Performance of FIR Filter with Wallace Multiplier over FIR filter with Truncated Multiplier

M.Gnanasekaran[1], Dr. M. Manikandan[2]
Research Scholar[1], Associate Professor[2]
St.Peter's University[1], Anna University[2], TN, India

Abstract – Performance analysis of finite impulse response (FIR) designs are presented by the concept of modified Wallace multipliers .This paper aims at reducing the leakage current , delays and power consumption of Wallace multiplier .This is accomplished by MCSA. An efficient verilog HDL has been written, successfully simulated and synthesized in Xilinx and the results shows that proposed design achieves the best delay and power than existing system which uses the concept of truncated multiplier.

Keywords: Faithful Rounding, Finite Impulse Response (FIR) filters, Truncated Multipliers, VLSI.

I. INTRODUCTION

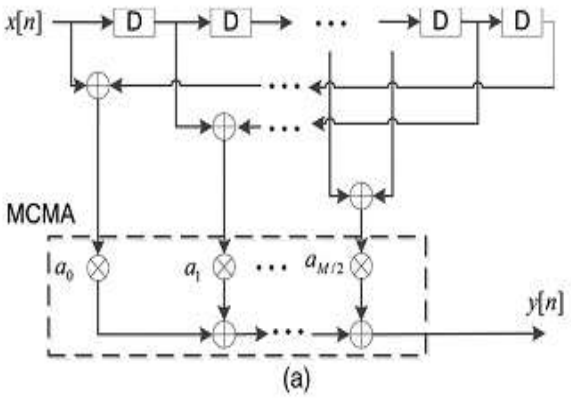
Finite impulse response (FIR) digital filters are one of the most widely used fundamental devices performed in DSP systems, ranging from wireless communications to video and image processing. This paper proposes new parallel FIR filter structures, which are beneficial to symmetric coefficients in terms of the hardware cost, under the condition that the number of taps is a multiple of 2 or 3. The 5 parallel FIR structures gives low area and high speed compare to existing 4 parallel architecture , exploit the inherent nature of symmetric coefficients reducing half the number of multipliers in sub filter section at the expense of additional adders in preprocessing and post processing blocks as in [9]. Exchanging multipliers with adders is advantageous because adders weigh less than multipliers in terms of silicon area. Due to reduction of multipliers, will get low area and high speed fir filter.

The explosive growth of multimedia application, the demand for high-performance and low-power digital signal processing (DSP) is getting higher and higher. The FIR digital filter in [9] is one of the most widely used fundamental devices performed in DSP systems, ranging from wireless communications to video and image processing. Some applications need the FIR filter to operate at high frequencies such as video processing,

ICGPC 2014
St.Peter's University, TN, India.

whereas some other applications request high throughput with a low-power circuit such as multiple-input– multiple-output systems used in cellular wireless communication. Furthermore in[14], when narrow transition band characteristics are required, the much higher order in the FIR filter is unavoidable. Due to its linear increase in the hardware implementation cost brought by the increase in the block size L, the parallel processing technique loses its advantage to be employed in [12]. In a truncated multiplier, several of the least significant columns of bits in the partial product matrix are not formed.

This reduces the area and power consumption of the multiplier. It also reduces the delay of the multiplier in many cases, because the modified carry save adder producing the product can be shorter for the Modified Wallace reduction method, once the partial product array (bits) is formed, adjacent rows are collected into non-overlapping groups of three. Each group of three rows is reduced by Applying a full adder to each column that contains three bits, Applying a half adder to each column that contains two bits, and Passing any single bit columns to the next stage without processing. There are two basic fir structures, direct form and Transposed form as in [8].



