International Journal of Scientific Research in Computer Science, Engineering and Information Technology

© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

Approximate Radix-4 Booth Multipliers for Error Analysis

M. Sainath[1], P. Muralikrishna[2]

[1]M.Tech Student, Department of ECE, SKU College of Engineering, Anathapur, Andhra Pradesh, India

[2]Assistant Professor, Department of ECE, SKU College of Engineering, Anathapur, Andhra Pradesh, India

ABSTRACT

The multiplication operation is found in many elements of a digital machine or digital computer, most appreciably in sign processing, portraits and scientific computation. With advances in era, diverse strategies have been proposed to layout multipliers, which give high speed, low power intake and lesser region.. Approximate radix-4 modified Booth encoding (MBE) algorithms and a regular partial product array that employs an approximate Wallace tree this quick gives Approximate booth multiplier layout the usage of the radix4 modified Booth encoders R4ABE1 and R4ABE2. The use of those sales space encoders we layout the approximate multipliers .In this approximate multipliers generate the mistakes the usage of NMED error tolerant computing is analyzed via using two approximate Booth encoders with recognize to approximate component. Results are shown in XILINX 14.3 ISE.

Keywords : Radix-4 Multiplier, Booth Encoder, Approximate Computing, Low Power.

I. INTRODUCTION

In arithmetic operations Multiplier plays a major role. With the rapid advances in multimedia and communication systems, real-time signal processing and large capacity data processing are increasingly being demanded. The multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined more by the multiplier. Furthermore, multiplier consumes much area and dissipates more power. An error is compensated with the outputs of the Booth encoders in the error compensation circuit proposed in mainly uses a simplified sorting network. Multipliers typically consist of a partial product matrix (PPM), which accumulates the partial products and reduces them to just two operands, and a last stage Carry Propagate Adders (CPA). The speed of multiplication can be bit (LSB) position of each patial product increased by reducing the number of partial products and/or accelerating the accumulationof partial products. approaches namely Booth algorithms using Wallace Tree 4-2 . Compressors, Carry Propagate Adders (CPA).Error tolarent Application we are using the Normalization of mean error distance . The approximate design of a radix-4 Booth multiplier as one of the most popular schemes for signed multiplication. a radix-4 Booth multiplier, a radix- 4 modified Booth encoding (MBE) is used to generate the partial products . a radix-4 MBE can reduce the number of partial products by a factor of two.

The implementation of the MBE significantly affects the area, delay and power consumption of Booth multiplier in the traditional MBE algorithm, an extra partial product bit is generated at the least significant of row due to the negative encoding. This leads to an irregular partial product array as requiring a complex reduction tree. A more efficient approximate radix-4 Booth encoder is proposed in this paper. The designs of both approximate radix-4 Booth encoders are presented and extensively analyzed.

Block	Partial Product
000	0
001	1*multiplicand
010	1*multiplicand
011	2*multiplicand
011	2*multiplicand
100	-2*multiplicand
101	-1*multiplicand
110	-1*multiplicand
111	0

Table 2 : Radix-4 Booth Encoding Table

Approximate Booth multipliers are proposed using approximate Booth encoders, in which the features of an approximate regular tree structure are illustrated in detail. An approximation factor is proposed to assist in the design of the approximate Booth multipliers and facilitate its error analysis. The proposed approximate Booth multipliers are comprehensively evaluated with respect to both hardware implementation and error analysis. The proposed approximate Booth multipliers are applied in image processing.

The multiplicand encoding process using Radix -4 Booth algorithm is based on the multiplier bits .It will compare 3 bits at a time with overlapping technique. Grouping starts from the LSB, and the first block uses only two bits of the multiplier and assumes a zero for the third bit. It consists of eight different types of states as we are comparing 3bits at a time and during these states we can obtain the outcomes, which are multiplication of multiplicand with 0,-1 and -2 consecutively. The state diagram presents various logics to perform the Radix-4 Booth multiplication in different states as per the adopting encoding technique.

