Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Design and Implementation of General Hardware Binary Multiplier (2^n x 2^n) Bits

To cite this article: Alaa Ghazi Abdulbaqia and Yasir Hashim 2022 *J. Phys.: Conf. Ser.* **2312** 012084

View the article online for updates and enhancements.

You may also like

- Stochastic emulation of quantum algorithms
Daniel Braun and Ronny Müller
- Multiparty quantum random access codes
Debashis Saha and Jakub J. Borkaa
- The Correlation between the Standard Potentials of Metal Electrodes and Redox Electrodes
Alexandr I. Chernomorskii

This content was downloaded from IP address 185.206.172.59 on 18/10/2022 at 08:30

Design and Implementation of General Hardware Binary Multiplier (2^n x 2^n) Bits

Alaa Ghazi Abdulbaqia[1], Yasir Hashim[2,*]

[1]Department of Information Technology, Tishk International University, Kurdstan Region, Erbil, Iraq

[2]Department of Computer Engineering, Tishk International University, Kurdstan Region, Erbil, Iraq

*yasir.hashim@ieee.org

Abstract. In this paper, a new general 2^n x 2^n bits hardware multiplier based on combinatorial has been designed, implemented and analysed. First, a new design for circuit to multiply two binary numbers with 2^n bits length, this new design starts with basic 2x2 bits circuit multiplier, n here equal to 1. Then based on this circuit, the 4x4 bits circuit multiplier has been designed. And based on 4x4, the 8x8 bits multiplier has been designed and continually the 16x16 bits multiplier. The final design for general 2^nx2^n bits multiplier has been presented. All these circuits have been mathematically proved and tested to get the final results.

1. Introduction

Multiplication hardware circuits is a significant function electronics circuits in arithmetic operations [1-2]. The multiplication of two binary numbers (2^n x 2^n) bits and then accumulate the product are among some of the repeatedly used calculation intensive mathematical functions [3-6] presently implemented in many Digital Signal Processing (DSP) applications such as, convolution, fast Fourier transform, filtering. The multiplication of two binary numbers (2^n x 2^n) bits also used in microprocessors arithmetic and logic units. As multiplication process dominates the executions of most mathematical operations in the most applications of signal processing, so there is a necessity always to design a more powerful hardware multiplier.

The implementation of hardware for binary multiplier can be distributing in to combinational and sequential hardware multiplier design. Combinational hardware multipliers are the basic and direct kind of multiplication. The types of combinational multipliers are Array multipliers, Booth multipliers, and Wallace Tree Multiplier [7], all of these types depends on the main definition of multiplication that deals with add and shift operations of numbers to find the final product, while the sequential hardware multipliers are mainly depends on a one circuit of addition to accumulate the result of multiplication. The sequential hardware multipliers are the smaller in size than the combinational hardware multipliers [8-10].

In this paper, a new general 2^n x 2^n bits hardware multiplier based on combinatorial has been designed, implemented and analyzed.

2. Design and Test Result of (2^n x 2^n) Bits Multiplier

The design of the new binary multiplier is construct based on the concept of recursive use of hardware multiplier circuit of 2^{n-1} x 2^{n-1} bits' circuit to construct the hardware multiplier of 2^n x 2^n bits.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

2.1. Design and Test Result of (2 x 2) Bits Multiplier

Firstly, the design will start with two binary numbers A, and B with 2 bits' length, then $2=2^n$, then n=1, it is possible to write:

$A = 2\,a_1 + a_0$ and $B = 2\,b_1 + b_0$

Where a_o, b_o are the minimum significant bits and $2a_1$, $2b_1$ are the most significant bits for the A and B binary numbers.

Then if we want to calculate the multiplication results (C):

C = A x B then

$$C = \sum_{j=0}^{2^2-1} c_j\, 2^j$$

While the number of bits for A and B are 2, So the maximum length of C will be $2^2 = 4$

Now, the multiplication of 2 x 2 bits can be summarized by Figure 1 below.

