Design and Implementation of Real-Time 16-bit Fast Fourier Transform using Numerically Controlled Oscillator and Radix-4 DITFFT Algorithm in VHDL

Priyanka Pateriya, Ankeet Chaora

Priyanka Pateriya, priyankapateriya03@gmail.com ,M.TECH Scholar, RCET Bhilai (C.G)
Ankeet Chaora, ankeet.chaora@rungta.ac.in, Assistant Professor, RCET Bhilai(C.G)
Dept. of Electronics & Telecommunication Engineering, Rungta College of Engineering & Technology, Bhilai, Chhattisgarh, India

---***---

Abstract - *This paper addresses the real-time demand of 16-bit FFT processor exploitation Numerically Controlled Oscillator (NCO) and devoted radix-4 Decimation-In-Time FFT in modern digital signal process domain. NCO is basically a discrete signal generator that produce discrete time, discrete worth illustration of sine or cosine wave relative to amendment in its input frequency control word. Radix-4 DITFFT recursively partitions a DFT into 4 quarter-length DFT of gathering of every fourth time sample. The output of these shorter FFT's are reused to work out the FFT, consequently enormously decreasing the aggregate computational expense. FFT Pipeline architecture is used for greater data preciseness (or exactitude) and dynamic range with high throughput. The result show that the design is area-power-time efficient for processing of 16-bit FFT. Execution of 16-bit FFT is synthesize in VHDL utilizing ISE tool of Xilinx 14.3. Finally, the simulation results in ModelSim ALTERA 10.4b shows that the FFT calculation of 16–bit complex input data is perform with success with lesser time and fewer space demand. This paper performs the real time high-speed digital signal processing requirements.*

Key Words: **DITFFT, Radix-4, NCO, VHDL, Pipelining.**

1.INTRODUCTION

Fast Fourier Transform (FFT) is an effective and quick Discrete Fourier Transform (DFT) algorithm, which is the core of digital signal processing algorithms. FFT is widely used in radar, communications, image processing, signal detection and other fields, most of those fields necessitate the FFT processor with high-speed and high precision real-time processing performance. A Fast Fourier transform (FFT) computes the discrete Fourier transform (DFT) of a sequence, or its inverse. Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and Inverse FFT performs the other way around. A FFT rapidly computes such transformations and thereby reduces the complexity of computing DFT from O(n^2) to O(n log n) where n is data size. FFT is widely used in digital signal process fields for prime speed and high preciseness real-time operation performance. The DFT differs from the DTFT in that its input and output sequences are each finite, it is therefore same to be the analysis of finite-domain discrete-time functions. Let x(n) be a finite duration sequence. The N-point DFT of the sequence x(n) is expressed by-

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-2\pi jnk}{N}} , \qquad k=0,1,..,N-1 \qquad \text{……….(1)}$$

And the corresponding IDFT is given as-

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{\frac{2\pi jnk}{N}} , \qquad n = 0,1...,N-1 \qquad \text{………..(2)}$$

Where, W_N be the complex-valued phase factor or rotational factor, which is Nth root of unity expressed by $W^{kn} = e^{-2j\pi kn/N}$.

By adopting divide & conquer approach, a computationally efficient algorithm for the DFT can be developed. This approach depends on decomposition of N-point DFT into successively small size DFTs. If N is factorized as N = $r_1,r_2,………,r_L$ where $r_1, r_2, ……,r_L=$ r, then N= r^L. Hence, the DFT will be of size "r", where this number "r" is called the radix of the FFT algorithm.

2. METHODOLOGY

2.1). Numerical Controlled Oscillator :-

In the digital signal processing, the NCO has wide range of applications. A numerically controlled oscillator (NCO) is a discrete signal generator that creates a synchronous(i.e., clocked), discrete-time, discrete-valued representation of a waveform, usually sinusoidal. NCO generates real or complex signal with freelance (i.e., independent) frequency and phase in each output. NCO are extensively used in the generation of sinusoidal waveform in DSP. NCO is a system that produces discrete approximation to sin or cos wave relative to vary in its input.

