Area Optimized Radix-2 8-Bit Reversible Booth Multiplier

Chepuru Sai Priyanka[1], T.K. Kalaiarasan[2]
[1, 2] Department of ECE
[1, 2] Swetha institute of Technology & science, tirupati

Abstract- *Reversible logic attains the attraction of researchers in the last decade mainly due to low-power dissipation and area efficient. Designers 'endeavours are thus continuing in creating complete reversible circuits consisting of reversible gates. This paper presents a design methodology for the realization of 8 bit Booth's multiplier in reversible mode. Booth's multiplier is considered as one of the fastest multipliers in literature have shown an efficient design methodology in reversible paradigm. The proposed architecture is capable of performing both signed and unsigned multiplication of two operands without having any feedbacks, whereas existing multipliers in reversible mode consider loop which is strictly prohibited in reversible logic design. This paper of 8x8 reversible Radix-2 Booth's Multiplier is designed in VHDL and synthesised and simulated in ALTERA QUATRUS –II 9.1. From the results have shown that the proposed design used 140 Combinational ALUTs and operating with 43.64MHz frequency.*

I. INTRODUCTION

The field of reversible logic is achieving a growing interest by its possibility in quantum computing, low-power CMOS, nanotechnology, and optical computing. It is now widely accepted that the CMOS technology implementing irreversible logic will hit a scaling limit beyond 2016, and thus the increased power dissipation is a major limiting factor.Landauer's principle states that, logic computations that are not reversible generate heat kTln2 for every bits of information that is lost. According to Frank , computers based on reversible logic operations can reuse a fraction of signal energy that theoretically can approach arbitrarily near 100%.

An n-input n-output function (gate) is called reversible if and only if it maps each input instance to a unique output instance. The only possible structure for a reversible network is the cascade of reversible gates. In practice, not all of the n! possible reversible functions can be realized as a single reversible gate. Several reversible gates have been existed in literature so far, where the synthesis of reversible circuits differs significantly from synthesis in traditional irreversible circuits. Two restrictions are added for reversible networks, namely fan-outs and back-feeds.

The aim of the paper is to design a Booth's multiplier in reversible mode which is capable of working with both signed and unsigned numbers. The reversible multiplier designed in works for unsigned numbers only, while the recently developed one in is based on booth recoding. On the other hand, the existed design is dedicated to eliminate these limitations and prove its supremacy thereby. This design also establishes its efficiency by assimilating all the good features of reversible circuits that are characterized by number of garbage outputs and number of gates.

BIT REVERSIBLE BOOTH'S MULTIPLIER

In this section, in a gradual approach we show the design of reversible array multiplier using Booth's algorithm. Implementing the Booth's method by a combinatorial array first requires a reversible multi-function cell capable of addition, subtraction and no operation (or skip), which we call as B cell according to the convention. The various function of B cell is selected by a couple of control lines named as H and D. The control signal is generated by another control cell which is named as C cell.

A. Design of C Cell

The C cell is the basic unit of control circuitry of the original array multiplier. The input of this cell (XiXi−1) implies two adjacent bits of the multiplier operand. The cell generates the required control signal named as H and D [17] according to the original multiplier algorithm. The calculation of H and D are determined by the following equations:

The reversible design of C cell (Fig. 3.1(a)) consists of a 3×3 TS-3 gate, a Fredkin gate. The third input of the Fredkin gate is set to zero, which act as a control input for the gate and generates the product (after complementing the first input) of other two inputs (denoted as 'D'). The third output of the TS-3 Gate is the control signal H. The block diagram of Fig. 3.1(b) shows the input and output line of C cell. The direct quantum realization of the C cell tenders a quantum cost of 7 as the quantum cost of 3×3 TS-3 Gate and Fredkin Gate is

2 and 5, respectively. However, it will produce a quantum cost of 4 if we design according to the one shown in Fig. 3.1(c).

