Design of High Performance Pipeline based Booth Algorithm

Megha jain
M.Tech Scholar, Dr. C.V. Raman university bilaspur, India.

Pallavee Jaiswal
Assistant Professor (ECE Department), Dr. C.V. Raman university bilaspur, India.

Abstract – In this proposed work a pipeline based algorithm for highly efficient Booth multiplier has been designed. In this work major focus is on Area, Power and Delay. In this work pipelining concept has been used. In case of Area no of LuTs is on focus, in case of Delay the no of stages are reduced by pipelining and for power reduction simultaneous clock has been used. The design has been implemented on the ISE 14.6 of XILINX. Simulation results are shown on modelsim simulator.

Index Terms - Multiplier, pipeline, high-speed, modified Booth algorithm,VHDL.

1. INTRODUCTION

The algorithm we have used is Booth‘s Algorithm which is a smart move for multiplying signed numbers. It comes with the ability to both add and subtract as there are multiple ways to compute a product. This algorithm is an algorithm that utilizes two‘s complement notation of signed binary numbers for multiplication.

The multiplication earlier was generally implemented via sequence of addition then subtraction, and then shifts operations, and before that is was a series of repeated addition where in the number which is to be added is known as the multiplicand and the number of times it is added is known as the multiplier. The result we get is the multiplication result. After Each step of addition a partial product is generated. When the operands are integers, the product in general is twice the length of operands in order to protect the information content. This repetitive addition method that is recommended by the arithmetic definition is slow as it is always replaced by an algorithm that makes use of positional depiction.So the decomposition of multipliers makes it of two parts where the first part is committed to the generation of partial products, and the second part collects those products and then adds them. The fundamental multiplication principle is twofold i.e. evaluation of partial products and gathering of the shifted partial products, where there was consecutive additions of the columns of the shifted partial product matrix.The multiplier is effectively shifted and gets the proper bit of the multiplicand. The delayed, gated case of the multiplicand must all be in the same column of the shifted partial product matrix. Then they are added to form the product bit for the particular form. Multiplication is thus a multi operand operation. To expand the multiplication to both signed and unsigned numbers, a suitable number system would be the depiction of numbers in two‘s complement format

2. RELATED WORK

i. Proposed Algorithm

Entity booth is port(a,b:in integer; clk : in std_logic; finalanswer:out integer;

Architecture…….

Signal declaration

…

…

Process(sensitivity list)

Variable declaration

….

….

Begin

Integer to binary conversion of multiplicand

Two’s complement to represent the negative number

…

…

Integer to binary conversion of multiplicand

Two’s complement to represent the negative number

.

.

.

In pipelining manner

When “00”

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
3a	00000 00111	00010	1	0	Add 2's compliment of multiplicand to left most 5 bits of product register and ignore any over flow.
3b	11110 00111	00010			Right Shift
4	01111 00011	00010	1	1	Do nothing just right shift
5a	00111 10001	00010	1	1	Do nothing just right shift
5b	00011 11000				

Shift right and place a zero at the left most bit or the MSB

When "01"

Add multiplicand (u3) to the left most 16 bits and ignore overflow

Shift right and place a zero at the left most bit or the MSB

When "10"

Add 2's complement of the multiplicand (u2) to the left most 16 bits and ignore overflow

Shift right and place a zero at the left most bit or the MSB

When "11"

Just shift right and place a zero at the left most bit or the MSB

Last 16 bits are the answer of the multiplication in binary form.

2's complement starting conversion binary to integer**.**

Starting conversion binary to integer.

End behavioural.

3. SIMULATION RESULT AND WAVEFORMS

The simulation results showing in the figures are representing the proof of Area, Power and Delay constraints. Actually the RTL View of the internal part of the Booth multiplier is very large so it is look like this. From the synthesis report the delay is find i.e. 6.271 ns, which is optimized delay. The calculated power by the Power analyzer is few miliwatt. At Last the Area is shown in the analysis table of the ISE tool is calculated though Slices.

