

A VLSI Implementation of FIR Filter Based on Approximate Dadda Multiplier

Pusarla Murali , P . Murali Krishna

Abstract— Finite impulse response (FIR) filters are used in Digital Signal Processing applications. Accuracy in Filter Designing is based on the Multiplication and accumulation of filter coefficients. This paper describes an approach to the VLSI implementation of digital filter which is flexible and provides superior to traditional approaches, low power, and area efficient Discrete Wavelet Transform architecture. A 4-tap finite impulse response (FIR) filter implemented by 4-2 compressor based dada Multiplier. Computation intensive applications such as DSP, image processing, floating point processors and communication technologies today require efficient binary multiplication which usually is the most power and time consuming block. This paper proposes an efficient design for unsigned binary multiplication to reduce delay. A 8x8 dadda multiplier has been designed which is based on Compressing arithmetic Units. It is optimized using adaptive and recursive approach combined with square-root carry-look ahead-adder. The designs have been coded in Verilog, synthesized in Xilinx Tool and physically verified with Spartan 3 XC 3S 200 tq 144 platform.

Keywords— Dadda Multiplier, Compressors, Dadda-Multiplier, Carry look ahead Adder, Adder, Digital signal processing (DSP), finite impulse response (FIR) filter, Multiplier, Four tap Fir Filter, VLSI design.

I. INTRODUCTION

In signal processing, the filter is used to remove some unwanted component or feature from a signal thereby improving the quality of signal. It alters the amplitude and/ or phase characteristics of a signal in a desired manner with respect to frequency. The primary function of filter are – to confine a signal into a prescribed frequency band, to decompose a signal into two or more sub-bands, to modify the frequency spectrum of a signal and to model the input output relationship of a system. Filters are extensively used in signal processing and communication system in applications like noise reduction, echo cancellation, image enhancement, speech and waveform synthesis etc.

Pusarla Murali ,ECE department, Sri Krishna devaraya University College of engineering and technology, Anantapur, (Email: murali.pusarla96@gmail.com)

P Murali Krishna, Assistant Professor, ECE department, Sri Krishna devaraya University College of engineering and technology, Anantapur.(Email : murali.43@gmail.com)

There are two main kind of filter: analog and digital filter. Analog filter has analog signal at both its input & output and are made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filters are fast and simple to realize but are little stable, sensitive to temperature variations and expensive to realize in large amounts. Digital filter on the other hand uses digital processor to perform numerical calculations on sampled values of the signal and eliminate the problems associated with their classical analog counterparts, thus are preferably used in place of analog filter. Broadly, digital filters are classified as: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filter. FIR filters have linear phase, stability, fewer finite precision errors, and efficient implementation hence preferred over IIR filter. This paper discusses the design and implementation of a FIR filter using dada Multiplier with other existing Multipliers. The results from the same have been obtained and observed separately. Further, the results have compared in terms of resource utilization, frequency of operation. As a result, it has been found that the Compressed dada multiplier performs the same operation by reducing the number of partial products obtained at each stage thereby simplifying the architecture in terms of delay, complexity parameters.

II. FINITE IMPULSE RESPONSE FILTER

A finite impulse response filter is a digital filter whose impulse response is finite in nature. The output of the FIR filter depends only on the present and the past values of the input that is it is non-recursive in nature. Hence, no feedback element is required for the filter implementation. The transfer function for the filter can be easily computed by taking the z transform of the frequency response of the FIR filter.

FIR filter transfer function can be expressed as:

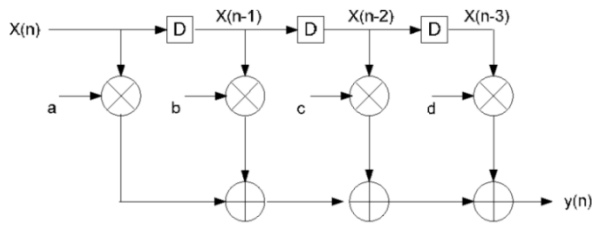
$$y[n] = b_0x[n] + b_1x[n - 1] + \dots + b_Nx[n - N] \\ = \sum_{i=0}^N b_i \cdot x[n - i],$$

$x[n]$ is the input signal,

$y[n]$ is the output signal,

N is the filter order;

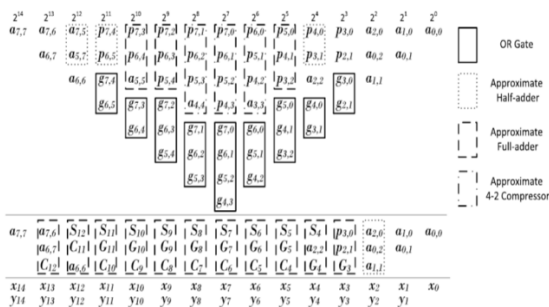
an N th-order filter has $N+1$ terms on the right-hand side.