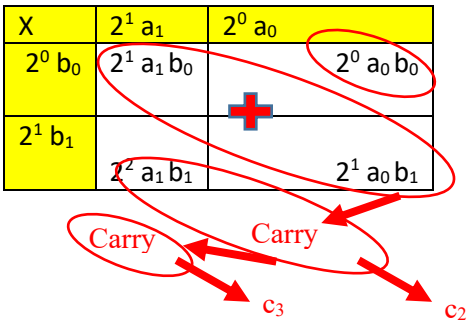

Figure 1. Multiplication of 2 x 2 bits

Then the results of multiplication of 2x2 bits can be expressed in terms of Low part (C_L) and High part (C_H) as following:

$C_L = 2\,c_1 + c_0$, $C_H = 2\,c_3 + c_2$

$$C = \sum_{j=0}^{3} c_j\, 2^j = c_3\, 2^3 + c_2\, 2^2 + c_1\, 2^1 + c_0\, 2^0 = 2^2\, C_H + C_L$$

Where:

$c_0 = a_0\, b_0$

$c_1 = a_1\, b_0 + a_0\, b_1$

$c_2 = a_2\, b_2 + carry\{c_1\}$

$c_3 = carry\{c_2\}$

So to implement the logic circuit to execute 2x2 multiplication using LogicCircuit software version 2.21.01.10, and depending on the above mathematical analysis, in it is possible to construct the hardware circuit of 2 x 2 multiplier using 4 AND Gates (each AND gate behaves like 1 x 1-bit multiplier and 2 Full Adder 1-bit circuits as shown in Figure 2. As illustrated in this circuit, the inputs are A and B as a 2-bits numbers and the output will be 4-bits number, 2-bits low and 2-bits high. As an example to test this circuit, Figure 3 and Figure 4 represent the results of multiplying 3 by 3 and 3 by 2 in binary form and the results are 9 and 6.

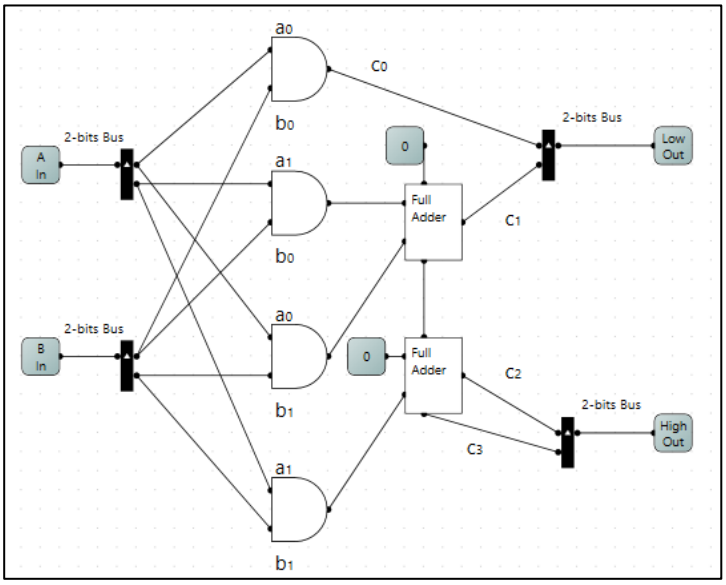

Figure 2. Hardware multiplier of 2x2 bits.

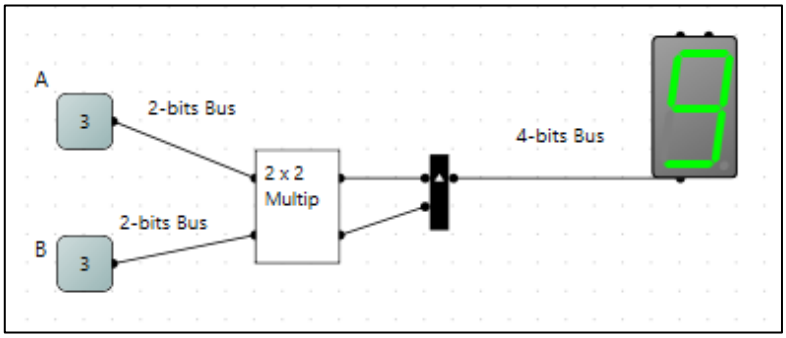

Figure 3. Test of 2x2 bits multiplier, by Appling (11_2x11_2), the result is (1001_2) (9_{16}).

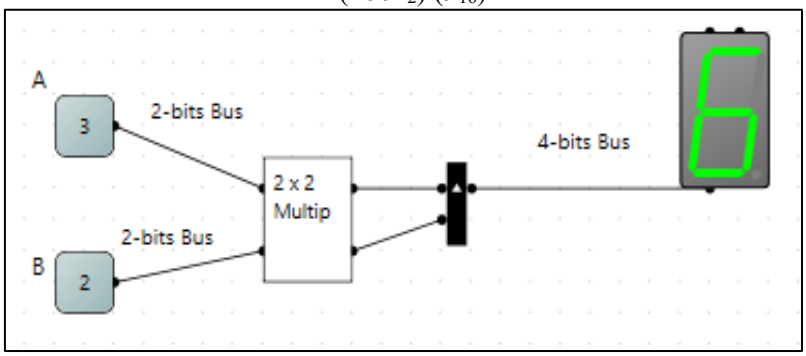

Figure 4. Test of 2x2 bits multiplier, by Appling (11_2x10_2), the result is (0110_2) (6_{16}).

