Architecture of FFT Processor Using Vedic Algorithm

Prof. A. S. Deshpande[1], Rutuja B. Patil[2], Pooja P. Pawar[3], Kiran Sable[4]
[1, 2, 3, 4] Department of Electronics and Telecommunication
[1, 2, 3, 4] Imperial College of Engineering and Research, Wagholi.

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, Urdhava Tiryakbhyam.

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. Urdhava-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.

Our basic concern is to increase the performance of the ineffective computational block of FFT processor by eliminating the complex critical path components and by using Vedic Sutra. Vedic algorithm depends on 16 different sutras dealing with several branches of mathematics like arithmetic, algebra, geometry. Therefore, to reduce the computational complexity and minimize the hardware cost the FFT.

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.

Thus, radix-2 algorithm reduces number of complex multiplications. Also, the numbers of stages required are also reduced by half.

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.

Urdhava 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.

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 Urdhava-tiryakbhyam algorithm to perform complex twiddle factor multiplications.

DESIGN IMPLEMENTATION

The design is implemented using the following steps:

i) Translate: The translate process runs NGD Build that merges all the input net lists (the logical design data) and the design constraints information to create the .ngd file.

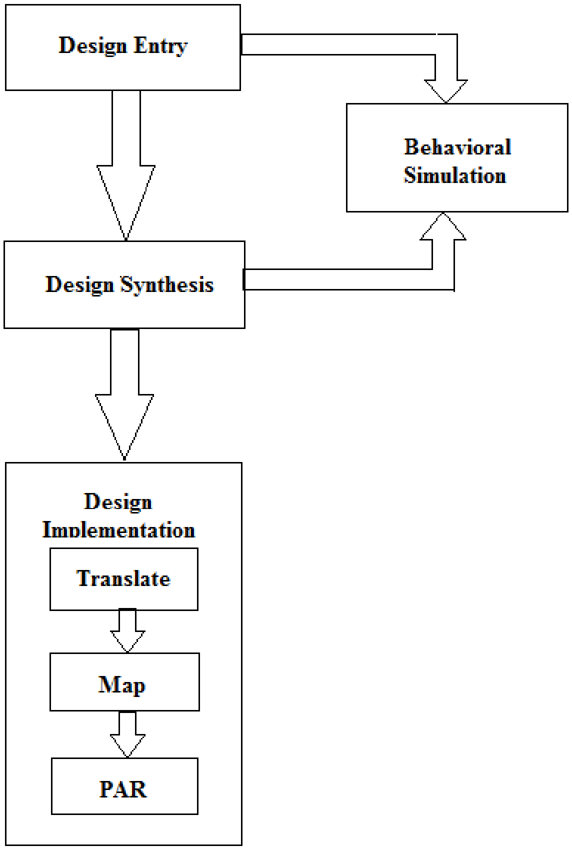


Fig. 4.2 ISE Design Implementation

ii) Map: The map process maps the logic designed in the .ngd file on the target FPGA. The output .ncd file contains the information of the resources such as the CLBs, registers and IOBs consumed on the target FPGA.

iii) Place and Route: Once the design is mapped, the design is then placed and routed (PAR) on the FPGA. This means that the resources described in the .ncd file are assigned specific locations on the FPGA. The connections between the resources are then mapped onto the FPGA's programmable interconnect network. The PAR process has a large impact on the speed of the design and therefore it generates an updated .ncd file with the static timing report. This report gives us the information for the clock frequency, the slack, whether there are any timing violations and the clock-to-pad and pad-to-clock delays.

The process of hardware implementation of the design on FPGA is completed if there are no errors and the timing constraints are met. The design summary generated after the complete process contains information of the resource utilization for the target FPGA and a detailed synthesis report. The design summary also displays the clock for the system in the clock report and the timing details are listed in the static timing report.

DESIGN FLOW:

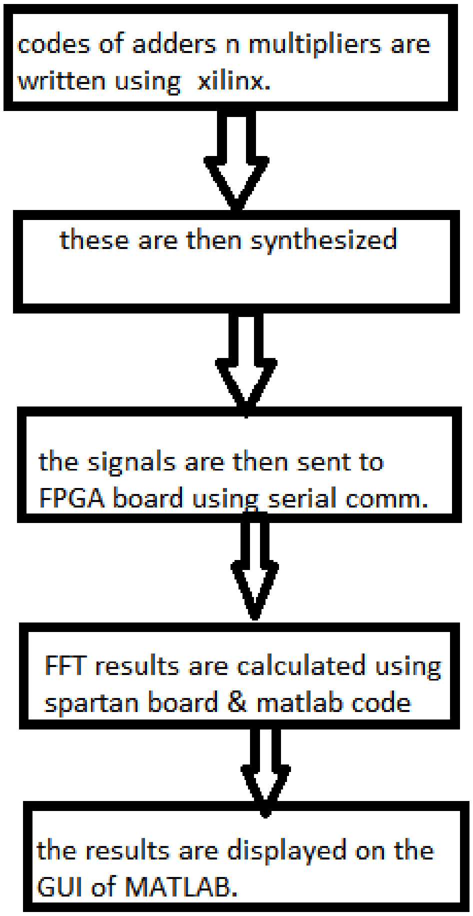

codes of adders n multipliers are written using xilinx.
these are then synthesized
the signals are then sent to FPGA board using serial comm.
FFT results are calculated using spartan board & matlab code
the results are displayed on the GUI of MATLAB.

Fig. Design flow

HARDWARE ARCHITECTURE

One of the main advantages of this Vedic multiplier algorithm (Urdhava-Tiryakbhyam Sutra) is that it can be realized in hardware. The hardware implementation of 4-bit multiplier using this Sutra is shown in Fig. 4.

