iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4579

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Efficient Mapping of a 64 Point Pipelined Architecture

Merlin Ponnachan[1], Anas. A. S[2]
[1, 2]*Department of ECE, APJ Abdul Kalam Technological University*

Abstract: ***In this paper, a pipelined Single-Path Delay Feedback (SDF) Fast Fourier Transform (FFT) architecture is efficiently mapped to Field-Programmable Gate Arrays (FPGAs). Several Algorithmic transformations are proposed that allow a better mapping of the architecture. To get better implementation results, target FPGAs architectural features are considered. A R2^{2}SDF FFT core for 64-point is mapped to Xilinx Virtex-6 device. The resulting Virtex-6 design gives an increase in throughput per slice, uses half as many DSP blocks. The architecture is simulated on Xilinx ISE 14.2 ISim with Speed grade -2 and at a maximum frequency of 167.259MHz.***

Keywords: Algorithmic transformations, Fast Fourier Transform, Single-path Delay Feedback, Comparison

I. INTRODUCTION

Nowadays, VLSI technology has become very popular and growing rapidly and the demand for handheld devices such as mobile phones, laptops etc. and DSP systems has increased tremendously. Three major issues for VLSI are area, power and speed. So there is a need to develop an efficient system. The FFT is a major block in DSP and communication systems. Fast Fourier transform is an algorithm that computes the frequency components of a signal. It is a faster algorithm that computes Discrete Fourier transform (DFT) efficiently, where DFT is the transform used commonly in Signal Processing applications. There are various FFT architectures that process different number of samples. Pipeline FFT architectures can be used for high speed applications. It can consume either one or several samples per iteration.

Field Programmable Gate Arrays (FPGAs) are semiconductor devices based on matrix of Configurable Logic Blocks (CLBs) connected through Programmable Interconnects. FPGAs can be reprogrammed after manufacturing for functionality requirements or desired applications. It has shorter design time and reduced nonrecurring expenses when compared with ASICs which are customized for particular design tasks. Due to the programmable nature of FPGAs, they are ideal fit for different markets. The configuration of FPGA is specified using a hardware description language (HDL).

The high performance requirements in signal processing applications can be obtained with the help of highly pipelined processors [3]. Various schemes can be used in pipelined processors and it includes R2MDC, R2SDF, R4SDF, R2^{2}SDF etc. Among these, the R2^{2}SDF has structural advantages on pipelined architectures over other algorithms [4]. In [5] different optimization and rounding techniques are explored over two different pipelined architectures that achieved better performance with lower resource usage. In [6] an area efficient algorithm is proposed for pipelined FFT architecture. In[7] multiplierless constant rotators based on Combined Coefficient Selection and Shift and Add Implementation(CCSSI) is proposed.

In this work, an FFT is designed for 64 point with some algorithmic transformations applied so that it could be efficiently mapped on an FPGA.

II. PIPELINED ARCHITECTURE

The structure of a 64 point SDF FFT is shown in Fig.1. The architecture consists of a pipeline of total $\log_2 N$ i.e., 6 radix 2 SDF Butterfly (BF) processing units. A total memory of N-1 is required and is minimal for single stream architectures [4]. Two types of BF units are used here. In between them, there is a trivial multiplier which performs the conditional multiplication of –j.

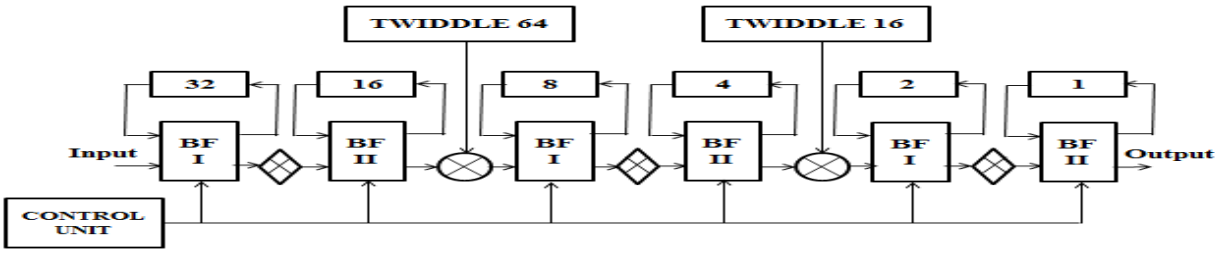


Fig. 1 64 point SDF pipeline architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