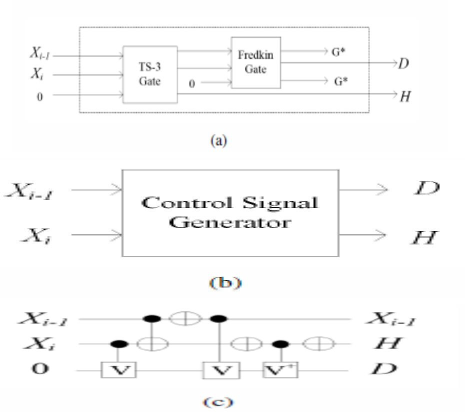


Figure 1. (a) Gates used in ’C’ Cell (b) ’C’ cell as a block diagram with input and output (c) Quantum circuit for ’C’ cell, where QC is 4

B. Design of B Cell

The B cell is a multi-function cell, where various functions include addition, subtraction, no-operation. These functions are defined by the following logic equations:

$$Z = a \oplus bH \oplus cH = a \oplus (b \oplus c)H$$

$$C_{out} = (a \oplus D)(b \oplus c) \oplus bc$$

Here Z is the result of addition or subtraction and Cout indicates the carry output. The cell operates on three operands a, b, c where a is the propagated result from a previous B cell, b is a multiplicand bit and c is the carry-in bit. H and D are the control signal generated by the corresponding C cell.

When HD=10, these equations reduce to the usual full adder equations:

$$Sum = a \oplus b \oplus c$$

$$C_{out} = ab \oplus c(a \oplus b)$$

On the contrary, when HD=11, the equations are converted to the corresponding full-subtracter equations:

$$Sum = a \oplus b \oplus c$$

$$C_{out} = \overline{a}b + \overline{a}c + bc$$

When H=0, Z becomes a and the carry lines play no role in the final result. Table II summarizes the function of this cell.

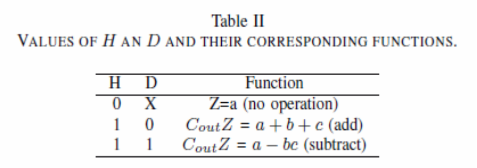


Table II
VALUES OF H AN D AND THEIR CORRESPONDING FUNCTIONS.

H	D	Function
0	X	Z=a (no operation)
1	0	$C_{out}Z = a + b + c$ (add)
1	1	$C_{out}Z = a - bc$ (subtract)

The B cell is designed (shown in Fig. 3.2) with the wellknown TS-3 Gate, MTSG, and Peres Gate. The MTSG Gate is a 4×4 Reversible gate which itself provides a full adder realization when the control bit is zero. Although, the MTSG[15] is a modified version of TSG [3], due to complex input output relationship of TSG, the gate is very much inefficient in terms of quantum realization (e.g., QC(TSG)=13), while the QC of MTSG is less than half of the QC of TSG (e.g.,QC(MTSG)=6). Hence, instead of using TSG gate, we use the modified TSG gate in our design methodology. The MTSG generates very simple output conserving the reversibility property.

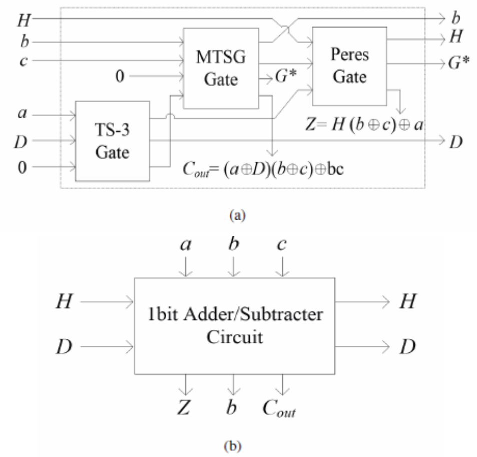

H
b
c
0
MTSG Gate
G*
Peres Gate
b
H
G*
a
D
0
TS-3 Gate
$Z= H(b\oplus c)\oplus a$
D
$C_{out}= (a\oplus D)(b\oplus c)\oplus$bc
(a)
a
b
c
H
D
1bit Adder/Subtracter Circuit
H
D
Z
b
C_{out}
(b)

Figure 2. (a) Circuit diagram of B cell (G* indicates the garbage output) (b) Block diagram of B cell

In addition, providing 0 in the D input, we can easily realize the Full-adder from MTSG. In Fig. 3.2(a), the output of TS-3 gate is fed as input to the MTSG gate and also to the Peres gate. The required output Z is produced from the Peres gate and the carry out bit is produced from the MTSG gate. The control signal HD and the same multiplicand bit b used in this cell is regenerated as a by product to activate the next cell. Since fan out is prohibited in reversible circuit, this additional function is taken into concern of each B cell.

