

Power Optimization using Reversible Gates for Booth's Multiplier

K. Veerender¹ | Ms. G.Laxmi²

¹PG Scholar, Vaagdevi Engineering College, Telangana.

²Assistant Professor, Vaagdevi Engineering College, Telangana.

To Cite this Article

K. Veerender, Ms. G.Laxmi, "Power Optimization using Reversible Gates for Booth's Multiplier", *International Journal for Modern Trends in Science and Technology*, Vol. 02, Issue 11, 2016, pp. 102-107.

ABSTRACT

Reversible logic attains the attraction of researchers in the last decade mainly due to low-power dissipation. Designers' endeavours are thus continuing in creating complete reversible circuits consisting of reversible gates. This paper presents a design methodology for the realization of Booth's multiplier in reversible mode. So that power is optimised Booth's multiplier is considered as one of the fastest multipliers in literature and we 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. Theoretical underpinnings, established for the proposed design, show that the proposed circuit is very efficient from reversible circuit design point of view.

KEYWORDS: Booth's Multiplier, Garbage Output, Low power Design, Quantum Cost.

Copyright © 2016 International Journal for Modern Trends in Science and Technology
All rights reserved.

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 [1] states that, logic computations that are not reversible generate heat $kT \ln 2$ for every bits of information that is lost. According to Frank [2], computers based on reversible logic operations can reuse a fraction of signal energy that theoretically can approach arbitrarily near 100%.

Rest of the paper is organized as follows: After illustrating the preliminaries of reversible logic gates in Section 2, we have presented the

input-output vectors of popular reversible gates along with their quantum costs. Section 3 concentrates on the main logic synthesis of the proposed reversible multiplier with the conclude in Section 5 discussing the main contribution and the future work.

II. LITERATURE REVIEW

In this section, basic definitions and ideas related to reversible logic are presented.

a) *Definition 1:* A Reversible Gate is a k -input, k -output (denoted by $k \times k$) circuit that produces a unique output pattern [5]–[8] for each possible input pattern. Reversible Gates are circuits in which the number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs, i.e., it can generate unique output vector

from each input vector and vice versa. always a good complexity measure for the circuit. It is always desirable to realize a circuit with minimum number of gates.

Let the input vector be I_v , output vector O_v and they are defined as follows, $I_v = (I_1, I_2, \dots, I_k)$ and $O_v = (O_1, O_2, \dots, O_k)$. For each particular k , there exists the relationship $I_v \leftrightarrow O_v$.

b) *Definition 2:* Unwanted or unused output of a re-versible gate (or circuit) is known as Garbage Output. More formally, the outputs which are needed only to maintain reversibility are called garbage outputs. While performing EX-OR operation with a Feynman gate (defined in Table I), the second output should be called as garbage, as shown in Fig. 1 with *.

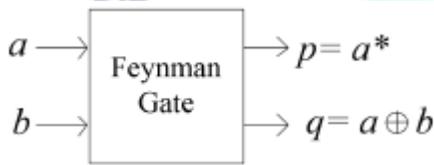


Figure 1. Feynman gate with target and garbage output.

Table I
POPULAR REVERSIBLE GATES WITH THEIR INPUT-OUTPUTS AND QUANTUM COST (QC)

Gate Name	Input Vector	Output Vector	Quantum Cost (QC)
Feynman (FG) [9]	(a, b)	(a, a ⊕ b)	1
Toffoli (TG) [10]	(a, b, c)	(a, b, ab ⊕ c)	5
Fredkin (FRG) [11]	(a, b, c)	(a, a' b ⊕ ac, a' c ⊕ ab)	5
Peres (PG) [12]	(a, b, c)	(a, a ⊕ b, ab ⊕ c)	4
TS-3 [13]	(a, b, c)	(a, b, a ⊕ b ⊕ c)	2
TSG [14]	(a, b, c, d)	(a, a' c' ⊕ b', a' c' ⊕ b' ⊕ d, (a' c' ⊕ b') d ⊕ ab ⊕ c)	13
MTSG [15]	(a, b, c, d)	(a, a ⊕ b, a ⊕ b ⊕ c, (a ⊕ b) c ⊕ ab ⊕ d)	6

c) *Definition 3:* The delay of a logic circuit is the maximum number of gates in a path from any input line to any output line. The delay of the circuit in Fig. 1 is obviously 1 as it is the only gate in any path from input to output.

d) *Definition 4:* The quantum cost (QC) of every 2×2 gate is the equal (=1) [16], while a 1×1 gate costs nothing since it can be always included to arbitrary 2×2 gate that precedes or follows it. Thus, in first approximation, every permutation of quantum gate will be built from 1×1 and 2×2 quantum primitives and its cost is calculated as the total sum of 2×2 gates that are used in the circuit. Now we define some popular reversible gates in Table I with their corresponding input-output vectors and quantum cost.

