ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

A High Speed Booth Wallace Multiplier Using Pipelining Technique

Juny Mary Jose[1], Reen Paul[2]

PG Student [VLSI], Dept. of ECE, College of Engineering, Munnar, Kerala, India [1]

Assistant Professor, Dept. of ECE, College of Engineering, Munnar, Kerala, India [2]

ABSTRACT: Multipliers have great importance in both digital signal processors and microprocessors. So designing a high speed multiplier is the need of the hour. There are several methods available to speed up a multiplier. This paper incorporates pipelining technique to a multiplier for improving its performance. The multiplier under consideration is Booth Wallace multiplier. A comparison between pipelined and non-pipelined booth Wallace multiplier in terms of delay and area utilization were also done in this work. Verilog HDL has been used for the coding. Xilinx ISE 14.2 design suite is used for synthesizing the code.

KEYWORDS: Pipelining, Booth Wallace, Xilinx.

I. INTRODUCTION

In a DSP processor for performing FFT, DCT, convolution etc. multipliers are important. Speed of such systems are dependent on the speed of the multiplier used, because it is the slowest element. A multiplication includes three steps. First step is the generation of the partial products then comes the reduction of the partial product to two rows-a sum row and a carry row. The final step is the addition of the two rows to obtain the product. This paper deals with one of the fastest multiplier namely Booth Wallace multiplier. This multiplier incorporates the advantages of booth multiplier and Wallace multiplier. Normal methods to improve the performance of this multiplier are:

- Improving radix
- Usage of compressors
- A faster adder for final addition

Another method of improving the performance is introducing pipelining. In this work along with the above said methods pipelining technique is also utilized for the speed improvement. Three stage pipelining is used for this work. The rest of this paper is organized as: section 2 deals with literature survey, section 3 deals with booth Wallace multiplier next sections gives a brief description about pipelining and pipelined booth Wallace multiplier, section 6 includes the work done and then followed by analysis of simulation results, finally concluding the work by section 8.

II. LITERATURE SURVEY

Several types of multipliers are available like serial multiplier [1], parallel multiplier [2], serial parallel multiplier. In a serial multiplier both the operands are given serially and also serial addition is used there. The unhealthy factor associated with this multiplier is the speed. Moving to serial parallel multipliers there multiplier is given serially and multiplicand is given in parallel. In this multiplier the speed depends on the addition of column of the partial products. Each column is added in each clock cycle. Therefore for an n*n bit operation it takes 2n clock cycles. Our area of interest is parallel multiplier, where both the operands are given at the same time. There the speed depends on the number of partial products to be added. Classification of parallel multipliers includes array and tree multipliers. Among array multipliers, the braun multiplier and baugh wooley [5] consumes more area. Braun multiplier is normally used for low power applications. It performs only with unsigned numbers. Both signed and unsigned numbers can be handled with baugh wooley multiplier. Then comes the booth multiplier, also handles both signed and unsigned numbers. It takes inputs as twos compliment numbers. Radix 2 booth multiplier is inefficient in the case of isolated 1s.Normally booth multiplier concentrates on partial product generation stage for improving the speed. Moving to modified booth algorithm, reduces the number of partial products, thereby improves the speed of operation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

Then comes the tree multipliers which includes Wallace and dadda tree. These multipliers concentrate on the partial product reduction stage for improving the performance. Normally they uses full adders and half adders for reduction of partial products to two rows. In Wallace tree[8] the partial products are divided into groups, each group consisting of three rows. The row which doesn't belong to any group will move to the next stage. The column having three numbers is passed through full adders and two is passed through half adders. This process is repeated until two rows are obtained. In the case of dadda tree initially $d_1=2$;then $d_{j+1}=\text{floor}(1.5\ d_j)$.First find the largest d_j that is less than maximum number of bits in any column. Then for every column , use full adders and half adders to make sure that the number of elements in each column will be $\leq d_j$. Actually Wallace tree performs a faster reduction whereas dadda tree provides only slower reduction. Also dadda tree requires a wider final adder. Incorporating the advantages of Wallace and booth multiplier, booth Wallace multiplier is generated. .A booth Wallace multiplier has three modules booth encoder and partial product generator, Wallace tree and a final adder. Work can be done on all these modules for improving the performance. Normal methods are improving the radix in the booth algorithm used, then using higher order compressors in Wallace tree then using a faster adder. In this paper along with these methods for further speed improvement pipelining is also incorporated. Three stage pipelining is used for this work.

