International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 02 | Feb 2019 www.irjet.net p-ISSN: 2395-0072

Wallace Tree Multiplier Using MFA Counters

Sheba K Sam[1], Mercy Mathew[2]

[1]M-TECH Student, Department of ECE, & Belivers Church Caarmel Engineering College, Kerala—689711, India.
[2]Asst.Professor, Department of ECE, & Belivers Church Caarmel Engineering College, Kerala—689711, India.

Abstract – *Multiplier circuits are the important part of a digital signal processing system, arithmetic and logic units. Binary multiplication results in partial products that are effectively added to get the final product. Addition of partial products occurs delay and power consumption of the multiplier. For generating partial products efficiently, Column compression is commonly used. Counters are widely used in a variety of applications. The proposed Multiplexer (MUX) based full adder counters are designed that is found to be faster than existing counters and has less power. Moreover, applying the proposed counters in wallace multiplier architectures reduces its delay and power consumption effectively.*

Key Words*:* Stacking, Multiplexer based full adder, Multiplier, Counter.

1. INTRODUCTION

Important feature of an electronic device such as mobile phones, laptops is its processor and its speed of operation. Everyone wants these devices to work with in a split of second. This happens only when the device has a fast and good processor. Multiplication represents one of important bottlenecks in many digital processing. Based on the size, several products are added to get the product. Generally multiplication is a slow process as a large number of partial products are added to evaluate the final product. For a 32 by 32-bit multiplication, 32 partial products are needed to evaluate the final product.

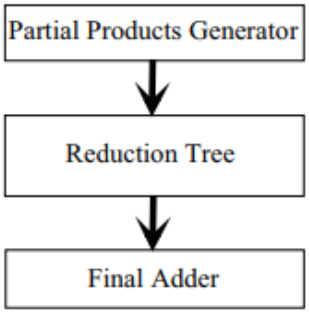

Fig -1: Block diagram of tree based multipliers.

In order to combine the partial products efficiently, column compression technique is commonly used. Many methods have been presented to increase the performance of the partial product summation, such as the well-known row compression techniques. These methods involve using full adders functioning as counters to reduce groups of 3 bits of the same weight to 2 bits of different weight in parallel using a carry-save adder tree. By several layers of reduction, the number of summands gets reduced to two stages, which are then added using a conventional adder circuit.

For higher efficiency, larger numbers of bits having equal weight are needed to be considered. The basic method when dealing with larger numbers of bits is the same: bits in one column are counted, producing fewer bits of different weights. For example, a 6:3 counter circuit accepts 6 bits of equal weight and counts the number of "1" bits. This count is then output using 3 bits of increasing weight.

A modified Booth encoding algorithm [4] will reduce the number of partial products. For a 16 by 16-bit multiplication, 16 partial products are added. This will reduce the number of partial products from 16 to eight that is still a large number and have high delay in comparison with other functional units in the system such as adders. In digital signal processing, this delay is unacceptable.

Dadda [5] gives the idea of using full adders to reduce the partial product matrix by the concept of (n,m) parallel counters. An (n,m) parallel counter is a counter having n inputs and m outputs. Thus, we can say that a full adder (3, 2) is a parallel counter. Main aim of Dadda multiplier is to reduce the height of the partial product matrix by suitable parallel counters. At every level, he used just enough parallel counters to reduce the height of the partial product matrix to the next lower number in the sequence. Dadda's scheme has the same delay as the Wallace multiplier and requires fewer gates, but has a less regular structure and might be more difficult to lay out in VLSI. We can implemented Dadda's scheme with parallel counters other than (3, 2) counters.

Wallace [3] gives an idea about pseudo-adders, which are arrays of full adders without rippling carries. In Wallace, it takes three inputs and reduces them to two outputs. Wallace used pseudo-adders at several levels for the summation of the partial products. Thus, for a 16 by 16-bit multiplication, there are total 16 partial products; so that, at the first level, the Wallace multiplier uses five pseudo-adders to reduce the 16 partial products to eleven. Then at the second level, it uses three pseudo-adders to get eight partial products from eleven products, two pseudo-adders at the next level to get six partial products, two more pseudo- adders to get four partial products, and at last one each at the following two levels finally gets two partial products. Here a fast carry propagate adder (CPA) such as a carry skip adder, carry

select adder, etc. are used to find the final products. So, a Wallace tree multiplier has a total delay of six full adder delays and one CPA delay. When using a shift-and-add algorithm for the same case, it has 15 CPA delay. But the advantage is that it uses less hardware.

