

High Speed Area Efficient 32 Bit Wallace Tree Multiplier

Nyamatulla M. Patel^{1*}, Kavya Damodar Shetty²

¹Asst. Prof, Dept. of Electronics and Communication Engg. Hirasugar Institute of Technology, Nidasoshi Belgaum, Karnataka, India

²Student, Dept. of Electronics and Communication Engg. Hirasugar Institute of Technology, Nidasoshi Belgaum, Karnataka, India

Abstract – A 32 bit high speed Wallace tree multiplier is designed using Verilog HDL and 4 bit multiplication is implemented in FPGA. The circuit is designed using carry save adder architecture and finally with one look ahead carry adder. The design is an improved version of tree based Wallace tree multiplier architecture. The proposed method aims at high speed multiplication of 32 bit Wallace tree multiplier. The entire design of 4 bit multiplication is coded in Verilog HDL, simulated with Modelsim and synthesized using Xilinx FPGA device and the similar algorithm is used for computing 32 bit multiplication. The result shows that the proposed architecture takes very less time for computing the multiplication of two 32 bit numbers. The proposed multiplier is much efficient than the existing methods.

I. INTRODUCTION

Multiplication is the basic arithmetic operation and is widely used everywhere during computation. In digital signal processing, most of the arithmetic operations require the use of multiplications. The performance of three dimensional computer graphics mostly depends on the performance of multiplications. Therefore, there has been much work on advanced multiplication algorithms and design also. Critical factors in the design of multipliers are chip area and speed of multiplication. There is highly demand of high-speed multiplications and require less hardware. The performance of multiplier is affected by the multiplication strategy and type of the multiplier used.

A 32 bit high speed area efficient Wallace tree multiplier is designed using verilog HDL and implemented in FPGA. The circuit is designed using carry save adder architecture and finally with one look ahead carry adder. The design is an improved version of tree based Wallace tree multiplier architecture. This paper aims at high speed multiplication and an area efficient 32 bit Wallace tree multiplier. The entire design is coded in Verilog HDL, simulated with Modelsim and synthesized using Xilinx FPGA device. The result shows that the proposed architecture takes very less time for computing the multiplication of two 32 bit numbers. In terms of area also, the proposed multiplier is much efficient than the existing methods.

II. MULTIPLIERS

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation. design targets – high speed, low power consumption, regularity of

The common multiplication method is “add and shift” algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both Modified Booth algorithm and Wallace Tree technique we can see advantage of both algorithms in one multiplier. However with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing. On the other hand “serial-parallel” multipliers compromise speed to

achieve better performance for area and power consumption. The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

2.1 Multiplication Algorithm

The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below:

Y= Y_{n-1} Y_{n-2}Y₂ Y₁ Y₀ Multiplicand
 X= X_{n-1} X_{n-2} X₂ X₁ X₀ Multiplier

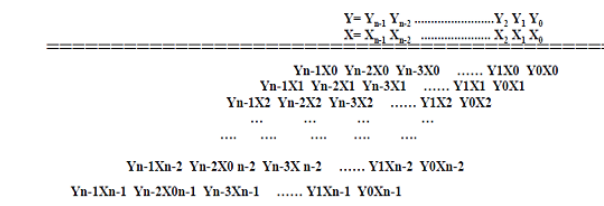


Figure-2.1 Multiplication Algorithm

AND gates are used to generate the Partial Products, PP, If the multiplicand is N-bits and the Multiplier is M-bits then there is N* M partial product. The way that the partial products are generated or summed up is the difference between the different architectures of various multipliers.

Multiplication of binary numbers can be decomposed into additions. Consider the multiplication of two 8-bit numbers A and B to generate the 16 bit product P.

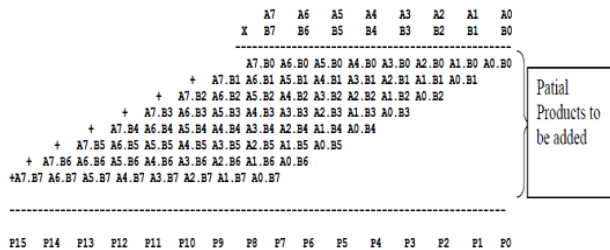


Figure-2.1.1 Generation of Partial Product

2.2 Array Multipliers

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added.

