International Journal of Computer Applications Technology and Research
Volume 2– Issue 6, 743 - 747, 2013, ISSN: 2319–8656

A New Design of Multiplier using Modified Booth Algorithm and Reversible Gate Logic

K.Nagarjun
Department of ECE
Vardhaman College of Engineering,
Hyderabad, India

S.Srinivas
Department of ECE
Vardhaman College of Engineering,
Hyderabad, India

Abstract: In this paper we propose a new concept for multiplication by using modified booth algorithm and reversible logic functions. Modified booth algorithm produces less delay compare to normal multiplication process. Modified booth algorithm reduces the number partial products which will reduces maximum delay count a the output. by combining modified booth algorithm with reversible gate logic it will produces further less delay compare to all other. In the past years reversible logic functions has developed as an important research area. Implementing reversible logic has the advantage of reducing the gate count, garbage outputs as well as constant inputs. Addition subtraction operations are realized using reversible DKG gate. This modified booth algorithm with reversible gate logic are synthesized and simulated by using Xilinx 13.2 ISE simulator.

Keywords: MBE, Reversible logic gate, DKG, reversible logic circuits. Carry save adder tree.

1. INTRODUCTION

The Continuous advances of the microelectronic technologies makes better use of input energy, to encode the data more efficiently, to transmit the information more faster and reliable, etc. In Particular, many of these technologies handle low power consumption to meet the requirements of various outboard applications. In these applications, a multiplier is a fundamental arithmetic unit and used in a great extent in circuits. The fastness of multiplication and addition arithmetic's decides the execution speed and performance of the total calculation. Because the multiplier needs the longest delay within the basic operational blocks in digital systems, the critical path is evaluated by multiplier in general. For high speed multiplication, the modified radix-4 booth's algorithm (MBA)[1] is generally used however this cannot completely solve the problem.

In general, the multiplier uses Booth's algorithm and array of full adders (FAs)[3], or Wallace tree rather than the array of FAs., i.e., this multiplier primarily consists of the three parts: Booth encoder[6], a tree to compact the partial products such as Wallace tree[1], and the final adder[1]. Because Wallace tree is to add the partial products. The most efficient way to gain the fastness of a multiplier is to cut down the number of the partial products as multiplication proceeds a series of additions for the partial products. To cut down the number of calculation steps for the partial products, MBA[1] algorithm has been employed mostly where Wallace tree has taken the role of increasing the fastness to add the partial products. To increase the fastness of the MBA algorithm, many parallel multiplication architectures have been explored and they have employed to various digital filtering calculations.

here In this design we use reversible logic gates[2] in the place of full adders to reduce the power and delay. A reversible logic circuit should have features like use minimum number of reversible gates[2], use minimum number of garbage outputs, use minimum constant inputs.

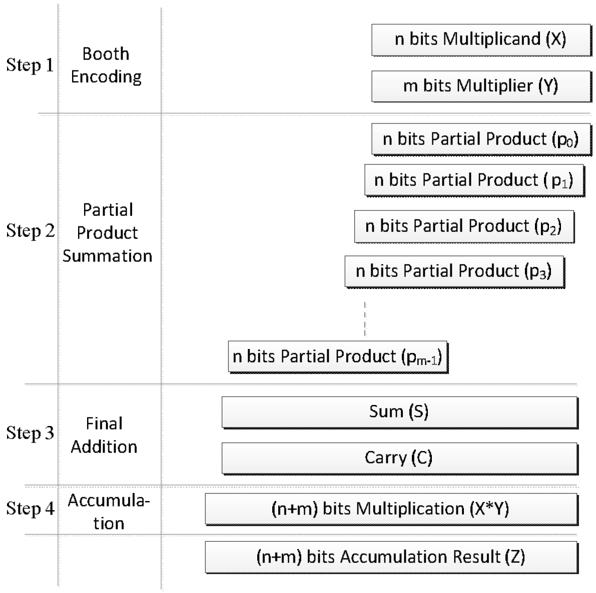

Step 1
Booth Encoding
n bits Multiplicand (X)
m bits Multiplier (Y)
Step 2
Partial Product Summation
n bits Partial Product (p_0)
n bits Partial Product (p_1)
n bits Partial Product (p_2)
n bits Partial Product (p_3)
n bits Partial Product (p_{m-1})
Step 3
Final Addition
Sum (S)
Carry (C)
Step 4
Accumula-tion
(n+m) bits Multiplication (X*Y)
(n+m) bits Accumulation Result (Z)

Fig 1: Basic arithmetic steps for multiplication and accumulation

In this paper, A New Design of Multiplier using Modified Booth Algorithm And Reversible Gate Logic are implemented.Section2 discusses overview of MAC.Section3 introduces reversible logic gate function Section4 covers the proposed MAC and architecture of CSA tree. Experimental results are showing the simulation results of proposed design.