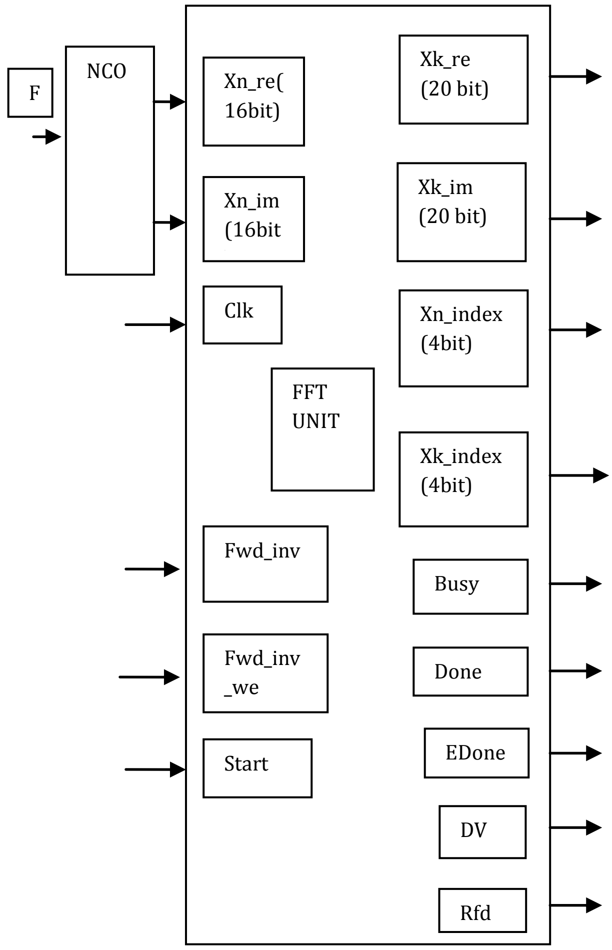


Fig-1: Proposed Methodology

We characterize discrete series of sine signal as:-
S(n)= sin($2\pi fnT_s$)
Where f denotes signal frequency and T_s denotes the sample interval. Its corresponding discrete phase series is ϕ(n) = $2\pi fnT_s$ where Linear phase increment Δϕ(n) = $2\pi fnT_s$ is a constant and is simply coupled to frequency f, that means when phase increment Δϕ(n) is fastened, accordingly frequency can be fastened. This is the start line of NCO technique. The phase that a cycle of sin signal expertise is 2π. If 2π is uniformly quantized into M equal elements, taking k equal parts of the part increment corresponding to frequency f, there will be -

$$\frac{f}{fs} = \frac{\Delta\phi(n)}{2\prod} = \frac{k}{M}$$

That is to mention, sine signal with frequency f is sampled by sampling frequency fs, its phase increment of adjacent sample point is a constant k. therefore dynamic k is to amendment frequency. Captions:-
F – Frequency word
A – Xn_real (16 bit)
B – Xn_img(16 bit)

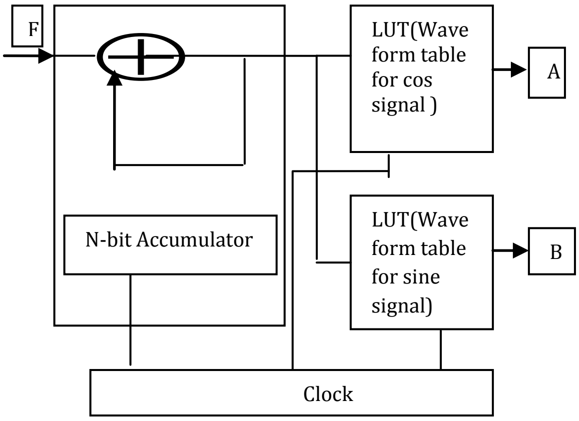

F
N-bit Accumulator
LUT(Wave form table for cos signal)
A
LUT(Wave form table for sine signal)
B
Clock

