

Design and Implementation of High Performance Multiplier Using HDL

Prajakta P. Chaure¹, G. D. Dalvi²

¹Electronics and Telecommunication Engineering, P.R. Pote (Patil) College of Engineering and Management, Amravati

²Professor, Electronics and Telecommunication Engineering, P.R. Pote (Patil) College of Engineering and Management, Amravati

Abstract: This represents an area efficient implementation of a high performance parallel multiplier. Radix-4 Booth multiplier with 3:2 compressors and Radix-8 Booth multiplier with 4:2 compressors are presented here. The design is structured for $m \times n$ multiplication where m and n can make up to 126 bits. Carry Look ahead Adder is used as the final adder to enhance the speed of operation. Finally the performance development of the proposed multipliers is validate by implementing a higher order FIR filter. The design entry is done in VHDL and simulated using ModelSim SE 6.4 design suite from Mentor Graphics. It is then synthesized and implemented using Xilinx ISE 9.2i targeted towards Spartan 3 FPGA. The multiplier is an essential element of the digital signal processing such as filtering and convolution. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). In the majority of digital signal processing (DSP) applications of critical operations are the multiplication and accumulation. In this paper we are implementing the other thing is the real time signal processing required high speed high throughput multiplier. This paper also investigates on various architecture of multiplier and adder which are suitable for throughput signal processing at the same time to achieved low power consumptions developed This project by various systems which are too difficult to implement. But we have design and implement this multiplier with very high performance using booth algorithm. The elements in FPGA which describe the complexity of design, which is reduced considerable. For this system development using wallace tree, carry look ahead adder, which giving high power and reducing the main factor which is time delay. Using this tools and different block we implementing the proper results as shown. So next we described the proposed work and system architecture which is based on booth algorithm and VHDL.

Keywords: FPGA, HDL, Carry Look ahead Adder, Carry Save adder, Wallace Tree, Booth Encoding

1. Introduction

With the fast advances in multimedia and communication systems, real-time signal processing and large capacity data processing are increasingly being demanded. The multiplier is an essential element of the digital signal processing such as filtering and complication. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). As they are basically proficient by recurring application of multiplication and addition, their speed becomes a major factor which determines the performance of the entire calculation. Since the multiplier requires the longest delay among the basic prepared blocks in digital system, the critical path is determined more by the multiplier. Furthermore, multiplier consumes much area and dissipates more power. Hence designing multipliers which offer either of the following design targets – high speed, low power consumption, less area or even a mixture of them is of substantial research interest. Multiplication operation involve generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products and/or accelerating the accumulation of partial products. Among the many methods of implementing high speed similar multipliers, there are two basic approaches namely Booth algorithm and Wallace Tree compressors.

This paper describes an efficient implementation of a high speed parallel multiplier using both these approaches. Here two multipliers are proposed. The first multiplier makes use of the Radix-4 Booth Algorithm with 3:2 compressors while

the second multiplier uses the Radix-8 Booth algorithm with 4:2 compressors. The design is structured for $m \times n$ multiplication where m and n can reach up to 126 bits. The number of partial products is $n/2$ in Radix-4 Booth algorithm while it gets reduced to $n/3$ in Radix-8 Booth algorithm. The Wallace tree uses Carry Save Adders (CSA) to accumulate the partial products. This reduces the time as well as the chip area. To further enhance the speed of operation, carry-look-ahead (CLA) adder is used as the final adder.

2. Literature Review

From the rigorous review work and published literature, it is observed that many researchers have design on implementation of high performance multiplier. In this paper we discuss an illustrate a number of approaches, these related work have been mentioned as follows:

Sumit.R.Vaidya and D.R.Dandekar's paper described on performance comparison of multipliers for power-speed-trade-off in VLSI design. Multiplication is an important fundamental function in arithmetic logic operation. It can be concluded that Booth Multiplier is superior in all respect like speed, delay, area, complexity, power consumption. However Array Multiplier requires more power consumption and gives optimum number of components required, but delay for this multiplier is larger than Wallace Tree Multiplier. Hence for low power requirement and for a lesser amount of delay requirement Booth's multiplier is suggested. Ancient Indian Vedic Mathematics gives efficient algorithms or formulae for multiplication which increase the speed of devices. [1].

