Comparative Study on Different Type of Multiplier in Verilog HDL

Nik Ahmad Afnan Bin Nik Azman
Department of Electronic Engineering
University Teknologi MARA,
40450 Shah Alam, Selangor Darul Ehsan, Malaysia
E-mail: afnan17595@gmail.com

Abstract— **This paper presents the comparative study on different type of multiplier in Verilog HDL. Multiplier is one of the most important components in digital design system and embedded applications. This research study on the different type of multiplier on the algorithm, implementation on verilog and performance analysis. Today researches already creates so many type of multiplier. This paper helps by doing comparative study of some of this multiplier for future references. Three type of multiplier used in this comparative study, that are Array, Vedic and Wallace with variation of 4-bits and 8-bits. This technical paper deals with design, synthesis and simulation using Quartus II and Modelsim. The performance of the multipliers is based on the power and area. Quartus II used to check the wire connectivity in logic and module of the design. To check the validity and functionality of the multiplier, Modelsim software are used. The same input data is used on each multiplier because it is expected to get the same output. In this study, it shows that the performance of multiplier are depending on it algorithm. The algorithm of multiplier help the design become more better in performance of power and area when the number of bits changed.**

Keywords—Binary Array Multiplier, Vedic Multiplier, Wallace Multiplier, 2bit, 4bit and 8bit

I. INTRODUCTION

Multiplier is one of the technology that have high request in Digital Signal Processing (DSP). There are many DSP applications that use multiplier for better performances.. As the sophistication of the technology the rate of modifying the multiplier also increased. There are many researchers already creating better performance multiplier. Some researchers creating new multiplier by modifying the existing multiplier to increase the performance. Reducing the time delay and power consumption are very essential requirements for many applications.

Computerized multipliers are the most normally utilized segments in any advanced circuit outline. They are quick, dependable and productive segments that are used to actualize any task. Contingent on the course of action of the segments, there are diverse kinds of multipliers accessible. Specific multiplier engineering is picked in view of the application [2].

.The multiplication operation consists of producing partial products and then adding these partial products and the final products is obtained. In this way the speed of the multiplier depends on the number of partial products and the speed of the adder and any logic with different algorithm

The comparison of 4-bit and 8-bit Array multiplier ,Vedic multiplier and Wallace multiplier shows the best design to use in DSP. The performance analysis are on area and power. The design implemented in Verilog HDL and tested in Quartus II for RTL circuit and Modelsim to check the multiplier operation without any error.

II. LITEREATURE REVIEW

A. Array multiplier

The basic structure of an array multiplier consists of AND gates, full adders and half adders. The method use for multiplication in this design is called partial product accumulation because rows of linked adders generate the accumulated partial products that would evolve from a manual multiplication of the data words [2].

By using Array Multiplier, the number of components required is optimized, but delay for this multiplier is larger. The delay can be noted by refer to the time constraint for worst delay because of the path traversed in the array [1][3]. Due to the usage of large number of gates the hardware becomes complex and area is increased [1].

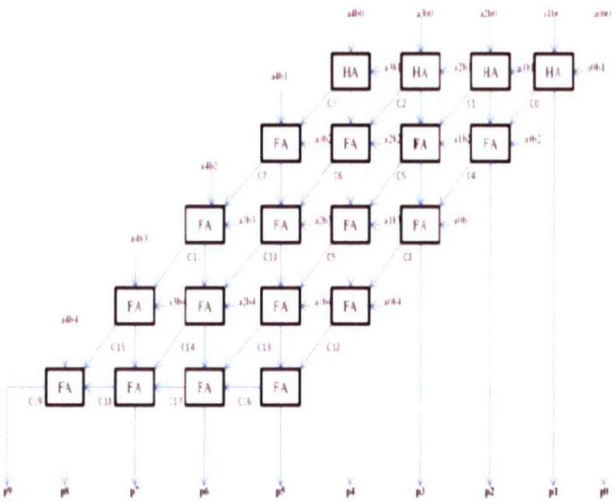

Figure 1.0 Example of Array multiplier design [4] .

