Indian Journal of Science and Technology, *Vol 7(4), 525–529, April 2014*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Booth Multiplier using Reversible Logic with Low Power and Reduced Logical Complexity

Amita Nandal[1*], T. Vigneswaran[2] and Ashwani K. Rana[1]

[1]Department of ECE, NIT Hamirpur-177005, Himachal Pradesh, India; amita_nandal@yahoo.com
[2]School of Electronics Engineering, VIT University, Chennai-600127, Tamil Nadu, India

Abstract

The proposed testable reversible architecture scheme yields significantly reduced complexity, low power and high speed features. It is a key issue in the interface of computation and physics, and of growing importance as miniaturization progresses towards its physical limits. With the advent of nanotechnology the fault detection and testability is of high interest for accuracy. This research work describes the reversible testable design of high-speed modified Booth multipliers. The proposed multiplier circuits are based on the modified Booth algorithm can be used to accelerate the multiplication speed with reduced power consumption. The resultant multiplier circuit shows better performance than others and can be used in the systems requiring very high performance. The proposed booth multiplier design shows 12% reduced logical complexity, 10% reduced power consumption and efficient device utilization achieved in comparison to existing reversible logic.

Keywords: Barrel Shifter, Booth Encoding, Booth Multiplier, Logical Complexity, Partial Products, Reversible Gate and Testability

1. Introduction

Garbage output refers to the output that is not used as a primary output or as an input to other gate. Reversible logic is likely to be in demand in high speed power aware circuits. The digital signal processing (DSP gates used and garbage outputs produced) is one of the core technologies in multimedia and communication systems. Many application systems based on DSP, especially the recent next-generation optical communication systems, require extremely fast processing of a huge amount of digital data. Most of DSP applications such as fast Fourier transform (FFT) require additions and multiplications. Since the multipliers have a significant impact on the performance of the entire system, many high-performance algorithms and architectures have been proposed to accelerate multiplication[1,9]. Well-structured design of the booth multiplier is given by Wang[2]. Further optimization can be achieved using reversible logic[11]. The reversible computation in a system can be performed only when the system comprises of reversible gates. These circuits can generate unique output vector from each input vector and vice versa, that is, there is a one-to-one mapping between input and output vectors. The major goal in reversible logic design is to minimize the number of reversible.

1.1 Reversible Logic

Reversible circuits are of high interest in low power CMOS design, optical computing[7], quantum computing and nanotechnology[3]. The most prominent application of reversible logic lies in quantum computers. A quantum computer can be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; each gate performs an elementary unitary operation on one, two or more two–state quantum systems called qubits. Each qubit represents an elementary unit of information corresponding to the classical bit values 0 and 1. Any unitary operation is reversible, hence quantum networks effecting elementary arithmetic operations such

**Author for correspondence*

as addition, multiplication and exponentiation cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible)[8]. There are various reversible gates available in literature[3,5,6,10]. The concept of testability is being introduced for higher accuracy concerns[4–6].

1.2 Booth Encoding and Multiplier

Booth[9] introduced Booth Multiplier (BM). They are also called radix-2 multiplier. The main advantage is that it involves no correction cycles for signed terms. To multiply X by Y using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {–2, –1, 0, 1, 2}. Table 1 shows the rules to generate the encoded signals by booth encoding scheme. After Booth recoding, a digit number is changed from two's complement representation to Booth representation. In some cases, the weight of the latter is less than that of the former one; the Booth algorithm can be used in multiplier to reduce the number of partial products. But it is not for all the cases. For example, the weight of binary number 010101 is 3, the Booth representation for this number is (+ 1 – 1 + 1 – 1 + 1 – 1), and its weight is 6. Multiple bits scanning per cycle in booth recoding reduces the number of partial products significantly, hence reduces the area, power and speeds up the designs.

2. Proposed Work

R1 gate is used along with R2 gate (a 4 input and 4 output Feynman Gate) to provide the feature of testability called CB(Combined Block) that is called Rgate logic[5,6]. The Haghparast Navi Gate (HNG)[3] was used along with 4*4 Feynman gate to provide testability feature and a name was proposed as New Testable Gate (NTG) shown in Figure 1. The proposed NTG logic has less logical computation as compared to existing testable reversible logic shown in Table 2. The truth table is shown in Table 3.

