ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 29-34 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

High Speed and Low Power Implementation of FIR Filter Design using DADDA & WALLACE Tree Multiplier

[1]D.G.Jignash,[2]Jami Venkata Suman

[1, 2]Department of Electronics and Communication Engineering, GMR Institute of Technology, Rajam
[1]jigu.gunavanth@gmail.com, [2]venkatasuman.j@gmrit.org

Abstract: The most area consuming arithmetic operations in high-performance circuit's Finite Impulse Response (FIR) multiplication is one. We are using different types of multipliers to reducing the cost of effective parameters in FIR filter design. We use only truncated multipliers in design, by using these multipliers we consume some more area and power, and ineffective results in transposed form. The structural adders and delay elements occupies more area and consumes power in these form so it was a reason to forward the proposed method. In prior FIR filters design with low cost effective results will done by the faithfully rounded truncated multipliers with the carry save additions, In direct form of FIR filter we use MCMAT for multiplication and accumulation operations. With compare of previous designs in direct form we reduce the area while decrease of the full adders, half adders and registers. In MCMAT and design the low cost FIR filters within the best area and power results we implement these improved truncated methods.

Key words: Finite Impulse Response (FIR) Filter, multiple constant multipliers/accumulators with faithfully rounded truncation (MCMAT),truncated multiplier, WALLACE tree multiplier, DADDA multiplier.

INTRODUCTION

In the field of electronic industry digital filters are used extensively. The noise ranges gradually increases by using analog filters for better noise performance can be obtained by using digital filters compared to analog filters. At every intermediate step in digital filter transformation able to perform noiseless mathematical operations. Our design includes the optimization of bit width and hardware resources without any impact on the frequency response and output signal precision [1].

Addition (or subtraction), Multiplication (normally of a signal by a constant)Time Delay i.e. delaying a digital signal by one or more sample periods are three basic mathematical operations used in digital filter is shows in figure 1.By using the mathematical operations mentioned above we can describe the behavior of the filter. The coefficients are multiplied by fixed-point constants using additions, subtractions and shifts in a multiplier block [5].

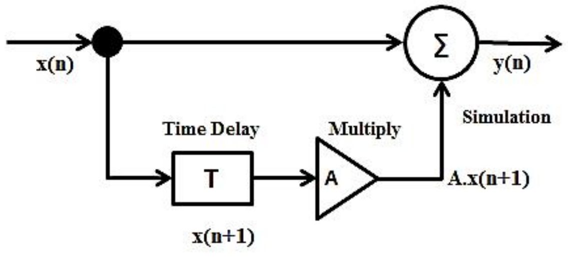

Fig1 Block diagram of a Simple Digital Filter

The unit sample of the signal defined by $\delta\ (n)$ is response one for generating the digital filter impulse response h (n). input sequence x(n) can whose response can be calculated easily if the impulse response is known at every sample index at n=0 a unit impulse is applied so that to attain non zero response for whose value of n is greater than or equal to 0 (i.e,n $\geq$ 0).The impulse response made to be idle so that to avoid uneven responses before applying input. It follows a time invariant property, in which the response delayed by a sample of δ (n-k).

The input $x(n)$ whose response is shown as below

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

In VLSI Signal Processing two types of digital filters are most widely used one is FIR (finite impulse response)and the other is IIR(infinite impulse response). FIR as indicates that the impulses are finite in this filter and phase is kept linear in order to noise distortions and no feedback is used for such a filters. As compared to IIR, FIR is very simple to design. Such type of FIR filters are used in DSP processors for high speed. In Digital Signal Processing Multiplication and addition is of times required. A high speed addition is done by parallel prefix adder and the better version of truncated multiplier with fewer components makes the reduction in delay [4].

For multi-rate applications FIR filters are suitable. i.e: for decrease in sampling rate called decimation or for increase in sampling rate called interpolation, or for both.

Either decimating or interpolating, the calculations are omitted by using FIR, which indeed used for maintaining. For limited calculations IIR is used because all output is found separately, even though there is a need of providing feedback.

In Digital Signal Processing, FIR filters define less number of bits which are designed by using finite-precision In IIR filter by using feedback problems will raise but in FIR filters limited bits are efficient in which there is no feedback. Using fractional arithmetic we can implement FIR filters. But in IIR filters, coefficients with magnitude of less than 1.0are always possible to implement a FIR filter. Using FIR filters is that they require more co-efficient than an IIR filter in order to implement the same frequency response, therefore needing more memory and more hardware resources to carry out mathematical operations.