The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

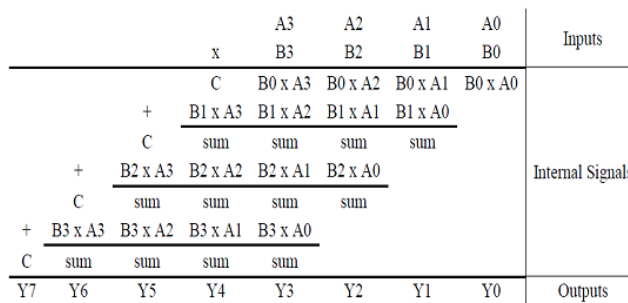


Figure-2.2 Array Multiplier

2.3 Wallace Tree Multiplier

Several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier, have been developed in past. Wallace introduced a very important iterative realization of parallel multiplier. This advantage becomes more pronounced for multipliers of bigger than 16 bits.

In Wallace tree architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. The most common counter used is the 3:2 counter which is a Full Adder.. The final results are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now O(logN). A block diagram of 4 bit Wallace Tree multiplier is shown in below. As seen from the block diagram partial products are added in Wallace tree block. The result of these additions is the final product bits and sum and carry bits which are added in the final fast adder (CRA).

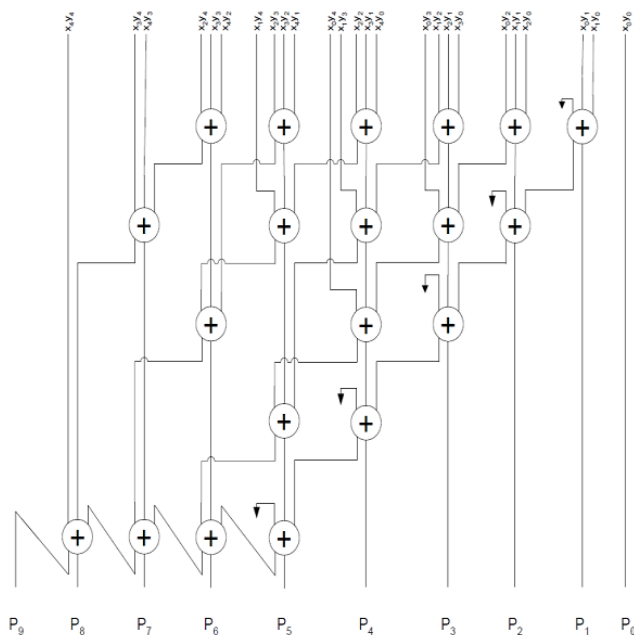


Figure-2.3 Generation of sum using Wallace tree for 4 bit input

Since Wallace Tree is a summation method, it can be used in conjunction with array multiplier of any kind including Booth array. The diagram below shows the implementation of 8 bit squarer using the Wallace tree for compressing the addition process.

III. METHODOLOGY

The use of carry save adder, to perform multiplication, first calculate the partial products of the multiplication, and then input them to the carry-save adder. For example, consider the multiplication of one argument ($a_3 a_2 a_1 a_0$) with another argument ($b_3 b_2 b_1 b_0$). The partial products are grouped into full adders and half adders. The grouping of partial products is done from downwards. The concept of carry save adder is that the sum is taken directly and the carry is propagated diagonally. Further the grouping is continued in the same fashion until we get the layer which consists of only half adders. At this stage we use carry look ahead adder instead of carry save adder as suggested by Wallace tree algorithm.[2,3]

A carry-save adder can add three values simultaneously, instead of just two. However, it does not output a single result. Instead, it outputs both a sum and a set of carry bits. The carry-save adder is essentially a group of full adders, each of which adds the bits in the same position of its three input sum operands. The full adder that adds bit i of its operands outputs bit S_i and carry bit C_{i+1} . Because the carry bits do not propagate through the adder, it is faster than parallel adders. Unlike the parallel adder, though, it does not produce a final sum of its inputs. A carry-save adder can add three values simultaneously, instead of just two. However, it does not output a single result. Instead, it outputs both a sum and a set of carry bits.