Shanthala S and S,Y,kulkarni has explained the VLSI design and implementation of low power MAC unit with block enabling technique. In the majority of digital signal processing (DSP) applications the dangerous operations usually absorb many multiplications and/or accumulations. A 8x8 multiplier-accumulator (MAC) is presented in this work. A full-adder circuit based on mux is used for MAC architecture. Compared to other full-adder circuits, the MUX based full adder has the highest operational speed and less transistor count. The basic building blocks for the MAC unit are recognized and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. With this block, the N-bit MAC unit is constructed and the total power consumption is intended for the MAC unit. With power reduction techniques adopted in this work, 27% of power is saved. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications. [2].

Manoranjan Pradhan, Rutupama Panda, Shushanta Kumar Sahu has worked on "MAC accomplishment using Vedic multiplication algorithm". They conclude that A conventional MAC unit consists of multiplier and an accumulator that contains the sum of the previous consecutive products. The 16x16, and 32x32 bit proposed MAC module show improved speed improvements over optimized Vedic multiplier architecture presented in [1]. Hence, the proposed MAC may be useful for elevated performance DSP processor. [3].

Padma Devi, Ashima Gridher, and Balwindar Singh has proposed the work "improved carry select adder with reduced area and low power consumption" explained that slowest in speed. These adders are more rapidly than ripple carry adders but slower than carry select adders. All the adders are designed using VHDL (Very High Speed Integration Hardware Description Language), Xilinx Project Navigator 9.1i is worn as a synthesis tool and ModelSim XE III 6.2g for simulation. FPGA Spartan3 is used for implementing the designs. Wherever there is need of smaller area and low power consumption, while a few raise in delay is tolerated, such designs can be used. These adders are faster than RCA and slower than CSA. [4]

3. Proposed Work

We proposed the project on design and implementation of high performance multiplier using VHDL system. The architecture of the proposed multiplier is shown in fig.1. It consists of four major modules: booth encoder, partial product generator, Wallace tree, carry look-ahead adder, booth encoder performs Radix2 and radix 4 encoding of the multiplier of the bits. Based on the multiplicand and the Encoded multiplier, partial products are generated by the producer. For large multiplier of 32 bits, the performance of the modified booth algorithm is limited. So booth recording together with Wallace tree structures have been used in the fast multiplier. The incomplete products are bring to Wallace tree and added appropriately. The result are finally Added

using a carry look-ahead adder (CLA) to get the final products.

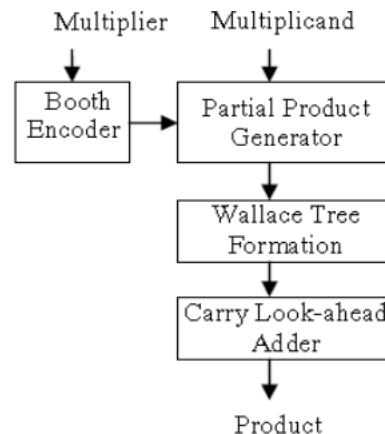


Figure 1: Block Diagram of Proposed multiplier.

Radix 4 booth algorithm is powerful algorithm for sign number multiplication, which treats of both positive and negative numbers uniformly. Since k bit binary number can be interpreted as k/2 digit radix 4 number, as k/3 digit radix 8 number as so on, it can be contract with more than one bit of the multiplier in each series by using high radix. Radix 8 booth copy applies same algorithm as that of radix 4. Radix 8 booth trim down the number of partial products to n/3, where n is the number of multiplier bit. The Wallace tree process is worn high speed design in order to produce two row of partial product that can be added in last stage. Wallace tree has in use the role of accelerating the accumulation of last products. The speed, area and power utilization of the multiplier will be in direct share to the competence of compressor.

