



# FPGA Implementation of Fast Fourier Transform Using Hybrid Pipelining Technique

G. Prasanna Kumar<sup>1</sup>, P. Subhash<sup>1</sup>, K. Siva Prasad<sup>1</sup>, M. Venkata Subba Rao<sup>2</sup>,  
B. V. V. Satyanarayana<sup>1</sup>, Prudhvi Raj Budumuru<sup>1</sup>, and B. Elisha Raju<sup>1</sup> (✉)

<sup>1</sup> Department of ECE, Vishnu Institute of Technology, Bhimavaram, India  
elisharaju.b@vishnu.edu.in

<sup>2</sup> Department of ECE, Sri Vishnu Engineering College for Women, Bhimavaram, India

**Abstract.** Implementation of Fast Fourier Transform with optimization of adder and multipliers is very important in FPGA implementation. There are many parallel and pipeline techniques were existing for optimizing the subcomponents. This paper presents a new hybrid pipelining architecture, which optimizes the multipliers. It is a dedicated design of real inputs, which is used in many practical applications. The design consists of single processing units for all the stages except the second stage. The memory required for the twiddle factors storage given by the linear life time chart and the analysis of data flow graph gives the optimized result of hybrid pipeline at the stage of insertion. We use ip-core multipliers for the multiplications of twiddle factors in the proposed method. The performance of the proposed method is compared in terms of power consumption which is reduced by the 20% of the existing techniques.

**Keywords:** FFT · FPGA · Hybrid Pipelining · IP Core · Twiddle Factor Multiplier

## 1 Introduction

Implementation of DSP applications on FPGA is one big area, in which much research work has been carried from the past decade. The real challenge is optimization of sub components. The implementation of FFT is the major problem of complexity so to challenging this, the number of multipliers and adders to be reduced. The applications of FFT includes in communication especially in cognitive radio [1, 2], in which pipelined FFT designs are necessary to reduce the hardware. The challenge in pipelined designs is delay, which has more delay. This ambiguity mentioned in [3], which is highly preferable and promising solutions were clearly explained in detail. This was a big solution for implementing higher order FFT.

One very important application of FFT is EEG, in which the main data input is real in nature. The RFFT has an advantage of eliminating number of computations and also improving processing speed. This is because of the symmetry property of the FFT and folding technique for optimizing the resources. The main modules in FFT implementation are adders in butterfly unit and multipliers for twiddle factor multiplication. FFT

designs with optimized resources addressed in [3, 4], these designs are pipelined designs which optimizes the resources in only one direction.

Storage is also one big factor in pipelined designs, [5] proposes an improved efficient storage FFT design and in [6] proposes FFT design with low latency high throughput. RFFT optimized designs was presented in [7, 8], which is mainly optimized in the sense of hardware in FPGA. The controller guide the all the modules in order to achieve a serial FFT output. The other important considerations in the design of FFT is twiddle factor multiplier which is addressed in [9, 10]. The cardiac rotation is one approach for effective twiddle factor multiplication. In [11], authors present UMA algorithm for multiplier which was very effective for FFT implementation. This subcomponent minimization is one big consideration in FFT implementation.

The FFT designs [12, 13] implemented are effective because of bit reversal operations performed in design itself. In [14] authors proposed a design which resource efficient in FPGA that is LUT's, FF pairs and other slice registers. In [15] authors presents FFT design with novel bit reversal module at the final stage for getting original output. The FFT module [16] gives the frequency measurements on FPGA. The overall literature of the FFT gives a way to further optimization. In [17–20] authors presents the pipelined FFT that is mixing of parallel design and pipelined design, which is a comprisable design between the power and latency and very effective for higher order FFT designs.

The literature review concludes that the reducing the power consumption increase the latency and vice versa. Most of the pipelined designs are fixed number of components in each stage, but there is further optimization possible at intermediate stages of FFT. The proposed design is hybrid pipelined design with two processing units in order to increase the speed of processing. This design implemented using ip core multipliers gives best results comparing to existing methods.

