ISSN 2395-1621

Time Efficient Architecture of FFT Processor Using Vedic Algorithm

[#1]Prof. A.S.Deshpande, [#2]Kiran Sable, [#3]RutujaPatil, [#4]PoojaPawar

[2]kimaya271994@gmail.com
[3]14rutuja.patil@gmail.com

[#1]Prof. Department of Electronics and Telecommunication
[#234]Department of Electronics and Telecommunication

JSPM Imperial College Of Engineering. And Research, Pune.

ABSTRACT

Fast Fourier Transform(FFT) is an important data processing technique in communication systems, digital image and signal processing systems. Here, we have proposed a high speed and an area efficient 4-point FFT processor by making use of Vedic algorithm.To decrease the computational complexity, area and to increase the speed we have developed the vedic FFT architecture by designing a Radix-2 algorithm and optimizing the throughput by Vedic algorithm. The obtained results are similar to that of the theoretical analysis and there is more than 15% reduction in terms of slices count. In addition, the power consumption is reduced thus increasing the speed by 16% using Vedic multiplier algorithm.

Keywords : **FFT, Vedic multiplier algorithm, Radix 2**

ARTICLE INFO

Article History

Received :16th March 2016

Received in revised form :

18th March 2016

Accepted : 20th March 2016

Published online :

23rd March 2016

I. INTRODUCTION

Now a days, the most widely used application of FFT is the OFDM[1]. It is mainly used in WLAN, digital audio broadcasting (DAB), digital video broadcasting-terrestrial (DVB-T) and digital video broadcasting-handheld (DVB-H). Due to such vast application of FFT, it is required to develop efficient FFT to meet the requirement of various OFDM communication standards[1]. The FFT is faster than Discrete Fourier Transform (DFT) and calculates DFT which is efficiently used in our work, thus reducing the computational complexity. Memory-based processors are widely used to design a FFT processor, which consists of butterfly processing elements and memory units. It has low power consumption but long latency and low throughput. To increase the efficiency of FFT architecture, radix-2 butterfly processing units along with dual port memory is adopted. Urdhva-Tiryakbhyam Sutra was first used in binary system and now is used to develop digital multiplier system. This Sutra reduces the N×N multiplier module into an efficient 4×4 multiplier structures effectively. This work represents a systematic mechanism for fast and an area efficient digital multiplier using Vedic mathematics.

II. SYSTEM DEVELOPMENT

Hence to improve performance, we have proposed a radix-2 FFT processor that operates on data length of 4-points with minimum area consumption. Fig.1 depicts the block diagram of FFT.

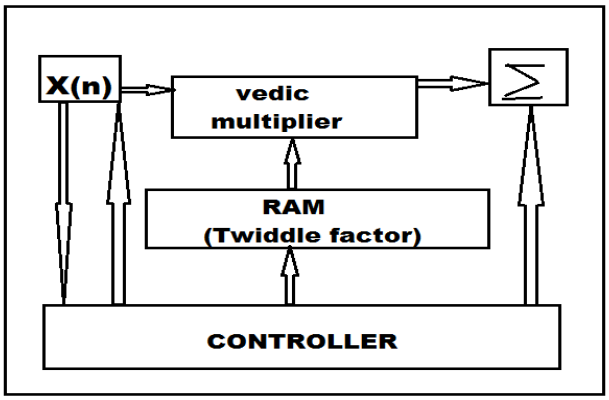


Fig 2.1. system block diagram

X[n] is the sample signal, on whose samples the multiplication is to be performed .The controller which is used is meant for controlling the values of twiddle factors and the multiplicants. RAM is used to store the values of twiddle factors. Vedic multiplier is the heart in this the actual multiplication of terms takes place, and we get the coefficients of FFT.

II. (A) RADIX-2 ALGORITHM

An efficient class of radix algorithm for designing FFT is radix-2 algorithm. In radix-2 algorithm, there are log4N stages and every stage has N/4 4-point butterfly processing elements. The radix-4algorithm operates by decomposing N point input data into N/4 different points. The N-point FFT is fragmented to repeat micro operations called butterfly operations. For realization of FFT hardware, only a single butterfly element is enforced in the chip, this butterfly element will carry out all the calculations repeatedly. If parallel as well as pipeline processing techniques are used, an N point radix-r FFT can be executed by (N/r) log r N clock cycles [2]. Thus, radix-2 algorithm reduces number of complex multiplications. Also, the numbers of stages required are also reduced by half.

II. (B) VEDIC ALGORITHM.

These Sutras were previously used for the multiplication of 2 numbers in decimal number system. In this project, we apply the resembling concept as that of binary system to make the proposed design match with the hardware.