Fig-2: Block diagram of NCO

FCW produce quantized phase sequence by timepiece. This procedure is typically completed by N-bit accumulator. Phase accumulator is primilarly a N-bit adder with feedback. It can unendingly add the output data to frequency word to produce regular N-bit phase address code. When discrete phase address sequence is input into the LUT, the discrete amplitude sequence of the corresponding circular functions like sine or cosine signal will be produced.
The data in sine or cosine LUT is stored as in the equation –

$$\text{Sin(n) = sin(}\frac{\phi(n)}{2^M}\times\frac{\pi}{4}\text{)} \quad ....(3)$$

$$\text{Cos(n) = cos(}\frac{\phi(n)}{2^M}\times\frac{\pi}{4}\text{)} \quad ....(4)$$

Where, M= Number of higher order bits represent the phase address code, obtained by truncating N number of bits.
And $\pi/4$ is used within the equation because, as analyzing the sine and cosine waveform, the sine samples from 0 to $\pi/4$ is equal to cosine samples from $\pi/4$ to $\pi/2$ and vice versa.
In this project, output discrete sine or cosine signal from waveform ROM or LUT is given as input signal to FFT processor. Discrete sine signal is given to imaginary part of data whereas discrete cosine signal is given to real part of data in FFT processor as shown in figure 3 and figure 4.

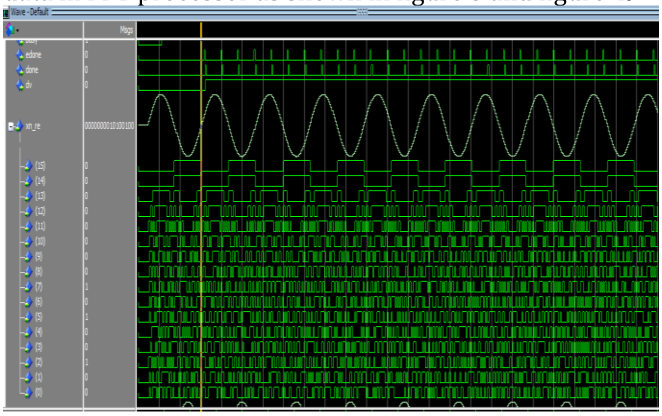


Fig-3: Discrete Cosine signal from NCO output to 16-bit FFT real input

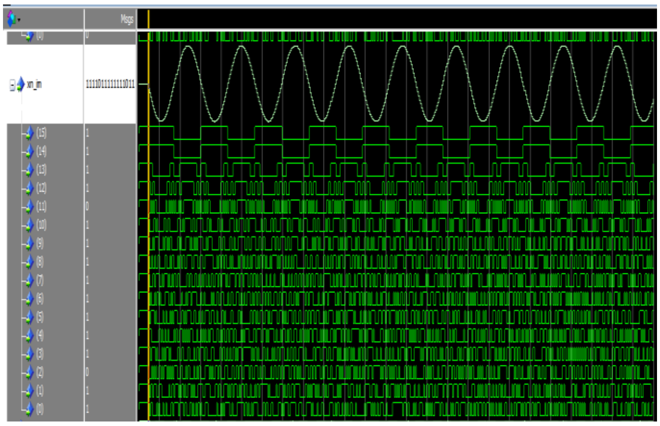


Fig-4: Discrete Sine signal from NCO output to 16-bit FFT imaginary input

2.2). Radix-4 DITFFT Algorithm:-

The decimation-in-time (DIT) radix-4 FFT recursively partitions a DFT into four quarter-length DFTs of groups of each fourth time sample. The outputs of these shorter FFTs are reused to compute several outputs, subsequently significantly lessening the aggregate computational expense (i.e.,total computational cost). The radix-4 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into four elements. The DFT sums over all groups of the every fourth discrete-time indexn=[0,4,8,…,N−4],n=[1,5,9,…,N−3],n=[2,6,10……,N-2]. (This works out only once the FFT length could be a multiple of four as shown in figure(5). The given sequence x(n) is decimated into 4 sequences of length N/4 each, a four-way split is used. The N point input sequence is split into four subsequences, x(4n),x(4n+1), x(4n+2), and x(4n+3), where n=0,1,…N/4-1. Then N=4^n