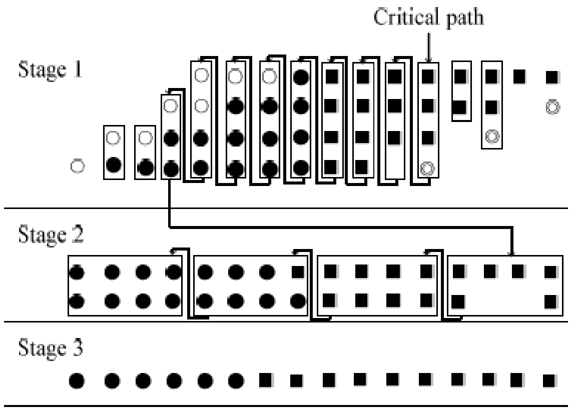


Figure 1 : radix-4 booth multiplication

II. APPROXIMATE RADIX-4 BOOTH MULTIPLIER

Radix-4 Booth multiplier in which they are two binary inputs as input, one is multiplicand and another one is multiplier. If both binary numbers are positive then it will perform two booth encoders. Approximate Booth multipliers are designed based on the approximate Booth encoder and the regular approximate partial product array.

Booth encoding has been proposed for improving the performance of multiplication of twos complement binary numbers ,it has been further improved by the MBE or radix-4 Booth encoding . The Booth encoder plays an important role in the Booth multiplier, which reduces the number of partial product rows by half. Approximate radix-4 booth multiplier designed based on two approximate booth encoders those are R4ABE1 and R4ABE2 .Approximate radix-4 Booth multiplier.

A. APPROXIMATE RADIX-4 BOOTH ENCODING METHOD 1

The advantage of the R4ABE1 design is that a very small error occurs, In the approximate Booth multiplier, the proposed approximate Booth encoder R4ABE1 used in the first part to generate the inexact partial products. The approximate Booth encoders can then be used in all or only part of the partial product generation process; therefore, an approximation factor p (p=1, 2, ..., 2N) is defined as the number of least significant partial product columns that are generated by the approximate Booth encoder.

This Booth encoder is Conventional design of Modified Booth encoder .The conventional Modified Booth encoder has More delay but this Booth encoder has less delay . This encoder reduce the complexity Then the conventional MBE . as only entries are modified however, all modifications change a '1' to a '0'. The absolute value of approximate product is always larger than its exact.

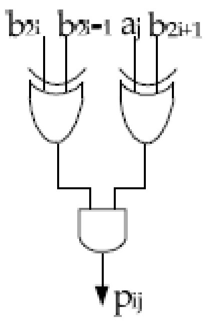


Figure 2 : Radix -4 Approximate booth encoder

The approximate radix-4 Booth multiplier (R4ABM1) uses R4ABE1 (to generate the p least significant partial product columns) and the regular approximate partial product array. The exact MBE is used for generating the 2N-p most significant partial product columns. The exact 4-2 compressors are used to accumulate both approximate and exact partial products.

Approximate radix-4 Booth multiplier (R4ABM2) uses R4ABE2 (to generate the p least significant partial product columns). The regular approximate partial product array and the exact 4-2 compressors. The error is controlled by the approximation factor a counterpart. Compared with the exact MBE ,R4ABE1 can significantly reduce the critical path delay of Booth encoding.

B. APPROXIMATE RADIX-4 BOOTH ENCODING METHOD 2 :

R4ABE1, the modification is achieved by not only changing a '1' to a '0', but also changing a '0' to a '1'. The approximate product can be either larger or smaller. Than the exact product and errors can complement each other in the partial product reduction process. R4ABE2 in a Booth multiplier, the error may not be larger than for a Booth multiplier with R4ABE1.

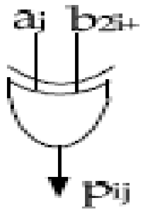


Figure 3 : Radix -4 approximate booth encoder 2 reasonable accuracy can be achieved for different applications.

C. APPROXIMATE WALLACE TREE DESIGN

The reduction of partial products using full adders as 4-2 compressors became generally Known as the “Wallace Tree ”.This architecture reduce the partial product rate .The tree reduction for an 8*8 bit partial product tree, The ovals around The dots represent either a full adder or a Half adder .This is reduce the need of carry propagation in the adder avoid latency of one addition is equal to gate delay of adder.

The chip/system designers add accuracy as a new constraint to optimize Latency-Power-Area (metrics.

In this paper, we present a new power and area-efficient Approximate Wallace Tree structure (AWTs) for error-tolerant applications.

