International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 12, December 2014

Implementation and Performance Evaluation of Parallel 8-point FFT using Vedic Multiplier

Mahesh Kumar Daryani, Kanak Kumar

Abstract— **A high speed fast fourier transform (FFT) design by using three algorithm is presented in this paper. In algorithm 1, 4-bit binary multiplier based technique are used in FFT. In this technique used 128 number of slice and 207 4-input LUT for virtex-2 device family. In algorithm 2, 4-bit adder based multiplier are used in FFT. In this technique used number of slice and 4-input LUT less compare to algorithm 1 technique. In algorithm 3, 4-bit Vedic multiplier based technique are used in FFT. In this technique used in three 4-bit ripple carry adder and four 2*2 Vedic multiplier. The main parameter of this paper is number of slice, 4-input LUTS and maximum combinational path delay were calculate.**

Index Terms—**FFT, Ripple Carry Adder, Carry Select Adder, Vedic Multiplier**

I. INTRODUCTION

Digital signal processing (DSP) is the mathematical manipulation of an information signal to modify or improve it in some way. It is characterized by the representation of discrete time, discrete frequency, or other discrete domain signals by a sequence of numbers or symbols and the processing of these signals [1].

The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals. The first step is usually to convert the signal from an analog to a digital form, by sampling and then digitizing it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of numbers. However, often, the required output signal is another analog output signal, which requires a digital-to-analog converter (DAC). Even if this process is more complex than analog processing and has a discrete value range, the application of computational power to digital signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression.

Manuscript received Nov 15, 2014.

Mahesh Kumar Daryani*, Electronic & Communication Department, Millennium Institution of Technology, Bhopal, India, mobile: 9584035225*

Kanak Kumar,Professor, in Electronic & Communication Department, Millennium Institution of Technology, Bhopal, India, 9074390323

DSP algorithms have long been run on standard computers, as well as on specialized processors called digital signal processor and on purpose-built hardware such as application-specific integrated circuit (ASICs). Today there are additional technologies used for digital signal processing

including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial apps such as motor control), and stream processors, among others [2-3]. The FFT is one of the most commonly used digital signal processing algorithm. Recently, FFT processor has been widely used in digital signal processing field applied for OFDM, MIMO-OFDM communication systems. FFT/IFFT processors are key components for an orthogonal frequency division multiplexing (OFDM) based wireless IEEE 802.16 broadband communication system; it is one of the most complex and intensive computation module of various wireless standards physical layer (ofdm-802.11a, MIMO-OFDM 802.11, 802.16,802.16e) [4].

However, the main constraints nowadays for FFT processors used in WiMAX and wireless communication systems are execution time and lower power consumption [5]. The main issue in FFT/IFFT processors is complex and large multiplication, which is the most addition and sub tractor arithmetic operation used in FFT/IFFT blocks. It is an expensive operation and consumes a large chip area and power especially when it comes to a large FFT point [6].

The speed of a processor depends on its multiplier's performance. This in turn increases the demand for high speed and low power multipliers, at the same time keeping in mind low area and moderate power consumption [6].

Over the past few decades, several new structures of multipliers have been designed and explored. Multipliers based on the booth's Algorithm [7], adder and shift [7], ROM based multiplier, and modified booth's algorithm [8] is quite popular in modern VLSI design but come along with their own set of disadvantages. In these algorithms, the multiplication process, involves several intermediate operations before arriving at the final output.

In this paper design the 4-bit Vedic multiplier using different adder and implementation 8-bit radix-2 FFT algorithm.

The paper is organized as follows: Section II discusses the FFT algorithm implementation radix-2 and complex multiplication used inside the butterfly-processing element. Section III devoted for an architectural description of the 4-bitadder used module. Section IV shows the proposed three algorithms. Section V shows the resulting implementation and finally a conclusion is given in section V.

ISSN: 2278 – 909X

II. FFT ALGORITHM

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Show the butterfly operations for radix-2 DIF FFT in figure 1. The radix-2 algorithms are the simplest FFT butterfly algorithm.