FPGA or Field Programmable Gate Arrays can be planned or configured by the user or expensive after manufacturing and during implementation. Hence they are or else known as On-Site programmable. Unlike a Programmable Array Logic (PAL) or other programmable device, their structure is similar to that of a gate-array or an ASIC. Thus, they are used to rapidly prototype ASICs, or as a substitute for places where an ASIC will finally be used. This is done when it is important to get the design to the market first. Later on, when the ASIC is produced in bulk to reduce the NRE cost, it can replace the FPGA. The programming of the FPGA is done using a logic circuit diagram or a source code using a Hardware Description Language (HDL) to specify how the chip should work. FPGAs have programmable logic components called 'logic blocks', and a hierarchy or reconfigurable interconnects which make possible the 'wiring' of the blocks together. The programmable logic blocks are called configurable logic blocks and reconfigurable interconnects are called switch boxes. Logic blocks (CLBs) can be programmed to perform complex combinational functions, or simple logic gates like AND and XOR. In most FPGAs the logic blocks also comprise memory elements, which can be as simple as a flip-flop or as composite as total blocks of memory. This proposed work having blocks and they worked as per their consideration. Main important factor is as follows

- 1) Booth multiplier- we are implement this structure using booth algorithm. Booth multiplication algorithm is that algorithm that multiplies two signed binary number in two's kind word notation. In this paper representation of the multiplicand and the product are not conduct. Specifically there are both also in two's praise representation. Like the multiplier, if the number that support addition and subtraction will work as condition. The order of the step is not be unwavering. Booth algorithm is the most important factor of this project. Next block is Wallace tree multiplier as we seen bellowed.
- 2) Wallace tree multiplier- Wallace tree multiplier is the another way to implementing high speed performance design which helps to neglecting the reply factor and circumstances which is occurring throughout processed. Wallace tree multiplier is an efficient hardware and software implementation of a digital circuit that multiplies two integers, devised by Australian computer scientist chris Wallace in 1964.

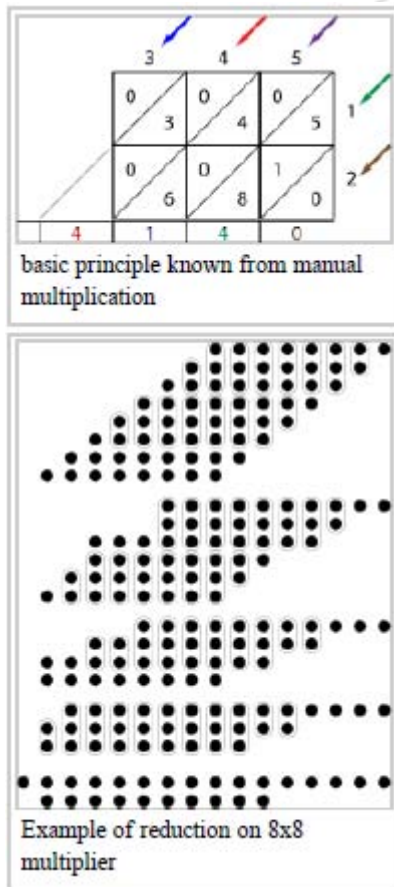


Figure 2: Wallace Tree Multiplier.

- This one is the simple example of wallace tree multiplier.the wallace tree multiplier has tree steps:
- a) First multiplies each bit of one of the arguments, by each bit of other one. During the result showing on position of the multiplied bits, the wires carry different weights.
 - b) Other step of the wallace tree multiplier is reducing the number of partial products to two by layers of full and half adders. This one is the second step of wallace tree multiplier.
 - c) Third one is group the wires in two numbers, and adds them with a conventional adder.

The second phase work as follows of Wallace tree, there are two three or more wires with the same weight add a following layers:

- Two three wires with the same weights and input them in to a full adder.
- During the process we find out the conclusion will be an output wires of the same weights and output higher weight for each three input wires.

In this phases two wires of same weight left, input them to the half adder condition, this one is the second phase of the multiplier solutions. Another step recommended to the next one wire left, which one is connected to the next layer.