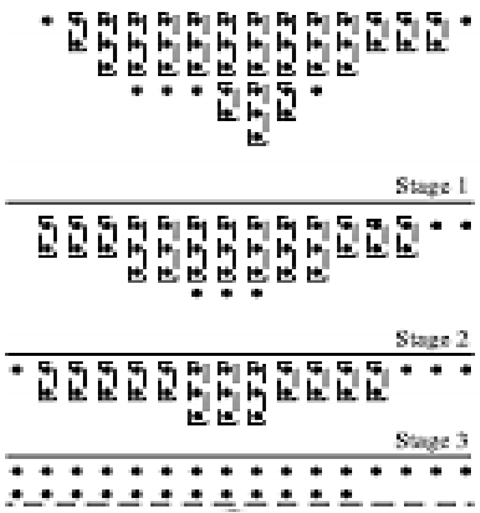


Figure 4 : Approximate Wallace tree structure

D. 4-2 COMPRESSOR:

There are various ways to implement the 4:2 compressor The simplest of which is cascading two full adder cells together. this structure was a popularized by santoro who built a 64*64 bit multiplier . the inter connection of these full adder must not make the carry out dependant on the carry in . it takes two full adders to build one 4:2 compressor .The speed of each 4:2 compressor is limits' by the delay of 3 XOR gates in series. Thus ,a 4:2 is slightly faster than the two full adder .Thus the tree compression is slightly faster using 4:2 with a trade off a little more hardware.

We propose a bit-width aware approximate multiplication algorithm for optimal design of our AWT. We employ a carry-in prediction method to reduce the critical path. It is further augmented with hardware efficient precomputation of carry-in. We also optimize our design for latency, power and area using Wallace trees.

Accuracy as well as design metrics are used to evaluate our approximate design of different bit widths, 4x4, 8x8 and 16x16. The simulation results show that we obtain a mean accuracy single cycle implementation of AWT gives reduction in latency. We achieve significant reduction in power and area that clearly demonstrates the merits of our proposed AWT design. Finally AWT is used to perform a real time application on a benchmark image.we obtain up to reduction in power and reduction in area without any loss in image quality.

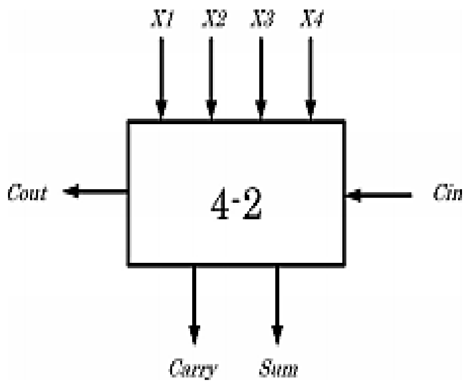


Figure 5

Compressor has 5 inputs A, B, C, D and Cin to generate 3 outputs Sum, Carry and Cout. The 4 inputs A, B, C and D and the output Sum have the same.
weight. The input Cin is output from a previous lower significant compressor and the Cout output is for the compressor in the next significant stage.

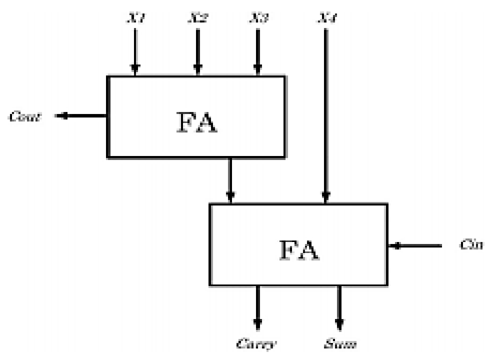


Figure 6 : 4-2 Compressor

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. An-2 compressor is usually a slice of a circuit that reduces n numbers to two numbers when properly replicated. A widely used structure for compression is the 4-2 compressor.

The give the outputs of the 4- 2 compressor, The common implementation of a 4-2 compressor is accomplished by utilizing two full-adder (FA) cells

$$Sum = x1 \oplus x2 \oplus x3 \oplus x4 \oplus Cin$$

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

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

E. CARRY LOOK AHEAD ADDER:

The ripple-carry adder, its limiting factor is the time it takes to propagate the carry. The carry look-ahead adder solves this problem by calculating the carry signals in advance, based on the input signals. The result is a reduced carry propagation time. To be able to understand how the carry look-ahead adder works, we have to manipulate the Boolean expression dealing with the full adder.