III. BOOTH WALLACE MULTIPLIER

Booth Wallace multiplier includes three modules. Booth encoding and partial product generation, then the Wallace tree for partial product reduction stage then for final addition a carry look ahead adder. Inorder to overcome the disadvantages of radix 2 booth algorithm modified booth algorithm is used, where more number of bits is considered at a time.Radix4 scans three bits at a time.Radix8 scans four bits at a time and so on. In the booth algorithm, multiplier is grouped in an overlapped manner initially, means the MSB of one group will become LSB of next group. Also before grouping add a zero as the LSB of the multiplier. Then based on each group the multiplier value is encoded. Each group is encoded to the range of {-2,-1,0,1,2} in the case of radix 4 and to the range of {-4,-3,-2,-1,0,1,2,3,4} in the case of radix 8.

Eg: 25*-10 = (00011001)*(11110110), taking the multiplier value as 111101100 then grouping is done in an overlapped manner. Each group and its encoded value using radix 4 are given below:

- 100 → +2X →000110010
- 011 → +2X →000110010
- 110 → -1X →11100111
- 111 → 0X → 00000000

By using radix4 booth encoding 4 partial products are generated. Actually inputs to theWallace tree is these values MSB extended to the bit width of the product. Now if encoding the same inputs using radix8 the encoded value is shown below:

- 1100 → -2X ⟶111001110
- 1101 → -1X ⟶11100111
- 1111 → 0X ⟶00000000

So by using radix 8 only three partial products are generated. By moving to higher order radix speed can be increased. Block diagram of booth Wallace multiplier is shown in figure1.After generating the partial products for the reduction of the partial products Wallace tree is used. Compressors can be used for the faster reduction. Compressors are nothing but cascading of full adders. A full adder is called 3:2 compressors. It reduces 3 inputs to 2 inputs. Higher order compressors [6] are also available like 4:2, 5:2 etc.4:2 compressor is just a cascade of two full adders. It actually compresses 5 inputs to 3 inputs.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

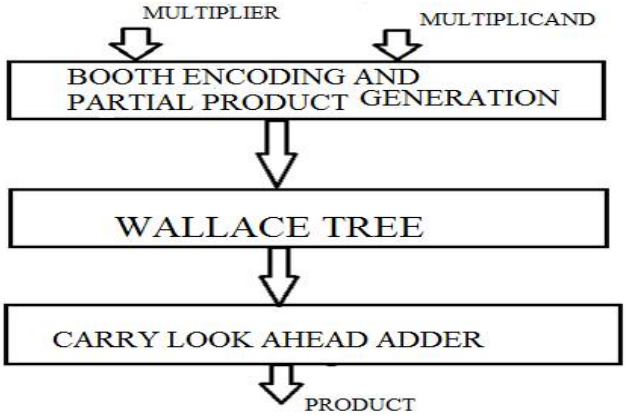

Fig. 1: Booth Wallace Multiplier

In Wallace tree addition is done in parallel way. Also more numbers are added at a time. In the case of 4:2 compressor sum[3] will go to the same position whereas the Carry and C_{out} goes to the next higher position.

