FPGA Implementation of Radix-2 based FFT Architecture for Real-Valued signal processing

D.Dinesh kumar, R.Varun prakash, G.Kirubakaran

Abstract: ***This paper investigates the various pipelined FFT architectures based on radix-2, radix-2^2 & radix-2^3 algorithms. The implemented FFTs are designed by employing techniques such as folded transform and register minimization. It maximizes the utilization of hardware resource and reduces the number of adders. It requires less area and achieves high throughput and low latency. For higher values of N, the FFT (Fast Fourier Transform) architecture has many butterfly structures which has been optimized. The FFT outputs are usually obtained in a bit reversed order and a new approach for reordering the bit-reversed orders has been proposed.***

Index Terms: Bit-reversed order, FFT, Folding Transformation, Redundancy, RFFT.

I. INTRODUCTION

Fast Fourier Transforms are extensively used in Discrete Fourier Transform in wireless networks, Asymmetric Digital Subscriber Line (ADSL), Digital Video Broadcasting, signal processing [1]. Fewer number of computations are required for FFT than that of the direct evolution of DFT (Discrete Fourier Transform) [13]. A novel single path delay commutator is proposed along with a pipelined FFT architecture to produce the output samples in normal order with less implementation complexity and reduction in the adders [2]. Based on the decimation process, FFTs are classified into two categories: Decimation in Time (DIT-FFT) and Decimation in Frequency (DIF-FFT).
There are three types of pipelined FFT architectures such as Single-Path delay feedback (SDF), Single-Path delay commutator(SDC), Multi-Path delay commutator(MDC). The pipelined FFT is categorized with real-time processor as the data sequence passing the processor. MDC architecture process multiple input streams as it has high throughput rate. SDF Architecture process single input data stream as the memory size is minimum and the multipliers are fully utilized whereas the adder utilization is low [2]. SDC architecture process single input streams, because it uses more memory resources than SDF and the control is complicated. In pipelined architecture, each stage of FFT uses separate arithmetic unit. This approach increases the throughput.
The RFFT (Real-Valued Fast Fourier Transform) play a vital role in various real-time applications such as Electroencephalography (EEG), Electrocardiography (ECG), the power spectral density (PSD) for the real-valued signals can be estimated [3]. The 4-parallel architecture has the amount of latency less compared to that of the 2-parallel architecture. Among the two scheduling algorithms, one has less complexity while the other has few delay elements. Since the two input samples can be processed parallel the frequency is reduced by 2. The FFT outputs are obtained in a bit reversed order. Circuit to reorder these bit reversed output sequences has been presented [4].
The optimized architecture for radix-2 in 32 point FFT architecture has been implemented in virtex 6 and the synthesis was done in ISE Design Suite 14.7 [5]. The low hardware complexity has been achieved and in In-place FFT structures which has larger butterfly blocks are the more efficient one based on area-time complexity and energy in consumption. In order to achieve high throughput, the in-place RFFT architecture could be scaled efficiently [6].
In order to produce the real and imaginary parts in parallel, some designs use reordering registers which leads to increase in register count. Here the scheduling structures using feedback are introduced to minimize the register count to half compared to previous described designs [7]. The multi path delay commutator has been used for processing the FFT computation for two independent data streams. Here the bit reversal implementation is performed by the architecture itself by reusing shift registers by interleaving of data [8].
This paper briefly describes about the 2-parallel 16-point architecture 1, 2-parallel 16-point architecture 2, 4-parallel 16-point architecture and 4-parallel 64-point pipelined architectures based on radix-2, radix-2^2 and radix-2^3 algorithm. The paper describes the following sections as RFFT in Section II and Feed-forward architecture in Section III and reordering the output samples in Section IV and implementation and design in Section V and the paper is concluded in Section VI.

