ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

DESIGN AND ANALYSIS OF AN EFFICIENT MULTIPLIER WITH IMAGE MULTIPLICATION

Saravanan.R[1], Sathiyapriya.J[2]
[1]PG Scholar, Department OF ECE, [2]Assitant Professor, Department of ECE
[1, 2] Sri Krishna College of Engineering and Technology, Kuniamuthur P.O., Coimbatore-641008, Tamil Nadu, India.
[1]saran14.rk@gmail.com,[2]sathiyapriya@skcet.ac.in

***Abstract*—Multipliers play an important role in today's digital signal processing and various other applications. In this paper a new multiplier design was implemented that incorporate with 4-2 compressors for utilization in a multiplier and these multiplier are analyzed for a Dadda multiplier. The multiplier design relies on the compressor and can meet with the respect to the circuit based design. In addition to that an application was implemented based on the multiplier design to the image processing for the image multiplication in order to calculate the peak signal to noise ratio. The extensive simulation result revealsthe efficient multiplier design and an application was implemented to the image processing based on the multiplier design**

Index terms*—Compressor, Dadda Multiplier,Carry Skip Adder (CSKA), Image Multiplication

1 .INTRODUCTION

MOST computer arithmetic applications are implemented using digital logic circuits, thus operating with a high degree of reliability and precision. However, many applications such as in multimedia and image processing can tolerate errors and imprecision in computation and still produce meaningful and useful results. Accurate and precise models and algorithms are not always suitable or efficient for use in these applications. The paradigm of inexact computation relies on relaxing fully precise and completely deterministic building modules when for example, designing energy-efficient systems. This allows imprecise computation to redirect the existing design process of digital circuits and systems by taking advantage of a decrease in complexity and cost with possibly a potential increase in performance and power efficiency.

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

Addition and multiplication are widely used operations in computer arithmetic; for addition full-adder cells have been extensively analyzed for approximate computing [2-4]. [1] has compared these adders and proposed several new metrics for evaluating approximate and probabilistic adders with respect to unified figures of merit for design assessment for inexact computing applications. For each input to a circuit, the error distance (ED) is defined as the arithmetic distance between an erroneous output and the correct one [1]. The tradeoff between precision and power has also been quantitatively evaluated in [1].

However, the design of approximate multipliers has received less attention. Multiplication can be thought as the repeated sum of partial products; however, the straightforward application of approximate adders when designing an approximate multiplier is not viable, because it would be very inefficient in terms of precision, hardware complexity and other performance metrics. Several approximate multipliers have been proposed in the literature [4] [5] [6] [7]. Most of these designs use a truncated multiplication method; they estimate the least significant columns of the partial products as a constant. In [4], an imprecise array multiplier is used for neural network applications by omitting some of the least significant bits in the partial products (and thus removing some adders in the array). A truncated multiplier with a correction constant is proposed in [5]. For an n×n multiplier, this design calculates the sum of the n+k most significant columns of the partial products and truncates the other n-k columns. The n+k bit result is then rounded to n bits. The reduction error (i.e. the error generated by truncating then-k least significant bits) and rounding error (i.e. the error generated by rounding the result to n bits) are found in the next step. The correction constant (n+k bits) is selected to be as close as possible to the estimated value of the sum of these errors to reduce the error distance.

A truncated multiplier with constant correction has the maximum error if the partial products in the n-k least significant columns are all ones or all zeros. A variable correction truncated multiplier has been proposed in [6].This method changes the correction term based on column n-k-1. If all partial products in columnn-k-1 are one, then the correction term is increased. Similarly, if all partial products in this column are zero, the correction term is decreased.

In [7], a simplified (and thus inaccurate) 2x2 multiplier block is proposed for building larger multiplier arrays. In the design of a fast multiplier, compressors have been widely used [8] to speed up the partial product reduction tree and decrease power dissipation. Optimized designs of 4-2 exact compressors have been proposed in [8, 11 - 16]. [17] [18] have also considered compression for approximate multiplication. In [17], an approximate signed multiplier has been proposed for use in arithmetic data value speculation (AVDS); multiplication is performed using the Baugh-Wooley algorithm. However, no new design is proposed for the compressors for the inexact computation. Designs of approximate compressors have been proposed in [18]; however, these designs do not target multiplication. It should be noted that the approach of [7] improves over [17] [18] by utilizing a simplified multiplier block that is amenable to approximate multiplication.