2. OVERVIEW OF MAC

In this section, general basic MAC[1] operation is introduced. A multiplier can be divided into three functional steps. The first is radix-2 modified Booth encoding[1] in which a partial product is produced from the multiplicand and the multiplier. second is the adder array or partial product compression to add all the partial products and change them into the form of sum and carry. The last step is the final addition in which the final multiplication result is produced by adding the sum and carry. If the process to accumulate the multiplied results is included, a MAC consists of four steps, as shown in Fig. 1, which shows the operational steps.

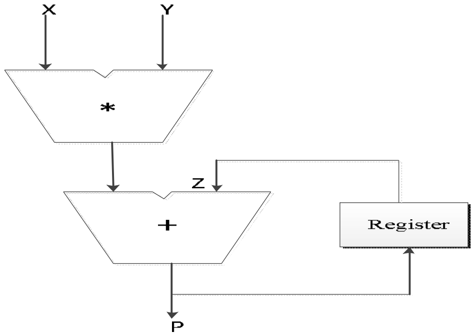


Fig 2: Hardware architecture of general MAC

A Basic hardware architecture of this MAC is shown in Fig. 2. It carries out the multiplication operation by multiplying the input multiplier X and the multiplicand Y. This result is added to the previous multiplication result Z as accumulation step. Fig. 2. Hardware architecture of the general MAC. The -bit 2's complement binary number can be expressed as

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i \quad x_i \in 0,1. \qquad \text{…. (1)}$$

If the equation (1) is expressed in base-4 type redundant sign digit form to apply the radix-2 Booth's algorithm, it would be.

$$X = \sum_{i=0}^{N/2-1} d_i 4_i \qquad \text{….. (2)}$$

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1.} \qquad \text{….. (3)}$$

If the equation (2) is used, multiplication can be expressed as

$$X \times Y = \sum_{i=0}^{N/2-1} d_i 2^{2i} Y \qquad \text{…..(4)}$$

If these equations are used, the before mentioned multiplication accumulation results can be expressed as

$$P = X \times Y + Z = \sum_{i=0}^{N/2-1} d_i 2^i Y + \sum_{j=0}^{2N-1} z_i 2^i. \qquad \text{…….(5)}$$

Each of two terms on the right side of (5) is evaluated independently and the final result is produced by adding the two results. The MAC architecture enforced by (5) is called the standard design . If the N-bit data are multiplied, the number of generated partial products is proportional to N . In order to add that partial products serially, the time of execution is also proportional to N . The fastest architecture of a multiplier, which uses radix-2 Booth encoding to generate partial products and a Wallace tree based on CSA as the adder array to add the partial products. If radix-2 Booth encoding is used, the number of partial products, i.e., the inputs to the Wallace tree, is reduced to half, resulting in the decrease in CSA tree step. In addition, the signed multiplication based on 2's complement numbers is also possible. Due to these reasons, most current used multipliers adopt the Booth encoding.

3.Reversible Logic DKG Gate

Reversible function is the main objective of the reversible logic theory. A 4* 4 reversible logic DKG[2] gate that can work uniquely as a reversible Full adder and a reversible Full subtractor is shown in Fig 3. It can be verified that input pattern representing to a particular output pattern can be uniquely determined. If the input A=0, the proposed gate acts as a reversible Full adder DKG gate, and if input A=1, then it acts as a reversible Full subtractor DKG gate. It has been proved that at least the two garbage outputs are required by the reversible full-adder circuit to make the output combinations unique.

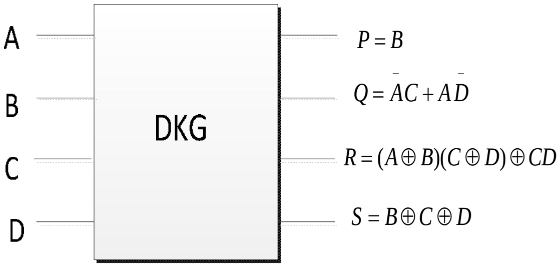

A
B
C
D
DKG
$P = B$
$Q = \bar{A}C + A\bar{D}$
$R = (A \oplus B)(C \oplus D) \oplus CD$
$S = B \oplus C \oplus D$

Fig 3: Reversible DKG gate

The binary full adder/subtractor is capable of handling the one bit of each input along with a carry in/borrow in generated as a carry out/ borrow from addition of previous lower order bit position. n binary full adders/subtractors are cascaded to add two binary numbers each of n bits.