2.2. Design and Test Result of (4 x 4) Bits Multiplier

Following the same design principal of 2X2 bits multiplier, the 4 x 4 bits' multiplier could be design, so $4=2^n$, then n=2, we can summarize the results to be as following:

$$A = 2^2 A_H + A_L \quad \text{, and} \quad B = 2^2 B_H + B_L$$
$$C = A\,B = (2^2 A_H + A_L)(2^2 B_H + B_L)$$
$$C = A_L B_L + 2^2 (A_H B_L + A_L B_H) + 2^4 A_H B_H$$
$$C = (A_L B_L)_L + 2^2 (A_L B_L)_H + 2^2 [(A_H B_L)_L + 2^2 (A_H B_L)_H + (A_L B_H)_L + 2^2 (A_L B_H)_H] + 2^4 [(A_H B_H)_L + 2^2 (A_H B_H)_H]$$
$$C = (A_L B_L)_L + 2^2 [((A_H B_L)_L + (A_L B_H)_L) + (A_L B_L)_H] + 2^4 [((A_H B_L)_H + (A_L B_H)_H) + (A_H B_H)_L] + + 2^6 [(A_H B_H)_H]$$

In this case we can separate the result (C) into Low and High parts C_H and C_L, and then separate each part to Low and High as below:

$$C = C_L + 2^4 C_H = C_{LL} + 2^2 C_{LH} + 2^4 (C_{HL} + 2^2 C_{HH})$$
$$C = C_L + 2^4 C_H = C_{LL} + 2^2 C_{LH} + 2^4 C_{HL} + 2^6 C_{HH}$$

By comparing the two equations we can conclude that:

$$C_{LL} = (A_L B_L)_L$$
$$C_{LH} = ((A_H B_L)_L + (A_L B_H)_L) + (A_L B_L)_H$$
$$C_{HL} = ((A_H B_L)_H + (A_L B_H)_H) + (A_H B_H)_L + carry\,\{C_{LH}\}$$
$$C_{HH} = (A_H B_H)_H + carry\,\{C_{HL}\}$$

*T*hose four equations can be realized by the circuit in Figure 5 utilizing the four 2 x 2 Multiplier Circuits that represents in Figure 2 and five 2-bits parallel Adders. As an example to execute this circuit, Figure 6 shows the results of (1111_2x1010_2) is (1001 0110_2) (96_{16}).

2.3. Design and Test Result of (8 x 8) and (16 x 16) Bits Multiplier

Similarly, we can expand the idea to construct 8 x 8 Multiplier, and 16 x 16 Multiplier as shown in the Figures 7, to 10.

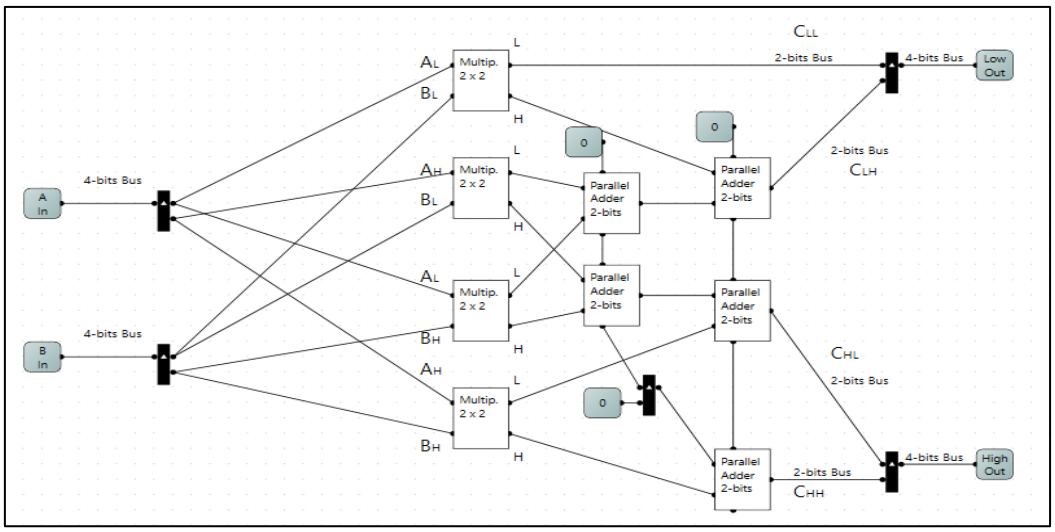

Figure 5. Hardware multiplier of 4x4 bits.