II. REAL-VALUED FAST FOURIER TRANSFORM

The N-point DFT sequence of x[n] is

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$$

where

$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author
D.Dinesh kumar*, Dept. of ECE, Mepco Schlenk Engineering College, Sivakasi, India.
R.Varun prakash, Dept. of ECE, Mepco Schlenk Engineering College, Sivakasi, India.
G.Kirubakaran, Dept. of ECE, Mepco Schlenk Engineering College, Sivakasi, India.

The input sequence in RFFT is assumed to be in real i.e., $\forall n, x[n] \in R$.If x[n] is the rea value, then the FFT output X[k] could be conjugate symmetric, i.e., X[N-k] = X*[k]

By using the above property, (N/2)-1 outputs can be removed. Since the outputs are real, the imaginary part i.e., Im(x[n]) = 0. Fig 1 represent the flow graph of N=16 point DIF FFT and the terms represented in the box are redundant terms. These redundant terms are removed. The input and output numbers in the graph are represented by indices of input and output samples respectively [9]. The DIF (Decimation in Frequency) FFT sequence breaks into smaller and smaller sequence. The number of samples and the sequence is represented as N and x[n] respectively

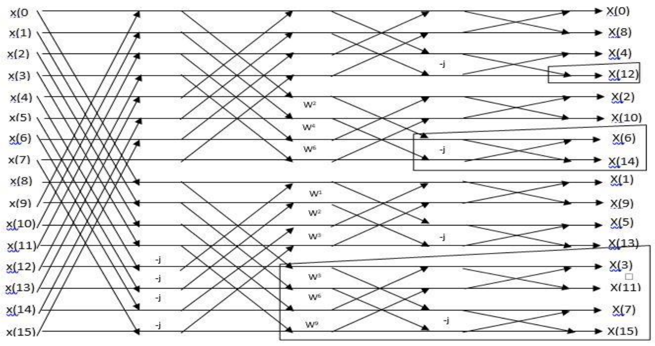


Fig.1. Flow graph for 16-point DIF FFT.

III. FEED-FORWARD FFT ARCHITECTURE

The radix-2^2 DIF RFFT algorithm has simple hardware structure due to direct mapping. Different butterfly structures are mapped to the nodes in the DFG (Data Flow Graph). Two different architectures are used for two different scheduling methods.

2-Parallel Radix-2^2 Architecture 1

The architecture for 2-parallel pipelined 16-point DIF FFT computation is shown in Fig 2.

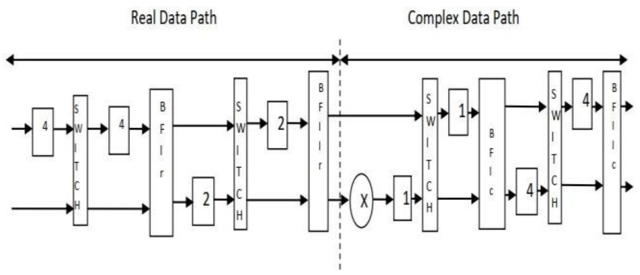


Fig.2. Architecture 1 for 16-point DIF FFT computation.

The first two stage butterfly has only real adders. According to the input data it has the following order: even samples (0 & 8), (2 & 10), (4 &12), (6 &14) and odd samples (1 & 9), (3 & 11), (5 & 13), (7 &15).This method first process even samples then followed by odd sample [9].This architecture can compute one 16 point FFT in 17 clock cycles as shown in data flow graph (DFG) in fig.3.

The scheduling can be derived using the folding sets as follows.

P = {P0, P2, P4, P6, P1, P3, P5, P7},
R = {R3, R5, Φ, R0, R2, R4, Φ, R1},
Q = {Q5, Q7, Q0, Q2, Q4, Q6, Q1, Q3},
S = {S2, S4, Φ, S1, Φ, S5, Φ, S0}