III. PROPOSED 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 ($X_i X_{i-1}$) 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:

$$H = X_i \oplus X_{i-1} \text{ and } D = X_i X_{i-1} \quad (1)$$

The reversible design of *C* cell (Fig. 2(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. 2(b) shows the input and output line of *C* cell. The direct quantum realization of the *C* cell tends 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. 2(c).

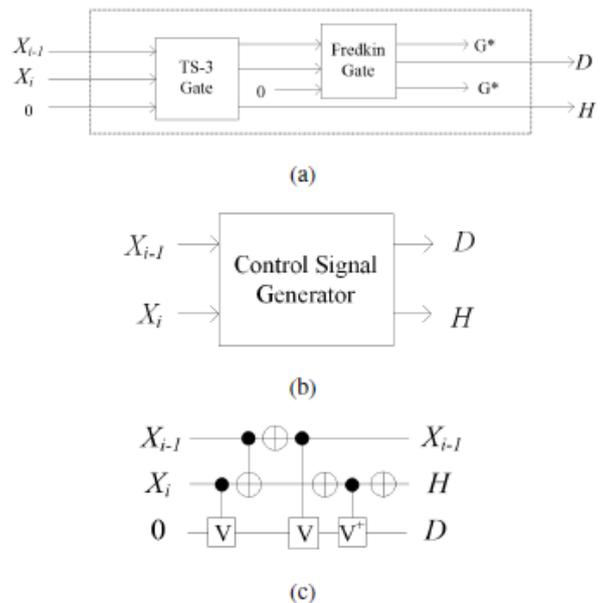


Figure 2. (a) Gates used in 'C' Cell (b) 'C' cell as a block diagram with input and output (c) Quantum circuit for 'C' cell, where QC is 4

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:

$$Z = a \oplus bH \oplus cH = a \oplus (b \oplus c)H \quad C_{out} = (a \oplus D)(b \oplus c) \oplus bc \quad (2)$$

Here Z is the result of addition or subtraction and C_{out} 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:

$$\begin{aligned} Sum &= a \oplus b \oplus c \\ C_{out} &= ab \oplus c(a \oplus b) \end{aligned} \quad (3)$$

Table II

VALUES OF H AND 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)

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

$$\begin{aligned} Sum &= a \oplus b \oplus c \\ C_{out} &= ab + ac + bc \end{aligned} \quad (4)$$

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.

The B cell is designed (shown in Fig. 3) with the well known TS-3 Gate, MTSG, and Peres Gate. The MTSG Gate is a 4x4 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. In addition, providing 0 in the D input, we can easily realize the Full-adder from MTSG. In Fig. 3(a), the output

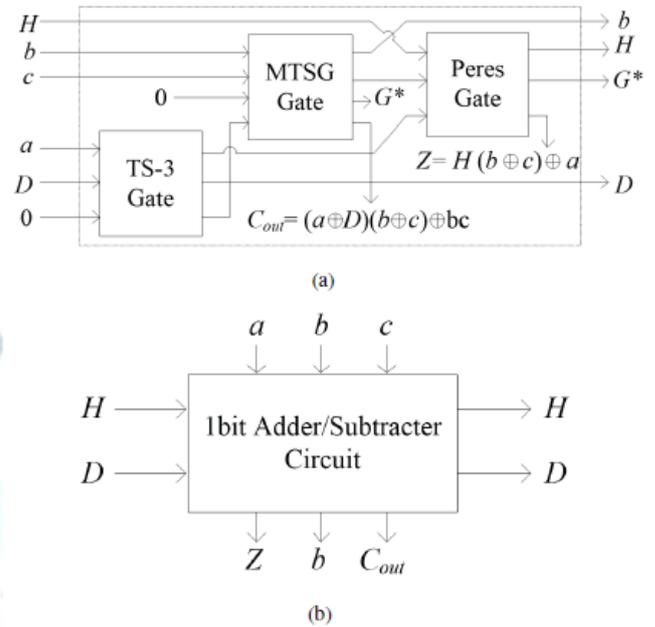


Figure 3. (a) Circuit diagram of B cell (G^* indicates the garbage output) (b) Block diagram of B cell

