International Journal of Scientific Research and Modern Education (IJSRME)
Impact Factor: 7.137, ISSN (Online): 2455 – 5630
(www.rdmodernresearch.com) Volume 3, Issue 1, 2018

FPGA IMPLEMENTATION OF 16-POINT FFT ALGORITHM USING DIGITAL SIGNAL PROCESSING

Hemlata Dalmia* & Sandeep Kumar**
Department of Electronics and Communication, Sreyas Institute of Engineering & Technology, Hyderabad

Cite This Article: Hemlata Dalmia & Sandeep Kumar, "FPGA Implementation of 16-Point FFT Algorithm Using Digital Signal Processing", International Journal of Scientific Research and Modern Education, Volume 3, Issue 1, Page Number 16-21, 2018.

Abstract:

DSP has become a key component, in many of the user, communications, medical and industrial products which implement the processing of signal using microprocessors, Field Programmable Gate Arrays (FPGAs), Custom ICs etc. Fast Fourier transform (FFT) has an important role in many digital signal processing (DSP) systems. E.g., in orthogonal frequency division multiplexing (OFMD) communication systems, FFT and inverse FFT are needed. The OFMD technique has become a widely adopted in several wireless communication standards. When operating in wireless environment the devices is usually to be powered using battery and, therefore, an energy efficient FFT implementation is needed. Signal processing concepts are often presented in a very mathematical and abstract format. This can discourage students from further exploration because of the apparent irrelevance to real world problems. In this paper, VLSI architecture for FFT algorithm is proposed. This architecture is authorized in Verilog language .Behavior simulation is done by using the Model Sim 6.0. PAR Simulation can be done by using the synthesis Xilinx ISE 10.1.

Key Words: FFT, FPGA, Verilog, DFT

1. Introduction:

In today's electronic world, all the systems are battery powered and hence forcing us to design the systems to be hardware efficient and power efficient. DSP (digital signal processing) including areas like audio and speech signal processing, digital image processing, communications, control of systems, biomedical signal processing, seismic data processing, etc, employ digital systems which carryout complex functionalities. The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals. Hardware efficient and power efficient designs for these systems are essential to achieve the best performance. DFT is one of the fundamentals used for representation of a discrete time sequence in frequency domain by its spectrum samples. The analysis (forward) and synthesis (inverse) equations of an N point FFT are given below.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \ldots, N-1.$$

The DFT algorithm can be used to approximate the transform of a continuous time function, subject to the limitations and difficulties: band limited signal, the high sampling rate to avoid aliasing, limited length of the signal for computational purposes and picket-fence effect. A fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and it's inverse. FFT is just a means of computing the DFT with a considerable reduction in the number of calculations required. The FFT is the basic data processing technique of the most of the systems and an important technique implemented on FPGA. DFT compute N points in the naive way, using the definition, takes O(N2) arithmetical operations, while an FFT can compute the same result in only O(N log N) operations. The most common FFT algorithm are the Cooley–Tukey algorithm. PFA algorithm, Winograd algorithm and so on. Due to increased utility of FFT in today's electronic systems, higher radix FFTs such as radix –4, radix – 8, radix – 2k, etc. is designed to improve timing and hardware constraints. The difference in the mentioned methods can be easily identified from of their butterfly structure of units.

Recently FPGAs has become a valid alternative as the technology is growing very fast. Therefore, in present days FPGAs play a key role in many areas due to their direct hardware performance as well as their inherent reprogrammability and flexibility features. In real-time applications, using FPGAs for FFT processing is becoming achievable. FFT in hardware is developing with speed due to its high throughput and low power.

The 16 samples are rearranged in the form of the radix-2 format. The number of stages is depending on the number samples taken into the account .The number of stage can be calculated by using this formula

$$\text{No. Stages} = Log_2^{numberofsamples} \qquad (2)$$

for radix-2 implémentation. Figure shows, the number of stages corresponding samples. The input samples are rearranged in such a way that for FFT process computation .