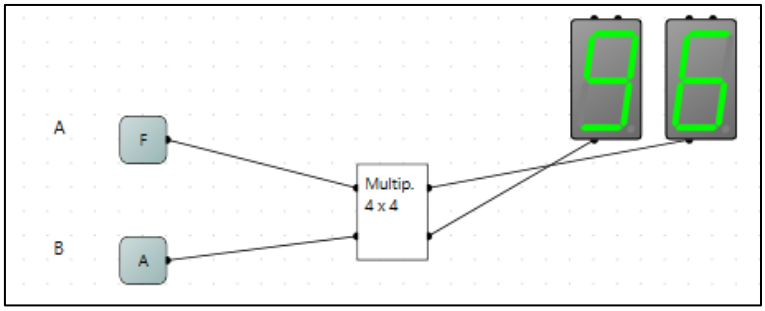

Figure 6. Test of 4x4 bits multiplier, by Appling (1111_2x1010_2), the result is ($1001\ 0110_2$) (96_{16}).

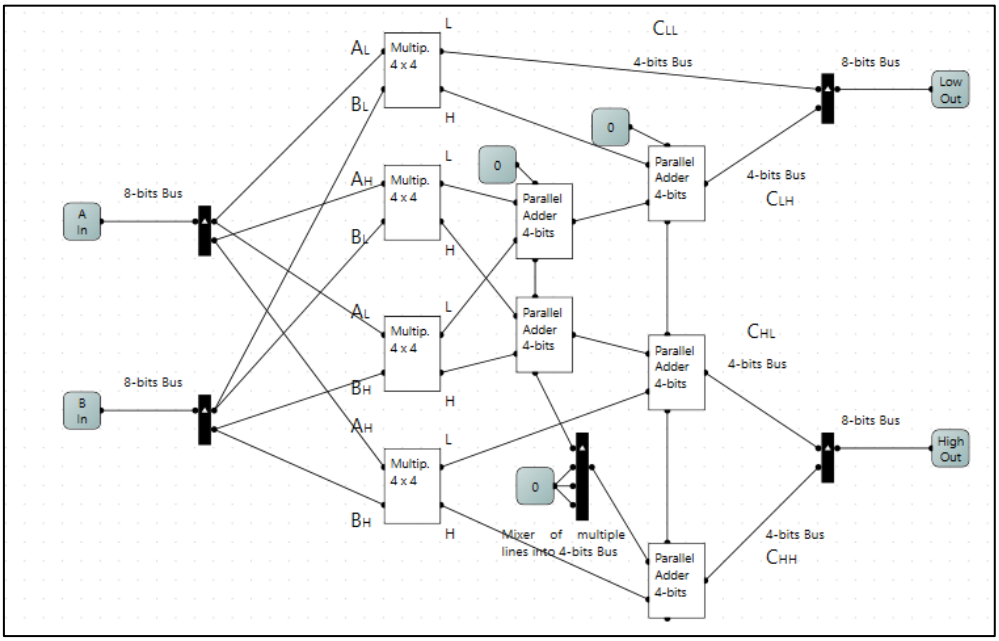

Figure 7. Hardware multiplier of 8x8 bits.

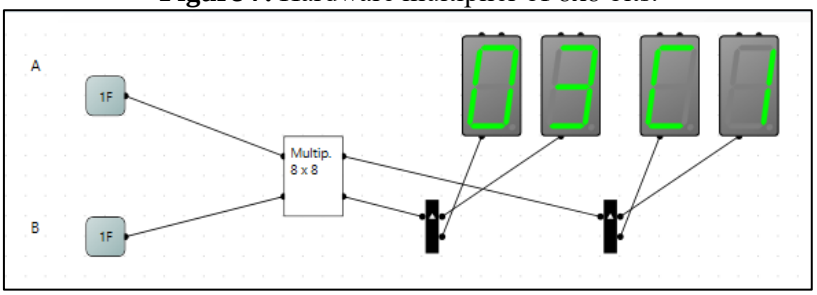

Figure 8. Test of 8x8 bits multiplier, by Appling ($0001\ 1111_2$x$0001\ 1111_2$), the result is ($0000\ 0011\ 1100\ 0001_2$) ($03C1_{16}$).

