Efficient VLSI Architecture for Xilinx Vertex E based FFT & IFFT Structure

[1]Mamta Raj, [2] Prof.Sanket Choudhary, [3] Dr. Soni Changlani
[1] M. Tech. Scholar, [2] Assistant Professor, [3] Head of Department
Electronics & Communication Department
Lakshmi Narain College of Technology & Science, Bhopal, (M.P) [India]

***Abstract* - Implementation of digital signal processing (DSP) algorithms in hardware, such as field VLSI, requires a large number of multipliers. Fast, low area multiply-adds have become critical in modern commercial and image DSP applications. A high speed fast Fourier transforms (FFT) and IFFT design by using 8-bit, 16-bit 32-bit and 64-bit algorithm is presented in this paper. My work focus is on two key ideas for improving FFT and IFFT algorithm performance: (1) develop new high performance efficient complex multiplier structure. (2) Parallel processing used in this design. In all algorithms are implemented Xilinx vertex 2 device family and simulated Modalism.**

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

I. INTRODUCTION

Due to the need of high-speed data transmission, today mobile telecommunications industry faces the problem of providing the technology that be able to support a variety of services ranging from voice communication with a bit rate of a few kbps to wireless multimedia in which bit rate up to 2 Mbps. Many systems have been proposed and OFDM system has gained much attention for different reasons. OFDM was first introduced in the 1960s, only in recent years; it has been recognized as an outstanding method for high-speed cellular data communication where its implementation relies on very high-speed digital signal processing. This method has only recently become available with reasonable prices versus performance of hardware implementation.

Since OFDM is carried out in the digital domain, there are several methods to implement the system. One of the methods to implement the system is using ASIC (Application Specific Integrated Circuit). ASICs are the fastest, smallest, and lowest power way to implement OFDM into hardware. The main problem using this method is inflexibility of design process involved and the longer time to market period for the designed chip.

Another method that can be used to implement OFDM is general purpose Microprocessor or Micro Controller. Power PC 7400 and DSP Processor is an example of microprocessor that is capable to implement fast vector operations. This processor is highly programmable and flexible in term of changing the OFDM design into the system.

The disadvantages of using this hardware are, it needs memory and other peripheral chips to support the operation. Besides that, it uses the most power usage and memory space, and would be the slowest in term of time to produce the output compared to other hardware.

The paper is organized as follows: Section II discusses the FFT & IFFT algorithm implementation radix-2 and complex multiplication used inside the butterfly-processing element. Section III shows the proposed three algorithms. Section IV shows the resulting implementation and finally a conclusion is given in section V.

II. FFT & IFFT ALGORITHM

The FFT is a complicated algorithm, and its details are usually left to those that specialize in such things. This section describes the general operation of the FFT, but skirts a key issue: the use of complex numbers.

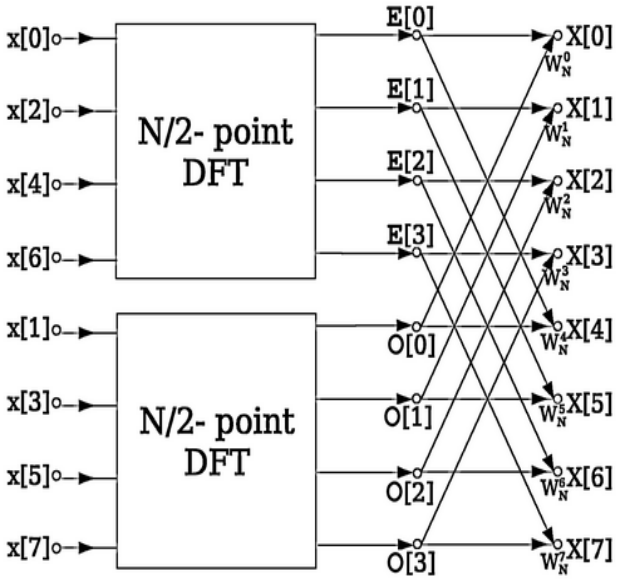


Figure 1: Block Diagram of 8-point FFT Algorithm

If you have a background in complex mathematics, you can read between the lines to understand the true nature of the algorithm. Don't worries if the details elude you; few scientists and engineers that use the FFT could write the program from scratch.