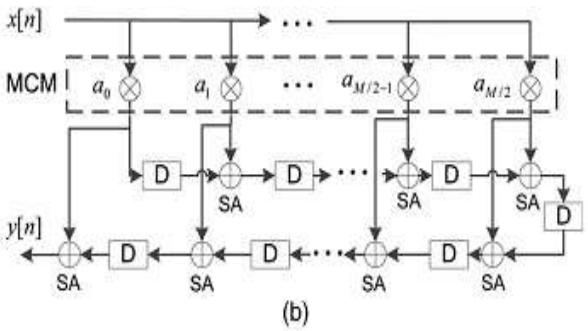


Fig.1. Structures of Linear-Phase Even-Order FIR Filters: (a) Direct Form and (b) Transposed Form.

II. FIR FILTER USING TRUNCATED MULTIPLIER

Truncated multiplier is a good candidate for digital signal processing (DSP) applications including finite impulse response (FIR) and discrete cosine transform (DCT). Through truncated multiplier a significant reduction in Field Programmable Gate Array (FPGA) resources can be achieved as in [6]. Multiplication is frequently required in digital signal processing. Parallel multipliers in [13] provide a high-speed method for multiplication, but require large area for VLSI implementations. In most signal processing applications, a rounded product is desired to avoid growth in word-size. Thus an important design goal is to reduce the area requirements of rounded output multipliers in [14]. The authors present a technique for parallel multiplication which computes the product of two numbers by summing only the most significant columns of the multiplication matrix, along with a correction constant. A method for selecting the value of the correction constant which minimizes the average and mean square error is introduced. Equations are given for estimating the average, mean square, and maximum error of the rounded product. With this technique, the hardware requirements of the multiplier can be reduced by 25 to 35%, while limiting the maximum error of the rounded product to less than one unit in the last place in [14].

A faithfully rounded truncated multiplier design is presented where the maximum absolute error is guaranteed to be no more than 1 unit of least position. The proposed method jointly considers the deletion, reduction, truncation, and rounding of partial product bits in order to minimize the number of full adders and half adders during tree reduction. Experimental results demonstrate the efficiency of the proposed faithfully truncated multiplier with area saving rates of more than 30%. In addition, the truncated multiplier design also has smaller delay due to the smaller bit width in the final carry-propagate adder [4]. The standard architecture of 6×6-bit parallel multiplier, where HA and FA are the half and full adders respectively. It can be expressed by the sum of two segments: the most-significant part MP and the least-significant part LP. The standard 6×6-bit parallel multiplier can also be divided into three subsets: the most-significant part MP, input correction IC and the least-significant part LP. The fixed width multiplier can be obtained directly by removing the LP region and introducing the IC region to obtain MP' region, which is truncated multiplier in [16].

Multipliers are present in almost all Digital Signal Processing systems. They are area and power demanding structures that constrain the timing and resolution parameters of the entire DSP unit, making the implementation of parallel multipliers desirable to achieve low-power arithmetic systems. Truncated multiplication reduces the power required by multipliers by only computing the most-significant cant bits of the product However, these results in fixed systems optimized for a given application at design time. A novel approach to truncation is proposed, where a full precision multiplier is implemented, but the active section of the partial product matrix is selected dynamically at run-time. This allows a power reduction tradeoff against signal degradation which can be modified at any time. Such architecture brings together the power reduction benefits from truncated multipliers and the flexibility of reconfigurable and general purpose devices [1].

Effective implementation of such a multiplier is presented in a custom dig- ital signal processor, where the concepts of software compensation and multi-level truncation are introduced in [16] and analyzed for different applications. Experimental results are studied, including power measurements from both post-synthesis simulations and a fabricated IC implementation, featuring the rst system-level DSP core using a one-grain programmable truncated multiplier. Fault-tolerant techniques are also studied from an energy efficiency point of view. The application of such techniques to the proposed DSP architecture shows that, not only energy reductions from both truncated multiplication and fault tolerance can coexist, but the existence of synergies between both techniques to obtain lower energy consumption for DSP architecture [2].

