

# Low Area Wallace Multiplier Using Energy Efficient CMOS Adder Circuit Analysis In Instrumentation

G. Sridhar\* and T. Reenaraj\*\*

**Abstract:** In most digital and high performance systems such as Microprocessor, FIR filter and Digital Signal Processor, Multiplier plays an important role. In this paper, the multiplier offers less area and power consumption is reduced. To reduce the hardware component, Energy efficient CMOS full adder plays an important role in Wallace tree multiplier. Modified Wallace multiplier have minimum adders than Standard Wallace Multiplier. In this paper, the energy efficient CMOS full adder is used in the modified Wallace multiplier at the place of full adder in the standard Wallace multiplier in order to reduce area and power consumption.

**Keywords:** Energy efficient full adder, CMOS full adder, Wallace multiplier.

## 1. INTRODUCTION

VLSI stands for “Very Large Scale Integration”. This is the field which involves packing more and more logic devices into small areas. Alongside, obeying Moore’s law, the capability of an IC has increased exponentially over the years, in terms of computation of power, utilization of available area, yield. The combined effect of these two advances is that people can now put diverse functionality into the IC’s, opening up new frontiers. Examples are embedded systems, where intelligent devices are put inside everyday objects, and ubiquitous computing where small computing devices proliferate to such an extent that even the shoes you wear may actually do something useful like monitoring your heartbeats.

The increasing prominence of portable systems and the need to limit power consumption (and hence, heat dissipation) in very-high density VLSI chips have led to rapid and innovative developments in low-power design during the recent years. The ever increasing demand for portable computing devices and wireless communication systems require low power VLSI circuits. Minimizing power dissipation during the VLSI design flow increases life time and reliability of the circuit. Numerous techniques for the design of low power VLSI circuits are reported where the dominant factor of power dissipation is caused by switching activity. While these techniques have reduced the circuit power dissipation during functional operation, testing of such low power VLSI circuits has recently become an area of concern. Therefore addressing the problems associated with testing low power VLSI circuits have become an important issue.

### Challenges

As microprocessors become more complex due to technology scaling, microprocessor designers have encountered several challenges which force them to think beyond the design plane, and look ahead to post-silicon:

- **Process variation** – As photolithography techniques tend closer to the fundamental laws of optics, achieving high accuracy in doping concentrations and etched wires is becoming more difficult and

\* Assistant Professor, PSN College of Engineering and Technology, India

\*\* Assistant Professor, PSN College of Engineering and Technology and Research Scholar, Sathyabama University, India

prone to errors due to variation. Designers now must simulate across multiple fabrication process corners before a chip is certified ready for production.

- **Stricter design rules** – Due to lithography and etch issues with scaling, design rules for layout have become increasingly stringent. Designers must keep ever more of these rules in mind while laying out custom circuits. The overhead for custom design is now reaching a tip point, with many design houses opting to switch to electronic design automation (EDA) tools to automate their design process.
- **Timing/design closure** – As clock frequencies tend to scale up, designers are finding it more difficult to distribute and maintain low clock skew between these high frequency clocks across the entire chip. This has led to a rising interest in multicore and multiprocessor architectures, since an overall speedup can be obtained by lowering the clock frequency and distributing processing.
- **First-pass success** – As die sizes shrink (due to scaling), and wafer sizes go up (to lower manufacturing costs), the number of dies per wafer increases, and the complexity of making suitable photomasks goes up rapidly. A mask set for a modern technology can cost several million dollars. This non-recurring expense deters the old iterative philosophy involving several “spin-cycles” to find errors in silicon, and encourages first-pass silicon success. Several design philosophies have been developed to aid this new design flow, including design for manufacturing (DFM), design for test (DFT), and Design for X.

## 2. OBJECTIVE OF THE WORK

A **Wallace tree** is an efficient hardware implementation of a digital circuit that multiplies two integers. The entire procedure is carried out into three steps:

- Partial product generation,
- Partial product grouping and reduction,
- Final adding.

The main objective of this project :

- **To reduce area** : The area can be reduced by minimizing gate counts in adder using common boolean logic.
- **To reduce power consumption**: The lower power consumption will be achieved by minimizing the area.

## PROJECT DESCRIPTION

### Wallace Tree Multiplier

A **Wallace tree** is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by an Australian Computer Scientist Chris Wallace in 1964.

