DESIGN OF LOW POWER AND LOW AREA MULTIPLIERS BY EVADING WASTAGE OF WASTAGE OF ENERGY

[1]**Dr.J.Sudhakar ,** Professor , [5]Department of Electronics and Communication Engineering , Vignan's Institute of Engineering for Women,Visakapatnam.
[2]**Mandapaka Haritha Ganga Bhavani**,[3]**Bheemisetti Kusuma Naga Lakshmi**,[4]**Pedapalli Srikavya Satya Sushma**,[5]**Buddha Aswini**
[2,3,4,5]Department of Electronics and Communication engineering, Vignan's Institute of Engineering for Women, Visakhapatnam

Abstract

Nowadays , low power and smaller area the most important criterion in fabrication of high performance systems and Digital Signal Processing since multiplication is an essential arithmetic operation for these applications.To reduce the power consumption number of operations are to be reduced.The multiplier mainly focuses on the four aspects to form an efficient multiplier, i.e., speed, power consumption, area, and accuracy. Inorder to achieve low power and smaller area this study implements the parallel multipliers such as Array multiplier,Carry-Save multiplier,Wallace Tree multiplier,Baugh-Wooley multiplier,Bypassing Multipliers and Adiabatic Multipliers and their comparitive analysis.This comparitive helps us to select one of the most suitable multiplier for a particular application.Using these multipliers power,area and speed could be simulated using Xilinx Vivado 2018.1 and finally these values could be analyzed.Among all these multipliers area and power consumption of Adiabatic Multipliers was smallest although array multiplier could be used for finest applications.

Keywords: Digital Signal Processing, **,**Partial Product,Multiply and Accumulate,Bypassing,Array,Carry-Save,Wallace,Baugh-Wooley,Booth.

1. Introduction

Multiplication is an important fundamental function in arithmetic operations.In fact,multiplication based operations such as Multiply and Accumulate(MAC) and inner product are some among the frequently used computation- intensive arithmetic functions currently implemented in many Digital Signal Processing(DSP) applications (such as convolution ,Fast Fourier Transform(FFT),filtering and others)**.**They usually contribute significantly to the time delay and take up a great deal of silicon area in the DSP system.Since multiplication dominates the execution time of most DSP algorithms,using a high speed multiplier is very desirable . Currently , multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.[1]

With an ever-increasing quest for greater computing power on battery-operated mobile devices,design emphasis has shifted from optimizing conventional delay time and area size to mininizing power dissipation while still maintaining high performance.A low-power design allows portable devices to operate longer with the same amount of battery charge.The subsequent sections present different types of parallel multipliers.The subsequent sections present different types of parallel multipliers.

2. Literature Review

Three important criteria to be considered in the design of multipliers are the chip area,speed of computation and power dissipation.Most advanced digital systems incorporate a parallel multiplication unit to carry out high-speed mathematical operations. Today,high-speed parallel multipliers with much larger areas and higher complexity are used extensively in RISC,DSP and graphics accelerators.Some examples of the parallel multipliers are the array multiplier,Baugh-wooley multiplier as well as tree multipliers like the Wallace Tree Multiplier.[2]

ISSN: 2278-4632
Vol-11 Issue-01 2021

2.1. Array Multiplier

An array multiplier is a digital circuit used for multiplication of two numbers by retaining an array of both full adders and half adders.It is having high degree of uniformity. This multiplier has been performed as a function of high speed and efficient area multiplier and it also involves AND of multiplier and multiplicand bits in the place of generation of partial products. Every partial product can be produced by interpreting multiplicand and single bit of multiplier for every period . The future addition operation is accepted by large speed carry-save algorithms and also, the concluding product is attained by retaining any faster adder and the amount of partial products depends on the number of multiplier bits.The design structure of the array Multiplier is regular, it is based on the add shift algorithm principle.[2]

A full adder has three input lines and two output lines, where we use this as a basic building block of an array multiplier where AND gates are used for the product, the summation is done using Full Adders and Half Adders where the partial product is shifted according to their bit orders.In an n*n bit array multiplier, n*n AND gates compute the partial products and the addition of partial products can be performed by using n* (n – 2) Full adders and n Half adders.

