IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified ⪤ Impact Factor 7.047 ⪤ Vol. 10, Issue 5, May 2022

DOI: 10.17148/IJIREEICE.2022.10562

Design of Power and Area Efficient Approximate Multipliers For Edge Detection Algorithm for Image Processing Application

Eindhumathy J[1], Pooja B[2], Rubiga M[3], Sofana M[4]

Assistant Professor - Department of Electronics and Communication Engineering, Saranathan college of Engineering, Trichy, TamilNadu [1]

UG - Department of Electronics and Communication Engineering, Saranathan college of Engineering, Trichy, TamilNadu [2,3,4]

Abstract : Approximate computing is tentatively applied in some Digital Signal Processing Applications which have an inherent tolerance for Errorneous Result. The Approximate Arithmetic blocks are utilized in them to improve Electrical Performance of these Circuits..The Result Shows that the Proposed Approximate multiplier design accomplish significant reductions in Power dissipation, Delay and Transistor count compared to an Exact accurate Multiplier design with a small loss in accuracy with a HighSpeed Output.This Design is Implemented by Verilog HDL and simulated by Modelsim 6.4c.The Performance is Measured by Xilinix tool Synthesis Process.The Proposed Sobel Edge detection algorithm uses approximation methods to replace the complex operations;This design is done by Matlab and Modelsim.This Proposed Multipliers are Replaced in the Sobel Operator Based Image Edge Detection. The Proposed Compressors Achieve Reduction in delay,power and area Respectively.

Keywords Approximate Compressors , Image Processing , Sobel Edge Detection , VerilogHDL , Dadda Multiplier.

I.INTRODUCTION

The tradeoff between reduction in accuracy and area, delay, power dissipation, which does not affect the typical operation for machine learning and multimedia applications. the inability of human eye to detect difference in finer details within images and videos, effectively take as advantage to implement machine learning and multimedia applications. This level of error tolerance is used to implement approximate computing circuits for Artificial intelligence (AI) and Digital Signal Processing (DSP) applications. Edge detection techniques have been successfully used for different applications. In edge detection, the abrupt changes in the pixel intensity are determined. These change in pixel intensities are determined by different techniques, in which different parameters are tuned to refine the edges of salient objects while suppressing the redundant objects from image. The edges obtained Since multiplication is a fundamental operation in many digital systems. While for multipliers of smaller bit widths, as their partial products are usually generated by simple AND gates, the approximation is applied in compression trees, which is arranged to accumulate all the generated partial products. 4-2 compressor is the core of such compression trees, which balances the compression efficiency and hardware cost [4]. It gets widely used in fast parallel multipliers to accelerate the compression process. Accordingly, many approximate designs are introduced in 4-2 compressors [5]-[12]. In [5], two novel approximate 4-2 compressors are proposed and embedded into the multipliers. The second design of [5] illustrates the rationality for canceling cin and cout of 4-2 compressors which is also adopted in the subsequent designs. Four designs of dual-quality 4-2 compressors are delivered in [6]. A high accuracy 4-2 compressor is presented in [7] through the modifying and simplification of the Karnaugh map. Further optimization is made in [8] with an error recovery module to enhance the error performance of the multiplier. A design of fast multiplier using approximate 4-2 compressors and modified Booth encoding is presented in [9]. Another derivation method on approximate 4-2 compressor is proposed and the transistor-level circuit optimization is presented for low supply voltage applications in [10]. Similar logical expression combined with a partial product altering method is proposed in [11] to achieve a better balance between electrical and error performances. A series of approximate compressors based on AND/OR logics is proposed in [12]. Besides, a new algorithm is also presented to embed them into efficient approximate multipliers.

II.EXACT 4-2 COMPRESSORS

In high-speed parallel multipliers, 4-2 compressors are employed to speed up the compression process of the partial products. The conventional way to implement a 4-2 compressor is cascading two full adders, which is shown in Fig.1

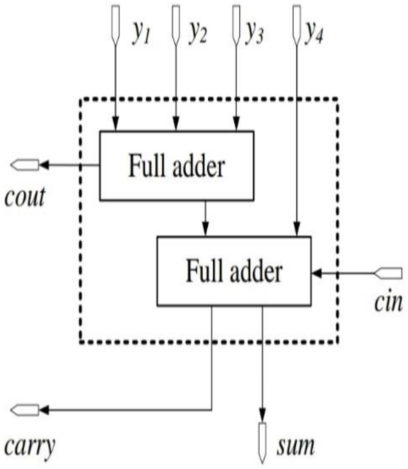


FIG.1. **Implementation of a conventional 4-2 compressor**

For each single 4-2 compressor, it employs 5 inputs (y1, y2, y3, y4, and cin) and 3 outputs (sum, cout, and carry). The outputs count the number of logic '1' in the five inputs. And the logical equations of its three output signals can be derived as below:

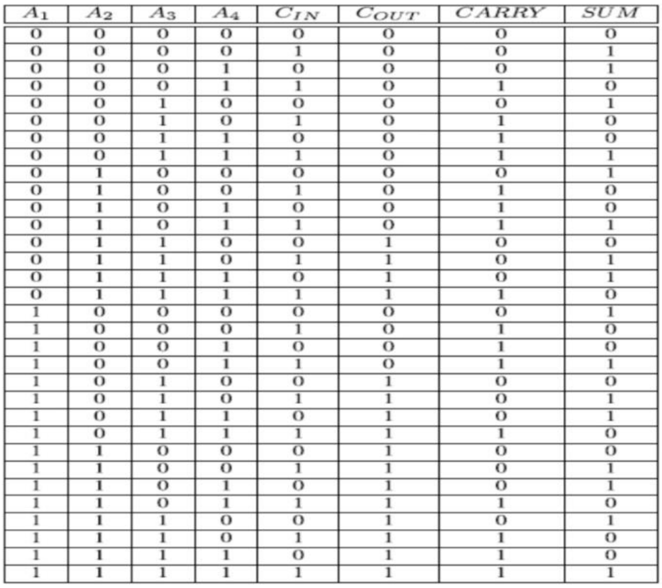
$$sum = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus cin \quad (1)$$

$$cout = (y_1 \oplus y_2)y_3 + \overline{(y_1 \oplus y_2)}y_1 \quad (2)$$

$$carry = (y_1 \oplus y_2 \oplus y_3 \oplus y_4)cin + \overline{(y_1 \oplus y_2 \oplus y_3 \oplus y_4)}y_4 \quad (3)$$

Among these three outputs, the cout and carry are propagated to the next bit. Therefore, these two signals keep the same weight and are more important than the sum.

The truthtable for the exact compressor is given below:

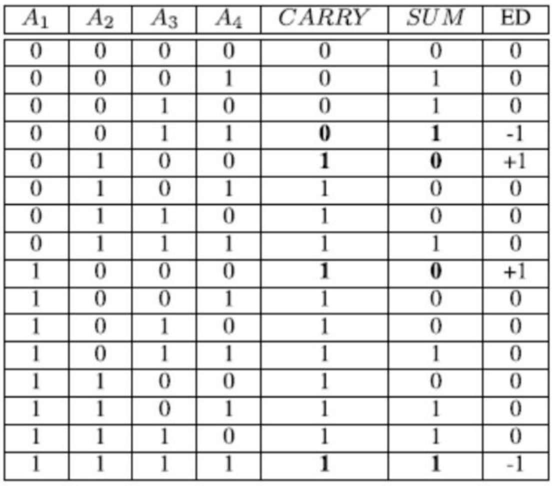


A_1	A_2	A_3	A_4	C_{IN}	C_{OUT}	$CARRY$	SUM
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	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	0	1	1
1	0	1	0	0	1	0	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	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	1	0	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

III.PROPOSED APPROXIMATE COMPRESSORS

The proposed high speed area-efficient 4-2 approximate compressor is proposed in this section. The compressor inputs are A1, A2, A3 and A4, outputs are CARRY' and SUM'. The input Cin and output Cout in the exact 4–2 compressor are completely ignored in the design of approximate 4–2 compressor.SUM can be generated using a multiplexer (MUX) based design approach.

Output of XOR gate(A1 XOR A2) acts as the select line for the MUX. When select line goes high, (A3 AND A4) is selected and when it goes low, (A3 OR A4) is selected. By introducing an error with error distance 1 in the truth table of the exact compressor, the proposed 4 -2 compressor , carry logic can be implemented with an OR gate. The logical expressions for realization of SUM' and CARRY' are given below.



A_1	A_2	A_3	A_4	$CARRY$	SUM	ED
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	**0**	**1**	-1
0	1	0	0	**1**	**0**	+1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	**1**	**0**	+1
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	**1**	**1**	-1

FIG 2:Truthtable of proposed 4-2 compressors

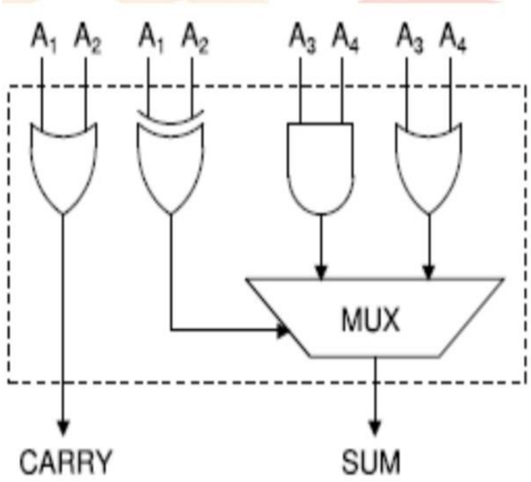


FIG 3:Proposed 4-2 compressors

IV.PROPOSED MULTIPLIER

In this section, the proposed design of an 8 × 8 approximate multiplier is presented. The multiplication operation can be divided into 3 parts as follows.

