www.ioirp.com
International Journal of Innovative Research in Technology, Science & Engineering (IJIRTSE)
ISSN: 2395-5619, Volume – 4, Issue – 4. June 2018

High Speed Binary Counters Based on Wallace Tree Multiplier in VHDL

E.Sangeetha[1]ASP and D.Tharaliga[2]
Department of Electronics and Communication Engineering,
Tagore College of Engineering and Technology, Salem, India.

Abstract— In this paper, a new high speed binary counter design is proposed. Producing high speed 7:3 counter circuit with no xor gates by VHDL simulation . Due to the avoidance of xor gates, we get a faster designs with the efficient power. Reduction of Wallace tree multiplier is used to improve the speed of the multiplier. By using VHDL simulation the proposed counters are 33% faster than the existing system. This avoidance of xor gates results in faster designs with efficient power and area utilization. In VHDL simulations, consume less power than other higher order counters. Additionally, using the proposed counters in existing counter-based Wallace tree multiplier architectures reduces latency and power consumption. The synchronous counter circuit is also designed by using VHDL simulation.

***Index Terms*—High speed counter, tree reduction, synchronous counter, VHDL, Wallace tree multiplier.**

I. Introduction

High speed, efficient addition of multiple operands is an essential operation in any computational unit. The speed and power efficiency of multiplier circuits is of critical importance in the overall performance of microprocessors. Multiplier circuits are an essential part of an arithmetic logic unit, or a digital signal processor system for performing filtering and convolution. The binary multiplication of integers or fixed-point numbers results in partial products that must be added to produce the final product. The addition of these partial products dominates the latency and power consumption of the multiplier.

Multiplication of large operands is one of the most extensively used operations in the public key cryptosystems. Wallace tree multipliers offer high speed operation and therefore are used extensively in high performance applications. The operation of Wallace tree multiplier is divided into three steps as shown in Figure 1.

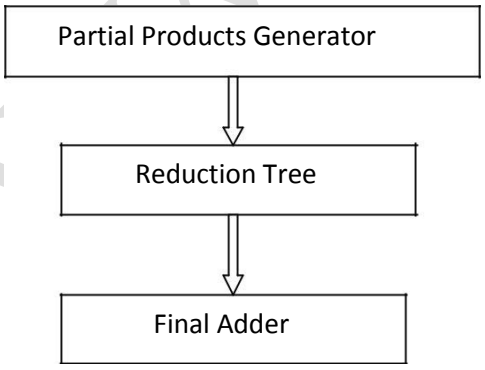

Figure 1 : Block Diagram of tree based multipliers

The partial product tree in the original Wallace tree multiplier is divided into groups where each group consists of three rows [1]. Then the addition is performed in every column using Full Adders (FAs) and Half Adder (HAs). This process is repeated until the tree is reduced to two rows. A large number of papers have been published in the literature to improve the performance of the Wallace multiplier. A Booth-encoded based Wallace multiplier is proposed in [2] which uses Booth-encoding to generate the partial products. In [3], a layout strategy is proposed to reduce the wiring delay of the tree reduction. This approach results in a slightly faster multiplier as compared to the Wallace tree multiplier. A modification is proposed in [4] to reduce the complexity of the traditional Wallace tree by reducing the number of half adders in the reduction process.

This strategy allows the modified architecture to have less area as compared to the Wallace multiplier while the speed of the both multipliers is same due to the same stages in the reduction process. A number of architecture use high speed counters in Wallace tree reduction to reduce the delay of the partial product tree reduction. The architecture in [5] uses a technique similar to Wallace reduction [1] to compute the sum of N inputs where all the inputs have the same weight. The architecture computes the 1s in the columns by using only the full adders. A modified form of this architecture is presented in [6] which uses Ripple Carry Adders (RCAs) and FAs to perform the counting.

The paper does not address the construction of large multipliers. A number of circuits are described in [8] for the construction of 6:3 and 7:3 counters using full custom design. Similarly, [9] presented a detailed investigation of the existing 6:3 and 7:3 counters as well as proposal of new circuits using CMOS transistors. However there is a lack of generic algorithm which can be used to construct large Counter Based Wallace (CBW) multipliers. This paper proposes a structural approach which can be used to implement the counter based Wallace multiplier of any size. The proposed algorithm can be easily employed for the implementation of CBW multiplier of any size on FPGA and ASIC platforms.