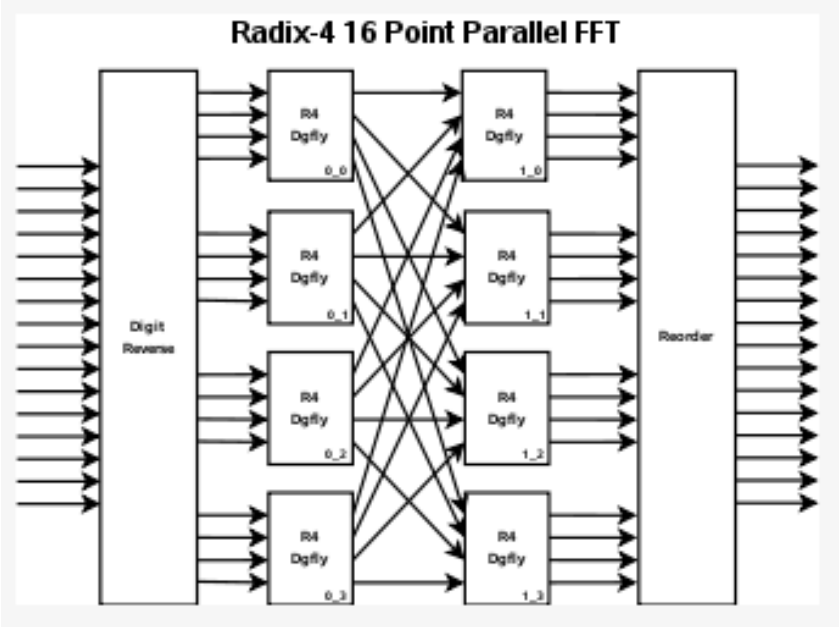


Figure 2: Block Diagram of 16-point FFT Algorithm

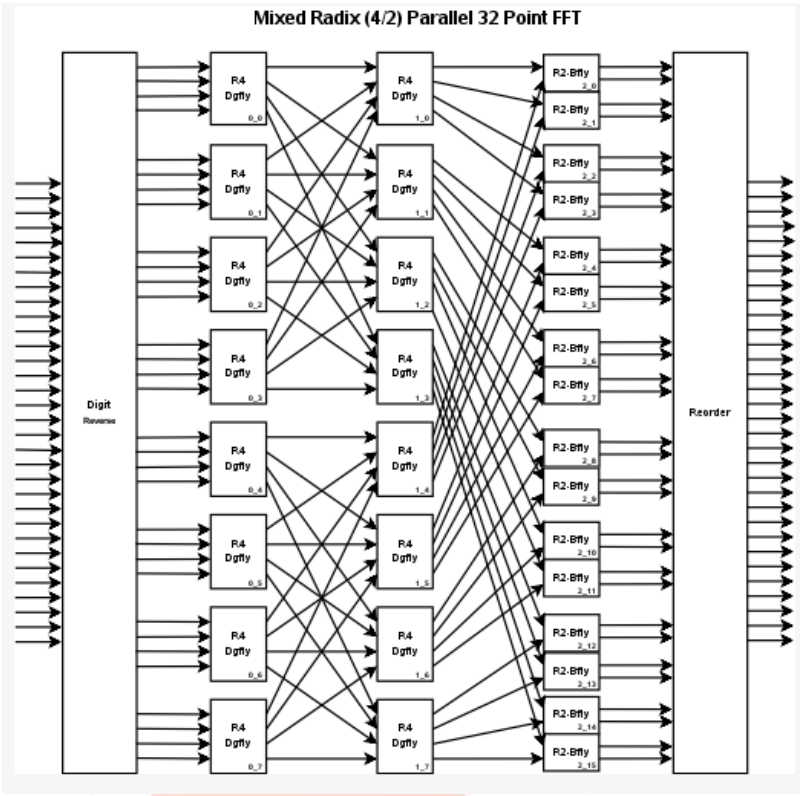


Figure 3: Block Diagram of 32-point FFT Algorithm

The FFT operates by decomposing an N point time domain signal into N time domain signals each composed of a single point. The second step is to calculate the N frequency spectra corresponding to these N time domain signals. Lastly, the N spectra are synthesized into a single frequency spectrum.

III. PROPOSED ARCHITECTURE

IFFT block is the main block of the proposed design can be divided into 6 stages. IFFT logic diagram is given in figure 4 with all 6 stages. Three main components used in these stages are parallel adder, parallel sub tractor and complex multiplier.