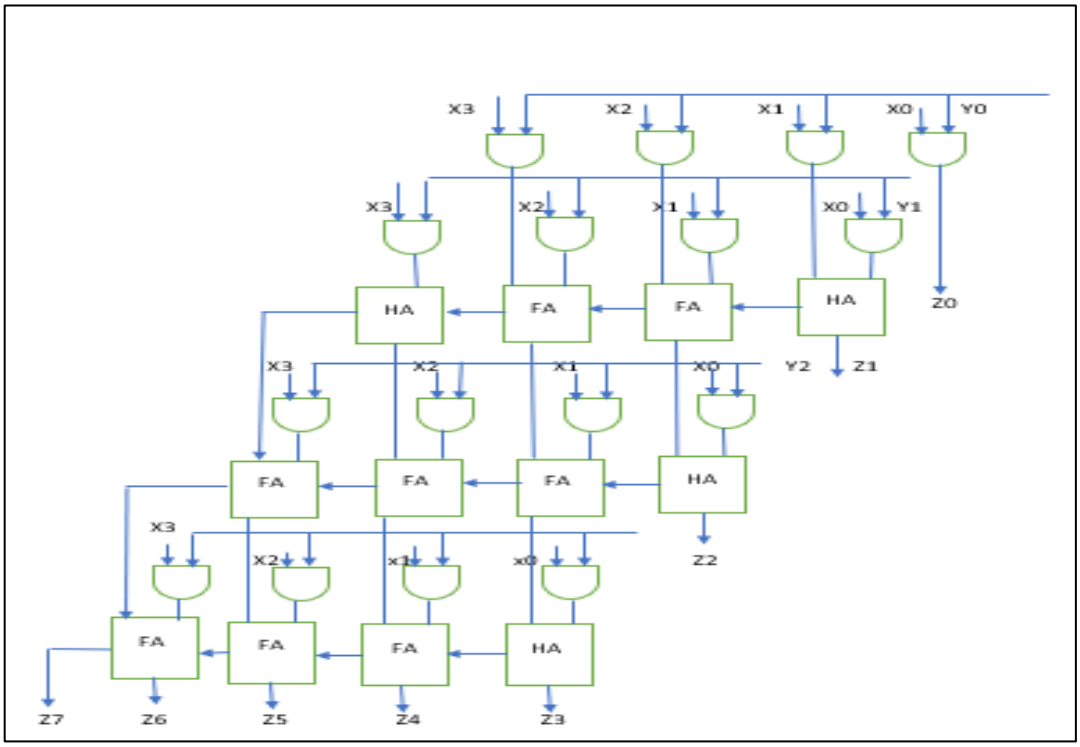

Figure 1: Schematic circuit of Array Multiplier

The **FIGURE 1** shows the architecture of a 4×4 array multiplier. Let X0,X1,X2,X3 be the multilicand bits and Y0,Y1,Y2,Y3 be the multiplier bits.The multiplication operation can be solved by finding partial products and then adding partial products together. Here the partial product can be obtained by multiplying the multiplicand with each bit of multiplier. The partial products are now shifted towards the left side on multiplication and they are added, this process is repeated until no two partial products exist for addition.Hence the final product generated is P7P6P5P4P3P2P1P0 where P7 is the MSB bit and P0 is the LSB bit.[3]

ISSN: 2278-4632
Vol-11 Issue-01 2021

2.2. Conventional Multiplier

An existing array multiplier is very much regular in its structure when compared to the conventional array multiplier and uses an only short wire that goes from one full adder cell to adjacent full adder cell.This multiplier saves carry for the next stage of addition hence also called Carry Save Array Multiplier[4].It consists of an array of AND gates and adders arranged in an iterative structure that does not require logic registers.This is also known as non-additive multiplier since it doesnot add an additional operand to the result of multiplication.A n*n bit multiplier requires n*(n-1) adders and n*n AND gates.One efficient implementation of this multiplier is the regular layout of the adder array as shown in figure 2.

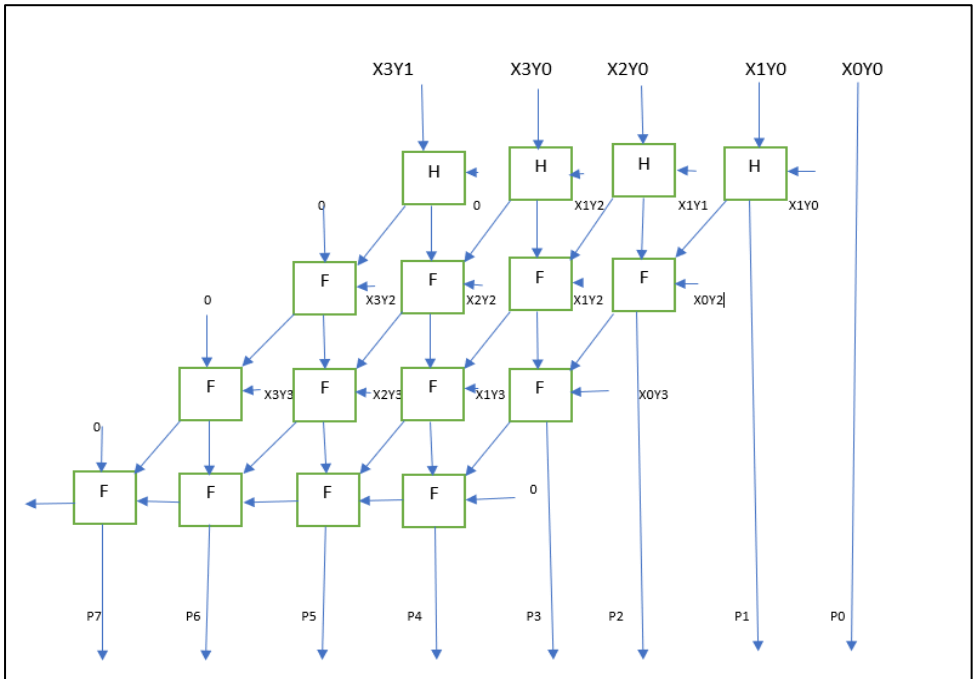

Figure 2: Schematic circuit of Conventional Multiplier

Each of the $X_i Y_j$ product bits is generated in parallel with the AND gates. Each partial product can be added to the previous sum of partial products by using a row of adders.The carry-out signals are shifted one bit to the left and are then added to the sums of the first adder and the new partial product.The shifting of the carry-out bits to the left is done by a Carry Save Adder (CSA).Instead,the respective carry bit is "saved" for the subsequent adder stage.This multiplier performs well for unsigned operands that are less than 16bits,in terms of speed,power and area.Besides,it has a simple and regular structure as compared to other multiplier schemes.However,the number of components required in building the conventional multiplier increases quadratically with the number of bits.This makes the multiplier inefficient and so it is rarely employed while handling large operands.Another pitfall of this multiplier is its potential susceptibility to glitching problems at the last of the full adders due to the exploitation of the Ripple Carry Adders(RCA)[5].