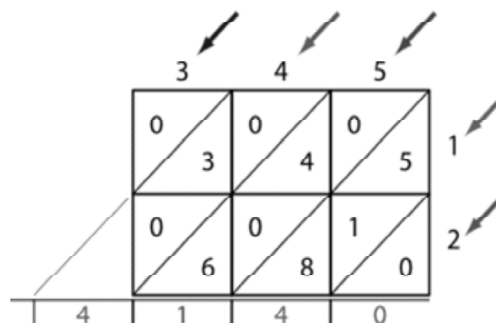


Figure 3.1: Basic Wallace Tree

The Wallace tree has three steps:

1. Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding  $n^2$  results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of  $a_2 b_3$  is 32 (see explanation of weights below).
2. Reduce the number of partial products to two by layers of full and half adders.
3. Group the wires in two numbers, and add them with a conventional adder.

The Wallace tree multiplier is considerably faster than a simple array multiplier because its height is logarithmic in word size, not linear. However, in addition to the large number of adders required, the Wallace tree's wiring is much less regular and more complicated. As a result, Wallace trees are often avoided by designers, while design complexity is a concern to them. Wallace tree styles use a log-depth tree network for reduction. Faster, but irregular, they trade ease of layout for speed. Wallace tree styles are generally avoided for low power applications, since excess of wiring is likely to consume extra power. While subsequently faster than Carry-save structure for large bit multipliers, the Wallace tree multiplier has the disadvantage of being very irregular, which complicates the task of coming with an efficient layout. The Wallace tree multiplier is a high speed multiplier. The summing of the partial product bits in parallel using a tree of carry-save adders became generally known as the "**Wallace Tree**". Three step processes are used to multiply two numbers.

- Formation of bit products.
- Reduction of the bit product matrix into a two row matrix by means of a carry save adder.
- Summation of remaining two rows using a faster Carry Look Ahead Adder (CLA).

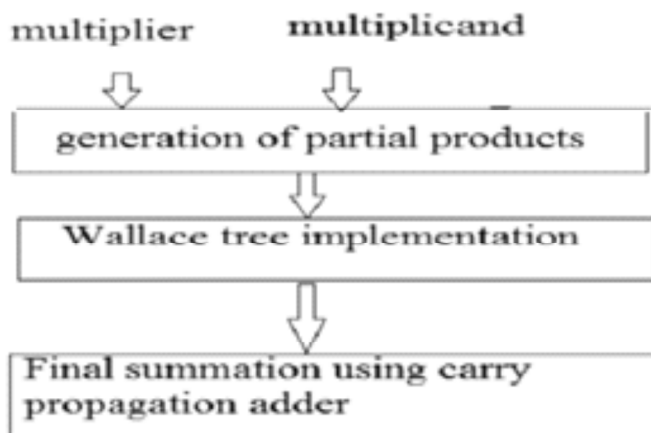


Figure 3.2: Flow diagram of Wallace tree

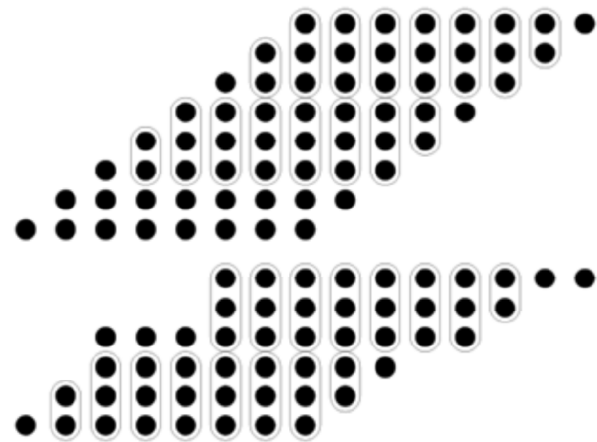


Figure 3.3: Wallace Multiplication

The second phase works as follows:

As long as there are three or more wires with the same weight add a following layer:

Take any three wires with the same weights and input them into full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.

- If there are two wires of the same weight left, input them into half adder.
- If there is just one wire left, connect it to the next layer.

The benefit of the Wallace tree is that there are only  $o(\log n)$  reduction layers, and each layer has  $o(1)$  propagation delay. As making the partial products is  $o(1)$  and the final addition is  $o(\log n)$ , the multiplication is only  $o(\log n)$ , not much slower than addition (however, much more expensive in the gate count). Adding

partial products with regular adders would require  $O(\log_2 n)$  time. From a complexity theoretic perspective, the Wallace tree algorithm puts multiplication in the class NC. These computations only consider gate delays and don't deal with wire delays, which can also be very substantial. The Wallace tree can be also represented by a tree of 3/2 or 4/2 adders.