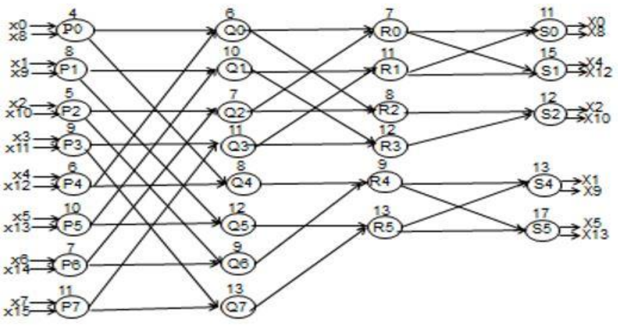


Fig.3. DFG of a 16 point DIF FFT

The nodes P0…..P7 represent 8 butterflies required in the first stage. The nodes Q0…..Q7 represent 8 butterflies required in the second stage and so on. The input samples in this architecture are processed in parallel. This design consists of real data path in first two stages and complex data path in other stages as shown in Fig 2. The switch represented in the architecture has two multiplexers as given in Fig 4.

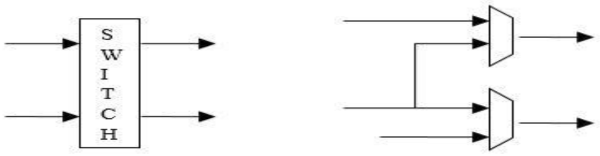


Fig.4. Switch equivalent circuit

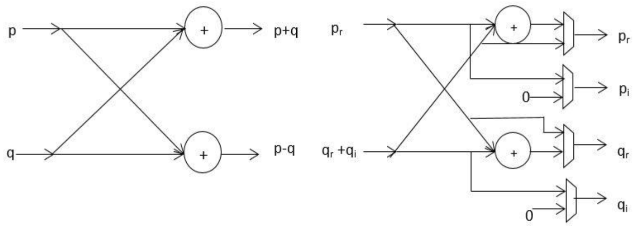


Fig.5. Butterfly structures for real data path

Four different butterfly structures are used to handle the complex and real data path as shown in Fig 5. BFIr and BFIIr are used in real data path. BFIr has two adders as one real adder and one real subtractor. BFIIr uses a butterfly with both real data and has logic to compute twiddle factor (coefficient) “-j” multiplication. In Fig6. BFIc and BFIIc are used in complex data path which contains the logic for twiddle factor(-j) multiplication and swapping of real part and imaginary part.

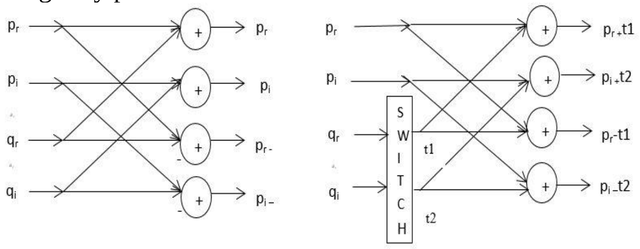


Fig.6. Butterfly structures for complex data path.

2-Parallel Radix-2^2 Architecture 2

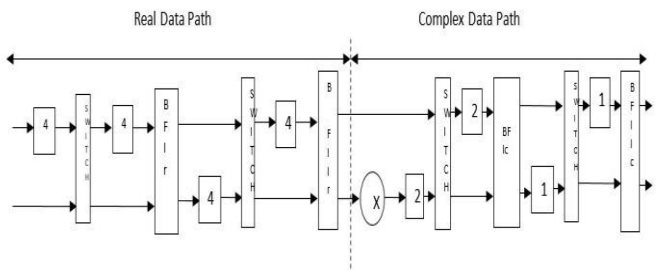


Fig.7. Architecture2 for the 16 point DIF FFT computation.

This scheduling method reduces the delay elements by processing the input samples consecutively, instead of separately sampling the even and odd inputs. When compared to first architecture, the numbers of clock cycles are reduced to 16 clock cycles. This is clearly seen in Fig 8 which represents the DFG for Fig 7 architecture [9].