Urdhva tiryakbhyam is a multiplication formula which can be applied to all the cases of multiplication. It actually means "Vertical and Crosswise multiplication". It is based upon simple concept by which all the partial products can be generated with the simultaneous addition of these products. The algorithm is used to generalize n x n bit number. Since the partial products and their respective sums are calculated in concurrent, this is not dependent on the clock frequency of the DFT processor. This Multiplier which is based upon this algorithm has an advantage that ,as the number of bits of data increases, gate delay and area also increases gradually as compared to other customary multipliers [3].

III.PROPOSED SYSTEM

In this project we have proposed a radix-2 FFT processor operates on data length of 4-points with minimum area consumption. driver cards are driven by microcontroller The initial step is to vertically multiply LSB's of two numbers. Carry bit generated. Due to this it is transferred to the second step and the result bit goes to the last result. In this step, it performs crosswise multiplication of adjacent number. Again, the previous carry bit is added here to yield final result and carry bit is propagated to next step. In the third step, the algorithm executes vertical-crosswise multiplication and previous carry bit is added to give final product. In this way, this algorithm performs multiplication of the two given numbers in vertical and crosswise manner until left with only MSB bits. The proposed FFT uses Urdhva-tiryakbhyam algorithm to perform complex twiddle factor multiplications. [4]

These Vedic multiplications which are performed, are nothing but the repeated addition of the numbers on which the operation has to be performed. These additions makes use of two different adders-adder 1 and adder 2.The programming and functioning of these are based on the coding done in VHDL, its simulation resultsand test bench waveforms are depicted in the figure below.In any multiplier Adder is a basic building block .Performance of multiplier is mainly dependent on propagation delay provided by adder.

Two types of adders are used,

1. Carry Save adder (CSA)
2. Ripple Carry Adder(RCA)

Ripple carry adder is used in only last stage of multiplier and Carry save adder is used in remaining all stages. Ripple carry adder generates more delay than the carry save adder.

1.Carry Save Adder

There are many cases where it is desired to add more than two numbers together. The straightforward way of adding together m numbers (all n bits wide) is to add the first two, then add that sum to the next, and so on. This requires a total of m − 1 additions, for a total gate delay of O (m*log n) (assuming look ahead carry adders). Instead, a tree of adders can be formed, taking only O(logm · log n) gate delays.

Using carry save addition, the delay can be reduced further still. The idea is to take 3 numbers that we want to add together, x + y + z, and convert it into 2 numbers c + s such that x + y + z = c + s, and do this in O(1) time. The reason why addition cannot be performed in O(1) time is because the carry information must be propagated. In carry save addition, we refrain from directly passing on the carry information until the very last step.

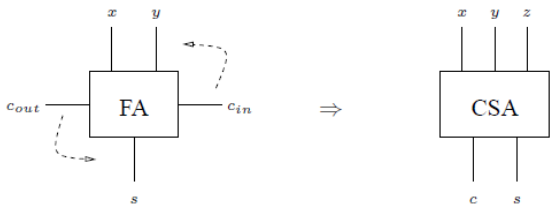
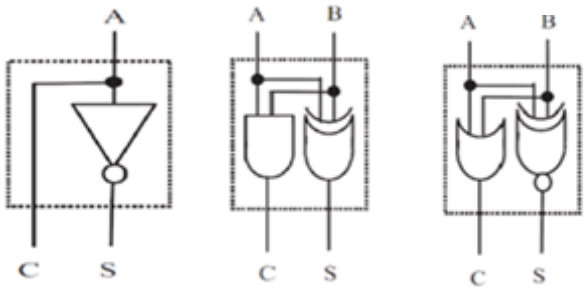


Fig 3.1 The carry save adder block.



(a)+1 Adder (b) A+B Adder (c) A+B+1 Adder

Fig 3.2 Adders

2. Ripple Adder

To obtain final result outputs of all carry save adders Ci and Si are added using Ripple Carry Adder.

s:		6	0	9	9	4
c:	+	1	0	1	1	
sum:		7	1	1	0	4

The important point is that c and s can be computed independently, and furthermore, each ci (and si) can be computed independently from all of the other c's (and s's). This achieves our original goal of converting three numbers that we wish to add into two numbers that add up to the same sum, and in O(1) time. The same concept can be applied to binary numbers.

3. Adder Cell

Structure of Adder cell is as shown below,

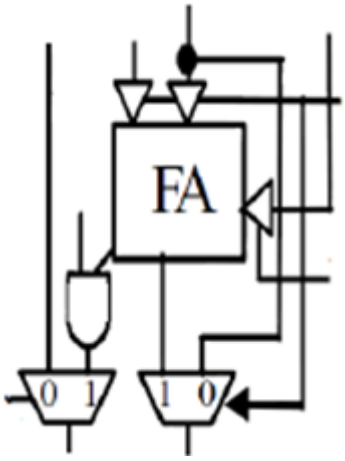


Fig 3.3 Adder cell

ALGORITHM

Following is the flow of the implementation. It helps to design Vedic Multiplier in effective manner and easy way.