2.3. Wallace Tree Multiplier

For large operand multipliers such as 32-bit and above ,the partial products are longer than 16 bits and are considered unacceptably large.In this case, the performance of the conventional multiplier is degraded.The Wallace tree multiplication algorithm, however can reduce the number of partial products by employing multiple input compressors capable of accumulating several partial products concurrently.It

can handle the multiplication process for large operands.This is achieved by minimizing the number of partial product bits in a fast and efficient way by means of a CSA tree constructed from 1-bit full adders. This algorithm uses pseudo adders to add three inputs and then produces two outputs whose sum is equivalent to the sum of the three inputs.The main advantage of using pseudo adders comes from having the ability to ability to avoid carry propogation , which increases the speed of multiplication . The main disadvantage of the wallace tree algorithm is that the architecture exhibits some irregularities in the layout because it has arelatively complicated interconnection scheme.[6]

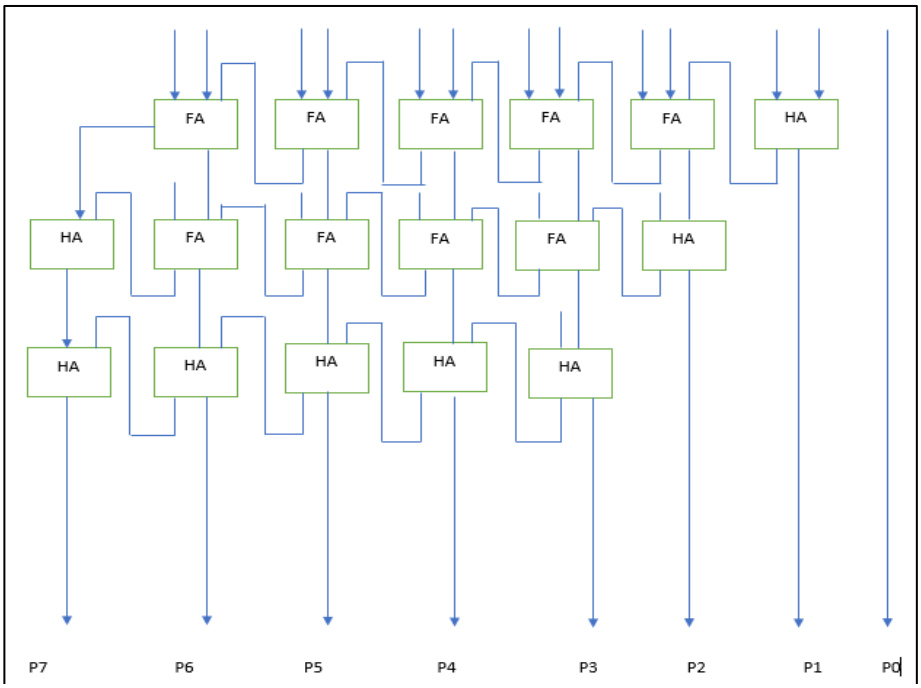

Figure 3: Schematic circuit of Wallace Tree Multiplier

In general,its multiplication process can be summarized as follows:
1. After generating the partial products,a set of counters reduces the partial product matrix but it doesnot propogate the carries.
2. The resulting matrix is composed of the sums and carries of the counters.
3. Another set of counters then reduces this matrix and the whole process continues until a two-row matrix is generated.
4. The two rows get summed up with a final adder ,preferably by a carry propagate adder.This method takes advantage of the carry save architecture in order to avoid the carry propogation until the final adder. In this scheme, the number of levels is crucial since they determine the speed of the multiplier.

2.4. Bypassing Multipliers

Bypassing with reference to multiplier means turning of some columns or rows or both in the multiplier array whenever certain multiplier or multiplicand or both bits are zero. In normal array type multiplier we have an array of Carry Select Adders/Full adders. If one of the inputs is zero the sum of adder is nothing but the input other than zero. Instead of unnecessary addition of zero we can skip the addition and provide input to the next level In that condition it is beneficial to bypass the other input to the sum. Here addition operation is not required to derive the sum output. The power that is saved here is unnecessary addition with zeros.

Types of bypassing schemes are:
1.Row Bypassing Multiplier
2.Column Bypassing Multiplier

ISSN: 2278-4632
Vol-11 Issue-01 2021

2.4.1: Row Bypassing Multiplier

Row bypassing technique is based on number of zeros in the multiplier bits. In this multiplier operation, some rows of adders in the basic multiplier array are disable during operation to save the power.

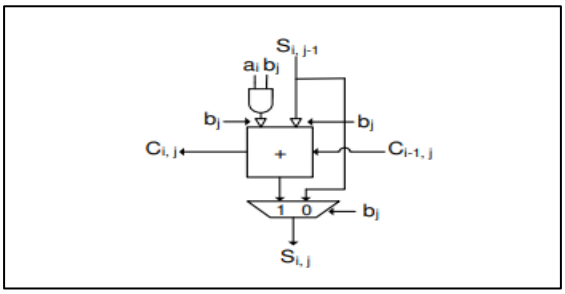

Figure 4: Modified Ripple Carry Adder

It consists of the rows of the ripple carry based full adder cells .Multiplexer is connected at output side of sum and carry to switch between bypassed path and normal path. Consider a multiplier with multiplier bits B and multiplicand bits A .A simple thought to improve performance is, as soon as bj was found to be zero , ie all partial products are zero, complete row is bypassed to avoid triggering those adding unit in the row to reduce power reduction.

