

VLSI Design, Implementation and Verification of Scalable FFT Processor

Mohammed Irfan M
M Tech, Department of Electronics and
Communication Engineering, BITM Ballari, India

Naseer Uddin
Assistant Professor, Department
Of Electronics and Communication Engineering,
BITM, Ballari, India

ABSTRACT

This paper presents design and implementation of a Pipelined FFT Architecture using Verilog HDL. The Pipelined Architecture is implemented using RS2DF (Radix-2 Single path Delay Feedback). Standard FPGA Flow is adapted to implement and was programmed on Spartan 3AN FPGA. Simulations and Synthesis are carried using Modelsim and Xilinx ISE. The Verilog Simulation results are compared with Inbuilt MATLAB FFT Core for verification of the design. The Speed achieved for this Core is 87.15 MHz

Keywords

FFT, Spartan 3, scalable architectures, FPGA, DFT

1. INTRODUCTION

The FFT processor can be designed by using radix 4 or radix 2. Radix 2 is more complex compared to radix 4, for reducing complexity greater radix calculations are used in implementation and it is utilized in many applications. By using FFT algorithm DFT complexity can reduce, also known as Cooley DIF algorithm. In this algorithm decay is fixed, input is shared repeatedly in branches N/r chain with length r this need $\log_r N$ stages. Equal hardware required for every stage, reading out information from memory and sending during processor, again storing in the same memory, this requires $\log_r N$ time performance for every pass. Radix r is using in this algorithm for reducing the passes during processor is done by increasing radix according to device cost. In many applications FPGAs are used for hardware operations. The FPGA's gives great achievement in less expense, adaptability; it takes less time to design. FPGA is more helpful in our project and in many more applications. Reconfigure the length of FFT by increasing or decreasing length from 16-bits to 1024-bits FFT using two architectures. Pipelined architecture and Memory based architecture. Pipelined architecture depends on radix-2 or radix-4 calculations. Radix-4 has less computational unpredictability estimated to radix-2 design. Significantly higher radix estimation can likewise be used for the usage to moderate the computational convolution in the architecture. Higher radix will bring down the computational issue. Tradeoffs for the decision of configuration are necessary in deciding length of FFT for the required application.

2. PIPELINED FFT ARCHITECTURES

This design is specific architecture for calculating DFT, using the algorithm called FFT; the sequence of data sending through this processor without any interruption and in real time. The calculation is done continuously with in time. This process achieved on a clock frequency, the advantages of this frequency is used in this project, to complete the processing with a lower clock for a low power result or to achieve high

speed processing. By using HDL, the FFT processor can scaled according to application requirement and it also parameterized. This design is flexible and commonly used to change the length and the architecture is characterized in two different types.

- (1) SDF [2] (Single Way Delay Feedback)
- (2) MDC (Multiple Way Delay Commutator)

The MDC and SDF are having special advantages, MDC having less delay components as compared to SDF. MDC have more advantages, like storage components and also used in MIMO. Refer figure 2.1, 2.2, 2.3, and 2.4 are different methods for designing pipeline processor. Where FIFO can identify by a symbol f , butterflies with 4 and 2 point can identify by a symbol BF4, BF2. Hence, reading twiddle factors and control is absent, the complexity in arithmetic and data operations.

2.1 Various proposals for pipeline FFT processor

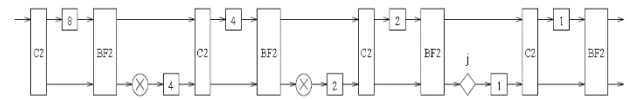


Figure 2.1: R2MDC (N=16)



Figure 2.2: R2SDF (N=16)

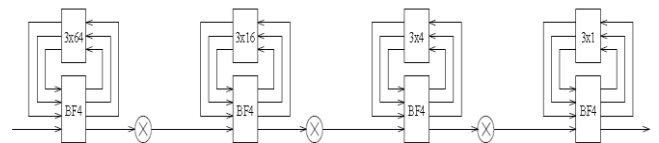


Figure 2.3: R4SDF (N=256)

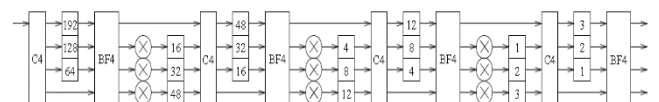


Figure 2.4: R4MDC (N=256)