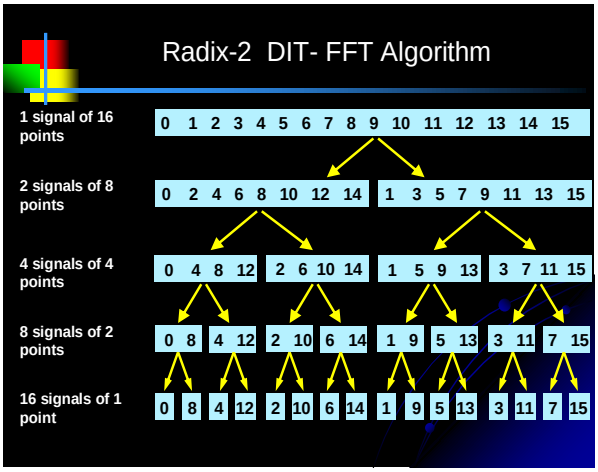

Figure 1: Samples distribution for the 16 samples of FFT

This process is called Bit-reversal format. The formation of bit reversal is the MSB bit is converted into LSB bit, the remaining bit are also follows same process only. This process is explained in given below

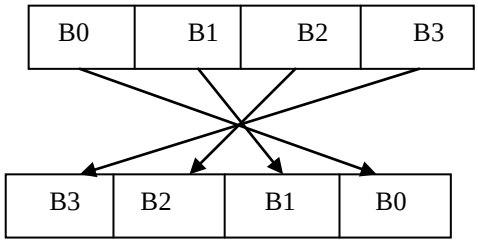

Figure 2: Bit Reversal Concept

This is done by using right shift by using the verilog programming.

2. Proposed Methodology:

The proposed architecture contains the four stages. Each stage is differed by the number of twiddle factors are included in the computation. The twiddle factor is a basic building element for the FFT processing. This indicates how the 360 degrees are distributed according to the number of samples. The Twiddle factor is calculated by using this formula

$$e^{j\theta} = \cos\theta + j\sin\theta \quad (3)$$

Where $\theta = (2\pi n / N)$ (4)

Where N indicates the number of samples and n is the integer in the range of 0 to 7. When substituting the n=0 to 7 values. Applying the n=0 to 7 the resulting values are used in Look Up table

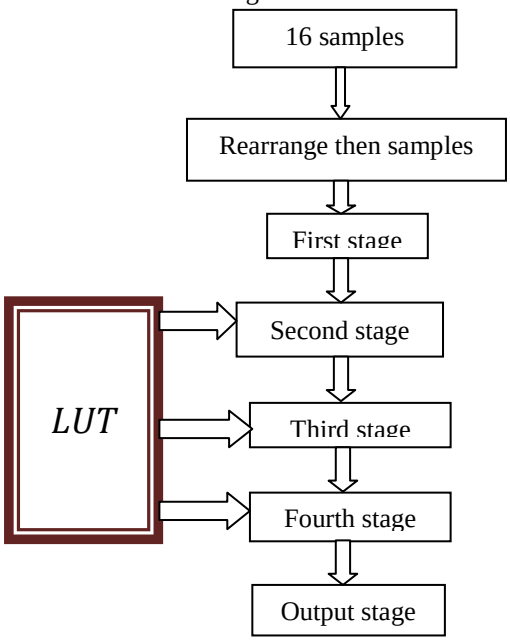

Figure 3: Flow Chart of Proposed Methodology

The stage 1 is having one twiddle factor. It is represented by using the W1160. The below table specifies the number twiddle factors with respect to the stages, the stages are derived from the number of samples

Table 1: Stages with Twiddle Factor

Stages	**Used Twiddle Factors**
First Stage	1
Second Stage	2
Third Stage	4
Fourth Stage	8