## 2 Implementation of FFT

RFFT is the FFT for real inputs which is designed using DFT's conjugate symmetry property, which reduces approximately half of the computations because of symmetry in output samples. The RFFT architecture for 16 samples using R-22 is shown in Fig. 1. The property is shown in equation 1, by this property complexity reduced at the output stage.

$$X(K) = X * (N - k) \quad (1)$$

## 3 Design Optimization

The existing architectures are 1 parallel, 2-parallel and several multi parallel designs which are standard designs where pipelining done in one way each parallel stage. Figures 1 and 2 shows the 1-parallel and 2-parallel designs. In these designs PU represents the processing unit, which generates the butterfly units of 2 inputs and gives 2 outputs. 1-parallel design consist parallel unit in each stage which increase the delay but equipment cost reduced. This delay effect the overall system performance. The two parallel designs compute the

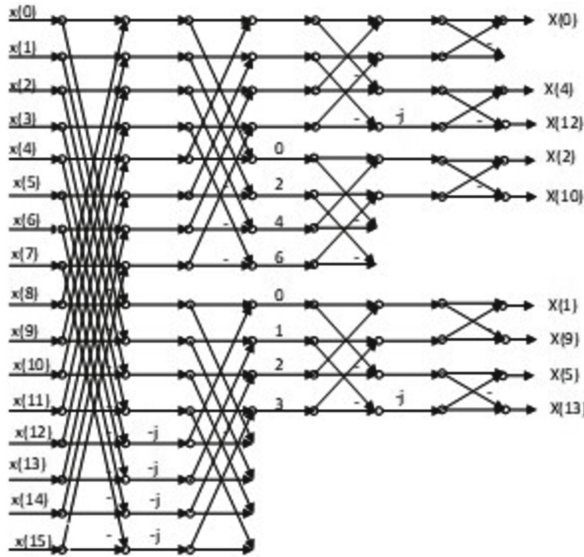


Fig. 1. 16- point FFT for Real signals.

2- parallel design shown in Fig. 3 which is comprisable design of power and reduces the latency, so in order to improve the performance of the overall system for power and latency. This paper presents the hybrid pipelined FFT design which comprisable design between power and latency. This design has an advantage of low power and low latency. Further increase of order gives the better performance.

The advantage of 2-parallel design is number of processing units and the multipliers are little and whereas for 4-parallel design is its delay is low. Further increase of parallel architecture will reduce the further latency. This parallel unit increase up to the order of the FFT. These pipelined methods are traditional and common for all the different architecture. The proposed method is hybrid pipelined method more focus on RFFT. Pipelining is necessary when the processing units are limited, here additional processing unit at the second stage increase the speed of processing at second stage. This reduces the overall latency of FFT module which is very important for wireless applications.

The propose hybrid pipelined FFT design for 16 points is shown in Fig. 4, in which PU represents the processing units. The processing is the mainly 2 input butterfly units which consist of two complex adders parallel compute the two samples at time in a stage. Stage 1 consists of one PU unit whereas stage 2 consists of two PU units which produces 4 samples at a time so that no delay in the stage-2.

In general latency of 1-parallel design for computing overall FFT output is  $3N/4-1$ , the latency decrease when more parallel stages are included at the intermediate stages of the FFT. Each stage of pipelining additional delay includes because of reusing the same module for more samples. Inclusion one additional PU unit reduces the Latency. The crucial part in this design is selection of the stage where additional PU unit includes for reducing latency.

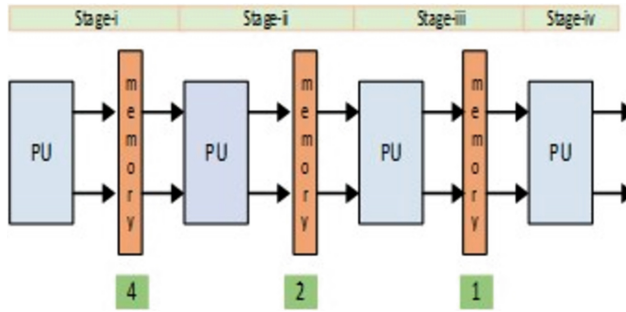


Fig. 2. 1-Parallel design of 16-values.