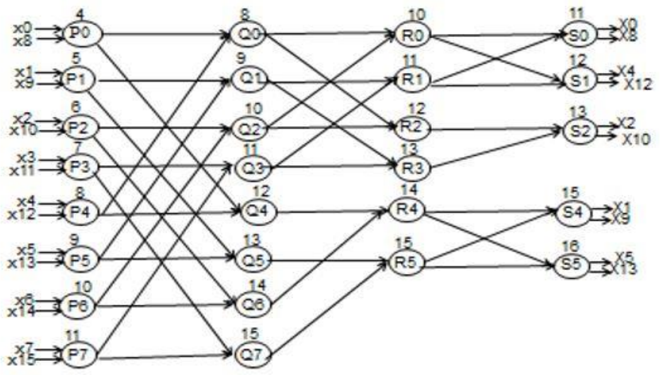


Fig.8. Data Flow Graph of 16 point DIF FFT.

This can be derived using the following folding sets.

P= {P0, P1, P2, P3, P4, P5, P6, P7},
R= {R2, R3, R4, R5, Φ, Φ, R0, R1},
Q= {Q4, Q5, Q6, Q7, Q0, Q1, Q2, Q3},
S= {S1, S2, Φ, S4, S5, Φ, Φ, S0}

4-Parallel Radix-2 architecture

The hardware complexity of the parallel architecture reduces based on radix-2 algorithm. The radix-2 architectures are described by N= 16 point DIF FFT.

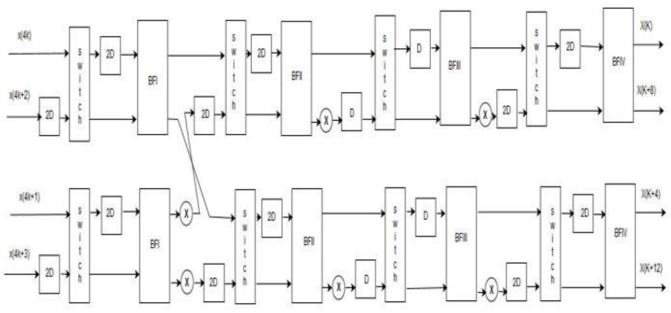


Fig.9. Architecture for the 16 point DIF RFFT computation.

The 1ststage has only real butterflies. Based on the input order of data, the upper part of butterfly structure has the pair of samples like (0,8), (1,9), (2,10), and (3,11) and the lower butterfly structure has pair of samples like (4,12), (5,13), (6,14), (7,15). The hardware resource complexity is twice that of the serial architecture and process four input samples in parallel manner. There is a 50% reduction in power consumption. Fig 9. represents the computation of 4 parallel radix-2 16 point DIF-FFT [10]. Fig.10 shows the pipelining and retiming cutsets for 16 point FFT architecture.

The architecture has the following folding sets

P = {P0, P1, P2, P3},	P’= {P’0, P’1, P’2, P’3},
Q = {Q1, Q3, Q0,Q2},	Q’={Q’1,Q’3,Q’0,Q’2},
R = {R2,R1,R3,R0},	R’={R’2,R’1,R’3,R’0},
S = {S3,S0,S2,S1},	S’={S’3,S’0,S’2,S’1}

For N-point FFT, the architecture takes multipliers of $4(log_4 N – 1)$ and delay elements of 2N-4.

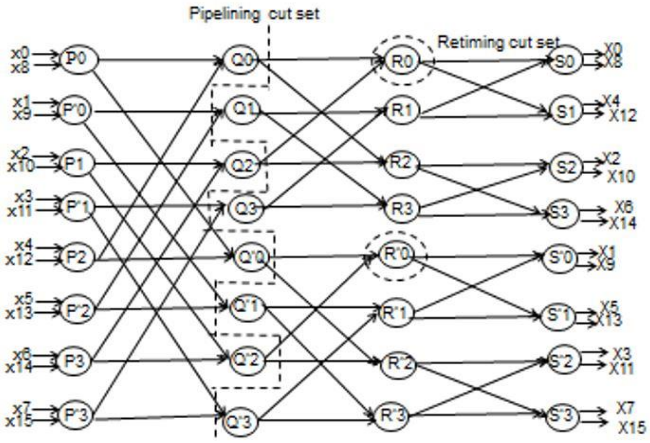


Fig.10. DFG for a radix-2^3 16-point DIF- FFT.