B. Urdhava Tiryakbhyam (Vedic) multiplier

The term 'Veda' implies storehouse of information. Vedic Mathematics is an ancient type of arithmetic reproduced from ancient Indian sacred texts referred to as Vedas. It depends on 16 sutras which execute distinctive branches of science like variable based math, geometry, arithmetic. [6]

Vedic multiplication is an algorithm that can solves a wide range of mathematical problems with the appropriate usage of the aphorisms. The fact that this methods are really fast when compared to the conventional mathematical methods that has led to their application in processors to increase their computational speed. An added advantage is that they also require less hardware which can be attributed to the symmetric computation evident in this technique.

The algorithm of the multiplication of this multiplier is based on *Urdhva Tiryakbhyam* which literally means "Vertically and Crosswise", is a general multiplication formula applicable to all cases of multiplications [2]. Figure below showing one of the Vedic multiplier.

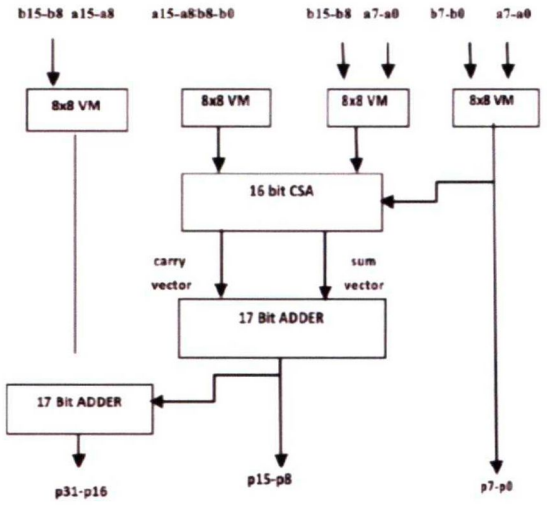

Fig 1.1 Example of Vedic multiplier design [2].

C. Wallace multiplier

Luigi WALLACE, the computer scientist has invented the WALLACE hardware multiplier during 1965. WALLACE multiplier is extracted form of parallel multiplier [7] It is slightly faster and requires fewer gates. Different types of schemes are used in parallel multiplier.

The Wallace tree multiplier is considered as speedier than a less difficult array multiplier and is an effective usage of an advanced circuit which is multiplies two integers. A Wallace multiplier is a parallel multiplier which uses the carry save addition algorithm to reduce the latency [5].

This is accomplished by utilizing full what's more, half adders to reduce the quantity of lines in the matrix number of bits at every summation stage. Despite the fact that the WALLACE increase has general and less intricate structure, the procedure is slower in way because of serial multiplication process [7].

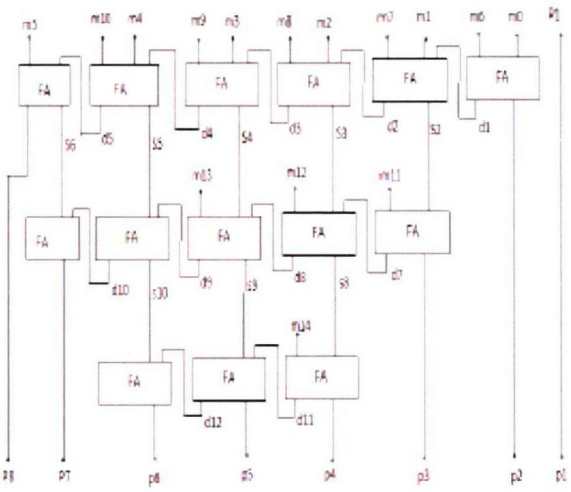

Figure 1.2 Example of Wallace multiplier design [5].

III. ALGORITHM OF MULTIPLIER

In this section, the algorithm of each multiplier will be presented. Each of the algorithm have their own way which make them easy to solve any multiplication. In digital design system the multiplication are used for computer processing design to speed up the performance of any processor. The algorithm of each multiplication will affect the number of logic gate in any design of multiplier. The algorithm below will be explained using simple calculation.