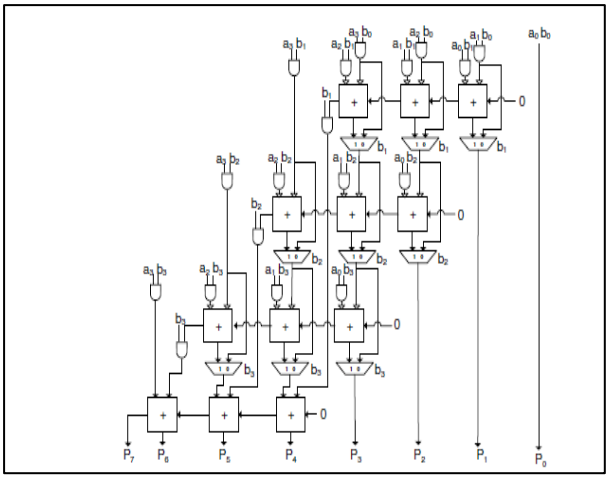

Figure 5: Schematic circuit of Row Bypassing Multiplier

To eliminate the redundant signal transitions, disable the adders whose partial product is zero, while shifting and bypassing the partial product of the previous adder rows to the next row of adder . The (j-1)th row of adders are bypassed to (j)th row if the corresponding bit in the multiplier is zero. This operation is performed by disabling the adder under the control of multiplier bit bj. Drawback of row bypassing multiplier is the requirement of additional correcting circuitry. For example, let b2 be zero. In this case, Ripple Carry Adder in the second row will be bypassed, and the outputs from the first row are fed directly to the third row of Ripple Carry Adder[7]. However, since rightmost full adder in the second row is disabled, it does not execute the addition, and the output is not correct. In order to solve this problem, extra components such as AND gates are added. Row bypassing multiplier reduces the switching activity by bypassing the row in which the multiplier bit is 0. It requires more consumption. To overcome this we are using column bypassing multipler[8].

2.4.2: Column Bypassing Multiplier

Column bypassing technique is based on number of zeros in the multiplicand bits. In this multiplication operation, some columns of adders in the basic multiplier array are disabled during operation to save power.

ISSN: 2278-4632
Vol-11 Issue-01 2021

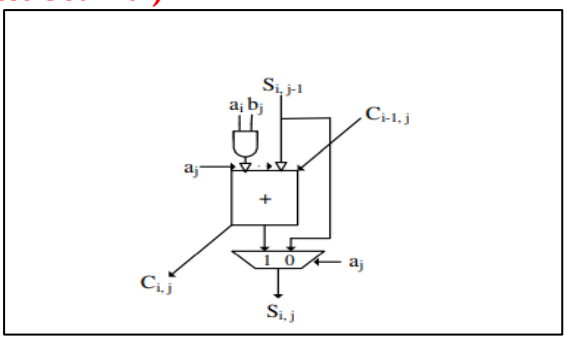

Figure 6: Modified Carry Save adder

This multiplier consists of rows of carry save adders. Multiplexer is connected at output side of sum to switch between bypassed path and normal path. The major focus of this multiplier is to reduce the switching transitions required to perform the computations[7].

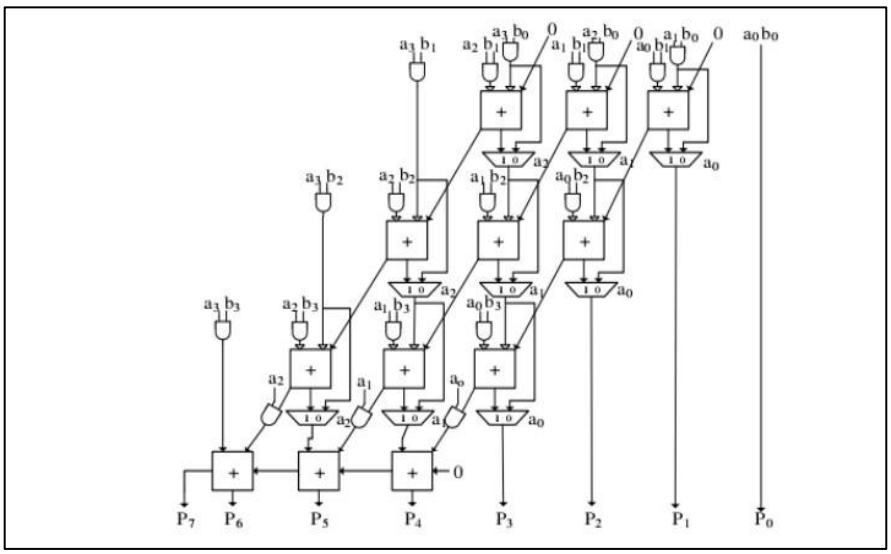

Figure 7: Schematic circuit of ColumnBypassing Multiplier

In column bypassing multiplier, if the bit A in the multiplicand is zero then addition operation in ith column can be bypassed for power reduction. The addition operation in (i-1)th column can be bypassed to (i)th column if the corresponding bit in the multiplicand is zero. This operation is performed by disabling the adder under the control of multiplicand bit ai. Column bypassing multiplier uses carry save adder hence speed of the operation increases when compared to Row Bypassing multiplier and consumes less power. The modified Full adder is simpler than that used in row bypassing multiplier[8]. Column bypassing multiplier requires extra correcting circuit . The above multipliers are restricted to unsigned binary numbers.