This hardware design is almost similar to the famous array multiplier where an array of the adders is required to arrive at the last step. All the partial products are then calculated concurrently and the delay associated with it, is mainly the time taken by the carry to propagate through adders which form the multiplication array.

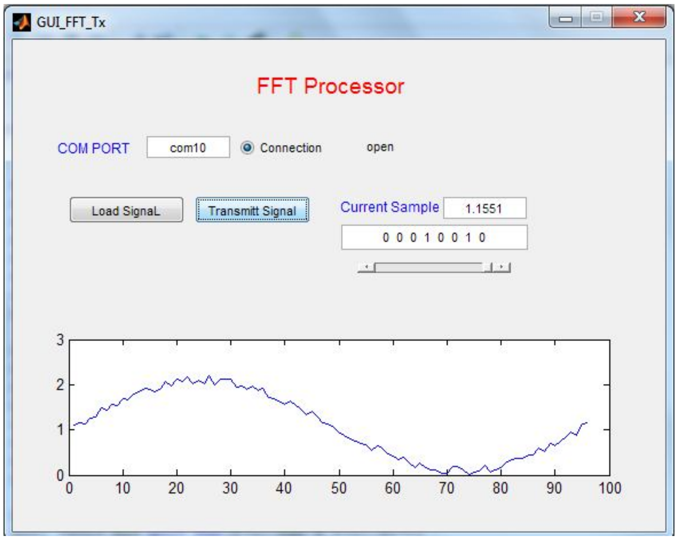
IV. CONCLUSION

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.

V. RESULTS


GUI_FFT_Tx

FFT Processor

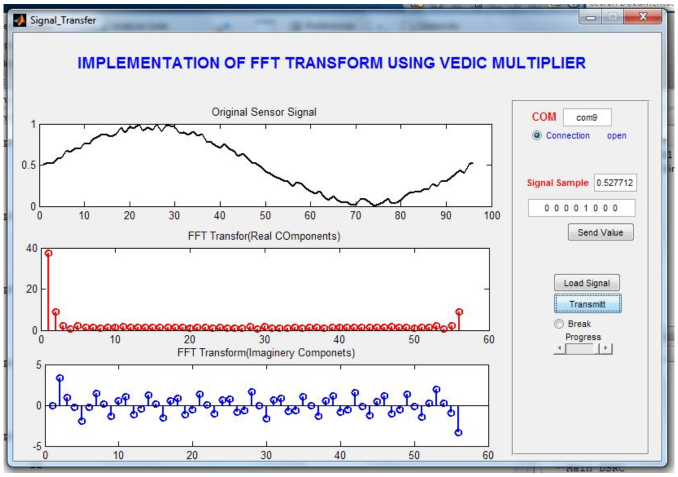
COM PORT com10 ● Connection open

Current Sample 1.1551

0 0 0 1 0 0 1 0

Fig 4.1.Transmission of single sample value on FFT Processor

For the output to be viewed, the hardware is made compatible with the PC by using prolific interface.


Signal_Transfer

IMPLEMENTATION OF FFT TRANSFORM USING VEDIC MULTIPLIER

COM com9 ● Connection open

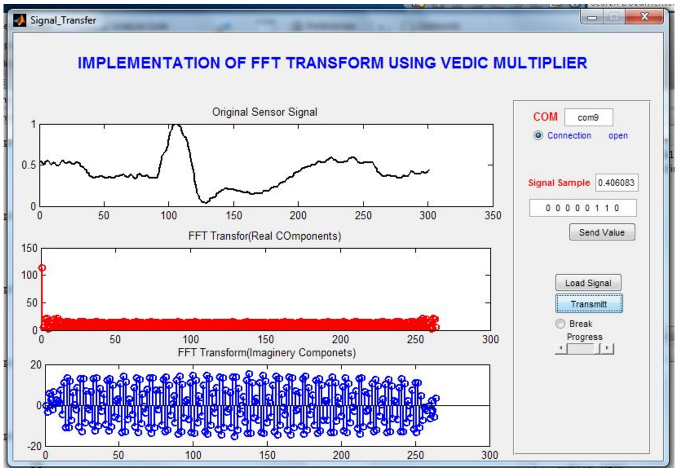
Signal Sample 0.527712

0 0 0 0 1 0 0 0

○ Break

Fig 4.2 Separation of Real and Imaginary Components of Original Signal

By selecting the COM port with the interfaces, we select the desired COM port and when the connection is OPEN then we load the signal and transmit it.



COM com9

● Connection open

Signal Sample 0.406083

0 0 0 0 0 1 1 0

Fig 4.3 FFT Transmission

Thus we have observed the output signal in real and imaginary form.

ACKNOWLEDGEMENT

We are pleased to acknowledge many individuals for their valuable guidance during the course of this project work.

We would like to acknowledge the foremost gratitude and deep appreciation to our internal Project Guide Prof. A. S. Deshpande for her valuable advice, guidance, encouragement and support for project, who continuously helped us throughout the project and without her guidance, this project would have been an uphill task.

REFERENCES

[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, Abhijit Mitra, “A Digital Multiplier Architecture using Urdhva Tiryakbhyam Sutra of Vedic Mathematics”,IITG , pp-1-4,2010.

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

[5] Asmita Haveliya, “FPGA implementation of a Vedic convolution algorithm”, International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp.678-684,Jan-Feb 201

[6] Leonard Gibson Moses S, Thilagar M, VLSI Implementation of High Speed DSP algorithms using Vedic Mathematics““‘, International Journal