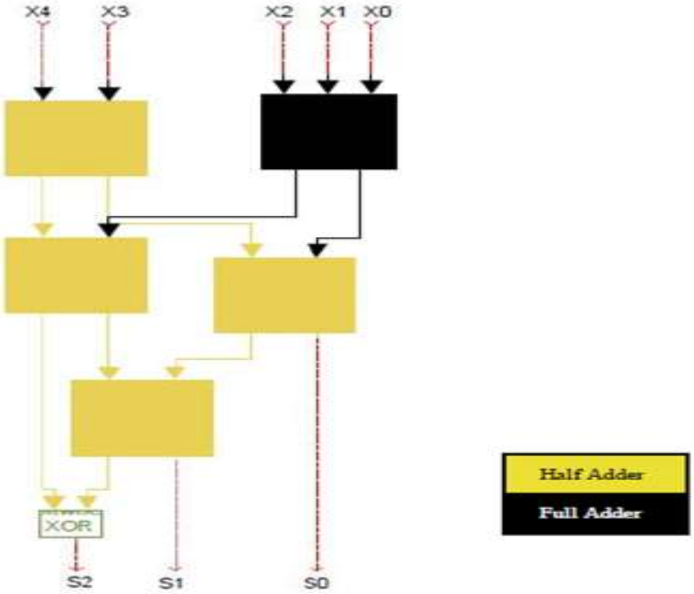
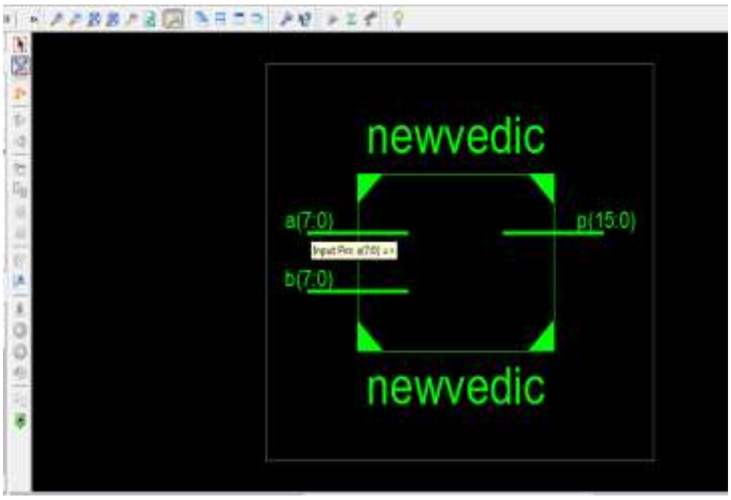
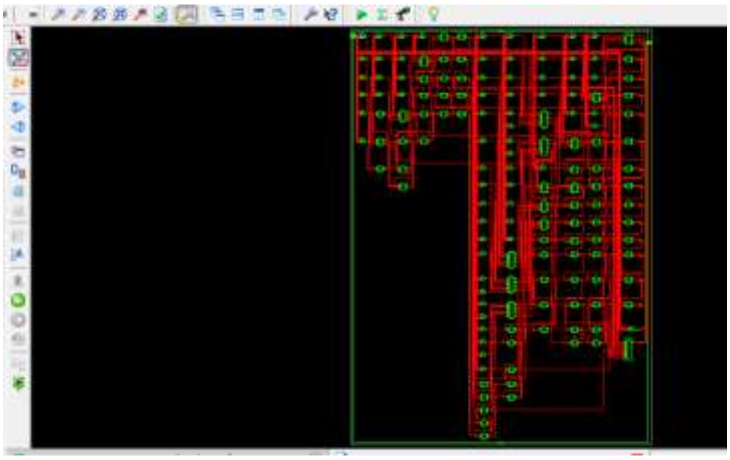


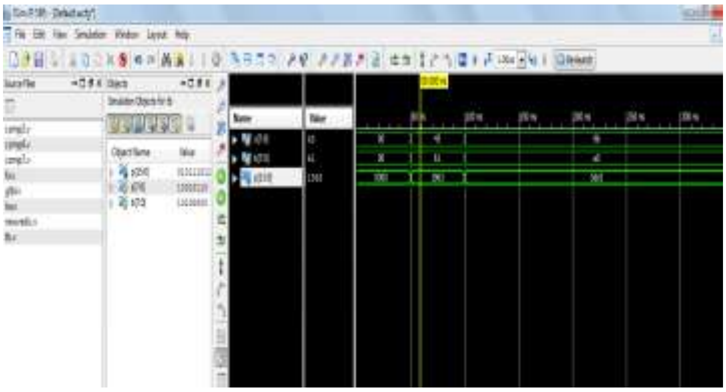
Figure 8: Compressor using full adders and half adders

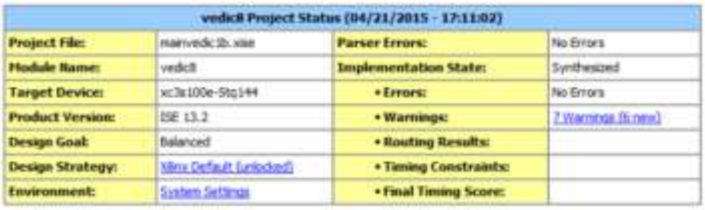
Let us consider the propagation delay of a gate to be tp. It is well known that a full adder has a total propagation delay of 2tp and a half adder has a propagation delay of tp. Considering this, the total propagation delay of a 4:2 adder using full adders and half adders can be calculated as 5tp and can be seen in Fig.5. On the other hand, it can be seen from Fig. 4.

that the propagationdelay of a 4:2 compressor remains only 3tp. Therefore, a 66.6% increase in speed can be recorded in comparison with an equivalent circuit made of full and half adders, proving to be a highly efficient architecture for addition. In order to add more than 5 bits at a time, yet another compressor architecture – the 7:2 compressor adder could be used and this is explained in detail below.

