ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

VLSI IMPLEMENT ATION OF 8-BIT FAST FOURIER TRANSFORM (FFT) PROCESSOR BASED ON RADIX -2 ALGORITHMS

[1]SANGITA KUMARI, [2]SWETA KUMARI, [3]MANSI WAGHELA

[1, 2, 3]Masters of Technology Scholar, Department of Electronics & Communication Engineering, Jayoti Vidyapeeth Women's University, Jaipur- INDIA

Email: sangitachoudhary853@gmail.com

ABSTRACT

In this paper, a modular approach is presented to develop parallel pipelined architectures for the fast Fourier transform (FFT) processor. The new pipelined FFT architecture has the advantage of underutilized hardware based on the complex conjugate of final stage results without increasing the hardware complexity. The operating frequency of the new architecture can be decreased that in turn reduces the power consumption. A comparison of area and computing time are drawn between the new design and the previous architectures. The new structure is synthesized using Xilinx ISE and simulated using ModelSim Starter Edition. The designed FFT algorithm is realizedin our processor to reduce the number of complex computations.

KEYWORDS – Radix, FFT, Xilinx, Verilog HDL, Complex conjugate.

1. INTRODUCTION

The Fast Fourier Transform (FFT) is commonly used in the field of digital signal processing (DSP) such as filtering, spectral analysis, etc., to compute the discrete Fourier transform (DFT). Using this transform, signals can be moved to the frequency domain where filtering and correlation can be performed with fewer operations. The FFT algorithm should be chosen here to consider the execution speed, hardware complexity, and flexibility and precision [1], [2], [3].

1.1 FFT ARCHITECTURES

This section will review common FFT structures used.

Sequential Processor: The basic sequential processor consists of a

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

processing element (PE) that can compute a butterfly. The same memory can be used to store the data, intermediate results and the twiddle factors. The amount of hardware involved is very small and it takes (N/2) log2N sequential operations to compute the FFT.

Pipeline Processor: To improve the performance of the sequential processor, parallelism can be introduced by using a separate arithmetic unit for each stage of the FFT. This increases the throughput by a factor of log2N when the different units are pipelined. This architecture is also known as cascaded FFT architecture and will be used in our proposed design.

Parallel Iterative Processor: By adding more processing elements to the processor in each sequential pipeline stage, performance can be improved even further. The butterflies can then be computed in parallel in any stage. The total execution time for the parallel iterative processor is log2N cycles. This scheme is used in the FFT processor for radar signal processing.

Array Analyzer: A fully parallel structure can be constructed by having a PE for each of the butterfly operations. This involves a lot of hardware and is not an attractive option for a large N. Parallel pipelined Architectures: Pipeline FFTs contain an amount of parallelism equal to logRN where N is the number of points for an FFT and R is the radix. They can be generally run at high-speeds and the amount of pipelining increased or decreased to meet timing. A pipeline implementation of the FFT was first proposed in which consisted of a series of computational blocks each composed of delay lines, coefficient storage, commutators, multipliers, and adders. Torkelson have classified different pipeline approaches and put into functional blocks with unified terminology [9]. The pipelined architectures can be classified into two types: single-path architectures and multi-path architectures. Several single-path architectures have been proposed: Radix-2 single-path delay feedback, Radix-4 single-path delay feedback, Radix-22 single-path delay feedback, Radix-24 single-path delay feedback, Split-Radix single-path delay feedback, and Radix-4 single-path delay commutator. The multi-path architectures: Radix-2 multi-path delay commutator, Radix-4 multi-path delay commutator, Split-Radix multi-path

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

delay commutator, and Mixed-Radix multi-path delay commutator. The delay feedback architecture is more efficient than the corresponding delay commutator in terms of memory utilization and Radix-2n has simpler butterfly and higher multiplier utilization [4], [1], [5].

1.2 FFT ALGORITHMS

Some of the FFT algorithms are discussed as follows.

RADIX-2 Algorithm: The most basic FFT algorithm is the Radix-2 Decimation in Frequency Algorithm. This algorithm decomposes even and odd-indexed frequency samples.