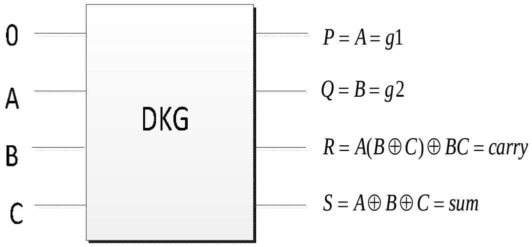

0
A
B
C
DKG
$P = A = g1$
$Q = B = g2$
$R = A(B \oplus C) \oplus BC = carry$
$S = A \oplus B \oplus C = sum$

Fig 4: DKG gate implemented as Full adder

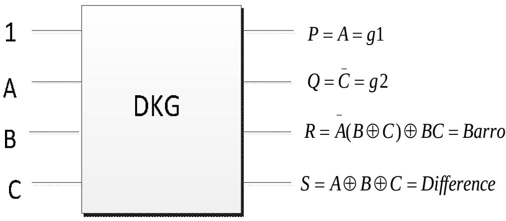

1
A
B
C
DKG
$P = A = g1$
$Q = \bar{C} = g2$
$R = \bar{A}(B \oplus C) \oplus BC = Barrow$
$S = A \oplus B \oplus C = Difference$

Fig 5: DKG gate implemented as Full subtractor

A Parallel adder/subtractor is an cascaded of full adders/subtractors and inputs are simultaneously applied. The carry/borrow produced at a stage is propagated to the next stage. When the control input A=0, the circuit behaves as a parallel adder, generates a 4 bit sum and a carry out, as shown in Fig 4. If the control input A=1, the circuit behaves as a parallel subtractor, generates a 4 bit difference and a borrow out, as shown in Fig 5.

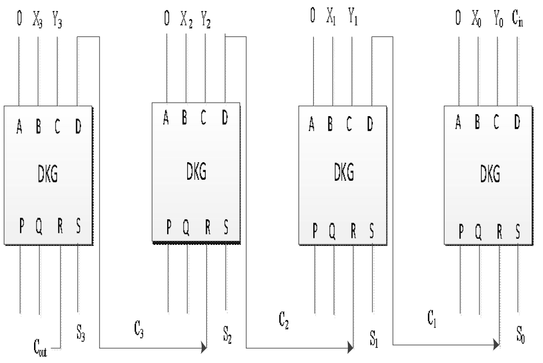


Fig 6: Reversible Parallel adder when A/S=0

4. PROPOSED ARCHITECTURE

In this section, By examining the various architectures, we have observed that delay can be improved by using higher radix MBA which reduces number of partial product rows that eventually reduces number of multiplication thereby improving speed. Thus we propose a new high speed and area efficient MAC architectures which will be an improvement over the existing Architecture by replacing Radix-2 with Radix-4 and Modified Booth Encoder in the Multiplication Stage. The output of multiplication and accumulation stages will be combined using hybrid reduction through CSA, CLA and HA which enhancing speed and efficiency. In addition, the Final Stage of CSA will include a New Reversible logic DKG gate was replaced a full adder. The new adder will be an improvement over the existing CSA. by replacing Reversible logic DKG[2] gate in CSA The design will be implemented using Verilog language and simulated using Xilinx ISE13.2 Simulator. We expect the proposed MAC will be useful in high performance signal processing system.

In Fig. 7 the proposed architecture were shown and it contain booth encoder and CSA & Accumulator tree and the tree contains half adder reversible logic DKG gate which will reduce the delay of the adder and a final adder to add the final sum and carry.

The architecture of the hybrid-type CSA that complies with the operation of the proposed MAC is shown in Fig.8, which performs 8X8-bit operation. It was formed based on the previous architectures. In Fig.8, S_i is to simplify the sign expansion and N_i is to compensate 1's complement number into 2's complement number. S[i] and C[i] and correspond to the i th bit of the feedback sum and carry. Z[i] is the i th bit of the sum of the lower bits for each partial product that were added in advance and Z[i]is the previous result. In addition corresponds to the i th bit of the I th partial product. Since the multiplier is for 8 bits, totally four partial products(P_0[7:0]~P_3[7:0]) are generated from the Booth encoder.