The structure of trivial multiplier is shown in Fig.2 on which the real and imaginary parts are swapped and the imaginary part is negated. The trivial multiplier is controlled by the control signals of the adjacent butterfly units i.e., the signals of the previous BF unit (as s) and the next BF unit (as t).

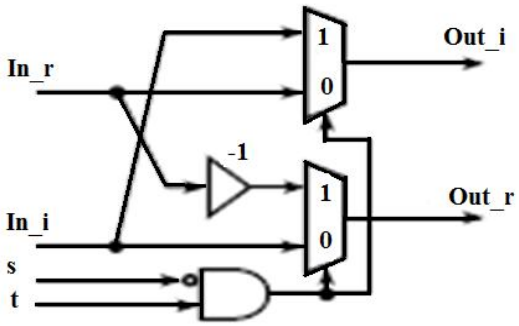


Fig 2. Structure of trivial multiplier

A $\log_2 N$ bit synchronous counter can be used to generate the control signals. Each bit of the counter controls each BF units in an order such that the LSB bit controls the last BF unit and the MSB bit controls the first BF unit. The architecture operates on normal input data and an output on bit-reversed order is obtained.

III.PROPOSED TRANSFORMATIONS

The constituent parts of the architecture are explained along with the transformations applied.

A. Complex Multiplier

Complex multipliers are used to multiply the twiddle factors with the processing datas. It includes four real multipliers and two adders as shown in Fig. 3. The figure shows the multiplication of (a + jb) and (c + jd). In the FFT cores, it is efficient to map the multipliers to DSP blocks if they are available. In the Virtex 6 DSP48E1 [8], the hard multiplier is 18 * 25. Each complex multiplier requires four DSP48E1 units and the latency is 4 clock cycles.

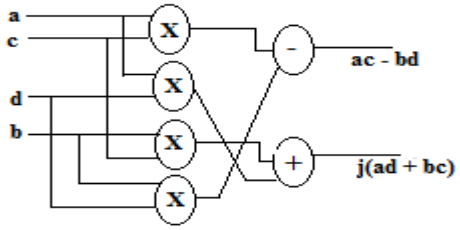


Fig 3. Complex Multiplier

B. Twiddle Factors And Mapping

For an N-point FFT, for the efficient memory storage and mapping only (N/8) twiddle factors are needed to be calculated. The remaining twiddles are related to that (N/8) factors as explained below:

Twiddle factor, $W_N = a - jb$

whereas and b are the real and the imaginary parts of the twiddle values. The real and imaginary values of the first (N/8) twiddles should be stored separately or retrieve the values separately so that it can be used for the remaining twiddle calculations.

Then for complex multiplication retrieve the real and imaginary values as given below,

The first 8 twiddle factors would be like,

$W_N^k = a_i + j(-b_i)$, $0 <= k <= N/8$ and $0 <= i <= N/8$

and the remaining twiddles would be,

$W_N^k = b_i + j(-a_i)$, $N/8 < k <= N/4$ and $N/8 > i >= 0$

$W_N^k = b_i + j(-a_i)$, $N/4 < k <= N/2$ and $N/4 > i >= 0$

$W_N^k = b_i + j(-a_i)$, $N/2 < k <= 3N/4$ and $0 < i <= N/4$

Thus for the non-trivial twiddle factor multiplication i.e., for the complex multiplication, only a total of first 8 twiddle factors are needed to be stored as the others are dependent on it and hence the memory for twiddle factor storage can be reduced. The twiddle factors calculated for the design is given in table I.