2. COMPRESSOR DESIGN

The main goal of either multi-operand carry-save addition or parallel multiplication is to reduce *n* numbers to two numbers; therefore, *n*-2 compressors (or *n*-2 counters) have been widely used in computer arithmetic. A*n*-2 compressor (Figure 1) is usually a slice of a circuit that reduces *n* numbers to two numbers when properly replicated. In slice *i* of the circuit, the *n*-2 compressor receives *n* bits in position *i* and one or more carry bits from the positions to the right, such as *i* – 1 or *i* – 2. It produces two output bits in positions *i* and *i* + 1 and one or more carry bits into the higher positions, such as i + 1 or i+2.

$$n + \psi_1 + \psi_2 + \psi_3 + \dots \leq 3 + 2\psi_1 + 4\psi_2 + 8\psi_3 + \dots \qquad (1)$$

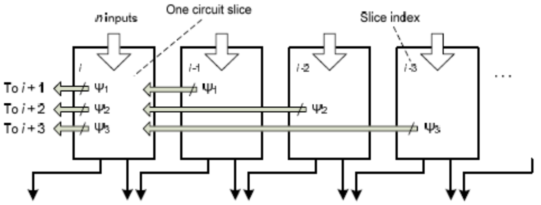


Figure 1.Schematic diagram of *n*-2 compressors in a multi operand addition circuit [13]

Where ¥ denotes the number of carry bits from slice ito slice i+ j.

A widely used structure for compression is the 4-2 compressor; a 4-2 compressor (Figure 2) can be implemented with a carry bit between adjacent slices. The carry bit from the position to the right is denoted as cin while the carry bit into the higher position is denoted as cout. The two output bits in positions i andi + 1are also referred to as the sum and carry respectively.

The common implementation of a 4-2 compressor is accomplished by utilizing two full-adder (FA) cells (Figure 3) [8]. Different designs have been proposed in the literature for 4-2 compressor [8, 11-16]. Figure 4 shows the optimized design of an exact4-2 compressor based on the so-called XOR-XNOR gates [8]; a XOR-XNOR gate simultaneously generates the XOR and XNOR output signals. The design of [8] consists of three XOR-XNOR (denoted by XOR*) gates, one XOR and two 2-1 MUXes. The critical path of this design has a delay of 3Δ, where Δ is the unitary delay through any gate in the design.

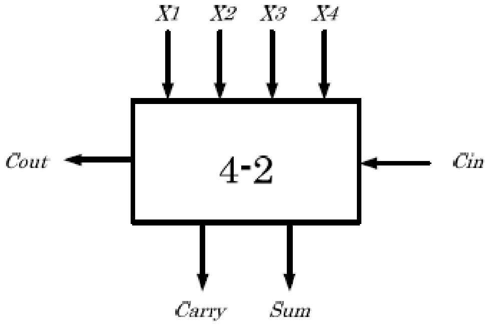


Figure2.4-2 compressor

$$Sum = x1 \oplus x2 \oplus x3 \oplus x4 \oplus Cin \qquad (2)$$

$$Cout = (x1 \oplus x2)x3 + \overline{(x1 \oplus x2)}x1 \qquad (3)$$

$$Carry = (x1 \oplus x2 \oplus x3 \oplus x4)Cin + \overline{(x1 \oplus x2 \oplus x3 \oplus x4)}x4 \qquad (4)$$

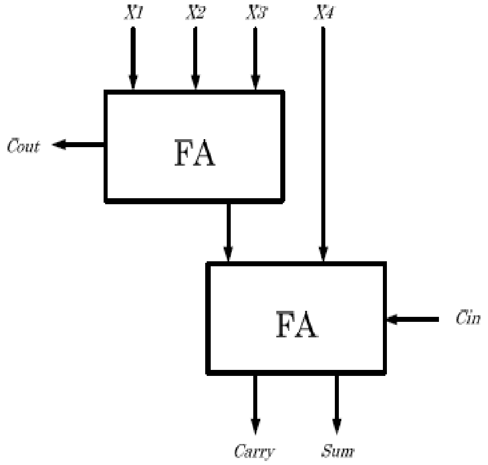


Figure 3. Implementation of 4-2 Compressor

TABLE I
TRUTH TABLE OF 4-2 COMPRESSOR

c_{in}	X_4	X_3	X_2	X_1	c_{out}	*carry*	*sum*
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0
0	0	1	0	0	0	0	1
0	0	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