RADIX-4 Algorithm: Decomposing the frequency samples into 2 bins (one each for odd and even samples) gives the Radix-2 Algorithm. Increasing the number of bins to 4 gives the Radix-4 Algorithm.

RADIX-8 Algorithm: Grouping 8 samples together gives the Radix-8 Algorithm.

RADIX-2/4 Algorithm: Split Radix algorithms in general use a separate decomposition for the odd and even samples. For example, the split radix 2/4 Algorithm uses radix-2 decomposition for the even samples and radix-4 decomposition for the odd samples. The advantage of this algorithm is that it provides the lowest known number of operations for computing length-2n FFTs. The disadvantage is that the resulting structure is irregular.

RADIX-2/8 Algorithm: Using the Radix-8 decomposition for the odd samples instead of Radix-4 decomposition gives the Split-Radix 2/8 Algorithm.

RADIX-2/4/8 Algorithm: Rearranging 8 samples together in a different manner gives the Radix-2/4/8 Algorithm.

2. IMPLEMENTATION

2.1 ARCHITECTURE IMPLEMENTED:

This architecture takes input as x0, x1, x2, x3, x4, x5, x6, x7 and these inputs are given to the first stage of the FFT architecture. Second stage inputs are taken from the first stage and we are getting second stage outputs X0, X 1, X2, X3, X4, X5, X6, X7. The system architecture is shown in Fig 1.

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

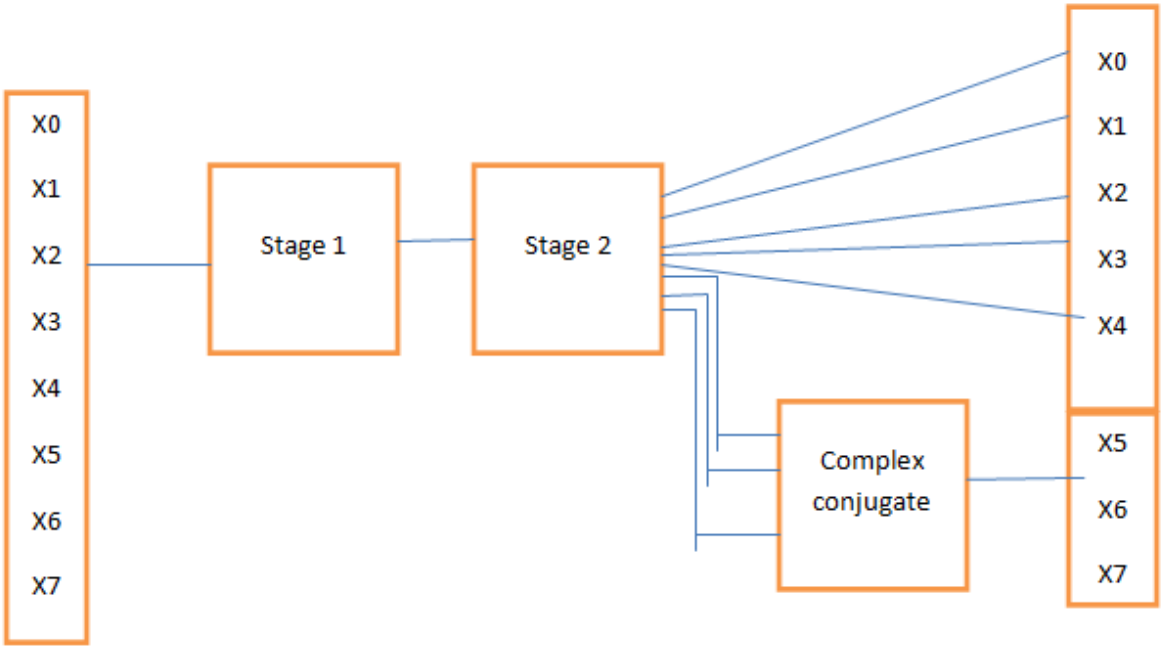

Figure 1: Architecture

The X3's complex conjugation is X5, X2's complex conjugation is X6 and X1's complex conjugation is X7.So this system simplifies the computation and reduce the computation time. Minimization of register is achieved by reducing the computation steps. This system reduces hardware complexity as well as power consumption and can be reduced up to 37% and 70%. The processor is compared to other implementations based on the area and power consumption. The results expose that our design achieves appropriate reduction in register as well as power. In order to get the register minimization it is necessary to take conjugate from some of the outputs. This conjugating technique uses a minimum registers comparing to the regular manipulations at the output stage. It reduces the overhead of reordering either the input or the output data. Hence no scrambling is required.