3.1 Wallace Tree Algorithm

Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, the idea of Wallace tree is given by Australian computer scientist Chris Wallace in 1964.

The Wallace tree has 3 steps.

Multiply the each bit of one of the arguments by each bit of the other, yielding n square results.

Reduce the number of partial products by using full adder and half adder.

Group the wires in two numbers and add them using CSA.

In last layer of algorithm we are using carry look ahead adder.

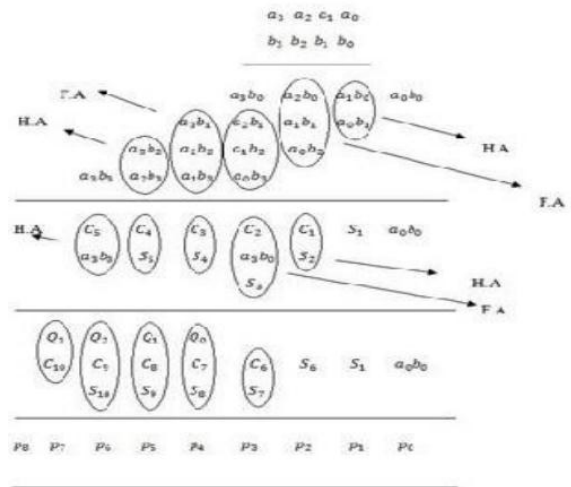


Figure-3.1.1 4*4 bit Wallace Multiplication

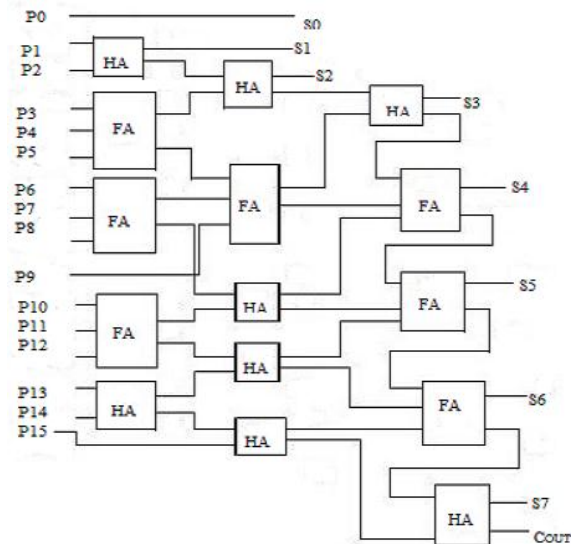


Figure 3.1.2-Block diagram of 4*4 Wallace tree multiplication

3.2 4 Bit Multiplication Algorithm

State 1: For 4*4 bit multiplication, initializing the Input x, y , and initializing the output sum.

State 2: Obtaining the partial product by multiplying each bit of x with each bit of y .

State 3: assigning the partial product from 0 to 15.

State 4: assign s_0 to p_0 .

State 5: level one of Wallace tree is calculated which consists of half adders and full adders.

State 6: assign s1.

State7: Level 2 is calculated by calling half adder and full adder.

State 8: assign s2.

State 9: use carry look ahead adder to calculate s3, s4, s5, s6, s7.

