SIMULATION OF RADIX-2 FAST FOURIER TRANSFORM USING XILINX

Remya Ramachandran
Department of EEE
Hindusthan College of Engineering and Technology,
Coimbatore, India

Vanmathi.k
Department of EEE
Hindusthan College of Engineering and Technology,
Coimbatore, India

Abstract— ***The Radix-2 decimation-in-time Fast Fourier Transform is the simplest and most common form of the Cooley–Tukey algorithm. The FFT is one of the most widely used digital signal processing algorithms. It is used to compute the Discrete Fourier Transform and its inverse. It is widely used in noise reduction, global motion estimation and orthogonal- frequency-division-multiplexing systems such as wireless LAN, digital video broadcasting, digital audio broadcasting. It is described as the most important numerical algorithm of our lifetime. The number of applications for this transform continues to grow. The Decimation-In-Time radix-2 FFT using butterflies has designed. The butterfly operation is faster. The outputs of the shorter transforms are reused to compute many outputs, thus the total computational cost becomes less. The 32 bit input FFT is synthesized using Verilog. The simulation results and the implementation details such as design summary, RTL schematic and others can be noticed. The design is developed using hardware description language VHDL / Verilog on Xilinx 14.2 xc3s500E.***

Keywords- Fast Fourier Transform (FFT); Decimation in Time (DIT); Radix-2

I. INTRODUCTION

The Discrete Fourier transform (DFT) is obtained by decomposing a sequence of values into components of different frequencies. The Fast Fourier transforms (FFTs) are the efficient algorithms to compute the DFT. The FFT algorithms are based on the principle of decomposing the computation of DFT into sequences of smaller DFTs [1]. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical. The FFT is used in various applications where the frequency-domain representation of a signal has to be analyzed. In the communications area, the FFT has gained attention because of its use in orthogonal frequency division multiplexing (OFDM) systems. For OFDM receivers, a FFT processing block is required [2]. Several communication systems require medium resolution (9–12 bits) analog-to-digital converters with bandwidths in the tens of MHz range [3]. The applications that use the FFT impose challenging specifications for its processing, such as small silicon area, high throughput, short processing time and reduced power consumption. For these applications, pipeline FFT architectures are accurate [4]. FFT has applications in mixed radix system, one of the popular numerical systems in which FFT numerical base or radix varies from one position to another position [5]. FFT is the most popular digital spectrum analysis technique [6].

The DFT is one of the fundamental operations in digital signal processing. The original computation of DFT with sample input requires complex multiplications. Cooley and Tukey first introduced the concept of FFT to demonstrate a significant computational reduction from to by making efficient use of symmetry and periodicity properties of the twiddle factors. These properties are:

$$\text{Symmetry property: } \omega_N^{k+N/2} = -\omega_N^k \qquad (1)$$

$$\text{Periodicity property: } \omega_N^{k+N} = \omega_N^k \qquad (2)$$

The related algorithms for the computation of the DFT are generally known as the FFTs. DFT and FFT are very popular signal processing tools [7]. An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition. Computing the DFT of N points in the naive way using the definition takes O(N2) arithmetical operations while a FFT can compute the same DFT in only O(N log N) operations. FFT module can be designed for the receiver and can be used for the transmitter IFFT with external conjugation either in hardware or software [8]. The discrete Hartley transform (DHT) is widely used in signal and image processing applications. The advantage of the DHT over the DFT is that it can be used to avoid complex operations when the input sequence is real. The forward and inverse DHTs differ from each other in their form only in the scaling factor [9]. The decimation in-time (DIT) and the decimation in- frequency (DIF) algorithms are the typical forms of the FFT algorithm.

II. 32 BIT DIT RADIX-2 FFT

The DIT radix-2 FFT recursively partitions a DFT into two half-length DFTs of the even-indexed and odd-indexed time samples. The outputs of the shorter FFTs are reused to compute many outputs, thus the total computational cost is decreasing. The radix-2 FFTs do not require explicit bit-reversal [10]. To lower the complexity, Cordic-based FFT/inverse FFT engines are used for spectral processing [11]. DIT FFT algorithm on $n = 2^p$ inputs with respect to a primitive n-th root of unity $\omega = \exp(2\pi j / n)$ relies on O(n log n) butterflies. In the case of the radix-2 Cooley–Tukey algorithm, the DFT of size-2 butterfly takes two inputs (x_0, x_1) and gives two outputs (y_0, y_1) by the formula

$$y_0 = x_0 + x_1\omega^k \quad (3)$$

$$y_1 = x_0 - x_1\omega^k \quad (4)$$

where k is an integer depending on the part of the transform being computed. The FFT processors use butterfly operations in which the arithmetic operations of complex valued data are performed [12]. An IEEE floating-point adder accepts normalized numbers, performs all IEEE rounding modes and will output the correctly normalized rounded sum/difference in the format required by the IEEE Standard [13]. The first step in the DIT algorithm is shown in Fig 1. The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to one-point sequences. For $N = 2^v$, this decimation can be performed $v = \log_2 N$ times.