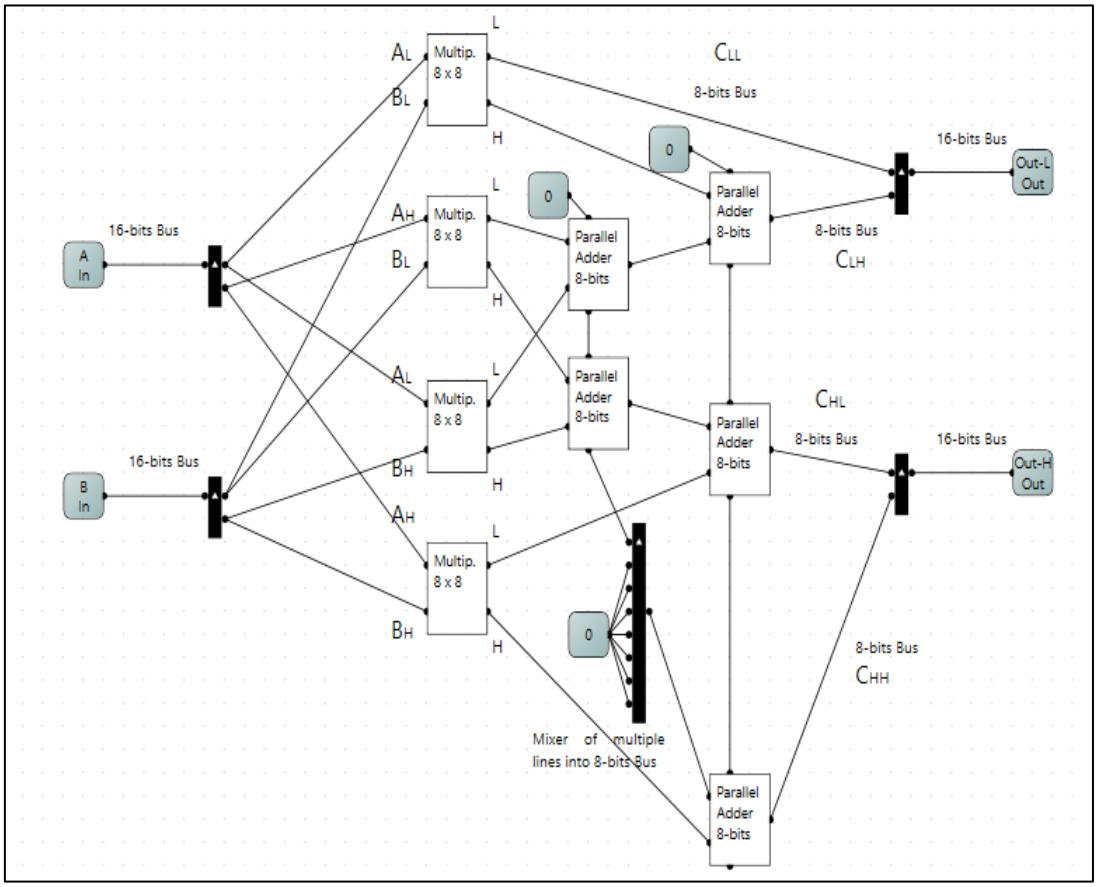

Figure 9. Hardware multiplier of 16x16 bits.