A. Array algorithm

The Array algorithm are the basic multiplying decimal numbers based on calculating the partial products, shifting them to the left and then adding them together. In this case, the basic multiplying are used for binary number to get the partial products [8]. The fact is most processor today used binary numbers. It is because by using binary numbers. each long number is multiplied by one digit (either 0 or 1) make them easier than in decimal as the product is 0 or 1 or the same number.

```
    1101  ( equal to 13 in decimal )
  X 1110  ( equal to 14 in decimal )
========
    0000  ( this is 1101 x 0)
   1101   ( this is 1101 x1, shifted left one position)
  1101    ( this is 1101 x1, shifted left two positions)
 1101     ( this is 1101 x1, shifted left three positions)
========
10110110  ( equal to 182 in decimal)
```

Fig 1.3:Example of simple multiplication using binary number.

The simple multiplication above show the algorithm of how the Array multiplication work. The understanding of this multiplier actually very easy but to convert it to verilog it may took time to finish it because of the complexity of the design that using many adder and AND gate.

B. Vedic Urdhva Tiryakbhyam algorithm

The Vedic multiplication actually have many type of mathematic algorithm but for this multiplier, it using algorithm that are consist of urdhvak (vertical multiplication) and tiryak (crosswise multiplication) operation. As for the multiplication this method are very useful to solve any complicated mathematical procedures.

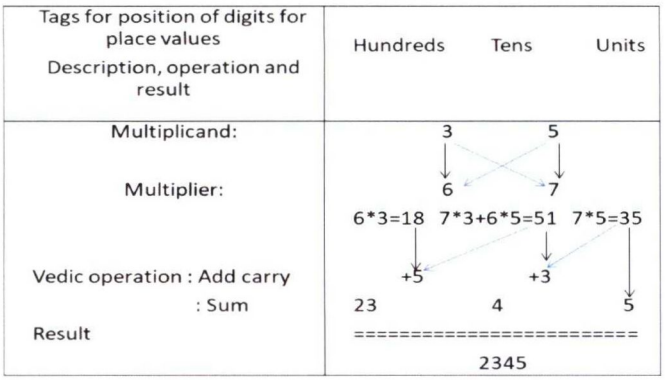

Fig 1.4: Example of vedic multiplication.

From example above the operation of multiplication show that the first operation are 5 * 7 = 35 then the carry = 3 added to next step, the number left = 5. Next is (7*3)+(6*5) = 51 the carry = 5 added to next step, the number left = 1+ 3(carry from last step) = 4. Last step, 6*3 = 18 there is no carry anymore, the number left = 18 + 5(carry from last step) = 23. So total from multiplication will equal to 2345 which make it same answer when using any calculator.

C. Wallace tree algorithm

Wallace multiplier is one among the proficient tree multipliers which are as often as possible utilized as a part of microchip circuits. Wallace multipliers work on an alternate calculation dissimilar to traditional Array multiplier. Wallace Multiplier is usually utilized where a productive deferral is significant necessity [6]. Wallace Multiplier calculation for 4-bit is specified beneath.

For a 4 bit multiplication firstly, the partial products are obtained. Group the first three rows of partial products and add them together by using adders. The carry generated by the adders in each column is rippled to preceding column as shown in fig 1.5. The outputs of these adders in the first stage are added with the remaining rows of partial products.

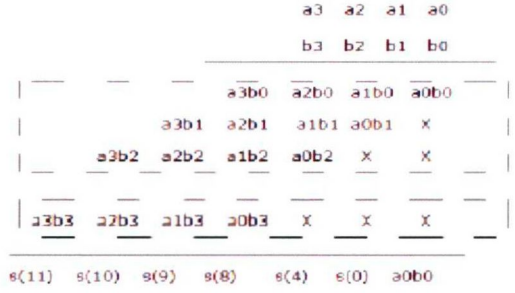

Fig 1.5: 4- bit Wallace multiplication.

The dot representation of Wallace multiplier is appeared in fig 1.6 where '.' shows fractional items and 'o' demonstrates carry created by the addition of partial products. Advance this multiplier is executed by utilizing different adder keeping in mind to minimize the memory and delay [6].

