International Journal of Scientific Research in Science, Engineering and Technology
© 2019 IJSRSET | Volume 6 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
doi : https://doi.org/10.32628/IJSRSET

Reversible Booth's Multiplication Implementation using Reversible Gates

Anup Kumar Biswas
Assistant Professor, Computer Science and Engineering, Kalyani Govt. Engineering College, Kalyani, Nadia, West Bengal, India

ABSTRACT

One of the attractive research area in Engineering and Technology is Reversible Computing / Reversible logic. The attraction of scientists and researchers in this field in the last decade is strongly due to low-power consumption for the execution of operations. To minimize the width of circuit, garbage, total gate numbers and delay are the objectives of the reversible logic synthesis. This paper explains the design methodology regarding reversible Booth's multiplier to be realized. Researchers' attempts are thus making complete reversible logic circuits made up of reversible gates. Reversible Booth's multiplication process as considered to be a speediest multiplier method. Being inspired in it, an efficient design-methodology for reversible paradigm is shown. The proposed architecture for reversible multiplication is capable of performing over signed and unsigned number multiplications for two input numbers without keeping any feedbacks. But the existing reversible mode of multiplication thinks of the feedback loop that is strongly prohibited in our present reversible multiplication mode. Theoretical underpinnings are able to show the efficiency of a reversible logic calculation and accordingly it is observed that our proposed circuit is more efficient with respect to the design viewpoint of a reversible circuit.

Keywords : Quantum Cost, Reversible Logic Booth's Multiplier, Permutation Matrix, Garbage Value

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 strongly taken that CMOS technology used for irreversible logic will create hit to a scaling limit beyond 2016, and this positive power consumption/dissipation is a considerable limiting factor. Landauer's principle [1] says that the logic computations, not reversible create heat $kT(\ln 2)$(where k = Plank′s constant) for any bit of information that is lost. In idle case, Frank [2] says that computers computations on reversible logic operations are able to reuse a fractional part of signal power that can approach close to 100%.

A gate which has k-number inputs and k-number of outputs is said to be reversible iff it maps each vector input to a unique vector output. The structure of a reversible network lies in the cascade or in sequence of reversible gates. In practice, not all of the n! It is important that the reversible functions can be considered as a reversible gate. There are two restrictions joined in reversible networks, they are back-feeds and fan-outs. There is a reversible multiplier designed in [3] that acts only on the unsigned numbers only, whereas, the recently developed reversible multiplier in [4] is on the basis of Booth recoding. On the other hand, the proposed

Received : 05 March 2019 | Accepted : 25 March 2019 | Volume 6, Issue 2, March-April-2019 : 849-859]

design is dedicated to eliminate these limitations and to prove its supremacy thereby. This design also develops its efficiency by means of assimilating all the fine characteristics of reversible circuits characterized by the positive number of garbage outputs as well as number of gates.

Rest of the paper is organized as follows: After illustrating the preliminaries of reversible logic gates in Sections 2 and 3 we have presented the input-output vectors of popular reversible gates along with their quantum costs. Section 5 concentrates on the main logic synthesis of the proposed reversible multiplier with the detailed description of each designed blocks. The theoretical underpinnings and the evaluation of the proposed Booth's multiplier are shown in Section 6. Some theorems with respect to the n × n Reversible Multiplier are given in section 8. We conclude in Section 9 discussing the main contribution and the future work.

II. LITERATURE REVIEW

Here, ideas and definitions relating to reversible logic are cited.

1) Related works in reversible gates

Reversible gates or reversible logic gates are the gates with a property of equal number of inputs and outputs, i.e. n-inputs and n-outputs. This reversible gates will minimize power consumption during the executions or computations provided the number of inputs equal to the number of outputs. This is possible because the outputs can be determined using the inputs and on the other hand equally the inputs can be recovered from the output uniquely [13,14,15]. Some works already have been done regarding reversible logics but this paper work is the first attempt in constructing it a structured type from the very fundamental logic gates [16]. This method of designing from logic gates improves the designing strategy from basic level which will have an effect on further higher level circuits and modules thus designed [17]. Hence, here an attempt is made to design reversible circuits from basic level to advanced level, by first designing logic gates and then using them to design higher level modules. This kind of structured approach is not carried out yet, to the best of the author's knowledge [18].

1.1.1. Controlled NOT gate [CNOT gate]

This gate is also called as Feynman gate, which gives the XOR operation of the inputs, as shown in Fig 1. Its quantum cost is 1. CNOT is widely used for fan-out purposes. Here P=A; Q= A ⊕ B.

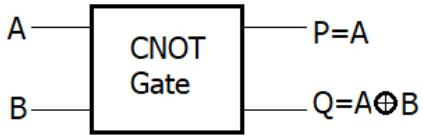


Fig. 1. CNOT gate.

1.1.2. Peres Gate

The new logic gate named "Peres gate" is a reversible gates having 3 inputs (A, B, C) and 3 outputs (P,Q,R) in Fig. 2 bearing quantum cost of 4. In the Figure, input-output relationships are: R = (A **AND** B) ⊕ C, Q = A ⊕ B and P = A [13].