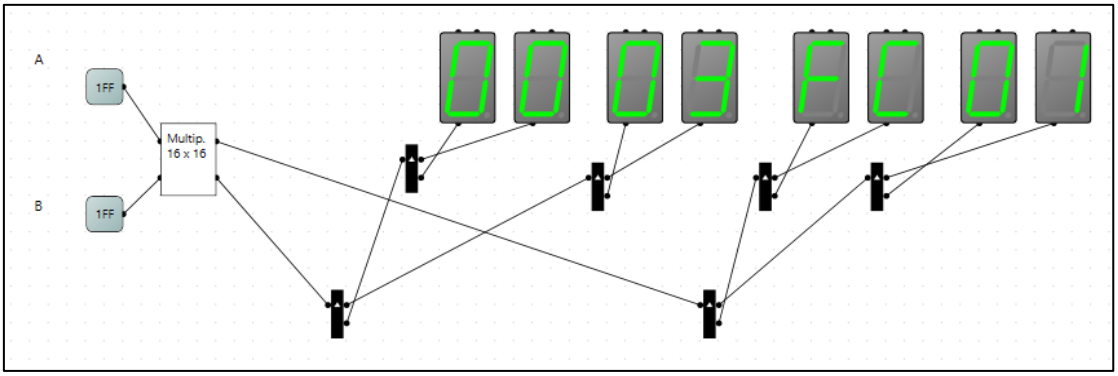

Figure 10. Test of 16x16 bits multiplier, by Appling ($0000\ 0001\ 1111\ 1110_2$x$0000\ 0001\ 1111\ 1111_2$), the result is ($0000\ 0000\ 0000\ 0011\ 1111\ 1100\ 0000\ 0001_2$) ($0003FC01_{16}$).

2.4. Design of General n x n Bits Multiplier

Let us assume we have the two binary numbers A, and B with maximum bit length of 2^n.

$$A = \sum_{j=0}^{2^n-1} a_j\, 2^j \quad , \quad B = \sum_{j=0}^{2^n-1} b_j\, 2^j$$

Then if we want to evaluate C = A x B then $C = \sum_{j=0}^{2^{(n+1)}-1} c_j\, 2^j$

So the maximum length of C will be 2^n

The hardware circuit should be able to accurately evaluate the bits of C given the bits of A and B. Each number like A can be separated into two parts with the same length, High part A_H and Low part A_L. Each has a maximum length of $2^{(n-1)}$ bits such that:

$$A = 2^{2^{(n-1)}} A_H + A_L$$
$$A = \sum_{j=2^{(n-1)}}^{2^n-1} a_j\, 2^j + \sum_{j=0}^{2^{(n-1)}-1} a_j\, 2^j = 2^{2^{(n-1)}} \sum_{j=0}^{2^{(n-1)}-1} a_{2^{(n-1)}+j}\, 2^j + \sum_{j=0}^{2^{(n-1)}-1} a_j\, 2^j$$

By comparing the two above equations we can conclude that

$$A_H = \sum_{j=0}^{2^{(n-1)}-1} a_{2^{(n-1)}+j}\, 2^j \text{ and } A_L = \sum_{j=0}^{2^{(n-1)}-1} a_j\, 2^j$$
$$A_H = a_{2^{(n)}-1} \dots\dots a_{2^{(n-1)}} \text{ and } A_L = a_{2^{(n-1)}-1} \dots\dots a_0$$

Similarly, we can express B also as two parts

$$B = 2^{2^{(n-1)}} B_H + B_L$$

Now the result of multiplication C = A B can be express in terms of 2 parts and then separate each part in High and Low also, so we end up with 4 parts:

$$C = 2^{2^{(n)}} C_H + C_L = 2^{[2^{(n)}+2^{(n-1)}]} C_{HH} + 2^{2^{(n)}} C_{HL} + 2^{2^{(n-1)}} C_{LH} + C_{LL}$$

Now we can evaluate each part of the output C separately in terms of the parts of the inputs A and B

$$C = A\,B = \left(2^{2^{(n-1)}} A_H + A_L\right)\left(2^{2^{(n-1)}} B_H + B_L\right)$$
$$C = 2^{2^{(n)}} A_H B_H + 2^{2^{(n-1)}} A_H B_L + 2^{2^{(n-1)}} A_L B_H + A_L B_L$$

Now each multiplication term can be further separated into Low and High parts with the length of $2^{(n-1)}$ bits:

$$A_L B_L = (A_L B_L)_L + 2^{2^{(n-1)}} (A_L B_L)_H$$
$$A_L B_H = (A_L B_H)_L + 2^{2^{(n-1)}} (A_L B_H)_H$$
$$A_H B_L = (A_H B_L)_L + 2^{2^{(n-1)}} (A_H B_L)_H$$
$$A_H B_H = (A_H B_H)_L + 2^{2^{(n-1)}} (A_H B_H)_H$$

$$C = 2^{2^{(n)}}\left[(A_H B_H)_L + 2^{2^{(n-1)}} (A_H B_H)_H\right] + 2^{2^{(n-1)}}\left[(A_H B_L)_L + 2^{2^{(n-1)}} (A_H B_L)_H\right] + 2^{2^{(n-1)}}\left[(A_L B_H)_L + 2^{2^{(n-1)}} (A_L B_H)_H\right] + \left[(A_L B_L)_L + 2^{2^{(n-1)}} (A_L B_L)_H\right]$$