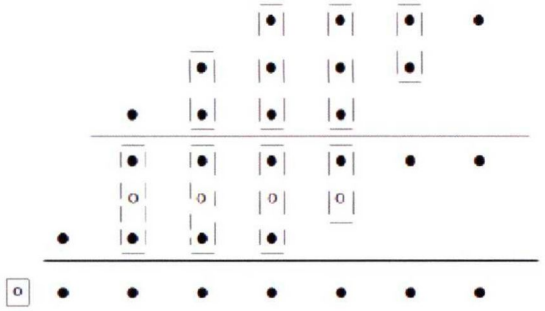

Fig 1.6 : Present the dot of 4-bit Wallace multiplication.

IV. METHODOLOGY

There are many algorithms for multiplier design, but for this study, Array, Vedic and Wallace multiplier are choosen. By studying the algorithms of each multiplier it is easier to predict the output. Manual calculation are also done to prove the functionality of the multipliers. The manual calculation of the multiplication of each multiplier will be compared to the result of the design after simulation of the test bench using Modelsim software. Study on multiplier algorithm also used to create a truth table for the multiplier design. In the truth table, the input and output are write based on the test bench code.

The multipliers are coded by using the verilog HDL format. The coding of the multiplier are compiled using Quartus II. In compilation, if the coding cannot be compiled, the code must be recheck and compile again until the compilation did not display any error. Furthermore, the Quartus II software also show the RTL circuit of the design. RTL circuit used to check whether the wire of the logic gate in the design connected or not. If there is a wire not connected to any logic gate or module, the coding must be change until the desired RTL circuit displayed. The performance analysis are also carried out using Quartus II.

Power reports on the design can be simulate using the Power Play Power Analyze Tool in Quartus II software. The power reports are determine by the Total Thermal Power Dissipation in miliwatt (mw). For area, the report are shown in Compile Design under Fitter (Place & Route). The area reports are determine by the Total logic elements.

The Modelsim software are used to simulate the test bench of the design to check the functionality of the multipliers. The results from Modelsim are compared with the truth table to verified the output. The test bench also need to be coded. If the simulation of the test bench have an error, the simulation need to be done again by repairing the coding until

there is no error occurred. The result of the Modelsim are shown in waveform. To check the functionality of the design, the test bench coding must be same format with the design. If not, error will occurred. The test bench are coded accordingly to the truth table. For example, the 4-bit multiplier 1st input set as 1000 (8) and 2nd input set as 1010 (10), after simulation the output is equal to 0101000 (80). The result should be same with the truth table. If not, there must be an error on the design or the test bench code.

The performance are referred to the report power and area of the multipliers. The lower the power dissipation the better the multiplier and for area, the lower the uses of total logic element the better the multiplier. The fig 1.7 below show the flow chart of this research.

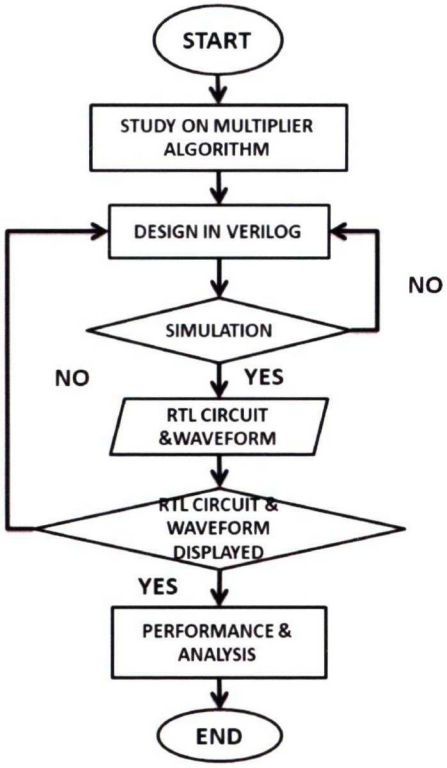

Fig 1.7 : Flow chart of the research.

V. Result and Discussion

The result are based on the output of each multiplier with different number of bit and based on the performance of the multiplier by referring to the power and area of the multiplier.