2.2 ALGORITHM IMPLEMENTED

Radix-2 is the first FFT algorithm. It was proposed by Cooley and Tukey in 1965.Though it is not the efficient algorithm, This algorithm is a special case in which N can be represented as a power of 2 .e., $N = 2^v$. This means that the number of complex additions and multiplications gets reduced to $N(N+6)/2$ and $N^2/2$ just by using the divide-and-conquer approach .When we also begin to use the symmetry and

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

periodicity property of the twiddle factor, it can be shown that the number of complex additions and multiplications can be reduced to N $\log_2 N$ and $(N/2)\log_2 N$ respectively. The entire process is divided into $\log_2 N$ stages and in each stage N/2 two-point DFTs are performed. The computation involving each pair of data is called a butterfly.Radix-2 algorithm can be implemented as: The algorithms appear either in

(A) **D**ecimation **in** **T**ime (DIT) Algorithm.

(B) **D**ecimation **in** **F**requency (DIF) Algorithm.

The radix-2 decimation-in-time and **decimation-in-frequency** fast Fourier transforms (FFTs) are the simplest **FFT algorithms**. Like all FFTs, they gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs.

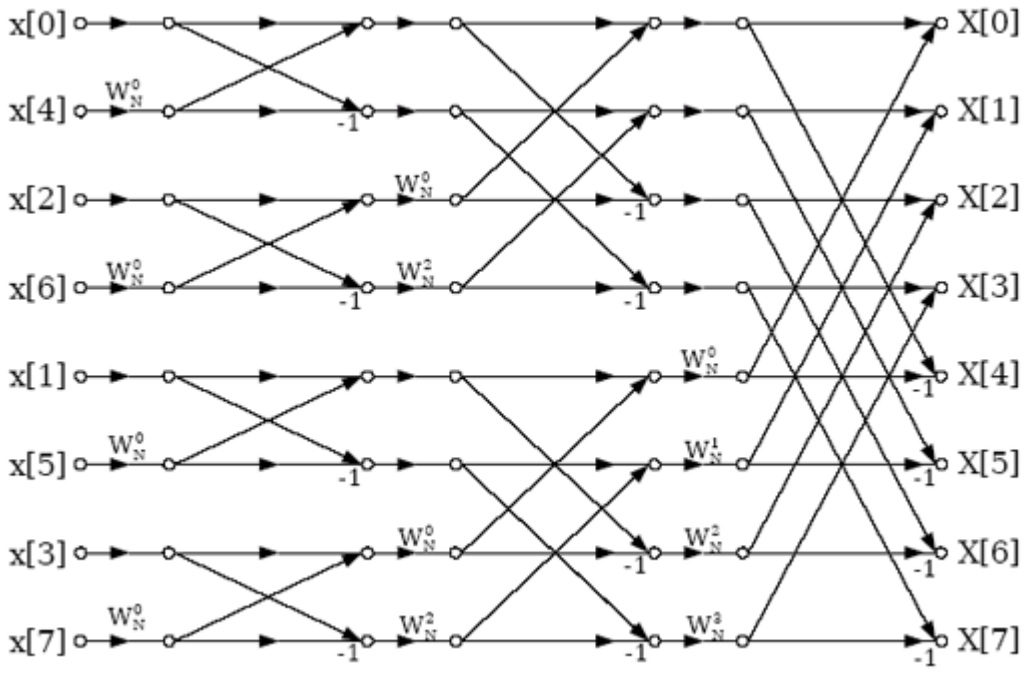

Figure 2: 8-pt radix-2 FFT using decimation-in-time FFT algorithm.

