International Journal of Hybrid Information Technology
Vol. 7, No. 6 (2014), pp. 325-336
http://dx.doi.org/10.14257/ijhit.2014.7.6.28

Design of Digital FIR Filter Based on MCMAT for 12 bit ALU using DADDA & WALLACE Tree Multiplier

Jami Venkata Suman[1], D. G. Jignash[2] and B. I. Neelgar[3]

[1]*Assistant Professor, Department of ECE, GMR Institute of Technology, Rajam, India, Mobile: +91-9985571930, Email: venkatasuman.j@gmrit.org.*
[2]*PG Student, Department of ECE, GMR Institute of Technology, Rajam, India, jigu.gunavanth@gmail.com.*
[3]*Professor & HOD, Department of ECE, GMR Institute of Technology, Rajam, India, hod_ece@gmrit.org*

Abstract

Over a period of time, there are different types of multipliers used to minimize cost of constructive parameters in digital FIR filter design in different forms. In these multipliers truncated multiplier design play's an essential role. Due to multiple adders and delay elements the circuit in transpose form occupies more area and consumes power. In the proposed method effectiveness in different parameters are achieved with low cost, high speed and effective results are done by faithfully rounded truncated multipliers with operations of carry save adder. Direct form of FIR filter utilizes based on MCMAT for multiplication and accumulation operations; it will reduce the area by decreasing the number of different components like structural adders and registers. By using MCMAT operation in the digital FIR filters, optimal results can be obtained in terms of power and area specifications. The proposed 12 bit Arithmetic and Logic Unit is designed along with MCMAT based digital FIR filter for reducing area and increasing speed in real time applications.

Keywords: *Finite Impulse Response Filter, multiple constant multipliers/accumulators with faithfully rounded truncation (MCMAT), TRUNCATED multiplier, WALLACE Tree multiplier, DADDA multiplier*

1. Introduction

In field of electronic industry, digital filters are used extensively. Noise range can be gradually increases by using analog filters. Better noise performance can be obtained by using digital filters compared to analog filters. The proposed design includes optimization of bit width and hardware resources without any impact on frequency response and output signal [2]. Three basic mathematical operations such as Addition (or) subtraction, Multiplication, delaying digital signal for one or more sample periods that are used in digital filter are shown in Figure 1.1. By using mathematical operations, behavior of the filter can be described. Coefficients are multiplied by fixed-point constants using additions, subtractions and shifts in a multiplier block [5].

ISSN: 1738-9968 IJHIT

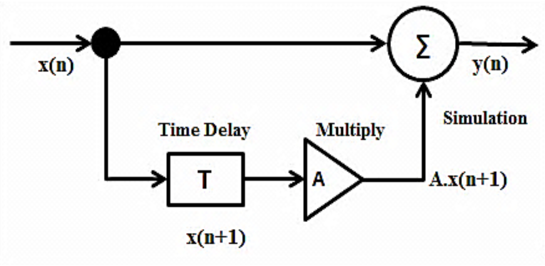


Figure 1.1. Block Diagram of a Simple Digital Filter

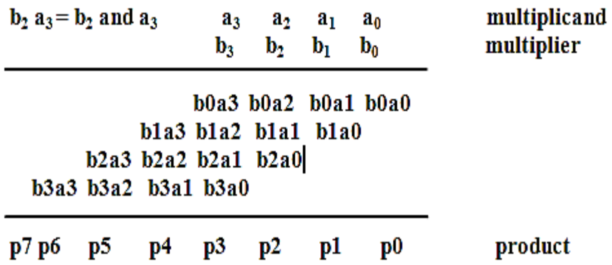
In VLSI Signal Processing, two types of digital filters widely used, one is FIR (Finite Impulse Response) and other is IIR (Infinite Impulse Response). FIR indicates that impulse is finite. In this filter phase is kept linear in order to avoid noise distortions and no feedback is used. As compared to 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 requires a lot of time. High speed addition is done by parallel prefix adder and better version of truncated multiplier with fewer components [4]. For limited calculations IIR is used because all output is found separately, even though there is need to provide feedback. Digital Signal Processing, FIR filters define less number of bits which is designed by using finite-precision. If IIR filter uses feedback problem will raise. In FIR filter limited bits are efficient, which there is no feedback. Using fractional arithmetic can implement FIR filters. FIR filters require more co-efficient than IIR filter in order to implement same frequency response, therefore needing more memory and hardware resources to carry out mathematical operations [1]. The proposed FIR filter design can have less number of coefficients.

2. Multipliers

Now a day’s fast co-processor, digital signal processing chips and graphics processors has created to satisfy customer for high speed and area efficient multipliers. Current design ranges from small, low-performance shift and ADD multipliers to large high-performance array of multipliers. Higher performance is gained by Tree structures than linear arrays, but tree interconnection is more complex, less regular. In digital signal processors and microprocessors multiplier is one of the key hardware blocks in most of digital and high performance systems. Main motivation behind this paper is to offer high speed and lower power consumption without increase in silicon area.

2.1. Binary Multiplications

Figure 2.1 represent multiplication process of two binary numbers, multiplicandand Multiplieraccording to the multiplier rules. If inputs are n bit then output should be 2n bits. The first step of method is to form the partial product matrix and this is obtained by adding the multiplicand and multiplier bits. If the multiplier bit is 0, partial product is 0. If multiplier bit is 1, partial product is equal to multiplicand, repeats for every multiplier bit. Note that number of partial products equal to the width of the multiplier. To obtain final product elements in columns (from right to left) are added using binary logic 7. Any carries are carried on to next column. Result of this operation is stored in one bit of product and operation is repeated for each remaining column.

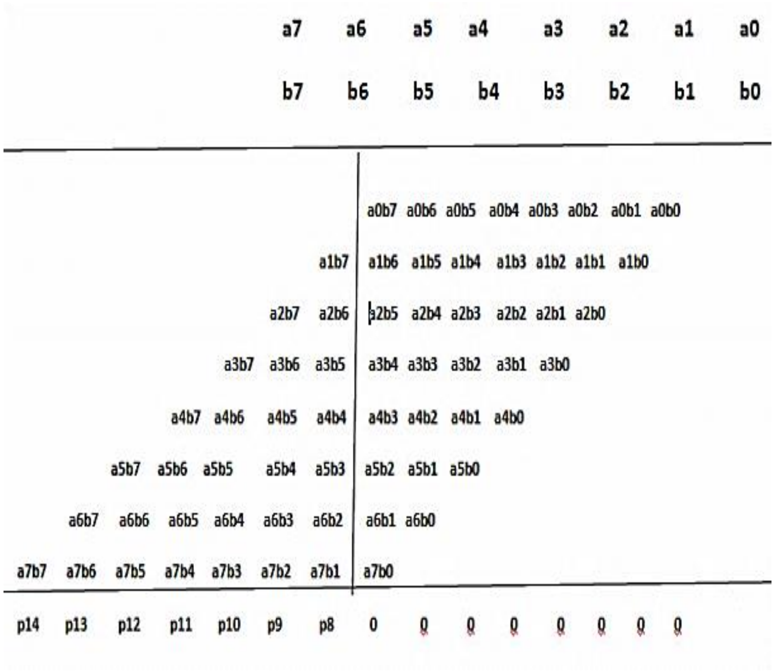


```
b2 a3 = b2 and a3          a3    a2    a1    a0          multiplicand
                           b3    b2    b1    b0          multiplier
--------------------------------------------------
                        b0a3  b0a2  b0a1  b0a0
                  b1a3  b1a2  b1a1  b1a0
            b2a3  b2a2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

Figure 2.1. 4x4 bit Binary Multiplications

3. Truncated Multiplier

Truncated multiplication is a technique, only most significant columns multiplication matrixes are used, therefore area requirements shrinkage will be done. Truncation is a method, least significant amount of columns not formed in this way. ‘T’ defines the degree of truncation and T least significant bits of the product always result in ‘0’. Algorithm behind truncated multiplication is same, when dealing with non-truncated multiplication regardless of truncation degree. Effect is illustrated in Figure 3, where truncation column in the partial product matrix is not formed. Degree of T=8 and T=12 is applied. Notice that columns to the right of maroon vertical line are missing. In implementation of truncated multiplier extended bit width of multiplier operands. By increasing bit width complexity will be increased in operation of multiplication. In truncated method 8 bit and 12 bit operations are implemented by three operations of truncated multiplier which follows operation of 4 bit truncated multiplier operation those are deletion, truncation, rounding concepts [3] shows in Figure 3.1. In truncated multiplier removal of unnecessary PBs is composed of three processes.



```
                                               a7   a6   a5   a4   a3   a2   a1   a0
                                               b7   b6   b5   b4   b3   b2   b1   b0
-------------------------------------------------------------------------------------
                                            | a0b7 a0b6 a0b5 a0b4 a0b3 a0b2 a0b1 a0b0
                                       a1b7 | a1b6 a1b5 a1b4 a1b3 a1b2 a1b1 a1b0
                                  a2b7 a2b6 | a2b5 a2b4 a2b3 a2b2 a2b1 a2b0
                             a3b7 a3b6 a3b5 | a3b4 a3b3 a3b2 a3b1 a3b0
                        a4b7 a4b6 a4b5 a4b4 | a4b3 a4b2 a4b1 a4b0
                   a5b7 a5b6 a5b5 a5b4 a5b3 | a5b2 a5b1 a5b0
              a6b7 a6b6 a6b5 a6b4 a6b3 a6b2 | a6b1 a6b0
         a7b7 a7b6 a7b5 a7b4 a7b3 a7b2 a7b1 | a7b0
-------------------------------------------------------------------------------------
         p14  p13  p12  p11  p10  p9   p8     0    0    0    0    0    0    0    0
```

Figure 3.1. 8x8 Bit Binary Multiplication with Truncation Degree T=8

3.1. Deletion

In truncated multiplier multiplication process is started with deletion operation only. In partial product more than half of bits are removed and then remaining bits become partial products in process. This is the main criteria of deletion.

3.2. Truncation

Truncation is a method where least significant columns in partial product matrix are not formed. 'T' defines degree of truncation. Least Significant Bits (LSB) of product always results in 0. Algorithm behind fixed width multiplication is same as when dealing with non-fixed width multiplication of the truncation degree. In FIR filter, zero order of non-uniform coefficient quantization is used to minimize cost and area [4].

3.3. Rounding

Conventionally an n-bit multiplicand and n-bit multiplier would render a 2n-bit product. Sometimes an n-bit output is desired to reduce number of stored bits. Let us consider instance of 8x8 bit multiplier.

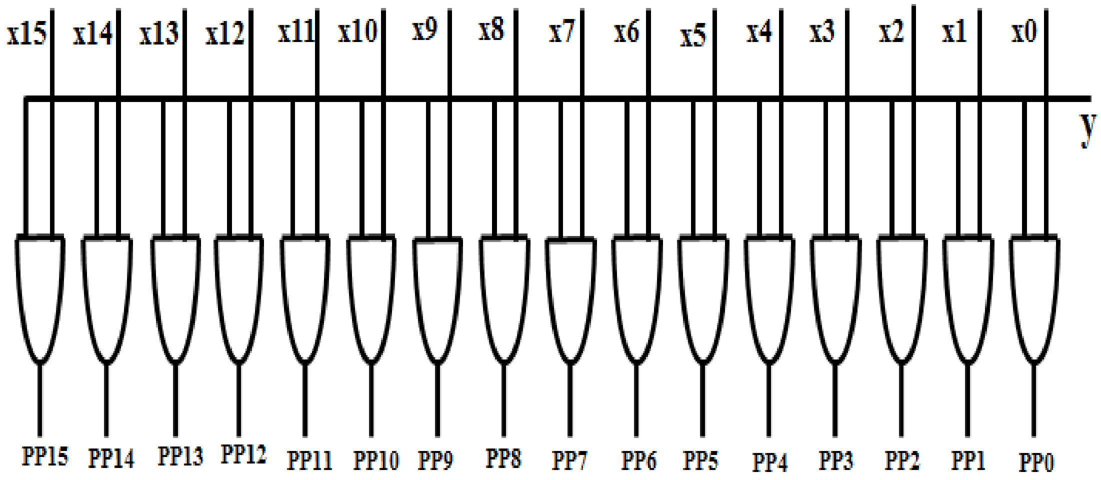


Figure 3.2. Partial Product Selection Logic for 8x8 Bit Multiplication

Let us consider an instance of 12x12 bit multiplier in Figure 3.3. Truncated multiplication provides an efficient method for reducing power dissipation and area of rounded parallel multiplier. In those huge bit widths of implementations mainly focus on the performance of multiplier in lot of complexity levels also. In case of truncated method area shrinkage will automatically done at any number of bit width. So, other parameters of power and delay with those parameters depend on cost effective will be decayed a lot.

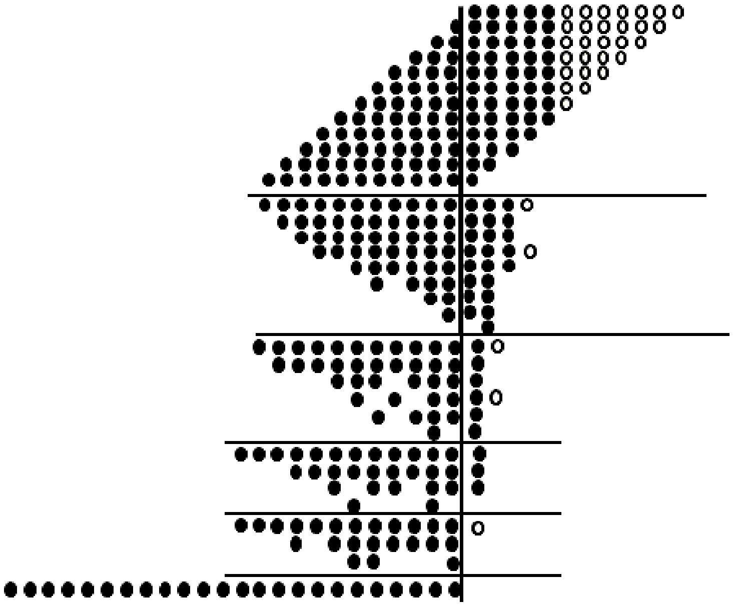


Figure 3.3. 12x12 Bit Multiplication with Truncation Degree T=12

4. Wallace Tree Multiplier

To reduce number of partial products that are to be added in final intermediate result Wallace Tree multiplier is used. Basic operation of Wallace Tree is multiplication of two unsigned integer. An efficient hardware is required to implement a digital circuit, that multiplies two integers is Wallace Tree multiplier that is designed by an Australian Computer Scientist Chris in 1964. There are three steps in Wallace tree multiplier.

4.1. Partial Product Generation Stage

First step of binary multiplier is generation of Partial product. Which are generated based on multiplier value. If multiplier bit is '0' (zero), then partial product row is also '0' (zero), if it is '1' (one). Each partial product row is shifted one unit to left from 2nd bit multiplication. Sign bit in signed multiplication also extended to left. For a conventional multiplier partial product generators are used. These contain a series of logic AND gates as shown in Figure 4.1. In this process of multiplication of two numbers, main operation is addition of partial products. Thus, performance and speed of multiplier depends on performance of the adder forms core of multiplier. Multiplier must be pipelined, to achieve higher performance.

4.2. Partial Product Reduction Stage

The design analyses begin with analysis of elementary algorithm for multiplication using Wallace Tree multiplier. Algorithm for 8-bits x 8-bits multiplication performs by Wallace Tree multiplier shows in Figure 4.1. The multiplication process is done in 5 stages. In each stage half adders are used and full adders that are denoted by 1 bit half adder and circle for 1-bit full adder. Partial products can be reduced by using half adders and full adders that are combined to build a carry-save adder (CSA). In next step remaining two rows are added by using a fast carry-propagate adder. Ripple-carry adder (RCA) uses the schematic of the conventional 8-bits x 8-bits. High speed Wallace multiplier is designed by referring to the algorithm. Block diagram for conventional high speed 8-bits x 8-bits Wallace Tree multiplier is shown in Figure 4.1. The main aim of proposed architecture is to reduce overall latency. WALLACE TREE and DADDA are two reduction techniques that are discussed in [8].

4.3. Partial Product Addition Stage

In this stage Wallace Tree multiplier method, ripple carry adders (RCA) are used to perform these addition operations.Three steps are used in Wallace method to process 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.

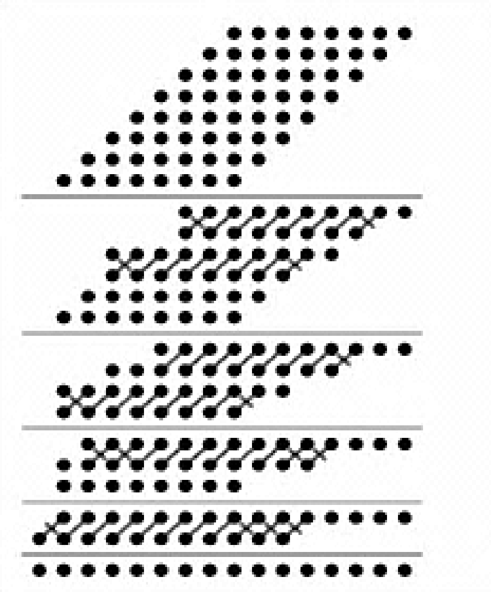


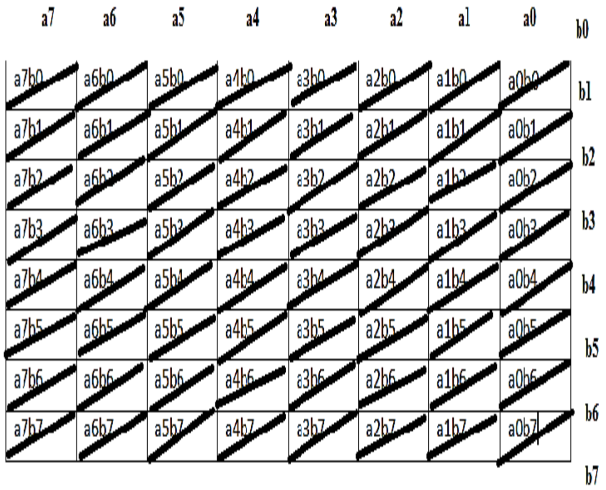
Figure 4.1. Method of Reduction on 8x8 Multiplier

5. Dadda Multiplier

DADDA multiplier is designed by Luigi Dadda, computer scientist during 1965. DADDA multiplier is mined form of parallel multiplier [6]. It increases speed and involves less number of gates. The parallel multiplier uses different type of schemes, DADDA is one of schemes that fundamentally minimize number of adder stages required to perform summation of partial products. By using full adders and half adders number of rows in matrix, number of bits at each summation stage can be reduced. 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.

5.1. Implementation of DADDA MULTIPLIER

Algorithm of DADDA multiplier is based on matrix form, as represents in Figure 5.1. The partial product bits are arranged in first stage are demonstrated in Figure 9. It represents the way of working process in DADDA multiplier.



a7	a6	a5	a4	a3	a2	a1	a0
a7b0	a6b0	a5b0	a4b0	a3b0	a2b0	a1b0	a0b0
a7b1	a6b1	a5b1	a4b1	a3b1	a2b1	a1b1	a0b1
a7b2	a6b2	a5b2	a4b2	a3b2	a2b2	a1b2	a0b2
a7b3	a6b3	a5b3	a4b3	a3b3	a2b3	a1b3	a0b3
a7b4	a6b4	a5b4	a4b4	a3b4	a2b4	a1b4	a0b4
a7b5	a6b5	a5b5	a4b5	a3b5	a2b5	a1b5	a0b5
a7b6	a6b6	a5b6	a4b6	a3b6	a2b6	a1b6	a0b6
a7b7	a6b7	a5b7	a4b7	a3b7	a2b7	a1b7	a0b7

Figure 5.1. 8x8 DADDA Algorithms

5.2. Steps Involved in DADDA Multipliers Algorithm

The wires carry different weights depending on situation of multiplied bits in Figure 5.2. To reduce number of partial products 2 layers of full adders are used. Group wires in two numbers are added with a conventional adder. Ripple Carry Adder is used to add more number of additions that are to be accomplished with carry in sand carry outs that are to be chained. By using several full adders it is possible to create a logical circuit to add multiple-bit numbers. Each full adder input is Cin, which is Cout of previous full adder. Since each carry bit "ripples" to next full adder, the architecture of DADDA multiplier algorithm must use RCA procedure. Data is taken with 3 wires and added by using adders. The carry of each stage is added with next two data's in same stage. At the final stage, same method of ripple carry adder is performed and hence product terms p1 to p8 is obtained in Figure 5.3.

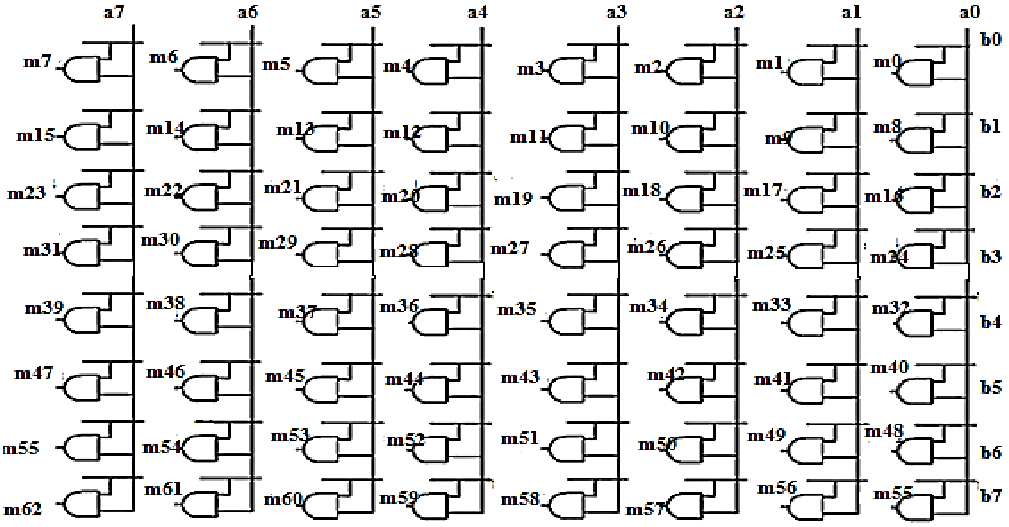


Figure 5.2. Product Terms Generated by a Collection of AND Gates

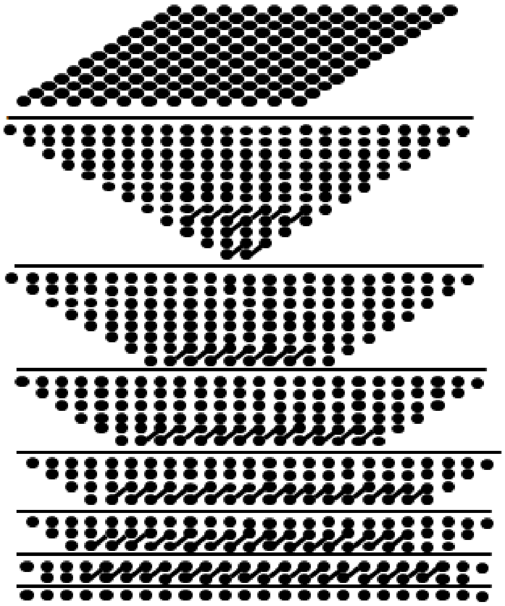


Figure 5.3. 12 x 12 DADDA Process of Multiplication

Figure 5.3 indicates the process of multiplication of partial products is same as dealing with reducing the number of stored bits. Each partial product is shifted one unit to the left from the 2nd bit multiplication ahead.

6. Experimental Results

Table 6.1. Performance Comparison for Normal FIR and Truncated Multiplier

Parameter	Normal FIR Multiplier			Truncated Multiplier		
	4 bit	8 bit	12bit	4 bit	8 bit	12bit
Power(mw)	0.26	1.11	2.93	0.33	1.53	5.88
Delay(ns)	11.409	4.040	4.040	10.937	4.040	4.283
No of LUT's	31	128	337	39	177	677
Memory(kb)	238284	254668	259372	240268	255116	299404

Table 6.2. Performance Comparison for Wallace Tree and Dadda Multiplier

Parameter	Wallace Tree Multiplier			Dadda Multiplier		
	4 bit	8 bit	12bit	4 bit	8 bit	12bit
Power(mw)	0.26	0.31	2.43	0.24	1.13	5.30
Delay(ns)	11.018	4.040	4.040	11.091	4.040	4.040
No of LUT's	31	36	280	35	165	610
Memory(kb)	241292	260428	317900	239884	263756	345548

Table 6.1 and Table 6.2 shows the implementation results using Xilinx ISE simulator. By analyzing the experimental results it is clear that the delay requirement for the entire WALLACE Tree, DADDA, Truncated and normal FIR multipliers are same for 8 bit as

compared to 12 bit operation. The speed of execution for the WALLACE Tree, DADDA, Truncated and normal FIR multipliers are different. In the same way the power consumption is better in DADDA and WALLACE Tree compared with the Truncated and normal FIR multipliers.

7. Integrated ALU

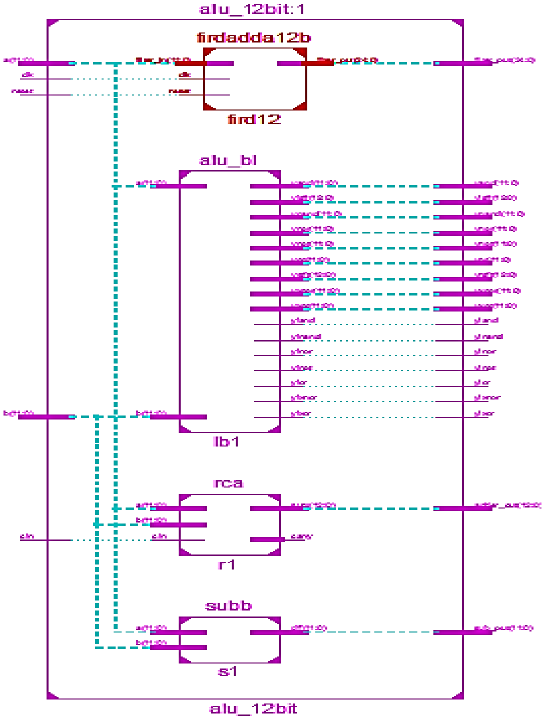


Figure 7.1. RTL Schematic of Integrated ALU

Figure 7.1 represents the RTL schematic for proposed design of 12 bit ALU with MCMAT based DADDA multiplier. When compared to normal ALU and FIR Filter, proposed design of 12 bit Integrated ALU will gives better performance. The execution process of 12 bit integrated ALU is faster than normal ALU and FIR Filter.

Table 7.1. Performance Comparation for Normal ALU, FIR Filter and 12bit Integrated ALU with Wallacetree and Dadda Multiplier

Parameter	12 bit Normal ALU	12 bit FIR Filter	12 bit Integrated ALU	
			12 bit ALU with DADDA multiplier	12bit ALU with Wallace Tree multiplier
Power(mw)	1.13	2.93	6.43	3.48
Delay(ns)	17.859	4.040	17.859	13.664
No of LUT"S	130	337	740	401
Memory(kb)	252136	259372	382764	301800

The performance comparisons of proposed 12 bit integrated ALU design is shown in Table 7.1. The12 bit ALU with MCMAT based WALLACE Tree multiplier will gives better results than 12 bit ALU with MCMAT based DADDA multiplier. The RTL schematic of 12 bit ALU with WALLACE Tree multiplier is similar to theRTL schematic of 12 bit ALU with DADDA multiplier.

8. Conclusion

In this paper direct form of digital FIR filter is recommended. It reduces area due to decrease of number of elements which are structural adders and storage elements. The design of 8 bit and 12 bit multiplier are proposed and the parameters such as power, area and delay for DADDA, normal multiplier and WALLACE Tree multiplier are compared. At last it is observed that truncated multiplier is not much efficient in terms of power factor and DADDA multiplier, WALLACE Tree multipliers are efficient in terms of delay and power analysis. The designing of 12 bit Arithmetic Logic Unit is done with MCMAT based 12 bit FIR using WALLACE Tree and DADDA multipliers. The results obtained shows that WALLACE Tree multiplier is efficient than DADDA multiplier, because speed and power are better in WALLACE Tree multiplier. By integrating ALU with MCMAT based digital FIR filter design in the real time applications, the overall speed and area can be improved. In future, it can be designed and implemented 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, **(2013)**, pp.287-291.

[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, **(2011)**, pp. 304–308.

[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, **(2011)**, pp. 126–136.

[4] Blad and O. Gustafsson, “Integer Linear Programming-Based Bit-Level Optimization for High-Speed Fir Filter Architecture”, Circuits Syst. Signal Process. Vol. 29, No. 1, **(2010)**, pp. 81–101.

[5] Y. Voronenko and M. Puschel, “Multiplierless Multiple Constant Multiplication”, ACM Trans. Algorithms, Vol. 3, No. 2, **(2007)**, pp. 1–38.

[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, **(2004)**, pp. 463–466.

[7] Bishwajeet Pandey, Jyotsana Yadav, M Pattanaik and Nitish Rajoria, “Clock Gating Based Energy Efficient ALU Design and Implementation on FPGA”, IEEE International Conference on Energy Efficient Technologies for Sustainability, **(2013)**, pp. 93-97.

[8] P. K. Meher, “New approach to look-up-table design and memory-based realization of FIR digital filter”, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, **(2010)**, pp. 592–603.

[9] P. K. Meher, S. Candrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic”, IEEE Trans. Signal Process., vol. 56, no. 7, **(2008)**, pp. 3009–3017.

[10] F. Xu, C. H. Chang, and C. C. Jong, “Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common sub expressions”,IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26, no. 10, **(2007)**, pp. 1898–1907.

[11] M. M. Peiro, E. I. Boemo, and L. Wanhammar, “Design of high-speed Multiplierless filters using a Nonrecursive signed common sub expression algorithm”, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 49, no. 3, **(2002)**, pp. 196–203.

[12] D.G.Jignash and Jami Venkata Suman, “High Speed and Low Power Implementation of FIR Filter Design using DADDA & WALLACE Tree Multiplier”, International Conference on Advances in Computer Science and Software Engineering, **(2014)**, October 10th, St.Anna’s College of Engineering& Technology, Chirala, Andhra Pradesh, India

Authors

Jami Venkata Suman, he received his Bachelor of Engineering in Electronics and Communication Engineering from Tontadarya College of Engineering, Gadag, under Visvesvaraya Technological University, Belgaum, Karnataka state, India in the year 2004 andMaster of Technology in VLSI System Design from Annamacharya Institute of Technology and Sciences, Rajampet, under JNTUH, Hyderabad, Andhra Pradesh state, India in the year 2008. He is currently working as an Asst. Prof in the Department of Electronics and Communication Engineering at GMR Institute of Technology, Rajam. His areas of interest include VLSI and Radar Signal Processing.

D. G. Jignash, pursuing Master of Technology in VLSI & Embedded Systems at GMR Institute of Technology, Rajam and Received Bachelor of Technology in Electronics and Communication Engineering from Khader Memorial College of Engineering and Technology, under JNTUH in the year 2012. His areas of interest include VLSI and Embedded Systems.

B. I. Neelgar, he received his B.E. in Electronics and Communication Engineering from Karnataka University, Dharawad, Karnataka State, India in the year 1984 and M. Tech in Digital Electronics and Advanced Communication from Karnataka Regional Engineering College (NITK), Surathkal, Mangalore University, Karnataka State, India in the year 1995. Presently he is pursuing his Ph.D from Jawaharlal Nehru Technological University (JNTUH), Hyderabad and waiting for final Viva-Voce. He is Professor and Head of the Department of Electronics and Communication Engineering at GMR Institute of Technology, Rajam. He has an Industry experience of 4.5 years and Teaching experience of 23 years. His areas of interest include Antennas, Antenna Arrays, Communication and VLSI.