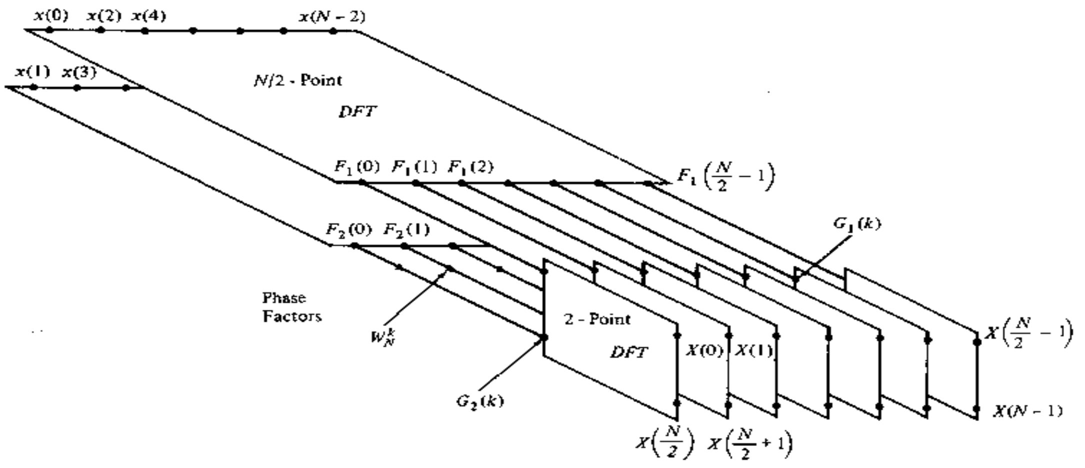
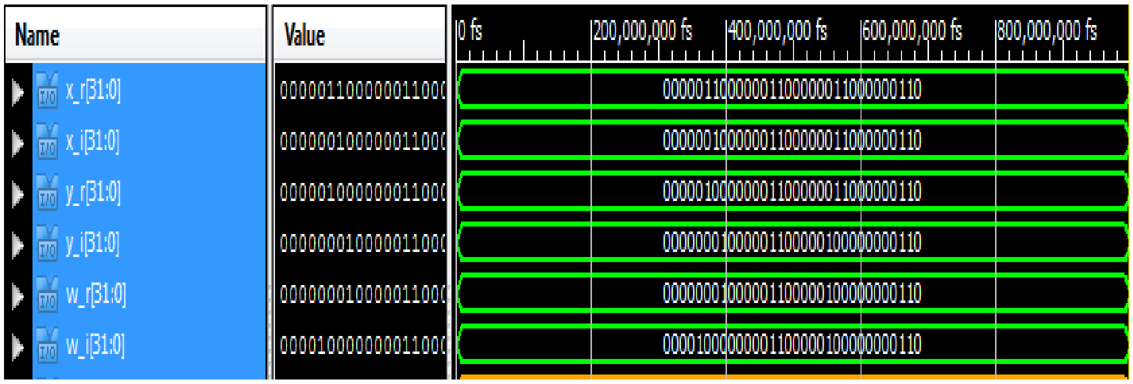


Figure 1: First step in the Decimation-In-Time Algorithm

A basic butterfly operation requires only N twiddle-factor multiplies per stage. FFTs of any factorable length may consist of successive stages of shorter-length FFTs with twiddle-factor multiplications in between. The twiddle factors may be computed or may be obtained from a memory. The efficient floating-point implementation of the butterfly units performs the computations in FFT processors. The twiddle factors are also computed in each stage according to the rules of butterfly operation. There are five stages in the 32 point FFT operations. The butterfly computation is simple and time consuming compared to other methods. The length-32, DIT FFT with the input data values or numbers scrambled and output data values or numbers in order of the 32 bit butterfly diagram. The twiddle factor computation is done at each stages of butterfly diagram. Twiddle factor are a set of complex roots of unity, fixed by the transform order and which must be applied to intermediate results in an FFT. This 32 bit radix-2 DIT FFT is designed and simulated in Spartan 3E xc3s500Eusing VHDL/Verilog synthesis using Xilinx 14.2.