X(k)=

$$\sum_{n=0}^{\frac{N}{4}-1} x(4n)W_N^{4nk} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)W_N^{(4n+1)k} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+2)W_N^{(4n+2)k} + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)W_N^{(4n+3)k} \quad ....(5)$$

Radix-4 uses log$_4$N stages, where each stage has N/4 butterflies. N/4 butterfly involves in each stage and number of stage is log$_4$N for N-point sequence. Therefore, the number of complex multiplications is 3N/4 log$_4$N and number of complex additions is 12N/4 log$_4$N. Eq(5)is written as below:-

$$X(k)= X_1(k) +W_N^{k} X_2(k) + W_N^{2k} X_3(k) +W_N^{3k} X_4(k) \quad …….(6)$$

The 16-bit FFT calculation performs in the high pipelined mode which implies in each clock cycle, one complex information (or data) is read from input buffer and complex result's is written in output buffer. It is partitioned into a few phases to perform 16-bit FFT count in pipelined mode.

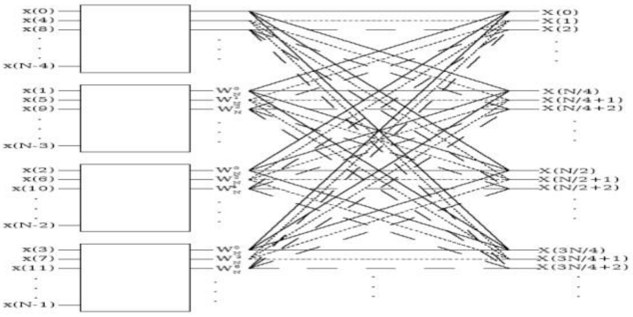


Fig-5: Radix-4 DITFFT Structure

3. RESULT AND CONCLUSIONS

3.1) RESULT:-

3.1.1). Performance Measurement :-This paper uses the VHDL to design. RTL view of 16-bit FFT processor designed in XILINX ISE 14.3 tool is shown in figure (6).). The time consumption for 16-bit FFT calculation is low as 9.576ns as shown in report figure (7). The design outline (i.e.summary) report is given in figure (8),which shows the slice consumption. The total power requirement for scheming or calculating 16-bit FFT is 0.140 Watt as shown in figure (9). And finally the area report is shown in figure (10). Compared the result with the existing system, this paper provides the time consumption to calculate 16-bit FFT using radix-4 DITFFT.

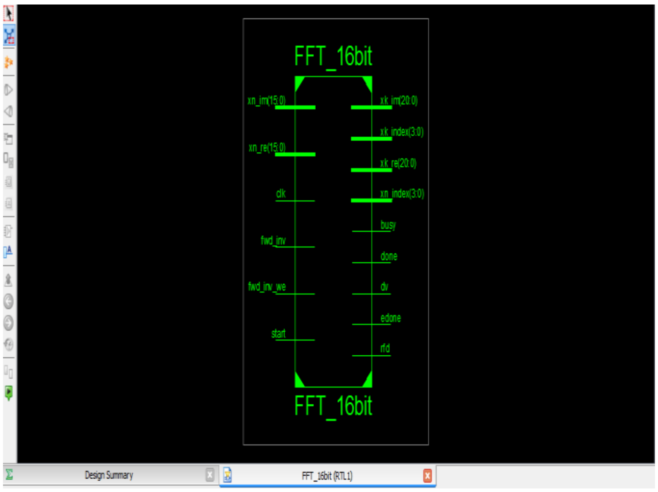
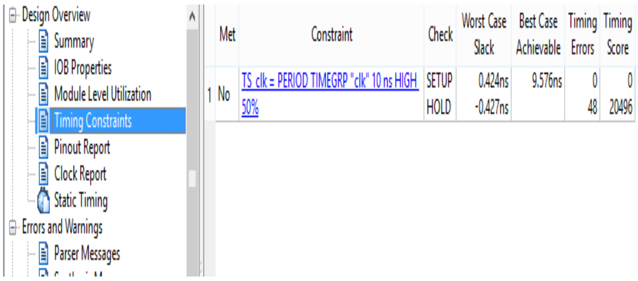
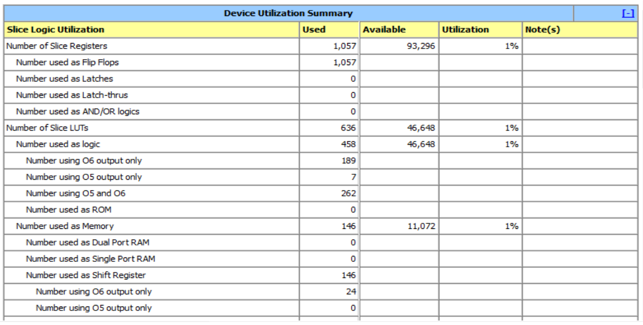


