Booth Algorithm for the Design of Multiplier

Bhavya Lahari Gundapaneni, JRK Kumar Dabbakuti

Abstract*: Most commonly used operation in many electronic and computing systems is multiplication operation. In order to meet the challenges that occur from advanced technology low power consumption is one of the important features in order to meet the various applications. Among the arithmetic operations the multiplication is one of the important operation that act as a basic operation to be used in every circuit to get efficient than other operations. Out the different types of multipliers the booth multiplier is one of the standard technique that allows a smaller, circuits to operate with fast and quick multiplication by using encoding techniques to the signed numbers of 2's complement. This standard technique is mostly used for the designing of the chip for any application and then provide improvements that are required to reduce the number of the partial products to half. The "Complex multiplication" techniques. In this way the booth multiplier can be able to reduce the number of iteration steps for performing the multiplication. When we consider the number of partial products of other conventional multiplier the booth multiplier can get less number of partial products. The main goal of any VLSI projects is to perform operations with high speed, low power consumption and also less area. Among the three features the speed is one of the most important factor that plays a vital role for every application. So, if we consider the process of algorithm for booth multiplier it generally consists of two basic steps which are generation and addition of partial products. The multiplier speed depends on the fastness of the partial products generated and how fast the addition is done by the multiplier. In this paper different techniques and algorithms are used for the design of the booth multiplier in order to get less consumption and less area to be consumed. Also focused on the improvement of speed of the multiplier and to reduce the delay.

Index Terms: Signed numbers, Booth multiplier, speed

I. INTRODUCTION

Most of the multiplication is used and many of the state-of -the-art systems operate. Generally in digital signal processors the fast multipliers are required[1].The main multiplication process is step by step, as each step finally has to add, the multiplication and add to itself the multiplying value between the multiplier and the multiplicand is a multiplier. This process leads to a wide operation which takes longer to get the result and the hardware required to do the circuit occupies additional areas and components, and leads to low speed. This process also takes up several areas. In many applications, generally a larger number of circuit sectors for Very Large Scale Integration (VLSI) use arithmetic operations.

Revised Manuscript Received on May 10, 2019.

Bhavya Lahari G, **M.Tech** student in ECM, KL deemed to be University, Vijayawada, India.
JRK Kumar Dabbakuti, Associate Professor in ECM, KL deemed to be University, Vijayawada, India.

In more frequent cases, multiplication relies strongly on the flexibility and quality of components that are useful in hardware processing multiplication, as multiplication in most applications is widely utilized. The necessary number of algorithms are given an extremely high interest in the application multiplication field and are implemented in the performance of the multiplication operation provided within the given literature. In particular, multipliers use a short-bit width multiplier that acts as building blocks for high-performance embedded processors and a core for digital signals processing. A reduction in the width of the multiplier plays an essential role in the processing of this kind of multiplication and acts as basic building blocks.

Advanced bit-width multipliers (less than 32 bits) are also very common in FPGAs. Until now, the steps taken and instructions used are the next step to multiply bits with lower width multipliers. In the process of multiplication we have three main steps. Thus, the first encoder takes the bits into the combined and produces partial products from the decoder depending on the type of multiplication. The results are then limited to one row in order to obtain the final sum and the single row produced gives the final carries, and then in the third phase the final sums and carries out the final output for the specific bits. We prefer Modified Booth Encoding mainly to reduce the number of partial products to half..

With the advancement in modern technology, multipliers design provide the high speed multiplication, less power consumption , order ness of layout and with reduced area. Or combination of all these make multiplier useful in achieving high performing ,less power consumption and dense implementations. So the best process of doing multiplication is modified booth multiplication as it reduces to half the number of partial products. Hence the modified booth algorithmic can be used for doing multiplication for both signed and unsigned bits.