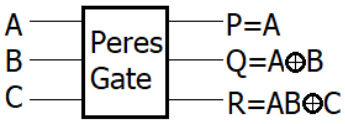


Fig. 2. Peres Gate

1.1.3. URG Gate (Universal Reversible logic Gate)

Like the ". Peres Gate" URG has the same number of input and outputs i.e, 3 inputs (A, B, C) and 3 outputs (P,Q,R), but quantum cost of 6 which is not the same as 4(for Peres Gate). The input-output relationships

are: R = (A.B) ⊕ C, Q = B and P = (A+B) ⊕ C. This relationship is depicted in Fig. 3 below.

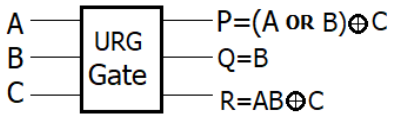


Fig. 3 URG Gate

2). Reversible gates

Reversible gates are being the system of circuits which perform one-to-one mapping betwixt vectors of inputs and outputs; as the mapping capability they have, the input vector states can always be retrieved from the output vector states if they got lost. Research performed by Landauer[1] and Bennett proved that the energy dissipation will not happen if computation is made reversible. With a view to obtain a number of reversible gates, many of such gates are designed and invented. Some examples concerning them are – (i) the Feynman gate, (ii) the Fredkin gate, (iii) the Toffoli gate, and (iv) the Peres gate. Extensive applications involved in Reversible logic is considered to be one of futuristic technologies. We are aware of the logic circuit constructed for a special purpose on the basis of logic gates, which are not reversible but non-reversible. In this paper we have designed logic gate by using reversible gates. Such reversible gates will give us assistance for implementing logic circuits having higher end (of degree).

Suppose the input vector is V_i, output vector V_o and they are defined like this, $V_i = (V_{i1}, V_{i2}, \ldots, V_{in})$ and $V_o = (V_{o1}, V_{o2}, \ldots, V_{on})$, then the relationship for a reversible gate gives $V_i \rightleftharpoons V_o$.

a) Feynman gate

In a reversible gate, some unwanted/unused outputs are known to be Garbage outputs. It can also be defined or treated as Garbage outputs. In the Feynman gate **(CNOT gate),** while it performs the exclusive-OR operation for one output, the other output is called the garbage depicted in Fig. 4(a).

This garbage (P=A) is essential for the purpose of controlling the reversibility of the output. The gate bearing garbage has its quantum cost equal to 1. CNOT can be used for fan-out purposes. The input-output relationships are P=A and Q= A⊕B. The Permutation Matrix for this gate is given in Fig. 4(b).

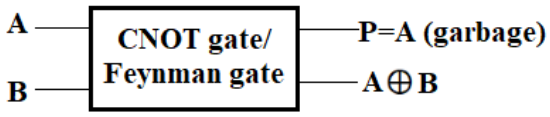


Fig. 4(a) Feynman gate (CNOT gate)

Truth table

INPUT		OUTPUT	
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Permutation matrix form

x1 y1 \ xo yo	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

Fig. 4(b) Truth Table and Permutation Matrix

b) Delay

The executing/quantum delay executing/quantum delay regarding to a logic circuit can be defined as sum of the executing/quantum delays of all gates connected sequentially in a path starting from an input point to a desired output terminal point. The executing/quantum delay of the circuit given in Fig. 4(a) is 1 because of its having only a gate in any path getting started from the input to the output terminal(s).

c) Quantum Cost (QC)

For a 2× 2 (2-inputs, 2-outputs) gate the quantum cost (QC) equal to (=1) [16], whereas a 1×1 gate costs nothing because this gate can always be included in arbitrary 2×2 gate and this gate can follow or precede it. Each permutation quantum gate can be constructed from 1×1 and 2×2 quantum primitives and its

quantum cost is measured as the sum total of 2×2 gates being used in the circuit.

Given below some popular reversible gates in Table 1 along with their corresponding quantum cost and input-output vectors.

Table-1 Some popular reversible gates with their input-output relations and quantum costs

Sl. No.	Gate Name	Input vector	Output vector	Quantum Cost
i.	Feynman(FG)[9]	(A, B)	(A, A⊕B)	1
ii.	Toffoli (TG)[10]	(A,B,C)	(A,B,AB⊕C)	5
iii.	Peres (PG) [12]	(A, B, C)	(A, A⊕B, AB⊕C)	4
iv.	Fredkin (FRG) [11]	(A, B, C)	(A, $\bar{A}$ B⊕AC, $\bar{A}$ C⊕AB)	5
v.	TS-3 [13]	(A, B, C)	(A, B, A⊕B⊕C)	2
vi.	TSG [14]	(A,B,C,D)	(A, $\bar{A}$ $\bar{C}$⊕$\bar{B}$, $\bar{A}$ $\bar{C}$⊕$\bar{B}$⊕D, ($\bar{A}\bar{C}$⊕$\bar{B}$)D⊕AB⊕C)	13
vii.	MTSG [15]	(A, B, C, D)	(A, A⊕B, A⊕B⊕C, (A⊕B)C⊕AB⊕D)	6
viii.	URG	(A, B, C)	((A+B)⊕C, B, (AB)⊕C)	6