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 byproduct to activate the next cell. Since fan out is prohibited in reversible circuit, this additional function is taken into concern of each B cell the quantum cost of the gate 'X'. To build up the generalized equation for the n bit version, the basic cells work as a building block. Thus, we summarize the number of gates, garbage outputs and the quantum cost of these cells in Table III.

Table III
COST FACTOR ANALYSIS OF THE PROPOSED CELLS. * ALTHOUGH $QC(TS-3) + QC(FRG) = 7$, ACCORDING TO THE DESIGN OF FIG. 2, THE $QC(C \text{ CELL})$ IS 4

Cell	Gate (count)	Garbage cost	Quantum Cost
B cell	TS-3 (1)	2	12
	MTSG (1)		
	Peres (1)		
C cell	TS-3 (1) Fredkin(1)	2	4*

C. Construction of $n \times n$ Reversible Twos Complement Array Multiplier

In this section, an $n \times n$ reversible Booth's multiplier is realized by the proposed B cell and C cell. The architecture of the $n \times n$ array multiplier, shown in Fig. 4 takes the form of a trapezium. All the C cells at the right together comprise the control circuitry. If $X = X_n, X_{n-1}, X_{n-2} \dots X_0$ and $Y = Y_n, Y_{n-1}, Y_{n-2} \dots Y_0$ denote the multiplier and multiplicand, respectively then the multiplier bits

are fed to the C cells, and an 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). 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.

D. Multiplication Example by a 4x4 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 -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 4x4 circuit (shown in Fig. 5) generates 1110001, which is -15 in two's complement.

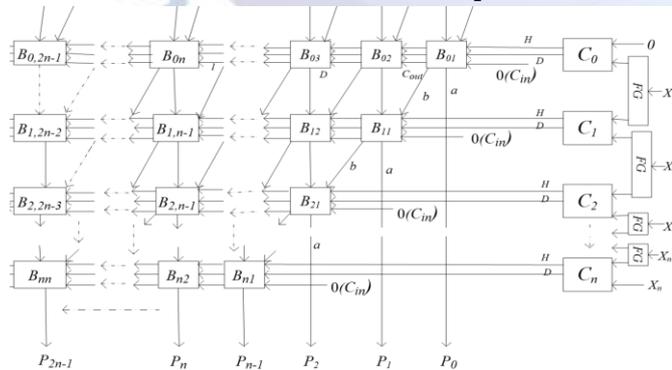


Figure 4. $n \times n$ reversible Booth's multiplier.

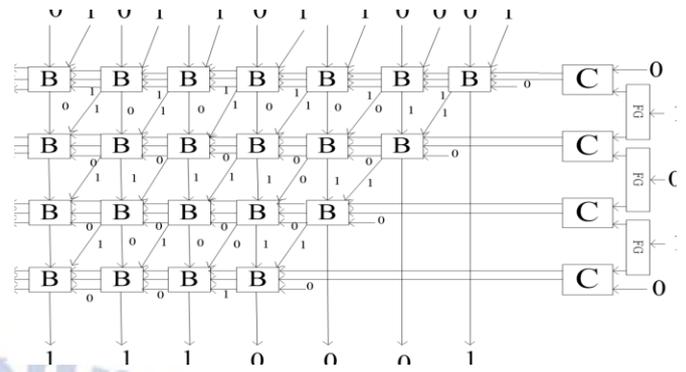


Figure 5. 4x4 reversible Booth's multiplier.

IV. EVALUATION OF THE PROPOSED DESIGN

In this section necessary theorems are given to evaluate the proposed design. All the theorems provide lower bounds for number of gates, garbage outputs, circuit delay and quantum cost.