Fig-6: RTL View of 16-bit FFT Processor in XILINX ISE 14.3.



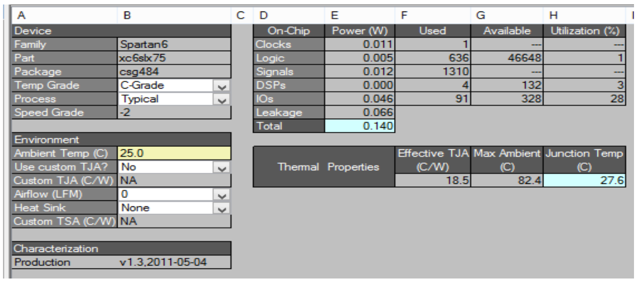
	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	No	TS clk = PERIOD TIMEGRP "clk" 10 ns HIGH 50%	SETUP	0.424ns	9.576ns	0	0
			HOLD	-0.427ns		48	20496

Fig-7: Timing Report of 16-bit FFT in XilinxISE 14.3



Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	1,057	93,296	1%	
Number used as Flip Flops	1,057			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	636	46,648	1%	
Number used as logic	458	46,648	1%	
Number using O6 output only	189			
Number using O5 output only	7			
Number using O5 and O6	262			
Number used as ROM	0			
Number used as Memory	146	11,072	1%	
Number used as Dual Port RAM	0			
Number used as Single Port RAM	0			
Number used as Shift Register	146			
Number using O6 output only	24			
Number using O5 output only	0			

Fig-8: Design Summary Report of 16-bit FFT Processor

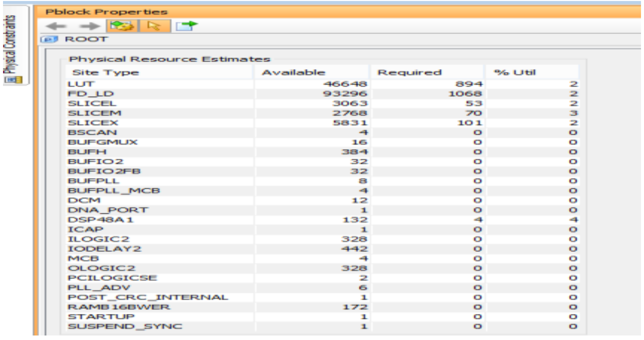


A	B
Device	
Family	Spartan6
Part	xc6slx75
Package	csg484
Temp Grade	C-Grade
Process	Typical
Speed Grade	-2
Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Heat Sink	None
Custom TSA (C/W)	NA
Characterization	
Production	v1.3,2011-05-04

D	E	F	G	H
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.011	1	---	---
Logic	0.005	636	46648	1
Signals	0.012	1310	---	---
DSPs	0.000	4	132	3
IOs	0.046	91	328	28
Leakage	0.066			
Total	0.140			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	18.5	82.4	27.6