4-Parallel radix-2^3 64-point Architecture

The pipelined architecture for 4-parallel Radix-2^3 algorithm is shown in Figure11. This architecture processes 4 sequential samples in parallel. This 4-parallel architecture requires three butterflies as BFI, BFII, and BFIII. Similar to 2-parallel architecture the first butterfly uses only real data [11].

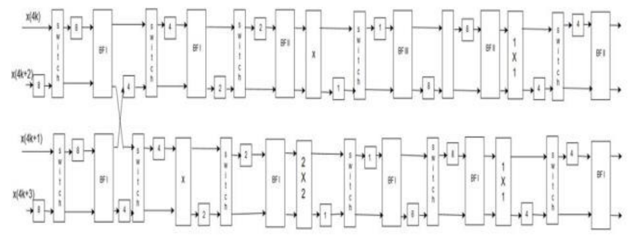
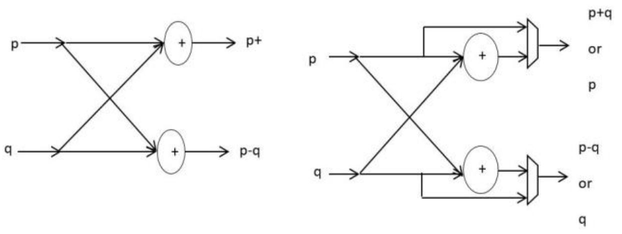
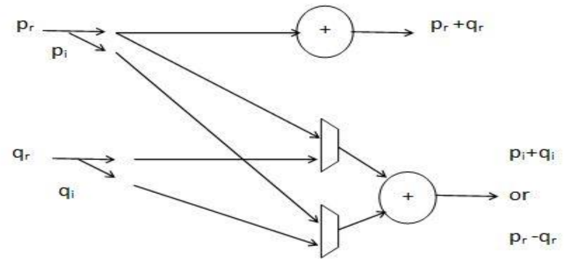


Fig.11. 4-parallel architecture for Radix-2^3 64 point FFT algorithm

Among the three butterflies the first butterfly has only real values whereas in second butterfly it has both real and complex values but the complex values are by-passed. Every data is a real number before it is multiplied to a complex multiplier. Third butterfly has both real and complex terms. Figure 12. shows the two butterfly structures which are meant for both real and imaginary data.



(a) BFI (b) BFII



(c) BFIII

Fig.12. (a),(b),(c) Butterfly structures

IV. REORDERING THE OUTPUT SAMPLES

Bit-reversal routine is considered to be an essential part of FFT because of its high possibility of degrading the overall execution time of FFT application if it is not perfectly designed. The output obtained in the FFT is in bit-reversed order [3] in serial architectures. In pipelined architecture, an additional memory of N addresses which is necessary to reorder the bit-reversed output samples.The input sequence is assumed to be in normal order for real-time processing in order to enable bit reversal circuits are used, So, the output is allowed to be in digit-reversed which can be applicable for DFT based communication system [2].Based on the type of algorithm, the output order changes for different folding sets. Fig 13. represents the basic reordering circuit. If the select line is choosing to be 1 in first mux, then the output is in the same order as that of input and if the select line is chooses to be 0 in second mux, the position is shuffled. The circuit is simple and consists of only of buffers and multipliers. These circuits use minimum number of registers and minimum latency as explained in [12].

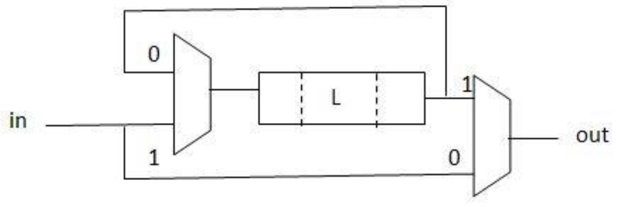


Fig.13. Basic reordering circuit.