$$C = \left[2^{2^{(n)}} (A_H B_H)_L + 2^{2^{(n)}+2^{(n-1)}} (A_H B_H)_H\right] + \left[2^{2^{(n-1)}} (A_H B_L)_L + 2^{2^{(n)}} (A_H B_L)_H\right] + \left[2^{2^{(n-1)}} (A_L B_H)_L + 2^{2^{(n)}} (A_L B_H)_H\right] + \left[(A_L B_L)_L + 2^{2^{(n-1)}} (A_L B_L)_H\right]$$

$$C = [(A_L B_L)_L] + \left[2^{2^{(n-1)}} (A_L B_H)_L + 2^{2^{(n-1)}} (A_H B_L)_L + 2^{2^{(n-1)}} (A_L B_L)_H\right] + \left[2^{2^{(n)}} (A_H B_L)_H + 2^{2^{(n)}} (A_L B_H)_H + 2^{2^{(n)}} (A_H B_H)_L\right] + \left[2^{2^{(n)}+2^{(n-1)}} (A_H B_H)_H\right]$$

$$C = [(A_L B_L)_L] + 2^{2^{(n-1)}}\,[((A_H B_L)_L + (A_L B_L)_H) + (A_L B_H)_L] + 2^{2^{(n)}}\,[((A_H B_L)_H + (A_L B_H)_H) + (A_H B_H)_L] + 2^{2^{(n)}+2^{(n-1)}}\,[(A_H B_H)_H]$$

By comparing with the equation

$$C = C_{LL} + 2^{2^{(n-1)}} C_{LH} + 2^{2^{(n)}} C_{HL} + 2^{[2^{(n)}+2^{(n-1)}]} C_{HH}$$

We can conclude that

$$C_{LL} = [(A_L B_L)_L]$$
$$C_{LH} = ((A_H B_L)_L + (A_L B_L)_H) + (A_L B_H)_L$$
$$C_{HL} = ((A_H B_L)_H + (A_L B_H)_H) + (A_H B_H)_L + Carry\{C_{LH}\}$$
$$C_{HH} = (A_H B_H)_H + Carry\{C_{HL}\}$$

So we can build the 2^n x 2^n utilizing four 2^{n-1} x 2^{n-1} multiplier circuits and five 2^{n-1} bits Full Adders, as shown in Figure 11.