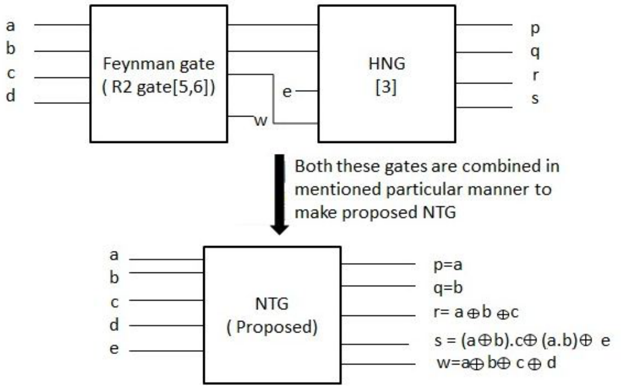


Figure 1. Proposed NTG (New Testable Gate) gate.

In proposed approach, any Boolean function can be implemented using new testable gate along with testing of fault. The testing of fault in HNG or FG can be done by comparing the output W of Feynman gate with output R of the HNG gate. The output W of Feynman gate should be complementary with the output R of the HNG gate. If they are not complementary, it means there is a fault either in FG or HNG. Based on the particular input combinations all types of gates can be implemented. Both combinational circuits and sequential reversible circuits can be designed using the proposed method.

3. Design and Implementation

In booth multiplier the reversible adders and reversible barrel shifters are used as basic building block, as shown in Figure 2 and Figure 3 respectively. The logic behind booth multiplier implemented using testable reversible NTG logic is shown in Figure 4. The computation results for delay, power and logical complexity are compared later.

Table 1. Booth Encoding

Binary number(a) 3-bit groups	Booth encoding
000	0
001	a
010	a
011	2a
100	–2a
101	–a
110	–a
111	0

Table 2. Logical complexity for various testable reversible logics

	Number of XOR operations(α)	Number of AND operations(β)	Logical Complexity
Rgate logic[5, 6]	12	4	$12\ \alpha + 4\ \beta$
NTG logic (Proposed)	8	2	$8\ \alpha + 2\ \beta$

Table 3. Truth table of proposed NTG gate

Inputs					Outputs				
a	b	c	d	e	p	q	r	s	w
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	0	1	0	1
0	0	1	0	1	0	0	1	1	1
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	0	0	1	1	0
0	1	0	0	0	0	1	1	0	1
0	1	0	0	1	0	1	1	1	1
0	1	0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1	1	0
0	1	1	0	0	0	1	0	1	0
0	1	1	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
0	1	1	1	1	0	1	0	0	1
1	0	0	0	0	1	0	1	0	1
1	0	0	0	1	1	0	1	1	1
1	0	0	1	0	1	0	1	0	0
1	0	0	1	1	1	0	1	1	0
1	0	1	0	0	1	0	0	1	0
1	0	1	0	1	1	0	0	0	0
1	0	1	1	0	1	0	0	1	1
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	1	1	0	1	0
1	1	0	0	1	1	1	0	0	0
1	1	0	1	0	1	1	0	1	1
1	1	0	1	1	1	1	0	0	1
1	1	1	0	0	1	1	1	1	1
1	1	1	0	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	0	0

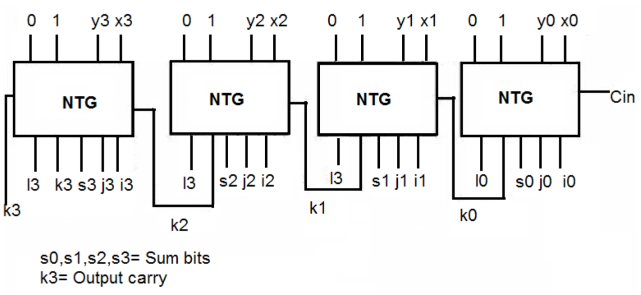


Figure 2. 4-bit ripple adder design using proposed NTG logic.