2. EXISTING SYSTEM

Symmetric stacking [1] is done by stacking all the input bits so that all the "1" bits are grouped together. After stacking all the given input bits, it can be converted into a binary count to output the 6-bit count. A small 3-bit stacking circuit is used first, to form 3-bit stacks. After that, These 3-bit stacks are combined to make a 6-bit stack using a symmetric technique that adds one extra layer of logic.

2.1. Three-Bit Stacking circuit

Stacker circuit is a circuit that consists of and gate and or gate. This stacker circuit avoids the use of the xor gate which is in the critical path and makes delay in the product of the multiplier and multiplicand to produce the output. X0, X1 and X2 are the three inputs which is said to be a 3-bit stacker circuit then it will produce three output which are Y0, Y1 and Y2 in which number of "1" bits in the outputs is the same as the number of "1" bits in the inputs, but "1" bits are grouped together to the left followed by the "0" bits. Fig-2 shows the 3- bit stacking circuit.

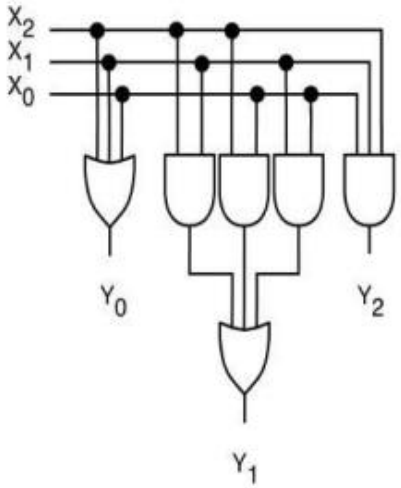

Fig -2: Three-bit stacker circuit

Outputs are then formed by:

Y0 = X0 + X1 + X2 --- (1)

Y1 = X0X1 + X0X2 + X1X2 ----- (2)

Y2 = X0X1X2. ----- (3)

First output will be "1" if any of the inputs is one, the second output will be "1" if any two of the inputs are one, and the last output will be one if all three of the inputs are "1." The output 2 is considered as a majority function which consists of both and gate and or gate which is the implementation of complex CMOS gate.

Figure-3 shows the symmetric stacking technique can be utilized to make a 7:3 counter too.

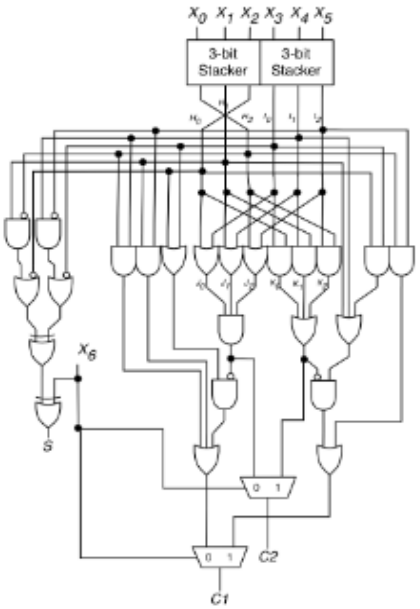

Fig- 3: 7:3 Counter based on symmetric stacking.

3. PROPOSED SYSTEM

3.1 MUX based Full adders (MFA)

In this, a binary counter design is proposed which is designed with the MUX based Full adders (MFA)[2]. The proposed modified full adder circuit shown in Fig-4 consists of two 2:1 MUX and an XOR gate. In this system, one XOR block in the conventional full adder is replaced by a multiplexer block so that the critical path delay gets minimized. The critical path delay is given by:

(i.e) delay= XOR + MUX

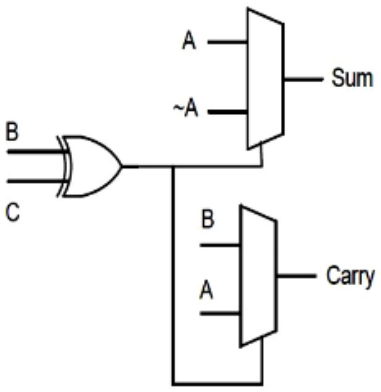

Fig-4: MUX based Full Adder

This proposed Full adder has three inputs. Out of these three given inputs, one input and its complement is given as inputs to the first multiplexer and the other two are given to the XOR gate as its inputs. The output of XOR gate will act as a select line for both the multiplexers. First and second inputs are given as input to the second multiplexer. This unique way of designing leads to the reduction of switching activity, which in turn reduces the power, delays in the critical path is also reduced compared to the existing designs which lead to reduction in delay and thus increasing the speed. Operation of the Full adder can be explained as,