3. BACKGROUND THEORY

In this subsection, a number of methods for calculating the DFT effectively. Analysis of DFT in different DSP

applications, for example correlation analysis etc, its effective calculation is subject that has reputable consideration by engineers researchers. Fundamentally, the computational issue for DFT is to build the string $\{X(k)\}$ of N complex numbers will obtained another string of information $\{x(n)\}$ of length N , as per the technique appeared in condition (1).

$$X[k] = \sum_{n=0}^{N-1} x(n) w_N^{kn} \quad 0 \leq k \leq N-1 \quad (1)$$

Where $w_N = e^{-j2\pi/N}$

3.1 Radix-2 FFT algorithms

Deriving FFT algorithms in this subsection by using divide and conquer method. Even and odd number data chain is derived by dividing the N Input points into sequence i.e. $F_2(k), F_1(k)$ resultant from $x(n)$ input sequence, refer figure 3.1 is 8-point DFT for calculating performs in 2-point block, 4 point block and 8point smaller DFT's. Implementing greater DFT's by using 8point smaller DFT's

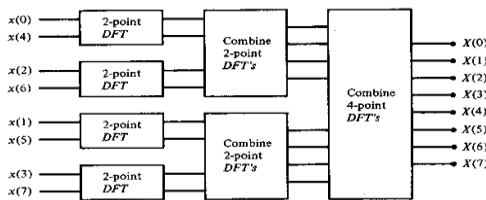


Figure 3.1 8-points DFT by using 3 stages.

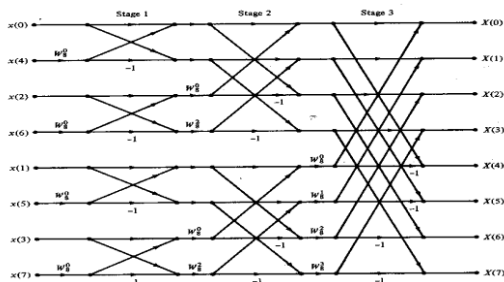


Figure 3.2 8-point decimation-in-time FFT algorithm.

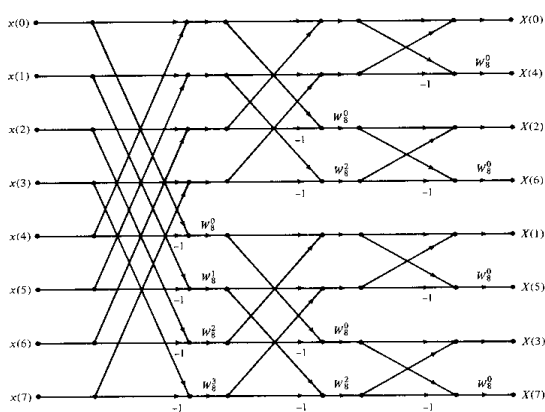


Figure 3.3 $N = 8$ -point decimation-in-frequency FFT algorithm.

3.2 Radix-4 FFT algorithms

When the number of data points N in the DFT is a power of 4 (i.e., $N = 4^v$), we can, of course, always use radix-2 algorithm for computation. However, in this case, it is more efficient to employ radix- r FFT algorithm. Refer figure 3.4a and 3.4b is radix-4 in more compact form. Note that each butterfly

involves three complex multiplications, since $w_N^0 = 1$, and 12 complex additions.

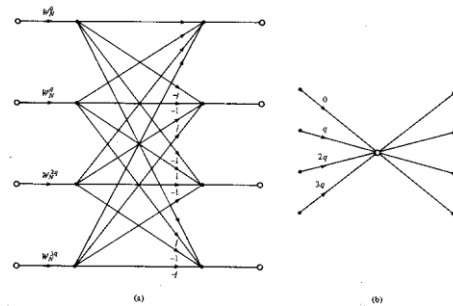


Figure 3.4 Basic butterfly computation in a radix-4 FFT algorithm.

4. SYSTEM DEVELOPMENT

Refer figure 4.1 is the design implementation of radix 2 single way delay feedback. This structure design uses feedback shift register to store up the output, multiplier and butterfly sections compare two R2MDC, and requirement of memory is minimal in this design. The DIT process is used to decompose the FFT algorithm in 2 approaches i.e. first half and second half approach, then output data $x(n)$ is branching into smaller subsections.