Fig-9: Power Requirement Report of 16-bit FFTProcessor

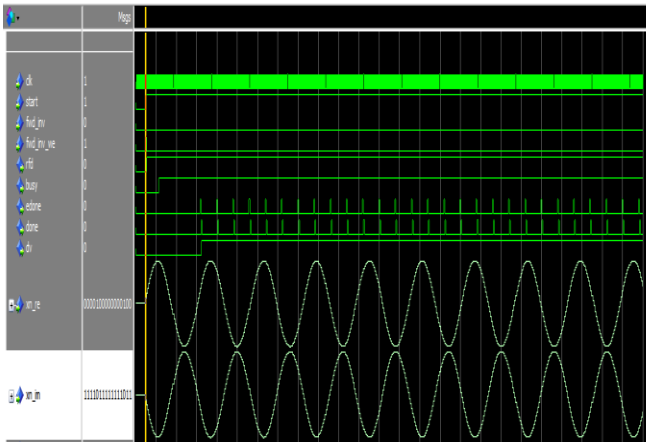


Physical Resource Estimates

Site Type	Available	Required	% Util
LUT	46648	894	2
FD_LD	93296	1068	2
SLICEL	3063	53	2
SLICEM	2768	70	3
SLICEX	5831	101	2
BSCAN	4	0	0
BUFGMUX	16	0	0
BUFH	384	0	0
BUFIO2	32	0	0
BUFIO2FB	32	0	0
BUFPLL	8	0	0
BUFPLL_MCB	4	0	0
DCM	12	0	0
DNA_PORT	1	0	0
DSP48A1	132	4	4
ICAP	1	0	0
ILOGIC2	328	0	0
IODELAY2	442	0	0
MCB	4	0	0
OLOGIC2	328	0	0
PCILOGICSE	2	0	0
PLL_ADV	6	0	0
POST_CRC_INTERNAL	1	0	0
RAMB16BWER	172	0	0
STARTUP	1	0	0
SUSPEND_SYNC	1	0	0

Fig-10: Area Requirement Report of 16-bit FFT Processor in Xilinx ISE 14.3

3.1.2). Simulation Result :-The test-bench of the FFT system with NCO is simulated in Modelsim Altera SE 10.4b as shown in figure(11). The system designs in pipeline and FFT output for different 16-bit input data occurs continously which meets high speed real-time DSP.



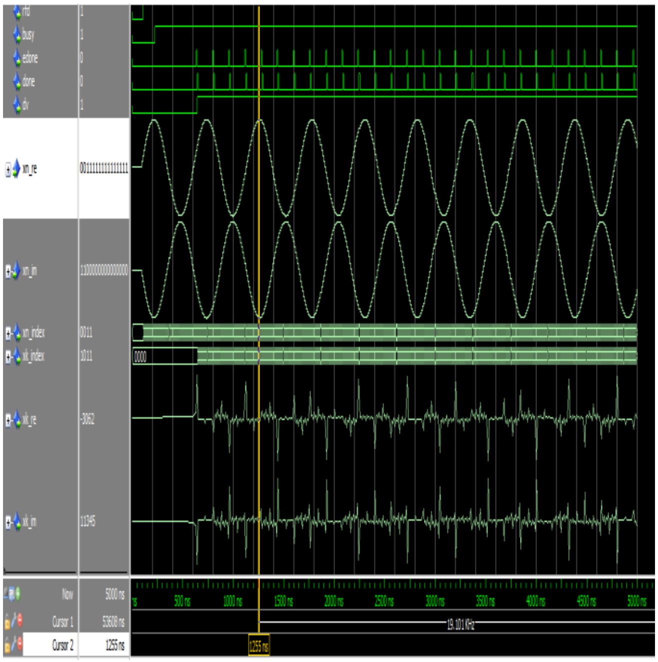


Fig-11a&b: Simulation result of 16-bit FFT Calculation in ModelSim ALTERA 10.4b

3.2).CONCLUSION

This paper analyses the usage of NCO for FFT input real and imaginary discrete signal, which provides the benefits of high frequency stability, high reliability and accuracy of discrete cosine & sine waveform and fine frequency resolution. With the utilization of feature of the radix-4 DITFFT algorithm for 16-bit FFT with pipeline design reduces the consumption of power and time requirement. The system introduces pipeline to get a high execution rate and accurate outturn. The implemented FFT processor meets the high-speed real-time digital signal processing necessities.

