Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

A SURVEY - COMPARISON OF MULTIPLIERS USING DIFFERENT LOGIC STYLE

Lakum kalpesh[1], B.H.Nagpara[2]

[1]Student M.tech EC in C U Shah Engineering Collage,Kothariya,Surendranagar.
[2]Assistnat proffesor at C U Shah Engineering Collage,Kothariya,Surendranagar.

Abstract--In a typical processor, Multiplication is one of the basic arithmetic operations and it requires substantially more hardware resources and processing time than addition and subtraction. In fact, approximately 8.72% of all the instruction in typical processing units is multipliers. In computers, a typical CPU allot a considerable amount of processing time in implementing arithmetic operations, multiplication operations. In this paper, comparision of different multipliers is done for low power requirement and high speed. The paper gives information of "booth" algorithm of Mathematics which is utilized for multiplication to improve the speed of multiplier and , area parameters of multipliers

Keywords- Different types of multiplier, Design logic style, multiplier algorithm

I. INTRODUCTION

Multiplication is an important fundamental function in arithmetic logic operation. Since, multiplication rules the performance time of most Digital Signal Procesing algorithm, therefore high speed multiplier is much preferred. Currently, multiplication time is still the main factor in determining the instruction cycle time of a DSP chips. With an ever-increasing pursuit for greater computing power on battery-operated mobile devices, design importance has shifted from optimizing conventional delay time area size to minimizing power dissipation while still maintaining the high performance. The low power and high speed VLSI can be implemented with different logic style. The three important thoughts for VLSI design are power, area and delay for different multipliers. There are many proposed logics (or) low power dissipation and high speed for every multiplier and each logic style has its own advantages in terms of speed and power. Fast multipliers are a main topic in the VLSI design of high speed processors. Most of the multipliers were designed using mainly Pass Transistor Logic circuits. Pass Transistor Logic is chosen to implement most of the logic function within the multiplier. In the design of arithmetic functions, Gate diffusion input logic requires fewer devices to implement basic logic function in arithmetic operation as compared to the CMOS; it is one of the important advantages of GDI Logic over CMOS. This transforms into Lower input gate capacitance and power dissipation as compared to static CMOS . The multiplier concurrently added the partial product bit generated with the accumulator bit.

The GDI approach allows implementation of wide range of complex functions using only two transistors. This method is hence suitable for design of fast low-power circuits using reduced number of transistors compared to CMOS and existing PTL techniques.

II. BASIC MULTIPLICATION OPERATION

The most basic form of multiplication consists of making the product of two binary numbers. (m×n) bit multiplication can be viewed as founding n partial product of m bits each, and then summation properly shifted partial Products to produce an (m+n) bit result P.

Let A and B be the operands with m and n bits are respectively. Using shift and enlarge type of approach the product P of these two operands can be represented as shown in equation.

$$A = [\sum_{j=0}^{m-1} a_j 2^j] \quad B = [\sum_{i=0}^{n-1} b_i 2^i]$$

$$P = [\sum_{j=0}^{m-1} a_j 2^j] \; [\sum_{i=0}^{n-1} b_i 2^i]$$

$$= \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} a_j b_i 2^{i+j}$$

A Binary multiplier is an electronics hardware device used in digital electronics or a computer or other electronics devices to perform swift multiplication of two numbers in binary symbol. It is made using binary adders. The rules for binary multiplication can be stated as follows

(a) If the multiplier digit is a 1, the multiplicand is simply derivative down and represents the product.
(b) If the multiplier digit is a "0" then product is also 0.

III. METHODS AND PERFORMANCES

There are numbers of methods that can be used to perform multiplication. In general, the choice is based on factors such as latency, throughput, area, and design complication. More effective parallel methodology uses some sort of array or tree of full adders to sum partial products. Array multiplier, Booth Multiplier algorithm and Wallace Tree multipliers are some of the standard methodologies to have hardware implementation of binary multiplier which are proper for VLSI implementation at CMOS level.

(A) Array Multiplier:

Array multiplier is an efficient layout of a combinational multiplier. Multipliscation of 2 binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array.

In array multiplier, consider two binary numbers W and Y, of m and n bits. There are mn summands that are produced in parallel by a set of mn AND gates. n x n multiplier needs n (n-2) full adders, n half-adders and n two AND gates. Also, in array multiplier poorest case delay would be (2n+1) td.

Array Multiplier gives more power consumption as well as optimum number of component necessary, but delay for this multiplier is largest. It also requires largest number of gates because of which area is also increased; due to this array multiplier is less cheap. Thus, it is a fast multiplier but hardware complexity is high.

