International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 06 | June-2016 www.irjet.net p-ISSN: 2395-0072

Implementation of Booths Algorithm i.e Multiplication of Two 16 Bit Signed Numbers using VHDL and Concept of Pipelining

Megha jain[1], Pallavee Jaiswal[2]

[1]*Mtech Scholar Dr. C.V. Raman university Bilaspur*
[2]*Assistant Professor ECE Department Dr.C.V Raman University Bilaspur chattisgarh*

Abstract - *This paper explains the pipeline architecture of high-speed modified Booth multiplier.The proposed multiplier circuits are based on the implementation of modified Booth algorithm and the pipeline technique which are the most widely used to increase the multiplication speed and used to reduce the burdon of ALU. In order to implement the optimally pipelined multipliers, many kinds of experiments have been conducted. The speed of the multipliers is greatly improved by properly deciding the number of pipeline stages and the positions for the pipeline registers to be used. We described the proposed modified Booth multiplier circuits in VHDL . The resultant multiplier circuits show better performance than others)*

Key Words: *multiplier, pipeline, high-speed, modified Booth algorithm,VHDL.*

1.INTRODUCTION

As we know, based on theory of multiplication there are so many things that happens in backend which earlier was done by us manually in order to get the result. So, as there are so many processes causes so many phases and as it is automated in computing system, it ended up with so huge hardware resources as well. Considering the complexity of circuitry and huge hardware resources that caused the system execution/operating speed to be utterly slow which was then tried to be taken care of by presenting many ideas for over three decades. The major area of improvement which was under the consideration that time was the VLSI Operating Speed, Area and the Power Consumption in order to make it an efficient architecture. Though these three were under the attention, Speed was having high priority. Basically as we know multiplication takes place in two steps, first one is producing and second one is adding. Now the speed of execution depends on how fast we are producing the products and how fast we are adding them together. Booth's Multiplier algorithm are meant to deal with the first one that is how fast it can produce the partial products to increase the operating speed. To achieve the second factor that is adding the products we need an architecture which can act as a fast adder. The performance of the system would have an impact of multiplier and in order to improve the performance many algorithms has come up with an architecture and some of them are used in Pipeline and Vector computers.The multimedia and communication system are now using Digital Signal Processing(DSP) which is using the both high speed Booth multipliers and pipelined Booth multipliers to improve the speed. Now the challenge is the power and area as we have achieved the speed. Then after high speed DSP computation applications like Fast Fourier Transform came up with two requirements those were additions and multiplications. This absolute goal was achieved by using the conventional modified Booth encoding (MBE) which generates an irregular partial product array because of the extra partial product bit at the least significant bit position of each partial product row. This concept have low overhead with a simple approach to generate a regular partial product with fewer rows which ended up with lowering the complexity, reducing the area, delay and the power of MBE multipliers. The whole simulation is done using VHDL/Verilog HDL Modelism and XILINX ISE design tool is used to have it synthesized.[1]

2.LITRETURE SURVEY

In the year 2012, M.Zamin ali khan et.al proposed in this paper that Multipliers are key components of many high performance systems such as microprocessors, digital signal processors, etc. Op-timizing the speed and area of the multiplier is major design issue which is usually conflicting constraint so that improving speed results mostly in bigger areas. A VHDL designed architecture based on booth multiplication algorithm is proposed which not only optimize speed but also efficient on energy use.[2]In the year 2012, Nishat Bano et.al proposesd the design and implementation of Booth multiplier using VHDL.This compares the power consumption and delay of radix 2 and modified radix 4 Booth multipliers. Experimental results demonstrate that the modified radix 4 Booth multiplier has 22.9%

power reduction than the conventional radix 2 Booth Multiplier.[3]In the year 2013, Deepali Chandel et.al proposed that multiplication in hardware can be implemented in two ways either by using more hardware for achieving fast execution or by using less hardware and end up with slow execution. The area and speed of the multiplier is an important issue, increment in speed results in large area consumption and vice versa. Multipliers play vital role in most of the high performance systems. Performance of a system depend to a great extent on the performance of multiplier thus multipliers should be fast and consume less area and hardware. This idea forced us to study and review about the Booth's Algorithm, modified Booth's algorithm and its radix-2, radix-4, radix-8 forms.[4]