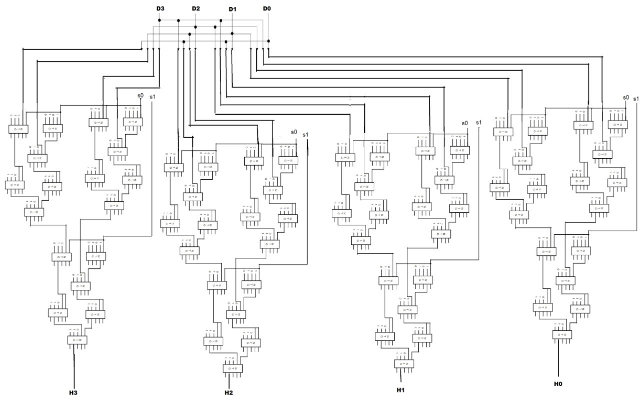
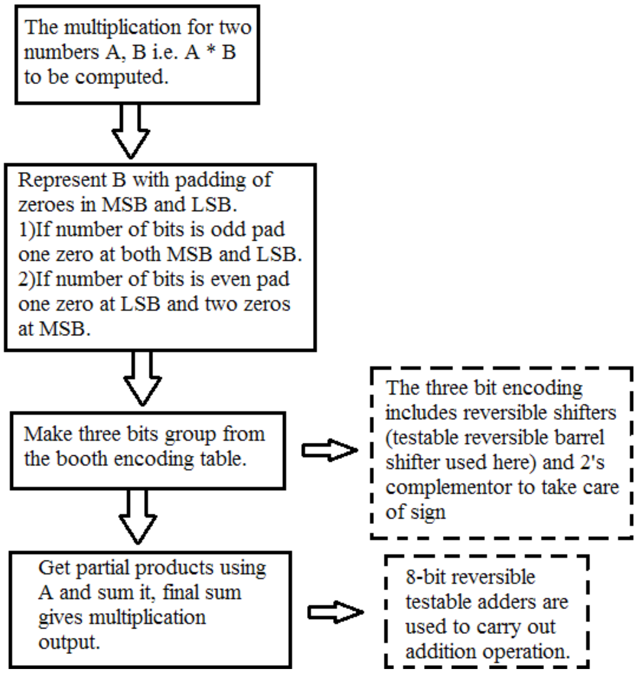


Figure 3. Barrel shifter design using proposed NTG logic.



The multiplication for two numbers A, B i.e. A * B to be computed.

Represent B with padding of zeroes in MSB and LSB.
1)If number of bits is odd pad one zero at both MSB and LSB.
2)If number of bits is even pad one zero at LSB and two zeros at MSB.

Make three bits group from the booth encoding table.

The three bit encoding includes reversible shifters (testable reversible barrel shifter used here) and 2's complementor to take care of sign

Get partial products using A and sum it, final sum gives multiplication output.

8-bit reversible testable adders are used to carry out addition operation.

Figure 4. Booth multiplication using proposed NTG logic.

The booth multiplication has been explained with the following steps:

Step1: Consider two numbers 'a' be multiplicand and 'b' be multiplier. Suppose a = 7 and b = 12, i.e output should be 84.

Step2: The number b can be represented as n = 4 bits i.e. b = 1100. So two zero padding is done in MSB and one zero padding in LSB. Now, b = 0011000.

Step3: The partial product computation shown in Table 4 where encoding is done with the help of Table 1.

Multiplication output = P1 + P2 + P3

= 00000000 + 11100100 + 01110000(8 bit reversible adders used here)

= 01010100 (Carry out = 1, For 4-bit multiplication the Carry out is ignored)

= 84

4. Results and Discussions

In this paper, reversible booth multiplier is implemented using VHDL. Functional simulation is done in MODELSIM and synthesis is carried out using Xilinx ISE-8.2i. Device utilization, delay and power results are compared among various logics. The power comparison is shown in Figure 5. From the analysis, savings in power is achieved using proposed NTG logic. The proposed booth multiplier using NTG logic shows 9% reduction in number of slices and 10% reduction in number of 4-input LUT used as compared to Rgate logic[5,6], Figure 6 and Figure 7 respectively. The NTG (proposed logic) provides

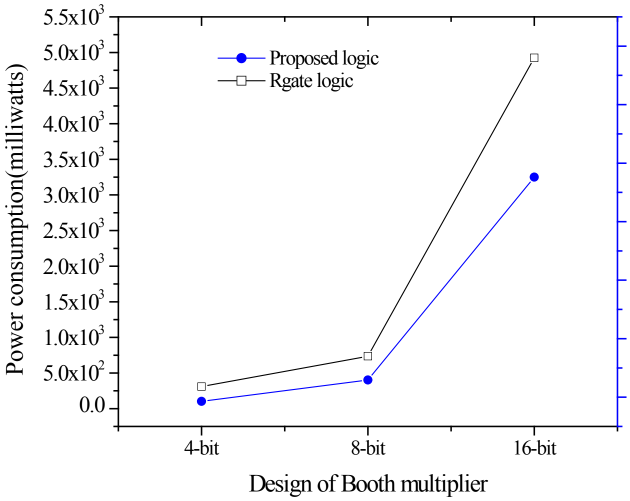


Figure 5. The power comparison results for reversible logic designs for booth multiplier.

