# ANALYSIS, VERIFICATION AND FPGA IMPLEMENTATION OF LOW POWER MULTIPLIER

**Garima Tiwari**

*M. Tech. Research Scholar, Mewar University, **Garimatiwari86@gmail.com***

## Abstract

*As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amounts of energy. While performance and area remain to be two major design goals, power consumption has become a critical concern in today's VLSI system design. Multiplication is a fundamental operation in most arithmetic computing systems. Multipliers have large area, long latency and consume considerable power. Previous work on low-power multipliers focuses on low-level optimizations and has not considered well the arithmetic computation features and application-specific data characteristics.*

*Binary multiplier is an integral part of the arithmetic logic unit (ALU) subsystem found in many processors. Booth's algorithm and others like Wallace-Tree suggest techniques for multiplying signed numbers that works equally well for both negative and positive multipliers. In this paper, we have used VHDL 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.*

***Keywords:*** *Multiplier, FPGA, ALU*

----------***----------

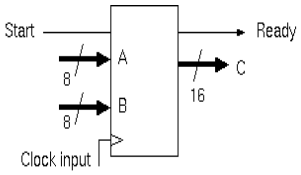
## 1. INTRODUCTION

Low-power design is a different goal from low-energy design although they are related. Power is a problem primarily when cooling is a concern. The maximum power at any time, peak power, is often used for power and ground wiring design, signal noise margin and reliability analysis. Energy per operation or task is a better metric of the energy efficiency of a system, especially in the domain of maximizing battery lifetime. In digital CMOS design, the well-known power-delay product is commonly used to assess the merits of designs.

Generally multiplication consists of three steps: generation of partial products or PPs (PPG), reduction of partial products (PPR), and final carry-propagate addition (CPA). Different multiplication algorithms vary in the approaches of PPG, PPR, and CPA. For PPG, radix-2 digit-vector multiplication is the simplest form because the digit-vector multiplication is produced by a set of AND gates. To reduce the number of PPs and consequently reduce the area/delay of PP reduction, one operand is usually recoded into high-radix digit sets. The most popular one is the radix-4 digit set {−2,−1, 0, 1, 2}. For PPR, two alternatives exist: reduction by rows, performed by an array of adders, and reduction by columns, performed by an array of counters. In reduction by rows, there are two extreme classes: linear array and tree array. Linear array has the delay of O(n) while both tree array and column reduction have the delay of O(log n), where n is the number of PPs. The final CPA requires a fast adder scheme because it is on the critical path. In some cases, final CPA is postponed if it is advantageous to keep redundant results from PPG for further arithmetic operations. Some low-level techniques that has been studied for multipliers include using voltage scaling, layout optimization, transistor reordering and sizing, using pass-transistor logic and swing limited logic, signal polarity optimization, delay balancing and input synchronization. However, these techniques have only achieved moderate improvement on power consumption in multipliers with much design effort or considerable area/delay overhead. The difficulty of low-power multiplier design lies in three aspects.

## 2. INTRODUCTION OF MULTIPIERS:

Multiplication is basically a shift add operation. There are, however, many variations on how to do it. Some are more suitable for FPGA use than others , some of them may be efficient for a system like CPU. This section explores various verities and attracting features of multiplication hardware.



