Available online at www.ijiere.com
International Journal of Innovative and Emerging Research in Engineering
e-ISSN: 2394 – 3343 p-ISSN: 2394 – 5494

64 Bit Signed Unsigned Multiplier Using VHDL

Mr. Mayuri Jagnale1, Prof. Ms. Pooja Thakre2
1PG Student, M.tech-IVth sem, Department of VLSI, NUVA College of Engg., Nagpur India
2HOD, Department of Electronics,NUVA College of Engg., Nagpur, India

ABSTRACT:
The 64-bit Signed/Unsigned Multiplier presents a Very high speed integrated circuit – Hardware Description Language (VHDL) based design and implementation of a fast unsigned multiplier. Booth's multiplication algorithm is a multiplication algorithm which is used to perform multiplication function for 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 algorithm used desk calculators that were faster at shifting than adding and created the algorithm to increase their speed. The multiplier uses a carry look ahead adder, which reduces the delay time caused by the effect of carry propagation through all the stages of a -carry adder. The main focus is on the enhancing the speed and complexity of the 64 bit Signed/Unsigned multiplier and it also provides a performance comparison between the fast multiplier and a ripple-carry adder based multiplier.

Keywords: Carry Look-Ahead Adder; Multiplier; VHDL Modeling & Simulation, signed/unsigned multiplier; Booth Encoding; Wallace Tree; fast adder.

I. INTRODUCTION

The multiplier is an essential which perform basic arithmetic operations hence it is important component of computer. However, complex multiplication not as an easy task asaddition or subtraction, because it takes more time to perform two suboperation, addition and shifting. Typically, a multiplication operation take between 8 and 2 cycles. Therefore, using high-speed multipliers is a crucial requirement for high presentation processors. Multipliers use the addition operation for all the partial products[1-2]. The adder can be a ripple-carry adder, a carry-look-ahead adder, or any extra adder. However, using a fast adder to complete multiplication operation improve the overall performance of the computer system.

Our study is alert on multipliers using unsigned data. VHDL, a Very high speed integrated circuit Hardware Description Language ,was[2] used to construction and model the multiplier design. Some researchers had addressed the adder performance issues and others did the same with view to the multiplier presentation.

Today, complex circuits are described in high-level description languages, like VHDL (VHSIC Hardware description language) or Vertilog, and synthesized to gate-level. A core operation in actual circuits, especially in digitalsignal processing such as Modulation, Filtering, or Neural Networks or Video Processing or Satellite Communication or Graphics or Control systems etc., is multiplication. Often, the computational performance of aDSP system is limited by its multiplication performance. This project presents fundamental of some multiplication algorithm including signed and unsigned multiplication and. Hardware multiplier multiplier sequential statements rather than implementing in any other higher language will give better speed in hardware multiplier implementation. Traditionally add algorithm and shift has been implement to devise but this is not suitable for VLSI implementation and also from delay point of view.

II. THEROTICAL STUDY

A. Adder

In modern computers adders exits in the arithmetic logic unit (ALU) where other operations are performedIn electronics, an adder is a digital circuit that performs addition of a numbersEven adders can be constructed for many arithmetic representations, such as excess-3 or Binary-coded decimal, the most common adders operate on binary numbers. In cases where two's complement is being used to signify negative numbers it is trivial to modify an adder into an adder-subtractor.modifyan adder into an adder-subtractor.

B.TypesOf Adder

1).Half Adder

A half adder is a logical circuit[3] perform an addition operation on two binary digits. The half adder produce a sum and a carry value which are both binary digits[3-4].

$$S = A \oplus B \quad (1)$$

$$C = A \cdot B \quad (2)$$

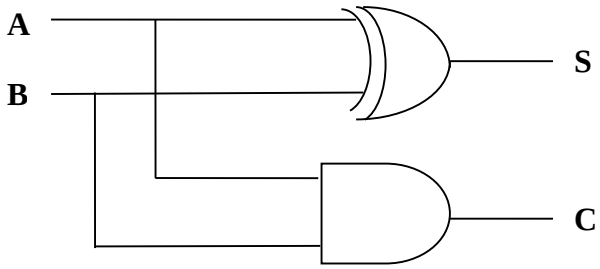

Figure.1 Half Adder[3]

2).Full Adder

A full adder is a logical circuit that[] perform an addition operation on three binary digits. The full adder[] produces a sum and carries value, which are both binary digit.

