VHDL Design and Implementation for Optimum Delay & Area for Multiplier & Accumulator Unit by 32-Bit Sequential Multiplier

SARITA SINGH[1], SACHIN MITTAL[2]
ECE Department, Mewar University Chittorgarh (Raj.)[1]R.D Engineering College Ghaziabad [2]

Abstract :- High performance systems such as microprocessors, digital signal processors, filters, ALU etc. which is need of hour now days requires a lot of components. One of main component of these high performance systems is multiplier. Most of the DSP computations involve the use of multiply-accumulate operations, and therefore the design of fast and efficient multipliers is imperative. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. This thesis investigates analysis of different multiplier for speed, area and delay usage. We try to present an efficient multiplier is produce fast, accurate and require minimum area. In this paper we will first study different types of multipliers: Then we compared the working of different multipliers by comparing the memory usage, speed and area by each of them. The result of this thesis helps us to choose a better option to choose a better multiplier out of different multipliers in fabricating different systems.

Keywords:**MAC,sequentialmultiplier,VHDL,Dataflow, waveform analyzer .**

I. INTRODUCTION

The addition and multiplication of two binary numbers is the fundamental and most often used arithmetic operation in microprocessors, digital signal processors, and data-processing application-specific integrated circuits. Therefore, binary adders and multipliers are crucial building blocks in VLSI circuits. High performance systems such as microprocessors, digital signal processors, filters, ALU etc. which is need of hour now days requires a lot of components. One of main component of these high performance systems is multiplier. Most of the DSP computations involve the use of multiply-accumulate operations, and therefore the design of fast and efficient multipliers is imperative. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. This paper investigates analysis of different multiplier for speed, area and propagation delay usage. We try to present an efficient multiplier is produce fast, accurate and require minimum area. In this thesis we will first study three different types of multipliers. Then we compared the working of different multipliers by comparing the different parameter by each of them. The result of this paper helps us to choose a better option to choose a better multiplier out of multipliers in fabricating different systems. . In this project, we have used VHDL as a HDL and XILLINX ISE for describing and verifying a hardware design based on Booth's and some other efficient algorithms. Timing and correctness properties were verified. Instead of writing Test-Benches & Test-Cases we used Wave-Form Analyzer which can give a better understanding of **Signals** & **variables** and also proved a good choice for simulation of design.

II.DIFFERNT MULTIPLER

A. Introduction of Multipliers

The main objective of design and implementation of a 32 Bit Sequential Multiplier. The programming objectives of 32 Bit Sequential Multiplier fall into following categories:

- Accuracy: The multiplier produces the correct result.
- Speed: The multiplier produces high speed.
- Area: The multiplier occupies less number of slices and LUTs.
- Power: The multiplier consumes less power.

Multiplication involves three main steps :

- Partial product generation
- Partial product reduction
- Final addition

For the multiplication of an *n*-bit multiplicand with an *m*-bit multiplier, *m* partial products are generated and product formed is *n* + *m* bits long.The multiplier architectures can be generally classified into following categories:

Sequential Multiplier
Booth Multiplier
Combinational Multiplier
Wallace Tree Multiplier

B. Booth Multiplier

Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallography at Birkbeck College in Bloomsbury, London. Booth used desk calculators that were faster at shifting than adding and created the algorithm to increase their speed. Booth's algorithm is of interest in the study of computer architecture.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**.

C. Combinational Multiplier

Basic Concept Throughout this section, we will look only at multiplication techniques for unsigned numbers. Alternatively, the hardware we present is suitable for sign and magnitude multiplication, but we concentrate on the manipulation of the magnitude part. Recall that the two numbers involved in a multiplication are called the *multiplicand* and the *multiplier*. Combinational Multipliers do multiplication of two unsigned binary numbers. Each bit of the multiplier is multiplied against the multiplicand, the product is aligned according to the position of the bit within the multiplier, and the resulting products are then summed to form the final result. Main advantage of binary multiplication is that the generation of intermediate products are simple: if the multiplier bit is a 1, the product is an appropriately shifted copy of the multiplicand; if the multiplier bit is a 0, the product is simply 0.

