ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

IMPLEMENTATION OF LOW POWER AND LESS AREA BOOTH MULTIPLIER

Sonia Sharma*, Anshul Soni
* AISECT UNIVERSITY, BHOPAL
AISECT UNIVERSITY, BHOPAL

DOI: 10.5281/zenodo.165633

ABSTRACT

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 power of the multiplier is an important issue, increment in speed and power 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.

KEYWORDS: Booth's, Algorithm, multiplication, multipliers.

INTRODUCTION

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, times that it is added is the multiplier, and the result is the product [8]. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. Product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an Algorithm that makes use of

GUIDE NAME- ANSHUL SONI
INSTITUTE NAME- AISECT UNIVERSITY, BHOPAL positional representation. It is possible to decompose multipliers into two parts to the generation of partial products, and the second one collects and adds them. The basic multiplication principle is twofold, i.e. evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive Additions of the columns of the shifted partial product matrix.[9] The multiplier is successfully shifted and gates the appropriate bit of the multiplicand The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement format. Multipliers are key components of many high performance systems such as FIR Filters, microprocessors, digital signal processors,[10] etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in The system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, whole spectrums of multipliers with different area-speed constraints are designed with fully parallel processing. In between are digit serial multipliers where single digits consisting of several bits are operated on. These multipliers have moderate performance in both speed and area. However, existing digit serial multipliers have been plagued by complicated switching systems and/or irregularities in design. Radix 2^n multipliers which operate on digits in a parallel fashion instead of bits bring the pipelining to the digit level and avoid most of the above problems. They were introduced by M. K. Ibrahim in 1993. These structures are iterative and modular. The pipelining done at the digit level brings the benefit of constant operation speed irrespective of the size of‟ the multiplier.[7]

BOOTH'S MULTIPLICATION ALGORITHM

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. 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

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).
2. A: Fill the most significant (leftmost) bits with the value of m. Fill the remaining (y + 1) bits with zeros.
3. S: Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining (y + 1) bits with zeros.
4. 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.
5. 4 Determine the two least significant (rightmost) bits of P.
6. If they are 01, find the value of P + A. Ignore any overflow
7. If they are 10, find the value of P + S. Ignore any overflow.
8. If they are 00, do nothing. Use P directly in the next step.
9. If they are 11, do nothing. Use P directly in the next step.
10. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.
11. Repeat steps 2 and 3 until they have been done y times.
12. Drop the least significant (rightmost) bit from P. This is the product of m and r.

DESIGN OF BOOTH MULTIPLIER

Booth's algorithm can be implemented in many ways. This experiment is designed using a controller and a data path. The operations on the data in the data path are controlled by the control signal received from the controller.[3] The data path contains registers to hold multiplier, multiplicand, intermediate results, and data processing units like ALU, adder/subtractor etc., counter and other combinational units Following is the schematic diagram of the Booth's multiplier which multiplies two 4-bit numbers in 2's complement of this experiment.

Here the adder/subtractor unit is used as data processing unit M, Q, A are 4-bit and Q-1 is a 1- bit register. M holds the multiplicand, Q holds the multiplier, A holds the results of adder/subtractor unit. The counter is a down counter which counts the number of operations needed for the multiplication. The data flow in the data path is controlled by the five control signals generated from the controller. These signals are *load* (to load data in registers), *add* (to initiate addition operation), *sub* (to initiate subtraction operation), *shift* (to initiate arithmetic right shift operation), *dc* (this is to decrement counter).

The controller generates the control signals according to the input received from the data path. Here the inputs are the least significant Q0 bit of Q register, Q-1 bit and count bit from the down counter.[7]

PERFORMANCE

The method of Booth multiplication reduces the numbers of adders and hence the area required to produce the partial sums. The high performance of booth multiplier comes with the drawback of power consumption. The reason is large number of adder cells required that consumes large power [5]. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better.