$$S = \left(A \oplus B\right) \oplus C_{in} \quad (3)$$

$$C_{out} + (B \cdot C_{in}) + (C_{in} \cdot A) \quad (4)$$

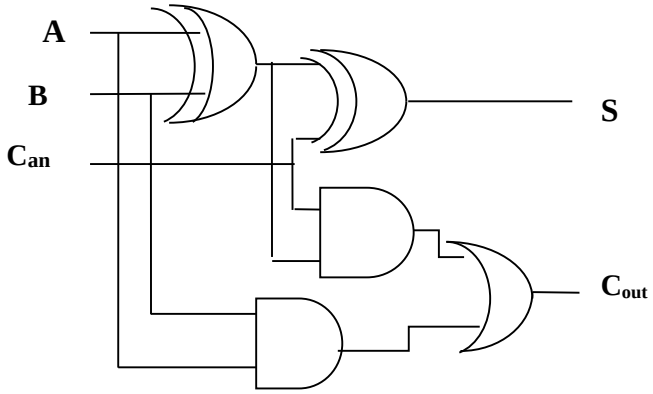

Figure.2 Full Adder[3]

A full adder can be construct from two half adders by between A and B to the input of one half adder, between the sum from that to an input to the second adder, connecting Cin to the other input and or the two carry outputs. Equivalently, S might be made the three-bit Xor of A, B, and Cin and Cout could be made the three-bit majority function of A, B, and Cin. The output of the full adder is the two-bit arithmetic sum of three one-bit[3-4] numbers.

3).Carry Look-AheadAdder

A carry-look-ahead adder is a kind of adder used in digital logic. A carry-look-ahead adder improve speed by falling the amount of time vital to determine carry bits. It can be contrasted with the simpler, butgenrally slower, ripple carry adder for which the carry bit is calculated beside the sum bit, and each bit must wait until the earlier carry has been designed to begin calculating its own result and carry bits. The carry look-ahead adder calculate one or more carry bits before the sum, which minimize the time reqiredto calculate the result of the larger value bits.

III. .MULTIPLIER

We have different techniques for multiplication. We can implement them in Xilinx software. The three multipliers which are implemented are:-

- Array Multiplier
- Booth Multiplier
- Wallace Tree Multiplier

We will be using an Algorithm which is faster i.e. Booth Algorithm to increase the performance of the multiplier.

A. Booth Multiplier

Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's balance data The algorithm was made-up by Andrew Donald Booth in 1951 while doing study on crystallography at Birkbeck College in Bloomsbury, London[6]. Booth used counter calculators that be faster at changing than adding and created the algorithm to improve their speed. Booth's algorithm is of importance in the study of computer architecture. One way to

speed up the multiplication is Booth encoding, which perform some ladder of multiplication at booth's algorithm looks at three bit of multiplier at a time to perform two stages of multiplication.

Booth's algorithm takes advantage of the detail that an adder subtractor is next to as fast and small as a simple adder. In the simple school algorithm, we shift the multiplicand X, then use one bit of multiplier Y if that shifted value is to be added into the partial product. The most regular form of booth's algorithm looks at three bit of multiplier at a time to perform two stages of multiplication.

The architecture of the proposed 64-bit signed/unsigned multiplier is shown in Fig. 3. The sign-control unit generate the MSBs of the multiplicand and multiplier and the choose signal for the line of multiplexer. Meanwhile, modified Booth encoding (MBE) is used to reduce the number of PPRs by a factor of two. After generating the PPRs, Wallace tree structures are used to efficiently add-up all PPRs in parallel. More specifically, [3:2], [4:2], [5:2] and [6:2] adders are combined to sum up all the PPRs pending only two rows are left. Carry Select adders are inserting in the second stage to decrees the third-stage long-length fast adder's delay; area and power without delay overhead. In the last step, a fast carry-propagation adder is used to add the final twoPPRs. The final adder is characterized by the fact that the input signals do not arrive simultaneously as a result of the Wallace tree compression. Ordinary single carry propagation adder design that suppose all the inputs enter simultaneously. A full adder combining both CSA and CCA is developed in the last stage. The control unit determine whether the multiplier operates on signed or unsigned numbers. This re-configurability results in a negligible 0.45% silicon area overhead.CSA, CCA and Carry Look-Ahead Adder (CLAs) can be used to implement the final fast addition. CLA is usually used and can be easily implement in dynamic CMOS logic with the check of full-custom design. For standard static CMOS circuit, CCA and CSA[6] are prefer and can easily be implemented using a standard cell library.In contrast to the CSA, CCA needs to use XOR logic to produce the final results. This translate in more delay as compared to a same bit-width CSA. The CSA needs to store both the conditional sum and carry together. As a result, more multiplexers are used than for a CCA.