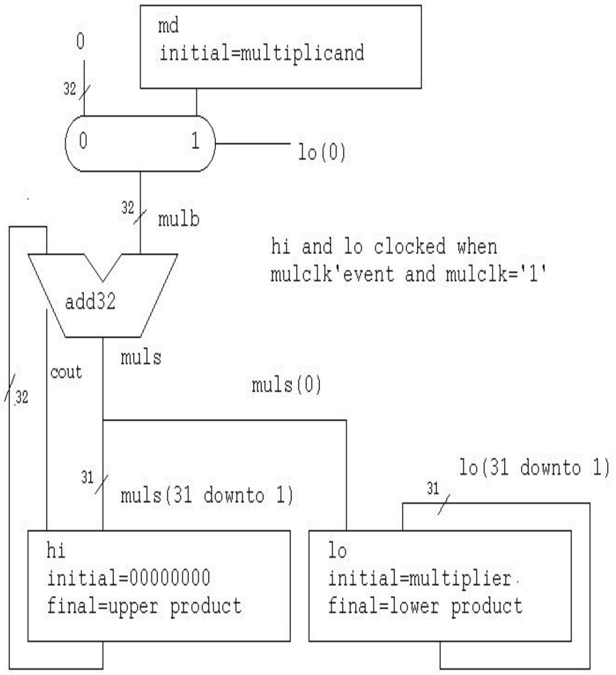
D. CSA Wallace-Tree Architecture:

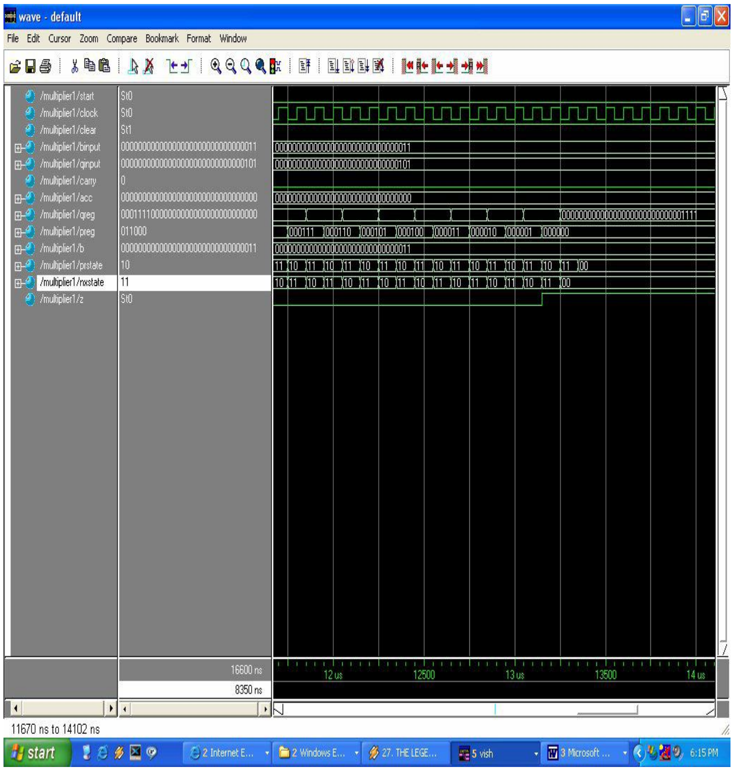
An unsigned multiplier using a carry save adder structure is one of the efficient Design inimplementation of Multipliers. Booth multiplier, two's complement 32-bit multiplicand by 32-bit multiplier input producing 64-bit product can be implemented using this special kind of Architecture.

D. 32 Bit Sequential Multiplier:

At the start of multiply: the multiplicand is in "md", the multiplier is in "lo" and "hi" contains 00000000. This multiplier only works for positive numbers. A booth Multiplier can be used for twos-complement values. The VHDL source code for a serial multiplier, using a shortcut model where a signal acts like a register. "hi" and "lo" are registers clocked by the condition mulclk' event and mulclk='1'.At the end of multiply: the upper product is in "hi and the lower product is in "lo."

A partial schematic of just the multiplier data flow is





```
/multiplier1/start      St0
/multiplier1/clock      St0
/multiplier1/clear      St1
/multiplier1/binput     00000000000000000000000000000011
/multiplier1/qinput     00000000000000000000000000000101
/multiplier1/carry      0
/multiplier1/acc        00000000000000000000000000000000
/multiplier1/qreg       00011110000000000000000000000000
/multiplier1/preg       011000
/multiplier1/b          00000000000000000000000000000011
/multiplier1/prstate    10
/multiplier1/nxstate    11
/multiplier1/z          St0
16600 ns
8350 ns
11670 ns to 14102 ns
```

III.RESULT

The results of efficient implementation of MAC unit with 32 bit sequential multiplier using VHDL design obtained from simulation and synthesis of various architectures of multipliers on the Xillinx 9.1i software.