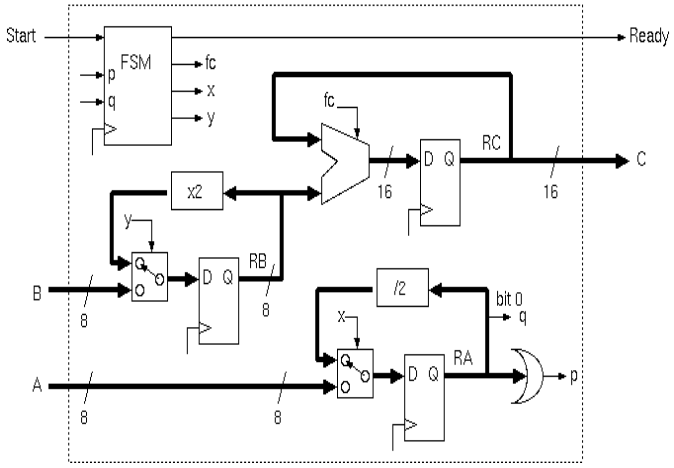




**Fig: 1** Multiplier Circuit

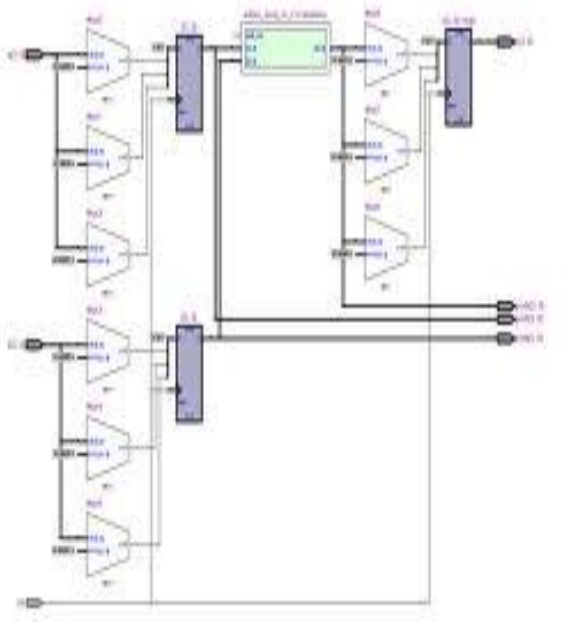


**Fig: 2** Modify Low power Multiplier

The multiplier area is quadratically related to the operand precision. Second, parallel multipliers have many logic levels that introduce spurious transitions or glitches. Third, the structure of parallel multipliers could be very complex in order to achieve high speed, which deteriorates the efficiency of layout and circuit level optimization. As a fundamental arithmetic operation, multiplication has many algorithm-level and bit-level computation features in which it differs from random logic. These features have not been considered well in low-level power optimization. It is also difficult to consider input data characteristics at low levels. Therefore, it is desirable to develop algorithm and architecture level power optimization techniques. We have design many multiplier and compare all of them that are given below

## 2.1 BOOTH MULTIPLIER:

Suppose we multiply a*b where is multiplicand and b is multiplier. The key to Booth's insight is to divide the groups bit of multiplier into 3 parts: the beginning, the middle, or the end of a run of 1s. More specific, the following table explains in detail:

| bi | bi-1 | Operation |
|---|---|---|
| 0 | 0 | Do nothing |
| 0 | 1 | Add a |
| 1 | 0 | Subtract a |
| 1 | 1 | Do nothing |