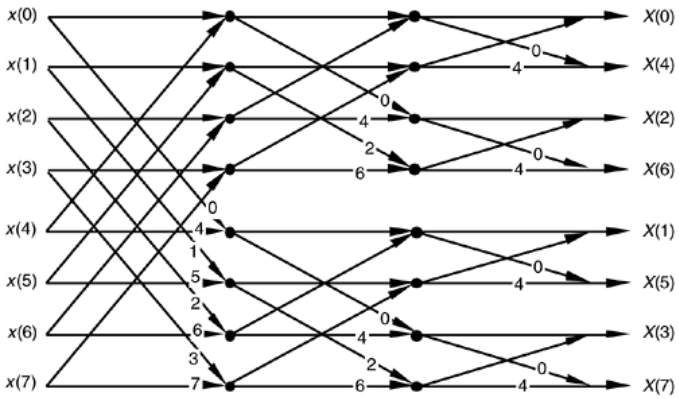

Figure 1: Radix-2 Decimation in Frequency Domain FFT Algorithm

III. 4-BIT ADDER

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logical unit, but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess 3, the most common adders operate on binary numbers. In cases where two's complement or one's complement is being used to represent negative numbers, it is trivial to modify an adder into an adder- sub tractor. Other signed number representation requires a more complex adder. The block diagram of 4-bit ripple carry adder is shown figure 2.

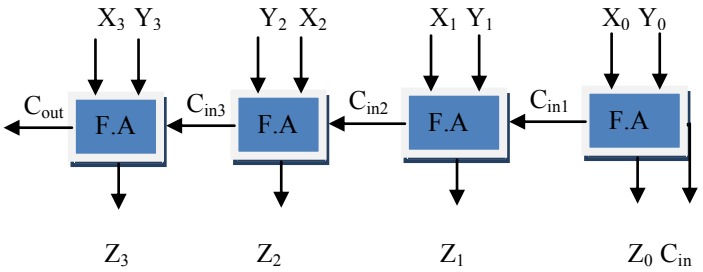

Figure 2: Block Diagram of 4-bit Ripple Carry Adder

IV. PROPOSED ARCHITECTURE

Hardware Description Language (HDL) is a language that can describe the behavior and structure of electronic system, but it is particularly suited as a language to describe the structure and the behavior of the digital electronic hardware design.

○ Algorithm 1

The FFT computation is accomplished in three stages. The x(0) until x(7) variables are denoted as the input values for FFT computation and X(0) until X(7) are denoted as the outputs.

The pipeline architecture of the 16-point FFT is shown in Figure 1 consisting of butterfly schemes in it. There are two operations to complete the computation in each stage.

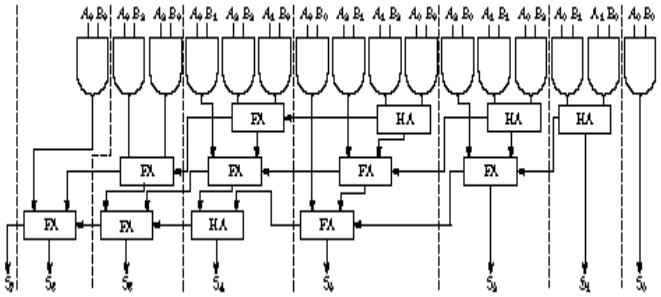

Figure 3: Block Diagram 4*4 Binary Multiplier

In algorithm 1, design radix-2 decimation in frequency domain FFT algorithm using 4*4 binary multiplier. A radix-2 FFT can be efficiently implemented using a butterfly processor which includes, besides the butterfly itself, an additional complex multiplier for the twiddle factor.

A radix-2 butterfly processor consists of a complex adder, a complex subtraction, and a complex multiplier for the twiddle factors. The complex multiplication with the twiddle factor is often implemented with four real multiplications and 2 add/ subtract operation.

○ Algorithm 2

To reduce the delay, a 4X4 multiplier is implemented using half adder, full adder and the 4-bit adder as shown in Figure 4.