An n × n reversible Booth’s multiplier is realized by the existed B cell and C cell. The architecture of the n×n array multiplier, shown in Fig. 3.3 takes the form of a trapezium. All the C cells at the right together comprise the control circuitry.

If X = Xn,Xn−1,Xn−2 . . . X0 and Y =Yn, Yn−1, Yn−2 . . . Y0 denote the multiplier and multiplicand, respectively then the multiplier bits are fed to the C cells, and a implicit zero is added with the multiplier bits. There are total n rows and each row contains a C cell, hence the total n number of C cells are required in the design. The top most row of this two dimensional architecture contains (2n-1) B cells.

The second row consists of (2n-2) B cells. Continuing in this fashion the bottom line only contains n number of B cell. All the multiplicand bits are fed to the upper layer B cells (through the input line indicated by 'b' in Fig. 3.2). The 'b' inputs of the left side of (n-1) B cells are set to the sign extended Y for addition and subtraction. The a inputs (indicates the result of sum or subtract from the corresponding upper layer cell) of the upper layer B cells and the carry inputs of the rightmost B cells are set to zero.

Multiplication Example by a 4×4 Reversible Booth's Multiplier This section illustrates an example of multiplication by the existed design. It shows the value of each input and output line for every single cell for the particular example. Assume that the two operands are −3 and 5, and so the result should be −15. Obviously the negative input that is the multiplicand will be in twos complement form. Hence, multiplicand Y =1101 (in twos complement form), multiplier X= 0101 (5), an implicit 0 is added, which becomes, X=01010 and they are fed into the C cells in the following manner.

01: HD=10 implies add.
10: HD=01 implies subtract.
01: HD=10 implies add.
10: HD=01 implies subtract.

Thus, the 4×4 circuit (shown in Fig. 3.4) generates 1110001,which is -15 in two's complement.

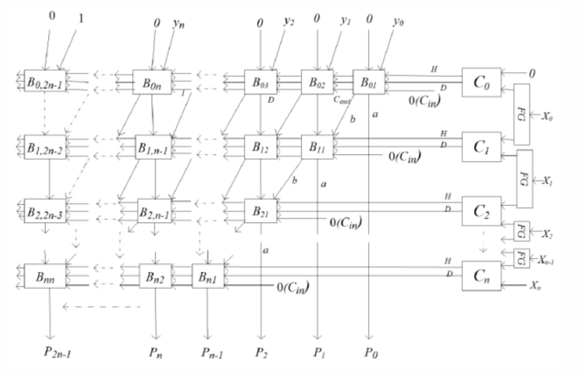


Figure 3. n × n reversible Booth's multiplier

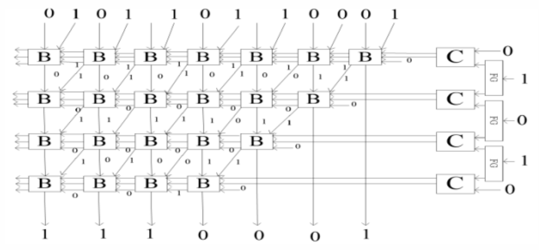


Figure 4. 4×4 reversible Booth's multiplier

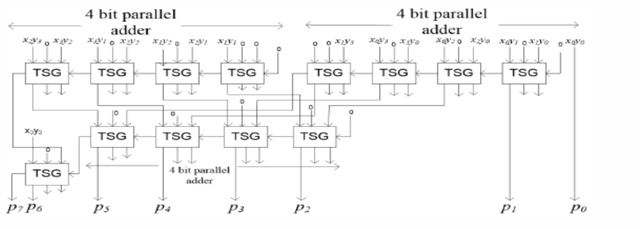


Figure 5. 4 × 4 reversible Booth's multiplier [3].

We also evaluate the 4×4 version of the existed Booth's multiplier with the two existing designs. To compute the necessary parameters for a 4×4 array multiplier the instance of the generalized equations are taken and the calculation is carried out by putting the value of n = 4. Existing method of Bhardaj and Deshpande [4] do not provide any generalized equation to calculate the delay of a circuit, while the other method in [3] (shown in Fig. 6 for n=4) uses fan out which is strongly prohibited in reversible logic design.