3). Implementation of array for booth Multiplication

Booth multiplication method implemented by a combinational array attracts a multifunctional cell called "Main Cell" which is capable of doing subtraction, addition and no operation or skip. Such a multifunctional "Main Cell" is depicted in Fig. 6(a) showing the different inputs (a, b, c, D, H) and outputs (z, b, C_{out}, D, H). The relationships between inputs and outputs can be established with the help of the following equations:

z=a⊕(b⊕c)H…….... (1)

C_{out}= (a⊕D) (b+c) + bc…… (2)

From the above equations, if the values of H and D are assumed to be 1 and 0 respectively, then these two equations reduce to full-adder equations like

z=a⊕b⊕c……..…….. (3)

C_{out}=a (b+c) + bc…… (4)

—where c and C_{out} do the roles of carry-in and carry-out.

If the values of H and D are assumed to be 1 and 1 respectively, then these two equations reduce to full-subtractor equations like

z=a⊕b⊕c……..…….. (5)

C_{out}=$\bar{a}$ (b+c) + bc…… (6)

here c and C_{out} do the roles of borrow-in and borrow-out

When the values of H and D are assumed to be 0 and 0 respectively, the equations reduce to (7) and (8), where z has no change, i.e., it takes the previous value. C_{out} indicates the carry-out.

z=a……..…….. (7)

C_{out}=a (b+c) + bc…… (8)

When the values of H and D are assumed to be 0 and 0 respectively, the equations reduce to (9) and (10), where z has no change, i.e., it takes the previous value. C_{out} indicates the borrow-out.

z=a……..…….. (9)

C_{out}=$\bar{a}$ (b+c) + bc…… (10)

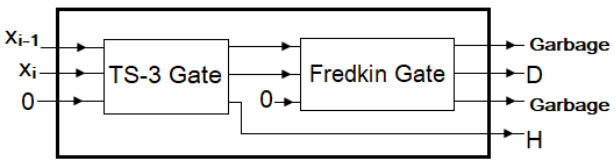


Fig. 5(a) HD Cell

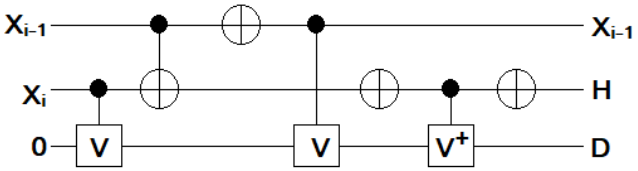


Fig. 5(b) Quantum Cell of HD Cell with quantum cost 3

Noted that the operations performed by the row i of "main Cells" [in Fig. 6(a)] are concluded by the values of $x_i x_{i-1}$ of the multiplier X. We can use every possible pair x_i, x_{i-1} for the purpose of controlling the row operations, by introducing another Cell indicated HD shown in Fig. 5(b). This HD Cell produces the controlling input signals H and D which are essential for the "Main Cells". The HD Cell must compare the values of x_i *and* x_{i-1} and deliver the decision value of HD inevitable for the Fig. 5(a).The values of H and D are produced by the following two equations.

$$H=(x_i \; XOR \; x_{i-1}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (11)$$

$$D=(x_i \; AND \; \bar{x}_{i-1}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (12)$$

5. Two Cells --"Main Cell" , "HD Cell"

For the purpose of designing of the proposed reversible array multiplier [KOREN 93], we must arrange combinational array multiplier which requires the reversible multifunctional cell (called "Main Cell") performing addition, subtraction and no operation (skip). The different functions performed by this Main Cell are done with the two control signals designated as H and D. These two control signals H and D are generated by another Cell called "HD Cell" The two inputs (x_i *and* x_{i-1}) of this HD Cell are taken from the Multiplicand (or Multiplier). Note that $x_{-1}$is always zero, i.e. $x_{-1} = 0$.

Both the "Main Cell" and "HD Cell" can be implemented by the popular reversible gates like TS-3 Gate, MTSG and Peres Gate or URG (Universal Reversible gate).The MTSG Gate is a Reversible gate having 4 inputs and four outputs which provides a full-adder [15].This MTSG is really a modified version of TSG [3], as the complex input-output relationship regarding TSG, this gate is wanting some inefficiency in terms of quantum costs. As the quantum cost of TSG is equal to13, whereas the QC of MTSG is 6. Therefore not accepting the TSG gate, we must prefer the modified TSG gate for the design methodology. The MTSG reversible gate creates very simple output maintaining the reversibility property. If D=0 and H=1, we can easily get a Full-adder from MTSG gate. In the circuit Fig. 6(a), we have drawn a "Main Cell" consisting of (i) one TS-3 gate, (ii) one MTSG gate and (iii) one Peres gate, one output is of the TS-3 is fed into the MTSG and another is also fed into the Peres gate. Similarly a single output of MTSG gate is fed into the Peres