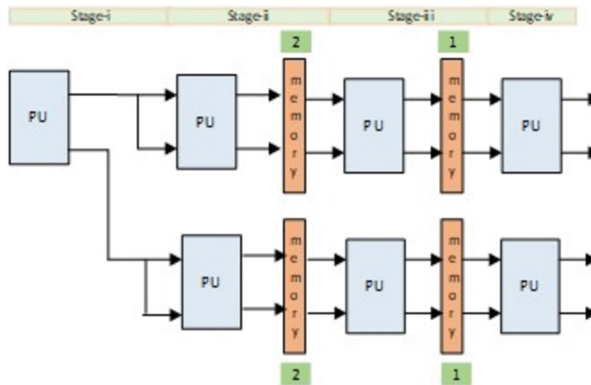


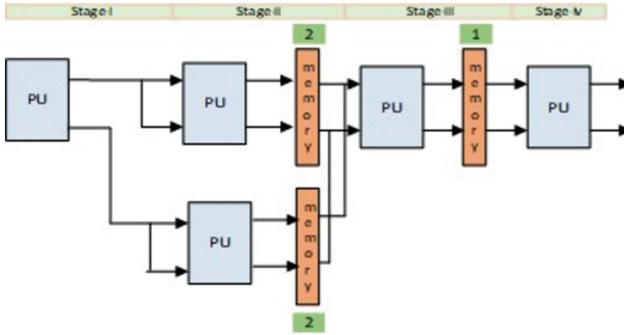
Fig. 3. 2-parallel design of 16-values

$Lat=3N/4-1$  is the expression of latency for 1-parallel design which is the overall latency of the architecture, if parallel unit is included in second stage then  $Lat\_2$  is the latency. Latency will be reduced by a factor of  $N/8$  if we include 1-parallel unit in second stage and it will be reduced by  $N/16$  for the inclusion of 3<sup>rd</sup> stage. The hybrid pipelining is the optimized solution helps us to reduce this latency by a factor of  $5N/8-1$  for being inclusion of 1-parallel stage. The proposed design follows the hybrid pipelining and also ip core multiplier for twiddle factor multiplication. Here memory is the FIFO for writing and reading the intermediate values at each stage of the output.

### 4 Design Optimization Using Folding Transform

The folding transformation to minimize the hardware cost for 2-parallel, 4-parallel and twin parallel processing is analyzed in section 2 & 3. In this section the folding transformation is applied to the new pipelined FFT architecture in order to optimize the hardware. Consider the DFG of the hybrid pipelining parallel processing architecture is shown in Fig. 4.

The data paths from set of nodes P to Q are splitting section and from Q to R, Q1 to R is the merging section. Section 3 will gives the elaborated analysis of the splitting



**Fig. 4.** Theme of hybrid pipelining FFT design

section in which merging analysis is also described. The folding sets corresponding to DFG is P is the set of nodes corresponding to first stage, Q, Q1 are the set of nodes in splitting section, R is the set of nodes in merging section. The folding transformation for this architecture at merging section is applied using the equation. The negative delay paths are retimed using the retiming equation, by adding one delay in the path of the nodes the resultant of modified data flow delay shown in 3rd column in the equation.

$$\begin{aligned}
 P &= \{P0, P1, P2, P3, P4, P5, P6, P7\} \\
 Q &= \{Q0, Q1, Q2, Q3\} \quad Q1 = \{Q10, Q11, Q12, Q13\} \\
 R &= \{R2, R3, R4, R5, R6, R7, R0, R1\} \\
 S &= \{S1, S2, S3, S4, S5, S6, S7, S0\}.
 \end{aligned}$$