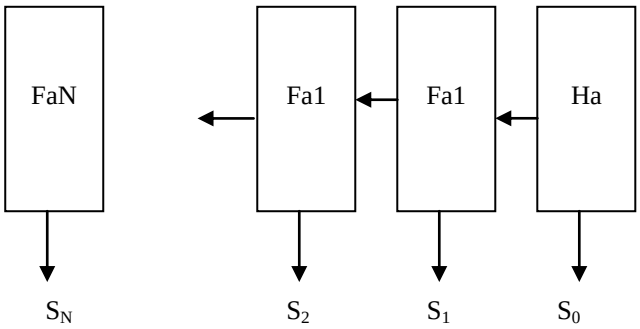

Figure 4: Adder Circuit for the Addition

This adder circuit is varying from one stage to another stage. This circuit is used as the upper portion of the total FFT process. The number of adder is calculated from the number of bits used to represents the samples. This paper discusses that by using Vedic mathematics in digital signal processing the it becomes easy and simple for FFT computation with the help of digital multipliers. Digital multipliers are required to implement important functions such as fast Fourier transforms (FFTs) and multiply accumulate (MAC). Two mostly used algorithms used for multiplication followed in the math coprocessor are array multiplication algorithm and booth multiplication algorithm [12]. The speed in computation increases by using array of multipliers because the partial products are calculated independently in parallel.The paper [13] has shown that the simple digital multiplier is designed based on the ancient Vedic mathematics Sutra (formula) called Urdhva Tiryakbhyam (Vertically and Cross wise) Sutra which was traditionally used for decimal system in ancient India. In , this Sutra is shown to be a much more efficient multiplication algorithm as compared to the conventional counterparts. Another paper [12] also representated the effectiveness of this sutra to reduce the N×N multiplier structure into efficient 4 × 4 multiplier structures. However, they have proved that this 4×4 multiplier section can be implemented using any efficient multiplication algorithm. We apply this Sutra to binary systems to make it useful in such cases. In particular, we develop an efficient 4 × 4 digital multiplier that calculates the partial products in parallel and hence delay decreases and speed is increased. The architecture of the designed Vedic multiplier looks similar to the popular array multiplier and hence the comaprison says that Vedic mathematics gives simpler and easier derivation of array multiplier than the conventional mathematics.

Step 1: **Right**

1 4
2 1
: 4 x 1 = 4

Step 2: **Middle**

1 4
2 1
: 1 x 1 + 2 x 4= 9

Step 3: **Left**

1 4
2 1
: 1 x 2 = 2

So this gives Result as: 294

2 (left)	9 (middle)	4(right)

Figure 5: Find the Product of 14 and 21

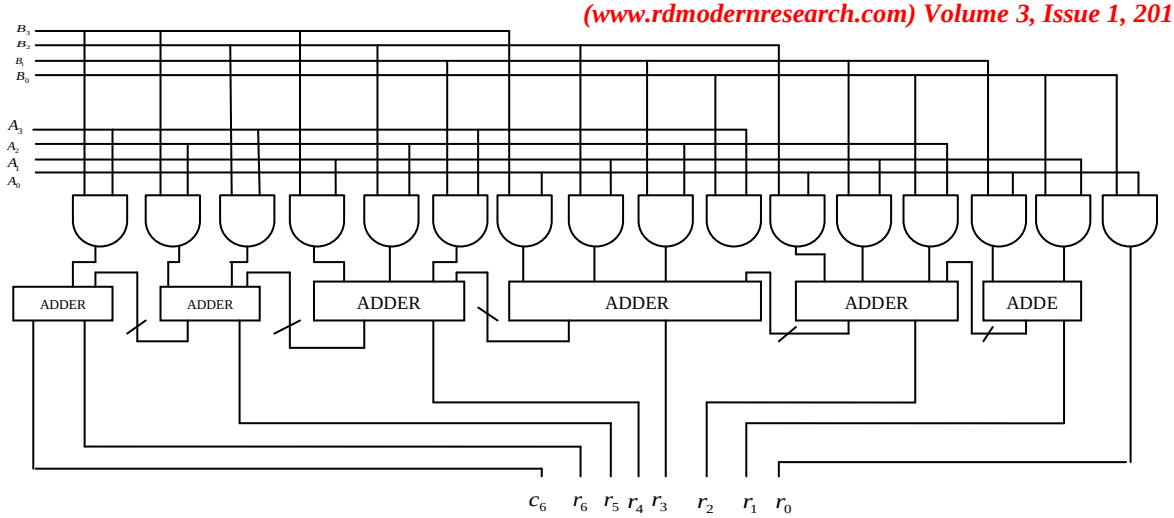

Figure 6: Hardware Architecture for proposed vertical and cross wise Multiplier

This multiplication is used as a component for the FFT multiplication process. This is the new method introduced for the multiplication operation. The multiplication process is extended to multiply any number of bits. This structure is extended for 8X8, 16X16, and 32x32… for any multiplication process applications. This method is fastest one for different application. In the second stage W160 and W164 component values are taken into account and multiplied with first stage outputs. In the third stage W160, W162, W164, W166 component values are taken into account and multiplied with second stage outputs. In the third stage W160, W161, W162, W163, W164, W165, W166, W167 component values are taken into account and multiplied with third stage outputs. All these W160, W161, W162, W163, W164, W165, W166, W167 values are stored LUT table. The LUT consists of 8 locations with one conation the W16N component

3. Result Analysis:

To perform synthesis this project uses Xilinx XST tool. The properties such as optimization effort, speed, power reduction and global optimization goal, apply to the Synthesize properties using the Xilinx® Synthesis Technology (XST) synthesis tool. Xilinx provide specified functions ranging in complexity from simple arithmetic operators such as adders, accumulators and multipliers, to system level building blocks including filters, transforms and memories. The Xilinx System can customize a generic functional building block such as a FIR filter or a multiplier to meet the needs of your application and simultaneously deliver high levels of performance and area efficiency.