2.3 SOFTWARE IMPLEMETATION

VERILOG HD L: It w as introduced by Gateway Design Automation in 1984 as a proprietary hardware description and simulation language [6]. VERILOG synthesis tools can create logic-circuit structures directly from VERILOG

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

behavioral descriptions, and target them to a selected technology for realization. Using VERILOG, one can design, simulate, and synthesize anything form a simple combinational circuit to a complete microprocessor system on a chip.

The Radix 2 presented above has been fully coded in Verilog Hardware Description Language. The design is coded in VERILOG, the Xilinx ISE Design Suite 14.6 [8] and the ModelSim-Altera 6.4b (Quartus II 9.0) Starter Edition [7] that gives the synthesis and simulation report. The net-list can be downloaded into the FPGA using the same Xilinx tools and Texas Instruments prototyping board. From the architecture of Radix 2 in Fig 2, the butterfly blocks BF2I and BF2II are described as building blocks in VERILOG code. The FFT is heavily pipelined to achieve as highest clock frequency as possible. Twiddle factors are generated by an external program and embedded to the Verilog code.

2.4 FPGA IMPLEMENTATION

The proposed Montgomery multiplier is implemented by using Spartan-3E device xc3s100e-4cp132 and is fairly compared with the virtex-5 device 5vlx30ff324-3. The comparison between those two devices has been done based on the parameters like number of slices, number of IO's, number of bonded IOBs, number of slice flip-flops and time consumption.

2.5 SPARTAN 3E

The Spartan-3 family of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The eight-member family offers densities ranging from 50,000 to five million system gates. Spartan-3F FPGAs are ideally suited to a wide range of consumer electronics applications including broadband access, home networking, and display / projection and digital television equipment. The Spartan-3E family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs.

Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

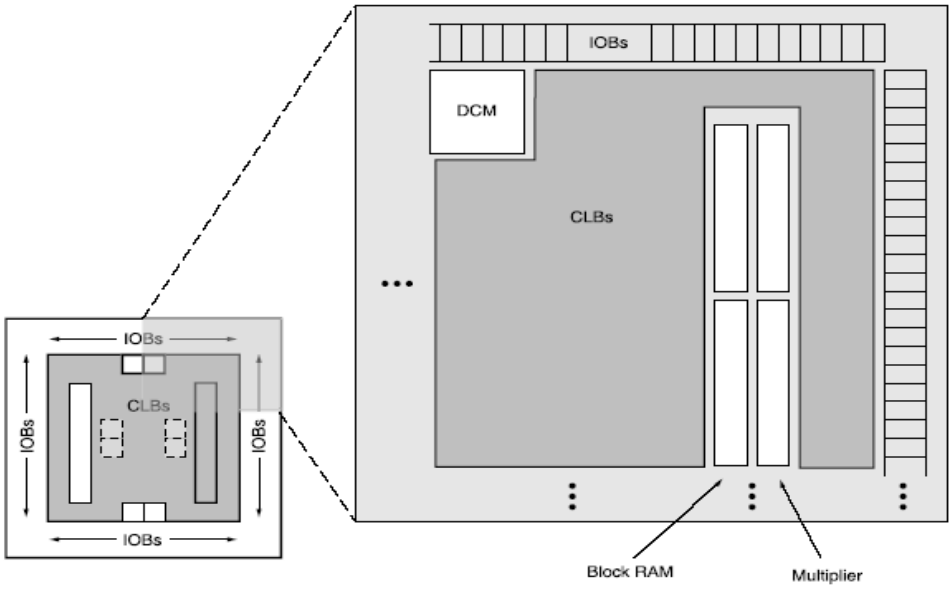

Figure 3: The Spartan-3E family architecture consists of five fundamental programmable functional elements

- **Configurable Logic Blocks (CLBs)** contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.

- **Input / Output Blocks (IOBs)** control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Supports a variety of signal standards, including four high-performance differential standards. Double Data-Rate (DDR) registers are included.

- **Block RAM** provides data storage in the form of 18-Kbit dual-port blocks.