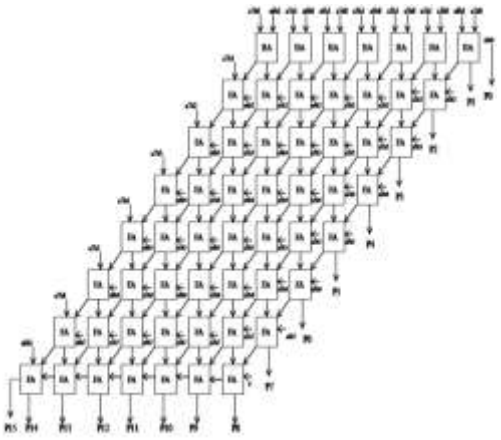


Fig.2. The architecture of a standard 8×8-bit parallel multiplier.

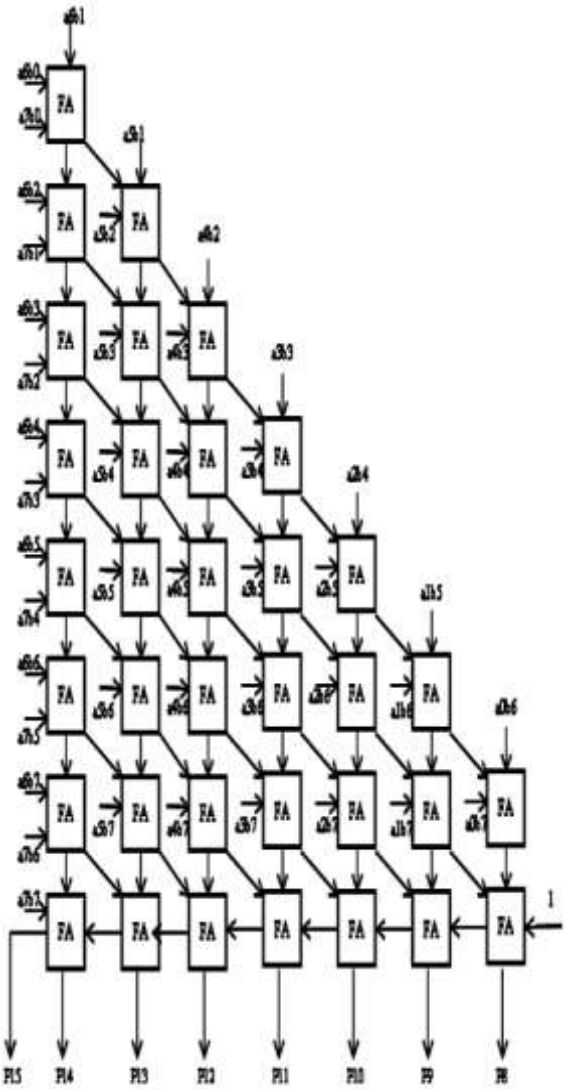


Fig.3. The architecture of a truncated 8×8-bit parallel multiplier.

Truncated multipliers offer significant improvements in area, delay, and power. The proposed method finally reduces the number of full adders and half adders during the tree reduction. While using this proposed method experimentally, area can be saved [5]. The output is in the form of LSB and MSB. Finally the LSB part is compressed by using operations such as deletion, reduction, truncation, rounding and final addition. In previous related papers, to reduce the truncation error by adding error compensation circuits. In this project truncation error is not more than 1 ulp (unit of least position). So there is no need of error compensation circuits, and the final output will be precise [3].

II. PROPOSED DESIGN OF WALLACE MULTIPLIERS

In proposed multiplier, each block addition operation in the column is performed using combination of half adders, full adders, 4:2 compressors and 5:2 compressors instead of full adders and they are selected appropriately based on the number of bits to be added. The first level (Stage A) computation is represented in the fig 4. Thus the generated partial sums are correctly divided and added again in the same manner which forms the second level (stage B) of computation. In the third level (stage C) the final product will arrive only after generated partial sums in the second level are added. The speed has been improved when compared with the truncated design without significantly increasing the amount of power in proposed Wallace Multiplier [7].