### Half Adder

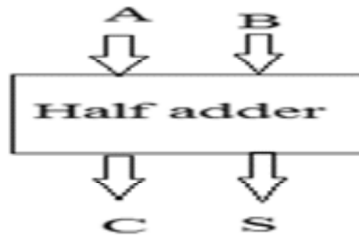


Figure 3.4: Block diagram of Half adder

The **half adder** adds two one-bit binary numbers  $A$  and  $B$ . It has two outputs,  $S$  and  $C$  (the value theoretically carried on to the next addition); the final sum is  $2C + S$ . The simplest half-adder design, pictured on the right, incorporates an XOR gate for  $S$  and an AND gate for  $C$ . With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder.



Figure 3.5: Logic diagram of Half adder

### 3.3. Full Adder

A **full adder** adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as  $A$ ,  $B$ , and  $C_{in}$ ;  $A$  and  $B$  are the operands, and  $C_{in}$  is a bit carried in from the next less significant stage. The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. binary numbers. The circuit produces a two-bit output sum typically represented by the signals  $C_{out}$  and  $S$ , where  $sum = 2 \times C_{out} + S$ . The one-bit full adder's truth table is:

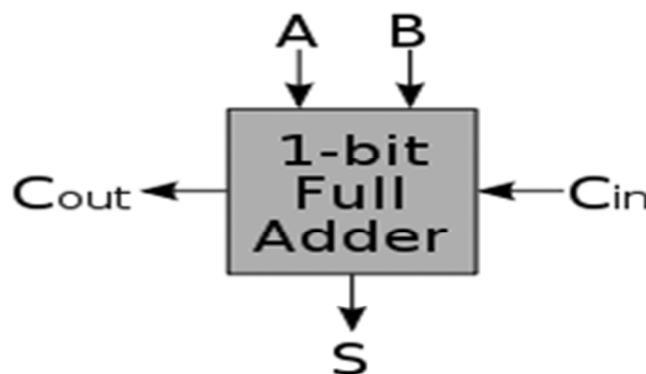


Figure 3.6: Block diagram of Full adder

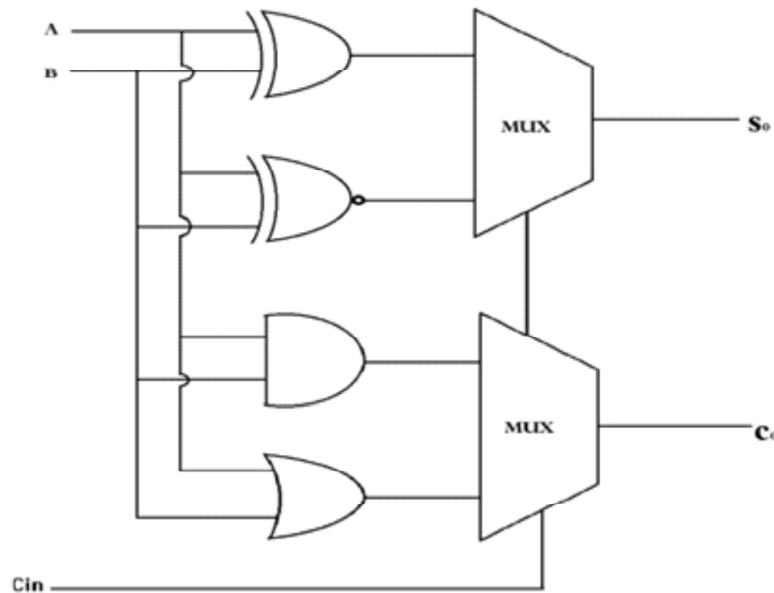


Figure 3.7: Logic diagram of Full adder

A full adder can be implemented in many different ways such as with a custom transistor-level circuit or composed of other gates. One example implementation is with

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B)).$$

In this implementation, the final OR gate before the carry-out output may be replaced by an XOR gate without altering the resulting logic. Using only two types of gates is convenient if the circuit is being implemented using simple IC chips which contain only one gate type per chip. In this light,  $C_{out}$  can be implemented as

$$C_{out} = (A \cdot B) \oplus (C_{in} \cdot (A \oplus B))$$

A full adder can be constructed from two half adders by connecting  $A$  and  $B$  to the input of one half adder, connecting the sum from that to an input to the second adder, connecting  $C_i$  to the other input and OR the two carry outputs. Equivalently,  $S$  could be made the three-bit XOR of  $A$ ,  $B$ , and  $C_i$ , and  $C_{out}$  could be made the three-bit majority function of  $A$ ,  $B$ , and  $C_i$ .