TABLE I
TWIDDLE FACTORS CALCULATED FOR 64-POINT FFT

Twiddle Factors	Values
W_{64}^{1}	0.995184 – j0.098017
W_{64}^{2}	0.980785 – j0.195090
W_{64}^{3}	0.956940 – j0.290284
W_{64}^{4}	0.923879 – j0.382683
W_{64}^{5}	0.881921 – j0.471396
W_{64}^{6}	0.831469 – j0.555570
W_{64}^{7}	0.773010 – j0.634393
W_{64}^{8}	0.707106 – j0.707106

The twiddle factors are mapped to the complex multipliers with the help of the counter used in the architecture. The mapping is done in such a way that, the twiddle 64 multiplication will occur in the 48th counter value which is the sum of the shift registers up to the first complex multiplier and the twiddle 16 multiplication will occur in the 60th counter value which is the sum of the shift registers up to the second complex multiplier from the Fig. 1.

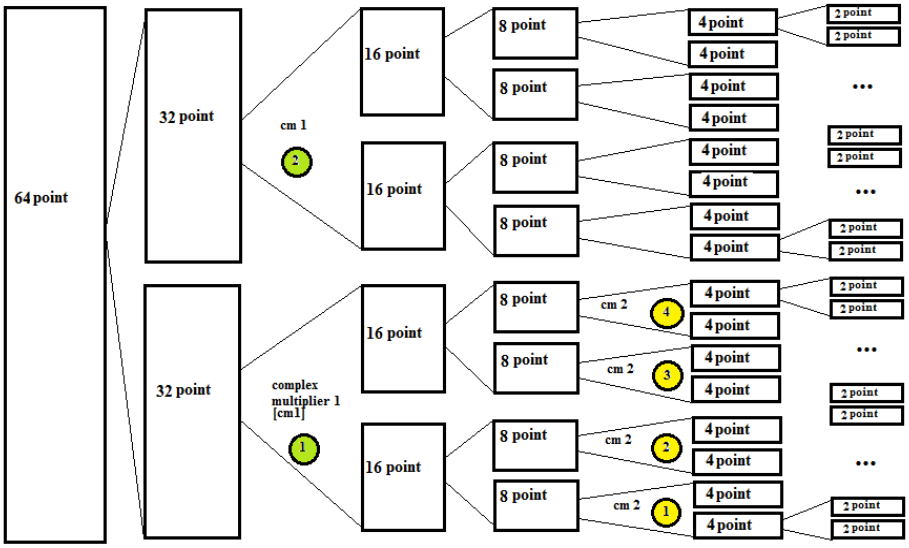


Fig 4. Order of mapping twiddle factors

Figure 4 shows the order of mapping twiddle factors using counters. cm 1 represents the first complex multiplier and cm 2 represents the second complex multiplier. The lower set of twiddle factors are first mapped as shown above figure.

C. Butterfly Units

The structure of the butterfly unit is shown in Fig. 5. x1_r, x1_i, x2_r and x2_irepresent the real and imaginary parts of the inputs. x1 represents the feedback input and x2 represents the direct input and y1 represents the direct output and y2 represents the feedback output. Butterfly unit performs the addition and subtraction of the inputs. Data multiplexers are used at the last part to select the inputs with respect to the control signal, s. If the control signal is zero, the inputs are directly passed to the output side and if the control signal is one, the added and subtracted outputs are passed to the output side.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