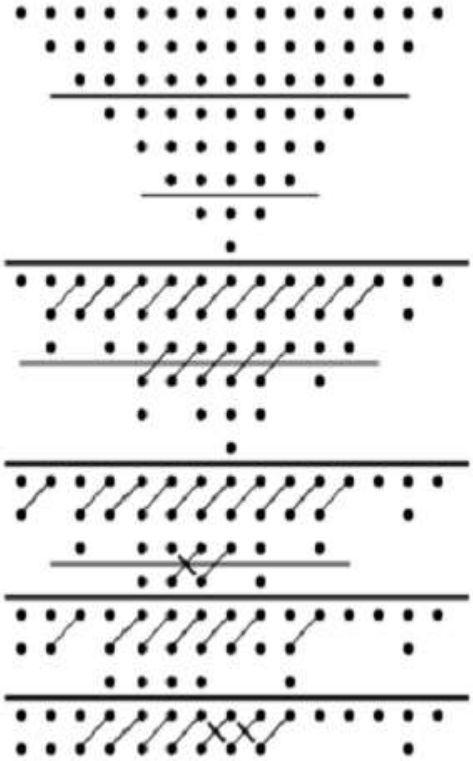


Fig.4. Proposed Design for modified Wallace multiplier 8x8

In Fig.4 shows the proposed modified Wallace tree multiplier block diagram. The proposed architecture differs from the Wallace multiplier logic represented by dot. 8x8 bit multiplier can be designed similarly by the same method. The Waveform obtained is correctly checked by giving different input bits combinations. The power & delay for both the truncated & proposed Modified Wallace multipliers are compared.

A comparison of power & delays are used in proposed and truncated method of 8x8 bits multiplier is given in table1. It is clear from the table that a significant power is reduced in a proposed design. 42.5% power can be saved in a proposed design as compared to truncated design without significant increase in area with a low power technique in 8x8 bit multiplier [10]. This reduction method is applied to each successive stage until only two rows remain. This process is illustrated by the conventional 8-bit by 8-bit Wallace multiplier. The reduction is performed in four stages (each with the delay of one full adder). The third phase will require a (2N-1-S) wide adder, where S – number of stages in reduction. After reducing from 8 stages to 2 stages modified MCSA adder is used to reduce the delay and area. Minimized adder graph (MAG) algorithm consumes more area but multiple constant multiplication algorithms consume less area [11].

3.1Reduced Complexity Wallace multiplier

It is the modified version of Wallace multiplier. It has less half adders than the normal Wallace multiplier. The partial products are formed by N^2 AND gates. The partial products are arranged in an "inverted triangle" order. The modified Wallace reduction method divides the matrix into three row groups [15].

1) Use full adders for each group of three bits in a column like the conventional Wallace reduction.

2) A group of two bits in a column is not processed, that is, it is passed on to the next stage (in contrast to conventional Method). Single bits are passed on to the next stage as in the conventional Wallace reduction.

3) The only time half adders are used is to ensure that the number of stages does not exceed that of a conventional

2.2 Modified Carry save Adder

The 16-bit regular CSA logic diagram is shown in the Figure.5. It has 17-half adders and 15-full adders. Since the ripple carry adder (RCA) is used in the final stage, this structure yields large carry propagation delay. To reduce this delay, the final stage of CSA is divided into 5 groups as shown in Figure 6. The first group includes n $1+\log_2$ n-bit value and other groups include $\log_2$n-bit value, where n is the bit size of the adder.