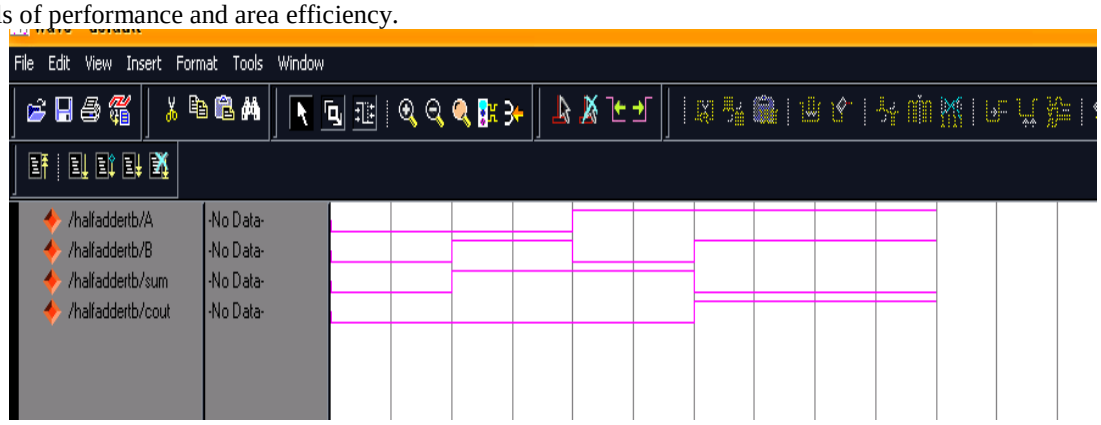

Figure 7: Half adder of two binary bits by using synthesis Xilinx ISE 10.1.

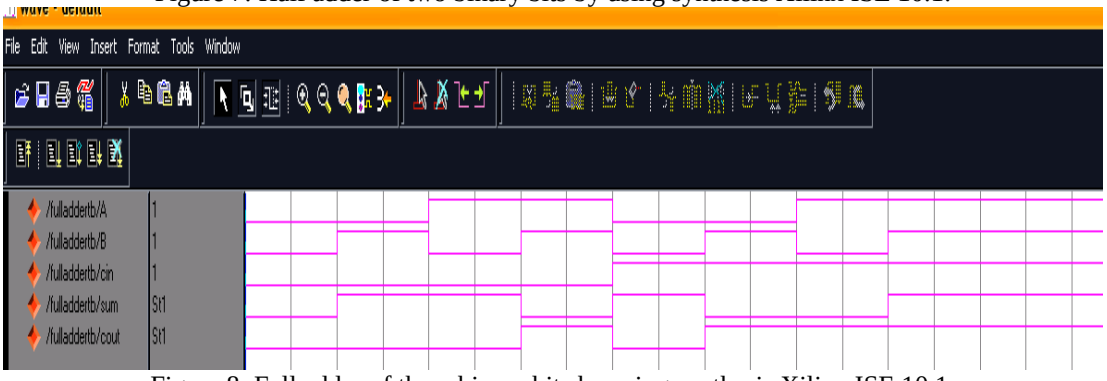

Figure 8: Full adder of three binary bits by using synthesis Xilinx ISE 10.1.

Fig. 1. Inverter for 16 bits by using synthesis Xilinx ISE 10.1.