Table 4. Partial products calculation and booth encoding

3-bit grouping	From booth encoding table(partial products)	Partial products computed for a = 7, i.e. 111 and b = 12, i.e. 1100
000	0	P1 = 00000000
110	−a	P2 = 11100100
001	a	P3 = 01110000

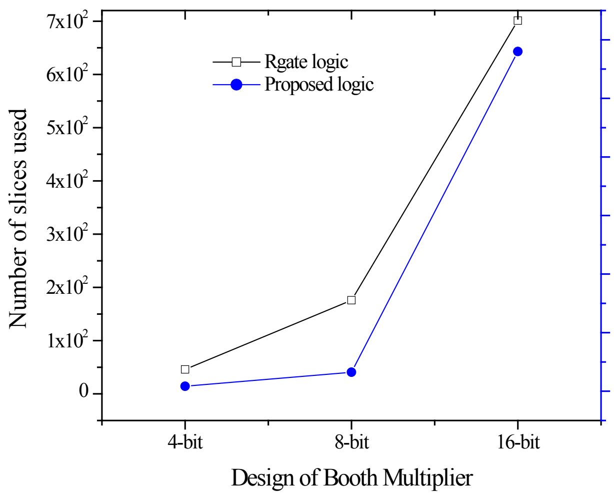


Figure 6. Comparison of number of slices used (device spartan3-3s200ft256-4) among Rgate[5,6] and proposed NTG reversible logic based booth multiplier.

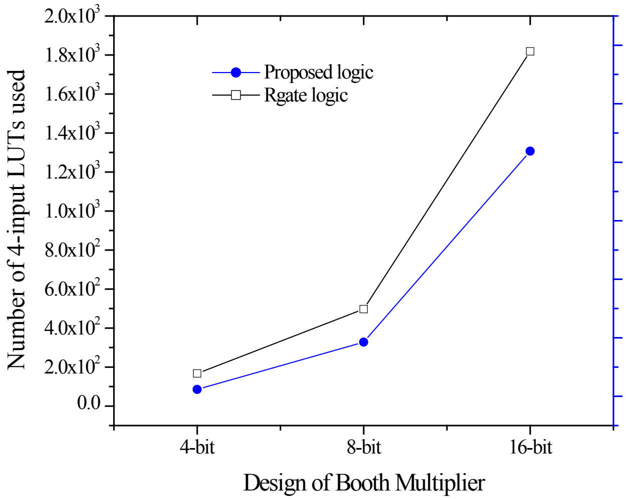


Figure 7. Comparison of number of 4-input LUT used (device spartan3-3s200ft256-4) among Rgate[5,6] and proposed NTG reversible logic based booth multiplier.

Table 5. Reversible booth multiplier logical complexity comparison

4-bit Reversible Booth multiplier	Total no. of reversible gates used	Logical Complexity
Rgate[5, 6]	352	$(12^{*}352)\alpha + (4^{*}352)\beta$
NTG(Proposed)	176	$(8^{*}176)\alpha + (2^{*}176)\beta$
Comments	The NTG (proposed logic) provides 12% reduced logical complexity which can be seen from the number of XOR operations (α) and number of AND operations (β).	

12% reduced logical complexity which can be seen from the number of XOR operations (α) and number of AND operations (β) in Table 5. The proposed booth multiplier using proposed NTG logic has speed improvement over existing Rgate logic[5,6] based reversible design (Figure 8). Figure 9 shows the comparison of overall performance metric i.e. Power Delay Product (PDP).

5. Conclusion

This research work is about efficient reversible realization of booth multipliers. The proposed gate (NTG) is shown better than the existing logic based testable reversible gate in terms of logical computation complexity, speed and power computation. The proposed work forms an important step in the building of complex logic circuits for quantum computers. An interesting future work would be to develop efficient reversible counter and further it's could be advantageous to design an arithmetic and logic unit. Proposed reversible logic can be used with technologies such as CMOS, in particular adiabatic CMOS, Optical, thermodynamic technology, Nanotechnology & DNA technology. Sometimes the problem lies with the tradeoff between parallelism and the combination of speed and power.