Future research work shall embody implementation of higher-point radix FFT processor and evaluating alternative efficient design with the projected architecture with low power and time techniques with energy-efficiency estimation to behave as high-level performance model of Fast-Fourier Transform.

ACKNOWLEDGEMENT

The author wishes to thanks professor Ankeet Chaora and professor Nitin Naiyyar who provided technical support and major guidance in this project and also guiding force behind this work. The author greatly indebted to them for their constant encouragement, invaluable advice. Author consider it as good fortune to have got an opportunity to work with such a wonderful person.

REFERENCES

[1] Ll Xiao-feng, Chen Long, Wang Shihu,"The Implementation of High-speed FFT processor based on Field Programmable Gate Array" Aug 2010 IEEE Publication (Conference on Computer, control &Electronics Engineering) 978-1-4244-7956-6/10.

[2] P.Augusta Sophy, R.Srinivasan, J.Raja, M.Avinash, "Analysis and Design of Low Power Radix-4 FFT Processor using Pipelined Architecture" 2015 *2* IEEE Publication (Conference on Computing and Communications Technologies) 978-1-4799-7623-2/15.

[3] Xiaobo Zou, Yunxue Liu, Yuhui Zhang "FPGA Implementation of Full Parallel and Pipelined FFT" Sept 2012 IEEE Publication (Conference on Wireless Communication) 978-1-61284-683-5/12.

[4] Xin Xiao, Erdal Oruklu and Jafar Saniie Xin Xiao, Erdal Oruklu and Jafar Saniie "Fast Memory Addressing Scheme for Radix-4 FFT Implementation" Jun 2009 IEEE Publication (Conference on Electronics/IT) 978-1-4244-3355-1/09.

[5] Yousri Ouerhani, Maher Jridi"Implementation techniques of high-order FFT into low-cost FPGA" Aug 2011 IEEE Publication (Conference on Circuits& system) 978-1-61284-857-0/11 .

[6] Ren Chen, Hoang Le, and Viktor K. Prasanna "Energy Efficient Parameterized FFT Architecture" IEEE Publication (Conference on Field PLA) 978-1-4799-0004-6/13.

[7] S. Suruthi and M. Arulkumar "Pipelined R22 SDF, R4SDC FFT Architecture Via Folding Transformation" IEEE Publication (Conference on Communication /signal processing) 978-1-4673-4866-9/13.

[8] Swapnil Badar & D.R.Dandekar "High Speed FFT Processor Design using Radix -24 Pipelined Architecture" IEEE Publication (Conference on Instrumentation control) 978-1-4799-7165-7/15.

[9] Mohammad Arif "A novel approach for DFT computation" IEEE Publication (Conference on Circuit, Power and Computing Technologies [ICCPCT]) 978-1-4799-7075-9/15.

[10] Wen-Chang Yeh and Chein-Wei Jen "High Throughput Feed Forward Pipelined Parallel Architecture for FFT and IFFT" IEEE Publication (Conference on Embedded & Communication system) 978-1-4799-6818-3/15 .

[11] Saikat Kumar Shome\ Abhinav Ahesh2, Durgesh Kr Gupta "Architectural Design of Highly Programmable Radix-2 FFT Processor with Efficient Addressing Logic" IEEE Publication (Conference on Circuit &system) 463-1-2744-2/12.

[12] HeJin\He SongBai "Design and Realization of NCO of Modulation Based on FPGA" July 2007 IEEE Publication (Conference on Communications, Circuits and System) 978-1-4244-1473-4.

[13] Gai Peng Ao "The Basic Principle and FPGA Implementation of NCO" 2012 IEEE Publication (International Conference on Instrumentation & Measurement, Computer, Communication and Control) 978-0-7695-4935-4.