- When both B and C are zero or one, sum = A, carry= B;
- When either of B or C is one and another is zero, sum=$\overline{A}$, carry=A;

ALGORITHM

T = B XOR C

T = 0 Sum = A

Carry = B

T = 1 Sum = $\overline{A}$

Carry = A

3.2 MFA 7:3 Counter

The 7:3 counter is shown in Fig-5. The MFA 7:3 Counter consists of 4 MFAs. The 7-bit inputs are X0, X1.....X7. The inputs X1, X2and X3 are given to the first MFA. There, X2 and X3 undergo XOR operation. The output of the XOR operation is denoted as "t0". The "t0" is given as the select lines to the first multiplexer. If "t0" is 0 then sum is X1and carry is X2. If "t0" is 1 then sum is X1and carry is X1. For all the Full adders the inputs are given and the output is obtained using the output of corresponding XORs. The third MFA gives the sum and last MFA gives the carry1 and carry2 outputs. This results the binary count of the given input.

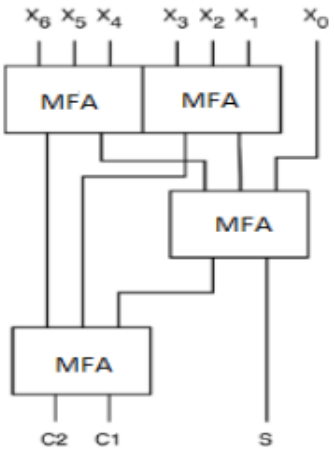

Fig-5: MFA 7:3 Counter.

Table -1: Comparison Table of Counters.

TYPES OF COUNTERS	DELAY(ns)	LUT	SLICE
Symmetric 7:3 Counter	11.589	19	10
MFA 7:3 Counter	7.981	8	4

4. WALLACE MULTIPLIER

The Wallace multiplier [3] is a hardware multiplier design, which consumes less power and its switching speed is faster than other multiplier architectures. Steps of Wallace multipliers are:

- Multiply (logical AND) each bit of one of the arguments, by each bit of the other, gets n^2 results. Depending on position of the multiplied bits, the wires carry different weight.
- Reduce the number of partial products into two by layers of full and half adders.
- Group the wires in two numbers, add them with a conventional adder.

Table -1: Comparison Table of Counters in Wallace Tree Multiplier.

TYPES OF COUNTERS	DELAY(ns)	LUT	SLICE	POWER(W)
Symmetric 7:3 Counter	22.664	204	117	0.034
MFA 7:3 Counter	19.180	154	87	0.027

5. CONCLUSION

MUX based Full adders (MFA) based 7:3 counter is designed and implemented in Wallace Tree Multiplier. The results are compared with the symmetric stacking 7:3 counter. It showed that 7:3 counters designed using MFA has low delay and hence achieve higher speed than other higher counter designs while reducing power consumption.

REFERENCES

[1] Christopher Fritz and Adly T. Fam "Fast Binary Counters Based on Symmetric Stacking," EEE Transactions on Very Large Scale Integration (VLSI) Systems (Volume: 25, Issue: 10 , Oct. 2017).

[2] SnehaP, Dr. M. Thiruveni "An MFA Binary Counter for Low Power Application" International Journal of Pure and Applied Mathematics. Volume 118 No. 20 2018, 4947-4954.

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

[4] Himani, Harmanbir Singh Sidhu "Design and Implementation Modified Booth algorithm and systolic multiplier using FPGA" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 11, November – 2013.

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

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

[7] S. Veeramachaneni, L. Avinash, M. Krishna, and M. B. Srinivas, "Novelarchitectures for efficient (m, n) parallel counters," inProc. 17th ACMGreat Lakes Symp. VLSI, 2007, pp. 188–191.

[8] Swartzlander, E E., Jr., (1973) "Parallel Counters" IEEE Trans. Computers, Vol. 22, pp 1021-1024.