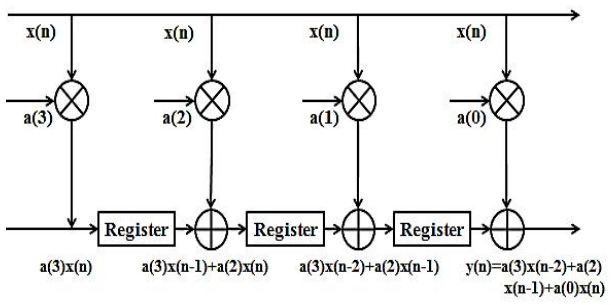

Fig 2: Direct Form FIR Filter Architecture

MULTIPLIERS

Now a day's fast co-processors, digital signal processing chips and graphics processors has created to satisfy customer needs for high speed and area efficient multipliers. Current design range from small, low-performance shift and add multipliers, to large high-performance array and tree multipliers. High performance is achieved in Conventional linear array multipliers, and require huge amount of silicon. Higher performance has gained for Tree structures than linear arrays but the tree interconnection is more complex and less regular.In paper various multiplier architectures have designed, during the past few years. In digital signal processors and microprocessors multiplier is one of the key hardware blocks in most of the digital and high performance systems. With the recent advances in technology ,more efficient multipliers have routedby many researchers. The main motivation behind this paper is to offer high speed and lower power consumption without increase in silicon area.

BINARY MULTIPLICATIONS

Figure 3 represents the process of multiplying two binary numbers, the *multiplicand* and the *Multiplier* according to the multiplier rules; if the inputs are n bit then the output should be 2n. The green box indicates the partial product matrix and the red box indicates a single partial product. The first step in this method is to form the partial product matrix are obtained by Adding the multiplicand and multiplier bits.

Another way to look at this is: If the multiplier bit is 0, the partial product is also 0. If the multiplier bit is 1, the partial product is equal to the multiplicand repeat for every multiplier bit Notice that this gives a number of partial products equal to the width of the multiplier. To obtain the final product the elements in the columns (from right to left) are added using binary logic7. Any carries are carried on to the next column. The result of this operation is stored in one bit of the product and the operation is repeated for each remaining column.

```
b2 a3= b2 and a3          a3   a2   a1   a0        multiplicand
                          b3   b2   b1   b0        multiplier
-------------------------------------------------
                    b0a3 b0a2 b0a1 b0a0
               b1a3 b1a2 b1a1 b1a0
          b2a3 b2a2 b2a1 b2a0
     b3a3 b3a2 b3a1 b3a0
-------------------------------------------------
p7 p6   p5    p4   p3   p2   p1   p0          product
```

Fig 3: 4x4 bit Binary Multiplication

TRUNCATED MULTIPLIER

Truncated multiplication is a technique where only the most significant columns of the multiplication matrix are used and therefore area requirements can be reduced. Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way 'T' defines the degree of truncation and the *T* least significant bits of the product always result in*'0'*.The algorithm behind truncated multiplication is the same as when dealing with non-truncated multiplication regardless of the truncation degree. The effect is illustrated in figure 2, where a truncation degree of *T = 3*, is applied. Notice that the columns to the right of the maroon vertical line are missing [2].

```
b2 a3= b2 and a3        a3 | a2   a1   a0    multiplicand
                        b3 | b2   b1   b0    multiplier
---------------------------+---------------
                      b0a3 |
                 b1a3 b1a2 |
            b2a3 b2a2 b2a1 |
       b3a3 b3a2 b3a1 b3a0 |
---------------------------+---------------
p7  p6   p5    p4    p3    | 0    0    0     product
```

Fig 4: 4x4 bit Binary Multiplication with truncation degree T=3

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 29-34 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

In the truncated multiplier the removal of unnecessary PPBs is composed of three processes:

A. Deletion
B. Truncation
C. Rounding

A) Deletion

In truncated multiplier we start the multiplication process with deletion only.In the partial product bits we remove the more than half of the bits, and then remaining bits become the partial products in the process. This is the main criteria of deletion.

B) Truncation

Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way 'T' defines the degree of truncation and the T Least Significant Bits (LSB) of the product always results in 0. The algorithm behind fixed width multiplication is the same as when dealing with non-fixed width multiplication regardless of the truncation degree. In filter the 0der of non-uniform coefficient quantization is used to minimize the cost of area[3].

C) Rounding

Conventionally an n-bit multiplicand and an n-bit multiplier would render a 2n-bit product. Sometimes an n-bit output is desired to reduce the number of stored bits.Let us consider an instance of 5x5 bit multiplier.

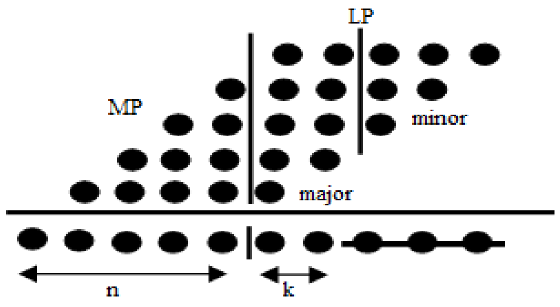

Fig 5: Partial-products matrix of a 5x5 bit multiplier

Truncated multiplication provides an efficient method for reducing the power dissipation and area of rounded parallel multiplier.

WALLACE TREE MULTIPLIER

To reduces the number of partial products to be added into 2 final intermediate results we use Wallace tree. The basic operation of Wallace tree is multiplication of two unsigned integer, an efficient hardware to implement a digital circuit that multiplies two integers is Wallace tree multiplier, designed by an Australian Computer Scientist Chris in 1964.

There are three steps in Wallace Tree:

I. Partial Product Generation Stage
II. Partial Product Reduction Stage
III. Partial Product Addition Stage

I Partial Product Generation Stage

The first step in binary multiplier is Partial product generation. These are the intermediate terms which are generated based on the multiplier value. If the multiplier bit is '0' (zero), then partial product row is also '0' (zero), and if it is '1' (one), then we can copy the multiplicand as it is. Each partial product row is shifted one unit to the left from the 2nd bit multiplication onwards is shown in the above mentioned example. The sign bit in signed multiplication also extended to the left. For a conventional multiplier partial product generators consist of a series of logic AND gates as shown in figure 6.

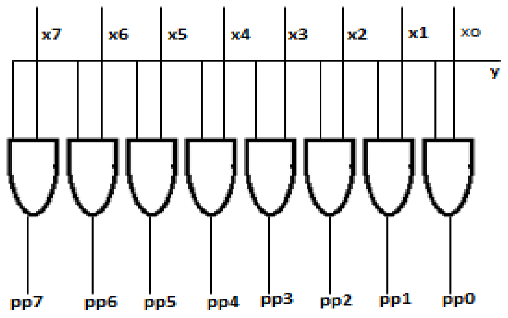

Fig 6: Partial product selection logic for simple multiplication.

In the process of multiplication of two numbers, the main operation is addition of the partial products. Thus, the performance and speed of the multiplier depends on the performance of the adder that forms the core of the multiplier. The multiplier must be pipelined, to achieve higher performance.

II Partial Product Reduction Stage

The design analyses begin with the analysis of the elementary algorithm for multiplication by Wallace Tree multiplier. The algorithm for 8-bits x 8-bits multiplication performs by Wallace Tree multiplier.To complete the multiplication process we have 5 stages. In each stage we used half adders and full adders that are denoted by the red circle for the 1 bit half adder and the blue circle for the 1-bit full adder. Reduce the partial products by using half adders and full adders that are combined to build a carry-save adder (CSA) until there were just two rows of partial products left.

In next step we add the remaining two rows by using a fast carry-propagate adder. For this project to get the final product of the two operands multiplication, ripple-

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 29-34 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

carry adder (RCA) is used, Secondly, the schematic of the conventional 8-bits x 8-bits high speed Wallace Tree multiplier is designed by referring to the algorithm, The block diagram for the conventional high speed 8-bits x 8-bits Wallace Tree multiplier. By the layers of full and half adders we can reduce the number of partial products to 2.The main aim of the proposed architecture is to reduce the overall latency. Thus we increases speed and reduce power consumption. In this design in place of full adders we used compressors. WALLACE tree and DADDA tree are two reduction techniques used which are discussed in paper [6].

III Partial Product Addition Stage

We use multiple half adders and full adder's in these addition stages to sum the products of the multiple bits. In this stage the Wallace multiplier method is using ripple carry adders (RCA) to perform these addition operations.

Three steps used in Wallace method to process the multiplication operation. They are
1. Construction of bit product(s)
2.Exhausting conventional adder, combine all product matrixes to form 2 vectors (carry and sum) outputs in first row.
3.Fast carry-propagate adder, remaining two rows are summed to produce the product

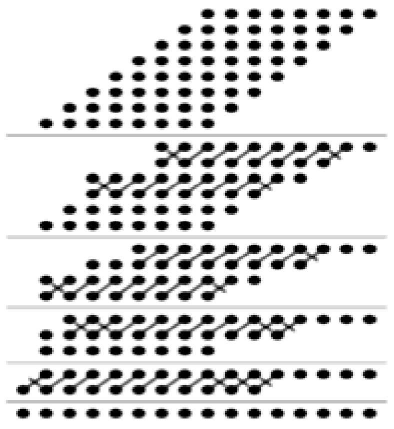

Fig 7: Method of Reduction on8x8 Multiplier

DADDA MULTIPLIER

DADDA hardware multiplier is designed by Luigi Dadda, the computer scientist during 1965. DADDA multiplier is mined form of parallel multiplier [5]. It is increases speed and involves less number of gates. In parallel multiplier we use different types of schemes, DADDA scheme is one of the schemes that fundamentally minimize the number of adder stages required to perform the summation of partial products. This exists succeeded by using full adders and half adders to reduce the number of rows in the matrix, number of bits at each summation stage. While the DADDA multiplication has regular and less complex structure, the process is slower in manner. Further, Wallace tree multiplier is expensive compared to that of DADDA multiplier. In this paper, DADDA multiplier is designed and analyzed by considering different methods using full adders involving different logic styles.

A. Implementation OF DADDA MULTIPLIER

The algorithm of DADDA multiplier is based on the matrix form, as represents in figure 8 the partial product bits are arranged in the first stage is demonstrated in Figure11 Represent way of working process in DADDA multiplier.

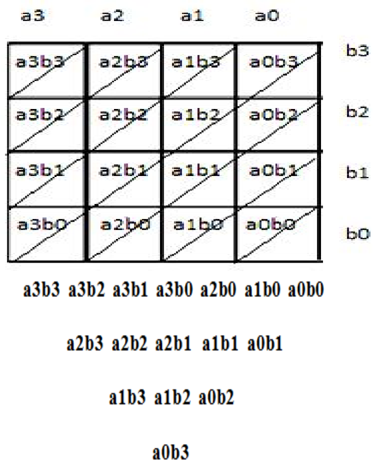

Fig 8: 4x4 DADDA algorithm

B. Steps involved in DADDA multipliers Algorithm

Multiply (that is AND) each bit of one of the arguments, by each bit of the other, producing N results. The wires carry different weights depending on situation of the multiplied bits in figure 9. Reduce the number of partial products to 2 layers of full adders. Group the wires in two numbers and add them with a conventional adder.

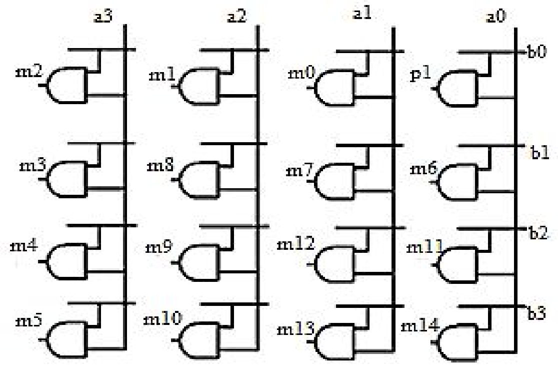

Fig 9: Product terms generated by a collection of AND gates

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 29-34 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Ripple Carry Adder is used to add more number of additions to be accomplished with the carry in sand carry outs that is to be chained. Hence multiple adders are used in ripple carry adder. By using several full adders it is possible to create a logical circuit to add multiple-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous full adder. This kind of adder is known as ripple carry adder. Since each carry bit "ripples" to the next full adder, recommended architecture of DADDA multiplier algorithm using RCA represent procedure. Take any 3 values with the same weights and give them as input to a full adder. The result will be an output wire of the same weight. When multiplication is taken at the first stage partial product obtained. The data's are taken with 3 wires and added by using adders and the carry of each stage is added with next two data's in the same stage.Partial products reduced to 2 layers of full adders with same procedure. At the final stage, same method of ripple carry adder method is performed and hence product terms p1 to p8 is obtained in figure 10.

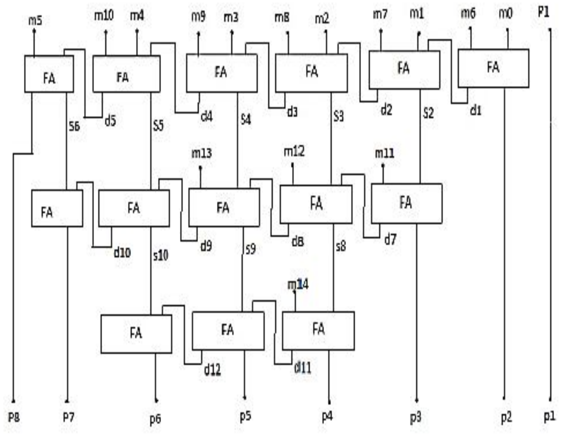

Fig 10: Implementation of 4x4 DADDA Multiplier

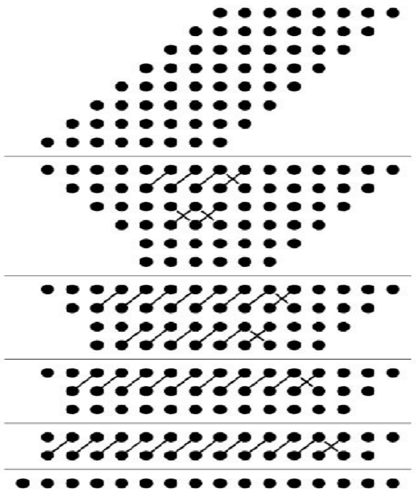

Fig 11: Method8X8 DADDA multiplier

Figure 11 indicates method of explaining the process of multiplication of partial products is same as dealing with reducing the number of stored bits, each partial product is shift one unit to the left from the 2nd bit multiplication ahead.

VII. EXPERIMENTAL RESULTS

Parameter	Normal FIR Multiplier		Truncated Multiplier		Wallace Multiplier		Dadda Multiplier	
	4 bit	8 bit	4 bit	8 bit	4 bit	8 bit	4 bit	8 bit
Memory	241316KB	257316KB	243300 KB	257828KB	243812 KB	263972 KB	242340 KB	267300 KB
Delay	11.409ns	4.040ns	10.937ns	4.040ns	11.018ns	4.040ns	11.091ns	4.040ns
Power	0.21mw	0.88mw	0.26mw	1.21mw	0.21mw	0.24mw	0.24mw	1.13mw

From the experimental result analysis it is clear that the delay requirement for the entire WALLACE, DADDA, Truncated and normal FIR multipliers are same for 8bit as compared to 4bit operation. That is the speed of execution is same for the WALLACE, DADDA, Truncated and normal FIR multipliers. In the same way the power consumption is better in DADDA and WALLACE when compare with the Truncated multipliers.

ISSN 2278-3091

CONCLUSION

In this paper the design of direct form is reduced the area due to decrease of the full adders, half adders and registers were compared to previous design. We had designed 4bit and 8bit multiplier and compared the power, area and delay reports for multipliers such as DADDA, normal multiplier and WALLACE tree. Finally the truncated one is not much efficient in term of power factor and DADDA, WALLACE tree is efficient in delay and power analysis. In future we can design and implement it for large bit width also.

REFERENCES

[1] Shen-Fu Hsiao, Jun-Hong Zhang Jian, And Ming-Chih Chen,"Low-Cost Fir Filter Designs Based On Faithfully Rounded Truncated Multiple Constant Multiplication/Accumulation",IEEE Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 60, No. 5, May 2013.

[2] H. J. Ko And S. F. Hsiao, "Design And Application Of Faithfully Rounded And Truncated Multipliers With Combined Deletion, Reduction, Truncation, And Rounding," IEEE Trans. Circuits Syst. Ii, Exp. Briefs, Vol. 58, No. 5, Pp. 304–308, May 2011.

[3] D. Shi And Y. J. Yu, "Design Of Linear Phase Fir Filters With High Probability Of Achieving Minimum Number Of Adders," IEEE Trans. Circuits Syst.I, Reg. Papers, Vol. 58, No. 1, Pp. 126–136, Jan. 2011.

[4] Blad and O. Gustafsson, "Integer Linear Programming-Based Bit-Level Optimization for High-Speed Fir Filter Architecture," Circuits Syst. SignalProcess. Vol. 29, No. 1, Pp. 81–101, Feb. 2010.

[5] Y. Voronenko and M. Puschel, "Multiplierless Multiple Constant Multiplication," Acm Trans. Algorithms, Vol. 3, No. 2, Pp. 1–38, May 2007.

[6] S. Hwang, G. Han, S. Kang, and J.-S. Kim, "New Distributed Arithmetic Algorithm for Low-Power Fir Filter Implementation," IEEE Signal Process.Lett. Vol. 11, No. 5, Pp. 463–466, May 2004

.

.