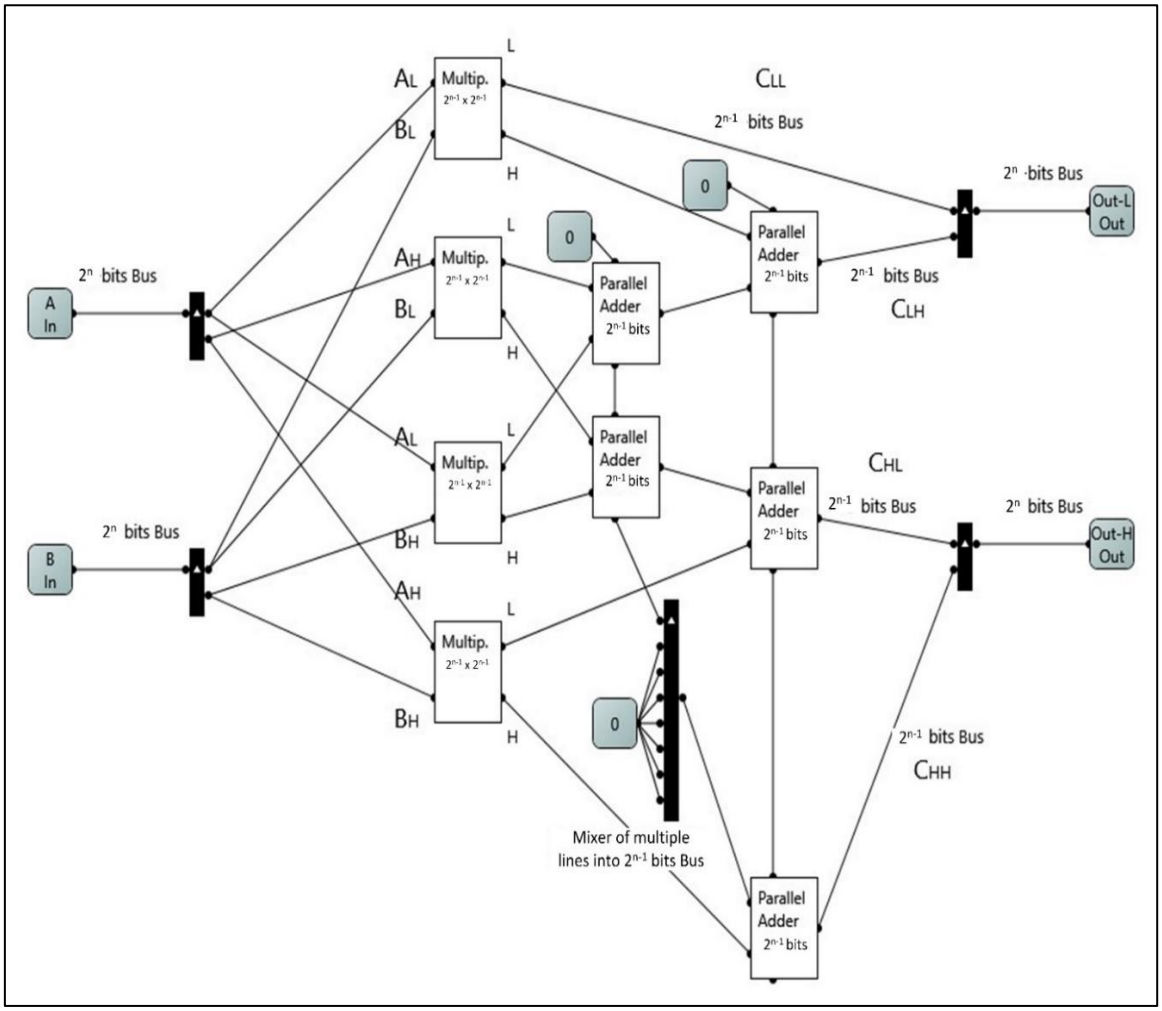

Figure 11. General 2^n x 2^n Hardware Multiplier Circuit.

3. Conclusion

This paper discussed the designing, analyzing and testing of general binary multiplier with bits' length of 2^n. the methodology of this new hardware design has been depends on building of a new basic 2x2 bits multiplier then based on this circuit proceed to build the more bits' length multiplier circuits. All designed circuits have been mathematically analyzed and then tested.

References

[1] Jhamb, Mansi, and Himanshu Lohani. "Design, implementation and performance comparison of multiplier topologies in power-delay space." *Engineering Science and Technology, an International Journal* 19. 1 (2016): 355-363.

[2] M. Moradi, R. F. Mirzaee and K. Navi, "Ternary Versus Binary Multiplication with Current-Mode CNTFET-Based K-Valued Converters," *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*, 2016, pp. 17-22, doi: 10.1109/ISMVL.2016.19.

[3] M. E. Paramasivam and R. S. Sabeenian, "An efficient bit reduction binary multiplication algorithm using vedic methods," *2010 IEEE 2nd International Advance Computing Conference (IACC)*, 2010, pp. 25-28, doi: 10.1109/IADCC.2010.5423043.

[4] P. Patel, A. Shandilya, N. Brahmbhatt, K. Raval and D. Deb, "Vedic and conventional methods of N × N Binary Multiplication with hardware implementation," *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, 2015, pp. 1-6, doi: 10.1109/SMARTSENS.2015.7873592.

[5] D. Kim and J. Jung, "Improved Multiplication Algorithm by Clearing Leading Zeros of Binary Numbers based on Big Data Analysis," *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2018, pp. 163-168, doi: 10.1109/SNPD.2018.8441151.

[6] M. P. Véstias and H. C. Neto, "Iterative decimal multiplication using binary arithmetic," *2011 VII Southern Conference on Programmable Logic (SPL)*, 2011, pp. 257-262, doi: 10.1109/SPL.2011.5782658.

[7] Krishnaveni, D., and T. G. Umarani. "VLSI implementation of vedic multiplier with reduced delay." *International Journal of Advanced Technology & Engineering Research (IJATER*) 2. 4 (2012): 10-14.

[8] Ahmed Salah Hameed and Marwa Jawad Kathem "Design and implementation of a fast sequential multiplier based on iterative addition architecture", *IOP Conf. Ser.: Mater. Sci. Eng.* 1076 (2021) 012040.

[9] Lee, Shing Jie, and Siti Hawa Ruslan. "A 2x2 Bit Multiplier Using Hybrid 13T Full Adder with Vedic Mathematics Method." *International Journal of Integrated Engineering* 10. 3 (2018).

[10] Floyd, Thomas L. Digital Fundamentals, 10/e. Pearson Education India, 2010.