On the other hand, the existed circuit is designed avoiding the fan outs. The design of [4] also failed to preserve the constraint of reversible logic design, i.e., loop in circ uit. The existed reversible multiplier works without using feedback and also can operate on both positive and negative numbers whereas the existing reversible multiplier work as serial multiplier. This achievement is obtained in expense of delay and preserving reversibility

II. PROPOSED DESIGN OF 8 BIT BOOTH MULTIPLIER

An 8x8 reversible Booth's multiplier is realized by the existed B cell and C cell.The architecture of the 8x8 array multiplier, shown in Fig. takes the form of a trapezium. All the C cells at the right together comprise the control circuitry. If X = X8,X8−1,X8−2 . . . X0 and Y =Y8, Y8−1, Y8−2 . . . Y0 denote the multiplier and multiplicand, respectively then the multiplier bits are fed to the C cells, and a implicit zero is added with the multiplier bits. There are total 8 rows and each row contains a C cell, hence the total 8 number of C cells are required in the design. The top most row of this two dimensional architecture contains 7 B cells.

The second row consists of 6 B cells. Continuing in this fashion the bottom line only contains n number of B cell. All the multiplicand bits are fed to the upper layer B cells (through the input line indicated by 'b' in Fig.3.6). The 'b' inputs of the left side of 7 B cells are set to the sign extended Y for addition and subtraction. The a inputs (indicates the result of sum or subtract from the corresponding upper layer cell) of the upper layer B cells and the carry inputs of the rightmost B cells are set to zero.

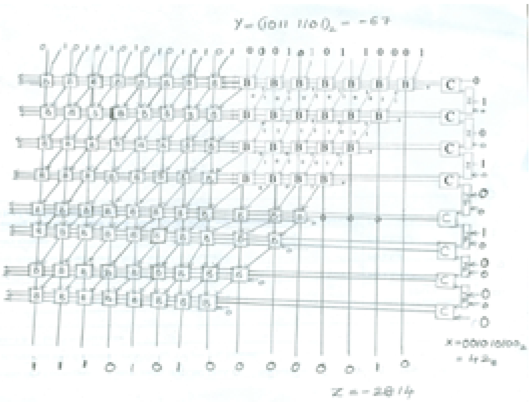


Figure 6. Proposed 8x8 Reversible signed Booth multiplier

Multiplication Example by a 8x8 Reversible Booth's Multiplier This section illustrates an example of multiplication by the proposed design. It shows the value of each input and output line for every single cell for the particular example. Assume that the two operands are −67 and 42, and so the result should be −2814. Obviously the negative input that is the multiplicand will be in twos complement form. Hence, multiplicand Y =1011 1101 (in twos complement form), multiplier X= 010 10100 (42), an implicit 0 is added, which becomes, X=0010 10100 and they are fed into the C cells in the following manner.

01: HD=10 implies add.
10: HD=01 implies subtract.
01: HD=10 implies add.
10: HD=01 implies subtract.

Thus, the 8x8 circuit generates 111010100000010,which is -2814 in two's complement as shown in fig 3.7 below.