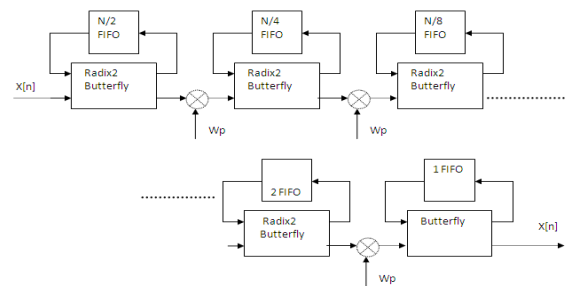


Figure 4.1: Radix 2 FFT Decimation in frequency Pipelined Architecture

4.1 Internal Architecture of each stage:

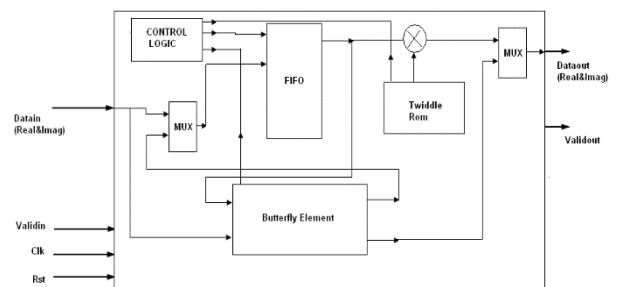


Figure 4.2: Internal Architecture of each stage of FFT

Refer figure 4.2 is 1 stage out of 10 stages for implementing 1K FFT processor. When the input i.e. Valid in is high the output of butterfly stores in FIFO block. Butterfly block in every stage has 3 control logic States. The imaginary and real data received by first stage, it stores into FIFO. This goes to next state when FIFO is full, reading and receiving the input information is taken place in this state. It generator two outputs by receiving two inputs, subtraction and addition operations is performed in second state.

FIFO storing the subtraction result and upcoming stage receives these addition results, incremented to 3rd state if

input information is received completely. This 3rd state used to send subtraction results for next stage which are previously stored in FIFO. Again it stores data in FIFO when valid in is high. Therefore, valid out and data out act as a input to next stage by connecting valid out signal to valid in and data out to data in. In this way 1K FFT processor is form by connecting 10 stages in series. The FFT processor can configure to any size according to our requirement, if we want 2K FFT processor one more stage has to connect in series after 10th stage. As shown in figure 4.3 is the State machine flow diagram.

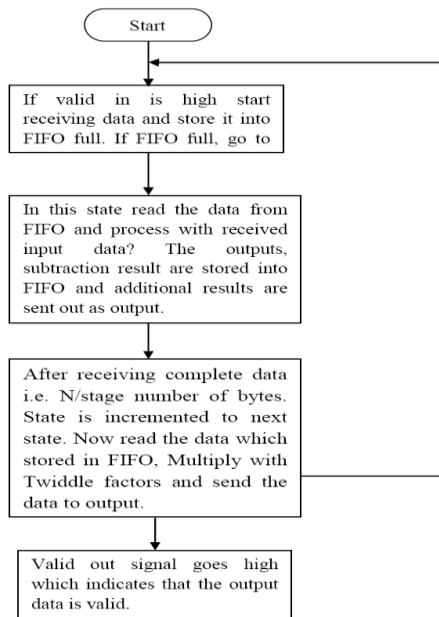


Figure 4.3 State machine flow diagram

4.2 Proposed Architecture

Refer figure 4.4 modification is done in generalized pipeline architecture to increase the performance of design. In this FIFO is modified for different stages of different length according to application requirement word length we are using 2 x 16 bits and 32 bits FIFO and storing N/2 points input to first FIFO. By this we can design FIFO with different size for N point FFT first stage is N/2 x 32 bits, second stage N/4 x 32 bits so on, therefore, different events write and read clock is taken.

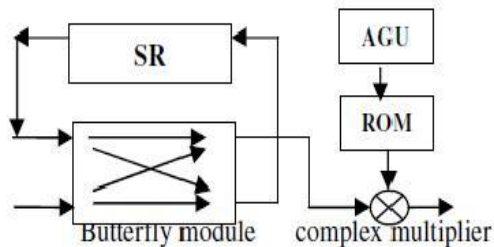


Figure 4.4: Proposed architecture for writing different output to FIFO

4.3 Field Programmable Gate Arrays

Standard FPGA Flow is adapted to implement this project. i.e right from specification to bit file generation, which is going to be programmed on Spartan 3AN FPGA.