- Generation of partial products
- Arrange of partial products into two rows
- The computation of final result generally using adders

DOI: 10.17148/IJIREEICE.2022.10562

The multiplier overall performance is mainly depends on the optimization of the second module. The 8 ×8 unsigned Dadda multiplier is designed using the proposed approximate compressors. The architecture of the Dadda multiplier designed using conventional 4–2 compressors is presented in [14].Each dot in the figure indicate a partial product obtained from AND gates. The reduction module contain half-adders, full adders and approximate proposed 4–2 compressors.

The approximate compressors are indicated with rectangles. The main objective of replacing all the conventional compressors with approximate compressors is to minimize the delay, power consumption and area significantly. If the approximate compressors are used at the least significant columns (rightmost columns in Fig.2.) then the performance of approximate multipliers increase in terms of accuracy. The overall error at the output of the approximate multiplier is minimized by rearranging the order of input bits to the approximate compressor.

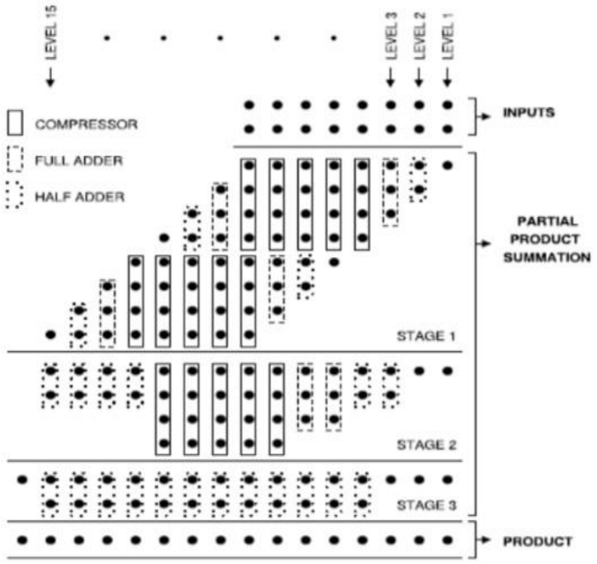
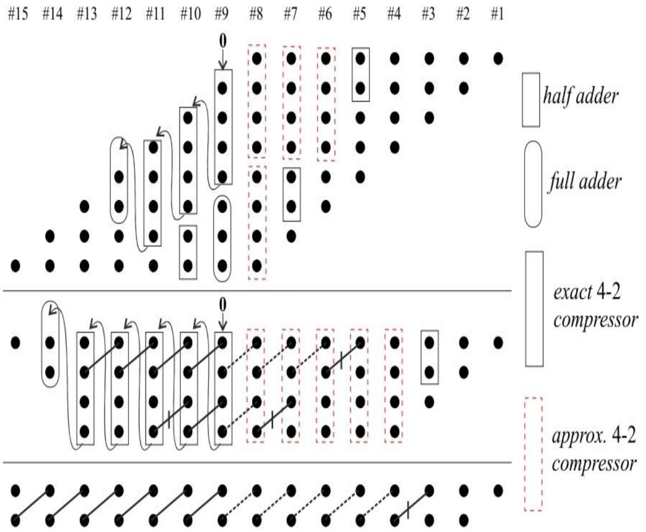


FIG 4: 8x8 Approximate Multiplier

Reduction scheme for an 8 × 8 unsigned multiplier with C – N configuration (approximate 4-2 compressors are used only in the 8 less significant columns of the partial-product matrix). Carry and Sum outputs of approximate 4-2 compressors are shown linked by dotted lines, while solid lines are used for full adders and exact 4-2 compressors .

DOI: 10.17148/IJIREEICE.2022.10562



V. MODULE & EXPLANATIONS FOR SOBEL PROCESS

MODULES NAME:
- Reading Image
- Compare the gradient magnitude with threshold value and find true edges
- Applying the convolution mask i and j on the input image
- Determine the gradient magnitude by computing
- Compare the gradient magnitude with threshold value and find true edge.