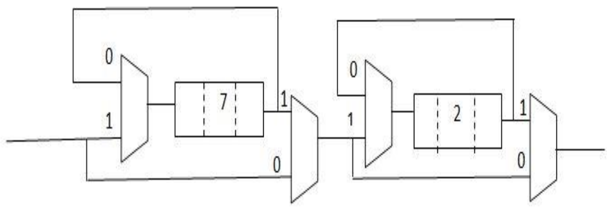


Fig.14. Bit-reversal circuits for a 16 point FFT

Fig 14 represents the bit reversal circuit of 16 point FFT. In case of bit reversal, as explained in [12][14], the permutation σ by interchanging the pair of dimensions x_i and x_{n-1-i}. The number of registers σ_i is represented by the general formula as

$$D(\sigma_i) = 2^{n-1-i} - 2^i \text{ .... (1)}$$

The total number of registers is given as

$$D(\sigma) = \sum_{i=0}^{\frac{n}{2}-1} D(\sigma_i) = (\sqrt{N}-1)^2 \text{ ....(2)}$$

Based on (2), the number of registers for 16 point is calculated as follows:

$D(\sigma 0) = 2^3 - 2^0 = 7$

$D(\sigma 1) = 2^2 - 2^1 = 2$

By using these calculations, the bit-reversal circuit for 16 point FFT is obtained. If the input sequence is represented in normal sequence order, the shuffle takes place and the output is obtained in the bit-reversed order. If the input sample itself is in the bit-reversed order, then the output is obtained in normal order using this circuit. These circuits can be used only if total length of the sequence is in the even power of 2.

By using this formula, the reordering for 1024 point FFT are given as

$D(\sigma 0) = 2^9 - 2^0 = 511$ $D(\sigma 1) = 2^8 - 2^1 = 254$

$D(\sigma 2) = 2^7 - 2^2 = 124$ $D(\sigma 3) = 2^6 - 2^3 = 56$

$D(\sigma 4) = 2^5 - 2^4 = 16$

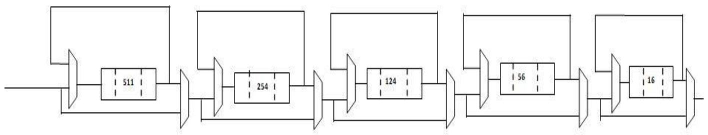


Figure 15. Circuit for bit reversal for 1024 point

Using this formula, the reordering for 1024 point can be done. For N point FFT, there are N/2 types of reordering circuit can be created. This is applicable only when N is even number.

IMPLEMENTATION AND RESULTS

2-parallel 16 point FFT architectures

Table 1

Resource Utilization of 2-parallel Architectures

DEVICE UTILIZATION	16-POINT ARCHITECTURE 1	16-POINT ARCHITECTURE 2
No. of Slice Registers	71	99
No. of Slice LUTs	49	68
No. of Occupied Slices	15	29

The table 1 shows the comparison of resource utilization between 2-parallel 16-point architectures. These architectures are implemented using different scheduling algorithms by altering the folding order for the butterfly structures. The slice registers, Slice LUTs and the occupied slice are realized for two-scheduling algorithms. It is inferred that Architecture 2 requires more resources than architecture1.

Table 2

Resource Utilization of 4-parallel Architectures

DEVICE UTILIZATION	16 - POINT ARCHITECTURE	64 - POINT ARCHITECTURE
No. of Slice Registers	127	150

No. of Slice LUTs	**124**	**113**
No. of Occupied Slices	**62**	**45**

The table 2 shows the comparison of resource utilization between 2-parallel for 16, 64 point architectures. The three algorithms are implemented based on the number of stages (2^n) as it expands based on the increasing power of 2. All the above FFT architectures have been simulated using Modelsim and implemented using FPGA.