3.Materials and methods

As low power consumption and smaller area is an important issue to reduce both power consumption and area Baugh-Wooley and Booth Multiplier are introduced.These multipliers are applicable to both signed and unsigned multiplication.

3.1. Baugh-Wooley Multiplier

Baugh-Wooley multiplier is an enhanced version of the conventional multiplier. It is designed to cater to multiplication of both signed and unsigned operands ,which are represented in the 2's complement number system.The partial products are adjusted so that the negative signs are moved to the last steps,which inturn maximize the regularity of the multiplication array.The architecture is also based on carry-save algorithm[3].

ISSN: 2278-4632
Vol-11 Issue-01 2021

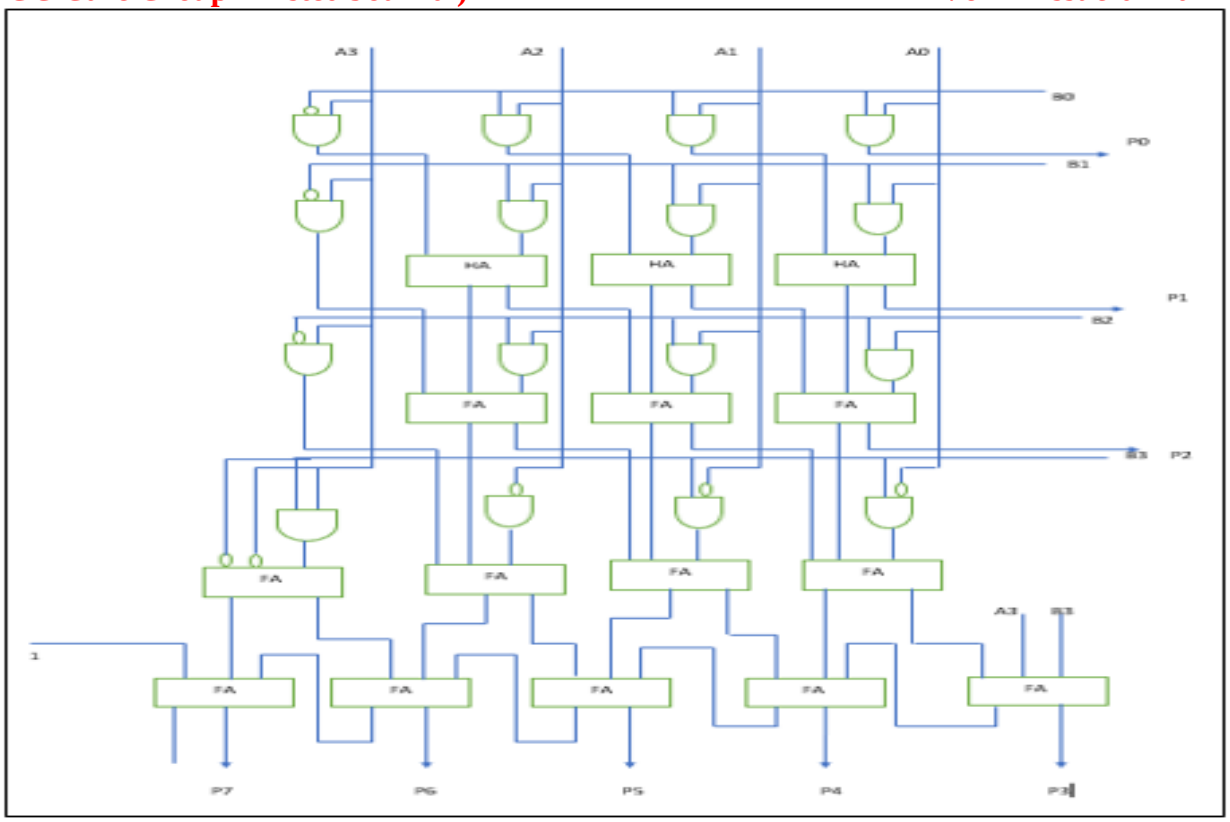

Figure 8. Schematic circuit of Baugh-Wooley Multiplier

Signed multiplicands must be first converted into their 2's complement representation before multiplication[9]. It is basically a combinational circuit that will either pass the input unchanged or convert it into the 2's complement form.When the Comp-sig goes high,the XOR gates invert the input bits and a 1 gets added to the result. The generated result is the 2's complement of the input bits.On the contrary, when the Comp-signal goes low,the multiplicand inputs donot get inverted and a 0 gets added to them.Once the signed multiplicand gets processed ,the MSB of the result would then indicate the sign of the result(1 for negative ,0 for positive).The area and power consumption of a number of multiplier structures vary with the number of bit operands and the layout strategies.Since the Baugh-Wooley multiplier is an evolvement of conventional multiplier ,its performance can also be improved.

3.2. Booth Multiplier

Area-efficient and fast multipliers are the essential blocks for high performance computing.Therefore,multipliers should be small enough so that a large number of them may be integrated on a single chip.The Booth multiplier makes use of the Booth encoding algorithm in order to the reduce the number of partial productsby considering two bits of the multiplier at a time,thereby achieving a speed advantage over other multipliers.This algorithm is valid for both signed and unsigned operands[10].