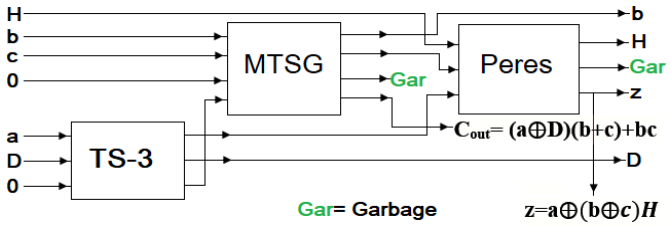


Fig. 6 (a) Main Cell

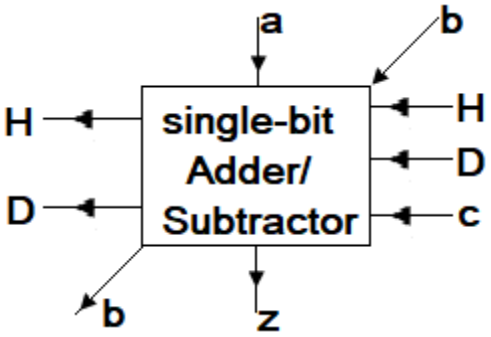


Fig. 6(b) Block diagram of Fig. 6(a)

The single-bit addition or subtraction results z; and carry-out or borrow-out C_{out} are produced fromTS-3 and MTSG gates respectively. Two control signals (H, D) and a one-bit multiplicand b are fed into the "Main Cell" and they are bypassed safely for activating the next Cells. To find the quantum cost of the main Cell we have to add all the quantum costs of TS-3 Gate, MTSG Gate and Peres Gate. As the quantum costs of them are 2, 6 and 4 respectively, hence the quantum cost of Fig. 6(a) is 12. When we are intended to making a generalized circuit diagram, the "Main cell" plays the role of a building block. Now we can summarize the number of gates requited, garbage

outputs and the individual quantum cost regarding the cells in Table-2 below. Note that although the sum of the quantum cost of TS-3 and FRG is 2+5=7, and the quantum cost for the circuit HD drawn in Fig. 5(a) is 4.

Table-2

Sl. No.	Cell Name	Gate Required	Garbage Cost	Quantum Cost
1	Main Cell	MTSG(one) Peres(one) TS-3(one)	2	12
2	HD Cell		2	4

6. Reversible Two's Complement Combinational Array multiplication Construction

An n×n reversible Booth's Multiplication can be realized with the help of (i) "Main Cell" and (ii) "HD Call". The architecture of this multiplier is depicted in Fig. 7(a). The extra Main cells set at the left alter the shape of the array from a parallelogram to a trapezium. They are employed to sign-extend the multiplicand Y= $y_n, y_{n-1}, y_{n-2}, \ldots, y_0$ for the purpose of addition/subtraction. Noted that, the diagonal line identified by b in Fig. 7(b) provide the sign-extended Y at each row of "Main Cell". If Y is positive, then it is sign-extended by leading 0s. On the contrary, if Y is negative, it is sign-extended by the leading 1s. As there are n numbers of row of "Main Cells" in Fig. 7(a), there are n number of "HD Cells". In the first row(upper row), the figure contains n+(n-1)=2n-1 "Main Cells", in the second it is (2n-2), in the 3rd it is (2n-3)…,in the last row it is n. Total number of "Main Cells" in Fig. 7(a) is

$= \{n+ (n-1)\} + \{n+ (n-2)\} +\ldots+(n+1)+n$

$= n^2+ \frac{n(n-1)}{2}$.

When n=4, total "Main Cells" is equal to 22.

All the bits of Y= $y_n, y_{n-1}, y_{n-2}, \ldots, y_0$ are fed to the upper-row "Main Cells", i.e. at the line marked b in the Fig. 6(b). Values of "a" are set to 0 initially for the first row. Carry in for each rightmost "Main Cells" is set to 0. The input-output relationships for the case of "Main Cell" and "HD Cell" is presented in Fig. 7(b) and Fig. 7(c).