Intuitively to design an approximate 4-2 compressor, it is possible to substitute the exact full-adder cells in Figure3 by an approximate full-adder cell (such as the first design proposed in [2]). However, this is not very efficient, because it produces at least 17 incorrect results out of 32 possible outputs, i.e. the error rate of this inexact compressor is more than 53% (where the error rate is given by the ratio of the number of erroneous outputs over the total number of outputs). Two different designs are proposed next to reduce the error rate; these designs offer significant performance improvement compared to an exact compressor with respect to delay, number of transistors and power consumption.

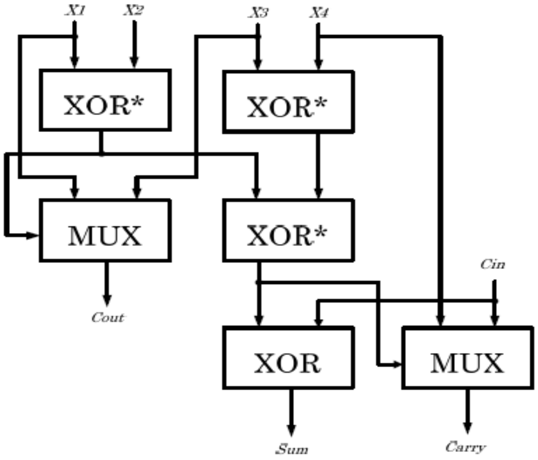


Figure4. Optimized 4-2 compressor of [8]

As shown in Table I, the *carry* output in an exact compressor has the same value of the input *cin* in 24 out of 32 states. Therefore, an approximate design must consider this feature. In Design 1, the *carry* is simplified to *cin* by changing the value of other 8 outputs.

$$Carry' = Cin \qquad (5)$$

Since the Carry output has the higher weight of a binary bit, an erroneous value of this signal will produce a difference value of two in the output. For example, if the input pattern is "01001" (row 10 of Table II), the correct output is "010" that is equal to 2. By simplifying the carry output to cin, the approximate compressor will generate the "000" pattern at the output (i.e. a value of 0). This substantial difference may not be acceptable; however, it can be compensated or reduced by simplifying the cout and sum signals. In particular, the simplification of sum to a value of 0 (second half of Table II) reduces the difference between the approximate and the exact outputs as well as the complexity of its design. Also, the presence of some errors in the sum signal will results in a reductions of the delay of producing the approximate sum and the overall delay of the design (because it is on the critical path).

$$Sum' = \overline{Cin}(\overline{x1 \oplus x2} + \overline{x3 \oplus x4}) \qquad (6)$$

In the last step, the change of the value of *cout* in some states, may reduce the error distance provided by approximate *carry* and *sum* and also more simplification in the proposed design.

$$Cout' = \overline{(\overline{x1x2} + \overline{x3x4})} \qquad (7)$$

Although the above mentioned simplifications of carry and sum increase the error rate in the proposed approximate compressor, its design complexity and therefore the power consumption are considerably decreased. This can be realized by comparing (2)-(4) and (5)-(7).Table II shows the truth table of the first proposed approximate compressor. It also shows the difference between the inexact output of the proposed approximate compressor and the output of the exact compressor. As shown in Table II, the proposed design has 12 incorrect outputs out of 32 outputs (thus yielding an error rate of 37.5%). This is less than the error rate using the best approximate full-adder cell of [2].