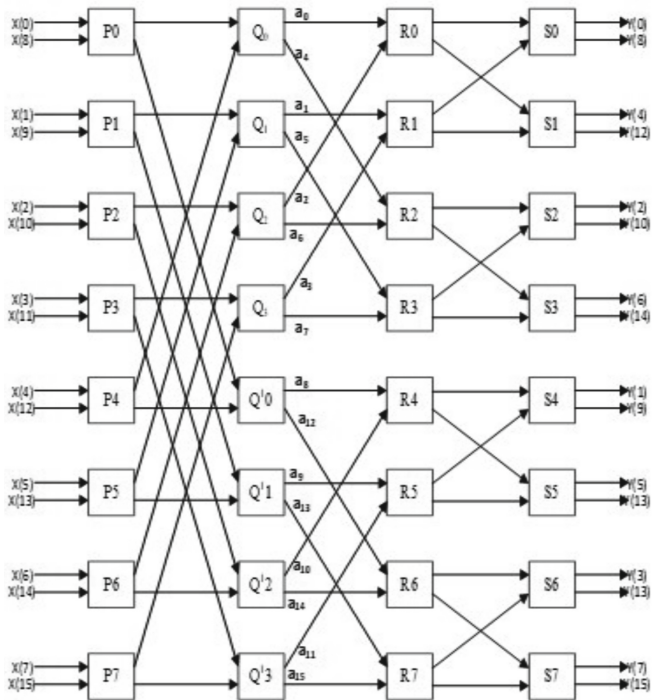
Consider the Linear life time chart at merging section that is shown in Fig. 5. The set of input nodes are Q and output nodes R in the merging section. The folding sets are shown in figure, here Q has two sets of nodes that Q and Q1, because it is splitting section there are two parallel processing units are induced in the architecture. The folding set R is the merging section in the architecture.

The outputs of Q and Q1 are a0,a1,----a15, {a0,a4} generated in 12th cycle a0 is utilized in 14th section ,a4 is utilized in 16th cycle, it shows that there is some latency because of merging section. The complete generation and utilization of all the outputs are shown in Fig. 5. This shows that the maximum registers required to store the intermediate values is 8, in FIFO based design it is FIFO of storage size 8 is the required one. In this architecture total FIFO's each of size 2 is used.

The FFT architecture for real valued signals is efficiently designed by eliminating the redundant data paths in the dataflow graph (DFG). The radix-22 DIF FFT with redundant data paths are shaded is shown in Fig. 5, these paths are eliminated and Fig. 7 depicts the resultant flow graph.

The redundant data paths are removed in DFG of Fig. 7, it shows that half of the output nodes are reduced. The Linear life time chart at the merging section of RFFT is shown in Fig. 8. It shows that at merging section, 6 registers are required. In the FIFO based design, FIFO of size 6 is required at merging stage of RFFT.

$Fl(Q0 \rightarrow R0) = 6 - 0 = 6$	$Fl(Q0 \rightarrow R4) = 2 - 0 = 2$	$Fl(Q0 \rightarrow R0) = 6$
$Fl(Q0 \rightarrow R2) = 0 - 0 = 0$	$Fl(Q0 \rightarrow R6) = 4 - 0 = 4$	$Fl(Q0 \rightarrow R2) = 0$
$Fl(Q1 \rightarrow R1) = 7 - 1 = 6$	$Fl(Q1 \rightarrow R5) = 3 - 1 = 2$	$Fl(Q1 \rightarrow R1) = 6$
$Fl(Q1 \rightarrow R3) = 1 - 1 = 0$	$Fl(Q1 \rightarrow R7) = 5 - 1 = 4$	$Fl(Q1 \rightarrow R3) = 0$
$Fl(Q2 \rightarrow R2) = 0 - 2 = -2$	$Fl(Q2 \rightarrow R4) = 2 - 2 = 0$	$Fl(Q2 \rightarrow R2) = 2$
$Fl(Q2 \rightarrow R0) = 6 - 2 = 4$	$Fl(Q2 \rightarrow R6) = 4 - 2 = 2$	$Fl(Q2 \rightarrow R0) = 4$
$Fl(Q3 \rightarrow R3) = 1 - 3 = -2$	$Fl(Q3 \rightarrow R5) = 3 - 3 = 0$	$Fl(Q3 \rightarrow R3) = 2$



**Fig. 5.** DFG of 16-point pipelined design with hybrid pipelining parallel processing

Figure 9 depicts the architecture of 16-point FFT for real valued signals by means of split merge parallel processing unit. Radix-22 DIF FFT data flow style is used for the design of this architecture. In which splitting section included in stage -2 and merging at stage-3. The splitting section consists of two 2-parallel butterfly units (BU2) and from merging section it is one only. The latency of this design is low compared to CFFT, because of this optimized design Latency and hardware are low compared several architectures.