Table 3

Comparison of FPGA Resource Usage for different FFTs

Pipelined Architecture	Slice Registers	Slice LUTs	Slices
16 point (2-parallel)	**71**	**49**	**15**
16 point (4-parallel)	**127**	**124**	**62**
64 point (4-parallel)	**150**	**113**	**45**

A comparison is made between different point FFTs for the hardware usage in Table 3. The throughput doubles based on the number of inputs given. 4-parallel 16 point. The 4-parallel pipelined FFT architecture has higher throughput compared to the 2-parallel pipelined FFT architecture.

Table 4

Comparison of Pipelined Architectures

	Adders	Multipliers	Normalized Throughput	Real Delays
2-parallel radix 2^2 architecture1	**$4\log_4 N$ - 2**	**$\log_4 N$ - 1**	**2**	**9N/8-2**
2-parallel radix 2^2 architecture2	**$4\log_4 N$ - 2**	**$\log_4 N$ - 1**	**2**	**N-2**
4-parallel radix 2	**$8\log_4 N$**	**$4(\log_4 N – 1)$**	**4**	**2N-4**
4-parallel radix 2^3	**$4\log_2 N$ - 2**	**$2(\log_8 N – 1)$**	**4**	**7N/4 - 4**

A comparison is made between the pipelined architectures based on the adders, multipliers and throughput. Normalized throughput doubles in the pipelined architecture for 4 parallel input which produces 4 samples as output each clock cycles. The timing details for 2-parallel 16-point architecture1 is given as minimum period is 3.370 ns and the frequency is 296.736 MHz and the minimum input arrival time before clock is 4.286 ns and the maximum output arrival time after clock is 5.280 ns. The maximum combinational path delay is 6.715 ns. For 2-parallel 16-point architecture2 the minimum period is 9.802 ns and the frequency is 102.020 MHz and the minimum input arrival time before clock is 11.628 ns and the maximum output arrival time after clock is 10.703 ns. The maximum combinational path delay is 11.565 ns. For 4-parallel 16-point architecture the minimum period is 9.980 ns and the frequency is 100.200 MHz and the minimum input arrival time before clock is 11.802 ns and the maximum output arrival time after clock is 6.053 ns. The maximum combinational path delay is 7.766 ns. For 4-parallel 64-point architecture the minimum period is 5.207 ns and the frequency is 192.049 MHz and the minimum input arrival time before clock is 6.274ns and the maximum output arrival time after clock is 10.109 ns. The maximum combinational path delay is 11.176 ns.

V. CONCLUSION

In this work, various pipelined FFTs are implemented and the hardware usage and latency performance are realized. The proposed FFT architecture is incorporated with circuit which converts the bit reversal order into a normal order which is not used in the previous architectures. It is inferred that throughput is doubled in 4-parallel architectures due to reduction in number of clock cycles to produce the samples at the cost of additional LUTs and registers compare to 2 parallel architectures. The future work focus on to extend the radix 2 architecture to radix 2^n with less hardware complexity and significant throughput performance.