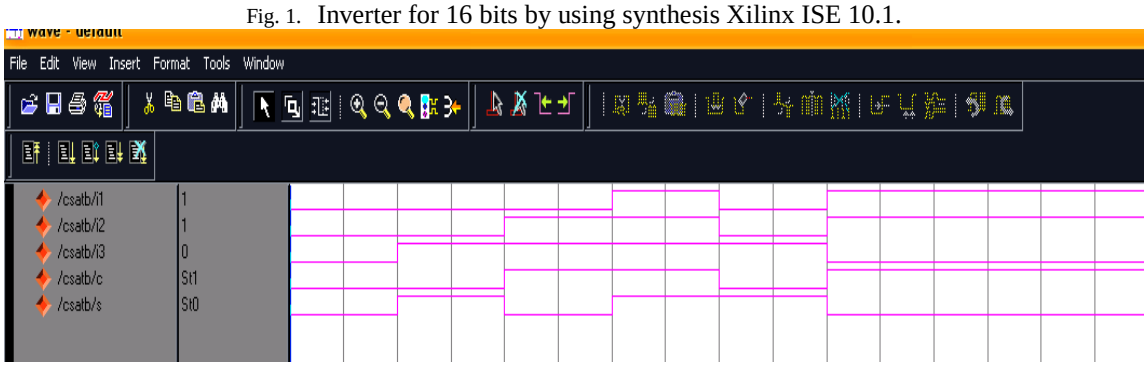

Fig. 2. CSA(Chirp Scaling Algorithm) in FFT by using synthesis Xilinx ISE 10.1.

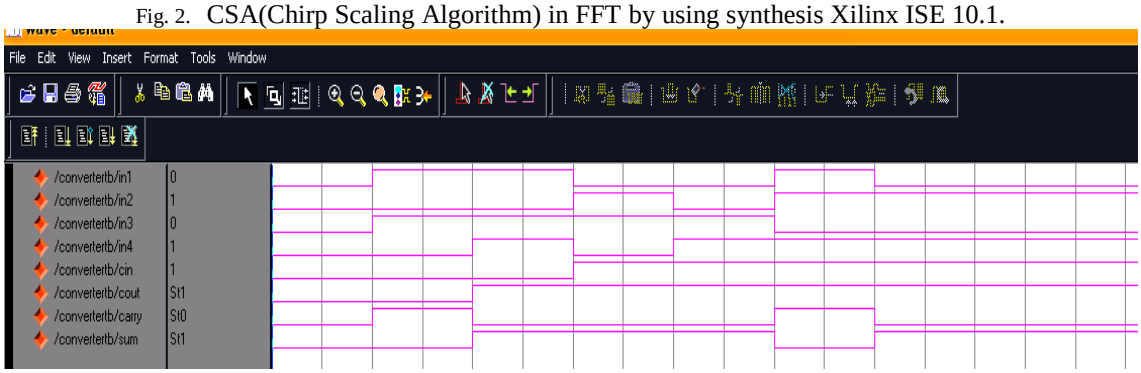

Fig. 3. Converter for 16 bits by using synthesis Xilinx ISE 10.1.

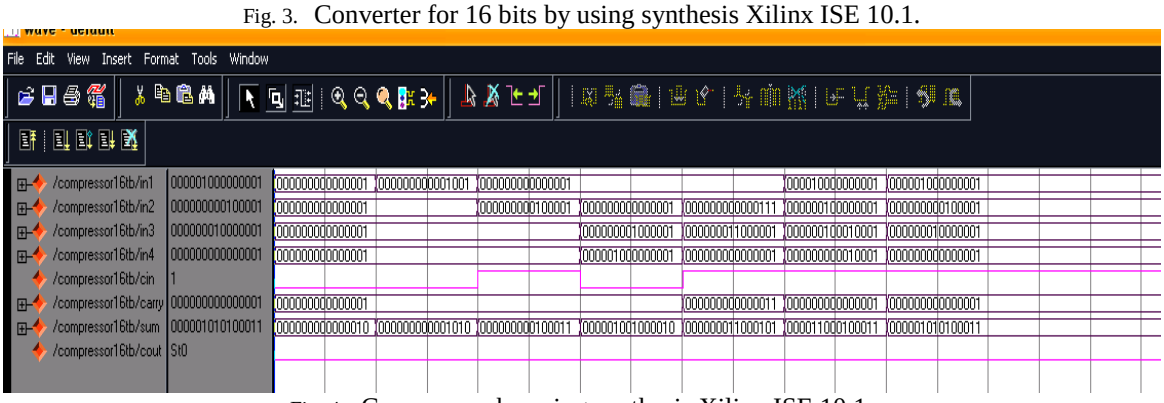

Fig. 4. Compressor by using synthesis Xilinx ISE 10.1.