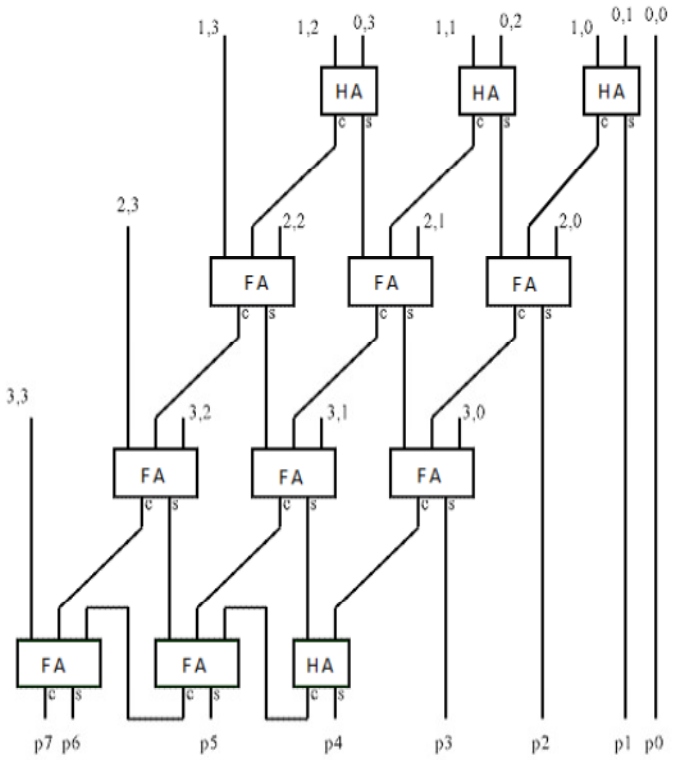

Fig.1 Array Multiplier

(B) Wallace tree multiplier

A fast process for multiplication of two numbers was established by Wallace tree. Using this method, a three steps procedure is used to increase two numbers; the bit products are made, the bit product matrixs is reduced to a two row matrix where sum of the row equals the sum of bit products, and the two resulting rows are sum with a fast adder to produce a final product.

In the Wallace tree technique, there are three bit signals are passed to a one bit full adder ("3W") which is called a three input Wallace tree circuit, and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal there of is supplied to the next stage of the full adder located at a one bit higher position.

Wallace tree is reducing the number of operands at initial chance. If you trace the bits in the tree, you will find that the Wallace tree is a tree of carry-save adders arranged as shown in figure. A carry save adder consists of full adders like the more familiar ripple adder, but the carry output from each bit is carried out to form second result vector rather being than wired to the next most significant bit. The carry vector is 'saved' to be combined with the sum later, hence the carry-save adder.

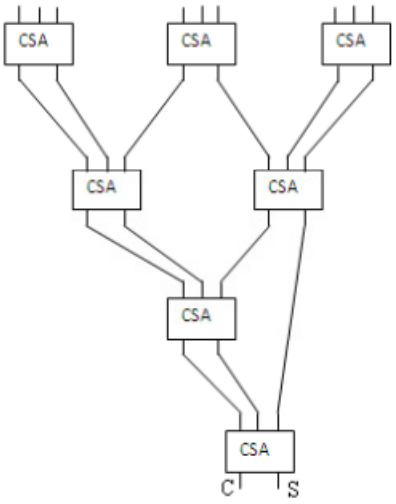

Fig.2 Wallace Tree Multiplier

In the Wallace tree technique, the circuit design is not easy although the speed of the operation is high since the circuit is quite irregular.

(B) Booth Multiplier

Another improvement in the multiplier is by reducing the number of partial products produced. The Booth record multiplier is one such multiplier; it scans the three bits at a time to reduce the number of partial products. There are three bits are: the two bit from the present pair; and a third bit from the high order bit of an adjacent lower order pair. After examining each trio of bits, the troikas are converted by Booth logic into a set of five control signals used by the adder cells in the array to control the operations performed by the adder cells.

To speed up the multiplication Booth encoding achieves some steps of multiplication at once. Booth's algorithm takes benefit of the fact that an adder sub tractor is nearly as fast and small as a simple adders.

From the essentials of Booth Multiplication it can be verified that the addition/subtraction operation can be avoided if the successive bits in the multiplicand are same. If 3 successive bits are same then addition/subtraction operation can be missed. Thus in most of the cases the delay associated with Booth Multiplication are smaller than that with Array Multiplier. However the performance of Booth Multiplier for delay is input data depended. In the poorest case the delay with booth multiplier is on per with Array Multiplier.

The method of Booth recording reduces the numbers of adders and hence the delay required to produce the partial sum by exploratory three bits at a time. The high performance of booth multiplier comes with the disadvantage of power consumption. The reason for this is the large number of adder cells (15 cells for 8 rows-120 core cells) that consume power.