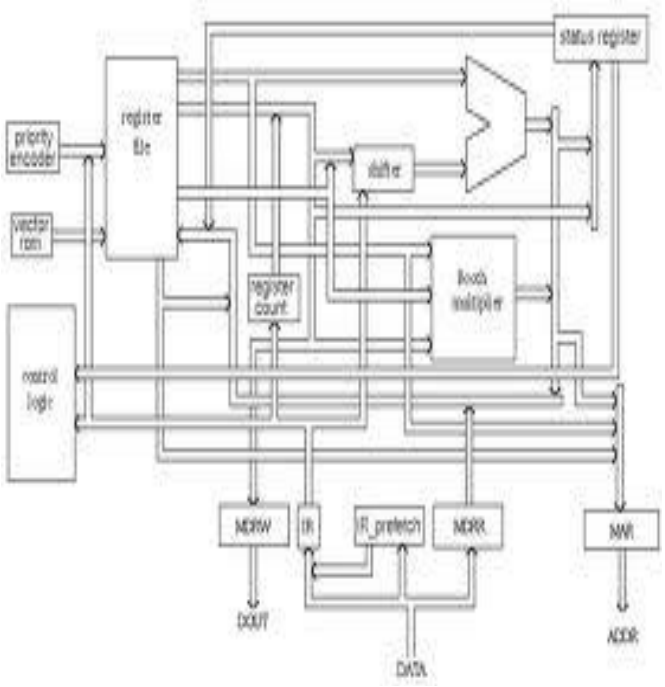



**Fig: 3**Modified Booth Multiplier

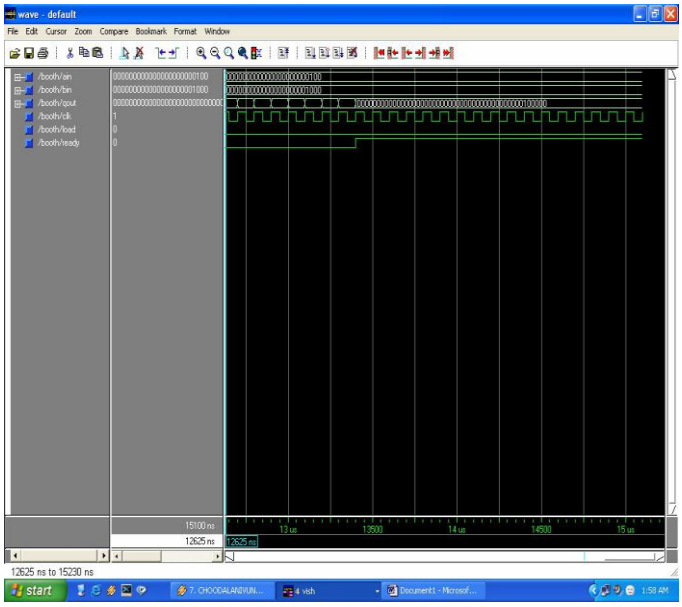


**Fig: 4** Simulation result for Modified Booth Multiplier

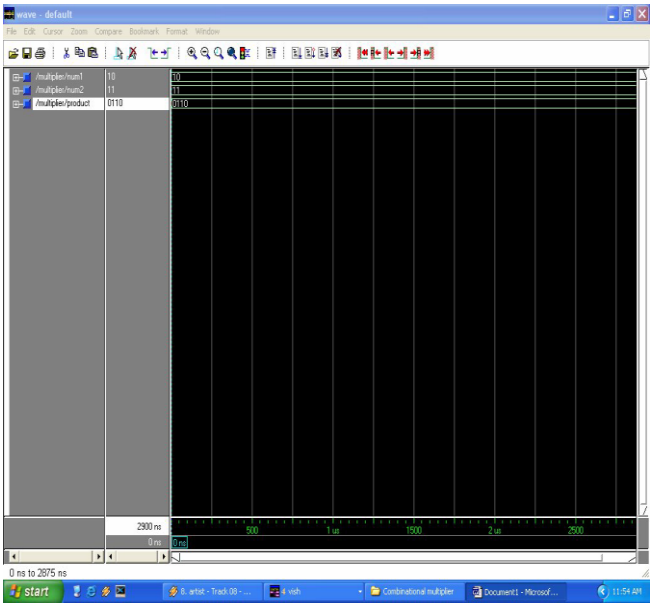


**Fig: 6** Simulation result modified Combinational multiplier

## 2.2 COMBINATIONAL MULTIPLIER

To understand the concepts in better way we have carried out the implementation of small size of combinational Multiplier. The size can be increased by just increasing the Array size of inputs and outputs.