The FIR filter consists of three basic blocks: an adder block, a multiplier and some delay elements. A D flip flop can serve the purpose of delay element. The adder performs the Proposed CSLA based BEC. The Karastuba multiplier block leads to maximum delay in the design and hence needs to be optimized.

III. DADA MULTIPLIER

The Dadda multiplier is a hardware binary multiplier design invented by computer scientist Luigi Dadda in 1965. It uses a selection of full and half adders to sum the partial products in stages (the Dadda tree or Dadda reduction) until two numbers are left. The design is similar to the Wallace multiplier, but the different reduction tree reduces the required number of gates (for all but the smallest operand sizes) and makes it slightly faster (for all operand sizes). We present four dual-quality reconfigurable approximate 4:2 compressors, which provide the ability of switching between the exact and approximate operating modes during the runtime. The compressors may be utilized in the architectures of dynamic quality configurable parallel multipliers. The basic structures of the proposed compressors consist of two parts of approximate and supplementary. In the approximate mode, only the approximate part is active whereas in the exact operating mode, the supplementary part along with some components of the approximate part is invoked.



Dadda Multiplier Using Proposed Designs

IV. APPROXIMATE ADDER DESIGN

Approximate Half Adder is performing Same Half adder Logic but The Sum XOR Operation is Performed as OR Operation. XOR gates tend to contribute to high area and delay. For approximating half-adder, XOR gate of Sum is replaced with OR gate as given in bellow equations. This results in one error in the Sum computation as seen in the truth table of approximate half-adder in Table II. A tick mark

denotes that approximate output matches with correct output and cross mark denotes mismatch.

$$\text{Sum} = x_1 + x_2$$

$$\text{Carr } y = x_1 \& x_2$$

TRUTH TABLE OF APPROXIMATE HALF ADDER

Inputs		Exact Outputs		Approximate Outputs		Absolute Difference
x1	x2	Carry	Sum	Carry	Sum	
0	0	0	0	0	0	0
0	1	0	1	0	1	0
1	0	0	1	0	1	0
1	1	1	0	1	1	1

A. APPROXIMATE FULL ADDER

In the approximation of full-adder, one of the two XOR gates is replaced with OR gate in Sum calculation. This results in error in last two cases out of eight cases. Carr y is modified and introducing one error. This provides more simplification, while maintaining the difference between original and approximate value as one. The truth table of approximate full-adder is given in Table

$$W = (x_1 + x_2)$$

$$\text{Sum} = W \oplus x_3$$

$$\text{Carr } y = W \& x_3$$

TRUTH TABLE OF APPROXIMATE FULL ADDER

Inputs			Exact Outputs		Approximate Outputs		Absolute Difference
x1	x2	x3	Carry	Sum	Carry	Sum	
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	1	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	1	1	1	0	1

B. APPROXIMATE 4-2 COMPRESSOR

In 4-2 compressor, three bits are required for the output only Sum computation, one out of three XOR gates is replaced with OR gate. Also, to make the Sum corresponding to the case where all inputs are ones as one, an additional circuit $x_1 \cdot x_2 \cdot x_3 \cdot x_4$ is added to the Sum expression. This results in error in five out of 16 cases. Carr y is simplified. The corresponding truth table is given in Table