- **Multiplier Blocks** accept two 18-bit binary numbers as inputs and calculate the product.

- **Digital Clock Manager (DCM)** Blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

3. SIMULATION AND RESULTS

The FFT processor was described with hardware description language Verilog and synthesized with XST tool in Xilinx ISE Design Suite 14.6 on Xilinx XC3S100e-4cp132FPGA chip, the high-performance signal processing applications chip with advanced serial connectivity, and simulated using ModelSim-Altera 6.4a (Quartus II 9.0)

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

Starter Edition. The top-level design shown in Fig. 6, the Xn is input data (8-bit real and 8-bit imaginary), Xk output data (8-bit real and 8-bit imaginary), synchronous reset; rst, clock, chip enable, start, busy, and finish. The results of the synthesis tool and the timing analysis using the ModelSim simulator indicate a maximum operating frequency of 40 MHz; this provides an execution time of 8 points transform in 249µS.Our Radix-2 processor achieves highest area and power consumption of all the processors implemented with Xilinx of Table I. The resulting figures show that our implementation outperforms the other implementations of that kind. The resulting speed nearly matches that of the Xilinx core but its throughput is more than 3 times higher due to its pipeline nature. The implementation results after implementing in Xilinx ISE Design Suite 14.6 are listed in Table I. shows the implementation results.

TABLE 1

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	299	1,920	15%	
Number of 4 input LUTs	118	1,920	6%	
Number of occupied Slices	216	960	22%	
Number of Slices containing only related logic	216	216	100%	
Number of Slices containing unrelated logic	0	216	0%	
Total Number of 4 input LUTs	132	1,920	6%	
Number used as logic	118			
Number used as a route-thru	14			
Number of bonded IOBs	17	83	20%	
Number of BUFGMUXs	2	24	8%	
Number of BSCANs	1	1	100%	