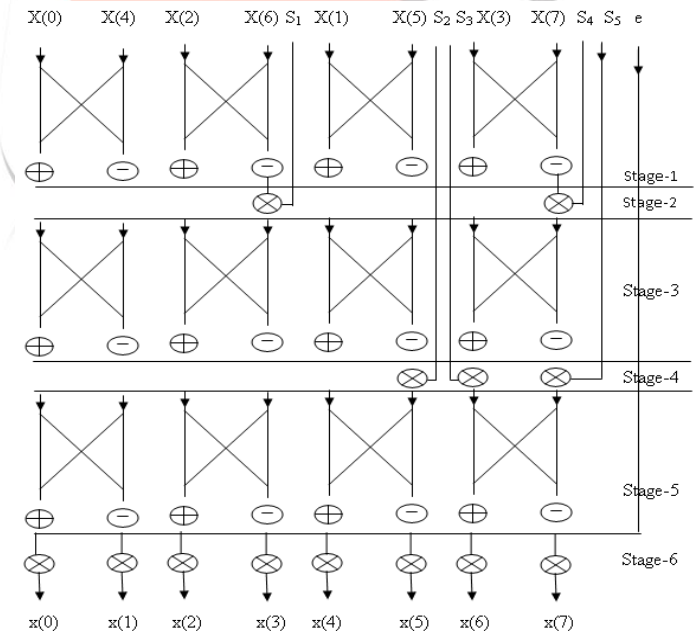


Figure 4: 8-point IFFT Structure

In the digital implementation of an IFFT operation, the single butterfly computation is implemented in the data path unit. A control unit controlling the data path and determine the stage of operations. The control unit coordinates the appropriate pairs of inputs into the butterfly computation and the output pairs is store in the memory. IFFT block can be divided into 6 stages. In 1st stage simultaneously addition and subtraction is done by using parallel adder for adding of inputs and parallel sub-tractor for Subtracting of input bits. The output comes out in 1st stage is multiplied by twiddle factor in 2nd stage.

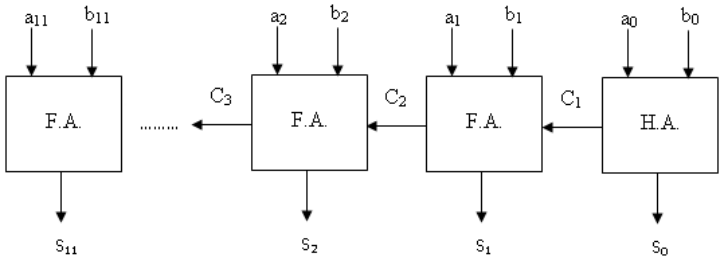


Figure 5: Logic Diagram of Parallel Adder

Twiddle factor multiplication is done by complex multiplier. In 3rd stage again simultaneously addition and subtraction is done of 2nd stage. In 4th stage again twiddle factor multiplication by complex multiplier.5th stage is also same as 1st and 3rd stage for adding and subtracting and in last stage i.e. 6th stage scaling is done by multiplying each output of stage 5th by 0.125.

In figure 5 shown the first component of parallel adder in 6-stage in IFFT algorithm. It is the combination of (n-1) full adder with one half-adder. For 8 bit 7 full-adder and a half adder is used.

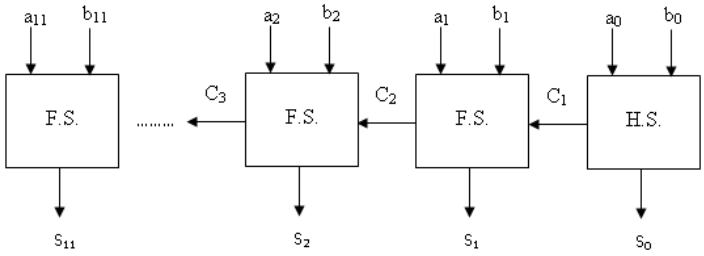


Figure 6: Logic Diagram of Parallel Sub tractor

In figure 6, shown the second component of parallel sub tractor in 6-stage in IFFT algorithm. It is the combination of (n-1) full sub-tractor with one half-sub-tractor. For 8 bit 7 full- sub-tractor and a half sub-tractor is used.