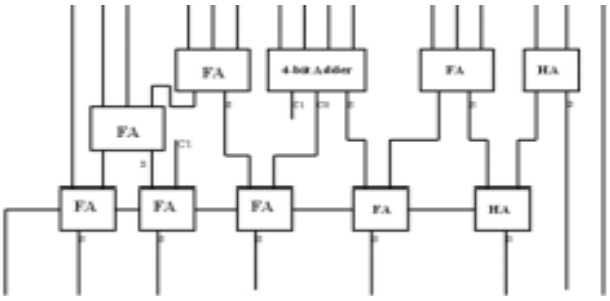

Figure 4: Block Diagram of 4-bit Multiplier using 4-bit Adder

○ Algorithm 3

In algorithm 3, Multiplication methods are extensively discussed in Vedic mathematics. Various tricks and short cuts are suggested by VM to optimize the process. The method is explained 2 bit numbers A and B where A = a1a0 and B = b1b0. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand

ISSN: 2278 – 909X

(crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit of the final product. The block diagram of 4-bit Vedic multiplier is shown Figure 5.

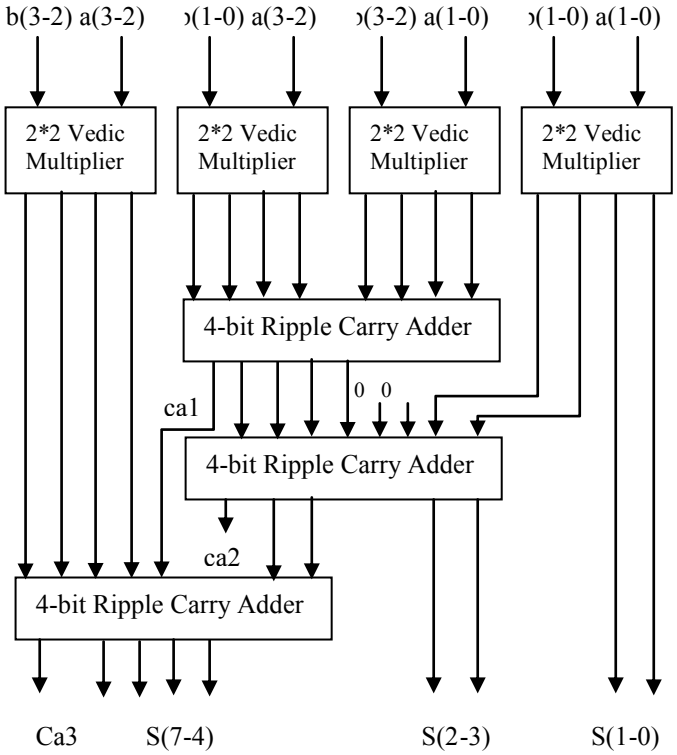

Figure 5: Block Diagram of 4-bit Vedic Multiplier

V. SIMULATION RESULT

The designs as were discussed in figure 3, 4 and 6 were implemented using VHDL and then were tested on Synopsys tools to determine the area and power constraints. In figure 46 and figure 7 have shown the resistor transistor logic (RTL) for FFT using 4-bit Vedic multiplier and output waveform of FFT using 4-bit Vedic multiplier respectively.