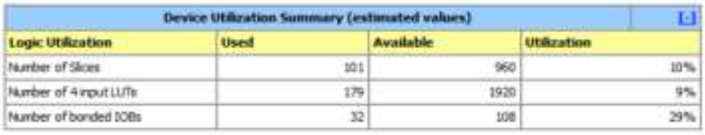








vedic8 Project Status (04/21/2015 - 17:11:02)			
Project File:	mainvedic1b.xise	**Parser Errors:**	No Errors
Module Name:	vedic8	**Implementation State:**	Synthesized
Target Device:	xc3s100e-5tq144	• **Errors:**	No Errors
Product Version:	ISE 13.2	• **Warnings:**	7 Warnings (6 new)
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	



Device Utilization Summary (estimated values) [-]

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	101	960	10%
Number of 4 input LUTs	179	1920	9%
Number of bonded IOBs	32	108	29%

Figure-9: Simulation Results

II. CONCLUSION

The behavioral simulation waveforms for the Vedic and Radix-8 multiplier are shown in figure 2 and figure 3. The proposed Radix-8 Multiplier may be used in DSP applications because it gives better performance in terms of power, delay and PDP. The proposed adder based multiplier can be used in high speed application because of its less power dissipation and delay. Also, this multiplier has the minimum number of nonzero partial products based on the CSD number property. The number of add/subtract operations is further reduced through the use of bypass techniques. Thus, the complexity of the hardware implementation is dramatically reduced as compared to conventional methods, including modified Booth recoding and competing CSD recoding techniques. This approach achieves an overall speedup as well as reduced power consumption which is particularly critical in mobile multimedia applications. We analyzed the architecture and compared it to the previous fast architecture extracted in terms of area, speed, and power consumption. We found that the new architecture has a better performance than the previous ones.

Application

1. Low Area
2. High speed multiplication
3. High Performance
4. High Efficiency

References

[1]. J. Choi, J. Jeon, and K. Choi, “Power minimization of function units by partially guarded computation,” in Proc. Int. Symp. Low Power Electron. Des., Jul. 2000, pp. 131–136.

[2]. Fayed A and M. A. Bayoumi, “A novel architecture for low-power design of parallel multipliers,” in Proc. IEEE Comput. Soc. Annu Workshop VLSI, Apr. 2001, pp. 149–154.

[3]. N. Honarmand and A. A. Kusha, “Low power minimization combinational multipliers using data-driven signal gating,” in Proc. IEEE Int. Conf. Asia-Pacific Circuits Syst., Dec. 2006, pp. 1430– 1433.

[4]. K.-H. Chen and Y.-S. Chu, “A spurious-power suppression technique for multimedia/DSP applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 1, pp. 132–143, Jan. 2009.

[5]. T. Yamanaka and V. G. Moshnyaga, “Reducing energy of digital multiplier by adjusting voltage supply to multiplicand variation,” in Proc. 46th IEEE Midwest Symp. Circuits Syst., Dec. 2003, pp. 1423–1426.

[6]. 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.

[7]. M. J. Schulte and E. E. Swartzlander Jr., “Truncated multiplication with correction constant,” in Proc. Workshop VLSI Signal Process., Oct.1993, pp. 388–396.

[8]. S. J. Jou, M. H. Tsai, and Y. L. Tsao, “Low-error reduced-width Booth multipliers for DSP applications,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 11, pp. 1470–1474, Nov. 2003.

[9]. K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, “Design of lowerrorfixed-width modified booth multiplier,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531, May 2004.

[10]. T.-B. Juang and S.-F. Hsiao, “Low-power carry-free fixed-widthmultipliers with low-cost compensation circuit,” IEEE Trans.Circuits Syst.II, Analog Digit. Signal Process. vol. 52, no. 6, pp.299–303, Jun. 2005.

[11]. Shiann-Rong Kuang and Jiun-Ping Wang “Design of powerefficient configurable booth multiplier” IEEE Trans. Circuits Syst. I Regular Papers vol. 57, no.3, pp. 568-580, March 2010.

[12]. T. Kitahara, F. Minami, T. Ueda, K. Usami, S. Nishio, M. Murakata,and T. Mitsuhashi, “A clock-gating method for lowpower.

[13]. LSI design,” in Proc. Int. Symp. Low Power Electron. Des., Feb. 1998, pp. 07–312.

[14]. X. Chang, M. Zhang, G. Zhang, Z. Zhang, and J. Wang, “Adaptive clock gating technique for low power IP core in SOC design,” in Proc. IEEE Int. Symp. Circuits Syst., May 2007, pp. 2120–2123.

[15]. L. Dadda, “Some schemes for parallel multipliers,” Alta Freq., vol. 34,pp. 349–356, May 1965.

[16]. C.Wallace, “A suggestion for a fast multiplier,” IEEE Trans. Electron.Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.