IV. DESIGN FLOW

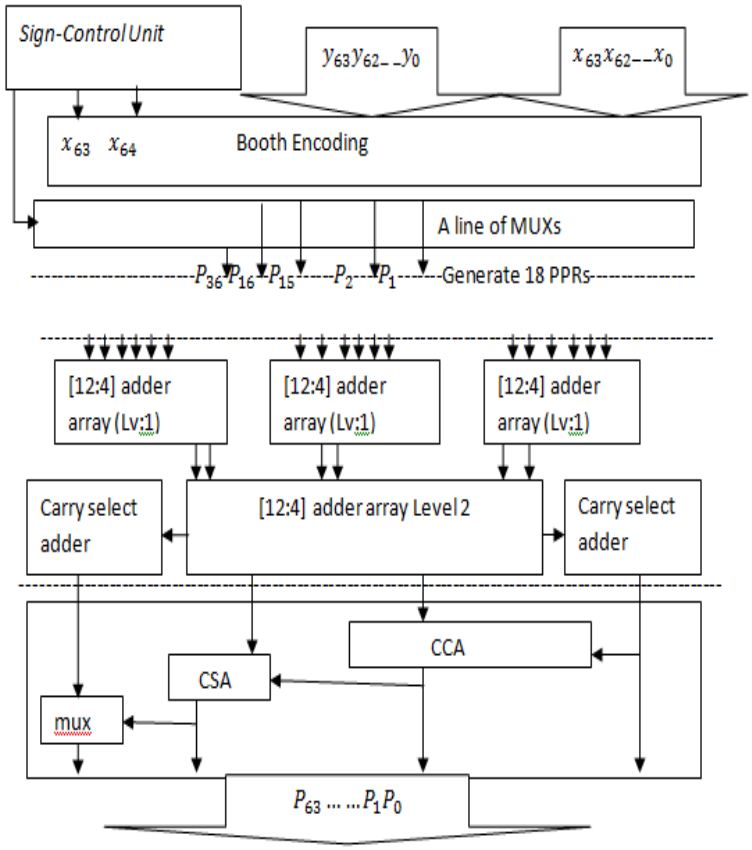

Figure 3. Architecture of the 64-Bit Signed/unsigned Multiplier

V.PPROPOSED WORK

64-bit Booth Multiplier

By implementing the above design we can see that the overflow bit is not required. The overflow bit shifts into the product register. To implement the 64 bit-register

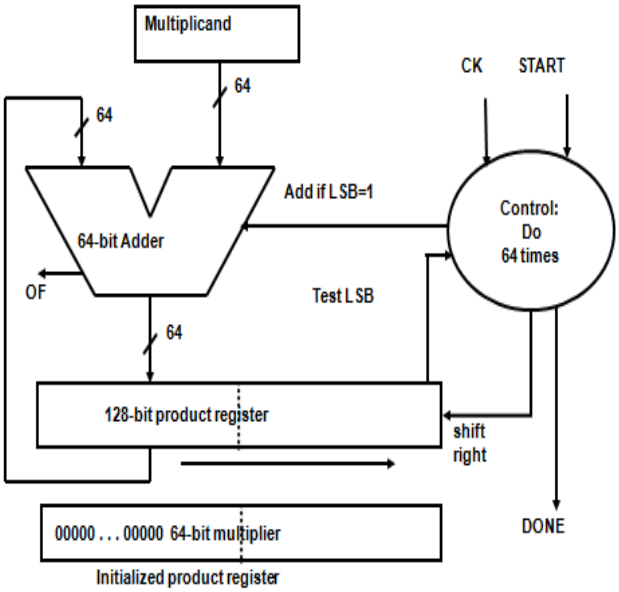

Figure 4. 64-bit Booth Multiplier

The multiplier result and the product register stored in the multiplier product register are shifted right by one bit. If the least significant bit of the multiplier product register is a '0', the bits in the multiplier product register are right shifted by one bit without the addition of the multiplicand product register. This is done 64 times. The result in the multiplier product register after 64 clock cycles is the final product

CONCLUSION

A low power 64 bit booth multiplier will be designed using VHDL and its performance will be compared with other available multipliers like parallel multiplier.

ACKNOWLEDGMENT

We take great pleasure & immense pride to present this project report on **"Low Power 64-Bit Signed/Unsigned Multiplier Using VHDL".** The sense of achievement, the satisfaction, the reward and the appreciation as regard to the completion of the report cannot be comprehended without the earnest support provided by various peoples associated with it.

REFERENCES

[1] R. Lyon, "2's Compliment Pipelined Multipliers", IEEE Transactions on Communication, April, 1976.
[2] Kai Wang, "Global and Modular two's Compliment Array Multipliers"IEEE Transactions on Computers, Vol-C-28 , issue 4, Apr 79.
[3] V. Oklobdzija, High speed VLSI arithmetic unit: Adders and Multipliers in "Design of high performance Microprocessor circuits. Ed. A. Chandrakasan, IEEE Press, 2000.
[4] ASIC Implementation of 4 Bit Multipliers" PravinkumarParate, IEEE Computer society.ICETET, 2008.25.
[5] E. E. Swartzlander. "Computer Arithmetic "IEEE Computer Society Press, 1990
[6] A.P. Chandrakasan, S. Sheng and R.W. Brodersen, "Low-power CMOS digital design," IEEE J. Solid-State Circuits, vol.27, no.4, pp.473-484, Apr.1992
[7] Chang C. H., GU J., Zhang M, "Ultra low-voltage low-powerCMOS 4-2 and 5-2 compressors for fast arithmetic curcuit IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 51, Issue 10, and pp.:1985 - 1997, oct-2014