

OPTIMIZATION OF MULTIPLIER IN REVERSIBLE LOGIC

N. Bhuvanewary

Assistant Professor, Department of ECE, Kalasalingam Academy of Research And Education,
Anand Nagar. Krishnankoil, Virudhunagar District, (India).

E-mail: bhuvanewary.n@klu.ac.in

ORCID: <https://orcid.org/0000-0001-6400-6602>

Adhi Lakshmi

Associate Professor, Department of ECE, Kalasalingam Academy of Research And Education,
Anand Nagar. Krishnankoil, Virudhunagar District, (India).

E-mail: lakshmi@klu.ac.in

ORCID: <https://orcid.org/0000-0002-6744-7048>

Recepción: 25/10/2019 **Aceptación:** 30/09/2020 **Publicación:** 30/11/2021

Citación sugerida:

Bhuvanewary, N., y Lakshmi, A. (2021). Optimization of multiplier in reversible logic. *3C Tecnología. Glosas de innovación aplicadas a la pyme, Edición Especial*, (noviembre, 2021), 113-123. <https://doi.org/10.17993/3ctecno.2021.specialissue8.113-123>

ABSTRACT

Reversible logic is leading area in power consumption. Based on its application, its emerging trend in power consumption. In ideal situations, reversible circuit yield nil power. Different methods of multiplier optimized in this paper. Like quantum computers, multipliers required to develop computational units. In the paper, two different methods of multiplier developed with very large bit width. Based on partial products hierarchical method is developed. Another method is Karatsuba's algorithm developed based on divide and conquers method. Finally, we compare the results of both two methods. The projected reversible multipliers are enhanced in terms of quantum cost, number of constant inputs, number of garbage outputs and complexity in hardware. In nanotechnology applications this multiplier can be used to construct complex system.

KEYWORDS

Karatsuba's algorithm, Hierarchical method, FFT, Reversible gates.

1. INTRODUCTION

One of the foremost issues in VLSI design is reducing power. In early days, due to information loss irreversible logic leads to power dissipation (Zhirnov *et al.*, 2003). When one bit information lost, it leads to dissipate at least $KT\ln 2$ joules of energy, where $K=1.380650 \times 10^{-23} \text{ m}^2\text{kg}^{-2}\text{K}^{-1}$ (joules Kelvin⁻¹) is the Boltzmann's constant and T is the temperature (Zhirnov *et al.*, 2003). Reversible logic blocks do not lose information because it has internally zero power dissipation. To eliminate $KT\ln 2$ joules of energy dissipation, circuit must be developed with reversible logic gates.

In the subsequent, based on the reversible logic multipliers are synthesized. Initially multipliers synthesized based on Toffoli circuit (Karatsuba & Ofman, 1963; Islam *et al.*, 2009). Also, multiplier can be synthesized based on different reversible gates. In addition to that, multipliers have been proposed with large bit-width.

This paper proposes a multiplier in reversible logic with large bit-width. Two methods are projected, the foremost method is hierarchical method and another one is Karatsuba method applied to the application of FFT.

Respite of the paper systematized as follows. Reversible logic gates ideas and necessity described in Segment II. In Segment III, basics of reversible functions and circuits are discussed. Segment IV provides algorithm of proposed work. Segment V provides the design of Hierarchical and Karatsuba method. Segment VI determines the conclusion.

2. REVERSIBLE LOGIC GATES

In segment II, basic reversible logic gates are described which is being used in the design. Figure 1 displays a 3*3 Toffoli gate. The input path is I (A, B and C) and the output path is O (P, Q, and R). The outputs of the gate are well defined by $P=A$, $Q=B$, $R=AB \oplus C$.

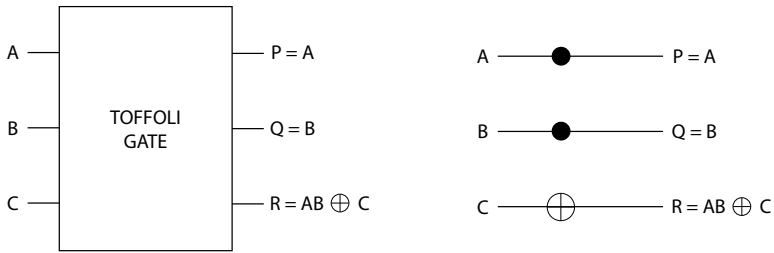


Figure 1. Toffoli Gate.
Source: own elaboration.