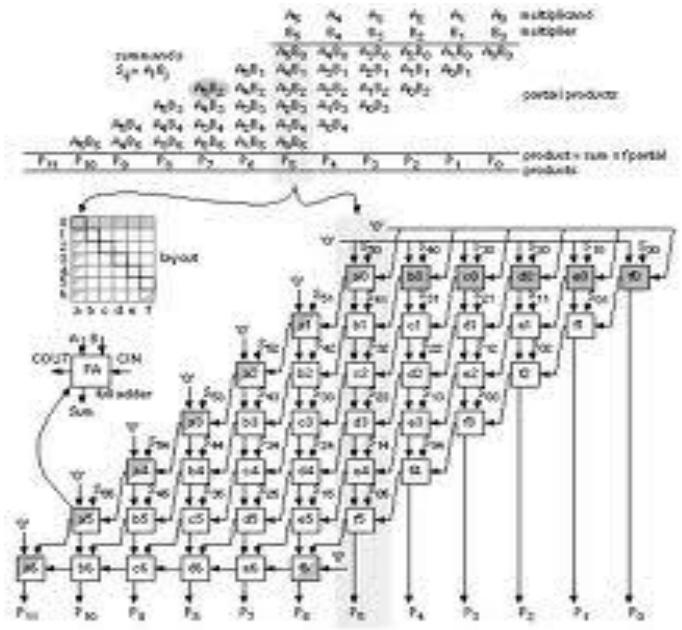


**Fig: 5** modified Combinational multiplier

## 2.3 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."

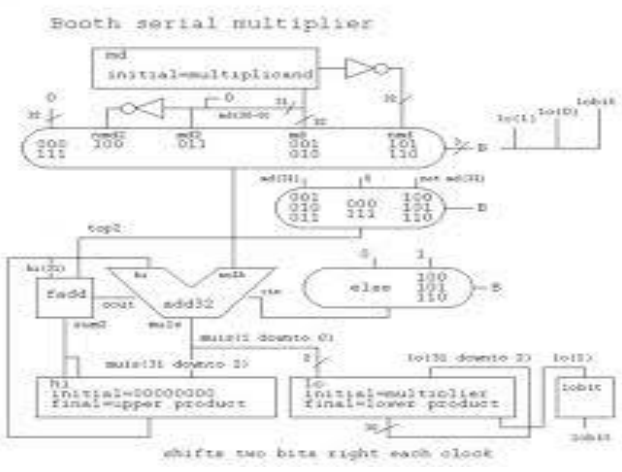



**Fig: 7** Modified Sequential Multiplier

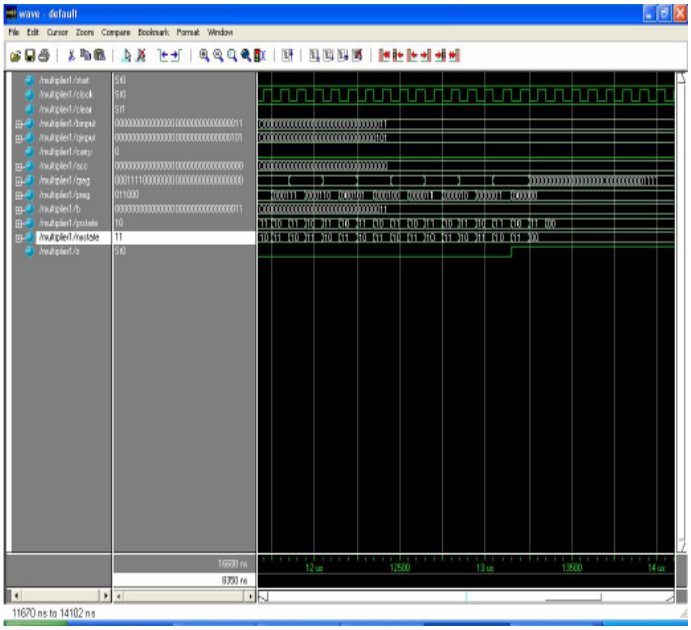


**Fig: 8** Simulation Result for Modified Sequential Multiplier

## 3. CSA WALLACE-TREE ARCHITECTURE:

An unsigned multiplier using a carry save adder structure is one of the efficient design in implementation 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.