Steps:

1. Start.
2. Implement the design components,
   - Adders
     a. Adder -1
     b. Adder -2
   - Vedic multiplier

3. Design the adder cells, using 'and' logic for bypassing and above components. Synthesize and test it's functionality. 4. Design the row by port mapping adder cells, Synthesize and test it's functionality.

5. Design the main multiplier by port mapping row module, Synthesize and test it's functionality.

IV. SIMULATION RESULTS AND TESTBENCH WAVEFORMS OF THE ADDERS

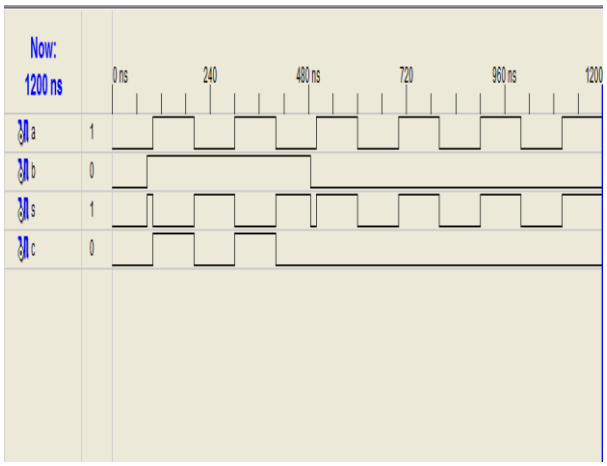


Fig 4.1. Testbench Of Adder1

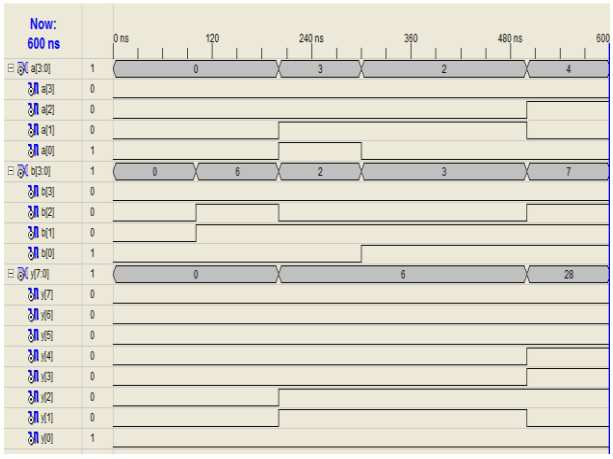


Fig 4.2. Testbench Of Adder2

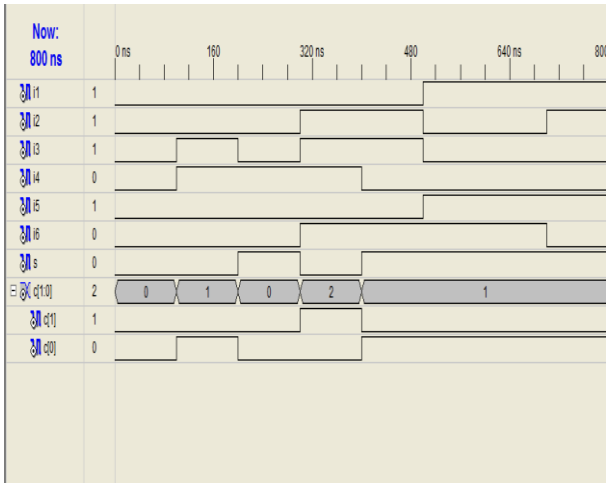


Fig 4.3 Final simulation result

V. CONCLUSION

Thus proposed architecture of this FFT processor was modeled using VHDL language. The entire architecture was synthesized and implemented by using Xilinx ISEv13.1.The functionality was then tested by creating the test bench waveform and used in behavioral and post layout

simulations. The hardware architecture of Vedic multiplier is also shown and found to be very much similar to the array multiplier. This is one of the many possible applications of Vedic Mathematics to Engineering and some serious efforts are required to fully utilize the potential of this interesting field for the betterment of Engineering and Technology. Knowingly or unknowingly we always use Vedic Sutras in everyday world of technology.

REFRENCES

[1] Nisha John, Prof. Sadanandan G.K, "FPGA Implementation of a Novel Efficient Vedic FFT/IFFT Processor For OFDM", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 3, September 2014.

[2] More T.V. , Panat A.R. "FPGA implementation of FFT using vedic algorithm",Computational Intelligene and Computing Research (ICCIC), pp-1-5,2013.

[3] Harpreet Singh Dhillon, AbhijitMitra, "A Digital Multiplier Architecture using UrdhvaTiryakbhyam Sutra of Vedic Mathematics",IITG , pp-1-4,2010.

[4] A.Ronisha Prakash, S.Kirubaveni, "Performance Evaluation of FFT Processor Using Conventional and Vedic Algorithm", IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology, pp-1-6, 2013.