$$W_1 = x_1 \cdot x_2$$

$$W_2 = x_3 \cdot x_4$$

$$\text{Sum} = (x_1 \oplus x_2) + (x_3 \oplus x_4) + W_1 \cdot W_2$$

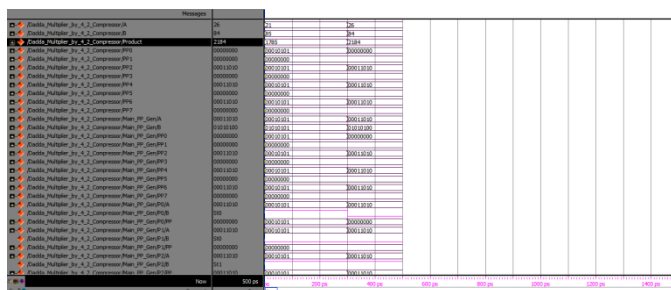
$$\text{Carr } y = W_1 + W_2$$

TRUTH TABLE OF APPROXIMATE 4-2 COMPRESSOR

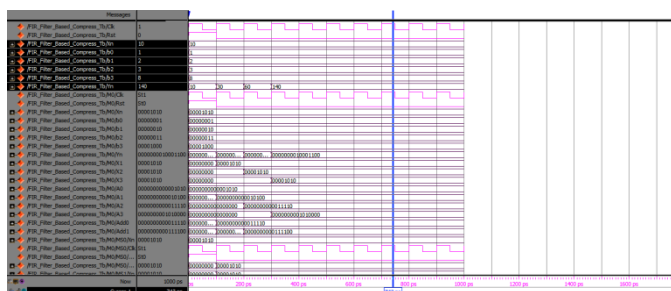
Inputs				Approximate outputs		Absolute Difference
x1	x2	x3	x4	Carry	Sum	
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	0	1	1
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	1

V. SIMULATION AND SYNTHESIS RESULTS

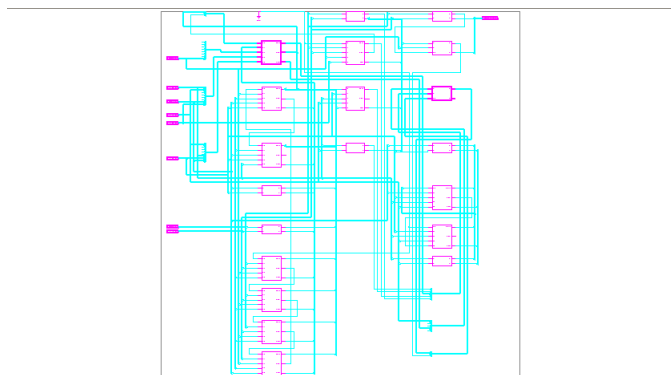
VI. COMPARISION AND ANALYSIS



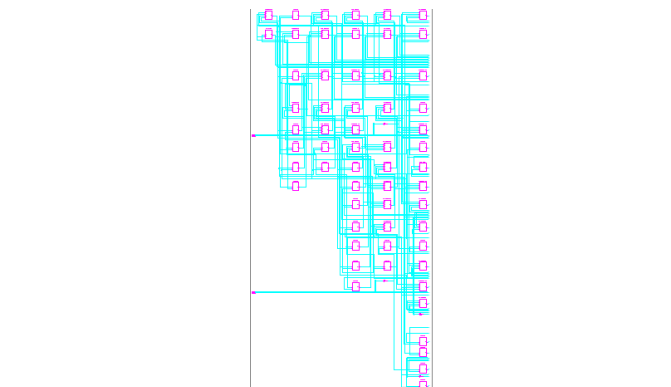
Dadda multiplier by 4-2 Compressor



FIR Filter Based on Multiplier Compressor



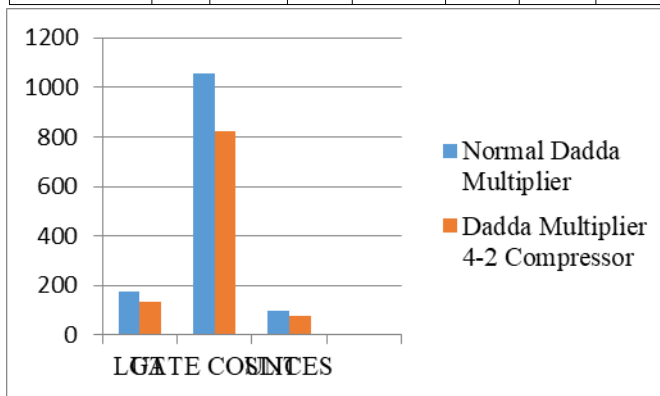
RTL View of Multiplier



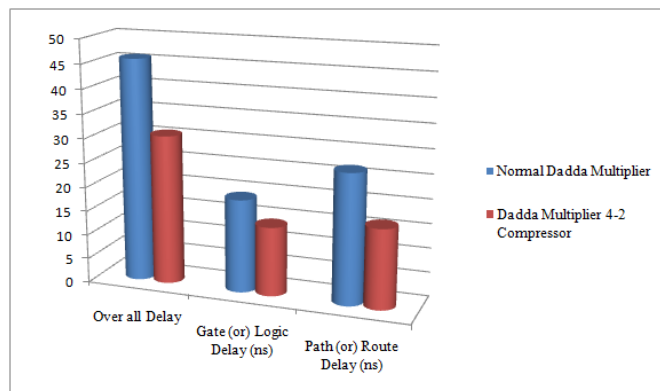
Technology View of Multiplier

The Proposed FIR Filter designs are implemented using Verilog HDL, synthesized using Xilinx for different bit sizes and the delay and area have been analyzed for comparison. From Fig. 4, it is observed that Compressor based Dadda Multiplier has minimum delay and Area. it becomes lesser compared to Normal Dadda Multiplier. The results show that the use of Compressed adders for addition achieves overall minimum delay in the proposed multiplier at the expense of higher area and delay dissipation.