II. LITERATURE SURVEY

In this section we are going to discuss some previous methodologies and their corresponding method of implementation. Sukhmeet et al. proposed worked on comparisonof Radix-2 booth multiplier with the Radix-4 booth multiplier. In this paper a parallel MAC is implemented with small possible delay. The parallel MAC is used in various applications of signal processing and video/graphical applications. In this proposed work the modified booth multiplier is designed with the help of high speed adder which is used to speed up the multiplication operation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The design implementation is done with the VHDL and simulation is done using Xilinx ISE 9.li software whereas FGPA XC3s50-5pq208 is used for hardware implementation.

Jani Basha Shaik et al. has proposed an efficient booth multiplier that is simulated with a software tool of Xilinx ISE design suite 14.2 and implemented on hardware device of nexus 2 kit ,FPGA.

Chinababu Vanama and M.Sumalatha implemented a modified booth multiplier of logical verification using Xilinx –ISE tool with the help of target technology and performed placement and routing operation for the system verification.

III. PROPOSED MODEL

In this proposed method the implementation of booth multiplier is considered due to the drawbacks that occur due to normal multiplication. In case of normal multiplication the multiplication is applied for two fixed numbers and also the multiplication is done for each and every bit of two numbers of signed numbers. If we consider the bits of '1' in the multiplier then the multiplication operation is done for those successive multiplicand bits and finally these bits are displayed. In case the bit is '0' then we get zero for the multiplicand bits and then those are displayed in the iteration steps.

So, it is a long process in case of multiplication as each and every bit is multiplied and it takes more time for the multiplication operation. This normal multiplication thus takes more time for the multiplication operation where each and every bit is multiplied due to this we get more numbers of iteration steps. As the number of iteration steps increases the delay also increases. So, in order reduce the number of iteration steps and other drawbacks we go for booth multiplier.

Example for the normal multiplication:

```
   0011
X     1010
      0000
     0011
    0000
   0011
0 0 11110
```

The basic booth multiplier is used for both signed and unsigned bits in multiplication. In case of this multiplier shifting operation is done in some cases directly and go for 2's complement in other cases. So it is one of the complex process where each and every bit is checked and then shifting takes place in multiplication.

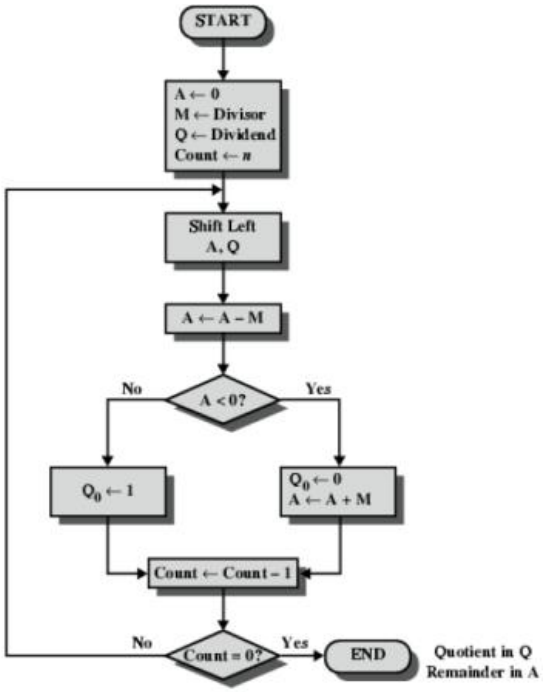
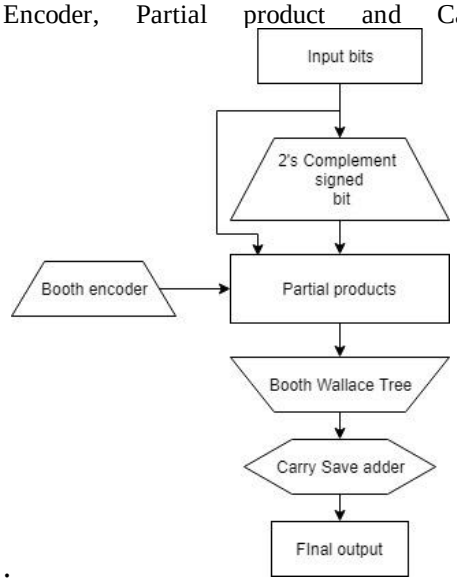