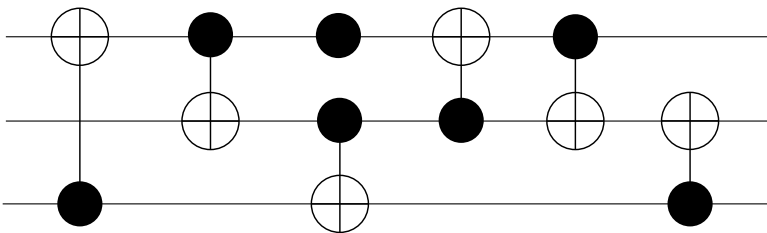


Figure 2. Toffoli circuit.
Source: own elaboration.

Figure 2 shows a Toffoli circuit with 6 gates, quantum cost value 10, cost of transistor 56 and 3 circuit lines respectively. Black circle ● is denoted as control lines and the cross line ⊕ denoted as target lines.

3. BASICS OF REVERSIBLE LOGIC

Demarcation 1: A reversible function $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ is defined as if (1) its amount of outputs is same as the amount of inputs and (2) it plots every input design to a distinctive output.

Reversible function is required to be embed in irreversible to reduce data loss, which requires circuit lines to produce constant inputs and garbage outputs. The least count of circuit lines is added, and it determined by number of existences of the most numerous output pattern (Toffoli, 1980). Reversible functions realized based on restrictions as if fan-out and feedback are not indorsed (Wille & Drechsler, 2009).

Demarcation 2: Inputs in reversible circuit G $X = \{x_i \mid 1 \leq i \leq n\}$ determined by reversible gates g_i cascaded, i.e. $G = \prod_{i=1}^d g_i$ where the number of gate denoted as d . In general, different reversible gates are there. In which most used gate is Toffoli because it is universal, multiple gate control.

Demarcation 3: Let the domain function denoted as $X := \{x_i \mid 1 \leq i \leq n\}$. $g(C; t)$ is the form of universal multiple control Toffoli gate, where $C = \{x_{ji} \mid 1 \leq i \leq k\} \subset X$ control lines set denoted as \mathbf{X} and with $t \in C$ the target line denoted as $t = x_l$. The target line is inverted when control lines set assigned as one, that is., When control line is empty the input vector of the gate has been charted to $\prod_{i=1}^{l-1} x_i \oplus 1, \prod_{i=l+1}^n x_i$, then the target is inverted. In the subsequent, Toffoli gates are referred as multiple control gate. Toffoli gate also called as Not gate when it has zero control lines.

4. PROPOSED REVERSIBLE MULTIPLIER

4.1. HIERARCHICAL METHOD

In this segment, multiplier synthesized based on hierarchical method by means of controlled increasers. To calculate partial products, multiply two factors $a = \sum_{i=0}^{n-1} a_i \cdot 2^i$ and $b = \sum_{i=0}^{n-1} b_i \cdot 2^i$ then add the products together, i.e. $a \cdot b = \sum_{i=0}^{n-1} (a_i \cdot \sum_{j=0}^{n-1} b_j \cdot 2^j) \cdot 2^i$. When the bit a_i assigned as one then the particular bit of b_j multiplied by particular rule of 2 is additional to the product. This can be apprehended by controlled functions. Therefore, by using hierarchical method number of multiple n -bit apprehends multiplier by distinct n -bit controlled increasers. The factors of multiple bit multiplication are $a = \prod_{i=0}^{n-1} a_i$ and $b = \prod_{i=0}^{n-1} b_i$ as well as the product $c = \prod_{i=0}^{2n-1} c_i$ here, the i -th controlled increaser is controlled by a_i . It conditionally adds the value of b to $\prod_{j=i}^{n-1} c_j$ i.e. to the n bits of the product c beginning from the i -th bit. To modify the j -th bit of the product, lower bit does not consider until the j -th controlled increaser. Therefore, by a bit shifter, we can realize the controlled increasers without considering lower bits. The $n + i$ -th position of the product carries carry, which further not used and, thus, carries zero.

$$\begin{array}{l}
 1 \text{ for } i = 0 \text{ to } n - 1 \\
 2 \{ \\
 3 \quad \prod_{j=i}^{n+i} c_j + a_i = \prod_{i=0}^{n-1} b_i \\
 4 \}
 \end{array}$$