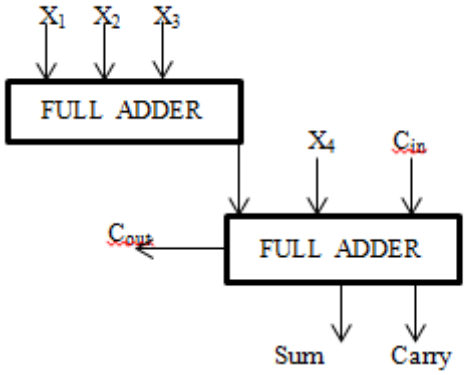

Fig. 2: 4:2 compressor

Carry look ahead adder is used for this work. This adder overcomes the problem of ripple carry adder. It includes a generate $G_i = X_iY_i$ and propagate term $P_i = X_i$ xor Y_i for finding carry. Carry, $C_{i+1} = G_i + P_iC_i$.

IV. PIPELINING

Pipelining [7] is a technique which is normally used in the case of advanced microprocessors inorder to improve its performance. Linear pipelining is used in this work .This technique is used in the cases where the stages are connected in a linear manner. By comparing a pipelined and a non-pipelined processor the advantage with pipelined processor is that it take only lesser number of clock cycles to finish all the tasks. But initially there will be latency. This latency is due to the presence of registers that are inserted in between the stages. In the case of pipelining the clock period is an important parameter. It is determined by the stage with maximum delay. In this work it is the carry look ahead adder. The data flow in a pipelined architecture can be synchronous and asynchronous. In the case of synchronous the data flow simultaneously, where as in asynchronous it is based on some handshaking protocols.

V. PIPELINED BOOTH WALLACE MULTIPLIER

Three stage pipelining is used for this work. In pipelining result of each stage is registered, so that it can be utilized by the next stage.Figure3 shows a pipelined architecture of booth Wallace multiplier. With pipelining arithmetic operations are performed simultaneously. During the first clock cycle first module works with first set of inputs, during the second clock cycle the second module, Wallace tree works with first set of inputs at the same time booth encoder

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

operates with second set of inputs. By the third clock cycle the first set of inputs will be handled by the carry look ahead adder, at this time Wallace tree is doing with second set of inputs and booth encoder with third set of inputs. At the third clock cycle first output is obtained which is followed by the other outputs.

VI. WORK DONE

8 bit booth Wallace multiplier with and without pipelining is done. It is done by using radix8 .This multiplier produces only three partial products. So 3:2 compressors were only used in the Wallace tree. Then the multiplier is analyzed by improving its bit width.

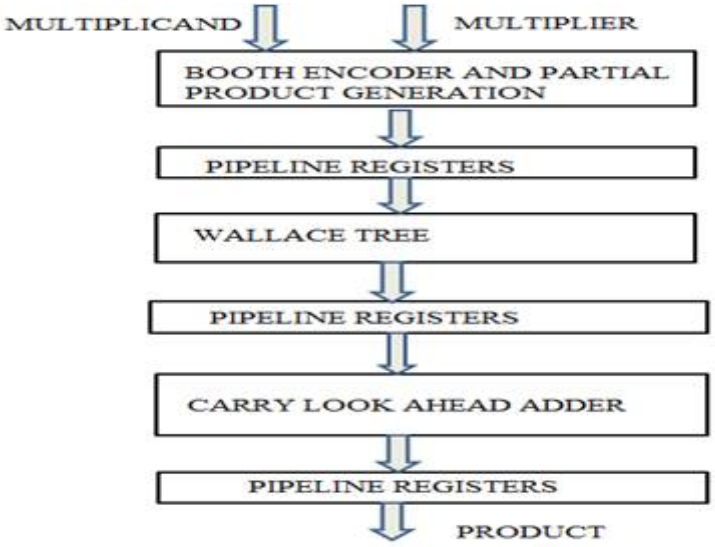

Fig. 3: Pipelined Booth Wallace Multiplier

Worked on 16 and 32 bit also. In the case of 16 bit multiplier radix 8 is used and 6 partial products are generated. There also in the Wallace tree 3:2 compressors were used. With 32 bit multiplier by using radix8 almost 11 partial products will be generated, working with that is a tedious task. So there an improvement in radix is made.32 bit is analyzed with radix16, which produces only 8 partial products. There in the Wallace tree analysis were done by using 3:2 and 4:2 compressors also. In this multiplier by incorporating 4:2 multiplier there is a reduction in the number of stages in the partial product reduction step.