A.32 Bit Sequential Multiplier: - Synthesis

```
->optimize .work.multiplier1.INTERFACE -target
xis2 -chip -auto -effort standard - hierarchy auto
-- Boundary optimization.
-- Writing XDB version 1999.1
-- optimize -single_level -target xis2 -effort standard
-chip -delay -hierarchy=auto
 Using wire table: xis215-6_avg
-- Start optimization for design
.work.multiplier1.INTERFACE
 Using wire table: xis215-6_avg
```

B.32 Bit Sequential Multiplier: - Simulation: wave form analyzer

IV CONCLUSION

The results obtained from simulation and synthesis of various architectures are compared and table 1. Sequential multipliers have maximum LUT in minimum area, minimum slack time increases the speed and have minimum delay due to constraints speed of clock in comparison to all the present multipliers. At First Instance It seems that combinational devices may work faster than the Sequential version of same devices. But this is not true in all the cases. In fact in complex system designing the sequential version of devises worked faster than the combinational multipliers. version because in sequential circuits the clock speed is constraint which does not get much affected from gate delays

S. No	Algorithms □ Performance/ Parameters	32-Bit Sequential Multiplier	Booth Multiplier	Combinational Multiplier	Wallace Tree Multiplier
1.	Optimum Area	110 LUTs	134 LUTs	4 LUTs	16 LUTs
2.	Optimum Delay	9 ns	11 ns	9.2 ns	9 ns
3.	Sequential Elements	105 DFFs	103 DFFs	----	----
4.	Input/Output Ports	67 / 71	50 / 49	4 / 4	24 / 18
5.	CLB Slices(%)	57(7.42%)	71(36.9%)	2(1.04%)	8(4.17%)
6.	Function Generators	114(7.42%)	141(36.72%)	4(1.04%)	16(4.17%)
7.	Optimum Clock	100	101.9	NA	100
8.	Slack time	0.89 ns	0.19 ns	Unconstrained path	1.48 ns

Different multiplier Comparison Table: 1

Asynchronous Problem is also a bigger drawback of the combinational circuits. So now days the computational part of systems are combinational and storage elements are sequential which is making a system robust, cheaper and highly efficient. From the 32 Bit sequential multiplier this drawback can be eliminated easily. In the future new algorithm can be developed to increasing the performance of the digital circuits devices

V. REFERENCES

[1]. John L Hennesy & David A. Patterson "Computer Architecture A Quantitative Approach" Second edition; A Harcourt Publishers International Company

[2]. C. S.Wallace, "A suggestion for fast multipliers," *IEEE Trans. Electron. Comput.*, no. EC-13, pp. 14–17, Feb. 1964.

[3]. M. R. Santoro, G. Bewick, and M. A. Horowitz, "Rounding algorithms for IEEE multipliers," in *Proc. 9th Symp. Computer Arithmetic*, 1989, pp. 176–183.

[4]. D. Stevenson, "A proposed standard for binary floating point arithmetic," IEEE Trans. Comput., vol. C-14, no. 3, pp. 51-62, Mar.

[5]. Naofumi Takagi, Hiroto Yasuura, and Shuzo Yajima. High-speed VLSI multiplication algorithm with a redundant binary addition tree. *IEEE Transactions on Computers, C-34(9)*, Sept 1985.

[6]. "IEEE Standard for Binary Floating-point Arithmetic", ANSUIEEE Std 754-1985, New York, The Institute of Electrical and Electronics Engineers, Inc., August 12, 1985.

[7]. Morris Mano,"Digital Design" Third edition; PHI 2000

[8.] J.F. Wakerly, *Digital Design: Principles and Practices*, Third Edition, Prentice-Hall, 2000.

[9] J. Bhasker, *A VHDL Primer*, Third Edition, Pearson, 1999.

[10]. M. Morris Mano,"Computer System Architecture", Third edition; PHI, 1993.

[11]. John. P. Hayes, "Computer Architecture and Organization", McGraw Hill, 1998.

[12]. G. Raghurama & T S B Sudarshan, "Introduction to Computer Organization", EDD Notes for EEE C391, 2003.