TABLE II
TRUTH TABLE OF THE FIRSTAPPROXIMATE 4-2 COMPRESSOR

c_{in}	X_4	X_3	X_2	X_1	c_{out}'	*carry'*	*sum'*	*Difference*
0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	-1
0	0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	0	0	0
0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	0	0	0
0	1	0	1	1	1	0	1	0
0	1	1	0	0	0	0	1	-1
0	1	1	0	1	1	0	1	0
0	1	1	1	0	1	0	1	0
0	1	1	1	1	1	0	1	-1
1	0	0	0	0	0	1	0	1
1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	0	1	1	0	1	0	-1
1	0	1	0	0	0	1	0	0
1	0	1	0	1	1	1	0	1
1	0	1	1	0	1	1	0	1
1	0	1	1	1	1	1	0	0
1	1	0	0	0	0	1	0	0
1	1	0	0	1	1	1	0	1
1	1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	0	0
1	1	1	0	0	0	1	0	-1
1	1	1	0	1	1	1	0	0
1	1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	0	-1

The gate level structure of the first proposed design (Figure 6) shows that the critical path of this compressor has still a delay of 3Δ, so it is the same as for the exact compressor of Figure 5. However, the propagation delay through the gates of this design is lower than the one for the exact compressor. For example, the propagation delay in the *XOR** gate that generates both the *XOR* and *XNOR* signals in [8], is higher than the delay through a *XNOR* gate of the proposed design. Therefore, the critical path delay in the proposed design is lower than in the exact design and moreover, the total number of gates in the proposed design is significantly less than that in the optimized exact compressor of [8].

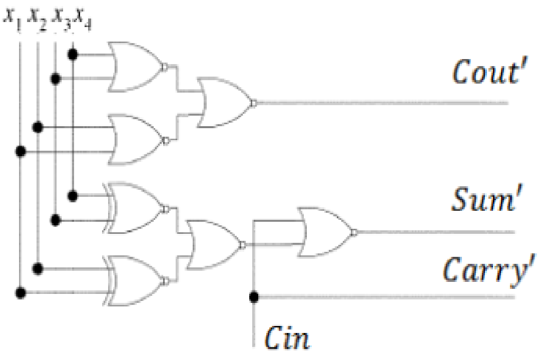


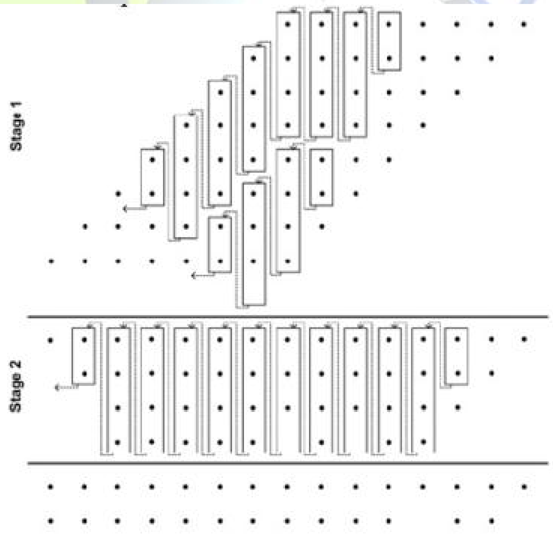
Figure 6. Gate level implementation of Design 1

3. MULTIPLICATION

In this section, the impact of using the proposed compressors for multiplication is investigated. A fast (exact) multiplier is usually composed of three parts (or *modules*) [8].

- Partial product generation.
- A Carry Save Adder (CSA) tree to reduce the partial products' matrix to an addition of only two operands
- A Carry Skip Adder (CSKA) for the final computation of the binary result. Christo Ananth et al. [10] proposed a system, Low Voltage Differential Signaling (LVDS) is a way to communicate data using a very low voltage swing (about 350mV) differentially over two PCB traces. It deals about the analysis and design of a low power, low noise and high speed comparator for a high performance Low Voltage Differential Signaling (LVDS) Receiver. The circuit of a Conventional Double Tail Latch Type Comparator is modified for the purpose of low-power and low noise operation even in small supply voltages. The circuit is simulated with 2V DC supply voltage, 350mV 500MHz sinusoidal input and 1GHz clock frequency. LVDS Receiver using comparator as its second stage is designed and simulated in Cadence Virtuoso Analog Design Environment using GPDK 180nm .By this design, the power dissipation, delay and noise can be reduced. In the first stage, 2 half-adders, 2 full-adders and 8 compressors are utilized to reduce

Figure 7. Reduction circuitry of an 8×8Dadda multiplier



4. RESULTS AND DISCUSSION

The simulation of the Multiplier design was done by using Modelsim SE 10.1c. The synthesis was implemented with XILINX ISE 14.5 design suite, here all the three stages of the multiplier design

Table III
Tabulation for Multiplier design

Parameters	Existing[8]	Proposed
Area Utilization (%)	16	7
Power (mW)	54	45
Delay (mS)	47.53	27.43
Transitor Count	2285	1155
Junction Temperture (C)	28	26

The delay of the reduction circuitry (second module) of a Dadda multiplier is dependent on the number of reduction stages and the delay of each stage. The power consumption of each multiplier is determined by the number and type of compressors used.