In the year 2013, S.Nagaraj et.al proposed in this paper to design a high speed multiplier with reduced error compensation technique. The fixed-width multiplier is attractive to many multimedia and digital signal processing systems which are desirable to maintain a fixed format and allow a little accuracy loss to output data. This paper presents the Design of error compensated truncation circuit and its implementation in fixed width multiplier. To reduce the truncation error, we first slightly modify the partial product matrix of Booth multiplication and then derive an effective error compensation function that makes the error distribution be more symmetric to and centralized in the error equal to zero, leading the fixed-width modified Booth multiplier to very small mean and mean-square errors. However, a huge truncation error will be introduced to this kind of fixed-width modified Booth multipliers. To overcome this problem, several error compensated truncation circuit approaches have been proposed to effectively reduce the truncation error of fixed-width modified Booth multipliers.[5]

3.BASIC BOOTH MULTIPLIER

Table 3.1 show that the Booth Multiplier Algorithm Rules, which are very important for this project. The algorithm rules give a procedure for multiplying binary integers in signed -2's complement representation

Table 3.1: Booth Multiplier Algorithm Rules

Xi	Xi-1	Operation
0	0	Shift Only
1	1	Shift Only
1	0	Add (-A) & Shift
0	1	Add (A) & Shift

3.3.1 Procedure and Example

Procedure Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P. Let m and r be the multiplicand and multiplier, respectively; and let x and y represent the number of bits in m and r.[3]

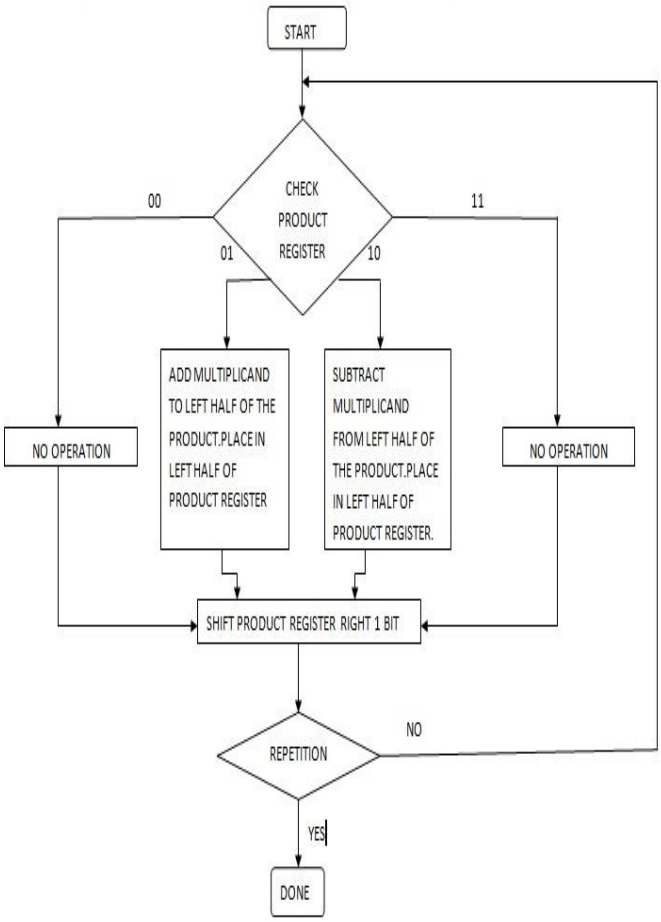
1.Determine the values of A and S, and the initial value of P. All of these numbers should have a length equal to (x + y + 1).

- A: Fill the most significant (leftmost) bits with the value of m. Fill the remaining (y + 1) bits with zeros.
- S: Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining (y + 1) bits with zeros.
- P: Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant (rightmost) bit with a zero.

1. Determine the two least significant (rightmost) bits of P.

- If they are 01, find the value of P + A. Ignore any overflow.
- If they are 10, find the value of P + S. Ignore any overflow.
- If they are 00, do nothing. Use P directly in the next step.
- If they are 11, do nothing. Use P directly in the next step.

2. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.
3. Repeat steps 2 and 3 until they have been done y times.
4. Drop the least significant (rightmost) bit from P. This is the product of m and r.



Example:

Find 2 × -4 ?

1 :- 00010 (5 bit multiplicand)

-2 :- 11110 (2's compliment of multiplicand)

-4 :- 11100 (5 bit multiplier)

Product register : 00000 11100

We will get the answer after 5 steps as we have taken a 5 bit multiplier .