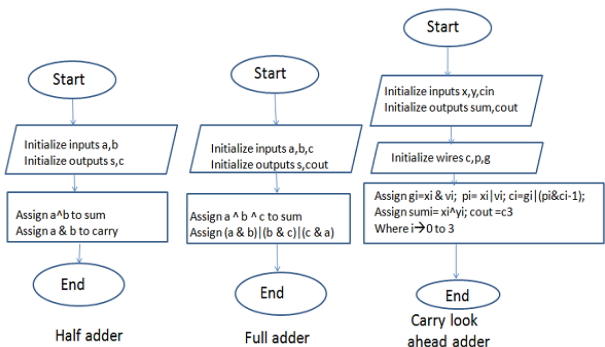


Figure-3.2.1 Flow chart for half adder, full adder and carry look ahead adder

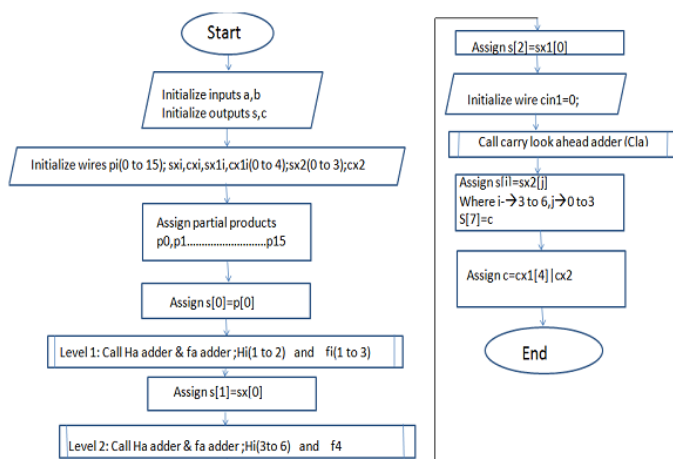
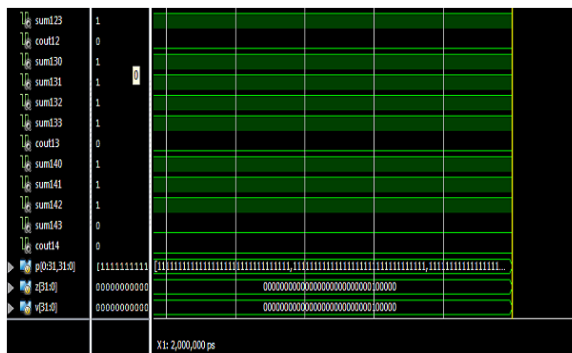
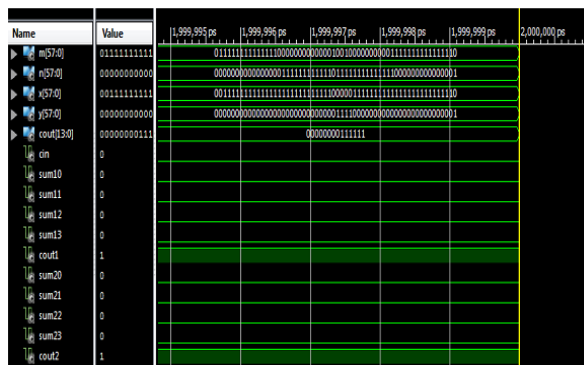
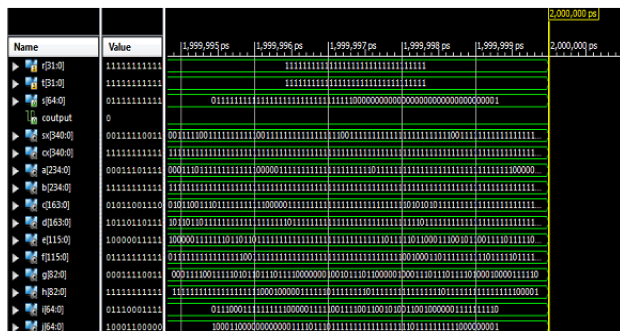


Figure-3.2.2 Flow chart for 4*4 bit Multiplication

IV. SIMULATION RESULT

4.1 32 Bit Multiplication



V. RTL VIEW

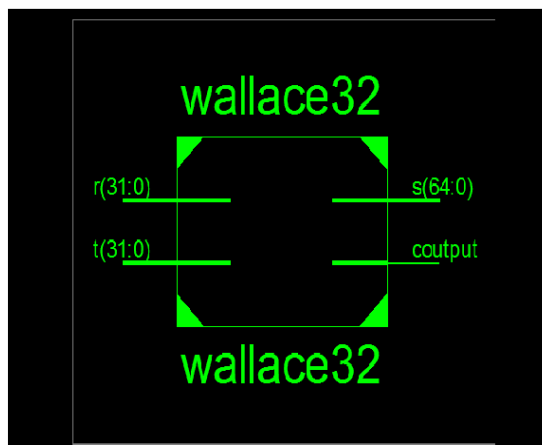


Figure-5.1 RTL View of Wallace tree multiplier.