Fig 1. Block diagram of Booth multiplier

When the sequence counter reaches to zero then the final result occurs. Thus it is considered as one of the longest process to proceed and takes long time in processing, due to this drawback the delay also increases and slowly leads to the increment of the delay and then reduces the speed of the multiplier. Due to this power consumption also increases. By considering all these drawbacks we go for the modified booth multiplier.

The number iteration steps will be reduced while performing the multiplication using booth multiplier. The architecture consists of four parts: Complement Generator, Booth Encoder, Partial product and Carry SaveAdder



.

Fig.2 Flow Chart of Booth multiplier

a) Complement Comparator

In case of this complement comparator the generation of 2's complement is done for the multiplicand or for the given data and finally the complemented results are obtained . these complemented results are used in the cases of requirement else the direct result is taken into consideration in some

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

cases due to some predefined cases.

b) Encoder

In order to do the multiplication the initial bits are given to the encoder and then the applied bits of encoder are considered as one bit for the two bits and finally the multiplication is applied for the bits.

c) Partial Product Generator:

The decoded bits are obtained at partial product generator and leads to the generation of less number of partial products during the multiplication of numbers. This leads to the reduction of number of partial products.

d) Wallace Multiplier:

This multiplier is similarly acts as array multiplier. This multiplier uses adders of half adder and full adder. In case of this every bit is multiplier by every other bit during the multiplication operation.

e) Carry save adder:

In this the fast addition of the partial products are done and the result is obtained so fast ,so this adder is considered than other adders.

The Booth multiplier identifies the operand that acts as a multiplier and can do multiplication for the algorithm as it reduce the number of steps while doing addition when compared with normal multiplication. In case of multiplication the operation is performed for every bits of multiplier with the multiplicand and then the generation of partial product occurs in respective order and then add all the partial products obtained. The most interesting thing is additions performed in this multiplication is data dependent, that makes this a perfect algorithm.

The multiplication of signed numbers is not possible as same as unsigned numbers because the signed numbers in 2's complement form cannot give the exact result if the same process of multiplication is applied for unsigned numbers. That is why booth algorithm is used and it deteriorates the sign of the final result. Thus booth algorithm performs high speed multiplication and find its way in different applications like digital signal processing , radar etc..

IV. BOOTH ALGORITHM

1. Adding '0'bit at rightside to LSB of the multiplier and consider from the right most of multiplier to make combinig of 2 bits from rightside to leftside and respective multiplier.
2. 00:11: does not perform any operation.
3. 01: mark the ending of string 1s and then adding multiplicand to partial products
4. 10 : mark the begin of the string 1s then subtracting multiplicand from partial products.

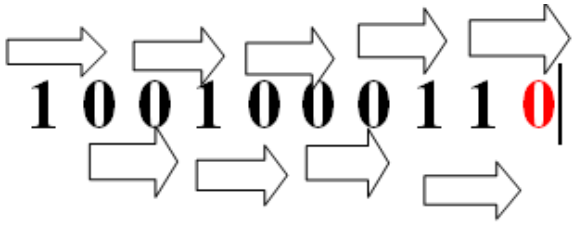


Fig 3. 2 bit combining of booth recorder

The required way of getting results for recognizing the highest speed multiplier is to augment parallelism was it useful in getting less no.of successive calculation levels. Similarly the booth algorithm for radix-4 which compares 3 bits with overlapping technique.

As this multiplication is able to reduce the number of partial products by half the total number of partial products to normal multiplication.