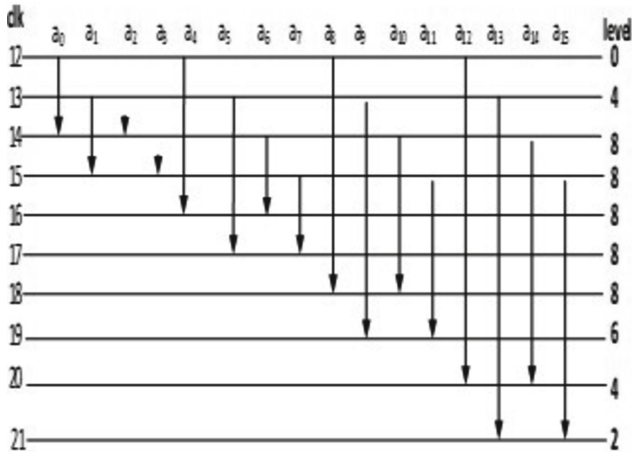


Fig. 6. Linear Life Time Chart at merging section

The proposed hybrid pipelining design of 16 point FFT is shown in Fig. 9. It consists of four stages and each stage has one BU unit except the second stage where hybrid pipelining is included. Two input multiplexers are included between intermediate stage where pipelining is required in order to guide the correct input to the next stage. The FIFO unit at first stage stores 8 samples. In second stage two FIFO's with storage of 4 is required. The twiddle factor multiplier are included after stage 2 and the coefficients are as follows (0, 2), (4, 6), (0, 1) and (2, 3). Latency comparison of hybrid pipelining technique with the existing technique is shown in Table 1, it indicates that 50 percent reduction in latency of RFFT compared to 1-parallel design and 20 percent reduction in PU units compared to 2-parallel design.

### 5 Simulation Results

The proposed designs are implemented on Vertex-7 AMD VIVADO tool. The FFT design was implemented using ip core multiplier in vivado and the detailed synthesized results shown in Table 2. The proposed hybrid pipelining technique for real valued signal achieves better power efficient results compared to existing techniques.

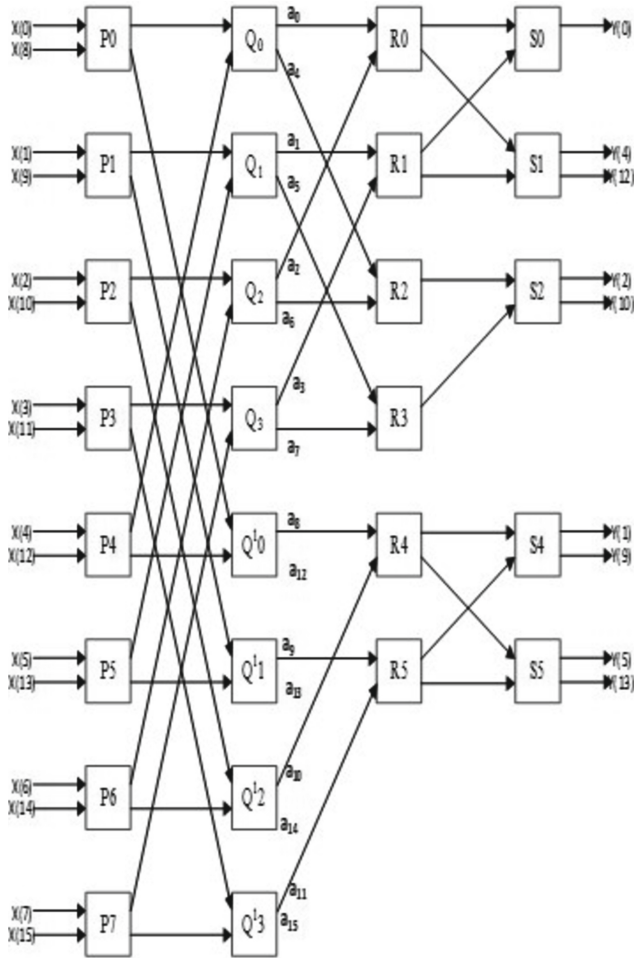


Fig. 7. DFG of 16-point pipelined design using hybrid pipelining parallel processing units