ISSN: 2278-4632
Vol-11 Issue-01 2021

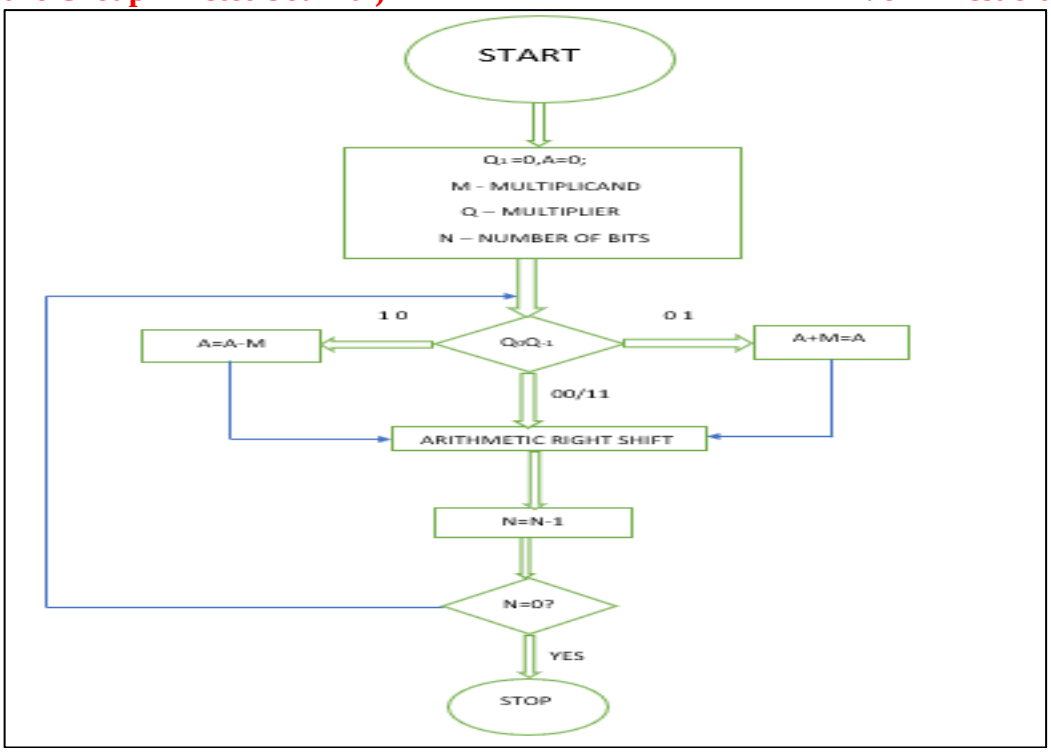

Figure 9. Flow chart of Booth multiplier algorithm

3.2.1 Standard radix-2 Booth multiplication rules

1. Append a zero to the right of the LSB of the multiplier.
2. Inspect groups of two adjacent bits of the multiplier, starting with the LSB and the appended zero.
   - If the pair is 00 or 11, then shift the partial product 1 bit to the right.
   - If the pair is 01,then add the multiplicand to the partial product and shift the partial product 1 bit to the right.
   - If the pair is 10,then subtract the multiplicand from the partial product and shift the new partial product 1 bit to the right[10][11].

3.Proceed with overlapping pairs of bits such that the MSB of a pair becomes the LSB of the next pair. In this manner, 1 bit of the multiplier number is eliminated in each pass through the algorithm.

4.When the last pair of bits is examined, the partial product is updated the rules expect that no shift is performed.

The main motivation behind the realization of the Baugh-Wooley multiplier is to multiply negative numbers based on the 2's complement representation.It resembles regularity in layout.Ever since Booth's algorithm got introduced,it has always been used and developed because it renders the advantage of reducing the number of partial products. This algorithm is useful for applications where the hardware cost is a major concern.

4.RESULTS AND DISCUSSION

The above implemented multipliers are simulates using Xilinx Vivado 2018.1,Spartan-7, xc7s25csga225-2.The synthesis results for all multipliers are obtained and their power and area are analyzed.Table 1 demonstrates the detailed comparative analysis of the implemented multipliers interms of area and power.