5. APPLICATION: IMAGE MULTIPLICATION

The simulation of the image multiplication was done by using MATLAB R2013a. In this section the application of the proposed approximate multipliers to image processing is illustrated. A multiplier is used to multiply two images on a pixel by pixel basis, thus blending the two images into a single output image.

The average NED and the Peak Signal-to-Noise Ratio (PSNR) that is based on the Mean Squared Error (MSE) are computed to assess the quality of the output image and compare it with the output image generated by an exact multiplier. The equations for the MSE and PSNR are given, *m* and *p* are the image dimensions and *I(i,j)* and *K(i,j)* are the exact and obtained values of each pixel respectively. In MAXIrepresents the maximum value of each pixel.

$$\text{MSE} = \frac{1}{mp}\sum_{i=0}^{m-1}\sum_{j=0}^{p-1}[I(i,j) - K(i,j)]^2$$

$$\text{PSNR} = 10log_{10}(\frac{MAX_I^2}{\text{MSE}})$$

Table IV
Tabulation for Multiplier design

Parameter	PSNR (dB)	Average NED ($x10^{-2}$)
Multiplier	49.87	0.28

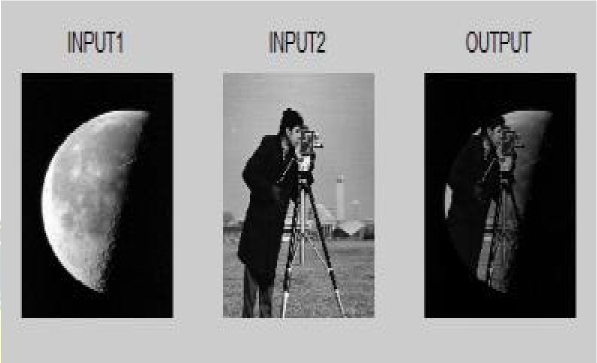


Figure 8. Image Multiplication example

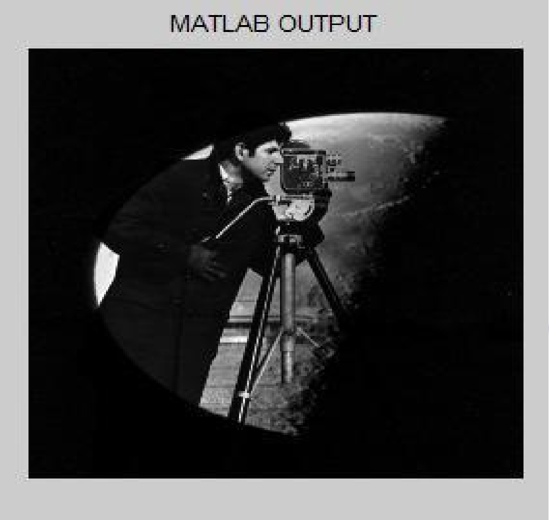


Figure 9. Image Multiplication MATLAB output

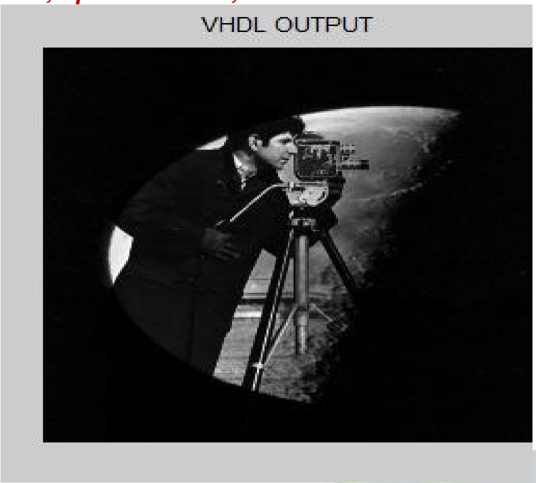


Figure 10. Image Multiplication VHDL output

6. CONCLUSION

Inexact computing is an emerging paradigm for computation at nanoscale. Computer arithmetic offers significant operational advantages for inexact computing; an extensive literature exists on approximate adders. However, this paper has initially focused on compression as used in a multiplier; to the best knowledge of the authors, no work has been reported on this topic.