II.HIGH SPEED BINARY COUNTERS

The aim of the proposed counter-based Wallace multiplier is to use only the 7:3 counters along with the full adder and half adder to construct the multiplier. However this would result in more reduction stages as well as the restriction of a generic formula to compute the number of reduction stages for a *N*-bit multiplier.

This issue is solved by incorporating 6:3 and 7:3 counter along with the preferred 7:3 counter. A digital circuit which has a clock input and have a number of count outputs which give the number of clock cycles. A circuit that will divide a binary input signal by two, producing one output. We want to get that output fast and in high speed. So, we use high speed binary counter. Then the application of binary counters is digital clocks and analog to digital converters. We used the 7:3 counter proposed in [10] due to its simple and fast circuit.

A. 6:3.Counter

A counter that follows the binary number sequence is called a binary counter. The circuit of the 6:3 counters is based on the concept of carry look ahead after which uses propagate and generate signals to speed up the addition. A 6:3 counter compress six partial products into three outputs. The circuit diagram of the 6:3 counters is given in Figure 2.

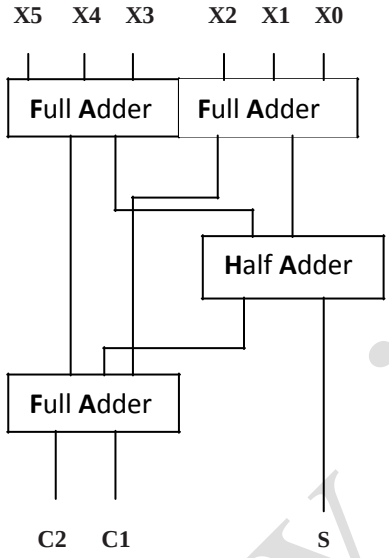

Figure 2: A 6:3 counter circuit built from full and half adders

The structure of this counter is composed of three full adders and one half adder. This kind of counter could be used in a high speed multiplier to reduce the number of partial products. The propagate and generate functions for the 6:3 counter are given as follows:

$P_0 = A\ B \quad P_1 = C\ D \quad P_2 = E\ \ F$
$G_0 = A.B \quad G_1 = C.D \quad G_2 = E.F$

The proposed 6:3 counter based on Wallace tree multiplier has no xor gates on its critical path, it operates nearly 33% faster than all other counter designs by using VHDL simulation. This novel method of counting allows construction of a counter for a substantial performance increase without increasing power consumption. A 6:3 counter built using this method uses no xor gates or multiplexers on its critical path. VHDL simulation results give our 6:3 counter is faster than existing counter designs.

A 6:3 counter has 6 inputs X1, X2, X3, X4, X5, X6 and three outputs SUM, COUT1 and COUT2. As a result, the circuit for COUT2 remains unchanged, with respect to the circuit for the COUT1 and SUM needs slight modification. The threshold voltages of all the transistors are changed because M = Vdd/6.

B. 7:3.Counter

The design of 7:3 counters is extensively studied in the literature and a number of architectures are proposed. The 7:3 counter proposed in [10] is selected for the proposed CBW multiplier due to its high speed operation. A 7:3 counter includes seven inputs and three outputs are called SUM, COUT1 and COUT2. The architecture of a conventional 7:3 counter is composed of four full adders. In this design the outputs are based on the number of bits with value '1' in truth table.

The actual performance of these counter cells heavily depends on the underlined technology. The different counter logics are designed upon the concept of the counter of a full adder. It could be defined as a single bit adder circuit which has four or five or six or seven inputs and three output. The design of counter circuits including full adder trees has a relatively high delay and consumes more power. These counter structures also grow quickly with the input vector size in terms of the needed number of full adder cells.

A 7:3 counter has a 7 inputs X0, X1, X2, X3, X4, X5, X6 and three outputs, SUM, COUT1, and COUT2. The simplified truth table of this circuit is similar to the simplified truth table of the 6:3 counter circuit. The circuit diagram of the 7:3 counter is given in Figure 3.