The Propagate P and generate G in a full-adder, is given as:
Pi = Ai ⊕ Bi Carry propagate
Gi = AiBi Carry generate
Notice that both propagate and generate signals depend only on the input bits and thus will be valid after one gate delay.
The new expressions for the output sum and the carryout are given by:
Si = Pi ⊕ Ci-1
Ci+1= Gi + PiCi T
Although it is impractical to have a single level of carry look –ahead logic for log adder
Hence equations show that a carry signal will be generated in two cases:

- if both bits Ai and Bi are 1
- if either Ai or Bi is 1 and the carry-in Ci is 1.

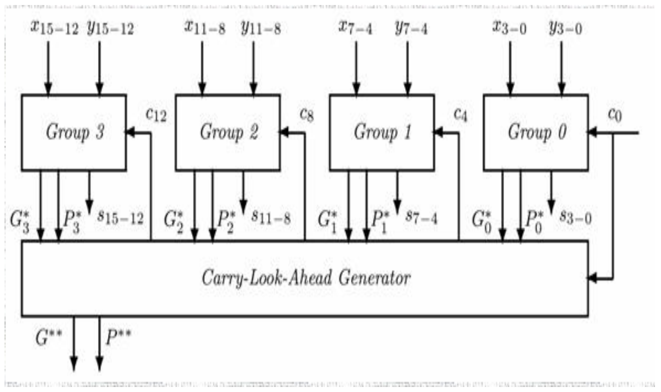


Figure 7 : carry look ahead adder

To make the carry generator from 4 bits to n bits, we need only add AND gates and inputs for the OR gate. The largest AND gate in the carry section has always n+1 inputs and the number of AND gates requirements is n. Therefore the design of a 16 bits adder needs the last carry generator section to have 16 AND gates, where the biggest AND gate has 17 inputs in group 1,2,3,4 the full adder will be there.

ERROR ANALYSIS AND EVALUATION:

The approximate designs, several metrics have been proposed to measure the error of approximate adders and multipliers including the mean error distance (MED), the relative error distance (RED) and the normalization of MED (NMED). Several error metrics are used to fairly compare different approximate designs of various sizes.wher ED =(M'-M),RED is ED/M.

Parallel-prefix adders, also known as carry-tree adders, pre-compute the propagate and generate signals. These signals are variously combined using the fundamental carry operator Due to associative property of the fco, these operators can be combined in different ways to form various adder structures.
The NMED is defined as the normalized MED by the maximum output of the accurate design. RED is

defined as the ED over the absolute accurate result. Mean RED (MRED) and PRED are usually used to evaluate the error distribution of approximate multipliers, where PRED is the probability of obtaining a RED smaller than a specific percentage value that is assumed throughout the paper.
The error metrics of the proposed approximate multipliers are provided in Table 5 for 8-bit and 16-bit designs; the accuracy of both approximate multipliers decreases. MRED increases with an increase and the MRED of R4ABM2 is generally larger than that of R4ABM1.

PRED (the probability of getting a RED smaller) of both the 8-bit R4ABM1 and R4ABM2 decrease rapidly when 4<p<12 the error rate is increased. The error rate of R4ABM1 is smaller than for R4ABM2 at the same value because the error rate of R4ABE-1 is significantly smaller than for R4ABE2 .

III. EXTENSION

It has been performed the design and implementation of a radix-8 booth multiplier. It has been proved that it can be useful to apply a radix-8 architecture in high speed multipliers because of the gain in time obtained due to reduction of partial products to n/3.The use of a radix-8 recoding is the less number of transistors resulting in a reduced power dissipation and area size.

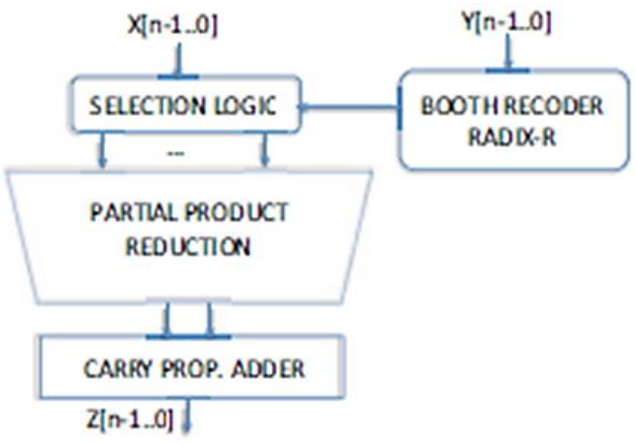


Figure 8 : Radix -8

.This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are Another important carry-tree adder known as the Brentkung adder also examined.