Example 1: Consider a 3-bit factors based on hierarchical $a = a_2a_1a_0$ and $b = b_2b_1b_0$ along with the product $c = 5c_4c_3c_2c_1c_0$. To realize this kind of multiplication we required 3 controlled increasers. a_0 is control the initial controlled increaser. In addition to that it temporarily added 2^{nd} factor b to last 3 LSB of product $c_2c_1c_0$. Carry writes over to third MSB bit c_3 . a_1 is control the 2^{nd} controlled increaser. In addition to that it temporarily added 2^{nd} factor b with product c , i.e. to $c_3c_2c_1$. Carry writes over to second MSB bit c_4 . a_2 controls the 3rd controlled increaser. In addition to that it temporarily added 2^{nd} factor b with product c , i.e. to $c_4c_3c_2$. Carry writes over to MSB bit c_5 .

4.2. KARATSUBA METHOD

In this segment Karatsuba’s algorithm, based multiplier designed with divide and conquer method. Based on this method multiplication realized by multiplying two factors with smaller bit-width and additionally perform some less expensive operations. Consider an n -bit multiplication with $m = 2 \cdot k$, Both multiple factors (example. $a = \sum_{i=0}^{2 \cdot k - 1} a_i \cdot 2^i$ are split as an upper part $\bar{a} := \sum_{i=k}^{2 \cdot k - 1} a_i \cdot 2^{i-k}$ and a lower part $\underline{a} := \sum_{i=0}^{k-1} a_i \cdot 2^i$ such that $a = \bar{a} \cdot 2^k + \underline{a}$. With this demonstration the subsequent equations are:

$$\begin{aligned}
 a \cdot b &= (\bar{a} \cdot 2^k + \underline{a}) \cdot (\bar{b} \cdot 2^k + \underline{b}) \\
 &= \bar{a} \cdot \bar{b} \cdot 2^{2 \cdot k} + (\underline{a} \cdot \bar{b} + \bar{a} \cdot \underline{b}) \cdot 2^k + \underline{a} \cdot \underline{b} \\
 &= \bar{a} \cdot \bar{b} \cdot 2^{2 \cdot k} + (\underline{a} \cdot \bar{b} + \bar{a} \cdot \underline{b} + \bar{a} \cdot \bar{b} + \underline{a} \cdot \underline{b} - \bar{a} \cdot \bar{b} - \underline{a} \cdot \underline{b}) \cdot 2^k + \underline{a} \cdot \underline{b} \\
 &= \bar{a} \cdot \bar{b} \cdot 2^{2 \cdot k} + (\underline{a} \cdot (\underline{b} + \bar{b}) + \bar{a} \cdot (\underline{b} + \bar{b}) - \bar{a} \cdot \bar{b} - \underline{a} \cdot \underline{b}) \cdot 2^k + \underline{a} \cdot \underline{b} \\
 &= \bar{a} \cdot \bar{b} \cdot 2^{2 \cdot k} + ((\underline{a} + \bar{a}) \cdot (\underline{b} + \bar{b}) - \bar{a} \cdot \bar{b} - \underline{a} \cdot \underline{b}) \cdot 2^k + \underline{a} \cdot \underline{b}
 \end{aligned}$$

These expressions indicate that 3 k -bit multiplications, bit shifts and subtractions realized from a $(2 \cdot k)$ -bit multiplication. In addition to that, circuit lines are necessary for reversible logic. For small bit width Karatsuba method is not applicable, the variable turning Point

signifies the bit-width lower. The Karatsuba method-based multiplication is used afar the turning Point.

Example 2:

Let us deliberate eight-bit Karatsuba multiplication and turning point $t = 6$. The turning point t is less than 8 bit and even the conditional lines 1 and 4 does not hold and computed K as four (line 7). Then, in line eight, new variables d, e, f are initialized which leads to 20 garbage lines i.e $4 \cdot 4 + 4 = 20$. Then, the 2 minor multiplications $c = c_7c_6c_5c_4c_3c_2c_1c_0 = a_3a_2a_1a_0 _ b_3b_2b_1b_0 = \underline{a} \cdot \underline{b}$ (line 9) and $\bar{c} = \bar{a} \cdot \bar{b}$ (line 10) are performed, respectively. In previous segment, same method realized using hierarchical method. Then, the result of the multiplications directly consigned to the product of the bits. To modify the product of the sums this result must be used that calculated next. These sums are $d = d_4d_3d_2d_1d_0 = a_7a_6a_5a_4 + a_3a_2a_1a_0 = \bar{a} + \underline{a}$ (line 11) and $e = \bar{b} + \underline{b}$ (line 12). To perform this operation, copy the 1st sum to the target until it uninitialized and increase the function by adding 2^{nd} sum. To get the 3^{rd} sub product, result of this 1^{st} and 2^{nd} sums are multiplied $h = d \cdot e$ (line 13).

```

1 if ( n < turningPoint)
2   c = MULTH(a,b)
3
4 if ( n % 2 = 1)
5   init an, bn, c2·n+1 with 0
6
7 k := ⌊n/2⌋
8 init d, e (k + 1 bits), f (2 · k + 2 bits ) with 0
9  $\bar{c} = MULT_K(\bar{a}, \bar{b})$ 
10  $\underline{c} = MULT_K(\underline{a}, \underline{b})$ 
11  $\bar{d} = \bar{a} + \underline{a}$ 
12  $e = \bar{b} + \underline{b}$ 
13  $h = MULT_K(d, e)$ 
14 h- =  $\bar{c}$ 
15 h- =  $\underline{c}$ 
16  $\prod_{i=k}^{3 \cdot k + 3} c_i + = h$ 

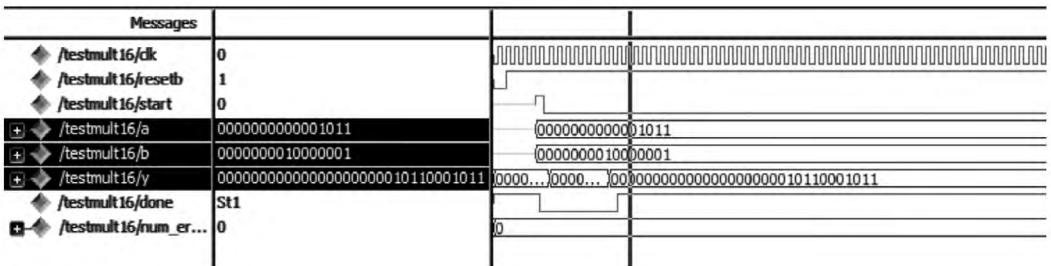
```

Multiplication performed by the hierarchical approach when turning point is greater than the bit.

5. DESIGN OF REVERSIBLE MULTIPLIER

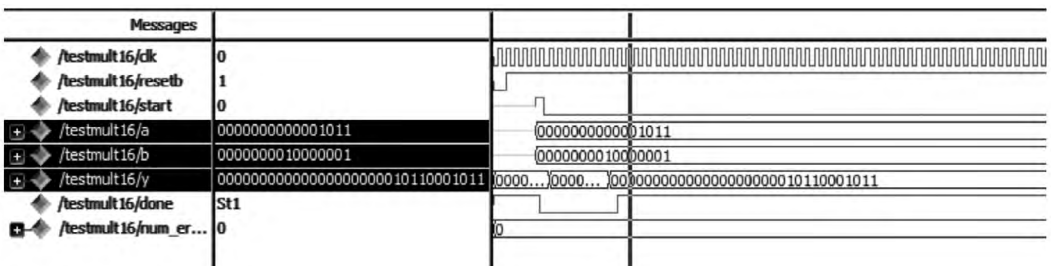
5.1. HIERARCHICAL METHOD

For example, we have to multiply the two binary values i.e. 0000000000001011 and 0000000010000001 representing with 16bits. The simulated waveform for this example is done using reversible logic; the result for this multiplication is 00000000000000000001 0110001011.



5.2. KARATSUBA METHOD

For example, the same values can be multiplied using karatsuba method representing with 16bits.



6. CONCLUSIONS

This paper proposes a multiplier with very large bit-width using reversible logic. The Hierarchical method only applicable for small bit-width but the Karatsuba method applicable for large bit-width. The results are compared using synthesized result of device

utilization of the two methods. The Karatsuba method is better than the Hierarchical method because the device utilization was more in the case of Hierarchical method that is mention in Table 1. The projected reversible multipliers are enhanced in terms of quantum cost, number of constant inputs, number of garbage outputs and complexity in hardware. In nanotechnology applications this multiplier can be used to construct complex system.