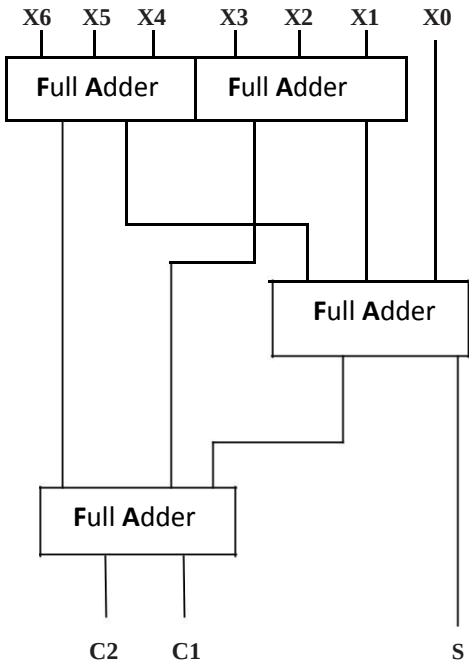

Figure 3. A 7:3 counter circuit built from full adders Hence, the circuit for COUT1 and COUT2 remains unchanged with respect to the 6:3 counter and only the circuit for the SUM needs a slight change. The threshold voltage of all the transistors are changed because M = Vdd/7.

III.WALLACE TREE MULTIPLIER

It is an efficient hardware implementation of a digital circuit that multiplier two integers. It reduce the number of partial products. Wallace multiplier is an efficient parallel multiplier it has three steps:

(i) Multiply each bit of one of the arguments, by each bit of the other yielding n2 results. Depending on position of the multiplied bits, the wires carry weights.
(ii) Reduce the number of partial products to two by layers of full and half adders.
(iii) Group the wires in two numbers, and add them with a conventional adder.

In this first stage, the multiplicand and the multiplier are multiplied bit by bit to generate the partial product terms. The second stage is the most important, as it is the most complicated and determines the overall speed of the multiplier. This stage includes addition of these partial product terms to generate the product 'p'. This paper will be more focused on the counter designs with VHDL simulation which consists of the addition of all the partial products [5].

In high speed design, the Wallace tree construction method is usually used to add the partial products. Although fast, since its critical path delay is proportional to the logarithm of the number of bits in the multiplier. In the last stage, the two row outputs of the tree are added using any high speed adder such as carry save adder to generate the output result.

Figure 4. Wallace Tree Multiplier

A.COUNTER BASED WALLACE MULTIPLIER

This section discusses the design of counter based Wallace (CBW) multiplier. The proposed algorithm uses a readjusted form of partial product tree which is rearranged as reverse pyramid [4]. Then the reduction is performed using the 6:3 and 7:3 counters along with the FAs and HAs. The CBW can perform the tree reduction in fewer stages as compared to the traditional Wallace multipliers [1]. Now, we will develop the equations to compute the maximum rows in each stage of CBW multiplier and total stages required for reduction process for *NXN* multiplier.

The first stage of an *NXN* multiplier has *N* rows. We need to find the maximum rows in subsequent stages until we are left with only two rows. Let us assume that the maximum rows in $stage_{i-1}$ are 16 and the number of rows in each column are same. In order to perform the reduction at column c, two 7:3 counters are used which will operate on 14 rows. The remaining two rows are reduced by using a 7:3 counters.

Figure 5 . Dot diagram of CBW Multiplier

Similarly, the columns c-1 and c-2 are reduced by using two 6:3 and one 7:3 counters. The two counters used at column c-1 will produce three Cout1 bits which are added to the column c of the stage $_i$. This will increase the rows in column c of stage $_i$ from 3 to 6. The two 7:3 counters at column c-2 will produce two C_{out2} bits which are also added to the column c of stage $_i$. Hence, the rows in column c of stage $_i$ will increase from 6 to 8.