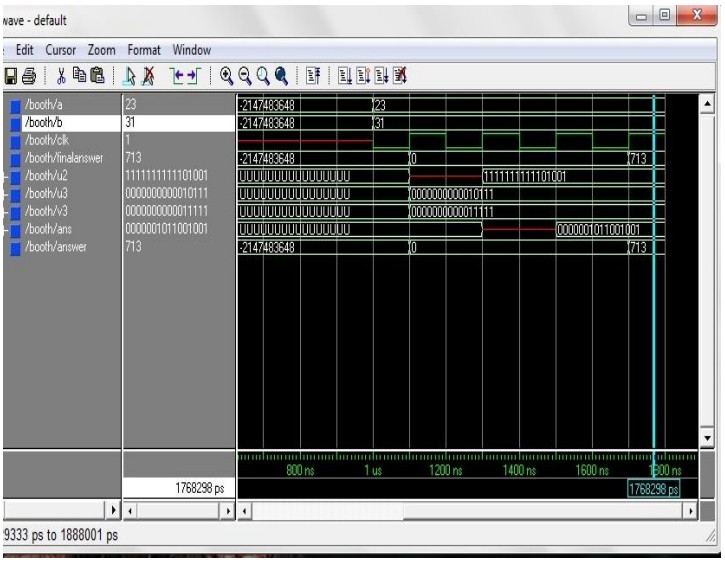
The whole procedure is shown in the following table :-

S. no	Product register	Multiplicand	Current bit	Previous bit	Operation product register
1	00000 11100	00010	0	0	Do nothing just right shift
2	00000 01110	00010	0	0	Do nothing just right shift
3 a	00000 00111	00010	1	0	Add 2's compliment of multiplicand to left most 5 bits of product register and ignore any over flow.
3 b	11110 00111	00010			Right Shift
4	01111 00011	00010	1	1	Do nothing just right shift
5 a	00111 10001	00010	1	1	Do nothing just right shift
5 b	00011 11000				

Table 3.2 Table For Booth Algorithm

4.Simulation Result

This is the one of the ALU part. Here we do simulation process with the help of the **Xilinx ISE 14.6** software. In this simulation process we force value of '**a**' as **23** and '**b**' as **31** and give 4 rising edges by forcing value of '**clk**' and we get the '**final answer**' as **713.**



5.1 Device Utilization Summary

By watching below table which is generated by the **Xilinx ISE 14.6** shows the area constraints utilized. That means number of LUT, Flip flops.Which provide us all details about implementation of above circuit or **Booth Multiplier**. in the device utilization summery the no of LUT is for 16 inputs is 3351 as per the no of flip flop to control the functioning of device is 152 which are moreover sufficient the slices which is in no 1143.

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	154	54,576	1%
Number used as Flip Flops	154		
Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	3,351	27,288	12%
Number used as logic	3,282	27,288	12%
Number using O6 output only	1,823		
Number using O5 output only	1,325		
Number using O5 and O6	134		
Number used as ROM	0		
Number used as Memory	0	6,408	0%
Number used exclusively as route-thrus	69		
Number with same-slice register load	9		
Number with same-slice carry load	60		
Number with other load	0		
Number of occupied Slices	1,143	6,822	16%

5. Conclusion

Design for 16x16 bit Booths multiplier i.e multiplier for two 16 bits signed numbers was successfully implemented and simulated using VHDL.Concept of pipelining was also used during programming.Pipelining gives higher speed. In large projects like this pipelined designs are very important for some blocks since it may act as a bottleneck for the performance of the whole design. On the other side there is a small disadvantage for pipelined designs. They introduce a small number of delay between input and output, in terms of clock cycle. For instance we have 3 stages in the pipelined code and hence the output comes only after 3 clock cycles, after the input is applied. But this disadvantage usually doesn't matter in most of the designs since after 3 clock cycles we can get continuous stream of output .

REFERENCES

[1].A. Booth, "*A signed binary multiplication technique*," Q. J. Me& Appl. March., vol. 4, pp. 236-240, 19.51

[2] M. Zamin Ali Khan1, Hussain Saleem2, Shiraz Afzal and Jawed Naseem. . "*An Efficient 16-Bit Multiplier based on Booth Algorithm"* (November-2012),International Journal of Advancements in Research & Technology, Volume 1, Issue 6, ISSN 2278-7763 Copyright © 2012 SciResPub

[3] Nishat Bano, " *VLSI Design of Low Power Booth Multiplier"(* February -2012*),*
International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February -2012 ISSN 2229-5518 IJSER © 2012 http://www.ijser.*or*
g

[4] Deepali Chandel 1 , Gagan Kumawat2 , Pranay Lahoty3 , Vidhi Vart Chandrodaya 4 , Shailendra Sharma, "*Booth Multiplier: Ease of multiplication* "(March 2013), International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013) 326".

[5]S.Nagaraj,R.Mallikarjua,,*"FPGA Implementation of Modified Booth Multiplier*",(March -April 2013),International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 2, March -April 2013.