- 1) Carry look ahead adder- there are so many adder but we used only carry look ahead adder. This carry look ahead adder solving the problem of calculating the carry signal. This condition is best on input signals. This processed will happened that carry signal will be generated in two case:
 - a) When the both a b are 1 as well.
 - b) This adder illustrate one of the two bits is 1 and carry-in

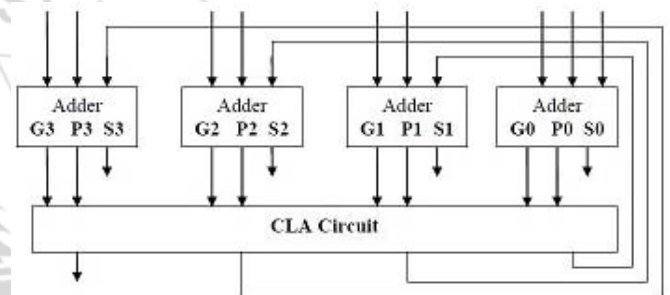


Figure 3: Carry Lookahead Adder

This adder generates and propagates the carry terms. This generates and propagates terms only depends on the inputs bits and thus should be valid after one two gate delay. That calculating the carry signals, one two not need to wait the carry to rippling all the steps and finally we getting the output or proper value.

4. Experimental Results

The test of proposed technique to implementation of different multiplier using booth algorithm. The image showed the delay timing and synthesis. The experimental result of different multiplier is as bellowed.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	31	4656	0%
Number of Slice Flip Flops	13	9312	0%
Number of 4 input LUTs	55	9312	0%
Number of bonded IOBs	16	232	6%
Number of GCLKs	1	24	4%

Figure 4: Synthesis of Proposed Multiplier.

Total REAL time to Xst completion: 0.00 secs
 Total CPU time to Xst completion: 0.15 secs

Figure 5: Timing Report of Proposed Multiplier.

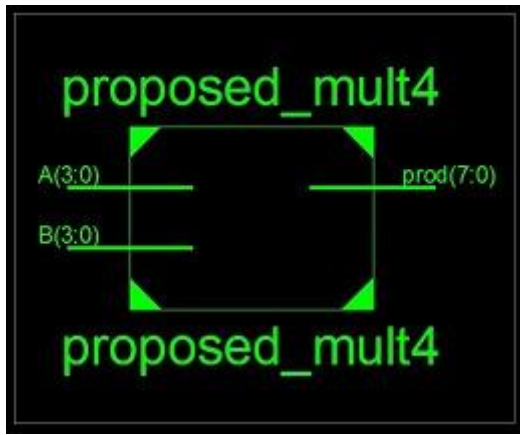


Figure 6: Synthesis of Proposed Multiplier.

Figure 7: Simulation of Proposed Multiplier

Above figures shown the experimental results of parallel multiplier using very high performance implementation. For this purposed we using the xilling and model sim software which are used for synthesis and simulation. Above fig showing the device utilization data which are access the detail reports data that message are filtering can also be performed. That synthesis of all the above multiplier is shows the RTL schematic diagram. Available process varies depending on these tools that are used. When select all the process in the flow, then it run automatically process to getting all the desired output. During the summery list including high level of information of our projects, utilization summery data gathered from the PAR report. All the environment variables and tools setting used during the implementation. Modelsim are used for simulation. We simulate the different multiplier, proposed multiplier, Wallace multiplier. We are implementing the different multiplier with using high performance. Timing report are shown the Wallace tree multiplier explaining the use of half and full adder for intermediate product terms obtained. This new device system design low power and delay and fast working.

5. Conclusion

The test of proposed technique to design and implement the high performance multiplier using booth algorithm.. By using the different software like modesim SE6.4 and xilinx we can be simulation and synthesis the parallel, wallax tree multiplier and proposed multiplier. Here we are concluding that using booth algorithm we designing the high and low power consumption multiplier.