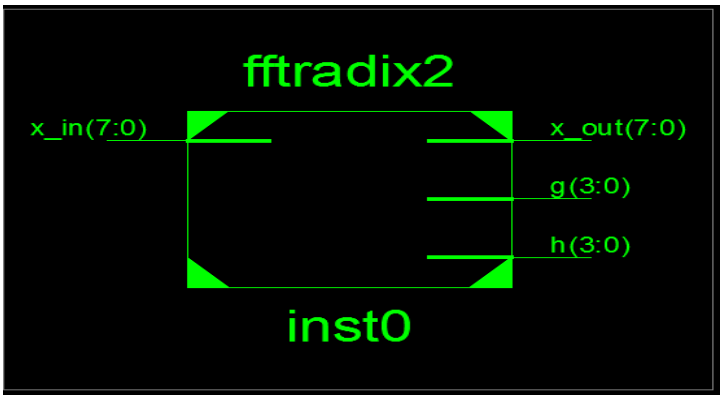

Figure 3: RTL Schematic of FFT

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

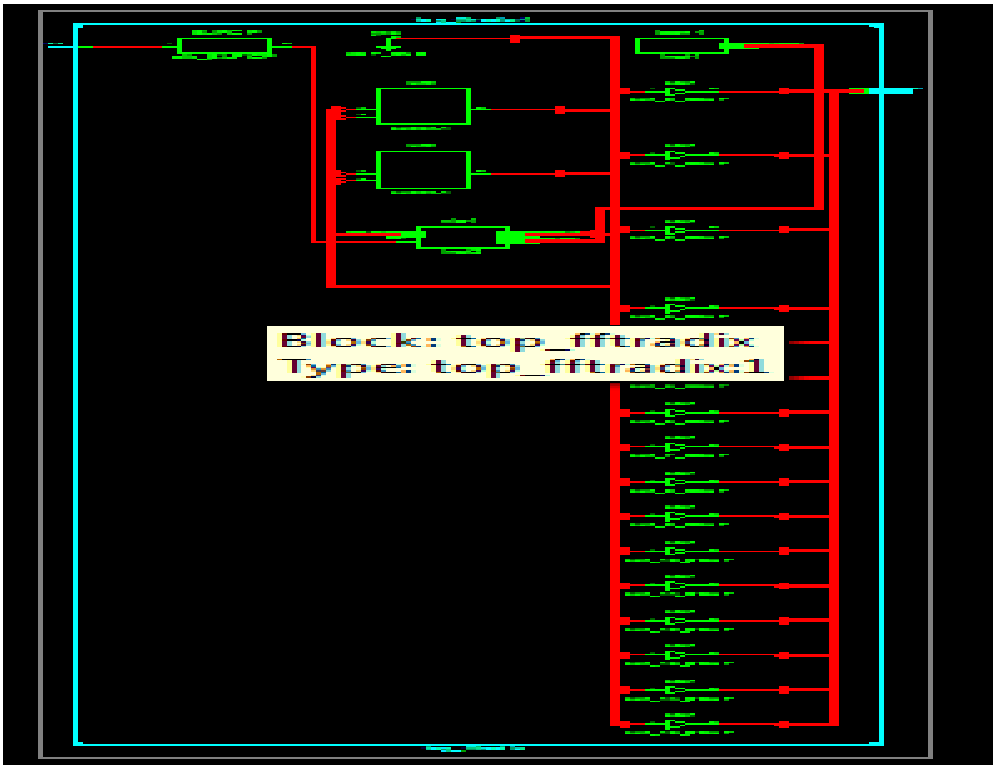

Figure 4: Technology Schematic

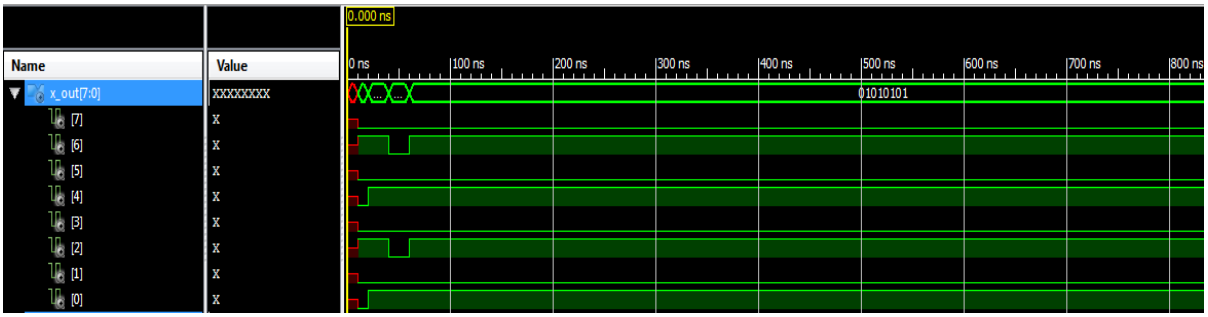

Figure 5: Output wave form

4. CONCLUSION

This FFT processor architecture optimized for speed of computation and area reduction has been designed. The algorithm used was a modified version of the DIF-FFT with the inputs and the outputs in natural order (not in bit reversed order). This design eliminates the need of scrambling the inputs and outputs. Although the processor designed is quite small and fast there are some improvements that can be made. Most of the cells used to build the FFT processor have been optimized for speed, area and power consumption. Implementing this technique of taking complex conjugate from some of the outputs recommended for higher point

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

FFTs. The power consumption can be reduced up to 70%.

REFERENCES

[1] Cortés, I. Vélez, I. Zalbide, A. Irizar, and J. F. Sevillano, "An FFT Core for DVB-T/DVB-H Receivers," VLSI Design, vol. 2008, Article.ID 610420, 9-pages , 2008.

[2] http://dangerousprototypes.com/docs/Xilinx_Spartan_3_FPGA_quick_start.

[3] http://cnx.org/content/m12016/latest

[4] A Guide to Digital Design and Synthesis, Second Edition by Samir Palnitkar.

[5] Verilog HDL Synthesis, "a practical premier", Third Edition by J. Bhasker.

[6] Ayinala and K. K. Parhi, "Parallel - Pipelined radix-22 FFT architecture for real valued signals", in Proc. Asilomar Conf. Signals, Syst. Comput., 2010, pp. 1274-1278