ISSN: 2278-4632
Vol-11 Issue-01 2021

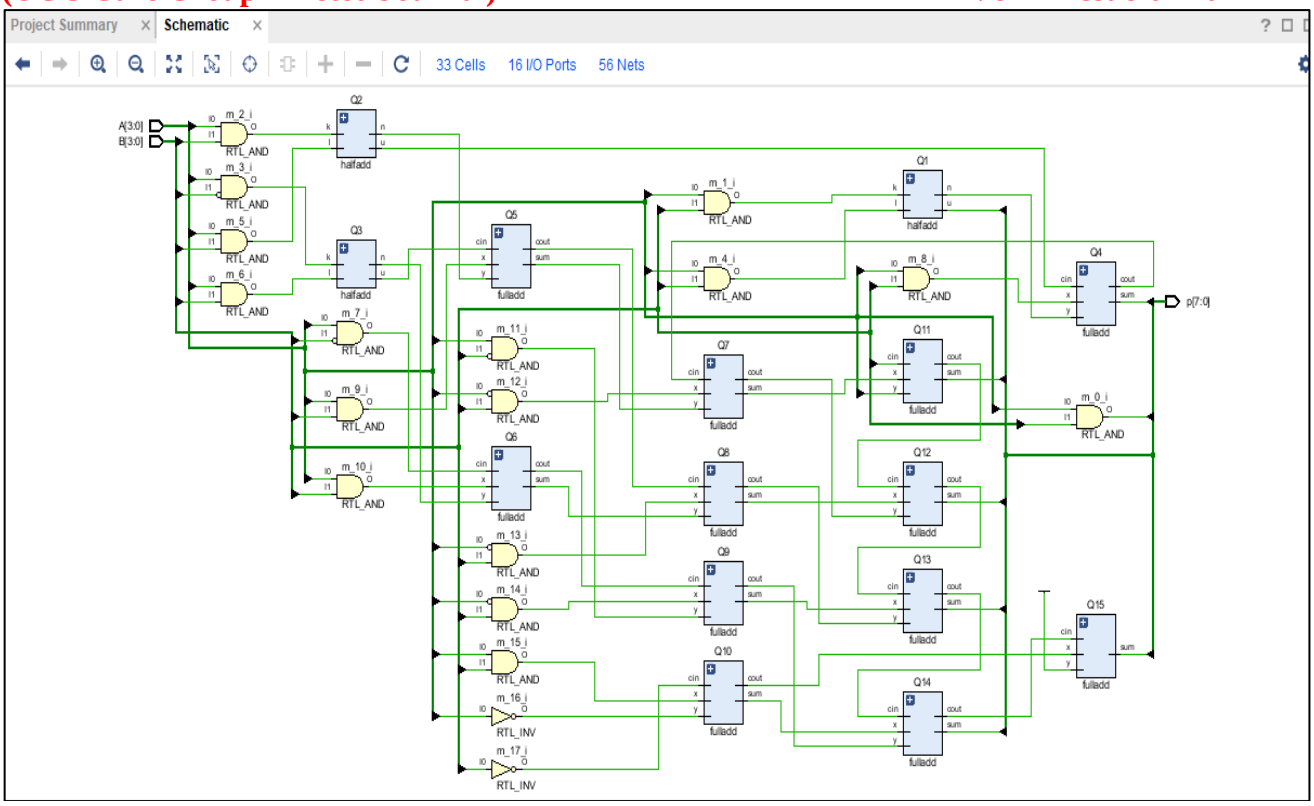

Figure 10: RTL schematic of 4-bit Baugh-Wooley Multiplier

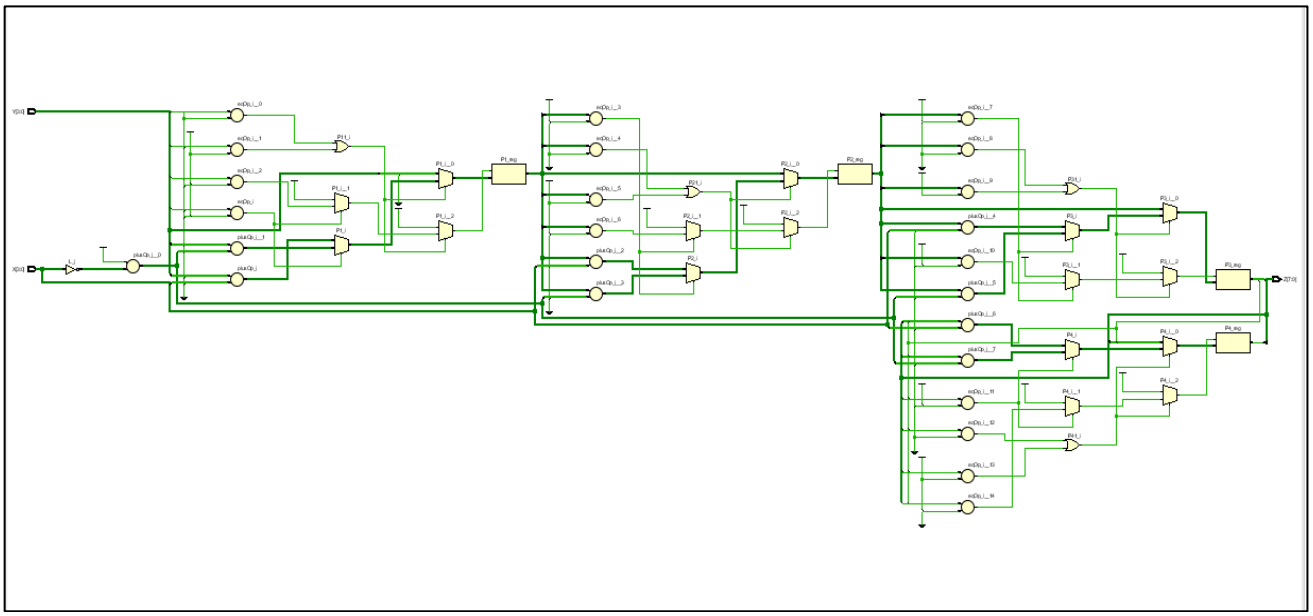

Figure 11: RTL schematic of 4-bit Booth Multiplier

ISSN: 2278-4632
Vol-11 Issue-01 2021

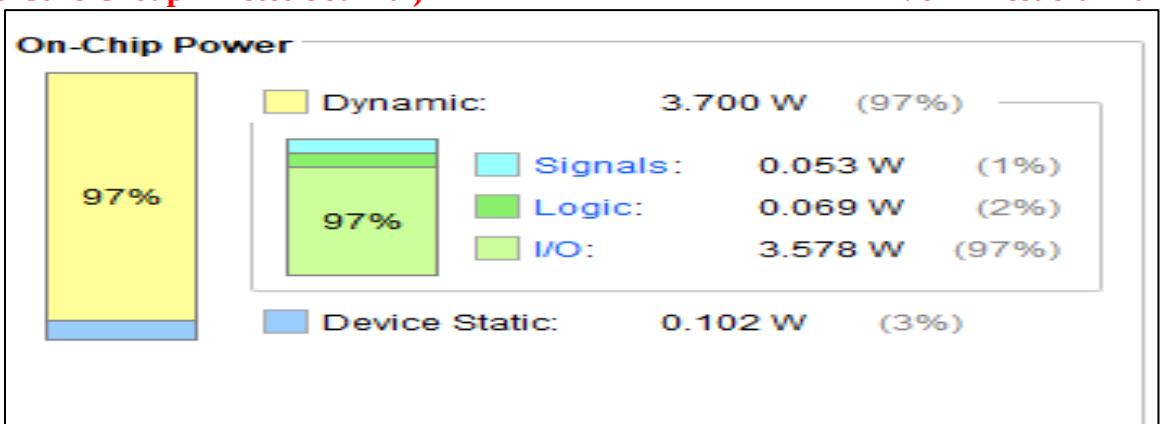

Figure 12: Power consumption of 4-bit Baugh-Wooley Multiplier

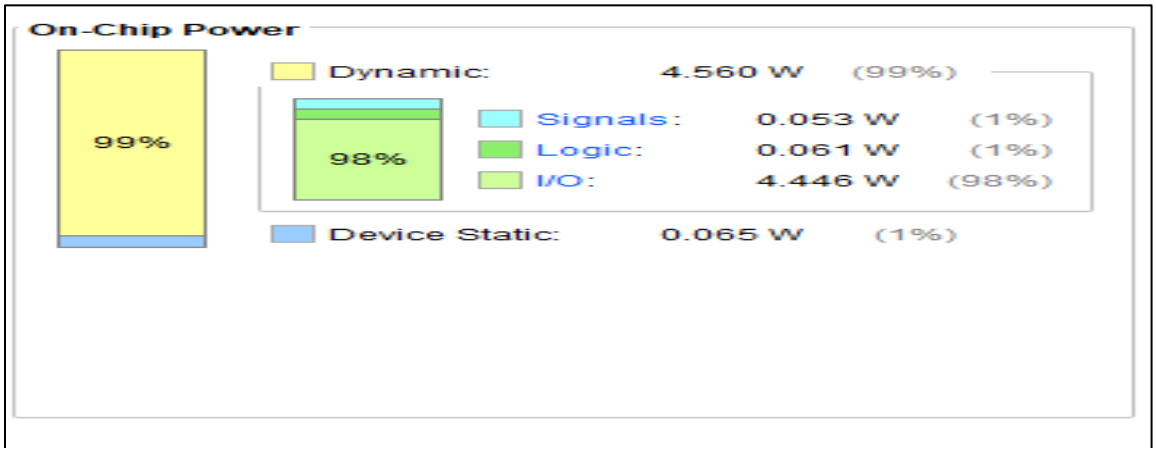

Figure 13: Power consumption of 4-bit Booth Multiplier

Table 1. Comparitive analysis of multipliers

Multiplier Type	Power Consumption(watts)	%of LUT's
Array Multiplier	4.36	1.02
RowBypassing Multiplier	4.15	1.23
Wallace Tree Multiplier	4.08	1.36
Conventional Multiplier	4.05	1.09
ColumnBypassing Multiplier	4.04	1.23
Booth Multiplier	4.60	0.68
Baugh-Wooley Multiplier	3.70	1.23

From the Table 1,Baugh-Wooley power consumption is 3.70 watts and device utilization summary is 1.23% and Booth multiplier power consumption From the table we observe that Baugh-Wooley has the lowest power consumption and Booth multiplier occupies small area.For low power applications Baugh-wooley can be used and for low area applications Booth-multiplier can be used.

5. CONCLUSION

The behaviour of these parallel multipliers Array Multiplier, Conventional Multiplier, Bypassing Multipliers,Walace Tree Multiplier,Baugh-Wooley Multiplier are described using VHDL and then simulation results and synthesis reports are obtained.The synthesis reports of multipliers shows that Baugh-Wooley consumes less power and Booth multiplier occupies smaller area.Further, study and comparitive analysis can be done on 16,32,64 bit multipliers and analysis can be done.