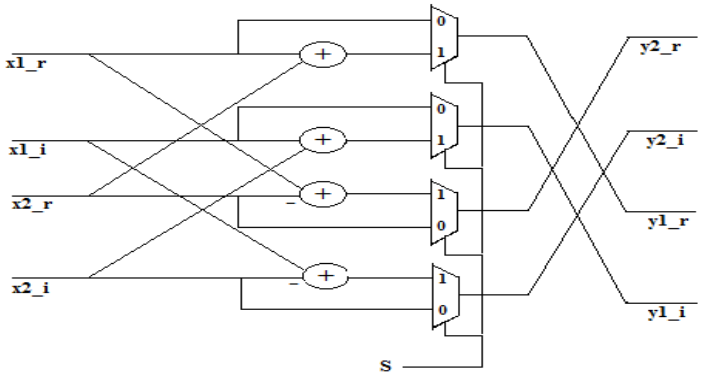


Fig 5. Structure of Butterfly Unit

For an N point FFT processor, the output should be scaled by 1/N due to finite word length effect [5]. The scaling can be distributed into 1/2 scaling throughout the operations and can be placed in between the adders and the data multiplexers in BF units. On implementation each components are mapped onto the LUTs and the efficient LUT usage can be obtained by means of some transformations.

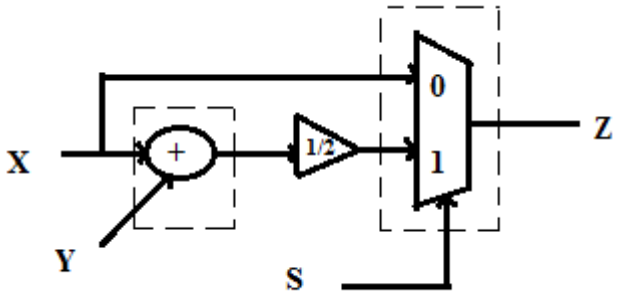


Fig 6. Scaling of data

In the BF unit at the place of adders, each of the adder and the multiplexer take one LUT and a total of 2 LUTs are used as shown in Fig. 6. In virtex6, LUT6 is used and both the adder and the multiplexer can be mapped onto a single LUT and this can be obtained by placing the scaling factor after the multiplexer by using the fact that Z= (Z + Z)/2 and is shown in Fig.7.

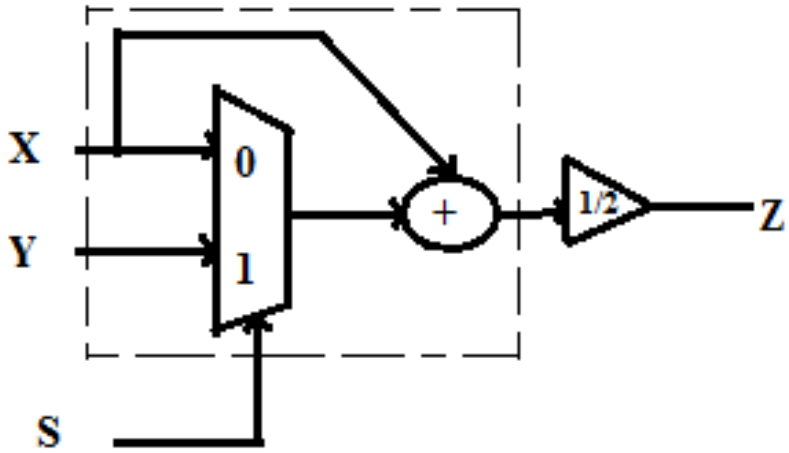


Fig 7.Efficient LUT usage

And hence the structure of BF I would be like as shown in Fig.8. After every BF II units, there is a complex multiplier. And these complex multipliers are mapped to DSP48E1 blocks.