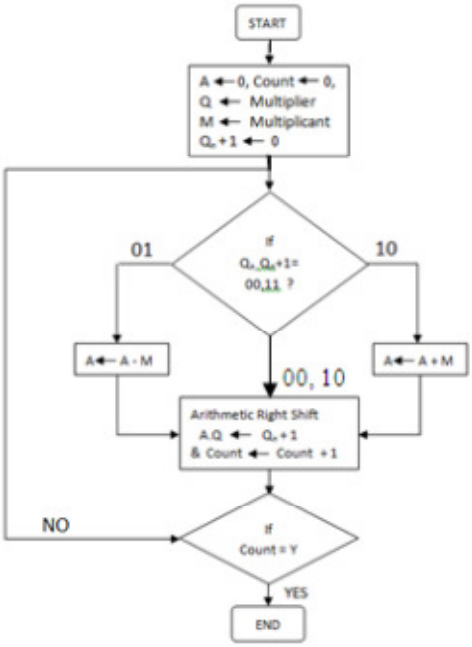

Fig.3 Booth Multiplication Algorithm

IV. MODIFIED BOOTH ALGORITHM

As explained in the chapter 1 that the modified booth algorithm can be of different types. Radix based modified booth encoding Multipliers are Radix-4, 8, 16, 32 and all of have its advantages and disadvantages. The Radix-2 disadvantages can be eliminated by examining three bits of Y at a time rather than two. The modified Booth algorithm is performed with recoded Multiplier which multiplies only a and 2a of the multiplicand, which can be obtained easily by shifting and/or complementation. The truth table for modified Booth recoding is shown below:

Yi+1	Yi	Yi-1	Zi+1	Zi	Zi/2	Expalination
0	0	0	0	0	0	No string Is in sight
0	0	1	0	1	1	End of string of 1s
0	1	0	0	1	1	Isolated 1
0	1	1	1	0	2	End of string of 1s
1	0	0	1	0	2	Beginning of string of 1s
1	0	1	-1	1	-1	End a string begin a new one
1	1	0	0	-1	-1	Beginning of string of 1s
1	1	1	0	0	0	Continuation of string of 1s

V. COMPARISION AND DISCUSSION

There are number of techniques for logic implementation at circuit level that improves the power dissipation, area and delay parameters in VLSI design. Implementation of parallel Multiplier in CPL logic shows significant improvement in power dissipation. CPL requires more number of transistors to implement as compared to the CMOS and provides only a little improvement in speed. Pass Transistor Logic which offers better performance over both the CMOS and CPL in terms of delay, power, speed and transistor count. The PTL outperforms the CMOS implementation in speed and great in power dissipation, with approximately same transistor count . When compared to CPL, PTL is faster and shows improvement in power and transistor count. Also GDI logic for batter implemention of logical circuit.

Parameter	Array Multiplier	Wallace Tree Multiplier	Booth's Multiplier
Opeartion speed	Less	High	Highest because the cycle length is as small as possible
Time Delay	More (n+1)tFA	Log(n)	Less (ntFA/2 + ntFA)
Area	Maximum area because it uses a large number of adders.	Medium area because Wallace Tree used to reduce operands.	Minimum area because adder/subtracter is almost as small/fast as adder.
Complexity	Less complex	More complex	Most complex
Power Consumption	Most	More	Less

Table 1. Comparison between Multipliers

Parameters	CMOS Logic	CPTL logic	PTL Logic	GDI logic
Number of Transistor	Most	More	Less	Less than PTL
Area	Maximum	Medium	Minimum	Minimum
Power	Most	More	Less	Less
Delay	Most	More	Less	Less
Speed	Less	Medium	High	Very High

Table.2 Comparison between multiplier designs in three different Logics

VI. CONCLUSION

It can be concluded that Booth Multiplier is superior in all respect like speed, delay, area, complexity, power consumption. However Array Multiplier requires more power consumption and

gives optimum number of components required, but delay for this multiplier is larger than Wallace Tree Multiplier. Hence for low power requirement and for less delay requirement Booth's multiplier is suggested.

REFERENCES

[1] "Performance Comparison of Multipliers for Power-Speed Trade-off in VLSI Design" Sumit R. Vaidya, D. R. Dandekar RECENT ADVANCES in NETWORKING, VLSI and SIGNAL PROCESSING
[2] " Simulation of IEEE 754 Standard Double Precision Multiplier using Booth Techniques" Puneet Paruthi, Tanvi Kumar, Himanshu Singh / International Journal of Engineering Research and Applications (IJERA)
[3] "ANALYSIS AND COMPARISON OF VARIOUS PARAMETERS FOR DIFFERENT MULTIPLIER DESIGNS" vidhi gupta, J.S ubhi, International Journal of Scientific Research and Management Studies (IJSRMS)
[4] "Principles of CMOS VLSI Design", N. Weste, and K. Eshraghian, Addison-Weslet Publishing Company
Jashanpreet kaur, Navdeep kaur, Amit Grover "A Review on Gate Diffusion Input (GDI) " International Journal of Advance Research in Electronics, Electrical & computer science Appliction of Engineering & Technology, volume 2, Issue 4, July 2014.
[5] Pooja Verma, Rachna Manchanda " Review Of Various GDI Techniques For Low Power Digital Circuits" International Journal of Emerging Technology and Advanced Engineering Volume 4, Issue 2, February 2014.
[6] Laxmi Kumre, Ajay Somkuwar, Ganga Agnihotri3 " Design of Low Power 8 bit GDI Magnitude Comparator" International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS) 2013.
[7] Aditya Kumar Singh, Bishnu Prasad De, Santanu Maity "Design and Comparison of Multipliers Using Different Logic Styles" International Journal of Soft Computing and Engineering (IJSCE) Volume-2, Issue-2, May 2012