Based on the observations of above example, we can obtain the rows in stage $_i$ by adding,
1) The number of traditional and proposed counters at column c and c-1 of $stage_{i-1}$.
2) The number of proposed counters (6:3 and 7:3) at column c-2 of $stage_{i-1}$.
3) The unprocessed row at column c of $stage_{i-1}$.

The number of stages for CBW multiplier are less as compared to the common Wallace tree multiplier. The purpose of this is to analyze the effects of different design strategy on the area utilization, power consumption, and

delay of the multiplier. The architectures differ in terms of the type of counters used at various places for reduction.

IV . SYNCHRONOUS COUNTER

The synchronous counters are the clock inputs of all the flip flops are connected together and are triggered by the input pulses. Thus all the flip flops are change state simultaneously in parallel. The most important advantage of synchronous counters is that there is no cumulative time delay because all flip flops are triggered in parallel.

Thus the maximum operating frequency for this counter will be significantly higher than for the corresponding ripple counter. Synchronous counters are so called because the clock input of all the individual flip flops within the counter are all clocked together the same time by the same clock signal. The synchronous counter the external clock signal is connected to the clock input every individual flip flop within the counter so that all of the flip flops are clocked together simultaneously.

The J and K inputs of FF0 are connected to high. FF1 has its J and K inputs connected to the output of FF0, and the J and K inputs of FF2 are connected to the output of an AND gate that is fed by the outputs of FF0 and FF1. Both outputs of FF0 and FF1 are high. The positive edge of the 4th clock pulse will cause FF2 to change its state due to the AND gate. The binary 4 bit synchronous up counter is shown in figure 6.

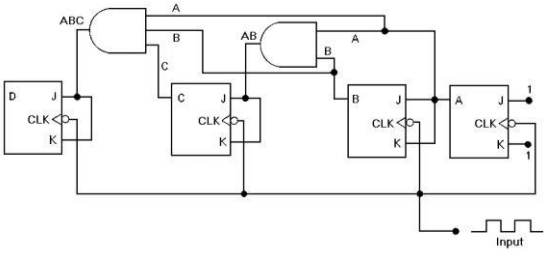

Figure 6. 4 bit synchronous up counter

JK flip flop to toggle based on whether or not all preceding flip flop outputs (Q) are HIGH we can obtain the same counting sequence as with the asynchronous circuit. Then as there is no inherent propagation delay in synchronous counters because all the counter stages are triggered in parallel at the same time.

As synchronous down counters are formed by connecting flip flops together and any number of flip flops can be connected or cascaded together to form a divide by n binary counter the modulo number till applies as it does for asynchronous counters. So a decade counter counts from 2n-1 can built along with sequences.

A synchronous down counter circuit also be built using synchronous binary counters to produce a count sequence from 0 to 9. This 4 bit synchronous counter counts sequentially on every clock pulse the resulting outputs count upwards from 0 to 15. Therefore this type of counter is also known as a 4 bit synchronous up counter.

The result of this synchronization is that all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect and therefore no propagation delay.

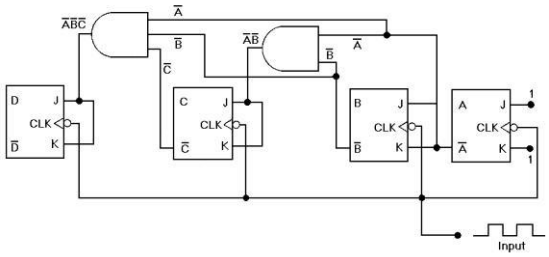

Figure 7. 4 bit synchronous down counter

V.VHDL SIMULATION

Very high speed integrated circuit **H**ardware **D**escription **L**anguage (VHDL) is used in electronic design automation to describe the digital and mixed signal systems as field programmable gate array and integrated circuits. VHDL can also be used as a general purpose parallel programming language. The idea of being able to simulate the ASICs from the information in this documentation was so obviously attractive that logic simulators were developed that could read the VHDL files.

The next step was the development of logic synthesis tools that read the VHDL, and output a definition of the physical implementation of the circuit. VHDL is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design. This collection of simulation models is commonly called a test bench.