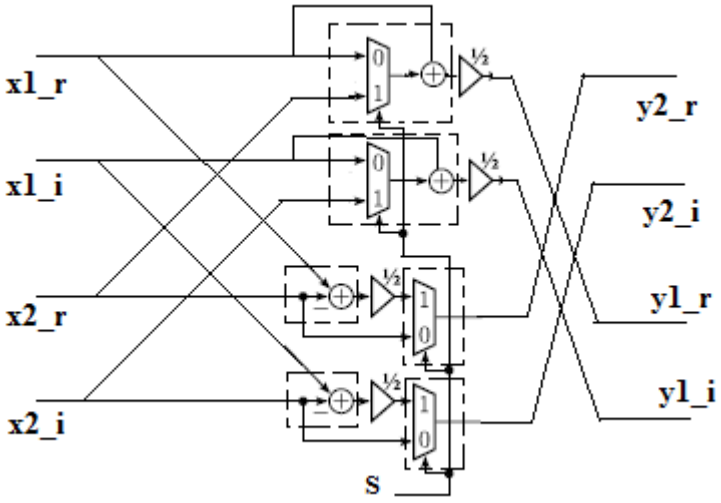


Fig 8. Butterfly Unit I

Some FPGA families exhibits preadders before hard multipliers and one among them is Virtex 6. If the adder of the BFII unit can be placed just before the complex multipliers, those adders and the multiplexers can be mapped along with the hard multipliers. The structure of BF II is shown in Fig. 9.

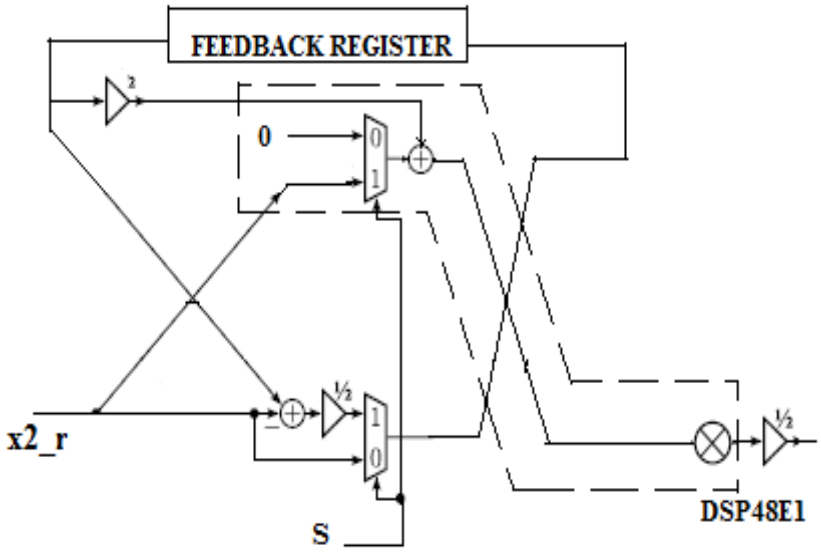


Fig 9. Real part of BF II

As it is a Single Path Delay Feedback architecture, one of the real and the imaginary parts of the butterfly unit's outputs are feedback to registers whose outputs are given back to that BF unit itself and it is done for the correct processing of an FFT and the feedback registers used are halved for each stages. The input datas are provided on the normal order and hence the outputs of the architecture would be in bit reversed order.

IV. SIMULATION RESULTS

18 bit fractional numbers are chosen as the coefficient and data values in its two's complement form. The ½ scaling in BF units can be obtained by right shifting the data by 1.The order of data processing in a $R2^2$ architecture is as shown in Fig. 10. The lower point units are first processed.