### 3. EXISTING WORK

An Existing Full adder design using alternative logic scheme gives low power delay product (PDP), in terms of speed, power consumption and reduced Area. Examining the truth table of EFA, it can be seen that the  $S_0$  output gives  $A \text{ XOR } B$  when  $C=0$ , and it also gives  $A \text{ XNOR } B$  when  $C=1$ . Thus, the Multiplexer can use to obtain the respective value taking  $C$  input as a selection signal. Following the same criteria taking  $C_0$  as a output equals to  $A \text{ AND } B$  when input  $C=0$ , and  $A \text{ OR } B$  when input  $C=1$ . Again  $C$  can be used as a respective value for the required result, using Multiplexer. An alternative logic scheme to design a full adder cell can be formed by a logic block to obtain the  $A \text{ XOR } B$  and  $A \text{ XNOR } B$  signals, another block can be obtain the  $A \text{ AND } B$  and  $A \text{ OR } B$  signals, and two Multiplexers driven by the  $C$  input to generate  $S_0$  and  $C_0$  outputs as shown in fig. Full adder design using swing restore complementary Pass-Transistor logic (SR-CPL) style [4] build the AND/OR gates And XOR/XNOR gates, and multiplexer based on pass transistor to obtain  $S_0$  and  $C_0$  output. The AND/OR gates build using a powerless and groundless pass-transistor logic.

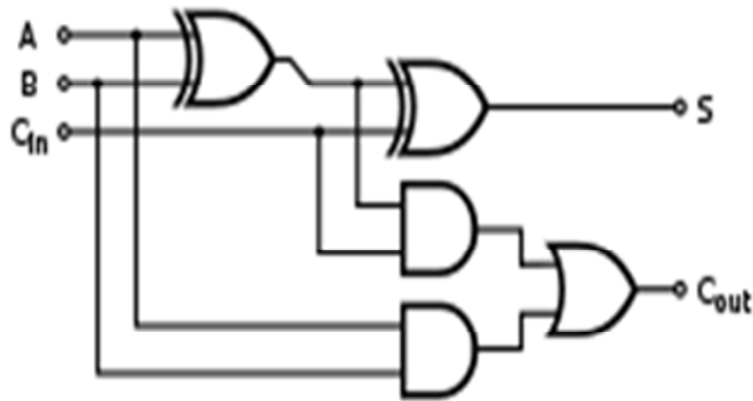


Figure 3.8: Logic diagram of Existing Full adder

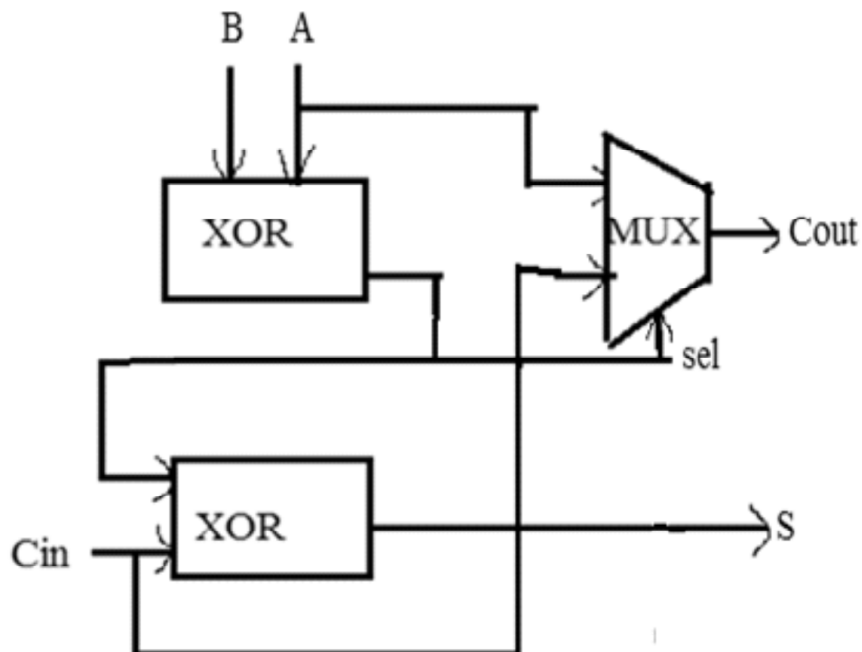
Double-pass transistor logic eliminates some of the inverter stages required for complementary pass transistor logic by using both N and P transistors, with dual logic paths for every function. While it has high speed due to low input capacitance, it has only limited capacity to drive a load

#### Drawbacks of Existing Work

- Higher number of great count in full adder circuit
- Higher power consumption
- Low performance

#### 4. PROPOSED WORK

The proposed full adder uses two XOR gates and one 2: 1 multiplexer unit. A and B are the inputs of first XOR gate and produce the output Y. Sum is generated by the XOR operation of  $C_{in}$  as one input and the output generated from the previous XOR gate Y is given as the another input . The carry is generated from the 2:1 multiplexer unit. The input of the multiplexer is A and  $C_{in}$ . Y acts as the control/selection signal of multiplexer. The carry  $C_{out}$  is generated depending upon the control signal.