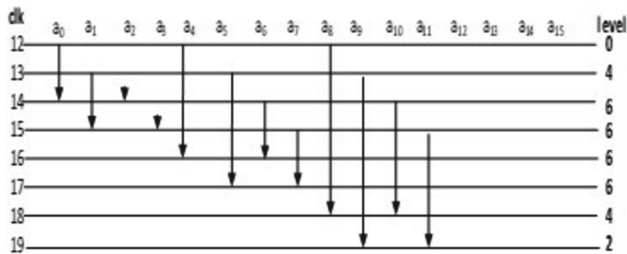
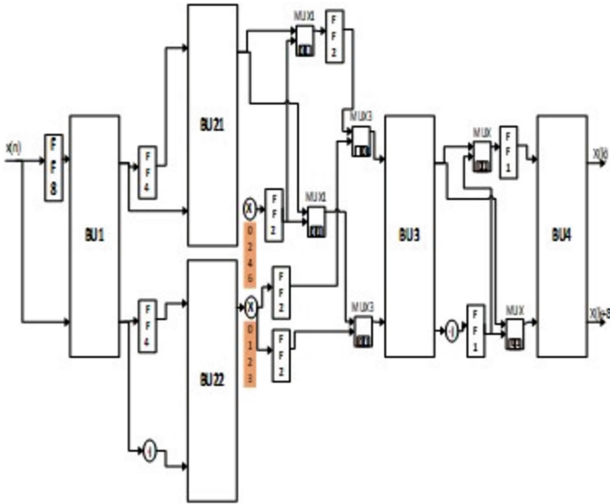


Fig. 8. Linear Life Time Chart at merging section for RFFT



**Fig. 9.** 16-point pipelined FFT architecture using hybrid pipelining parallel processing for real signals

**Table 1.** Latency comparison of pipelining architectures

Design Type	Delay(cycles)							Total	PU Units
	1	2	1	4	1	2	1		
1-Parallel Design	1	2	1	4	1	2	1	12	4
2-Parallel Design	1	2	1					4	7
Proposed hybrid pipelining (CFFT)	1	2	1	2	1	2	1	10	5
Proposed hybrid pipelining (RFFT)	1	2	2	1					5

**Table 2.** Comparison of synthesized results

Type of design	Slice registers	Slice LUTs	Power(mw)	Execution time (ns)
Floating point FFT [16]	420	726	42	3.1
Reconfigurable radix-22 FFT s [14]	340	629	30	--
FFTwith MUMA [12]	317	565	25	--
<b>FFT using IP core multiplier</b>	305	536	22	

## 6 Conclusion

A new hybrid pipelined FFT architecture is proposed in this paper, which is optimized for real valued signals. The multiplier in the resultant design was implemented using IP core multiplier. The results show that FFT with ip core multiplier gives better power efficiency compared to existing methods. This architecture is preferred for the implementation of higher order FFTs because it consumes less power compared to other existing architectures, especially for higher order FFTs. In future these designs can be implemented for DSP applications like DCT.