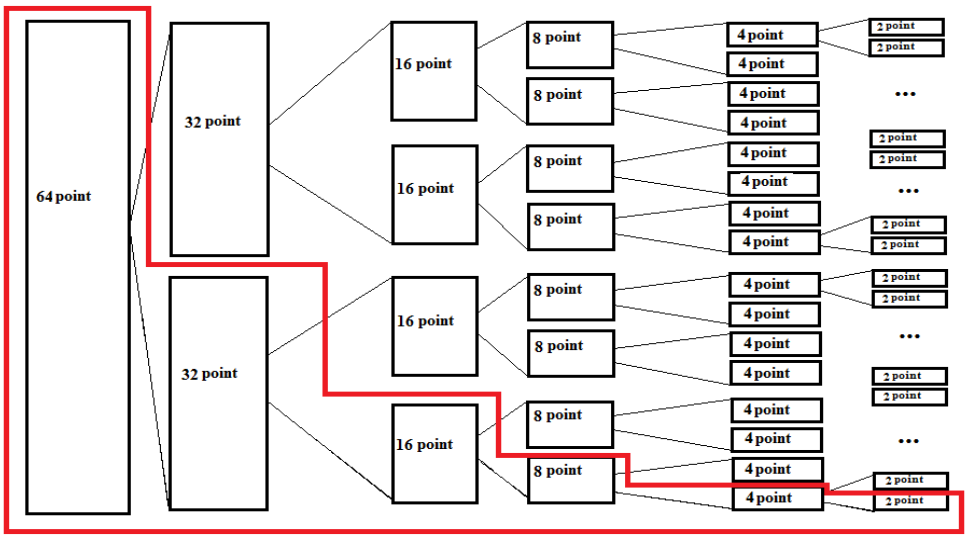


Fig. 10. Data processing order

The RTL schematic of Butterfly Unit is as shown in Fig. 11.

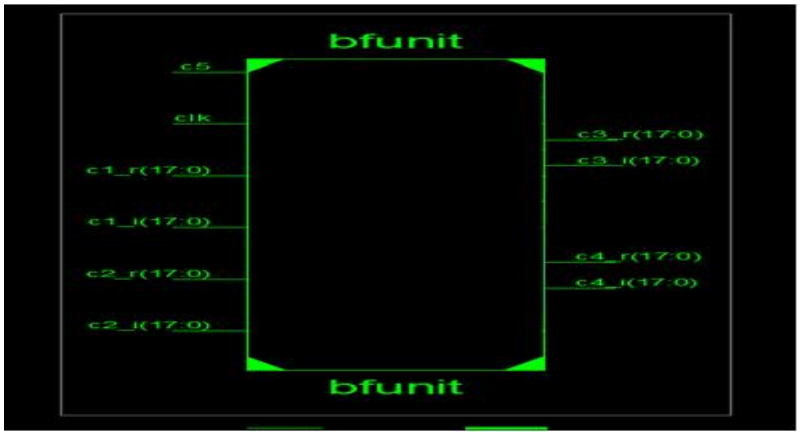


Fig 11. RTL schematic of Butterfly Unit

The RTL schematic of the architecture is shown in Fig. 12.

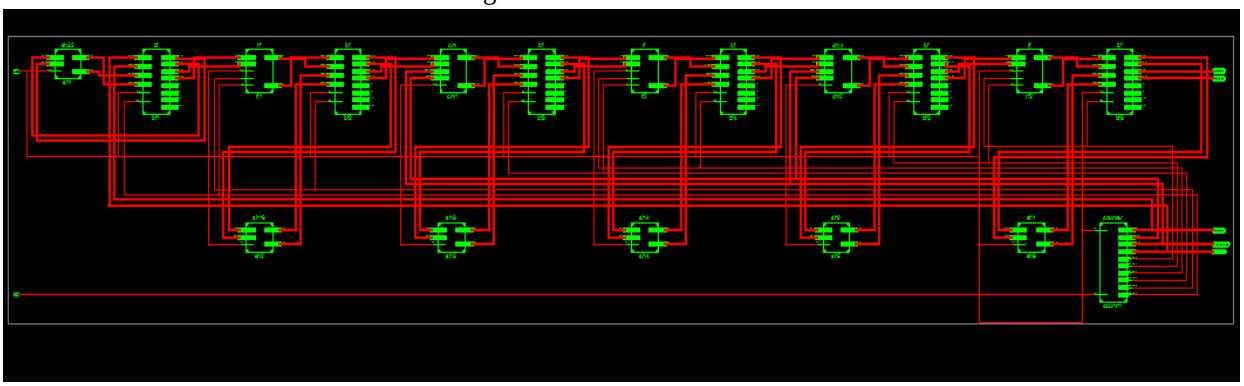


Fig 12. RTL Schematic of the architecture

The output can be obtained from the last count value of the first counting and the process ends before the second full counting of the counter. Output waveform of the architecture is shown in Fig. 13.c and d are the inputs and r and s are the outputs. Count represents the values of 6 bit counter. clk and rst are the clock and the reset respectively.