## Advantages of Proposed Work

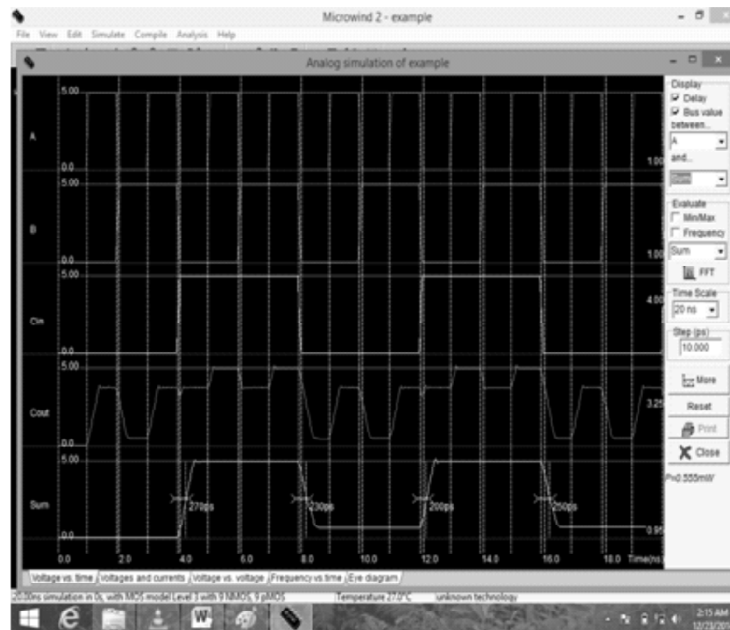
- Reduce power consumption
- Reduce area
- High performance

## 5. SIMULATION PROCEDURE

### MICROWIND AND DSCH – NOR EXAMPLE

- We will learn both the design flow and the CAD tools.
- The specifications we are going to see may be different for different foundry and technology.
- Design Example (2 Levels) : NOR Gate
  - Logic design
  - Circuit design

## 6. DESIGN AND SIMULATION RESULTS



## 7. CONCLUSION

In this project, the proposed full adder design which consumes less area and power than the existing full adder. The layout simulation have been done for the proposed circuit in order show the improvement in power consumption over supply voltages, operating frequency and temperature. The proposed adder has been designed using 45nm technology and proved it to be a better option for low power complex system design. The net effect is that proposed full adder shows a much better performance compared to existing full adder.

## 8. FUTURE ENHANCEMENT

In future work, the proposed full adder design will be implemented in Wallace multiplier to reduce the gate count and also to reduce power consumption.

### References

- [1] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II*, vol. 43, pp.
- [2] Vasudev G. and RajendraHegadi, Member, IACSIT" Design and Development of 8-Bits Fast Multiplier for Low Power Applications" *IACSIT International Journal of Engineering and Technology*, Vol. 4, No. 6, December 2012.
- [3] Gyanesh Savita<sup>1</sup>, Vijay Kumar Magraiya<sup>2</sup>, Gajendra Kulshrestha<sup>3</sup>, VivekGoyal" Designing of Low Power 16-Bit Carry Select Adder with Less Delay in 45 nm CMOS Process Technology" *International Journal of Emerging Technology and Advanced Engineering*.
- [4] Ila Gupta, Neha Arora, Dr. B.P.Singh" Design and Analysis of 2:1 Multiplexer for High Performance Digital Systems" *IJECT* Vol. 3, Issue 1, Jan. - MarCh 2012ISSN.
- [5] Ila Gupta, NehaArora, Prof. B. P. Singh / "New Design of High Performance 2:1 Multiplexer" *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, Mar-Apr 2012, pp.1492-1496.
- [6] Padma Devi, Ashima Girdher, Balwinder Singh Improved "Carry Select Adder with Reduced Area and Low Power Consumption" *International Journal of Computer Applications* (0975 – 8887) Volume 3 – No.4, June 2010.