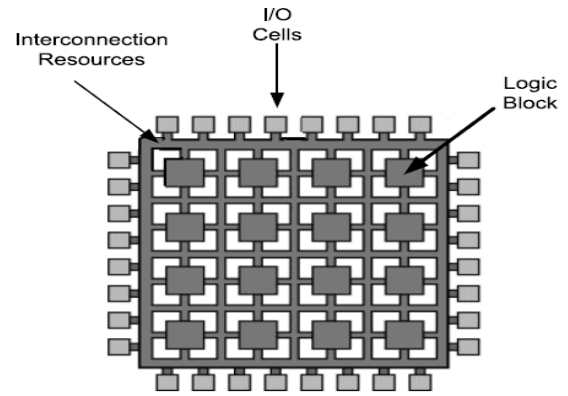


Figure 4.5: FPGA

5. TESTING AND VALIDATION

Below are the simulations Results for 8 point, FFT. As you can see in Figure 5.1 valid in is kept high and real in and image in inputs are given. Valid out goes high and we get real data out and image data out.

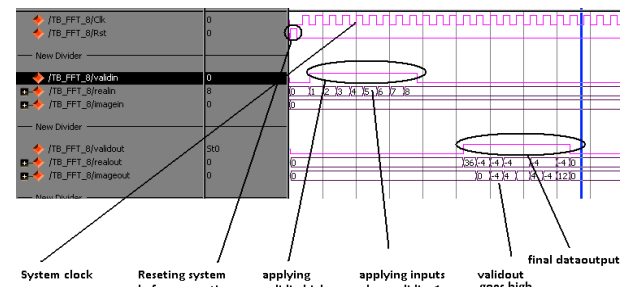


Figure 5.1: 8 point simulations

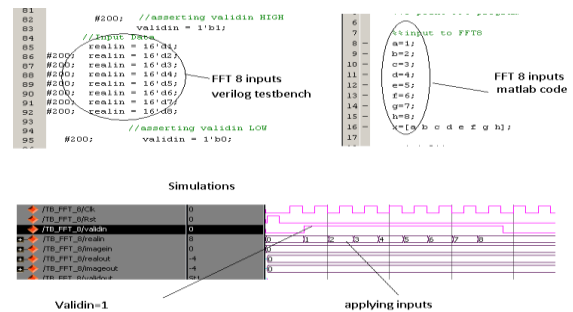


Figure 5.2 shows stage2 output comparison between Verilog and Matlab

5.1 FFT 1024k point simulations results

We have verified 8 point FFT it is working , we are sure that it will work for other points as well i.e. 16, 32, 64...1024. The reason being is that the basic building block of our architecture remains same for all stages. In order to verify for 1024 point, what we did was, we generated 2 frequencies i.e. 50 Hz and 120 Hz, Added up the two signals. This signal now acts as the input to 1024Point FFT refer figure 5.3.

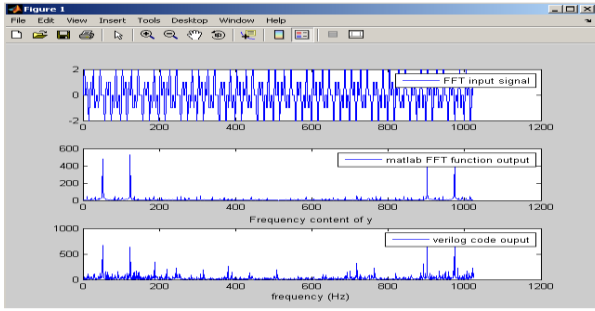


Figure 5.3 Frequency domain view of FFT input signal, MATLAB FFT function output and Verilog code output

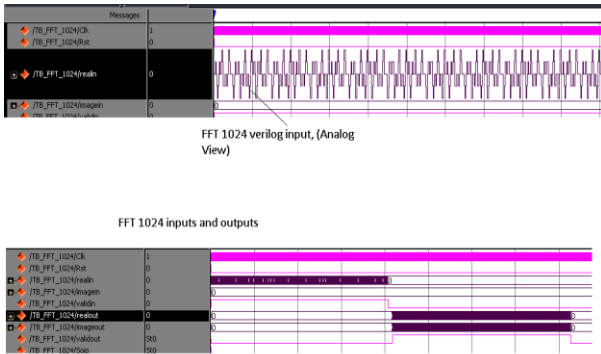


Figure 5.4: FFT 1024 verilog input and FFT 1024 points' inputs and output.