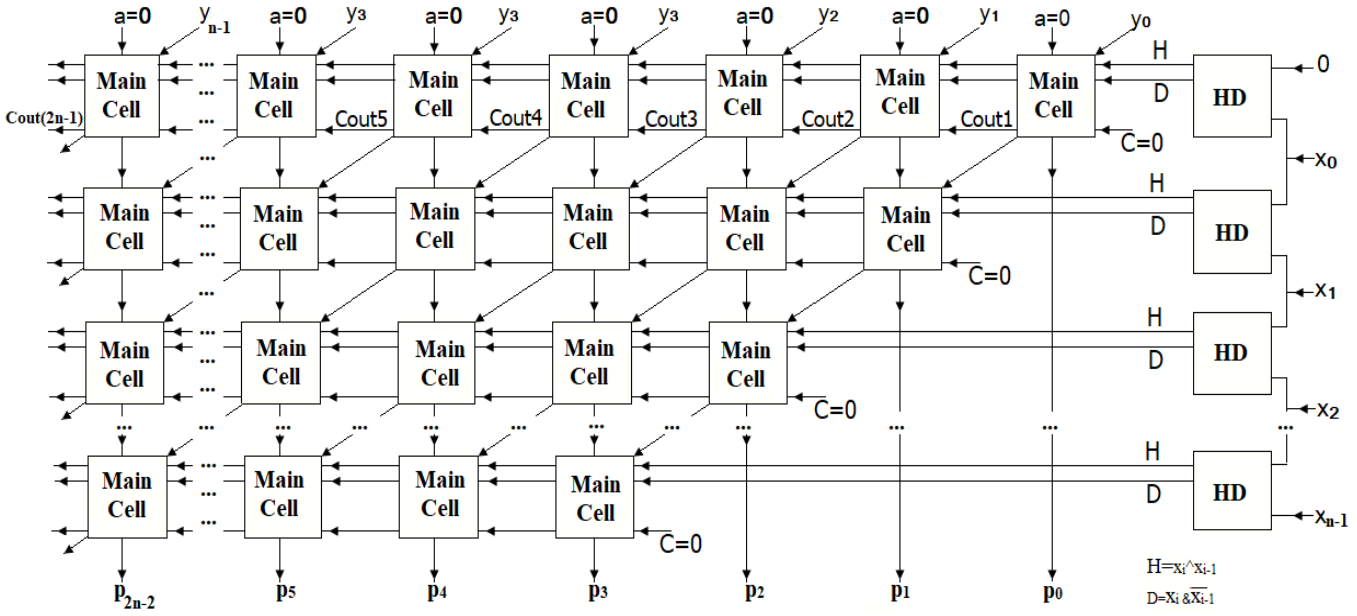
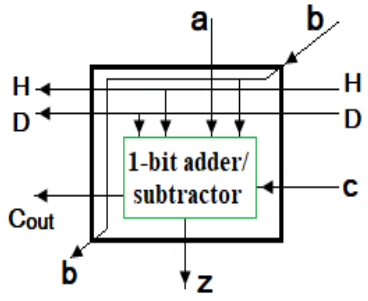