The divided groups are listed below,

i). {c4,s[4:0]}

ii).{c7,x[7:5]}

iii).{c10,x[10:8]}

iv).{c13,x[13:11]}

v). x[17:14]

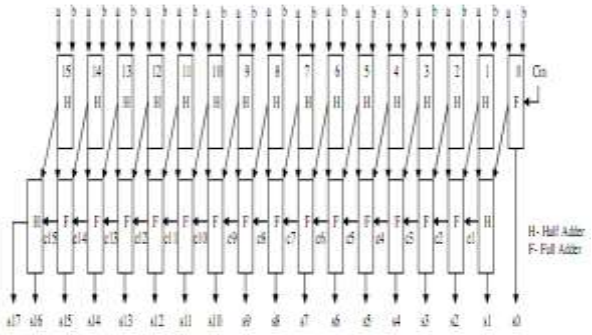


Fig.5. Block diagram of Regular CSA

The first group of output s[4:0] are directly assigned as the final output; the second group {c7,x[7:5]} manipulates the partial result by considering c4 is 0; the third group {c10,x[10:8]} manipulates the partial result by considering c7 is 0; the fourth group {c13,x[13:11]} manipulates the partial result by considering c10 is 0 and the fifth group x[17:14] manipulates the partial result by considering c13 is0. Depending on c4 of the first group, the second group mux gives the final result without the carry propagation delay from c4 to c7; depending on c7 of the second group final result, the third group mux gives the final result without the carry propagation delay from c7 to c10; depending on c10 of the third group final result, the fourth group mux gives the final result without the carry propagation delay from c10 to c13 and depending on c13 of the fourth group final result, the fifth group mux gives the final result without the carry propagation delay from c13 to s17.

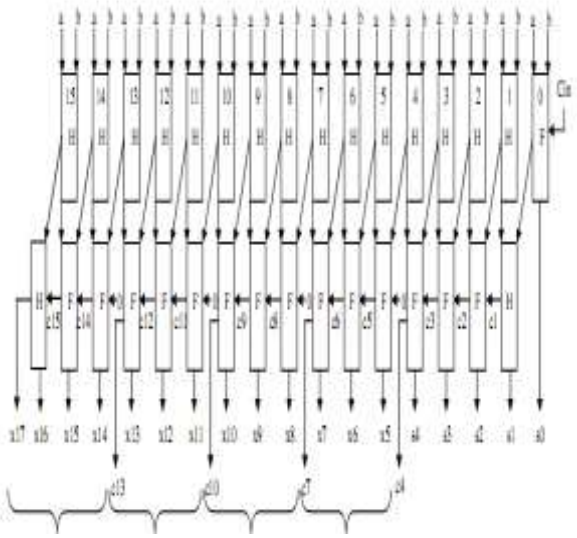


Fig.6. Modified carry save adder.

The main advantage of this logic is that each group computes the partial results in parallel and the muxes are ready to give the final result "immediately" with the minimum delay of the mux. When the Cin of each group arrives, the final result will be determined "immediately". The area indicates the total cell area of the design and the total power is sum of leakage power, internal power, net power and dynamic power. The proposed result shows that the CLA and CSA have reduced area and consume lesser power than MCSA. But the speed of the MCSA architecture has significantly improved and has the least value of power-delay product compared to the conventional CSA and CLA.

IV. SIMULATION RESULTS

In this work we are evaluating the performance of the proposed FIR filter using low power consumption using Wallace multipliers with modified carry save adders. These multipliers can be implemented using Verilog coding. In order to get the power report and delay report we are synthesizing these multipliers using Xilinx and ModelSim. Simulation result for the FIR filter using modified wallace multiplier are given in figure7.

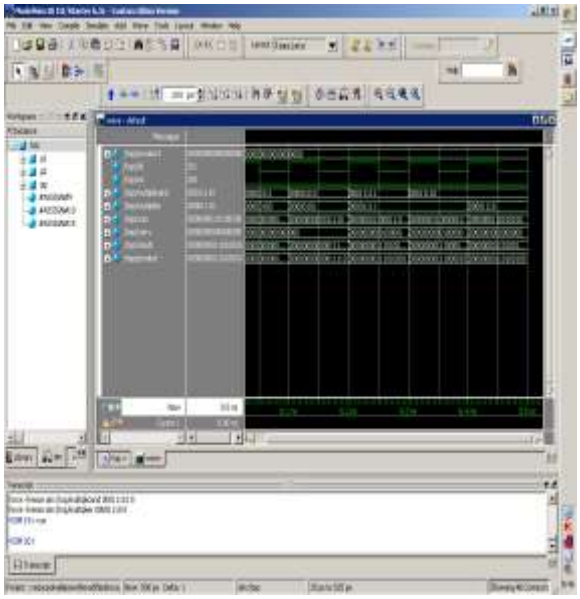


Figure.7. Simulation result of proposed FIR filter with truncated multiplier.