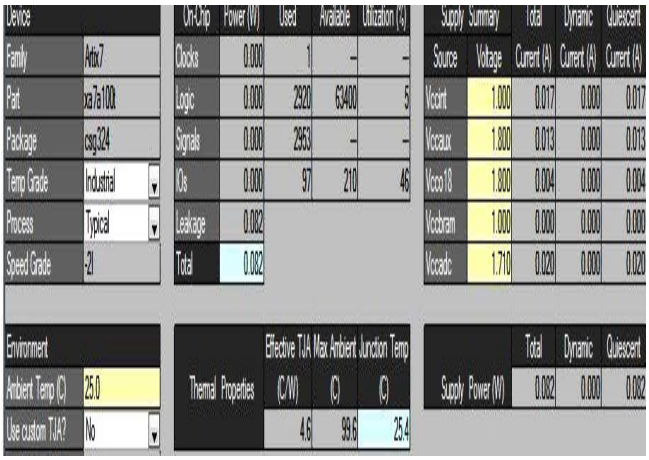

Fig.1. Figure shows the calculated power for proposed booth algorithm

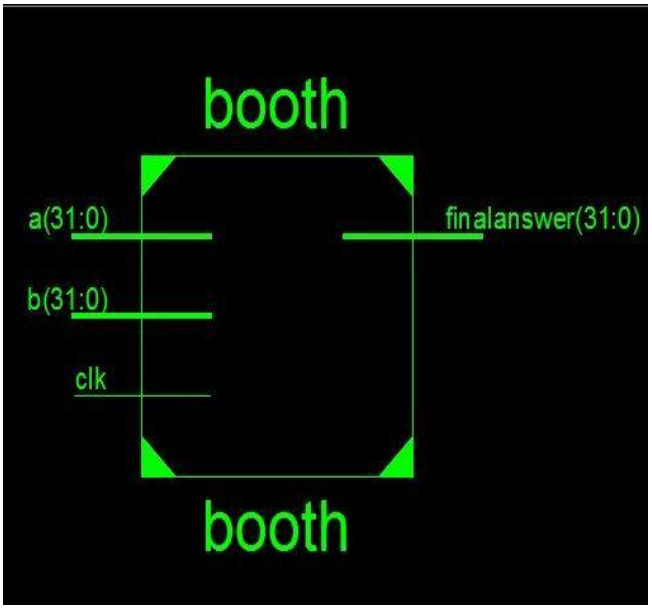

Fig.2 figure shows the RTL Top view of Booth Algorithm

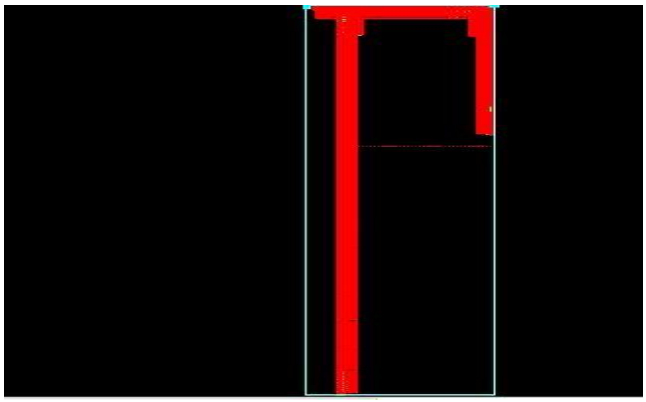

Fig.3 figure shows internal view of RTL

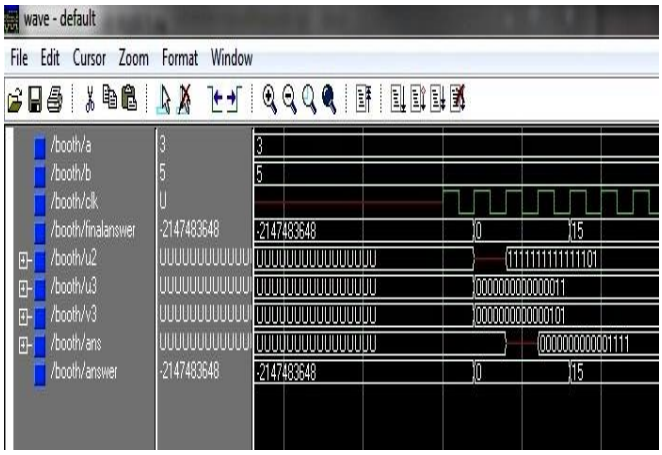

Fig.4 Figure shows the simulation waveform of Booth Multiplier.

Synthesis report:

Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'

Offset: 6.271ns (Levels of Logic = 1)

Source: answer_15 (FF)

Destination: finalanswer<15> (PAD)

Source Clock: clk rising

Data Path: answer_15 to finalanswer<15>

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

FD:C->Q 1 0.619 0.240 answer_15 (answer_15)

OBUF:I->O 5.412 finalanswer_15_OBUF (finalanswer<15>)

Total 6.271ns (6.031ns logic, 0.240ns route)

(96.2% logic, 3.8% route)

All the mathematical equations should be numbered as shown above.

4. CONCLUSION

Design for 16x16 bit Booths multiplier i.e multiplier for two 16 bits signed numbers has been successfully implemented and simulated using VHDL Concept of pipelining is used. The final results from the performance point of view are acceptable. As power is few miliwatt, delay is few nano second, and the slices used in the design are the 12 percent of the total present in the device. The Tool used ISE14.6 also have some advanced option to calculate the various parameters and optimizing capability at the extent.

REFERENCES

[1] 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.org

[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.Mallikarjuna Reddy,"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.

[6] K. Jeswanth Singh, B. Vamsi Krishna ,International "Design and Implementation of Modified Booth Encoder Multiplier using Carry Select Adder",(November 2014) Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-3 Issue-5, November 2014,

[7] Prof. V.R.Raut1 , P. R. Loya2 Prof. Ram Meghe, Dhamangaon (Rly), "FPGA Implementation of Low Power Booth Multiplier Using Radix-4 Algorithm"(August 2014)

Authors

Megha Jain
Mtech Scholar
Dr. C.V. Raman university Bilaspur(C.G.)

Miss Pallavee Jaiswal
Assistant Professor
Dr. C.V. Raman University Bilaspur