International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 4, No. 2
May 2015

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 4, No. 2, May 2015

Research Paper

DESIGN AND IMPLEMENTATION BIT LEVEL OPTIMIZATION OF ADDER-TREES FOR MULTIPLE CONSTANT MULTIPLICATION FOR EFFICIENT FIR FILTER

Gunda Hymavathi[1]*, K Kiran Kumar[2] and D Sailaja[3]

Corresponding Author: **Gunda Hymavathi ✉ gunda.hymavathi301@gmail.com*

The most area and power consuming arithmetic operation in high-performance circuits like Finite Impulse Response (FIR), multiplication is one. There are different types of multipliers to reducing the cost and effective parameters in FIR filter design. Among those this paper use modified Wallace multiplier in the fir design. The structural adders and delay elements occupies more area and consumes power in this form so it was a reason to forward the proposed method. In prior FIR filters design with low cost effective results will done with the carry skip additions. Along with that the proposed method modified Wallace multiplier based fir filter is also designed in this paper to make the FIR filter design is suitable for low power applications.

Keywords: Finite Impulse Response (FIR) Filter, Modified Wallace tree multiplier, Carry skip adder

INTRODUCTION

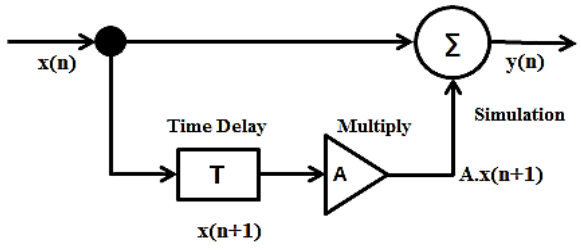
In the field of electronic industry digital filters are used extensively. The noise ranges gradually increases by using analog filters for better noise performance can be obtained by using digital filters compared to analog filters. At every intermediate step in digital filter transformation able to 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 (Bull and Horrocks, 1991). 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 (Voronenko, 2007).

The unit sample of the signal defined by $\delta(n)$ is response one for generating the digital filter

[1] M. Tech Student, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.
[2] Associate Professor, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.
[3] Associate Professor, Department of ECE, Sree Kavitha Engineering College, Khammam, Telangana 507 112, India.

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 ≥ 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).

Figure 1: Block Diagram of a Simple Digital Filter

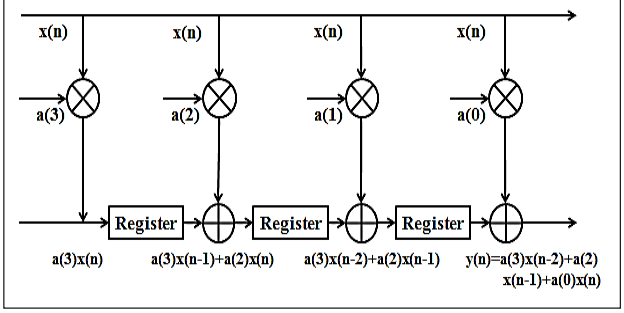


The input $x(n)$ whose response is shown as below:

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

In VLSI Signal Processing two types of digital filters are most widely used one is Finite Impulse Response (FIR) and the other is Infinite Impulse Response (IIR). 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 (Park and Kang, 2001). For multi-rate applications FIR filters are suitable 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 1.0 are 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.

Figure 2: Direct Form FIR Filter Architecture



MULTIPLIERS

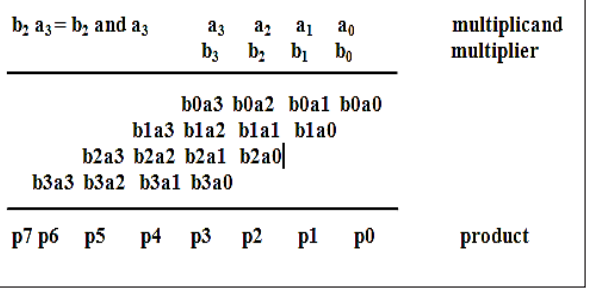
Now a day's fast co-processors, digital signal processing chips and graphics processors has created to satisfy customer needs for high speed and area efficient multipliers. Current design range from small, low-performance shift and add multipliers, to large high-performance array and tree multipliers. 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 routed by many researchers. The main motivation behind this paper is to offer 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 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 logic 7. 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.

Figure 3: 4x4 bit Binary Multiplication



WALLACE TREE MULTIPLIER

To reduces the number of partial products to be added into two 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:

- Partial Product Generation Stage
- Partial Product Reduction Stage
- Partial Product Addition Stage

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. 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.

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-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 (Meher and Pan, 2011).

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 two vectors (carry and sum) outputs in first row.
3. Fast carry-propagate adder, remaining two rows are summed to produce the product.

Figure 4: Method of Reduction on 8x8 Multiplier

MODIFIED WALLACE MULTIPLIER

In the paper a MAC (Multiplication and accumulation) is implemented using modified Wallace multiplier and for performing the addition the Ripple Carry Adder (RCA) is used . Modified Wallace multiplier is also known as DADDA multiplier.For improving the performance the addition is implemented using reversible 3:2 compressor through which the number of adders required for performing the operation is reduced.A modified Wallace multiplier is an efficient hardware implementation of digital circuit which multiplies two integers. Generally in the reduction phase of conventional Wallace multipliers, many full adders and half adders are used when compared to modified Wallace multipliers. As we know that half adders do not reduce the number of partial product bits. Therefore, it is necessary to minimize the number of half adders used in a multiplier which reduces the hardware complexity. Hence, a modification to the Wallace reduction is done in which the delay is the same as for the conventional Wallace reduction. The modified

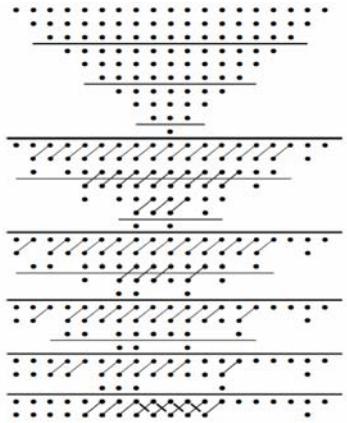
reduction method greatly reduces the number of half adders with a very slight increase in the number of full adders. Reduced complexity Wallace multiplier reduction consists of three stages. First stage the N x N product matrix is formed and before passing on to the second phase the product matrix is rearranged to take the shape of inverted pyramid. During the second phase the rearranged product matrix is grouped into non-overlapping group of three as shown below, single bit and two bits in the group will be passed on to the next stage and three bits are given to a full adder. The number of rows in each stage of the reduction phase is calculated by the formula.

$$r_{i+1} = 2[r_i / 3] + r_i \bmod 3$$

if $r_i \bmod 3 = 0,$ then

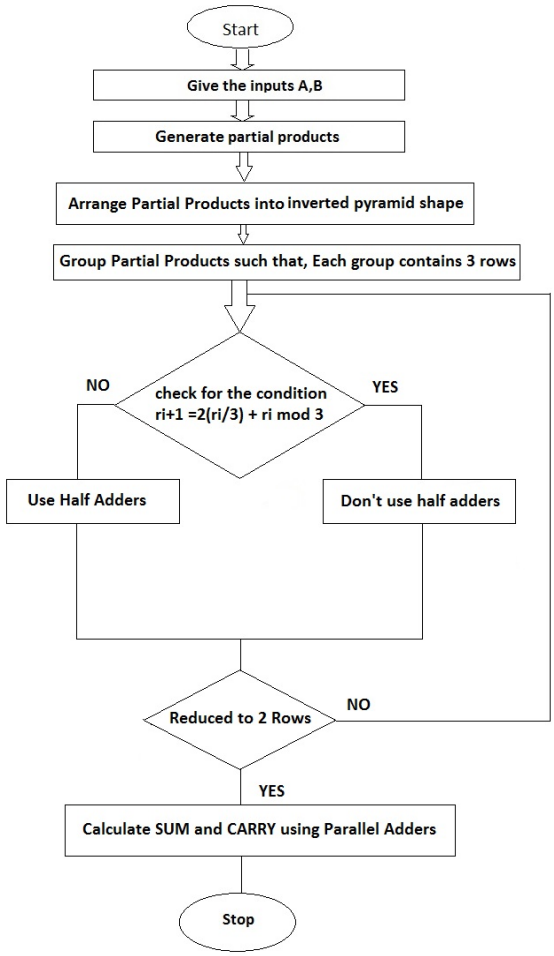
If the value calculated from the above equation for number of rows in each stage in the second phase and the number of rows that are formed in each stage of the second phase does not match, only then the half adder will be used. The final product of the second stage will be in the height of two bits and passed on to the third stage. During the third stage the output of the second stage is given to the carry propagation adder to generate the final output.

Figure 5: Modified Wallace Reduction Block Diagram



MODIFIED WALLACE FLOW CHART

Figure 6: Modified Wallace Flow Chart

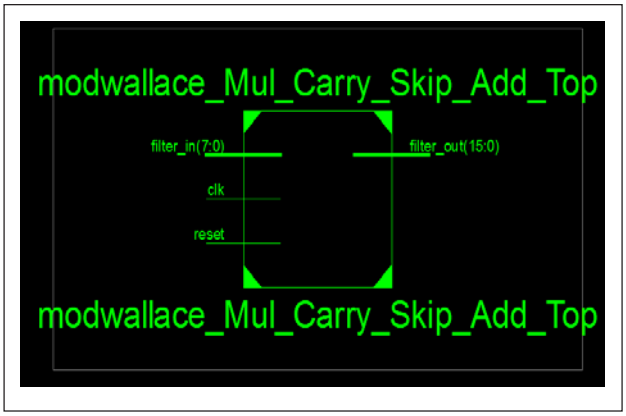


RESULTS

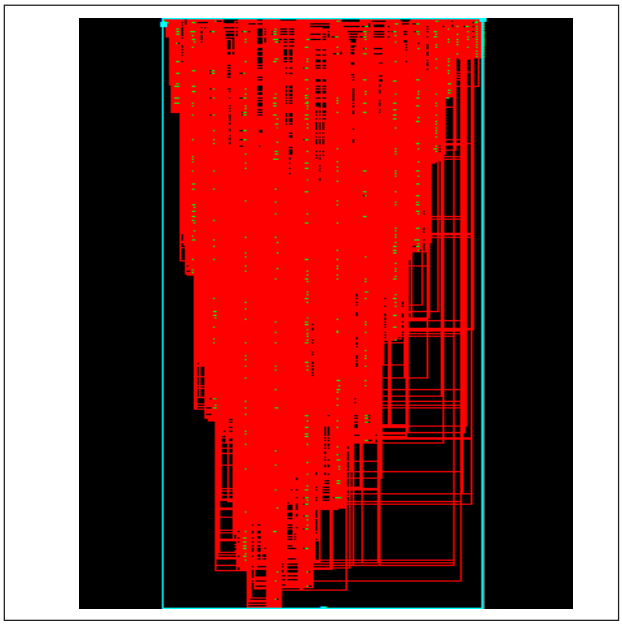
In our paper we make use of modified Wallace multiplier and carry skip adder to design FIR filter. The inputs to the design are 8 bit width filter_in, clk, and also reset. The reset which we are using is synchronous with clock and at the first most clock cycle we make reset as “0” in order to initialize our design, so that there is no possibility

of getting unknown values. The coefficients value i.e coeff0 will be multiplied to filter_in and added to the value obtained on multiplying coeff1 with filter in.

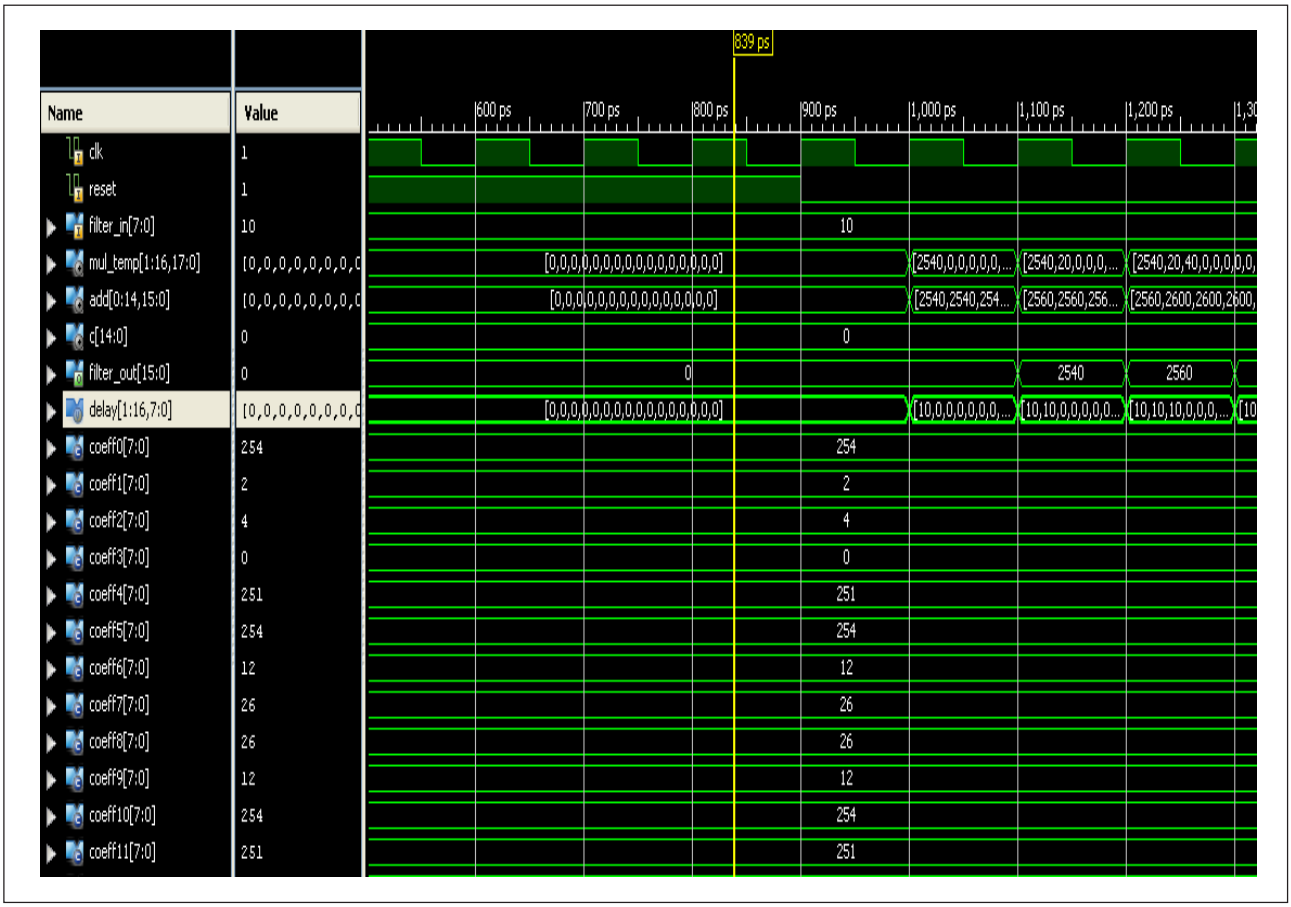
RTL SCHEMATIC



TECHNOLOGY SCHEMATIC



WAVEFORM



CONCLUSION

Digital filters are becoming ubiquitous in audio applications. As a result, good digital filter performance is important to audio system design. Digital filters differ from conventional analog filters by their use of finite precision to represent signals and coefficients and finite precision arithmetic to compute the filter response. In this project, FIR filter is implemented in Xilinx 12.3i using Verilog language. Verilog coding for the FIR filter was done. Even the processing speed increases, there are little variations in terms of time and power consumed. The power consumption is high when the n-input bits increases. Overcoming these variations can further enhance the system performance.

REFERENCES

1. Aksoy L., Lazzari C, Costa E, Flores P, and Monteiro J (2013), "Design of digit-serial FIR filters: Algorithms, architectures, and a CAD tool," *IEEE Trans. Very Large Scale Integration (VLSI) Syst*., Vol. 21, No. 3, pp. 498-511.
2. Bull D R and Horrocks D H (1991), "Primitive operator digital filter," *IEE Proceedings-G,* Vol. 138, No. 3, pp. 401-412.
3. Dempster A G and Macleod M D (1995), "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analod Digit. Signal Process.*, Vol. 42, No. 9, pp. 569-577.
4. Gately M B, Yeary M B, and Tang C Y (2012), "Multiple real-constant multiplication with improved cost model and greedy and optimal searches," *in Proc. IEEE ISCAS*, May, pp. 588–591.
5. Huang R, Chang C H H, Faust M, Lotze N, and Manoli Y (2011), "Signextension avoidance and word-length optimization by positive-offset representation for FIR filter design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, Vol. 58, No. 12, pp. 916–920.
6. Hwang S, Han G, Kang J, and Kim J S (2004), "New distributed arithmetic algorithm for low-power FIR filter implementation," *IEEE Signal Process. Lett.*, Vol. 11, No. 5, pp. 463–466.
7. Ko H J and Hsiao S F (2011), "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.
8. Kumm M, Zipf P, Faust M and Chang C H (2012), "Pipelined adder graph optimization for high speed multiple constant multiplication," *in Proc. IEEE ISCAS*, May, pp. 49–52.
9. Mehendale S D S M and Venkatesh G (1995), "Synthesis of multiplier-less FIR filters with minimum number of additions," *in Proc. IEEE ICCAD*.
10. Meher P K and Pan Y (2011), "Mcm-based implementation of block fir filters for high-speed and low-power applications," *in Proc. VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th Int. Conf.,* Oct., pp. 118–121.
11. Meher P K (2010), "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, pp. 592–603.

12. Meher P K, Candrasekaran S, and Amira A (2008), "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," *IEEE Trans. Signal Process.*, Vol. 56, No. 7, pp. 3009–3017.

13. Park I C and Kang H J (2001), "Digital filter synthesis based on minimal signed digit representation," *in Proc. Design Autom. Conf*. (DAC).

14. Voronenko Y and Püschel M (2007), "Multiplierless multiple constant multiplication", *ACM Trans. Algorithms,* Vol. 3, No. 2.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com