Theorem 1: Let N_{GT} be the number of gates required to realize an $n \times n$ Reversible Multiplier where n is the number of bits, then

$$N_{GT} \geq \frac{9}{2}n^2 + 1 \tag{5}$$

Proof: An $n \times n$ Reversible Multiplier requires $3/2 n(n-1)B$ cells and each B cell contains 3 reversible gates. Moreover, n LB cells (B cells in left most side) are required along with the B cell each of which contains 2 reversible gates. To perform the control operation n C cells are required, where each of them consists of 2 gates. Furthermore, $(n/2+1)$ FGs are required to perform the copy operation. As N_{GT} is the total number of gates to realize the $n \times n$ Multiplier, according to the above definition

$$N_{GT} \geq \frac{3}{2}n(n-1)3 + 2n + 2n + \frac{n}{2} + 1$$

$$N_{GT} \geq \frac{9}{2}n^2 + 1 \tag{6}$$

Similarly, we propose the following theorems that can be proved in the similar way.

Theorem 2: Let n be the number of bits in the Reversible Multiplier and N_{GB} denotes the number of garbage outputs, then

$$N_{GB} \geq n(4n + 1) - 1$$

Proof: Each B cell generates 2 garbage output and an $n \times n$ Reversible Multiplier requires $3/2 n(n-1)$ B cells. The LB cell, comprises the last column of the reversible multiplier do not need to generate the carry equation as well as to propagate the control signal. Realization of a LB cell requires

no less than 3 garbage output. Further, each C cell generates 2 garbage output and Total $2n$ number of garbage is added for n C cells. Moreover, $(n-1)$ B cells of the last row generates extra $(n-1)$ garbage that is due to the propagation of prime input b. As NGB is considered as the minimum number of gates to realize the reversible multiplier, hence

$$N_{GB} \geq \frac{3}{2} n(n-1) + 3n + 2n + (n-1) \quad (7)$$

$$N_{GB} \geq n(4n+1) - 1$$

Theorem 3: Let P , P and P are the delay of B, LB and C cell respectively in the $n \times n$ reversible multiplier. Let, D_F be the delay of a FG and D_{RM} denotes the total delay of the reversible multiplier, then

$$D_{RM} \geq (2n-2)P + nP + P + D_F$$

LB cell, one C cell and a FG along with the B cell. Hence, considering D_{RM} as the total delay,

Theorem 4: Let QC(RM) be the total quantum cost to realize an $n \times n$ reversible multiplier where n is the number of bits, then

$$QC(RM) \geq 18n^2 + 13\frac{n}{2} + 1 \quad (9)$$

Proof: Each B cell tenders a quantum cost of 12 ($QC(TS-3) + QC(MTSG) + QC(PG) = 2+6+4$). An $n \times n$ reversible multiplier requires $n/2(3n-5)$ such B cells. The B cell of the column before the last one uses FG instead of TS-3 since it does not need to feed control signal D to the last level of LB cell. Thus, each B cell of this specific column incurs a quantum cost of 11. Moreover the quantum cost of each individual LB cell is 5 ($QC(FG) + QC(PG) = 1+4$). Beside this, the n number of C cells contribute $7n$ to the quantum cost and the remaining $(n/2+1)$ FGs are responsible for a QC of $(n/2+1)$. As QC(RM) is the total quantum cost to realize the $n \times n$ multiplier, according to the aforementioned definition:

$$QC(RM) \geq n_2(3n-5)12 + 11n + 5n + 7n + n_2 + 1$$

$$\geq 18n^2 + 13n_2 + 1$$

We also evaluate the 4×4 version of the proposed 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 proposed 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 circuit. The proposed 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.

V. CONCLUSION

This paper presents a Radix-2 Booth's Multiplier implementation using Reversible Gates. A full design of $n \times n$ reversible array multiplier is proposed which is based on the conventional irreversible design. The evaluation of the proposed circuit is performed from all the aspects of reversible logic. Additionally, the quantum cost of the proposed cell (different sub-sections of the entire circuit) as well as the whole design has been analyzed. 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 design is, it is capable which is not present in the existing circuits considered in this paper. Current research is investigating the extension of the proposed 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] M. Perkowski, M. Lukac, P. Kerntopf, M. Pivtoraiko, D. Lee, H. Kim, W. Hwangbo, J.-w. Kim, and Y. W. Choi, “A hierarchical approach to computer-aided design of quantum circuits,” in *International Symposium on Representations and Methodology of Future Computing Technology*, 2002, pp. 201–209.
- [17] J. P. Hayes, “Computer organization and architecture,” 1978.