This result is taken by 16bit multiplication of two binary values 0000000000001011 and 0000000010000001.

Table 1. Comparison of proposed systems.

	NO. OF FLIP FLOPS	NO. OF 4INPUT LUTS	NO. OF IOBS
Hierarchical method	81	122	68
Karatsuba method	36	72	64

Source: own elaboration.

REFERENCIAS

- Bennett, C. H.** (1973). Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6), 525-532. <https://doi.org/10.1147/rd.176.0525>
- Desoete, B., & De Vos, A.** (2002). A reversible carry-look-ahead adder using control gates. *Integration-the VLSI journal*, 33(1-2), 89–104. <https://biblio.ugent.be/publication/160416>
- Fazel, K., Thornton, M. A., & Rice, J. E.** (2007). ESOP-based Toffoli gate cascade generation. In *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 206-209. <https://ieeexplore.ieee.org/document/4313212>
- Gupta, P., Agrawal, A., & Jha, N. K.** (2006). An algorithm for synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(11), 2317–2330. <https://ieeexplore.ieee.org/document/1715418>

- Haghparast, M., Jassbi, S. J., Navi, K., & Hashemipour, O.** (2008). Design of a novel reversible multiplier circuit using HNG gate in nanotechnology. *World Applied Sciences Journal*, 3(6), 974–978. [https://www.idosi.org/wasj/wasj3\(6\)/19.pdf](https://www.idosi.org/wasj/wasj3(6)/19.pdf)
- Islam, M. S., Rahman, M. M., Begum, Z., & Hafiz, M. Z.** (2009). Low cost quantum realization of reversible multiplier circuit. *Information Technology Journal*, 8(2), 208–213. <https://scialert.net/abstract/?doi=ij.2009.208.213>
- Karatsuba, A., & Ofman, Y.** (1963). Multiplication of many-digital numbers by automatic computers. *Doklady Akad. Nauk SSSR*, 145(2), 293-294. http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=dan&paperid=26729&option_lang=eng
- Landauer, R.** (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3), 183-191. <https://doi.org/10.1147/rd.53.0183>
- Maslov, D., & Dueck, G. W.** (2004). Reversible cascades with minimal garbage. *Computer-Aided Design of Integrated Circuits and Systems*, 23(11), 1497–1509. <https://ieeexplore.ieee.org/document/1350877>
- Miller, D. M., Maslov, D., & Dueck, G. W.** (2003). A transformation based algorithm for reversible logic synthesis. *Proceedings 2003. Design Automation Conference (IEEE Cat. No. 03CH37451)*, pp. 318-323. <http://web.cecs.pdx.edu/~mperkows/temp/March17/dac.pdf>
- Nielsen, M. A., & Chuang, I. L.** (2000). *Quantum Computation and Quantum Information*. Cambridge University Press. <http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf>
- Shende, V. V., Prasad, A. K., Markov, I. L., & Hayes, J. P.** (2003). Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(6), 710–722. <https://ieeexplore.ieee.org/document/1201583>
- Shor, P. W.** (1994). Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134. <https://ieeexplore.ieee.org/document/365700>

- Thapliyal, H., & Srinivas, M. B.** (2006). Novel reversible multiplier architecture using reversible TSG gate. In *International Conference on Computer Systems and Applications*, pp. 100–103. <https://ieeexplore.ieee.org/document/1618339>
- Toffoli, T.** (1980). Reversible computing. In Bakker, J. W. de., and Leeuwen, J. van (eds.) *Automata, Languages and Programming* Springer.
- Vandersypen, L., Steffen, M., Breyta, G., Yannoni, C. S., Sherwood, M. H., & Chuang, I. L.** (2001). Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414, 883–887. <https://doi.org/10.1038/414883a>
- Wille, R., & Drechsler, R.** (2009). BDD-based synthesis of reversible logic for large functions. In *46th ACM/IEEE Design Automation Conference*, pp. 270–275. <https://ieeexplore.ieee.org/document/5227151>
- Zhirnov, V. V., Cavin, R. K., Hutchby, J. A., & Bourianoff, G. I.** (2003). Limits to binary logic switch scaling – a gedanken model. *Proceedings of the IEEE*, 91(11), 1934–1939. <https://ieeexplore.ieee.org/document/1240081>