IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 60-64

Hybrid Recoding Driven Approximated Partial Product Reduction Multiplier for DIP Applications

[1]M.Induja, [2]C.Neha, [3]H.Mahashree, [4]Ms.G.Anuvidhya,
[1,2,3]UG students, [4]Assitatnt professor, Department of ECE
Apollo engineering College, Poonamalle.

Abstract : *In This Paper, We Analyze The Tradeoff Accuracy And Energy Conservation Of Fixed Width Multiplier In DIP Applications. In Most Cases Fixed Point Arithmetic Is Used To Keep The Dynamic Ranges Within The Certain Limit Based Accuracy Trade Off Over Resource Constraints Our Proposed Approximated Multiplier Use Consecutive 4:2 Compressor Units Which Applies On Partial Products Accumulated Based On Dadda Multipliers That Includes The Leading Ones From Two Operands To An M×M Multiplier. In Most Cases Direct Truncation Approach Identifies Only The MSB Part Leads Error Propagation. An Exact Approximation Followed By Tree Based PP Reduction Gives Considerable Complexity Reduction And Also Consumes Much Less Energy And Area Than Direction Truncation Approaches. The Efficiency Of This Improved Energy And Area Efficiency Are Further Improved Through Radix-8 Booth Recoding Method. Finally, We Proved The Performance Accuracy Through Hardware Synthesis And Exhaustive Test Bench Simulation Results. We Also Demonstrate That The Quality Retention Of 4:2 Compressor Level Approximation Circuit Using Image De-Noising Application And Its Impact On De-Noised Output Is Analyzed And It Is Well Matched With Exact Lossless Computation In Terms Of PSNR Reconstruction Quality Measure.*

Keywords *- 4:2 Compressors, Accuracy, Very Large Scale Integration (VLSI), Digital image processing (DIP).*

I. INTRODUCTION

The applications of the digital images are more and more extensive today than ever before. In recent years, the design for high speed with low power consumption has become one of the greatest challenges in high-performance very large scale integration (VLSI) design. As a consequence, many techniques have been introduced to reduce the power consumption with high performance of new VLSI systems. However, most of these methods focus on the power consumption and quality enhancement by replacing multipliers with distribute arithmetic approach [1]-[2]. But this increase the overall delay makes them unsuitable for high speed applications.

In the last decade many research have been carried out to reduce the computation complexity, and quality enhancement of truncation error. Most of them are efficient with software implementation, but unfortunately due to their irregular structure and complex routing, these algorithms are not suitable for VLSI implementation. Our goal is to design a high speed and area efficient approximated multiplier design most suitable for the application in image processing. To reduce the area of the circuit, the fixed-width multipliers will be only kept the most significant half part of the products. It will lead large error, many compensation methods are used to solve this problem [3]-[4]. This work is also permitted to design high-speed and unique MAC hardware structure using single adder unit, thereby making them suitable for any DSP applications. To prove the efficiency of the proposed multiplier unit it is compared with state-of-the-art methods like high speed vedic and high radix booth multipliers. Moreover, this methodology has several attractive features such as simplicity, regularity and modularity of architecture. Also, the Distributed arithmetic (DA) technique can be designed to meet high speed demand requirements of MAC design, where all bits of one tap unit are processed within the bounded delay. Here we attempt to increase the speed of multiplication by reducing number of partial products generated using high radix booth algorithm and combine it with partial addition of using approximated 4:2 compressor technique. The main concerns are speed and area optimization compensation in MAC computation.

II. FPGA IMPLEMENTATION

Field programmable gate arrays were actually invented only for prototyping the digital design which is later to be used in IC's. But in recent days FPGA's are started to use as a product in many fields. So Field programmable gate arrays are ideally suited for the implementation of fixed width mulplier based digital image processing. However, there are several issues that need to be solved. When performing fixed width computation, LSB truncations are carried out with finite precision loss. But in FPGA resources available to perform fixed

point arithmetic with full width precision is unjustified, and measures to be taken to account for this. Another concern is the partial product generation itself.