METHOD NAME	AREA IN NUMBER OF LUT			MEMORY IN Kilobytes	DELAY		
	LUT	GATE COUNT	SLICES		DELAY	GATE OR LOGIC DELAY	PATH OR ROUTE DELAY
Normal Dadda Multiplier	176	1056	97	243608 kilobytes	45.844ns	19.138ns 41.7% logic	26.706ns 58.3% route
Dadda Multiplier 4-2 Compressor	137	822	77	177360 kilobytes	30.528ns	14.179ns 46.4% logic,	16.349ns 53.6% route



Area Comparison



Delay Comparison

VII. CONCLUSION

Conventionally the FIR filters which have huge application in Digital Signal Processing were developed using traditional DSP algorithms. With the advancement in the technology, the FIR filters are being developed using VLSI technology. This leads to the extensive decrease in the area occupied on chip and power consumed by the filter. The FIR filter consists of three blocks: the multiplier, adder and the delay block. Out of all three, the multiplier is the slowest of all. The research work presented in this paper has achieved adequate results and has demonstrated the efficiency of high level optimization techniques. In this work, the FIR filter has been designed using Dadda Multiplier and CLA Adder. From this work, it is concluded that chip area of FIR filter designed using dada compressor multiplier is significantly reduced and thereby making the system faster. A 8×8 -bit multiplier has been proposed and designed to showcase the technique with the primary objective of minimizing the delay & area so that it can find application in DSP, Image Processing and computation intensive ASIPs. It is based on the Compressed dada Multiplier that generates lesser number of partial product terms. The algorithm is further optimized using adaptive concept for computation of the third product term to yield faster speed.

REFERENCE

- [1] G. Challa Ram, D. Sudha Rani, R. Balasaikesava and K. Bala Sindhuri, "Design of delay efficient modified 16 bit Wallace multiplier", 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, pp. 1-4, 20-21 May 2016.
- [2] V. Buddhe, P. Palsodkar, "Design and Verification of Dadda Algorithm Based Binary Floating-Point Multiplier", IEEE International Conference on Communication and Signal Processing, pp.1073-1077, April 2014.
- [3] Ron S. Waters and Earl E. Swartzlander, "A Reduced Complexity Wallace Multiplier Reduction", IEEE Transactions on Computers, vol. 59, Issue 8, pp. 1134 – 1137, May 20, 2010.
- [4] Damarla Paradasaradhi, M. Prashanthi and N Vivek, "Modified Wallace tree multiplier using efficient square root carry select adder", 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, pp. 1-5, 6-8 March 2014.
- [5] S Arish and R. K. Sharma, "An efficient binary multiplier design for high speed applications using Karatsuba algorithm and Urdhva-Tiryagbhyam algorithm", 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, India, pp. 192-196, 23-24 April 2015.
- [6] Dipankar Pal and Akhilesh G. Naik, "Delay estimation, chip-power analyses and comparison of single-level and multi-level recursive vedic algorithm with conventional algorithms for digital multiplier", 2018 International Conference on IC Design & Technology (ICIDT), Otranto, Italy, pp. 29-32, 4-6 June 2018.
- [7] N. Nedjah, "A Review of Modular Multiplication Methods and Respective Hardware Implementation", Informatica, vol. 30, no. 1, pp. 112-115, April 18, 2005.
- [8] Kore Sagar Dattatraya and V.S. Kanchana Bhaaskaran, "Modified Carry Select Adder using Binary Adder as a BEC-1," European Journal of Scientific Research, ISSN 1450-216X / 1450-202X Vol.103 No.1 (2013), pp. 156- 159, January 2013.
- [9] Rupashree Sahu and Asit Kumar Subudhi, "An Area Optimized Carry Select Adder", 2015 IEEE Power, Communication and Information Technology Conference (PCITC) Siksha 'O' Anusandhan University, Bhubaneswar, India, pp. 1-6, 15-17 Oct. 2015.