,

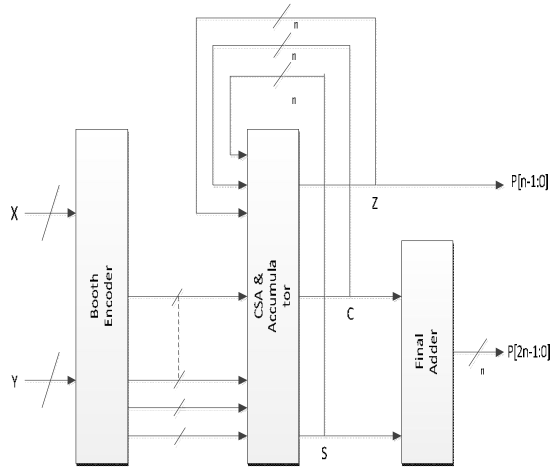


Fig 7: Hardware architecture of the proposed MAC

This CSA requires at least four rows of DKG full adders for the four partial products. Thus, totally five DKG full adder rows are necessary since one more level of rows are needed for accumulation. For an 8-bit MAC operation, the level of CSA is (n/2+1) . The white square in Fig. 8 represents an reversible DKG full adder and the gray square is a half adder (HA). The rectangular symbol with five inputs is a 2-bit CLA with a carry input.

The critical path in this CSA is determined by the 2-bit CLA. It is also possible to use FAs to implement the CSA without CLA. However, if the lower bits of the previously generated partial product are not processed in advance by the CLAs, the number of bits for the final adder will increase. When the entire multiplier or MAC is considered, it degrades the performance.

the characteristics of the proposed CSA architecture have been summarized and briefly compared with other architectures. For the number system, the proposed CSA uses 1's complement, but ours uses a modified CSA array without sign extension. The biggest difference between ours and the others is the type of values that is fed back for accumulation. Ours has the smallest number of inputs to the final adder.

In the carry save adder architecture more number of full adders were used and that will replaced by reversible DKG gate in the proposed system by this the overall performance was further increased the half adders in carry save adder were used as it is because no bigger difference in the normal half adder and reversible DKG half adder.

In addition, we compared the proposed architecture with that the previous booth multiplier. Because of the difficulties in comparing other factors, only delay is compared. The sizes of both MACs were 8X8 bits and implemented by using Xilinx 13.2 simulator. The delay of ours was 30.38ns while in previous it was 34.35ns,which means that ours improved about 4.2ns of the delay performance. This improvement is mainly due to the reversible logic DKG full adder. The architecture from the previous should include a normal full adders that will consumes maximum delay for that we used is very effective and gives maximum performance.