Fig. 7(a) General Reversible Two's Complement Array multiplication Construction

z=a^bH^cH = a **XOR** (b **XOR** c)H

C_{out}= ((a **XOR** D) (b|c)) |bc



Main Cell

Fig. 7(b) Main Cell

when HD=00 then it implies z= no change

z=a Cout=ab | bc | ca

when HD=01 then it implies z= no change

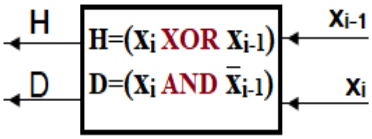
z=a Cout=a'(b | c) | bc

when HD=10 then it implies Full-adder

z=a XOR b XOR c Cout=ab | bc | ca

when HD=11 then it implies Full-subtractor

z=a XOR b XOR c Cout=a'(b | c) | bc



HD Cell

Fig. 7(c) HD Cell

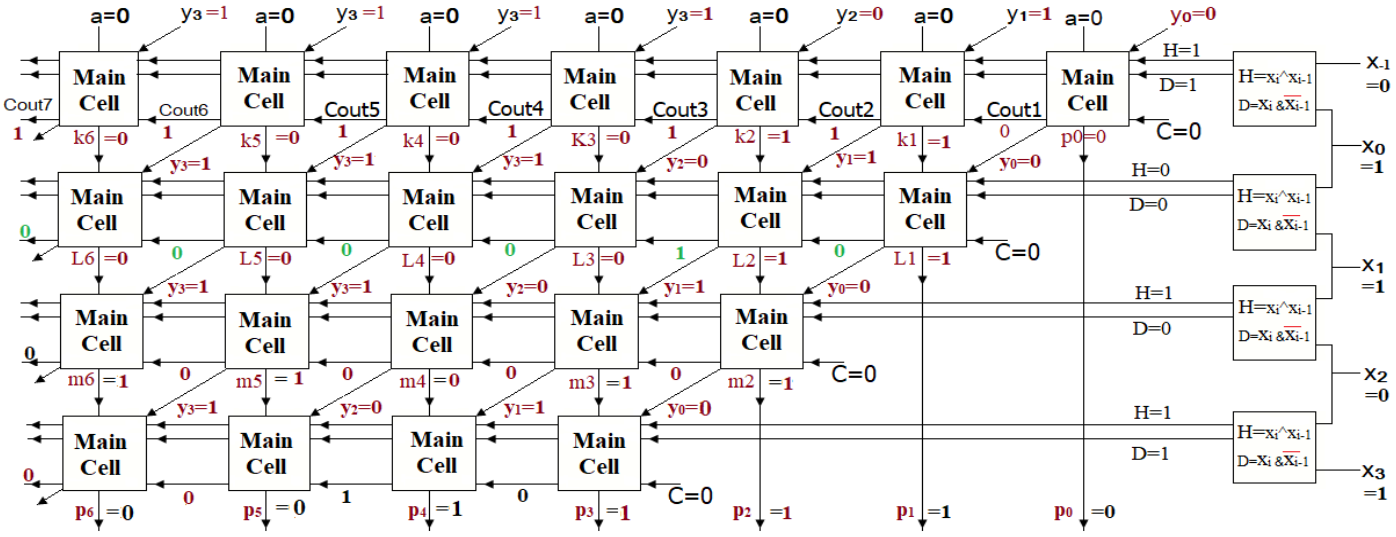
7. Observation of results

For the verification of the 'Reversible Array multiplication of Booth's Algorithm', we have analyzed the circuit results for different inputs. For our convenience, a 4×4 (4-bit by 4-bit) multiplication is performed in three cases: (i) two signed numbers, (ii) one signed number and on unsigned number and (iii) two unsigned numbers. For the first case in Fig 8(a), Multiplicand X and multiplier Y are chosen as

X= x_3, x_2, x_1, x_0 = 1011= -5

Y= y_3, y_2, y_1, y_0 =1010= -6

And output we obtained is=$[0011110]_2$=30 which satisfies the arithmetic operation value(-5×-6).



Multiplication of two negative numbers -5 and -6

X=$[-5]_{10}$=$[1011]_2$ Y=$[-6]_{10}$=$[1010]_2$ X×Y=$[1011]_2$×$[1010]_2$ = $[0011110]_2$= $[30]_{10}$

Fig. 8(a) Circuit for Multiplication of two signed numbers (−5× −6)

Similarly, for the second case in Fig 8(b), multiplicand X and multiplier Y are chosen as

X= x_3, x_2, x_1, x_0 = 0010= 2

Y= y_3, y_2, y_1, y_0 =1000=-8

And output we obtained from the diagram is=$[1110000]_2$=−16 which satisfies the arithmetic operation value (2× −8) = −16.

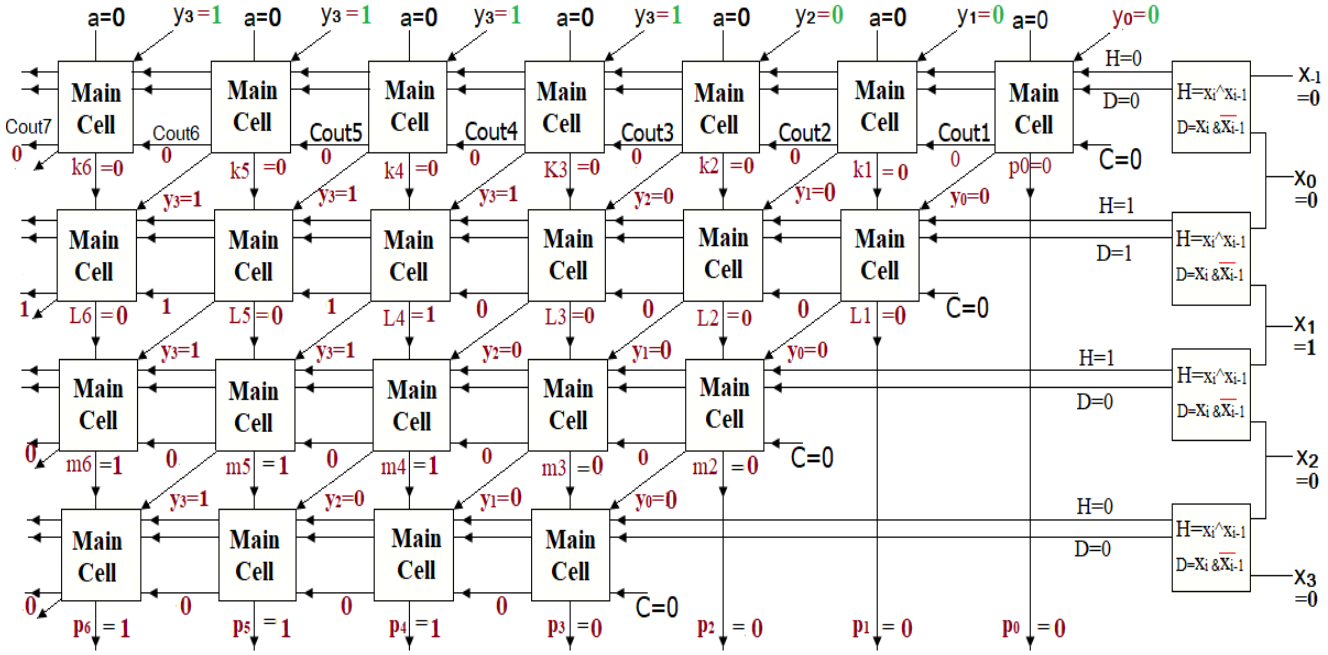


Fig. 8(b) Circuit for Multiplication of 1 one unsigned and signed number (2×-8)

In the same way, we can investigate the third case in Fig 8(c) here, multiplicand X and multiplier Y are chosen as

$$X = x_3, x_2, x_1, x_0 = 0100 = 4$$

$$Y = y_3, y_2, y_1, y_0 = 0100 = 4$$

And output we obtained from the circuit is=$[0010000]_2$=16 which satisfies the arithmetic operation value (4× 4) = 16.