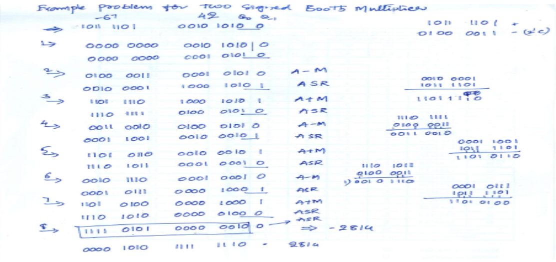

Example Problem for Two Signed Booth Multiplier
-67 42 Q₀ Q₋₁
1011 1101 0010 1010 0
1011 1101 +
0100 0011 - (2'c)
1 0000 0000 0010 1010 0
0000 0000 0001 0101 0
2 0100 0011 0001 0101 0 A−M
0010 0001 1000 1010 1 ASR
3 1101 1110 1000 1010 1 A+M
1110 1111 0100 0101 0 ASR
4 0011 0010 0100 0101 0 A−M
0001 1001 0010 0010 1 ASR
5 1101 0110 0010 0010 1 A+M
1110 1011 0001 0001 0 ASR
6 0010 1110 0001 0001 0 A−M
0001 0111 0000 1000 1 ASR
7 1101 0100 0000 1000 1 A+M
1110 1010 0000 0100 0 ASR
8 1111 0101 0000 0010 0 ASR ⇒ -2814
0000 1010 1111 1110 - 2814
0010 0001
1011 1101
1101 1110
1110 1111
0100 0011
0011 0010
0001 1001
1011 1101
1101 0110
1110 1011
0100 0011
0010 1110
0001 0111
1011 1101
1101 0100

Figure 7. Example for 8x8 reversible signed booth multiplier

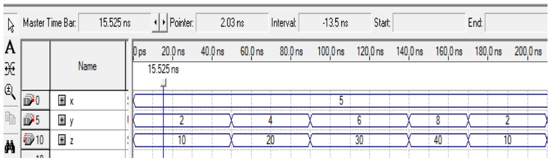
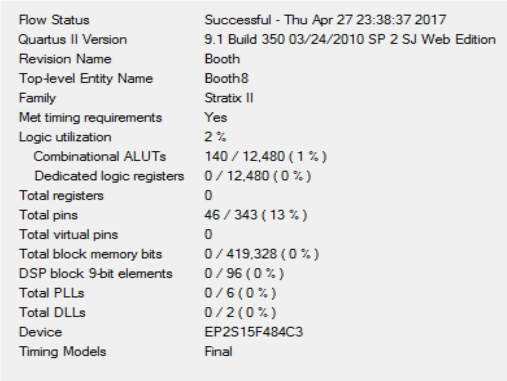


Figure 8. Simulation Results for 4 bit booth multiplier using TSG gate

The above figure is Simulation Results for 4 bit booth multiplier using TSGgate.. WhenX =5(0101) ,Y=2(0010) ,4(0100), 6(0110) and 8(1000) signal are given to 4 bit multiplier using TSG gate the output of multiplier generates 10(0001010),20(0010100),30(0011110) and 40(0101000) respectively.



Flow Status	Successful - Thu Apr 27 23:38:37 2017
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	Booth
Top-level Entity Name	Booth8
Family	Stratix II
Met timing requirements	Yes
Logic utilization	2 %
Combinational ALUTs	140 / 12,480 (1 %)
Dedicated logic registers	0 / 12,480 (0 %)
Total registers	0
Total pins	46 / 343 (13 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP2S15F484C3
Timing Models	Final

Figure 9. bit booth multiplier design summary

The above figure is representation of design summary for 8bit booth multiplier design summary.. The above 4bit booth multiplier takes 140 combinational ALUTs and totally 46 Input and output pins to functioning the 4bit booth multiplier.

The above 4bit booth multiplier takes <1% logic utilization with device used is EP2S15F484C3 and this design has not been used any memory and DSP elements.

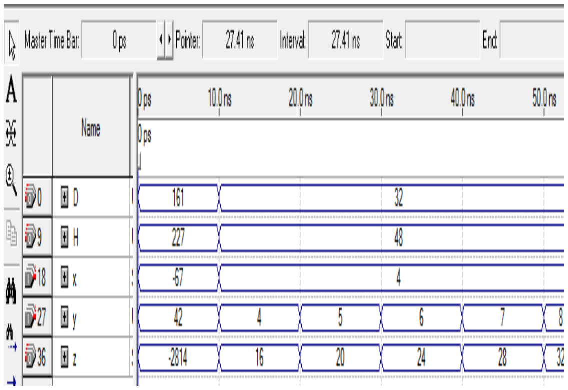
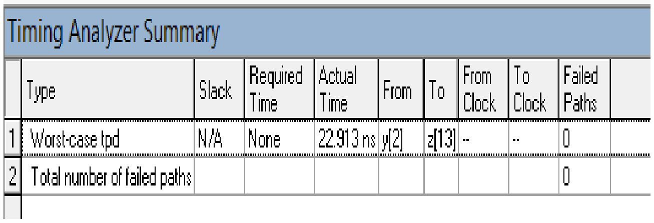


Figure 10. Simulation Results for 8 bit booth multiplier

The above figure is Simulation Results for 8 bit booth multiplier.When X=42(00101010),Y=-67(10111101) signals are given to 8 bit multiplier the output of multiplier generates 2'c form output is -2814(0000 1010 1111 1110).



Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tpd	N/A	None	22.913 ns	y[2]	z[13]	--	--	0
2	Total number of failed paths								0

Figure 11. Timing analysis for 8 bit booth multiplier

III. CONCLUSION

This paper presents a Radix-2 Booth's Multiplier implementation using Reversible Gates. A full design of 4x4 reversible array multiplier is existed which is based on the conventional irreversible design. The evaluation of the proposed circuit is performed from all the aspects of reversible logic. The proposed reversible multiplier architecture outperforms the existing design in terms of design methodology by preserving the constraints of reversible logic synthesis. The key achievement of the proposed design is capable of working with both signed and unsigned numbers, which is not present in the existing circuits considered in this paper. This paper of 8x8 reversible Radix-2 Booth's Multiplier is designed in VHDL and synthesised and simulated in ALTERA QUATRUS –II 9.1.From the results it is evident that the proposed design used 140 Combinational ALUTs and operating with 43.64MHz frequency. Current method is applied to the extension of the existed logic for Radix-4 approach.

REFERENCES

[1] R. Landauer, "Irreversibility and heat generation in the computing process," IBM J. Res. Dev., vol. 5, no. 3, pp. 183–191, Jul. 1961.

[2] M. P. Frank, "Introduction to reversible computing: Motivation, progress,and challenges," in Conference on Computing Frontiers, 2005, pp. 385–390.

[3] H. Thapliyal and M. B. Srinivas, "Novel reversible multiplier architecture using reversible tsg gate," in IEEE Conference on Computer Systems and Applications, 2006, pp. 100–103.

[4] K. Bhardwaj and B. M. Deshpande, "K-algorithm: An improved booth's recoding for optimal fault-tolerant reversible multiplier," in VLSI Design, 2013, pp. 362–367.

[5] H. H. Babu, R. Islam, A. R. Chowdhury, and S. M. A. Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit," in Euromicro Symposium on Digital Systems Design, 2003, pp. 50–54.

[6] H. H. Babu, R. Islam, S. M. A. Chowdhury, and A. R. Chowdhury, "Synthesis of full-adder circuit using reversible logic," in VLSI Design, 2004, pp. 757–760.

[7] A. R. Chowdhury, R. Nazmul, and H. M. H. Babu, "A new approach to synthesize multiple-output functions using reversible programmable logic array," in International Conference on VLSI Design (VLSID), 2006, pp. 6–11.

[8] S. Mitra and A. Chowdhury, "Minimum cost fault tolerant adder circuits in reversible logic synthesis," in International Conference on VLSI Design (VLSID), 2012.

[9] R. P. Feynman, "Quantum mechanical computers," Optics News, vol. 11,no. 2, pp. 11–20, Feb 1985.

[10] T. Toffoli, "Reversible computing," in Colloquium on Automata, Languages and Programming, 1980, pp. 632–644.

[11] E. Fredkin and T. Toffoli, "Collision-based computing," vol. 21, pp.219–253, 1983.

[12] A. Peres, "Reversible logic and quantum computers," Phys. Rev. A,vol. 32, pp. 3266–3276, Dec 1985.

[13] H. Thapliyal, S. Kotiyal, and M. B. Srinivas, "Novel bcd adders and their reversible logic implementation for ieee 754r format," in VLSI Design, 2006, pp. 387–392.

[14] H. Thapliyal and M. B. Srinivas, "A novel reversible tsg gate and its application for designing reversible carry look-ahead and other adder architectures," in Asia-Pacific Conference on Advances in Computer Systems Architecture, 2005, pp. 805–817.

[15] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu, "Efficient approaches for designing reversible binary coded decimal adders," Microelectronics Journal, vol. 39, no. 12, pp. 1693–1703, 2008.

[16] J. P. Hayes, "Computer organization and arc

[17] Jakia Sultana, Sajib Kumar Mitra and Ahsan Raja Chowdhury “On the Analysis of Reversible Booth’s Multiplier” in 28th International Conference on VLSI Design and 2015 14th International Conference on Embedded Systems,2015