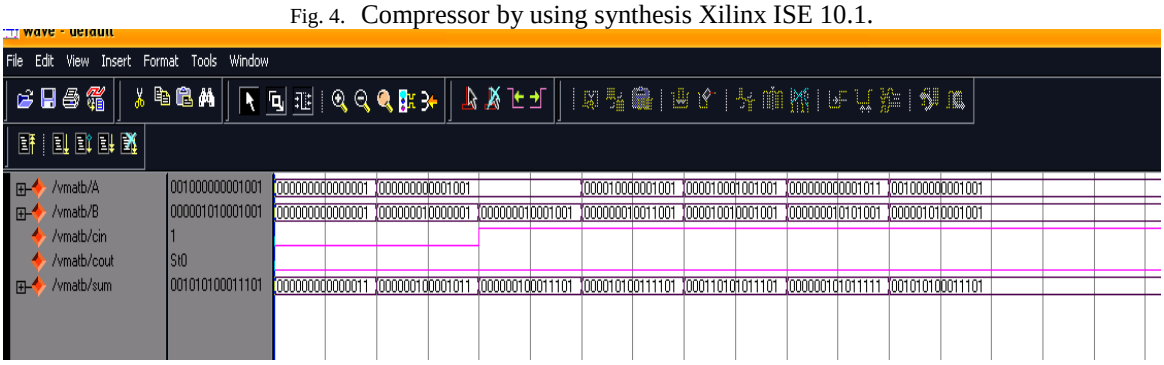

Fig. 5. VMA (Vedic multiplier Algorithm) by using synthesis Xilinx ISE 10.1.

4. Conclusion:

In this project we proposed Efficient VLSI architecture for FFT algorithm. This architecture functionality is verified by using Modelsim 6.0. And synthesized by using the XILINX 9.1I. The proposed VLSI architecture is implemented on the FPGA as it provides the flexibility of reconfigurability and reprogrammability. This proposed architecture is novel architecture for FFT algorithm. This is a scalable architecture for any length. Little bit of hardware architecture modification we can implement any FFT with different communication applications like OFDM and pulse compression in radar signal application.

5. References:

1. Despain, Alvin M. "Very Fast Fourier Transform Algorithms Hardware for Implementation," IEEE Trans. On Computers, Vol. 28, No. 5, pp. 333-341,1979.
2. Patli, Sudhir Subhashchandra. "Hardware Implementation of a Radix-4 Fast Fourier Transform Technique with Improved Two-Tone Resolution for Electronic Warfare Applications." PhD diss., Wright State University, Dayton, August 2004.
3. Tsui, James Bao-Yen. "Digital Techniques for Wideband Receivers," IEEE Tans. on Microwave Theory and Techniques, Vol. 45, No. 12, December 1997.
4. Sarathy, Vivek. High Spurious-Free Dynamic Range Digital Wideband Receiver for Multiple Signal Detection and Tracking, MS Thesis, Wright State University, Dayton, OH, December 2007.
5. Tsui, James Bao-Yen. Digital Techniques for Wideband Receivers, (Second Edition). Norwood, MA, Artech House, Inc., 2001.
6. Xilinx System Generator v2.1 for Simulink User's Guide. Online, www.mathworks.com/applications/dsp_comm/xilinx_ref/guide.pdf.
7. Spezio, Anthony E. "Electronic Warfare Systems," IEEE Transactions on Microwave Theory and Techniques, Vol. 50, No. 3, pp. 633-644, March 2002.
8. Harris, Fredric J., Chris Dick, and Michael Rice. "Digital receivers and transmitters using polyphase filter banks for wireless communications." IEEE transactions on microwave theory and techniques, Vol. 51, No. 4, pp. 1395-1412, 2003.
9. Xilinx Virtex-4 FPGA Overview. Online, www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex4/overview.htm
10. Fowler, Kim. "Analog-to-Digital Conversion in Real-Time Systems," IEEE Instrumentation and Measurement Magazine, September 2003.
11. D. Pok, C.-I. H. Chen, J. Schamus, C. Montgomery and J. B. Y. Tsui, "Chip Design for Monobit Receiver," IEEE Trans. Microwave Theory and Techniques, Vol. 45, No. 12, pp. 2283-2295, December 1997.
12. Laxman P. Thakre, Suresh Balpande, Umaeh Akare, Sudhair Lande 2010, " Performance evaluation and Synthesis of Multiplier Used in FFT Operation Using conventional and Vedic Algorithm", International Conference on emerging trends in Engineering and Technology, 614-619.
13. Kaustubh M. Gaikwad, Mahesh S. Chavan "Vedic Mathematics for Digital Signal Processing Operations: A Review" International Journal of Computer Applications (0975 – 8887), Vol .113 – No. 18, March 2015