These approximate compressors are utilized in the reduction module of four approximate multipliers. The approximate compressors show a significant reduction in transistor count, power consumption and delay compared with an exact design.

Moreover, the application of these approximate multipliers to image processing has confirmed that two of the proposed designs achieve a PSNR of nearly 50dB in the output generated by multiplying two input images, thus viable for most applications.

In conclusion, this paper has shown that by an appropriate design of an approximate compressor, multipliers can be designed for inexact computing; these multipliers offer significant advantages in terms of both circuit-level and error figure of merit/

REFERENCES

[1] J. Liang, J. Han, F. Lombardi, “New Metrics for the Reliability of Approximate and Probabilistic Adders,” IEEE Transactions on Computers,vol. 63, no. 9, pp. 1760 - 1771, 2013.

[2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, K. Roy, “IMPACT: IMPrecise adders for low-power approximate computing,” Low Power Electronics and Design (ISLPED) 2011 International Symposium on. 1-3 Aug. 2011.

[3] S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, “A probabilistic CMOS switch and its realization by exploiting noise,” in Proc. IFIP-VLSI SoC, Perth, Western Australia, Oct. 2005.

[4] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, “Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850-862, April 2010.

[5] M. J. Schulte and E. E. Swartzlander, Jr., “Truncated multiplication with correction constant,” VLSI Signal Processing VI, pp. 388–396, 1993.

[6] E. J. King and E. E. Swartzlander, Jr., “Data dependent truncated scheme for parallel multiplication,” in Proceedings of the Thirty First Asilomar Conference on Signals, Circuits and Systems, pp. 1178–1182, 1998.

[7] P. Kulkarni, P. Gupta, and MD Ercegovac, “Trading accuracy for power in a multiplier architecture”, Journal of Low Power Electronics, vol. 7, no. 4, pp. 490--501, 2011.

[8] C. Chang, J. Gu, M. Zhang, “Ultra Low-Voltage Low- Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits,” IEEE Transactions on Circuits & Systems, Vol. 51, No. 10, pp. 1985-1997, Oct. 2004.

[9] D. Radhakrishnan and A. P. Preethy, “Low-power CMOS pass logic 4-2 compressor for high-speed multiplication,” in Proc. 43rd IEEE Midwest Symp. Circuits Syst., vol. 3, 2000, pp. 1296–1298.

[10] Christo Ananth, Bincy P Chacko, “Analysis and Design of Low Voltage Low Noise LVDS Receiver”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 2, Ver. V (Mar - Apr. 2014), PP 10-18

[11] J. Gu, C. H. Chang, “Ultra Low-voltage, low-power 4-2 compressor for high speed multiplications,” in Proc. 36th IEEE Int.

Symp. Circuits Systems, Bangkok, Thailand, May 2003.

[12] M. Margala and N. G. Durdle, “Low-power low-voltage 4-2 compressors for VLSI applications,” in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 84–90.

[13] B. Parhami, “Computer Arithmetic: Algorithms and Hardware Designs,” 2nd edition, Oxford University Press, New York, 2010.

[14] K. Prasad and K. K. Parhi, “Low-power 4-2 and 5-2 compressors,” in Proc. of the 35th Asilomar Conf. on Signals, Systems and Computers, vol. 1, 2001, pp. 129–133.

[15] Ercegovac, Miloš D., and Tomas Lang. Digital arithmetic. Elsevier, 2003.

[16] Baran, Dursun, Mustafa Aktan, and Vojin G. Oklobdzija. "Energy efficient implementation of parallel CMOS multipliers with improved compressors."Proc. of the 16th ACM/IEEE international symposium on Low power electronics and design. ACM, 2010.

[17] D. Kelly, B. Phillips, S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation", in Proc. of the conference on design and architectures for signal and image processing, 2009

[18] D. Kelly, B. Phillips, S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation", in Proc. of the conference on design and architectures for signal and image processing,