REFERENCES:

[1]PrajwalGaharwar,AyoushJohari,Design and implementation of multipliers, DOI:10.1109/SCEECS.2016.7509342,2016

[2]Sumit Vaidya and Deepak Dandekar,DELAY-POWER PERFORMANCE COMPARISION OF MULTIPLIERS IN VLSI CIRCUIT DESIGN,OM College of Engineering,Wardha,Maharasthra,India.

[3]Rajasree Shanmuganathana , Kathirvel Brindhadevi,Comparative analysis of various types of multipliers for effective low power. Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam,2019

[4]E.L. Braun,Digital Computer Design,Logic circuitary,synthesis,Academic press,Newyork,1963.

[5]D.A.Pucknell and K.Eshraghin,Basic VLSI Design, 3 rd ed.,Prentice Hall,1994.

[6]Yamini devi Ykuntam ; Katta Pavani,Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs,DOI: 10.1109/ICCCNT49239.2020.9225404,2020,IEEE.

[7]Pankaj Kumar and Rajender Kumar Sharma,LOW POWER MULTIPLIER DESIGN WITH IMPROVED COLUMN BYPASSING SCHEME,Natioal Institute of Technology,Kurukshetra,India,2016.

[8]Jin-Tai-Yan;Zhi-Wei Chen, Low-Power multiplier design with row and column bypassing,Chung Hua University,Hsinchu,Taiwan,DOI: 10.1109/SOCCON.2009.5398054,2010,IEEE.

[9]Pramodhini Mohanty,An Efficient Baugh-Wooley Architecture for Signed and Unsigned Fast Multiplication,Noida Institute of Engineering,Greater Noida,2013.

[10]Bhavya Lahari Gundapaneni,JRK kumar Dabbakuti,Booth Algorithm for the Design of Multiplier,2019.