## References

1. Sundararajan, M., Govindaswamy, U.: Multicarrier spread spectrum modulation schemes and efficient FFT algorithms for cognitive radio systems. *Electronics*. **3**(3), 419–443 (2014)
2. Dawwd, S.A., Nori, S.: Reduced Area and Low Power Implementation of FFT/IFFT Processor. *Iraqi Journal for Electrical & Electronic Engineering*. **14**(2), 108–119 (2018)
3. Kanders, H., Mellqvist, T., Garrido, M., Palmkvist, K., Gustafsson, O.: A 1 million-point FFT on a single FPGA. *IEEE Transactions on Circuits and Systems I: Regular Papers*. **66**(10), 3863–73 (2019)
4. Elango, K., Muniandi, K.: VLSI implementation of an area and energy efficient FFT/IFFT core for MIMO-OFDM applications. *Annals of Telecommunications*. **75**, 215–227 (2019)
5. Noor, S.M., John, E., Panday, M.: Design and implementation of an ultralow-energy FFT ASIC for processing ECG in cardiac pacemakers. *IEEE transactions VLSI systems* **27**(4), 983–987 (2018)
6. Ordóñez, A., Argüello, F., Heras, D.B.: GPU accelerated FFT-based registration of hyper-spectral scenes. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **10**(11), 4869–4878 (2017)
7. Saini M., Verma S., Sharan A.: Multi-view ensemble learning using rough set based feature ranking for opinion spam detection. In: *Advances in Computer Communication and Computational Sciences: Springer Singapore. Proceedings of IC4S, Volume 1, 3–12* (2019)
8. Yang, L., Zhang, K., Liu, H., Huang, J., Huang, S.: An efficient locally pipelined FFT processor. *IEEE transactions on circuits And systems II: Express Briefs* **53**(7), 585–589 (2006)
9. Ayinala, M., Brown, M., Parhi, K.K.: Pipelined parallel FFT architectures via folding transformation. *IEEE Transactions VLSI Systems* **20**(6), 1068–1081 (2011)
10. Garrido, M., Parhi, K.K., Grajal, J.: A pipelined FFT architecture for real-valued signals. *IEEE Trans. Circuits Syst. I Regul. Pap.* **56**(12), 2634–2643 (2009)
11. Hasanin, M.E.H.A.: Design and implementation of low complexity fast Fourier transform architecture (Master's thesis) (2020)
12. Nguyen, H.N., Khan, S.A., Kim, C.H., Kim, J.M.: A pipelined FFT Processor using an optimal hybrid rotation scheme for complex multiplication Design FPGA implementation and analysis. *Electronics* **7**(8), 137 (2018)
13. Tang T., Pan J., Yang D., Wang E.: Frequency measurement by FFT based on FPGA. In: *IEEE 3rd advanced information technology, electronic and automation control conference (IAEAC)*, 453–456 (2018)
14. Glittas, A.X., Sellathurai, M., Lakshminarayanan, G.: A normal I/O order radix-2 FFT architecture to process twin data streams for MIMO. *IEEE Transactions VLSI Systems* **24**(6), 2402–2406 (2016)

15. Rani, M.R., Kumar, G.R. and Kumar, G.P.: Implementation of March Algorithm based MBIST Architecture for SRAM. *International Journal of Advanced Research in Computer Engineering & Technology*, 1(3). (2012)
16. Indirapriyadarsini, K., Kamalakumari, S., Prasannakumar, G.: VLSI implementation of pipelined fast fourier transform. *Int. J. Adv. Res. Comput. Eng. Technol.* **1**(4), 427–432 (2012)
17. Satyanarayana, B.V.V., et al.: Design and Performance Analysis of 11-T SRAM Cell with BTI Reliability, In: 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, pp. 166–173. (2023)
18. Satyanarayana, B.V.V., Lakshmi, B.K.S., Kumar, G.P., Srinivas, K.: (2024) LUT-Based Area-Optimized Accurate Multiplier Design for Signal Processing Applications. In: Pareek, P., Gupta, N., Reis, M.J.C.S. (eds.) *Cognitive Computing and Cyber Physical Systems. IC4S 2023. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 536 (2023)
19. Kumar, G.P., Krishna, B.T., Kotipalli, P.: Split Radix Fast Fourier Transform Design with Reduced Multipliers. (2020)
20. Raju Kalidindi, S.N., Terlapu, S.K., Krishna, M.V.: Implementation of efficient reconfigurable FIR filter with control logic for 5G applications. *Soft Comput* **25**, 10509–10518 (2021)