VII. RESULT AND DISCUSSION

Analysis were done on Xilinx ISE design suite 14.2.FPGA used is SPARTAN3E, xc3s500e-5ft256 device is considered. In Table 1 shows the delay analysis of 8 bit and 16 bit using radix 8.Table 2 shows the delay analysis of 32 bit multiplier using different compressors.

Table 1: Delay Analysis

Multiplier	8 bit(ns)	16 bit(ns)
Without Pipelining	24.651	41.101
With Pipelining	18.704	21.416

.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

Table 2: Delay Analysis of 32 bit multiplier

Multiplier	32bit using 3:2 compressor(ns)	32bit using 3:2 and 4:2 compressors(ns)
Without Pipelining	75.925	69.818
With Pipelining	28.213	25.238

Table 3: Logic Utilization

Multiplier	No. of slices(4656)	No. of 4 input LUT(9312)
8 bit non pipelined	97	117
8 bit pipelined	110	186
16 bit non pipelined	430	810
16 bit pipelined	453	859
32 bit non pipelined	2513	4525
32 bit pipelined	2624	4711

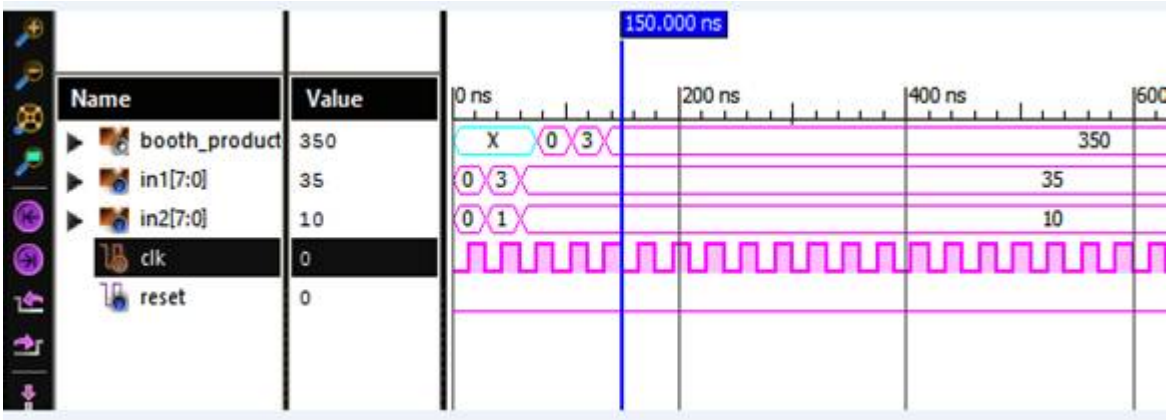

Fig. 4: Simulation of 8 bit pipelined multiplier

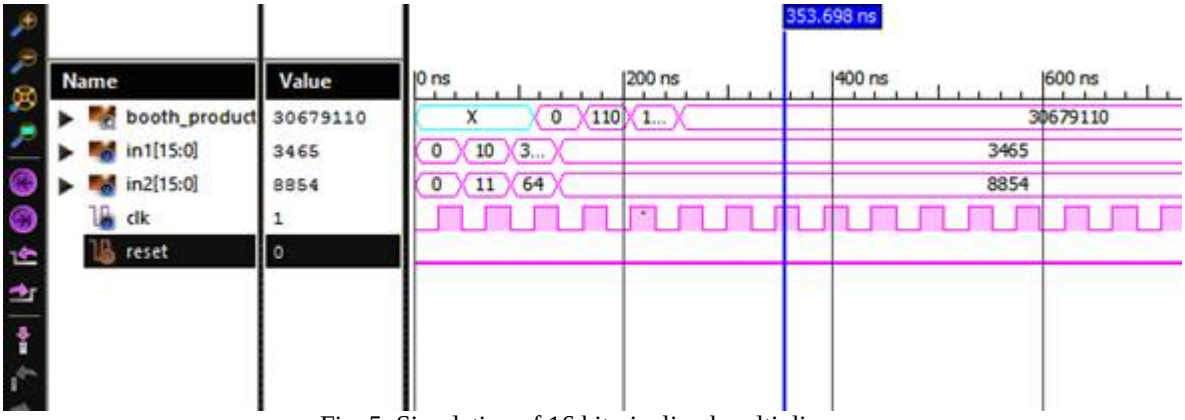

Fig. 5: Simulation of 16 bit pipelined multiplier

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

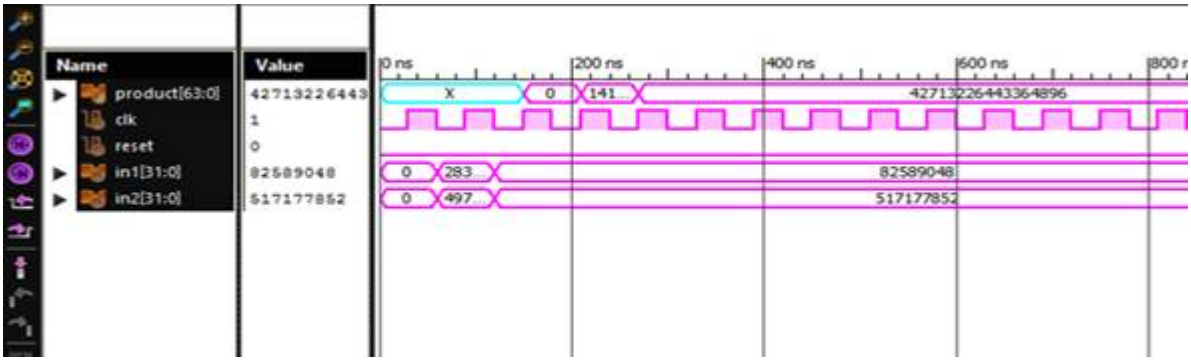

Fig. 6: Simulation of 32 bit pipelined multiplier

VIII.CONCLUSION

A comparison between pipelined and non-pipelined multiplier were done in terms of area and delay. SPARTAN3E FPGA is used for the analysis and the device under consideration is xc3s500e-5ft256 .Pipelined multiplier has high speed compared with non-pipelined at the cost of area. Also by incorporating higher order compressors high speed is achieved.