6. RESULTS AND DISCUSSIONS

Refer below is the power analysis report of 1024 point FFT architecture, this report contains on chip clocks, logic, signals, DSPs, IOs, and Leakage etc... totally 0.118 watts supply power is consumed. This power consumption is inclusive of dynamic power of 0.004W and quiescent power of 0.114W. Refer tables from 1 to 7 for the obtained area, power and delay synthesis reports .

6.1 Power Analysis Report

Table 1: Device

Device	
Family	Spartan 3adsp
Part	Xc3sd1800a
Package	Cs484
Temp Grade	Commercial
Process	Typical
Speed grade	-5

Table 2: power (W)

On-chip	Power(W)	Used	Available	Utilization (%)
Clocks	0.004	4	--	--
Logic	0.000	599	33280	2
Signals	0.000	1085	--	--
DSPs	0.000	8	84	10
Ios	0.000	73	309	24
Leakage	0.000			
Total	0.118			

Table 3: Supply summary

Supply summary		Total current(A)	Dynamic current(A)	Quiescent current(A)
Source	Voltage			
Vccint	1.200	0.046	0.003	0.042
Vccaux	2.500	0.025	0.000	0.025
Vcco25	2.500	0.000	0.000	0.000

Table 4: Thermal Properties

Thermal Properties	Effective TJA (C/W)	Max ambient (C)	Junction Temp (C)
	18.0	82.9	27.1

Table 5: Supply power (W)

Supply power(W)	Total	Dynamic	Quiescent
	0.118	0.004	0.114

Table 6: Characterization

Characterization
PRODUCTION v1.1.06-26-09

Table 7: Environment

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA(C/W)	NA
Airflow (LFM)	0

7. CONCLUSION AND FUTURE SCOPE

VLSI Design, Implementation and verification of scalable FFT processor, which contains multiplication complexity butSFG maintains radix 2 butterfly design. New radix 4 architecture is designed based on radix 2 algorithm. The requirement of hardware in our design is less comparing to other designs as, the memory, complex multipliers and adders estimate additionally the control complicationis scheduledfor distinguish. For simple understanding, this logarithm is utilized at whatever suitable. This proves it a perfect structural design for VLSI Design, Implementation and verification of scalable FFT processor and it verified by simulation.

Future work includes focusing on the implementation of scalable FFT processor and thus implementing the model practically and calculating the performance by applying the technique in real time and also considering all the practical parameters. The research is going on in 5G and IP cores, thisused for specific application (ASIC)like communication systembasedfor reducing computation time,multistandard audio/videochannelization and so advance.

8. REFERENCES

- [1] ShouSheng and MatSTorKelson. A New Approach to Pipeline FFT Processor, IEEE proceedings of IPPS 1996.
- [2] YoUng W.K.W, SwartZlander E.E and JoSepH S J. Radix 4 Delay Commutator For FFT Processor Implementation, IEEE, 1984.
- [3] S.K. SriVatsa And M. KaNnanLow Power Hardware Implementation of High Speed FFT Core, IEEE, 2010.
- [4] A.T. ErdoGan T. ArslUn, WeiHUn, M. HaSan. The

- Development Of High Performance FFT IP Cores Through Hybrid Low Power Algorithmic Methodology, IEEE. Compute, 2005.
- [5] ZE KeWaA, XuE Liu, FeNg YU. Pipelined Architecture for Normal Input /Output Order FFT, JZUS, C-2011.
- [6] A.M. DeSpain and E.H. WOld. Pipeline and Parallel Pipeline FFT Processors For VLSI Implementation, IEEE, 1984.
- [7] DavIdHwAng, YinGningPeNg, Pipeline FFT Architectures Optimized For FPGA, International Journal, Volume 2009 (2009),
- [8] A.M. DeSpain, Very Fast Fourier Transform Algorithms Hardware For Implementation, IEEE, MAY 1979, c 28(5):333–341.
- [9] VeNuGoPal B, VasAnthaSudHeer N, FPGA Implementation Of 64 Point FFT Processor, IJIT and Exploring Engineering ISSN: 2278-3075, SEPTEMBER 2012.