The two major objectives considered through approximation model are as follows:

- To propose hybrid multiplier by combining booth with dada multiplier to perform approximated multiplication.
- To achieve maximum speed and area optimization

III. BOOTH MULTIPLIER

In the modern world, we need system which will run at high speed. Multipliers play an important part in today's DSP and DIP applications. In many cases for image processing applications multiplications are used larger in number. Therefore, speed improvement in multiplier is important. Advances in technology have permitted many researchers to design multipliers which offer both high-speed and unique hardware structure, thereby making them suitable for specific VLSI implementation.

In any multiplication algorithm, the multiplication operation is carried out by summation of decomposed partial products. For high-speed multiplication we need to apply a booth radix recoding multiplication algorithm. In recent days in all high radix booth algorithm recoding is changed from 2s-complement format to a signed-digit representation from the defined set. This is called modified booth algorithm.

3.1 Radix-8 Algorithm

Here we reduce the number of partial products using a higher radix in the multiplier recoding. Recoding of binary numbers was first invented by Booth [5]. The modified Booth's algorithm is done by appending a zero to the right of M. Figure 2 shows recoding of 0101000102. In radix-8 recoding is similar to radix-4 [6] but here we take four bits instead of three bits and then we represent that coded values in signed-digit representation using table 1.

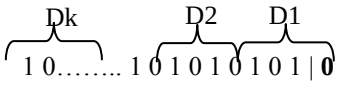

Fig. 1: Recoding representation

Where *k* is the number of partial products to be generated.

Table 1: Radix-8 sign digit values

Coded bits	signed-digit value
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

From the table 1 we need to have 2N, 3N, 4N and its 2's complement respectively.

3.2 Preprocessing Stage.

Both 2N and 4N is achieved by simple left shift of N. and 3N is calculated by adding 2N and N. If the bit width of N is high this will increase the delay in preprocessing stage. After this stage only we can generate partial products. In order to reduce the delay here we use Carry select adder. The resource used in CSA adder later will be used for partial product addition.

3.3 Dadda Tree

As we discuss in section 1 truncation errors will occur when we remove least significant bits from multiplied result. Here we use Wallace array to represent partial products array and its summation, which gives the multiplication result. First four partial products are processed using 4-2 compressor which is made up of two full adders. In later stages we used only Full adder for partial product addition.

In radix-8 partial products are left shifted by three bits. So in partial products some of the LSB bits will become 0's. It is predefined one. So these bits are not considered here in Wallace tree structure in order to save the hardware resource. In Wallace tree partial products are divided into main part and truncation part. Resource used in truncation part is reconfigured based on number of columns need to be selected for error compensation.

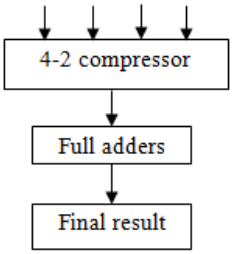

Fig.2: Dadda tree structure

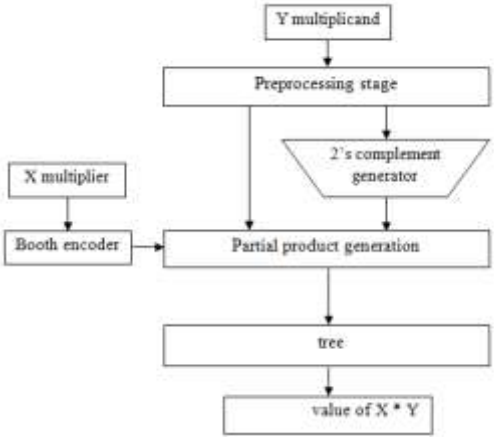

Fig. 3: Block diagram of proposed booth multiplier.

IV. DISTORTION ANALYSIS

The images can be distorted in embedding process because of changing pixel bits. Distortion is measured by means of two parameters namely, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

MSE can be calculated by,

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - Y_{ij})^2$$

The PSNR is calculated using,

$$PSNR = 10\log_{10}\left\{\frac{I^2_{\max}}{MSE}\right\}dB$$

Imax is the maximum intensity value of each pixel which is equal to 255 for 8 bit gray scale images. Higher value of PSNR leads to better image quality. Results of approximated multiplier driven image de-noising for Lena is represented in Figures 8.

The size of these images is 256*256 pixels. As shown here, message embedding is done with no dramatic changes in image quality.