A. Output Waveform of multiplier

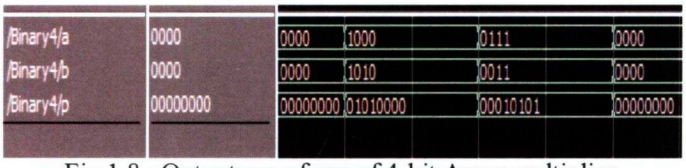

Fig 1.8 : Output waveform of 4-bit Array multiplier.

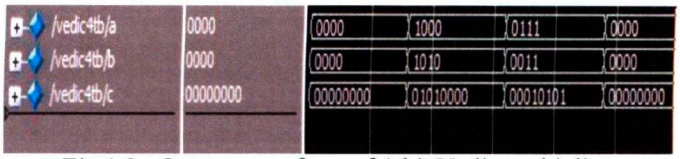

Fig 1.9 : Output waveform of 4-bit Vedic multiplier.

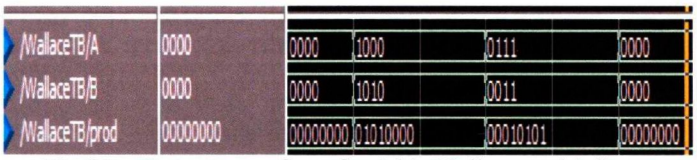

Fig 2.0 : Output waveform for 4-bit Wallace multiplier.

Fig 2.1: Output waveform for 8-bit Array multiplier

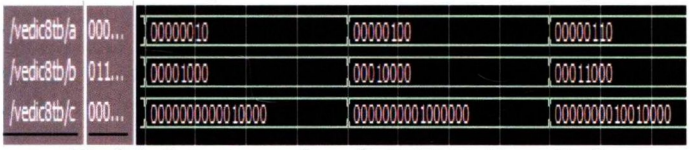

Fig 2.2: Output waveform for 8-bit Vedic multiplier

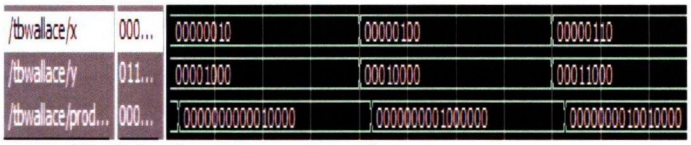

Fig 2.3 : Output waveform for 8-bit Wallace multiplier

The figure 1.8, 1.9, 2.0, 2.1, 2.2 and 2.3 above show the output waveform of each multipliers after done test benh in Modelsim. Inputs of multipliers are set with the same value , in 4-bit multiplier the 1st wave, *input a* is equal to 1000 (8), *input b* is equal to 1010, then the 2nd wave, *input a* equal to 0111 and *input b* equal to 0011. For the 8-bit multiplier, the *input a* are set by increment of 00000010 and *input b* set by increment of 00001000.

The multiplication of each multiplier shows the same output if the multiplier doesn't have any error in the design. So from this the result, the purpose of the test bench actually to check whether the multiplier are functioning without error and to check the coding of the design have issue with the increasing of the number of bit on output . Sometimes error like no output and the design undetected can occurred while simulating the test bench. The truth table below shows the input and output of the waveform.

Table 1: Truth table 4-bit multiplier

Number of wave for 4-bit multiplier	Input A	Input B	Output
1st wave	1000	1010	01010000
2nd wave	0111	0011	00010101
3rd wave	0000	0000	00000000

Table 2 : Truth table 8-bit multiplier

Number of wave for 8-bit multiplier	Input A	Input B	Output
1st wave	00000010	00001000	0000000000010000
2nd wave	00000100	00010000	0000000100000000
3rd wave	00000110	00011000	0000000010010000

The truth table from table 1 and table 2 above shows the expected result for the test bench of the 4-bit multiplier and 8-bit multiplier. It can be seen that all of the output from figure above are all the same with the truth table. So from this the design, the three different type 4-bit and 8-bit multiplier are valid. There will be no output waveform if the design cannot be compile or undetected by the Modelsim. The output bits increases cause by adder in the designs. For example, when 0000 multiply with 0000, the output will be 00000000.

A. Power and area

For this section the result are based on the performance of the multiplier by referring to the report power and area using Quartus II software.