VI. TECHNOLOGICAL VIEW

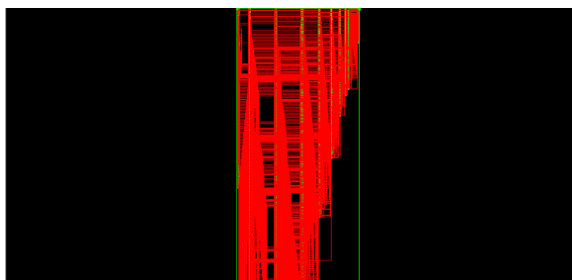


Figure-6.1 Technological View of Wallace tree multiplier.

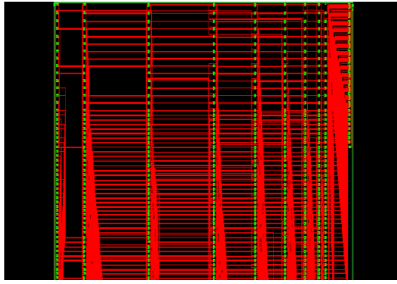


Figure-6.2 Technological View of Wallace tree multiplier.

VII. ADVANTAGES AND APPLICATION

7.1 Advantages

- The benefit of the Wallace tree is that there are few reduction layers, and each layer has small propagation delay.
- It has minimal latency and number of pipeline stages are achieved through a decay driven design scheme.
- The speed of operation is increased by using this multiplier.
- Required less memory space.
- More efficient.
- Less power required.

7.2 Applications

- Used in DSP applications.
- Useful for the study of integral operators
- Used in digital circuits.

CONCLUSION

The entire design of a 32 bit Wallace tree multiplier is coded in verilog and implemented in Xilinx FPGA. The RTL schematic view of the design is presented in figure 6.2. Simulation results and technological view are shown in figure 6.1 and 6.3 respectively. The delay is 16.56ns and speed has been increased and due to the reduced layer the area required is also less.

REFERENCES

Akther S, European conference on circuit theory and design, August 2007, VHDL implementation of fast NxN multiplier based on vedic mathematics.

C Vinoth, V S Kanchana Bhaskaran, IEEE 2011, A novel low power and high speed wallace tree multiplier for RISC Processor.

Damarla Paradhasaradhi, N Prashanti, N Vivek, IEEE 2013, Modified Wallace tree multiplier using efficient square root carry select adder.

Keshaveni N. August 2015, Volume 124-no.13, "Area efficient Wallace tree multiplier" International journal of computer application.

King Fai Pang, IEEE 1990, Architecture for pipelined Wallace tree multiplier- accumulators

Monika Vaishnav, October 2012, Volume 1, No.1, IJISCS, Design of multi-precision reconfigurable Wallace Tree Multiplier for high performance applications.

M Ravindra Kumar, August 2013, International journal of innovative research and studies, ISSN 2319-9725, Design and implementation of 32*32 bit high level Wallace tree multiplier.

Savita Niar, R.H Khade, Ajit Saraf, International journal of Advanced Research in Electrical and Electronics and Instrumentation engineering Vol-4, Issue 7, July 2015, Design and Analysis of Various 32 bit Multiplier in an approach towards a Fast Multiply.

Reminder Preet Pal Singh, Parveen Kumar, Balwinder Singh, International journal of Recent Trends in Engineering, Vol-2, no-6, November 2009, Performance Analysis of 32-Bit Array Multiplier with Carry Save Adder and with a Carry Look ahead Adder.

Corresponding Author

Nyamatulla M. Patel*

Asst. Prof, Dept. of Electronics and Communication Engg. Hirasugar Institute of Technology, Nidasoshi Belgaum, Karnataka, India

E-Mail – nyamatp@gmail.com