Fig. 4: Simulated output

V. PERFORMANCE ANALYSIS

The image is converted into blocks using MATLAB and the values are stored as a text file. The text file is accessed by the Modelsim and the corresponding pixels values are then fed to image filtering design module which returns the data sequence. These data are written to a text file. The image can be reconstructed from the text file using MATLAB coding.

5.1 Speed and area trade off performance

Here both speed and hardware utilization rate is totally depends on the number of bits used in the input operands and the re-configuration rate. , and the key sizes used in each stage of hierarchical matching steps. The performance measure of multiplication units with various categories are shown in Table 2.

Table.2. Performance report using CYCLONE II FPGA family (EP2C35F672C6).

Multiplication type	Area(LE's used)	Fmax report
Direction truncation	377	125.58 MHz
4:2 compressor	213	140.81 MHz
Approximated 4:2 compressor	181	143.95 MHz
Booth recoded- tree model	163	404.86 MHz

Fig. 5: Area summary output

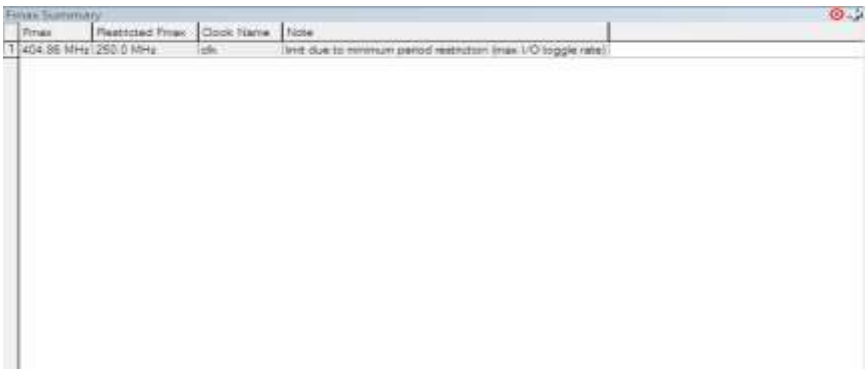

Fig. 6: Fmax summary output

VI. CONCLUSION

In this work, we proved the hardware efficiency of approximated tree multiplier by utilizing high-level transformation techniques with variation in modes. Here we construct the reconfigurable 4:2 compressor design which will reduce hardware complexity slightly as the as compared to conventional method. Compared with all other existing multiplication methods, in our proposed methodology, the booth recoding followed by tree based PP reduction will give significant complexity reduction and speed enhancement as well. We analyze the complexity trade-off between the direct truncation and 4:2 compressor models. Finally in order to prove the validity of approximation image de noising is evaluated and its metrics are compared with exact computation methods.

REFERENCES

[1] S. Yu and E. E. S. ,Jr., "DCT implementation with distributed arithmetic"IEEE Trans. Comput., vol. 50, no. 9, pp. 985–991, Sep. 2001.
[2] P. K. Meher, "Unified systolic-like architecture for DCT and DST using distributed arithmetic," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 12, pp. 2656–2663, Dec. 2006.
[3] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixedwidth multipliers," IEEE Trans. Circuits Syst. I, vol. 52, no. 8, pp. 1608– 1619, Aug. 2005.
[4] C. H. Chang and R. K. Satzoda, "A low error and high performance multiplexer-based truncated multiplier," IEEE Trans. VLSI Syst., vol. 18, no. 12, pp. 1767–1771, Dec. 2010.
[5] C. Lu, S. P. A.D. Booth, "A signed binary multiplication technique," Quarterly J. Mechan. Appl. Math., vol IV. Part 2, 1951.
[6] Yuan-Ho Chen and Hsin-Chen Chiang "Modified Booth Encoding Radix-4 8-bit Multiplier" .2012
[7] Prasad, K,"Low-power 4-2 and 5-2 compressors" vol. 1, pp. 129 – 133, Nov. 2001.
[8] Jongsun Park · Kaushik Roy "A Low Complexity Reconfigurable DCT Architecture to Trade off Image Quality for Power Consumption", Springer Science Business Media, April 2008.
[9] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, Nov. 2015, pp. 418–425.
[10] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.
[11] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired 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.
[12] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.
[13] C. H. Lin and I. C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Comput. Design (ICCD), Oct. 2013, pp. 33–38.