Table 3: Report power of multiplier

Number of bit	*Multiplier*	*Power (mw)*
4bit	Binary Array	**64.11**
	Vedic	**120.29**
	Wallace	**135.06**
8bit	Binary Array	**142.05**
	Vedic	**142.04**
	Wallace	**137.01**

Table 4: Report area of multiplier

Number of bit	*Multiplier*	*Area(total logic element)*
4bit	Binary Array	**32 / 114,480 (< 1 %)**
	Vedic	**32 / 114,480 (< 1 %)**
	Wallace	**35 / 114,480 (< 1 %)**
8bit	Binary Array	**166 / 114,480 (< 1 %)**
	Vedic	**160 / 114,480 (< 1 %)**
	Wallace	**117/ 114,480 (< 1 %)**

The result from table 3 and 4 are collected by using different type multiplier using 4-bit and 8-bit input. For Binary Array multiplier in 4-bit, the power are the lowest compare to Vedic and Wallace. This showing that the binary array are the best when using the 4-bit multiplier. The area of the multiplier showing that the 4-bit multiplier of Binary Array and Vedic multiplier are the same equal to 32 total logic element except Wallace multiplier that have 35 total logic element make it the worst multiplier.

In 8-bit multiplier the result are contrast than the 4-bit multiplier, it can be see that the Wallace are the best among the others multiplier. Wallace multiplier have lowest power and area compare to Vedic and Array multiplier. In this case, the Array and Vedic have almost similar result. The total logic element of Array equal to 166 and Vedic equal to 160 also the power both multiplier equal to 142.05 mw and 142.04 mw.

Increasing the bit of the multiplier might change the result and the performance of the design because the coding of the multiplier become more complicated at higher bits. The result show that the Wallace is the best when it increasing the bit. The area and power suggest the best design for 4-bit is the Arrray multiplier while for 8-bit design, the results shows the best design is Wallace. Due to increasing the bit the algorithm of the Wallace multiplier help it to be better at power and area compared to 4-bit.

CONCLUSION

The comparative study of Array, Vedic and Wallace multiplier shows that the performance of the multiplier is based on their algorithm. The conclusion is the algorithm of multiplier help the design become more better in performance of power and area when the number of bits changed. The result from the project showing that Wallace have good performance for 8-bit multiplier and binary Array for 4-bit multiplier. In the end, the multiplier will be implemented in DSP, but without the direct comparison of the multiplier the DSP cannot be optimize by the performance of the multiplier.

REFERENCES

[1] R. Dhanabal, V. Bharathi, N. Anand, G. Joseph, S. S. Oommen, and S. K. Sahoo, "Comparison of existing multipliers and proposal of a new design for optimized performance," *Int. J. Eng. Technol.*, vol. 5, no. 2, pp. 1704–1709, 2013.

[2] M. H. Ali and A. K. Sahani, "Study, Implementation and Comparison of Different Multipliers based on Array, KCM and Vedic Mathematics Using EDA Tools - ijsrp-p18135.pdf," vol. 3, no. 6, pp. 1–8, 2013.

[3] M. Hatamian, D. Hung, Z. Kurmas, J. Frenzel, J. Pinter-Lucke, and P. Zhao, *In Praise of Digital Design and Computer Architecture*. 2016.

[4] P. V Rampur, M. Jagadish, and G. Yogeesha, "Design and Implementation of Advanced Array Multiplier for Binary Multiplication on FPGA," no. August, pp. 2–5, 2016.

[5] I. American and I. Technical, "Design and Implementation of Wallace Tree Multiplier using Higher Order Compressors," vol. 04, no. 06, pp. 442–448, 2016.

[6] A. Kant and S. Sharma, "Applications of vedic multiplier designs - A review," *2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015*, 2015.

[7] G. C. Ram, D. S. Rani, R. Balasaikesava, and K. B. Sindhuri, "Design of delay efficient modified 16 bit wallace multiplier," *2016 IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2016 - Proc.*, pp. 1887–1891, 2017.

[8] A. Khatibzadehl, K. Raahemifar, and M. Ahamdi, "a Novel Multiplier for High-Speed Applications," *Power*, pp. 305–308, 2005.