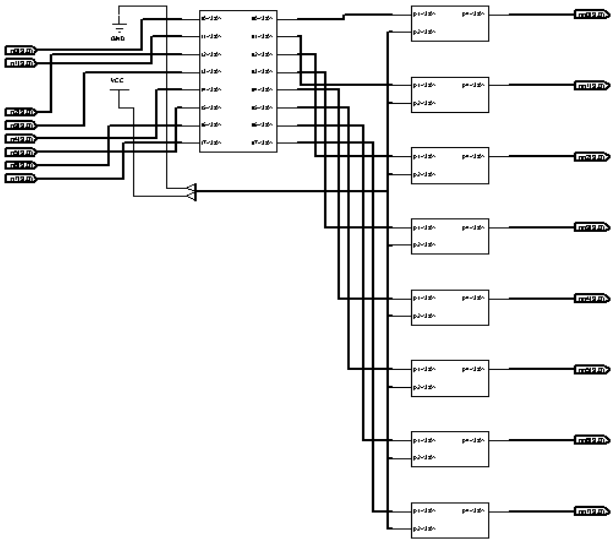

Figure 6: Resister transistor Logic for FFT using 4-bit Vedic Multiplier

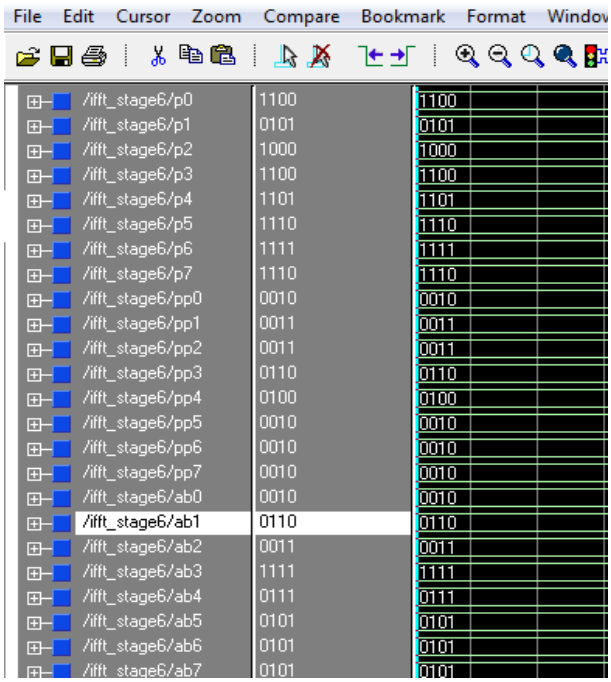

Figure 7: Test Bench Waveform for FFT using 4-bit Vedic Multiplier

Table 1: Synthesis Result for FFT Algorithm 1, Algorithm 2 and Algorithm

Architecture	Number of slice	Number of 4-input LUTs	Maximum combinational path delay
Algorithm 1	128	207	18.375 nsec
Algorithm 2	112	196	16.587 nsec
Algorithm 3	104	185	14.355 nsec

VI. CONCLUSION

A 4-bit adder, 4-bit ripple carry adder, 2*2 Vedic multiplier, 4-bit binary multiplier, 4-bit multiplier using 4-bit adder and 4-bit multiplier using ripple carry adder have been proposed in this paper and implementation in 8-point FFT. It is seen that the speed of the algorithm 3 is higher than that of normal array multiplier and multiplier using 4-bit adder. This multiplier can be used in applications such as digital signal processing, encryption and decryption algorithms in cryptography.

REFERENCES

[1] Charles. Roth Jr., "Digital Systems Design using VHDL",Thomson Brooks/Cole, 7th reprint, 2005.

[2] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, "Implementation of Vedic multiplier for Digital Signal Processing", International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Joural of Computer Applications® (IJCA), pp.1-6.

ISSN: 2278 – 909X

[3] Himanshu Thapaliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad, India.
[4] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", Delhi(2011).
[5] Harpreet Singh Dhillon and Abhijit Mitra, " A Reduced-bit Multiplication Algorithm for Digital Arithmetic", International Journal of Computational and Mathematical Sciences, Febrauary 2008, pp.64-69.
Sumit Vaidya and Depak Dandekar. "Delay-power perfor-mance comparison of multipliers in VLSI circuit design". International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.
[6] H. Thapliyal and H.R Arbania. "A Time-Area-Power Eficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", Procedings of the 204 International Conference on VLSI (VLSI'04), Las Vegas, Nevada, June 204, p. 434-439.
[7] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Procesing", Global J. of Eng. Edu, Vol.8, No.2, 204, UICEE Published in Australia.
[8] Thapliyal H. and Srinivas M.B, "High Sped Eficient NxN Paralel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Transactions on Enginering, Computing and Technology, 204, Vol.2.
[9] Harpret Singh Dhilon and Abhijt Mitra, "A Reduced–Bit Multiplictiion Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2.2 @ www.waset.orgSpring208.

ISSN: 2278 – 909X