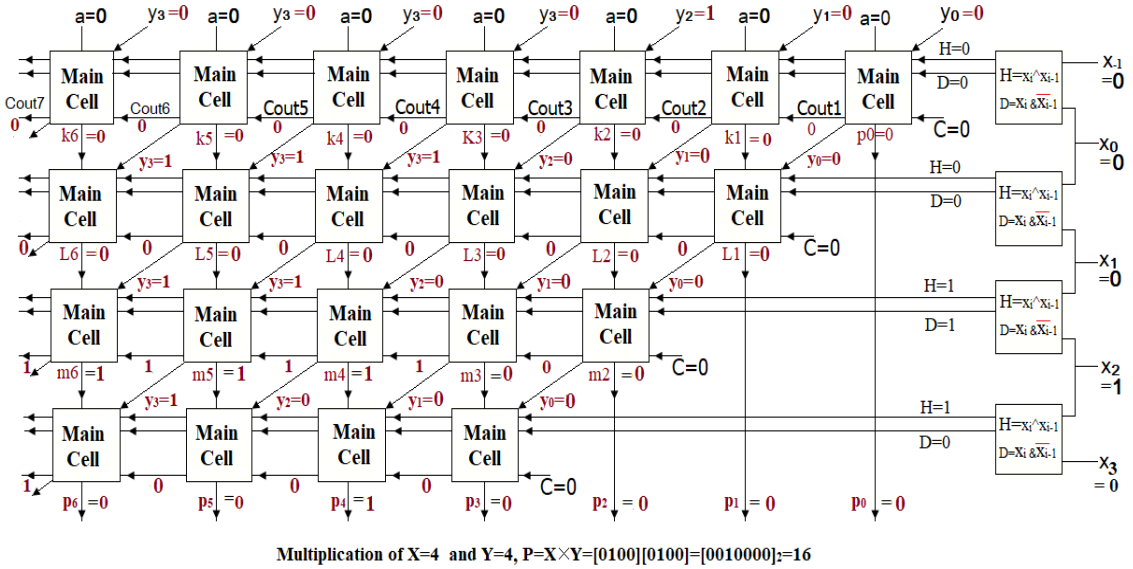


Fig. 8(c) Circuit for Multiplication of two unsigned numbers like 4 *and* 4 (4 ×4)

According to the reversible 4x4 multiplication circuit, some inputs, different carry-outs and output products are given below in the Table-3.

Table-3

Sl. No.	a	c	X= $x_3x_2x_1x_0$	Y= $y_3y_2y_1y_1$	Carry of row1 $c_7c_6c_5c_4c_3c_2c_1$	Carry of row2 $c_7c_6c_5c_4c_3c_2$	Carry of row3 $c_7c_6c_5c_4c_3$	Carry of row4 $c_7c_6c_5c_4$	Product P=X×Y $p_6p_5p_4p_3p_2p_1p_0$
1	0	0	$4=[0100]_2$	$3=[0011]_2$	0000000	000000	11111	1110	0001100=12
2	0	0	$2=[0010]_2$	$-6=[1010]_2$	0000000	111110	00010	1110	1110100=−12
3	0	0	$-4=[1100]_2$	$3=[0011]_2$	0000000	000000	11111	1110	1110100=−12
4	0	0	$2=[0010]_2$	$6=[0110]_2$	0000000	111110	11100	0000	0001100=12
5	0	0	$5=[0101]_2$	$3=[0011]_2$	1111111	111110	11111	1110	0001111=15
6	0	0	$-6=[1010]_2$	$-2=[1110]_2$	0000000	111110	00000	0000	0001100=12
7	0	0	$0=[0000]_2$	$15=[1111]_2$	0000000	000000	00000	0000	0000000=0
8	0	0	$-2=[1110]_2$	$7=[0111]_2$	0000000	111111	11100	1110	1110010=−14
9	0	0	$1=[0001]_2$	$7=[0111]_2$	1111111	111100	00111	0000	0000111=7
10	0	0	$4=[0100]_2$	$-2=[1110]_2$	0000000	000000	11110	0000	1111000=−8
11	0	0	$6=[0110]_2$	$-1=[1111]_2$	0000000	111111	00000	0000	1111010=−6
12	0	0	$-4=[1100]_2$	$-2=[1110]_2$	0000000	000000	11110	0000	0001000=8

8. Some theorems with respect to the $n \times n$ Reversible Multiplier are given below:

Theorem1: Total number of gates T_G required to construct an $(n \times n)$ Reversible Multiplier is not less than $4.5n^2$+1.