The key advantage of VHDL when used for systems design, is that it allows the behavior of the required system to be described and verified before synthesis tools translate the design into real hardware. Another benefit is that VHDL allows the description of a concurrent system. VHDL is a dataflow language unlike procedural computing languages such as basic C and assembly code.

A.VHDL code of Full Adder

```
Entity full_add is
Port (a : in STD_LOGIC);
      b : in STD_LOGIC;
     Cin : in STD_LOGIC;
```

```
        Sum : out STD_LOGIC;
        Cout : out STD_LOGIC;
End full_add;
```

B. OUTPUT WAVEFORM OF SYNCHRONOUS COUNTER

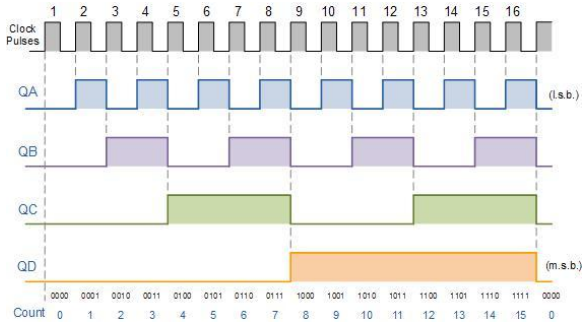

Figure 8. Output of synchronous counter

VI. CONCLUSION AND FUTURE WORK

In this paper, the circuit design of 6:3, 7:3 and synchronous counter circuits in the partial tree reduction process which enables it to perform the operation in less stages. The VHDL simulation shows that the results of counter circuits with a best performance. The proposed approach allows the fast and easy implementation of large circuit designs. By the avoidance of xor gates and the reduction of Wallace tree multipliers are used to improve the speed of the counter. It consume less power and reduce the power consumption than the existing system. By the implementation of VHDL simulation the proposed approach is 33% faster than the existing system.

REFERENCES

[1] T.Fam, Adly and Christopher Fritz, "Fast binary counters based on symmetric stacking", IEEE Trans.Comput., 2017.

[2] S. Asif and Y. Kong, "Design of an algorithmic wallace multiplier using high speed counters," in *Proc. IEEE Comput. Eng. Syst. (ICCES)*, Dec. 2015, pp. 133–138.

[3] S. Asif and Y. Kong, "Analysis of different architectures of counter based Wallace multipliers," in *Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2015, pp. 139–144.

[4] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comput.*, vol. 45, no. 3, pp. 294–306, Mar. 1996.

[5] A. Dandapat, S. Ghosal, P. Sarkar, and D. Mukhopadhyay, "A 1.2-ns 16 × 16-bit binary multiplier using high speed compressors," *Int. J. Elect. Electron. Eng.*, vol. 4, no. 3, pp. 234–239, 2010.

[6] D. Radhakrishnan, "Low-voltage low-power CMOS full adder,"*IEEE Proc.-Circuits, Devices Syst.*, vol. 148, no. 1, pp. 19–24,Feb. 2001.

[7] S.-F. Hsiao, M.-R. Jiang, and J.-S. Yeh, "Design of high-speed lowpower 3-2 counter and 4-2 compressor for fast multipliers," *Electron. Lett.*, vol. 34, no. 4, pp. 341–343, Feb. 1998.

[8] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[9] L. Dadda, "Some schemes for parallel multipliers," *Alta Freq.*, vol. 34, pp. 349–356, May 1965.

[10] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," *IEEE Trans. Comput.*, vol. 44, no. 8, pp. 962–970, Aug. 1995.

[11] M. Mehta, V. Parmar, and E. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits," in *Proc. 10th IEEE Symp. Comput. Arithmetic*, Jun. 1991, pp. 43–50.

[12] S. Veeramachaneni, L. Avinash, M. Krishna, and M. B. Srinivas, "Novel architectures for efficient (m, n) parallel counters," in *Proc. 17th ACM Great Lakes Symp. VLSI*, 2007, pp. 188–191.

[13] S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M. B. Srinivas, "Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors," in *Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2007, pp. 324–329.

[14] J. Gu and C.-H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2. Apr. 2003, pp. 661–664.

[15] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in *Proc. Conf. Rec. 35th Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Nov. 2001, pp. 129–133.