IMPLEMENTATION AND SIMULATION

Booth's algorithm can be implemented in many ways. This experiment is designed using a controller and a data path. The operations on the data in the data path are controlled by the control signal received from the controller Once the all VHDL modules are prepared, they should be simulated before they are put in actual Hardware chip. We can generate a test counter waveform from the Project, New Source menu of ISE and it will support in setting up the simulation. Once we simulate our design and feel it is functioning properly

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

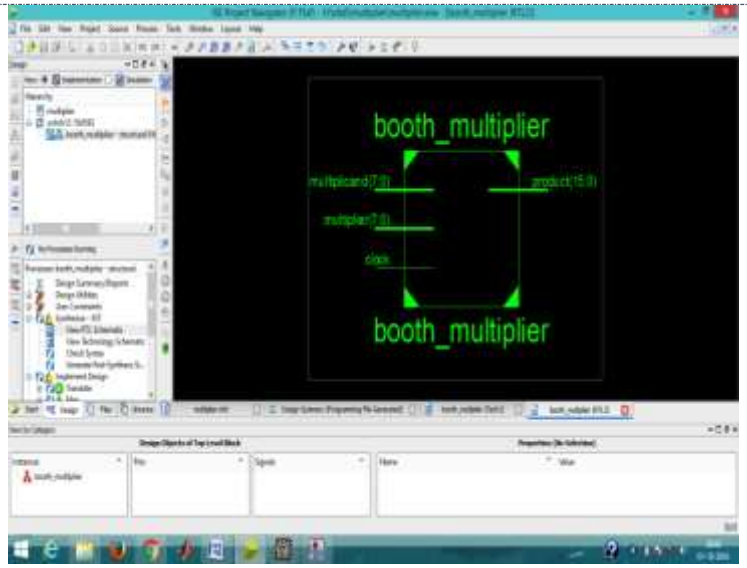

Fig. 1 Shown RTL view of 7 bit booth multiplier

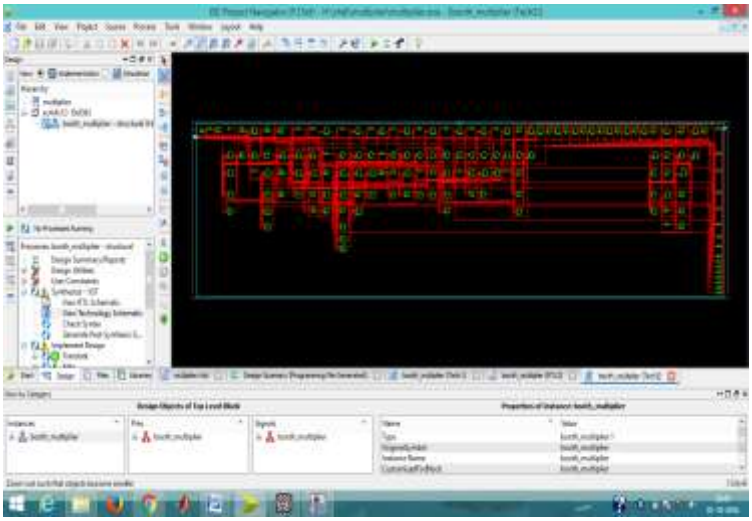

Fig. 2 Shown RTL view of 7 bit booth multiplier

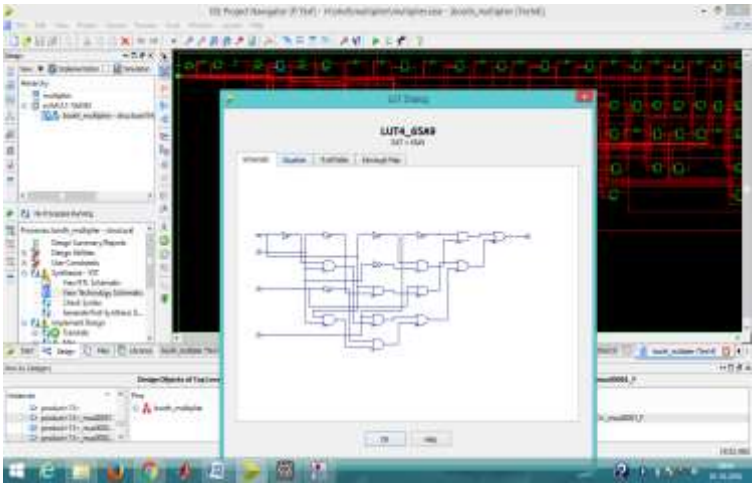

Fig. 3 Shown LUTs view of 7 bit booth multiplier

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

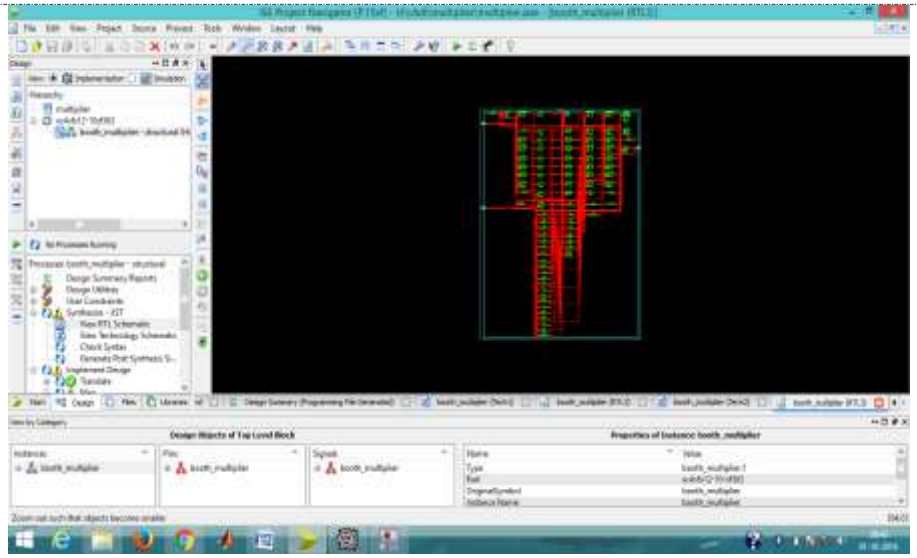

Fig. 1 Shown technology view of 7 bit booth multiplier

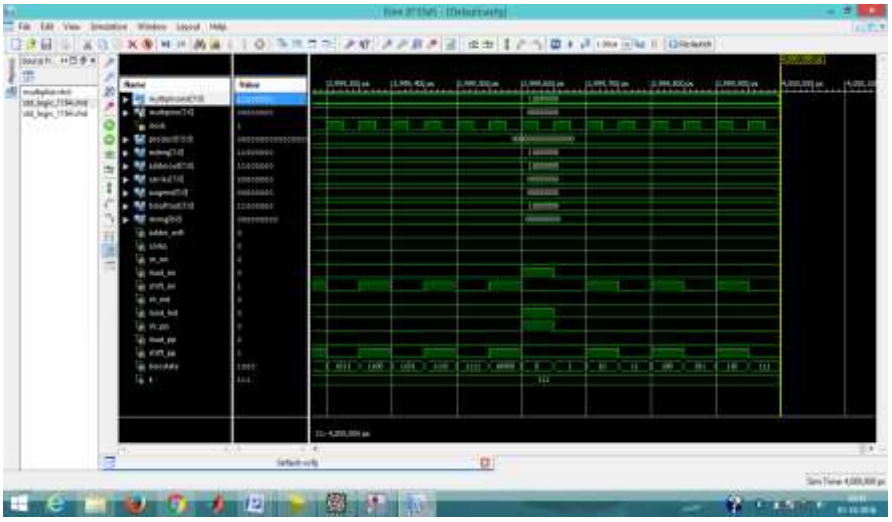

Fig. 1 Shown simulation of 7 bit booth multiplier

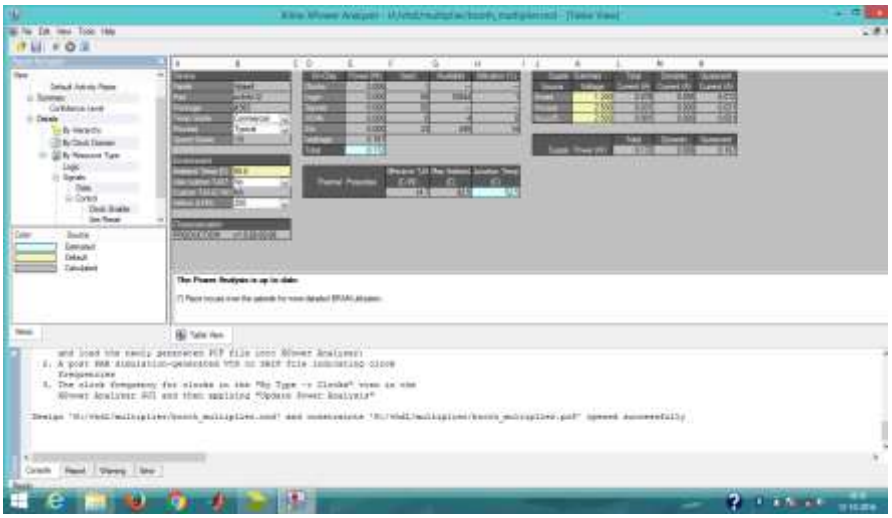

Fig. 1 Shown power and area of 7 bit booth multiplier