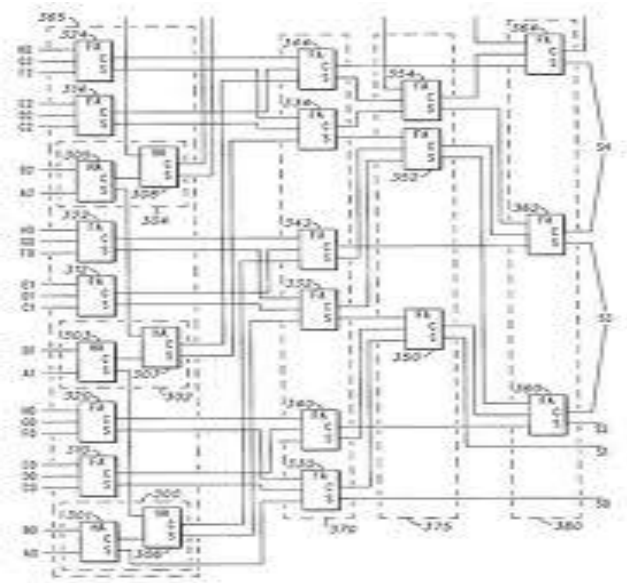


**Fig: 9** Modified and Optimize CSA Wallace Tree Multiplier Architecture

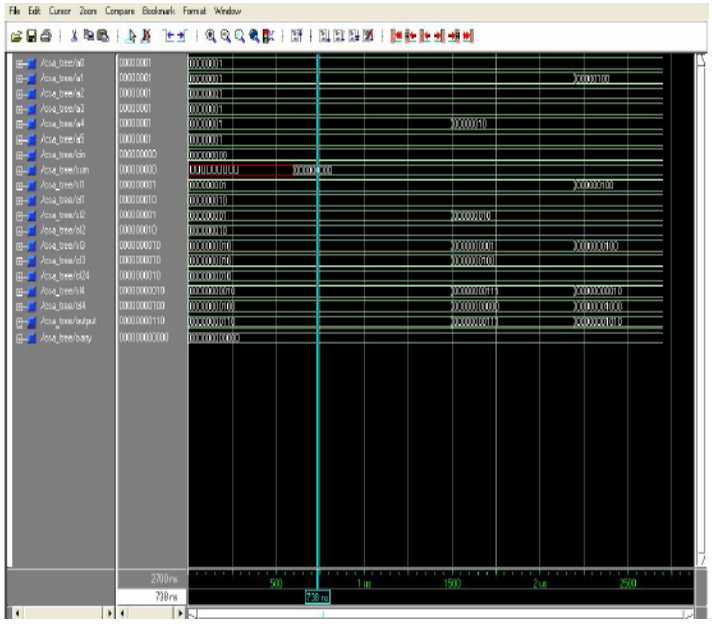


**Fig: 10** Simulation Result for Modified and Optimize CSA Wallace Tree Multiplier Architecture

## RESULT ANALYSIS

**Table 1:** Compression between multipliers

| S.No | Algorithms→ Performance/Parameters | Serial Multiplier (Sequential) | 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 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(36.98%) | 2(1.04%) | 8(4.17%) |
| 6. | Function Generators | 114(7.42%) | 141(36.72%) | 4(1.04%) | 16(4.17%) |
| 7. | Data Required Time/ Arrival Time | 9.54 ns 8.66 ns | 9.54 ns 9.36 ns | NA 8.61 | 10 ns 8.52 ns |
| 8. | Optimum Clock (MHz) | 100 | 101.9 | NA | 100 |
| 9. | Slack | 0.89 ns | 0.19 ns | Unconstrained path | 1.48 ns |

## CONCLUSIONS

Power dissipation in multiplier designs has been much-researched in recent years, due to the importance of the multiplier circuit in a wide variety of microelectronic systems. The focus of multiplier design has traditionally been delay optimization, although this design goal has recently been

supplemented by power consumption considerations. Our goal has been first to understand how power is dissipated in multipliers, and secondly to devise ways to reduce this power consumption.

In this paper, we described previous work which has been done in the area of multiplier delay and power optimization. We identified methods by which multiplier delay has been reduced, and we concentrated on understanding how these various speedup techniques impact the power dissipation of the multiplier as a whole. Power savings of up to 25% were achieved, along with reductions in die area and interconnect. We have presented an investigation of multiplier power dissipation, along with some techniques which allow reductions in power consumption for this circuit. Given the importance of multipliers, it is likely that further research efforts will be directed at optimizing this block for delay and power efficiency.

## REFERENCES

[1] A. Abnous and J. Rabaey, "Ultra-low-power domain-specific multimedia processors," in VLSI Signal Processing, IX, pp.461-470, Oct. 1996.

[2] I. Abu-Khater, A. Bellaouar, and M. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers", IEEE J. Solid-State Circuits, vol.31, no.10, pp.1535-1546, Oct. 1996.

[3] H. A. Al-Twaijry, Area and Performance Optimized CMOS Multipliers. Ph.D. dissertation, Stanford University, Aug. 1997.

[4] M. Alidina, et. al., "Precomputation-based sequential logic optimization for low power," IEEE Trans. VLSI Systems, vol.2, no.4, pp.426-436, Dec. 1994.

[5] F.S. Anderson, et. al., "The IBM system 360/91 floating point execution unit," IBM J. Res. Develop., vol.11, pp.34-53, Jan. 1967.

[6] E. de Angel and E.E. Swartzlander, Jr., "Survey of low power techniques for VLSI design," in Proc. 8th Annual IEEE Int. Conf. Innovative Systems in Silicon, pp.159-169, Oct. 1996.

[7] E. de Angel and E.E. Swartzlander, Jr., "Low power parallel multipliers,"in VLSI Signal Processing, IX, pp.199-208, Oct. 1996.

[8] E. de Angel, Low Power Digital Multiplication. Ph.D. dissertation, The University of Texas at Austin, 1996.

[9] E. de Angel and E.E. Swartzlander, Jr., "Switching activity in parallel multipliers," in Proc. 35th Asilomar Conf. Signals, Systems and Computers, pp.857-860, Nov. 2001.

[10] TSMC 0.18μm Process 1.8-Volt SAGE-X Standard Cell Library Databook. Artisan Components, Inc., Oct. 2001.

[11] A. Bellaouar and M. Elmasry, Low-power Digital VLSI Design: Circuits and Systems. Kluwer Academic Publishers, 1995.

[12] G.W. Bewick, Fast Multiplication: Algorithms and Implementation. Ph.D. dissertation, Stanford University, Feb. 1994.

[13] M. Bhardwaj, R. Min, and A. Chandrakasan, "Power-aware systems," in Proc. 34th Asiloma