REFERENCES

[1] Akhter, Shamim, and Saurabh Chaturvedi. "HDL based implementation of N× N bit-serial multiplier." *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*,2014 International Conference on.IEEE,2014,PP.470-474.

[2] Wen, Ming-Chen, Sying-Jyan Wang, and Yen-Nan Lin. "Low power parallel multiplier with column bypassing." *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE, 2005..

[3] Sharma, Neeta, and Ravi Sindal. "Modified Booth Multiplier using Wallace Structure and Efficient Carry Select Adder." *International Journal of Computer Applications* 68.13 (2013): 39-42.

[4] Kshirsagar, Rahul D., et al. "Implementation of pipelined Booth Encoded Wallace tree Multiplier architecture." *Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on*. IEEE, 2013.

[5] Själander, Magnus, and Per Larsson-Edefors. "High-speed and low-power multipliers using the Baugh-Wooley algorithm and HPM reduction tree."*Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*. IEEE, 2008.

[6] Jagadeshwar Rao, M., and Sanjay Dubey. "A high speed wallace tree multiplier using modified booth algorithm for fast arithmetic circuits." *IOSR Journal of Electronics and Communication Engineering (IOSRJECE)* 3.1 (2012): 07-11.

[7] Hamacher, Carl, Zvonko Vranesic, and Safwat Zaky. *Computer organization*. McGraw-Hill, 2002.

[8] C. S. Wallace, "A suggestion for a fast multiplier," Electronic Computers, IEEE Trans-actions on, no. 1, pp. 14–17, 1964.