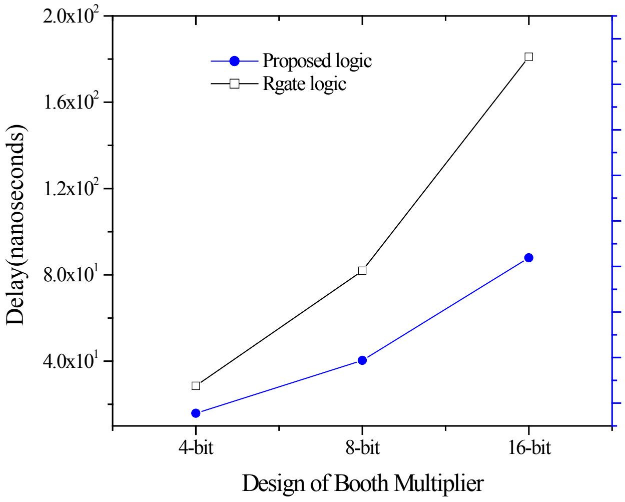


Figure 8. Comparison of delay (device spartan3-3s200ft256-4) among Rgate[5,6] and proposed NTG reversible logic based booth multiplier.

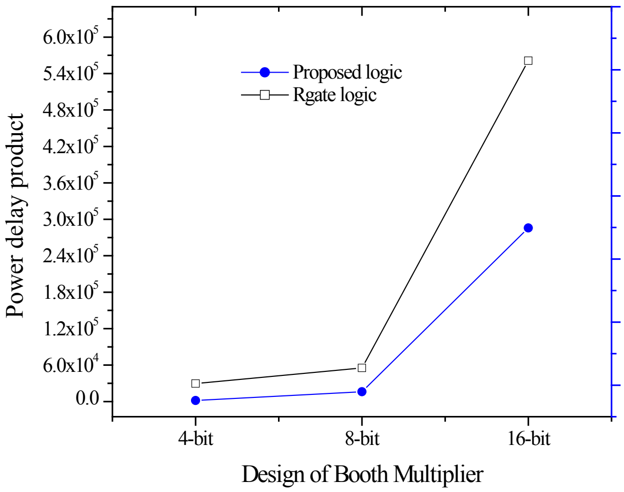


Figure 9. Comparison of overall performance metric Power Delay Product (PDP).

6. References

1. Kang J-Y, Gaudiot J-L. A simple high-speed multiplier design. IEEE Trans on Comput. 2006; 55(10):1253–58.
2. Wang L-R, Jou S-J, Lee C-L. A well-structured modified booth multiplier design. IEEE International Symposium on VLSI Design, Automation and Test. VLSI-DAT; 2008 Apr 23–25; Hsinchu. IEEE. p. 85–8.
3. Haghparast M, Navi K. A Novel reversible BCD adder for nanotechnology based systems. Am J Applied Sci. 2007; 5(3):282–8.
4. Boykin PO, Roychowdhury VP. Reversible fault-tolerant logic. Proceedings International Conference on Dependable Systems and Networks. 2005. p. 444–53.
5. Vasudevan DP, Lala PK, Parkerson JP. Online testable reversible logic circuit design using NAND blocks. 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. DFT. 2004 Oct 10–13. IEEE. p. 324–31.
6. Vasudevan DP, Lala PK, Parkerson J. Reversible-logic design with online testability. IEEE Transactions on Instrumentation and Measurement. 2006 Apr; 55(2):406–14.
7. Milburn GJ. Quantum optical Fredkin gate. Phys Rev Lett. 1989; 62(18):2124–27.
8. Bennett CH. Logical reversibility of computation. IBM Journal Research and Development. 1973; 17(6), 525–32.
9. Booth AD. A signed binary multiplication technique. Q J Mech Appl Math. 1951; 4(2); 236–40.
10. Fredkin E, Toffoli T. Conservative Logic. Int J Theor Phys. 1982; 21(3–4):219–53.
11. Khlopotine A, Perkowski M, Kerntopf P. Reversible logic synthesis by gate composition. Proceedings of IWLS. 2002; 261–66.