The comparison table 1 shows the power and delay of both the existing and proposed FIR filter structures. The delay of FIR filter gets increased for lower and reduced statically when go for higher order of FIR filter.

Table 1 Delay and Power Comparison of 8x8 Multiplier

FIR with different Multiplier	Delay(ns)	Power(mw)
FIR with truncated multiplier	27.9	3.9
FIR with Wallace Tree multiplier	25.6	2.7

V. CONCLUSION

In this paper, we have proposed design for a FIR filter using Modified Wallace Multiplier with Modified carry save adder. The FIR filter is designed to increase the speed and decrease the power taken by the multiplier unit. It has been easily concluded that proposed FIR filter design has consumed less power than the Truncated Multiplier design. A tool for generating structural Verilog models of specific instances of these filter is discussed. We have presented a formal synthesis methodology that can be automated and thus it not only ensures formally verified synthesis results but also is very easy to use for end users who do not have any background in formal semantics and reasoning. Our synthesis methodology achieves correctness by construction and thus eliminates the post synthesis verification requirements, which in turn reduces design time. At the end we calculate and compare the power delay product of various adders and multipliers and conclude that for the minimal power design of FIR filter Modified Wallace multiplier using carry save adder is minimum as compare to other adders and multipliers.

REFERENCES

[1] J. G. Chung and K. K. Parhi, "Frequency-spectrum-based low-area low-power parallel FIR filter design," EURASIP J. Appl. Signal Process. , vol. 2002, no. 9, pp. 444–453, 2002.

[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley, 1999.

[3] Z.-J. Mou and P. Duhamel, "Short-length FIR filter and their use in fast nonrecursive filtering," IEEE Trans. Signal Process. , vol. 39, no. 6, pp. 1322–1332, Jun. 1991.

[4] J. I. Acha, "Computational structures for fast implementation of L-path and L-block digital filter," IEEE Trans. Circuit Syst. , vol. 36, no. 6, pp. 805–812, Jun. 1989.

[5] C. Cheng and K. K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," IEEE Trans. Circuits Syst. I, Reg. Papers , vol. 51, no. 8, pp. 1492–1500, Aug. 2004.

[6] C. Cheng and K. K. Parhi, "Further complexity reduction of parallel FIR filter," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS 2005), Kobe, Japan, May 2005.

[7] O. Gustafsson, "Lower bounds for constant multiplication problems," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 11, pp. 974–978, Nov. 2007.

[8] Y. Voronenko and M. Puschel, "Multiplier less multiple constant multiplication," ACM Trans. Algorithms, vol. 3, no. 2, pp. 1–38, May 2007.

[9] 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.

[10] R. Huang, C.-H. H. Chang, M. Faust, N. Lotze, and Y. Manoli, "Sign extension 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, Oct. 2011.

[11] 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, pp. 592–603, Mar. 2010.

[12] 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, pp. 3009–3017, Jul. 2008.

[13] Deepshikha Bharti, K. Anusudha, "High Speed FIR Filter Based on Truncated Multiplier and Parallel Adder" International Journal of Engineering Trends and Technology (IJETT) – Volume 5 Number 5 - Nov 2013

[14] S. Karunakaran, Dr. N.Kasthuri, "Area and Power Efficient VLSI Architecture for FIR Filter using Asynchronous Multiplier," British Journal of Science 61 December 2011, Vol. 2

[15] Muhammad H. Rais, *member IEEE* "Efficient Hardware Realization of Truncated Multipliers using FPGA" International Journal of Engineering and Applied Sciences 5:2 2009

[16] 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.