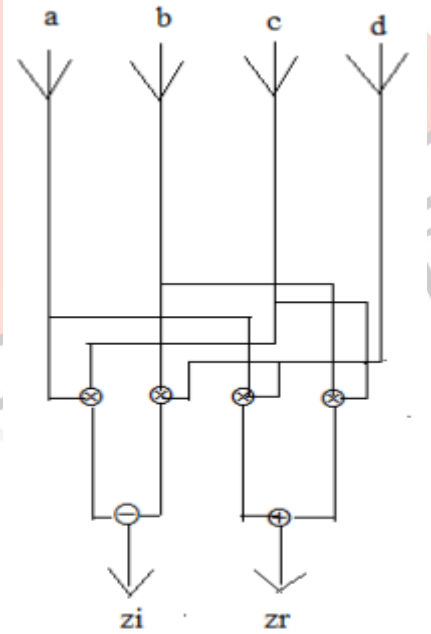
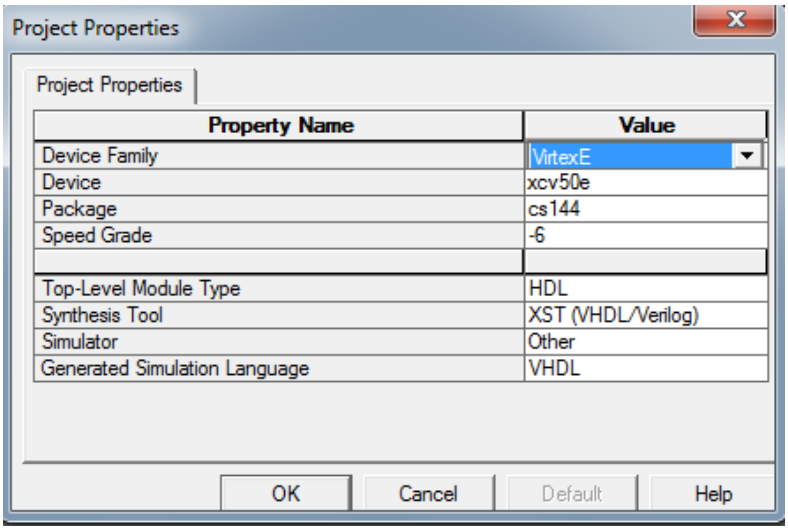


Figure 7: Logic of complex multiplier

In figure 7, shown the third component of complex multiplier in 6-stage in IFFT algorithm. It is used for multiplication of complex numbers such as A + JB with C + JD. After multiplication output comes in a form of (AC – BD) +J (AD + BC) component of complex multiplier are parallel-adder, parallel sub-tractor and multiplier.

IV. SIMULATION RESULT

VHDL is an acronym for VHSIC (Very high Speed Integrated Circuit) Hardware Description Language. It is a hardware description language that can be used to describe the structure and/or behavior of hardware designs and to model digital systems. VHDL is very adaptive, owing to its architecture, allowing designers, electronic design automation companies and the semiconductor industry to experiment with new language concepts to ensure good design tolls and data interoperability. Having designed the various DSP configurations, we now proceed to the software synthesis of this designs using VHDL. In the following sections, we have established the desired filter outputs using separate VHDL codes for every design. The structure so was discussed successfully implemented or synthesized on XILINX Project Navigator: 12.1i.



Implementing the all the stage up to six stages has been captured by VHDL and the functionality is verified by RTL and gate level simulation. To estimate the timing, area and power information for ASIC design, we have used Synopsys Design Compiler to synthesize the design into gate level.

Table1: Synthesized Results for stage up to six levels in IFFT.

	Number of slice	Number of Slice Flip Flops	Number of 4 input LUTs	Required time after clock (nsec)
Stage-I	116	203	256	10.752
Stage-II	101	177	256	13.991
Stage-III	195	342	256	19.382
Stage-IV	217	381	247	21.940
Stage-V	289	506	247	27.825
Stage-VI	356	621	192	49.528

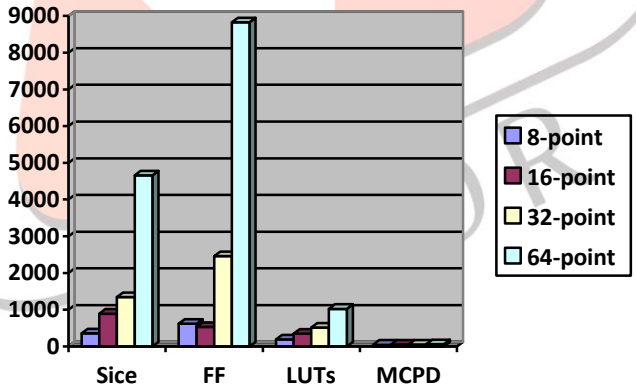


Figure 8: Chart diagram of 8-point, 16-point, 32-point and 64-point FFT and IFFT

MCPD: Maximum Combinational path Delay
Slice: Number of Slice
FF: Number of Slice Flip Flop
LUTs: 4-input LUTs

V. CONCLUSION

FFT and IFFT algorithm is successfully developed using VHDL software. The output from each module was tested using appropriate software to ensure the correctness of the output result. This is to ensure that the hardware module was correctly working when implemented in the FPGA hardware. During the implementation stage, the operation for IFFT is tested using Xilinx software.

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.

[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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] Harpret Singh Dhilon and Abhijt Mitra, "A Reduced– Bit Multipliction Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2.2 @ www.waset.orgSpring208.