III. SYNTHESIS RESULTS AND DISCUSSION

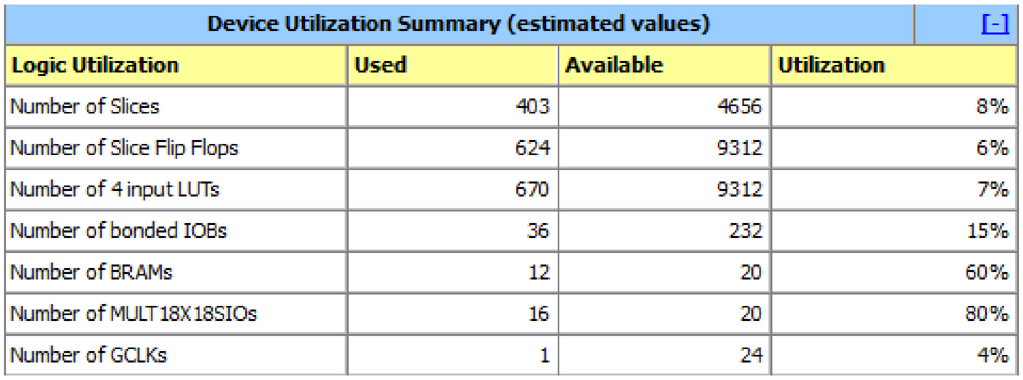
The 32 bit radix-2 DIT FFT is synthesized in Spartan 3E starter board as the evaluation development board. The family is Spartan 3E, the device used is xc3s500E, the package is FG320 and the speed is -4. The top level source type is HDL, the synthesis tool is XST(VHDL/Verilog), the simulator is Isim(VHDL/Verilog) and the VHDL source analysis standard is VHDL-93. For 32 bit, the 32 bit binary numbers in x_r[31:0] and x_i[31:0] are given as the input of real and imaginary parts and after the transformation using Verilog coding, the output of real and imaginary parts are obtained in y_r[31:0] and y_i[31:0]. The transformation is done with the help of butterfly diagrams. The result obtained after simulation in the Isim window is shown in Fig 2.



Name	Value	
x_r[31:0]	0000011000000011000	000001100000001100000011000000110
x_i[31:0]	0000001000000011000	000000100000001100000011000000110
y_r[31:0]	0000010000000011000	000001000000001100000011000000110
y_i[31:0]	0000000100000011000	000000010000011000001000000000110
w_r[31:0]	0000000100000011000	000000010000011000001000000000110
w_i[31:0]	0000100000000011000	000010000000011000001000000000110

Figure 2. Simulation Output of 32 Bit Input DIT Radix-2 FFT

The implementation results will be obtained in the synthesis of 32 bit radix-2 DIT FFT. The implementation details include the design summary from which the logic utilization is obtained, floor planning, I/O planning and timing constraints of clock domain, inputs, and outputs are shown below. The device xc3s500E design summary of 32 bit radix-2 DIT FFT is shown in Figure 3. The number of logics used in the design is obtained from the device utilization summary. The design can be done in this device since the percentage of logic utilization comes within 100%. The logic utilization details of 32 bit shows that the number of slices is 8%, slice FF is 6%, 4 input LUTs is 7%, bonded IOBs is 15%, BRAMs is 60%,18*18 SIOs is 80% and Gclks is 4%.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	403	4656	8%
Number of Slice Flip Flops	624	9312	6%
Number of 4 input LUTs	670	9312	7%
Number of bonded IOBs	36	232	15%
Number of BRAMs	12	20	60%
Number of MULT18X18SIOs	16	20	80%
Number of GCLKs	1	24	4%

Figure 3. Design Summary of 32 Bit Input DIT Radix-2 FFT

The timing constraints used in the user constraints while synthesizing 32 bit radix-2 DIT FFT is shown below. This includes timing of clock domain, inputs and outputs. The timing constraint of clock domain is shown in Figure 4 in which the clock pulse is given as the input. The timing constraint of inputs is shown in Figure 5. The timing constraint of output is shown in Figure 6.