REFERENCES

1. Yen-Nan Chang,"An Efficient VLSI Architecture for Normal I/O Order Pipeline FFT Design", IEEE Transactions on Circuits and Sys—II: Express Briefs, Vol. 55, No. 12, December 2008.
2. XueLiu, Feng Yu, Ze-keWang, "A pipelined architecture for normal I/O order FFT",IEEE Transactions on Circuits and Systems II: Express Briefs, Volume: 55 ,Issue:12, Dec. 2008.
3. M. Garrido, K. K. Parhi, J. Grajal, "A pipelined FFT architecture for real-valued signals", IEEE Transactions Cir Sys. I, Reg. Papers, vol. 56, no. 12, pp. 2634 - 2643,Dec. 2009.
4. Manohar Ayinala, Student Member, IEEE, Michael Brown, and Keshab K. Parhi, Fellow, IEEE," Pipelined Parallel FFT Architectures via Folding Transformation", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 20, NO. 6, JUNE 2012.
5. Tarek Belabed, Sabeur jemmali, Chokri Souani,"FFT Implementation and Optimization on FPGA", 4th International Conference on Advanced Technologies for Signals and Image Processing (ATSIP), 2018.
6. Basant K. Mohanty, Senior Member, IEEE, and Pramod Kumar Meher, Seniour Member, IEEE," Area-Delay-Energy Efficient VLSI Architecture for Scalable In-Place Computation of FFT on Real Data", IEEE Transactions on Circuits and Systems-I Regular Papers, Vol66, Issue 3, 2019.
7. Antony Xavier Glittas, Mathini Sellathurai, Gopalakrishnan Lakshinarayanan," A Normal I/O Order Radix-2 FFT Architecture to Process Twin Data Streams for MIMO", IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24(6) .1-5, 2016.
8. Antony Xavier Glittas, Mathini Sellathurai, Gopalakrishnan Lakshinarayanan, "Two-parallel pipelined fast fourier transform processors for real-valued signals", IET Circuits, Devices & Systems, Vol10, Issue 4, 2016
9. Manohar Ayinala, Member, IEEE, and Keshab K. Parhi, Fellow, IEEE, "FFT Architecture for Real-Valued Signals Based on Radix-23 and Radix-24 Algorithms", IEEE Transactions on Cir and Sys-I: Regular Papers.
10. Manohar Ayinala, Keshab K. Parhi, "Parallel-Pipelined Radix-22 FFT Architecture for Real Valued Signals", Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers,7-10 Nov,2010.
11. Kodakandla Abhilash and P.Reena Monica," FFT Architectures for Real Valued Signals based Different Radices Algorithm", Indian Journal of Science and Technology, Vol8(20), DOI:10.17485/ijst/2015/ v8i20/78462, August 2015.

12. Mario Garrido, Member, IEEE, Jesus Grajal, and Oscar Gustafsson, senior member, IEEE,"Optimum circuits for bit reversal", IEEE Transactions on Circuit0s and Sys—II: Express Briefs, Vol. 58, No. 10, October 2011.
13. Ching-Hsien Chang, Chin-Liang Wang, Member, IEEE and Yu-Tai Chang,"Efficient VLSI Architectures for Fast Computation of the Discrete Fourier Transform and Its Inverse", IEEE Transaction on Signal Processing, Vol. 48, No.11, November 2000.
14. Mario Garrido, Member, IEEE, J.Grajal, M.A.Sanchez, Oscar Gustafsson. Senior member, IEEE, " Pipelined Radix-2k Feedforward FFT architectures", IEEE Transactions on Very Large Scale Integration(VLSI), Vol21, NO.1, January 2013.

AUTHORS PROFILE

D. Dineshkumar was born in the year 1988 in India. He received his B.E degree in Electronics and Communication Engineering from Mepco Schlenk Engineering College and M.E degree in VLSI Design from College of Engineering Guindy in the year 2010 and 2012 respectively. He is presently working as an Assistant Professor in the Department of Electronics and Communication Engineering in Mepco Schlenk Engineering College. His area of interest includes VLSI signal processing and architectures.

R.Varun Prakash was born in the year 1988 in India. He received his B.E degree in Electronics and Communication Engineering from Mepco Schlenk Engineering College and M.E degree in Applied Electronics from SSN College of Engineering in the year 2010 and 2014 respectively. He is presently working as an Assistant Professor in the Department of Electronics and Communication Engineering in Mepco Schlenk Engineering College. His area of interest includes robotics perception.

G. Kirubakaran was born in the year 1993 in India. He received his B.E degree in Electronics and Communication Engineering from Kalasalingam Institute of Technology and M.E degree in VLSI Design from College of Engineering Guindy in the year 2014 and 2016 respectively. He is presently working as an Assistant Professor in the Department of Electronics and Communication Engineering in Mepco Schlenk Engineering College. His area of interest includes low power VLSI design and optimization.