MODULE EXPLANATION:
READING IMAGE:

Sobel operator is used to detect edges of the test images used. This procedure is applied on more than one test image as shown in Fig.5 . Firstly the image data is read as an array with the dimension of image size. The number of elements of this array is calculated in order to resize the array of image to another array. The resizing that be used is (256×256) in MATLAB program.

FIG 5:Applying the Convolution Mask i and j on the Input Image

The horizontal template and vertical template shown in Fig. 5 are used to get convolution with input image by using equation (1) and (2). The result matrix after this operation is got the same size of two gradients Matrix Gx and Gy as the original image as shown in Fig 6.

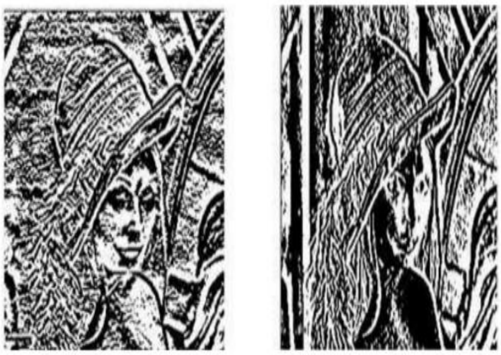


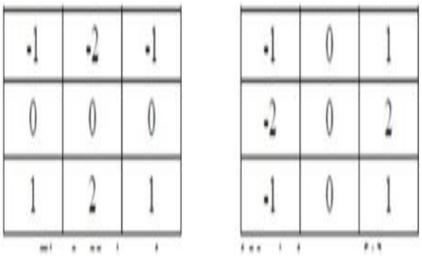
FIG 6:MATRIX IMAGE

DETERMINE THE GRADIENT MAGNITUDE BY COMPUTING

The gradient magnitude is determent by squaring the pixels values of each filtered image, Then Adding of the two results and computing their root to get the total gradient value (Gr) are done.

SOBEL EDGE DETETOR OPERATOR:

Sobel Edge Detection, there are two masks, one mask identifies the horizontal edges and the other mask identifies the vertical edges. Each of the masks has the effect of calculating the gradient in both vertical and horizontal direction. These Sobel masks are convolved with smoothed image and giving gradients in I and j
directions is given by sobel masks are showing in Fig 7.



-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

FIG 7:SOBEL MASKS

COMPARE THE GRADIENT MAGNITUDE WITH THRESHOLD VALUE AND FIND TRUE EDGES

Finally, the edges can be detected by applying the threshold by using equation (5) to the total gradient (Gr). If (Gr) is greater than the threshold, then pixel should be identified an edge as shown in Figure8 Else it's not identified as an edge. This Edge Detection logic is made by Schostic Logic Circuit.

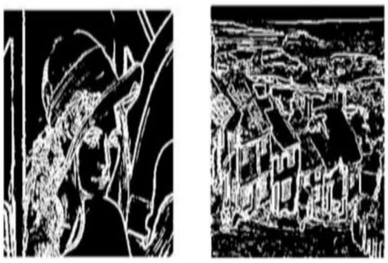
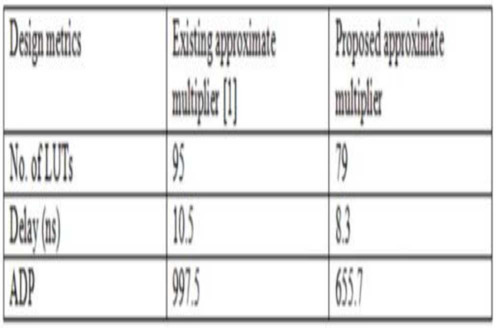


FIG 8 : EDGE IDENTIFIED IMAGE

This work is licensed under a Creative Commons Attribution 4.0 International License

VI.SIMULATION RESULTS



Design metrics	Existing approximate multiplier [1]	Proposed approximate multiplier
No. of LUTs	95	79
Delay (ns)	10.5	8.3
ADP	997.5	655.7

From the table it is observed that proposed multiplier will take 79 LUTs for designing but existing multiplier take 95 LUTs, from that proposed multiplier will take 16 LUTs less than existing multiplier i.e. 21% area efficient. proposed multiplier will take 8.3ns delay but existing multiplier take 10.3ns delay, from that proposed multiplier is faster as compared to existing approximate multiplier i.e. 21% speed improved as compared to existing. proposed multiplier will take 655.7 but existing multiplier take 997.5 , from that proposed multiplier ADP is 341.8 less than existing multiplier i.e. 35% ADP is efficient as compared to existing approximate multiplier.