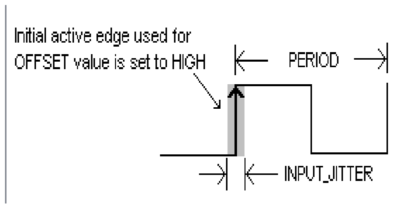


Figure 4. Clock Domain Timing Constraint

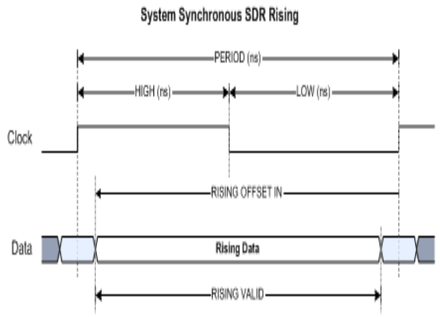


Figure 5. Input Timing Constraint

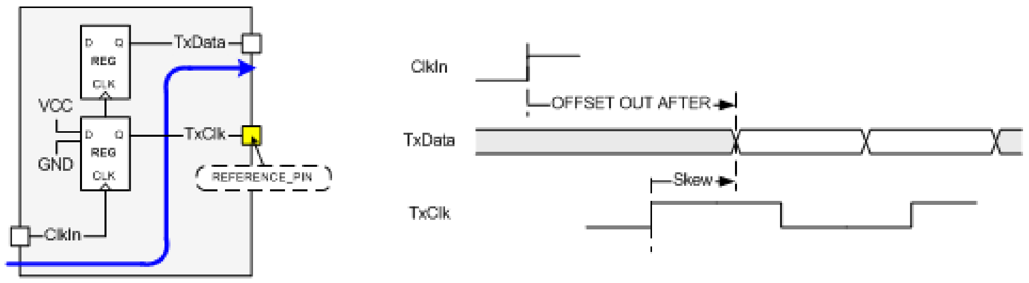


Figure 6. Output Timing Constraint

The floorplanning is shown in Figure 7. Floorplanning is the process of identifying structures that are placed close together, and allocating space for them so as to meet the conflicting goals of available space (cost of the chip), required performance, and desired to have everything close to everything else. The goals of floorplanning are to increase density, routability, or performance and to reduce route delays for selected logic by suggesting a better placement. Floorplanning allows you to choose the best grouping and connectivity of logic in a design and manually place blocks of logic in an FPGA device. Even a good floorplan does not guarantee that a design will meet timing. A bad floorplan will lead to waste-age of die area and routing congestion**.**

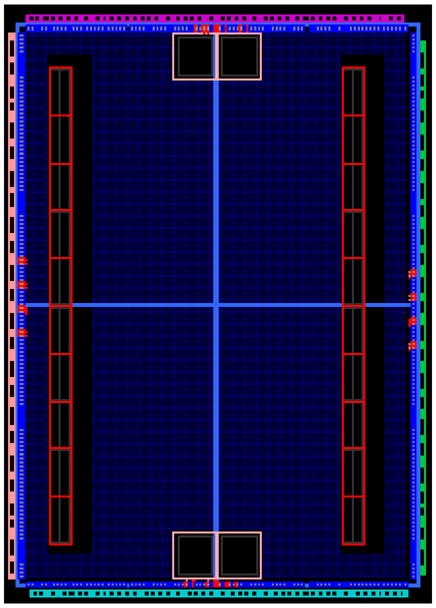


Figure 7. Floor Planning

The I/O pin planning or plan ahead of 32 bit radix-2 DIT FFT is shown in Figure 8. Input and output (I/O) signals can be assigned to package pins in the design. This process launches the PlanAhead™ software for FPGA designs and Pinout and Area Constraints Editor for CPLD designs. This process operates on the top module in the design before the design is synthesized. This allows assigning input and output signals to package pins before the underlying logic in the design has been developed. Constraints are saved to a user constraints file User Constraints File. You can create this file prior to running this process. The PlanAhead software extracts the top-level I/O port information from the associated HDL source files. Complicated design constraints are generated in practice to guide the I/O placement. Performing I/O pin assignment for FPGA devices describes the procedure for creating and assigning I/O ports to physical package pins.