Reference

- [1] Sumit.R. Vaidya and D.R.Dondekar's department of electronics engineering om clg maharashtra university India." performance comparison of multipliers for power speed trade off in vlsi design published in IEEE ISSN: 1790-5117 recent advance in networking vlsi and signal processing.
- [2] Shanthala. s and s.y. Kulkarni "vlsi design and implementing of low power MAC unit with block enabling technique." Published on European journal of scientific research ISSN 1450-216x vol. 30 no.4 (2009) p.p 620-630.
- [3] Manoranjan Pradhan, Rutu Parna and Shushanta Kumar Sahu "MAC implementation using vedic multiplication algorithm" published on international journal of computer application (0975-8887) vol 21-no.7,may 2011.
- [4] Padma Devi ,Ashima Girdher and Balwider Siingh , "improved carry select adder with reduce area and low power consumption" published on IEEE international journal of computer research volume 3-no.4 june 2010.
- [5] R. Caves ,M. Clerc, "The Swarm and the Queen: Towards a sssDeterministic and Adaptive Particle Swarm Optimization," In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 1951-1957, 1999.
- [6] H.H. Crokell, "Specialization and International Competitiveness," in Managing the Multinational Subsidiary, H. Etamad and L. S, Sulude (eds.), Croom-Helm, London, 1986.
- [7] K. Deb, S. Agrawal, A. Pratab, T. Meyarivan, "A Fast Elitist Non-dominated Sorting Genetic Algorithms for Multiobjective Optimization: NSGA II," KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.
- [8] J. Geralds, "Sega Ends Production of Dreamcast," vnunet.com, para. 2, Jan. 31, 2001.
- [9] Dong-Wook Kim, Young-Ho Seo, "A New VLSI Architecture of Parallel Multiplier-Accumulator based on Radix-2 Modified Booth Algorithm", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, vol.18, pp.: 201-208, 04 Feb. 2010.
- [10] Prasanna Raj P, Rao, Ravi, "VLSI Design and Analysis of Multipliers for Low Power", Intelligent Information Hiding and Multimedia Signal Processing, Fifth International Conference, pp.: 1354-1357, Sept. 2009.
- [11] Lakshmanan, Masuri Othman and Mohamad Alauddin Mohd.Ali, "High Performance Parallel Multiplier using Wallace-Booth Algorithm", Semiconductor Electronics, IEEE International Conference , pp.: 433- 436, Dec. 2002.
- [12] Jan M Rabaey, "Digital Integrated Circuits, A Design Perspective", Prentice Hall, Dec.1995.Louis P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication", Computers, IEEE Transactions,vol.24, pp.: 1014-1015, Oct. 1975.
- [13] Rajendra Katti, "A Modified Booth Algorithm for High Radix Fixedpoint Multiplication", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, vol. 2, pp.: 522-524, Dec. 1994.

- [14] C. S. Wallace, "A Suggestion for a Fast Multiplier", Electronic Computers, IEEE Transactions, vol.13, Page(s): 14-17, Feb. 1964..
- [15] Hussin R et al , "An Efficient Modified Booth Multiplier Architecture", IEEE International Conference, pp.:1-4, 2008.
- [16] John L Hennesy & David A. Patterson "Computer Architecture A Quantitative Approach" Second edition; A Harcourt Publishers International Company C. S.Wallace, "A suggestion for fast multipliers," IEEE Trans. Electron. Comput., no. EC-13, pp. 14– 17, Feb. 1964.
- [17] M. R. Santoro, G. Bewick, and M. A. Horowitz, "Rounding algorithms for IEEE multipliers," in Proc. 9th Symp. Computer Arithmetic, 1989, pp. 176–183.
- [18] D. Stevenson, "A proposed standard for binary floating point arithmetic," IEEE Trans. Comput., vol. C-14, no. 3, pp. 51-62, Mar.

Author Profile

Prajakta P. Chaure B.E in electronics and telecommunication engineering in Raison collage Amravati. Now pursuing M.E in Electronics and telecommunication in P.R.Pote Collage of Engineering and Management in Amravati.