VII . COMPARISON RESULTS

S no	**METHOD NAME**	**AREA**			**DELAY** (in ns)		
	MAC Based	**LUT**	**Slices**	**Gates**	**Max Delay**	**Gate Delay**	**Path Delay**
1	**Exisiting Multiplier**	**176**	**97**	**1056**	**45.84**	**19.13**	**26.70**
2	**Proposed Compressor**	**137**	**77**	**822**	**30.52**	**14.17**	**16.34**

RESULT 1

S no	**METHOD NAME**	**AREA**			**DELAY (in ns)**		
	Type	**LUT**	**Slices**	**Gates**	**Max Delay**	**Gate Delay**	**Path Delay**
1	**Exisiting Multiplier based MAC**	**246**	**134**	**1616**	**7.430**	**6364**	**1066**
2	**Proposed Compressor based MAC**	**206**	**109**	**1376**	**7.430**	**6364**	**1066**

RESULT 2

VIII . CONCLUSION

All approximate multipliers are designed for n = 8. The multipliers are implemented in Verilog and synthesized using This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. we will design a Efficient Array Multiplier using our proposed Multiplier. The proposed approximate compressors are proposed and analyzed for a Array multiplier. This Proposed Multiplier is used in sobel operator design. Sobel operator executed in matlab and modals software. The Proposed method is implemented using Verilog HDL and Simulated and Synthesised by modelsim and Xilinx tools. The Design was analysed by Xilinx Tool.A novel approximate 4 -2 compressor designs are presents in this paper. Firstly, a high speed area efficient compressor design is proposed, which attained a considerable reduction in area, delay and power when compared to other state-of-the-art approximate compressor designs. The proposed approximate multiplier design has accuracy with 25% error rate and equal positive and negative absolute error deviation of 1.The proposed approximate multiplier shows a significant improvement in terms of area, power consumption and delay as compared to the existing approximate multiplier. In conclusion, this work has shown that multiplier can be implemented for approximate computing by an approximate design of a compressor; this proposed multiplier offers advantages in terms of design parameters compared to existing approximate multipliers, and in terms of accuracy metrics, area, delay and power consumption.

REFERENCES

[1] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and Probabilistic Adders,” IEEE Trans. Computers, vol. 63, no. 9, pp. 1760–1771, Sep. 2013.

[2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “IMPACT: IMPrecise adders for low-power approximate computing,” in Proc. Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 409–414.

[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,” presented at the IFIP Int. Conf. Very Large Scale Integ., Perth, Australia, Oct. 2005.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

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

[6] E. J. King and E. E. Swartzlander Jr., “Data dependent truncated scheme for parallel multiplication,” in Proc. 31st Asilomar Conf. Signals, Circuits Syst., 1998, pp. 1178–1182.

[7] P. Kulkarni, P. Gupta, and M. D. Ercegovac, “Trading accuracy for power in a multiplier architecture,” J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.

[8] C. Chang, J. Gu, and M. Zhang, “Ultra low-voltage low- power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits,” IEEE Trans. Circuits Syst., 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. IEEE 43rd Midwest Symp. Circuits Syst., 2000, vol. 3, pp. 1296–1298.

[10] Z. Wang, G. A. Jullien, and W. C. Miller, “A new design technique for column compression multipliers,” IEEE Trans. Comput., vol. 44, no. 8, pp. 962–970, Aug. 1995.

[11] J. Gu and C. H. Chang, “Ultra low-voltage, low-power 4-2 compressor for high speed multiplications,” in Proc. 36th IEEE Int. Symp. Circuits Syst., Bangkok, Thailand, May 2003, pp. v-321–v-324.

[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 ed. London, U.K.: Oxford Univ. Press, 2010.

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

[15] M. D. Ercegovac and T. Lang, Digital Arithmetic. Amsterdam, The Netherlands: Elsevier, 2003.

[16] D. Baran, M. Aktan, and V. G. Oklobdzija, “Energy efficient implementation of parallel CMOS multipliers with improved compressors,” in Proc. ACM/IEEE 16th Int. Symp. Low Power Electron. Design, 2010, pp. 147–152.

[17] D. Kelly, B. Phillips, and S. Al-Sarawi, “Approximate signed binary integer multipliers for arithmetic data value speculation,” in Proc. Conf. Design Architect. Signal Image Process., 2009, pp. 97–104.

[18] J. Ma, K. Man, T. Krilavicius, S. Guan, and T. Jeong, “Implementation of high performance multipliers based on approximate compressor design,” presented at the Int. Conf. Electrical and Control Technologies, Kaunas, Lithuania, 2011.

©IJIREEICE This work is licensed under a Creative Commons Attribution 4.0 International License