Brent-Kung adder is a very popular and widely used adder. It actually gives an excellent number of stages from input to all outputs but with asymmetric loading of Intermediate stages. It is one of the parallel prefix adders.

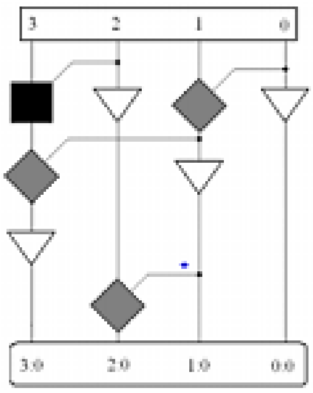
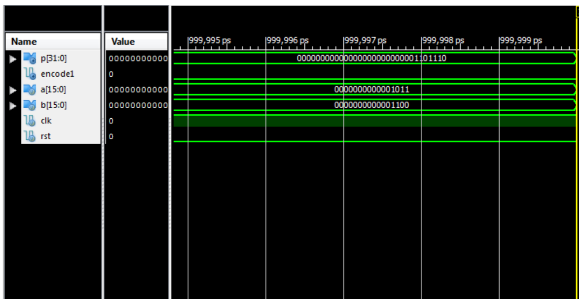


Figure 9 : Brentkung adder

It is one of the parallel prefix adders where these adders are the ultimate class of adders that are based on the use of generate and propagate signals. In case of Brent kung adders along with the cost, the wiring complexity is also less.

IV. RESULTS



```
p[31:0]    00000000000   00000000000000000000000001101110
encode1    0
a[15:0]    00000000000   0000000000001011
b[15:0]    00000000000   0000000000001100
clk        0
rst        0
```

Figure 10 : R4BM1

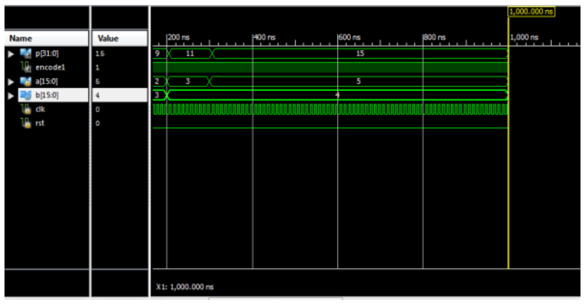


Figure 11 : R4BM2

V. CONCLUSION

Designs of approximate radix-4 Booth multipliers The aim of paper is design and implementation of the optimized ApproximateRadix-4 Booth multiplier. Proposed method is based on Booth encoders, 4-2 compressor and CLA, thus it is more area and more delay. The approach used for increase performance we use prefix adder design. This method has more speed.

VI. REFERENCES

1. S Venkataramani, S. T. Chakradhar, and K. Roy. "Computing approximately, and efficiently," Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp.748-751.
2. J Han and M. Orshansky, "Approximate computing: an emerging paradigm for energy-efficient design," Proc. 18th IEEE European Test Symposium, 2013, pp.1-6.
3. H Mahdiani, A. Ahmadi, S. 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, pp. 850-862, 2010.
4. V Gupta, D. Mohapatra, S. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise Adders for Low-Power Approximate Computing," Proc. Int. Symp. Low Power Electronics and Design (ISLPED), pp. 1-3, 2011.
5. W Liu, L. Chen, C. Wang, M. O'Neill and F. Lombardi, "Design and analysis of floating-point adders", IEEE Trans. Computers, vol. 65, pp. 308-314, Jan. 2016.
6. J Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Computers, vol. 63, pp. 1760-1771, Sep. 2013.
7. A Booth, "A signed binary multiplication technique," Quarterly J. Mechanics and Applied Mathematics, vol. 4, pp/236-240, June 1951.
8. O MacSorley, High-speed arithmetic in binary computers, Proc. IRE, vol. 49, pp. 67-91, 1961.
9. K-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low error fixed-width modified Booth multiplier," IEEE Trans. VLSI Systems, vol. 12, no. 5, pp. 522-531, 2004.
10. M. J. Schulte and E. E. Swartzlander Jr., "Truncated multiplication with correction constant," Proc. Workshop VLSI Signal Process. VI, 1993, pp. 388-396.