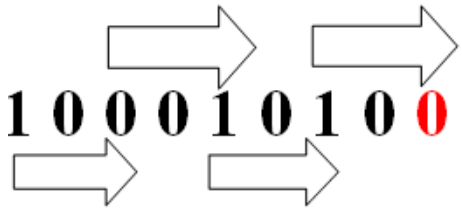


Fig 4. 3 bit pairing as per booth recorder

So, by considering the 3 bits the speed of thr multiplication can be improved and the numbers of multiplication steps are reduced to half than the original conventional multiplication. The Booth multiplication also have so advantages like when three bits are same then no operation can be performed and due to this the number of adders are reduced and the complexity of the multiplier can be reduced.

This multiplier has specific operation for successive bit operation and not required to perform addition and subtraction operation for every step of multiplication. Also the multiplication of signed numbers is not possible as same as unsigned numbers because the signed numbers in 2's complement form cannot give the exact result if the same process of multiplication is applied for unsigned numbers. That is why booth algorithm is used and it deteriorates the sign of the final result. Thus booth algorithm performs high speed multiplication and find its way in different applications like digital signal processing , radar etc.

Xi	Xi-1	Qi-1	Multiplier value	condition
0	0	0	0	Zeros String
0	1	1	+1	Ending string 1s
1	0	-1	-1	Beginingstring 1
1	1	0	0	String 1s

Table 1 booth recording table for Radix -2

V. SIMULATIONS AND RESULTS

The simulations of the booth multiplier can be obtained by using XilinX tool.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