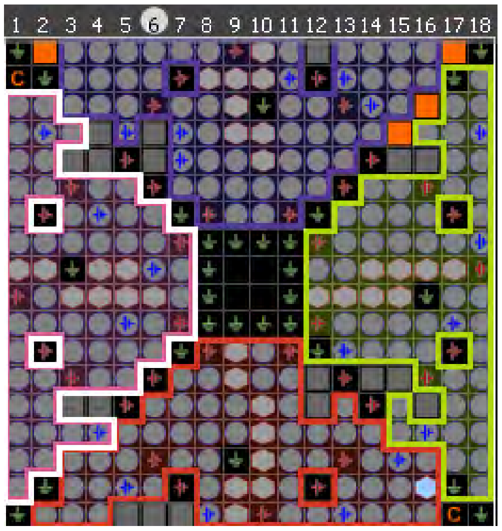


Figure 8. I/O Pin Planning

IV. Conclusion

In this work, the design of 32 bit radix-2 DIT FFT is implemented in the FPGA Spartan 3E. The total computational cost is less as the shorter ones are reused to compute many outputs. The simulation of 32 bit radix-2 FFT is done and outputs are obtained in Isim window. The design summary describes the logic utilization details which shows that the number of slices is 8%, slice FF is 6%, 4 input LUTs is 7%, bonded IOBs is 15%, BRAMs is 60%,18*18 SIOs is 80% and Gclks is 4%. The logic utilization of xc3s500E for implementing DIT radix-2 FFT increases as the number of bits for processing increases. The implementation details are noticed after synthesizing using the Xilinx 14.2. The timing constraints of clock domain input and output, floorplanning and I/O planning are noticed.

References

[1] Yuke Wang, Yiyan (Felix) Tang, Yingtao Jiang, Jin-Gyun Chung, Sang-Seob Songand Myoung-Seob Lim, "Novel memory reference reduction methods for FFT implementations on DSP processors", *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2338-2349, May 2007.

[2] NimaSadeghi, Vincent C. Gaudetand Christian Schlegel, "Analog DFT processors for OFDM receivers: circuit mismatch and system performance analysis", *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 56, no. 9, pp. 2123-2131, September 2009.

[3] J. Ankesh, V. Muthusubramaniam, and P. Shanthi, "Analysis and design of a high speed continuous-time δ∑ modulator using the assisted opamp technique", *IEEE Journal of Solid-State Circuits*, vol. 47, no. 7, pp. 1615-1625, July 2012.

[4] Ainhoa Cortés, IgoneVélezand Juan F. Sevillano, "Radix r^k FFTs: matricial representation and SDC/SDF pipeline implementation", IEEE TRANSACTIONS ON SIGNAL PROCESSING, vol. 57, no. 7, pp. 2824-2839, JULY 2009.

[5] Youngsun. H, Peter. H, Seon. W.K, Jong. K.K and Chulwoo. K, "A novel architecture for block interleaving algorithm in MB-OFDM using mixed radix system", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 1020-1024, June 2010.

[6] Jie. Q, Joseph. D.C, Bradley. F.D, George. J.S, Fa. F.D and Charles. E.S, "Selective spectrum analysis for analog measurements", *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4960-4971, October 2011.

[7] Tiziano Bianchi, Alessandro Pivaand Mauro Barni, "On the implementation of the discrete fourier transform in the encrypted domain", *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 86-97, MARCH 2009.

[8] R.S Sherrat, O. Cadenas and N. Goswami, "A low clock frequency FFT core implementation for multiband full-rate ultra-wideband (UWB) receivers", *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 798-802, August 2005.

[9] Longyu. J, Huazhong. S, Jiasong. W, Lu. W and Lotfi. S, "A novel split-radix fast algorithm for 2-d discrete hartley transform", *IEEE Transactions on circuits and systems—I: regular papers*, vol. 57, no. 4, pp. 911-924, April 2010.

[10] Jaime Seguel, Dorothy Bollman, and John Feo, "A framework for the design and implementation of FFT permutation algorithms", *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 625-635, July 2000.

[11] Hamid. E, Sang. V.T and Ahmed. M.E, "Design and implementation of a scalable channel emulator for wideband MIMO systems", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4698-4709, November 2009.

[12] E.E. Swartzlander Jr., and H.M. Saleh, "FFT implementation with fused floating-point operations", *IEEE Transactions on Computers*, vol. 61, no. 2, pp.284-288, February 2012.

[13] P.M. Seidel and G. Even, "Delay-optimized implementation of IEEE floating-point addition", *IEEE Trans. Computers*, vol. 53, no. 2, pp. 97-113, February 2004.

Remya Ramachandran received B.Tech Degree in Electronics and Communication Engineering in the year 2008 from Calicut University, Kerala, India. She is currently doing M.E. Degree in Applied Electronics under Anna University, Chennai, Tamil Nadu, India. She has published many papers in National / International Conferences to her credit. Her areas of interest are Signal Processing, Image Processing, Communication and VLSI.

K.VANMATHI received B.E and M.E Degrees in the years 1990 and 2004 respectively from PSG College of Technology, Coimbatore, Tamil Nadu, India. She has teaching experience of sixteen years. She is presently working as Associate Professor in the Department of Electrical and Electronics Engineering at Hindusthan college of Engineering and Technology. Her areas of interest are Linear machines, FEA of machines EMI and EMC issues. She has published many papers in National / International Conferences to her credit.