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

CONCLUSION

This from the result of power consumption and the total area is reduced, Booth's algorithm can be implemented in many ways Hence the power consumption is also less. This is clearly depicted in our results. This speeds up the calculation and makes the system faster. Table 1 show the power consumption and area reduced as previous work.

Table 1 Comparison Table

Architecture	Area	Power(w)
previous	2558	3.83
Proposed	1%(74 out of 10944 LUTs)	0.172

REFERENCES

[1] Tso-Bing Juang and Yu-Ming Chiu High-Speed Arithmetic LAB, Department of Computer Science and Information Engineering, National Pingtung Institute of Commerce (NPIC), Taiwan
[2] J. M. Rudagi, V. Amblr, V. Munavalli, R. Patil, V. Sajjan, "Design and implementation of efficient multiplier using Vedic Mathematics," in *proc. IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, *November* 2011, pp. 162-166.
[3] T. Arunachalam, S. Kirubaveni, "Analysis of High Speed Multipliers," in *proc. IEEE International Conference on Communication and Signal Processing*, April 2013, pp. 211-214.
[4] *Communication and Applications*, June 2011, pp. 854-859.K. S. Yeo, K. Roy, Low Voltage, Low Power VLSI, Tata Mcgraw-Hill Education, 2nd
[5] J. Sharma, S. Kumar, "Digital Multipliers-A Review," *International Journal of Engineering and Management (IJEMR)*, June 2014, Vol. 4, No. 3.
[6] D. Garg, S. Arya, "Design of Configurable Booth Multiplier Using Dynamic Range Detector," *International Journal of Electronics*
[7] S. R. Kuang, J. P. Wang, "Design of power
[8] efficient configurable booth multiplier",
[9] *IEEE International Transaction on Circuits and Systems*, March 2010, Vol. 57, No. 3.
[10] J. W. Townsend, A. J. Abraham, E. Swartzlander, "A comparison of Wallace and dadda multiplier delays," *International Society for Optical Engineering (SPIE)*, December 2003, Vol. 52, No. 5
[11] S. Shah, A. J. Khabb, D. A. Khabb, "Comparison of 32-bit Multipliers for VariousPerformance Measures," in *proc. IEEE International Conference on Microelectronics*, November 2000, pp. 75-80.
[12] Behrooz Parhami, Computer Arithmetic,Algorithms and Hardware Design, Oxford University Press, 2000.
[13] K. L. S. Swee, L. H. Hiung, "Performance Comparison Review of 32 bit Multiplier Designs", in *proc. IEEE International Conference on Intelligent and Advanced Systems*, June 2011, pp. 854-859.