Proof. As known that an $n \times n$ Reversible Multiplier contains $\frac{3}{2}$n(n-1) number of "Main Cells" and n number of "HD Cells". Every "Main Cell" has 3 reversible gates, So total number of gates needs to be taken for "Main Cells" is $=3\times\frac{3}{2}$n(n-1). Every "HD Cell" has two reversible gates, so 2n number of gates are needed for this cell. Moreover, n *number of LMain-Cell* cells (Main Cell in left most side) are needed along with the *Main Cell and* each of which has 2 reversible gates, so here 2n number of gates requires. Again $(\frac{n}{2}+1)$ number of FGs needs performing the copy operation.

Number of gates needed = $3\times\frac{3}{2}$n(n-1)+2n+2n + $(\frac{n}{2}+1)=\frac{9}{2}n^2-\frac{9}{2}$n+$\frac{9}{2}$n +1=$4.5n^2$+1

So at least $(4.5n^2+1)$ number of gates required.

i.e. $T_G \geq (4.5n^2+1)$

Theorem 2 Total number of garbage output $T_{Garbage}$ required to construct an $(n \times n)$ Reversible Multiplier is not less than $3n^2$+3n-1.

Proof: 2 garbage outputs are created for every "Main Cell", as there are $\frac{3}{2}n(n-1)$ number of "Main Cells" in $(n \times n)$ Reversible Multiplier, so $3n(n-1)$ number of garbage outputs are there. There are n *number of LMain-Cell* cells (Main Cells in left most side) and everyone has 3 garbage output. There are n number of HD cells and everyone has 2 numbers of garbage output. In addition, $(n-1)$ number of "Main Cells" in the last row creates $(n-1)$ garbage outputs for the purpose of propagating the prime input b.

For realizing the $(n \times n)$ Reversible Multiplier the circuit requires the at least

$\{+\frac{3}{2}n(n-1)\times 2$ + 3n+2n+ (n-1) = $3n^2$-3n+3n+2n+n-1=$3n^2+3n-1\}$ number of garbage outputs

So, $T_{Garbage} \geq 3n^2$+3n-1

Theorem 3: If the processing delay of "Main Cell", "L-Main Cell" and "HD Cell" are D_M, D_{LM} and D_{HD} respectively of the $n \times n$ reversible multiplier. In addition, assume the delay of forwarding gate (FD) is D_{FD} and delay of n × n reversible multiplier is D_{RM}, then

$$DRM \geq (2n-1)\ D_M + D_{LM} + D_{HD} + D_{FD}$$

Proof: In the first row, the longest path in the $n \times n$ reversible multiplier contains the maximum number of "Main Cells" which is (2n-1), in addition there is extra one "L-Main Cell" in this longest path in the leftmost position. In this path there is also a "HD Cell" and one forwarding gate "FD". Hence the total delay $DRM \geq (2n - 1)\ D_M + D_{LM} + D_{HD} + D_{FD}$

9. Conclusion

Reversible Booth's Multiplication is elaborately discussed in this paper. Here, for multiplication purpose, Booth's Multiplier is implemented keeping in view the use of Reversible Gates. Depending on the reversible logic aspects, the evaluation of our proposed multiplication circuit is done. The quantum cost of different gates and more complex circuits like proposed Cell are measured. And keeping track of the quantum value(s) the whole system is analyzed and designed. The performance of this reversible Booth's multiplier is better than that of the design existing in terms of design methodology. In this case the constraints of reversible logic synthesis are preserved. The main role of this design is that it is working both for the unsigned and signed numbers. A Radix-2 or base-2 Booth's Multiplier is implemented with the help of reversible gates in this work.

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. [Online]. Available: http://dx.doi.org/10.1147/rd.53.0183

[2] M. P. Frank, "Introduction to reversible computing: Motivation, progress, and challenges," in Conference on Computing Frontiers, 2005, pp. 385– 390. [Online]. Available: http://doi.acm.org/10.1145/1062261.1062324

[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. [Online]. Available: http://dx.doi.org/10.1109/AICCSA.2006.205074

[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. [Online]. Available: http://dx.doi.org/10.1109/VLSID.2013.215

[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. [Online]. Available: http://dl.acm.org/citation.cfm?id=942792.943112

[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. [Online]. Available: http://dl.acm.org/citation.cfm?id=962758.963476

[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, pp. 334–339.

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

11–20, Feb 1985. [Online]. Available: http//www.osa-opn.org/abstract.cfm?URI=on-11-2-11

[10] T. Toffoli, "Reversible computing," in Colloquium on Automata, Languages and Programming, 1980, pp. 632–644. [Online]. Available: http://dl.acm.org/citation.cfm?id=646234.682540

[11] E. Fredkin and T. Toffoli, "Collision-based computing," vol. 21, pp. 219–253, 1983. [Online]. Available: http://dl.acm.org/citation.cfm?id=644307.644311[12]

[12] A. Peres, "Reversible logic and quantum computers," Phys. Rev. A, vol. 32, pp. 3266–3276, Dec 1985. [Online]. Available:http://link.aps.org/doi/10.1103/PhysRevA.32.3266

[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. [Online]. Available: http://dx.doi.org/10.1109/VLSID.2006.122

[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. [Online]. Available: http://dx.doi.org/10.1007/11572961 66

[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," 1998 McGraw Hill international Editions.