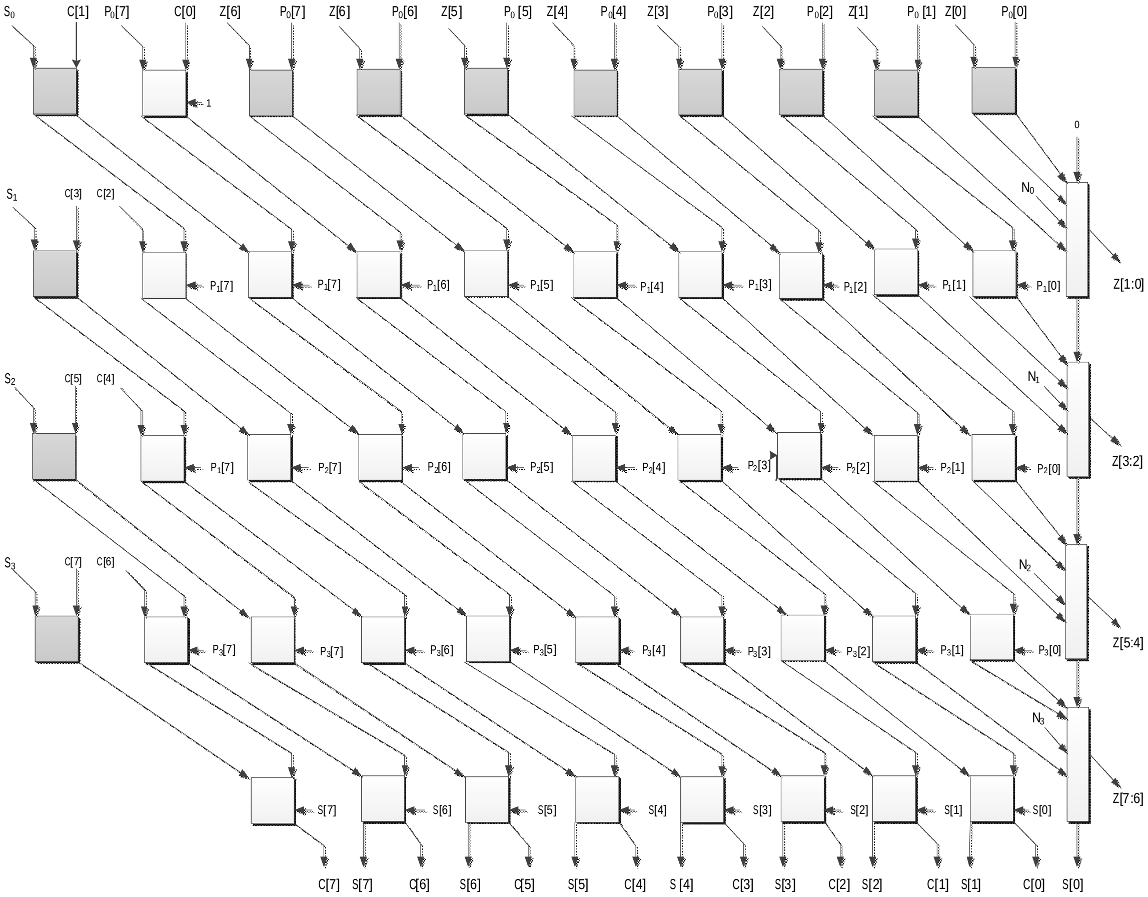


Fig 8: Hardware architecture of proposed CSA tree

5. EXPERIMENTAL RESULTS

The proposed architecture was simulated and synthesized by using Xilinx 13.2 tool and cadence virtuoso

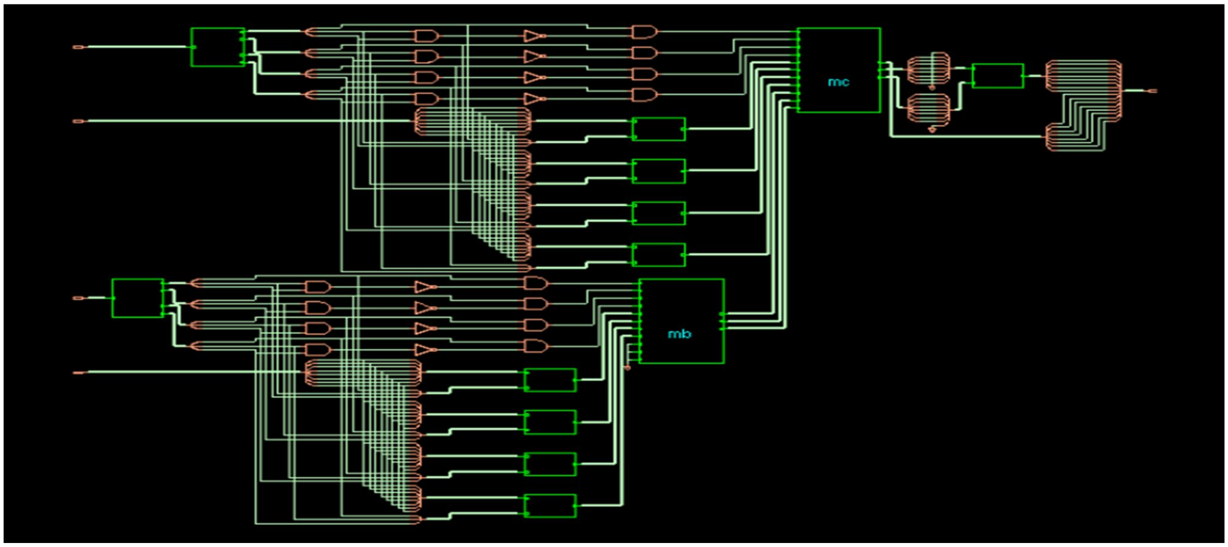
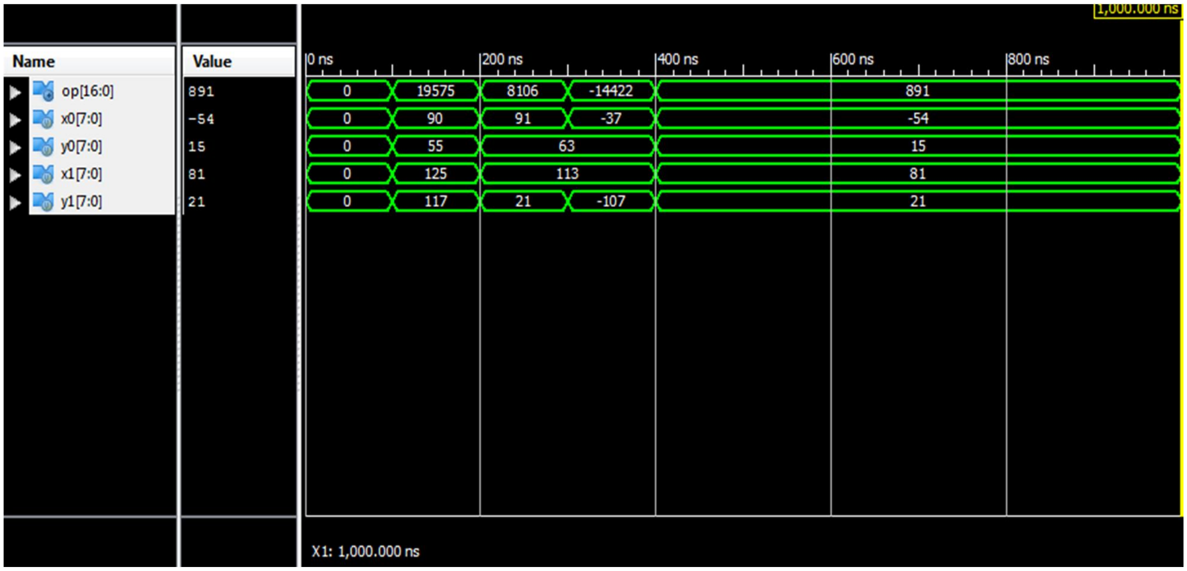