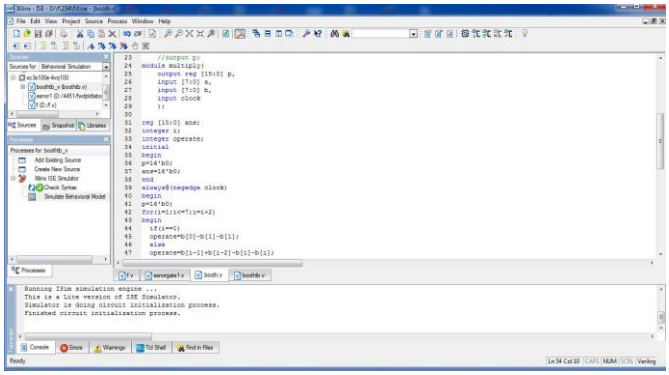


Fig 5 Implementation of synthesis model

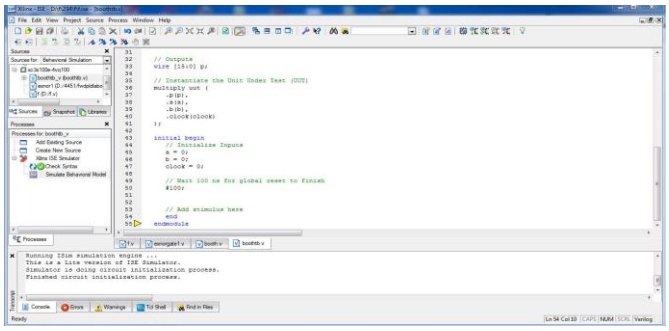


Fig 6 test bench model

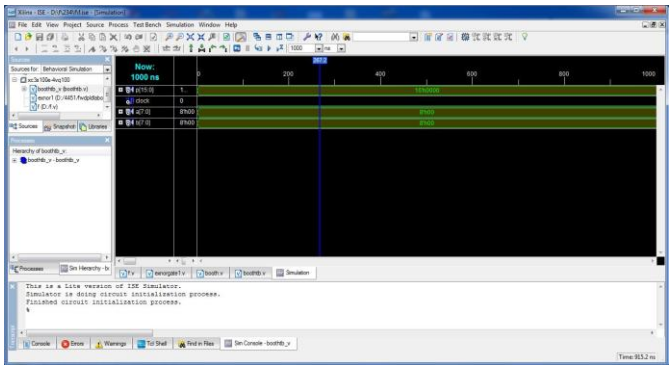


Figure 7 Simulation result of Radix 2 series

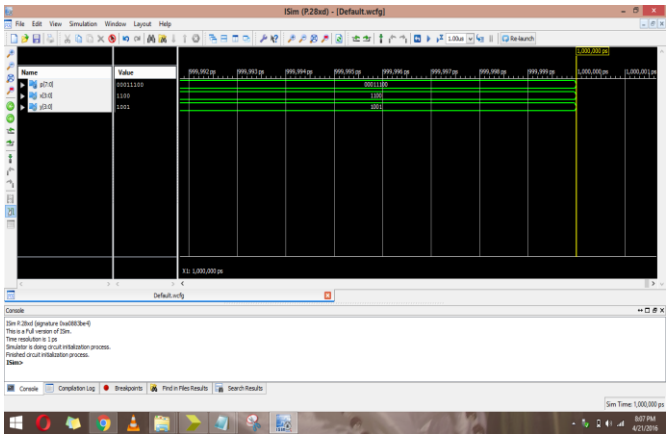


Fig 8 Multiplication of Radix 4 series

VI. CONCLUSION

The booth multiplication is most efficient one which is has more capability. The multiplication is suitable for the both signed and unsigned numbers. The booth algorithm is a process which will reduce the number of partial products during the multiplication .This multiplication process of booth and their representations in terms of multiplication increases the speed, the no of calculations required to implement ,and the size of hardware is implemented can be reduced.

REFERENCES

1. Bano, Nishat. "VLSI design of low power booth multiplier." *International Journal of Scientific & Engineering Research* 3.2 (2012): 2-4.
2. Yeh, Wen-Chang, and Chein-Wei Jen. "High-speed Booth encoded parallel multiplier design." *IEEE transactions on computers* 49.7 (2000): 692-701.
3. Kuang, Shiann-Rong, Jiun-Ping Wang, and Cang-Yuan Guo. "Modified booth multipliers with a regular partial product array." *IEEE Transactions on Circuits and Systems II: Express Briefs* 56.5 (2009): 404-408.
4. Cho, Kyung-Ju, et al. "Design of low-error fixed-width modified booth multiplier." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 12.5 (2004): 522-531.
5. Rajput, Ravindra P., and MN Shanmukha Swamy. "High speed Modified Booth Encoder multiplier for signed and unsigned numbers." 2012 UKSim 14th International Conference on Computer Modelling and Simulation. IEEE, 2012
6. Kuang, Shiann-Rong, Jiun-Ping Wang, and Cang-Yuan Guo. N "Modified booth multipliers with a regular partial product array." IEEE Transactions on Circuits and Systems II: Express Briefs 56.5 (2009): 404-408.
7. Sukhavasi, SusruthaBabu, and SuparshyaBabu Sukhavasi. "Power Competent Cmos Comparator For Analog To Digital Converter Circuits." International Journal of Advances in Engineering & Technology 6.5 (2013): 2196.
8. Lin, Hsin-Lei, Robert C. Chang, and Ming-Tsai Chan. "Design of a novel radix-4 booth multiplier." The 2004 IEEE Asia-Pacific Conference on Circuits and Systems, 2004. Proceedings.. Vol. 2. IEEE, 2004.
9. Chengdong Liang ; Lijuan Su," An Innovative Booth Algorithm", IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),2016.
10. Razaidi Hussin ; Ali Yeon Md. Shakaff," An Efficient Modified Booth Multiplier Architecture", IEEE, International Conference On Electronic Design,2008.

AUTHORS PROFILE

Author-1 Photo

Bhavya Lahari G, **M.Tech** student in ECM, KL deemed to be UNIVERSITY, VIJAYAWADA, INDIA.

Author-2 Photo

JRK Kumar Dabbakuti, Associate Professor in ECM, KL deemed to be UNIVERSITY, VIJAYAWADA, INDIA.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication