

International Conference on Emerging Trends in Engineering, Science and Technology  
(ICETEST - 2015)

## Area Efficient Low Power Modified Booth Multiplier for FIR Filter

Greeshma Haridas<sup>a,\*</sup>, Dr. David Solomon George<sup>b</sup>

<sup>a</sup>PG Scholar, Department of ECE, Government Engineering College Idukki, Kerala - 685606, India

<sup>b</sup>Associate Professor, Department of ECE, Government Engineering College Idukki, Kerala - 685606, India

---

### Abstract

Finite impulse response (FIR) filters are extensively used in various digital signal processing applications such as digital audio, image processing, data transmission, biomedical etc. In some applications, the FIR filter circuit must be capable to operate at high sample rates, while in other applications, the FIR filter circuit must be a low power circuit operating at moderate sample rates. FIR filters design implementation consist a large number of multiplications, which leads to excessive area and power consumption. The topology of the multiplier circuit affects the resultant area and power consumption. In this paper a new area efficient low power FIR filter design is proposed using a spanning tree based modified Booth multiplier realized in direct form. An area efficient modified spanning tree adder is also proposed, which enhances the area efficiency of FIR filter. The design is implemented using Xilinx 14.2 ISE tools, Model Sim, programming in VHDL. For implementation of the FIR filter, MATLAB Simulink tool is employed to determine various filter coefficients.

© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the organizing committee of ICETEST – 2015

**Keywords:** FIR filter; Modified booth multiplier; Spanning tree adder

---

### 1. Introduction

Digital filters are very important part of Digital Signal Processing(DSP). Infact their extraordinary performance is one of the key reasons that DSP has become so popular. Multiplication is the most basic function used in digital filters. With advances in technology, various techniques have been proposed to design multipliers, which offer high speed, low power consumption and lesser area. Thus making them suitable for various high speed, low power compact VLSI implementations. However, these three parameters i.e. power, area and speed are always traded off.

Multiplication is the most basic functions used in digital signal processing. It requires more hardware resources and processing time than addition and subtraction. In fact, 8.72% of all instructions in a typical processing unit is multiplier[1]. Since multiplication dominates execution time, there is a need for high speed multiplier. In the past many novel ideas for multipliers have been proposed to achieve high performance [2-7]. As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of

---

\* Corresponding author. Tel.: +0-949-693-3622 ; fax: +0-000-000-0000.

E-mail address: [greeshmaharidas@rocketmail.com](mailto:greeshmaharidas@rocketmail.com)

energy. A quicker style with lower power consumption and smaller space is implicit to the trendy electronic styles. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. Therefore a compromise has got to be wiped out speed of the circuit for a larger improvement in reduction of space and power.

This paper presents an FIR filter based on spanning tree based modified booth multiplier. Multiplier based on modified booth algorithm and wallace addition is one of the fast and low power multiplier. Area of modified booth wallace multiplier can further be improved by using a modified spanning tree adder for final addition.

## 2. FIR Filter Architecture - Overview

The most straightforward way to implement a digital filter is by convolving the input signal with the digital filters impulse response. In general, FIR filtering is described by a simple convolution operation as expressed in the equation (1).

$$y[n] = \sum_{k=0}^{N-1} h[k] * x[n-k] \quad (1)$$

where  $x[n]$ ,  $y[n]$ , and  $h[n]$  represent data input, filtering output, and a coefficient, respectively and  $N$  is the filter order [8]. In FIR filter, the multiplier is the major constraint which defines the performance of the desired filter. Usually modified booth multipliers are used for good performance.

### 2.1. Modified booth multipliers

The Booth multiplier makes use of Booth encoding algorithm in order to reduce the number of partial products by processing three bits at a time during recoding [9]. This recoding method is widely used to generate the partial products for implementation of large parallel multipliers, which adopts the parallel encoding scheme.

The modified booth algorithm is shown in Table 1.

Table 1. Modified booth algorithm

$Y_{2i+1}$	$Y_i$	$Y_{2i-1}$	Recoded Digit	Operand Multiplication
0	0	0	0	0*Multiplcand
0	0	1	+1	+1*Multiplcand
0	1	0	+1	+1*Multiplcand
0	1	1	+2	+2*Multiplcand
1	0	0	-2	-2*Multiplcand
1	0	1	-2	-2*Multiplcand
1	1	0	-1	-1*Multiplcand
1	1	1	0	0*Multiplcand

#### 2.1.1. Wallace tree multiplier

Trees are an extremely fast structure for summing partial-products. Tree structures require only order  $\log N$  stages to reduce  $N$  partial products by performing parallel additions. The tree multiplication algorithm can reduce the number of partial products by employing multiple input compressors capable of accumulating several partial products concurrently [1]. This is achieved by minimizing the number of partial product bits in a fast and efficient way by means of a Carry Save Adder (CSA) tree constructed from 1-bit full adders [10].

## 3. Proposed FIR Filter

Nowadays, many finite impulse response (FIR) filter designs aim at either low area or high speed or reduced power consumption are developed. It can be observed that, with the increase in area, hardware cost of these FIR filters

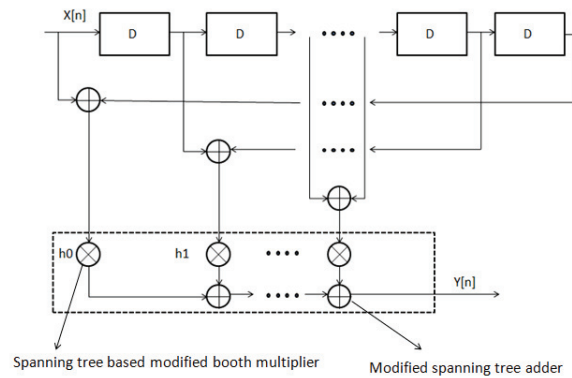


Fig. 1. Architecture of proposed FIR filter

are increasing. This observation leads to design a low area cost FIR filter with the advantages of reduced power consumption and moderate speed performance. To reduce the hardware cost, the hardware area should be optimized.

### 3.1. Structure of proposed filter design

In the direct form FIR filter in Fig. 1, the spanning tree based modified booth multiplier module performs the concurrent multiplications of individual delayed signals and respective filter coefficients, followed by accumulation of all the products. Accumulation is also done using a modified spanning tree adder. To reduce the area of the multiplier an area efficient adder, namely the spanning tree parallel prefix adder, is used here for final addition. A modified spanning tree adder is developed here such that it enhances the performance further.

### 3.2. Spanning tree adder

Propagate and generate signals are pre-computed in Parallel-prefix adders. Then these signals are combined using the fundamental carry operator (fco) [11].

$$(g_L, p_L) o (g_R, p_R) = (g_L + p_L * g_R, p_L * p_R) \quad (2)$$

Fundamental carry operator (fco) can be combined in different ways to form various adder structures due to the associative property of the fco. For, example the four-bit carry-lookahead generator is given by:

$$c_4 = (g_4, p_4) o [(g_3, p_3) o [(g_2, g_2) o (g_1, p_1)]] \quad (3)$$

A more efficient tree structure can be made by rearranging the order of operations which allows parallel operation. This is shown below:

$$c_4 = [(g_4, p_4) o (g_3, p_3)] o [(g_2, g_2) o (g_1, p_1)] \quad (4)$$

The arrangement of the prefix network gives rise to various families of adders. This paper focuses on Spanning Tree Adder. An operator known as black cell is used here. Black cell generates the ordered pair in equation (2) [12].

Spanning Tree Adders are hybrid form of a tree structure with a ripple-carry design which is shown in Fig. 2. This design terminates with a 4-bit ripple carry adder.

### 3.3. Proposed modified spanning tree adder

The architecture of modified spanning tree adder is shown in Fig. 3a. It comprises of 4-bit ripple carry adder and carry predictor logic. The carry predictor shown in Fig. 3b, predicts the carry input to the 4-bit ripple carry adder before the actual carry-out comes from the proceeding 4-bit ripple carry adder. The delay of carry out from a ripple carry adder is 8 stages, whereas the carry predictor logic can predict the carry-in to the second 4-bit ripple carry adder

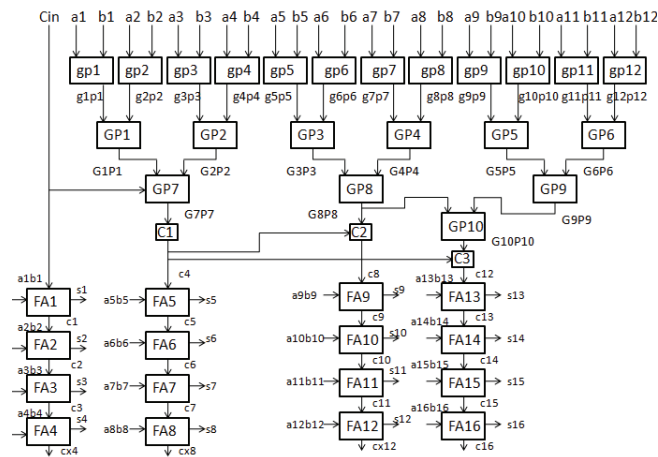


Fig. 2. 16-bit Spanning Tree Adder

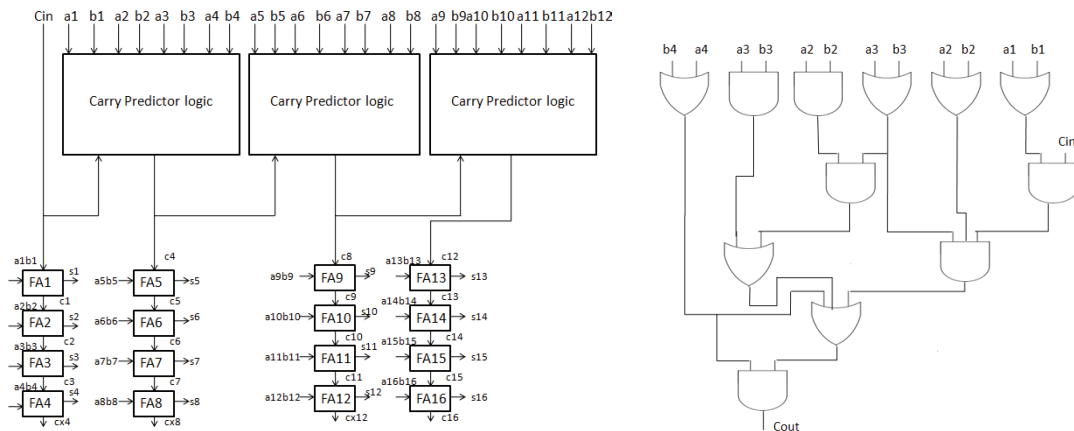


Fig. 3. a) Modified spanning tree adder, b) Proposed carry predictor logic

in 5 stage delay and 4 stage delay for the rest ripple carry adders. Thus 4-bit ripple carry adders can operate almost parallel and thereby enhances the speed performance.

In comparison with the general spanning tree adder structure, the carry predictor logic consumes only 12 gates, whereas in spanning tree adder each black cell consumes 3 gates, resulting 21 gates. So modified spanning tree adder is more area efficient than spanning tree parallel prefix adder.

### 3.4. Spanning tree based modified booth multiplier

Multiplier architecture consists of Booth encoder, Partial product generator, Wallace tree and a modified spanning tree parallel prefix adder. The first block is the modified booth algorithm which encodes the multiplier bits and the partial product generator produces the partial products by operating on multiplicand. This will reduce the number of partial products. Partial products so formed are added using Wallace tree. Block diagram of Booth Encoded Wallace Tree multiplier is shown in Fig. 4. The outputs of Wallace tree i.e. final sum and carry are added using modified spanning tree parallel prefix adder and adder give the final product.

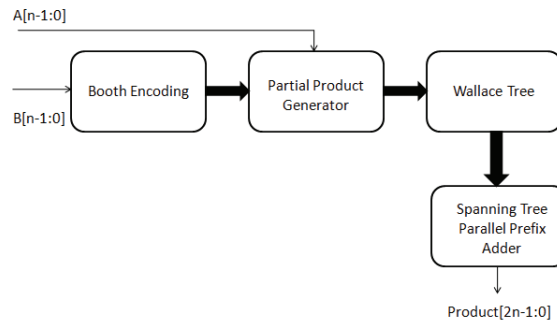


Fig. 4. Architecture of proposed modified booth multiplier

#### 4. RESULTS AND DISCUSSION

The symmetric form structure of proposed FIR filter is implemented. FIR filter coefficients design has been performed by using MATLAB. VHDL has been used to enter hardware description. The functionality of the designs were verified via simulation with ModelSim. The Xilinx ISE 14.2 software was used to synthesize the designs onto the FPGA Spartan 3E device. To test the correctness of the design the observed output from MATLAB co-simulation with modelsim implementation is compared with the simulated output results that confirms the effectiveness of the design. The FPGA device is configured with the .bit file using the JTAG boundary scan method. After the Spartan device is configured for the spanning tree based modified booth multiplier design, its working is verified by applying different inputs.

Table 2. Comparison of FIR filters

FIR filter topology	Existing [13]	Filter based on spanning tree adder	Filter based on modified spanning tree adder
No. of 4 i/p LUT	969	836	770
Occupied Slices	572	473	443
Slice flip-flop	215	32	32
IOB	57	57	57
Delay(min.period)(ns)	12.736	1.933	1.933
Power(W)	0.033	0.032	0.032

A detailed comparative study is done in order to analyze how much the proposed FIR filter using spanning tree based modified booth multiplier is better than the existing FIR filter design. The comparison is done in terms of area, delay, power consumption. The simulated FIR filter delay and area are obtained from the Xilinx ISE synthesis reports and are shown in Table 2. Power is measured using power estimator tool provided by Xilinx.

The area consumption of the FIR filters is measured in terms of 4 input Look-Up Tables (LUT) units, slice flipflops, occupied slices and IOB's which represent configurable logic units within the FPGA Spartan 3E. Conventional FIR filter utilizes 969 LUT's, whereas FIR filter using spanning tree based modified booth multiplier and modified spanning tree based booth multiplier utilizes 836 and 770 4 input LUT's respectively. The area comparison is clearly depicted in Fig. 5. From the graph it is evident that the area consumed by FIR filter using spanning tree based modified booth multiplier is least. While comparing filters using spanning tree adder and modified spanning tree adder, filter using modified spanning tree adder is consuming lesser area, since the proposed carry predictor logic has reduction in number of gates.

The power consumption is represented in Watts. Spanning tree based FIR filter consumes less power than that of existing FIR filter. The reduction in number of gates accounted for the reduction in power in spanning tree based FIR filter. Speed comparison is done using the timing report obtained in the synthesis report. According to timing report the minimum period for the spanning tree based FIR filter is lesser than that of existing FIR filter. Power and delay comparison are shown in Fig. 6. Both power and delay of spanning tree and modified spanning tree based FIR filter



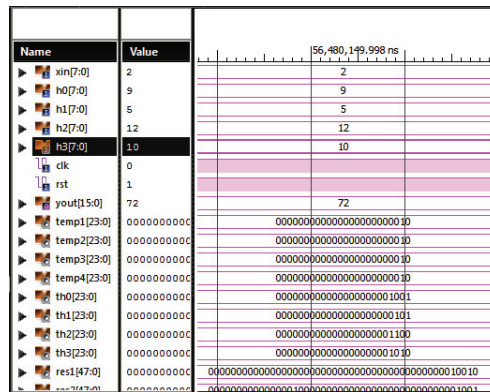


Fig. 8. Output Waveform of FIR filter based on modified spanning tree adder

## 5. Conclusion

In the design of Integrated Circuits, area occupancy plays a vital role because of increasing the necessity of portable systems. A highly area efficient FIR filter using spanning tree based modified Booth multiplier is designed based on the symmetric form realization. By employing spanning tree adder and modified spanning tree adder in FIR filter, area is reduced by 23.29% and 29.10% respectively when compared to conventional FIR filter. And the power is reduced by 3.03%.

## References

- [1] Rabey, Nikolic, and Chandrasekhran, "Digital Integrated Circuits: A Design Perspective" 2nd Edition, Prentice Hall, pp. 586-594, 2003.
- [2] A. A. DelBarrio, S. O. Memik, M.C. Molina, J. M. Mendias, and R. Hermida, "A distributed controller for managing speculative functional units in high level synthesis", *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 3, pp. 350363, March 2011.
- [3] N. Petra, D. D. Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Design of fixed-width multipliers with linear compensation function", *IEEE Trans. Circuits Syst.*, vol. 58, no. 5, pp. 947960, May 2011.
- [4] Jiun-Ping Wang, Shiann-Rong Kuang, and Shish-Chang Liang, "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications", *IEEE Trans. Circuits Syst.*, vol. 19, no. 1, pp. 52-60, January 2011.
- [5] Yuan-Ho Chen, T.-Y. Chang, and C.-Y. Li, "Area-Effective and Power-Efficient Fixed-Width Booth Multipliers Using Generalized Probabilistic Estimation Bias", *IEEE Trans. Circuits Syst.*, vol. 1, no. 3, pp. 277-287, September 2011.
- [6] Yuan-Ho Chen and T.-Y. Chang, "A High-Accuracy Adaptive Conditional-Probability Estimator for Fixed-Width Booth Multipliers", *IEEE Trans. Circuits Syst.*, vol. 59, no. 3, pp. 594-603, March 2012.
- [7] Shin-Kai Chen, Chih-Wei Liu, "Design and Implementation of High-Speed and Energy-Efficient Variable-Latency Speculating Booth Multiplier (VLSBM)", *IEEE Trans. Circuits Syst. I*, vol. 60, no. 10, pp. 26312643, October 2013.
- [8] 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 Trans. Circuits Syst. II*, Express Briefs, 2013.
- [9] O. L. MacSorley, "High speed arithmetic in binary computers", *Proc. IRE*, vol. 49, pp. 67-91, 1961.
- [10] Parhami, Behrooz, "Computer Arithmetic: Algorithms and Hardware Designs", Oxford University Press 2000.
- [11] David H. K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli, "Design and Characterization of Parallel Prefix Adders using FPGAs", *Proc. IEEE*, pp. 168172, 2011.
- [12] Srinivasasamanoj R. ,M. Sri Hari and B. Ratna Raju, "High speed VLSI implementation of 256-bit Parallel Prefix Adders", *International Journal of Wireless Communications and Networking Technologies*, vol. 1, no. 1, 2012.
- [13] Shelja Jose, Shereena Mytheen, "Modified Booth Multiplier Based Low-Cost FIR Filter Design", *International Journal of Engineering Science and Innovative Technology*, vol. 3, September, 2014.