Fig 9: TOP Level Schematic of proposed MAC



Name	Value
op[16:0]	891
x0[7:0]	-54
y0[7:0]	15
x1[7:0]	81
y1[7:0]	21

Fig 10: Output of Modified booth algorithm with reversible gate logic

Table 1.Comparison table of different parameters

Parameter	MBE [1]	MBE with reversible gate logic
No. of slices used	178 out of 4656 (3%)	162 out of 4656 (3%)
No. of LUT’s	355 out of 9312(3%)	294 out of 9312 (3%)
Total Delay	34.535ns	30.38ns

6.CONCLUSION

In this paper, a new MAC architecture to perform the multiplication-accumulation, To efficiently process the digital signal processing and multimedia application this architecture was proposed. the overall MAC performance has been improved By eliminating the independent accumulation process that has the greatest delay and merging it to the compression process of the partial products, almost twice as much as in the previous work and by replacing full adder in the CSA with reversible logic gate further improves the performance. The proposed hardware was implemented and synthesized through Xilinx ISE 13.2 tool. Consequently, the proposed architecture can be used effectively where we requiring high throughput such as a real-time digital signal processing.

7.REFERENCES

[1] “A New VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2 Modified Booth Algorithm “Young-Ho Seo, Member,IEEE,and Dong-Wook Kim, Member, IEEE

[2] “A Distinguish Between Reversible And Conventional Logic Gates” B.Raghu Kanth, B.Murali Krishna , M. Sridhar , V.G. Shanti Swaroop,

[3] O. L. MacSorley, “High speed arithmetic in binary computers,” Proc.IRE, vol. 49, pp. 67–91, Jan. 1961.

[4] S. Weser and M. J. Flynn, Introduction to Arithmetic for Digital Systems Designers. New York: Holt, Rinehart and Winston, 1982.

[5] A.R. Omondi ,Computer Arithmetic Systems. Englewood Cliffs, NJ:Prentice-Hall, 1994.

[6] A.D.Booth,“A signed binary multiplication technique,” Quart. J.Math., vol. IV, pp. 236–240, 1952.

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

[8] A.R. Cooper, “Parallel architecture modified booth multiplier”, Proc. Inst. Electr Engg. Vol. 135, pp 1251-1988.