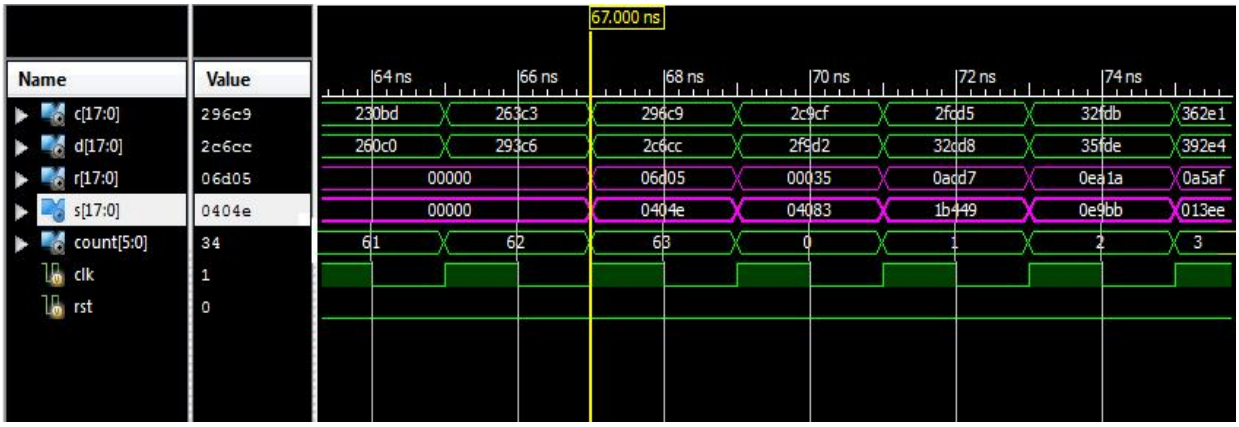


Fig 13. Output Waveform of the FFT architecture

The architecture is simulated on Xilinx ISE 14.2 and the targeted FPGA device is Virtex 6 xc6vlx365t-2 with speed grade -2 and at a maximum frequency of 167.259MHz. Figure 14 shows the simulation results of the optimized architecture.

POINT64VIRTEX6 Project Status (03/19/2018 - 09:57:49)			
Project File:	point64virtex6.xise	**Parser Errors:**	No Errors
Module Name:	POINT64VIRTEX6	**Implementation State:**	Synthesized
Target Device:	xc6vlx240t-2ff1156	**• Errors:**	No Errors
Product Version:	ISE 14.2	**• Warnings:**	4 Warnings (2 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	1414	301440	0%
Number of Slice LUTs	2024	150720	1%
Number of fully used LUT-FF pairs	1241	2197	56%
Number of bonded IOBs	80	600	13%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	8	768	1%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**

Fig 14. Simulation Results

The results of the 64 point DIF FFT architecture obtained are compared with the previous works and the performance analysis is given in the table II.

TABLE II
PERFORMANCE ANALYSIS

Parameters	Sarath et al.[2]	Ramesha et al.[1]	Proposed design
No. of Slices	4944	1991	1414
No. of bonded IOBs	1024	87	80
LUT-FF pairs used	0	3084	1241
DSP blocks	-	48	8
No. of used Logic	-	-	1546
No. used as memory	-	-	478
(ns) Minimum Period	7.666	10.333	5.979
Maximum Frequency(MHz)	-	96.778	167.259

Based on the simulation results, it is found that the hardware resources are efficiently mapped with an increase in maximum frequency and a decrease in minimum period compared with the previous works. Table 2 shows the results on 64 point FFT on FPGA devices along with the results of the proposed design. The performance analysis shows that, the proposed design has a percentage decrease of 28.98% of slices used compared with the paper [1] and 71.40% reduction compared with the paper [2], a percentage reduction of 92.18% on bonded IOBs when compared with the paper [2]. Only 8 DSP blocks are used in the proposed design, a percentage design of 59.76% on fully used LUT-FF pairs when compared with the paper [1]. And also, the minimum period is reduced to 5.979 and the maximum frequency is increased to 167.259.

V. CONCLUSIONS

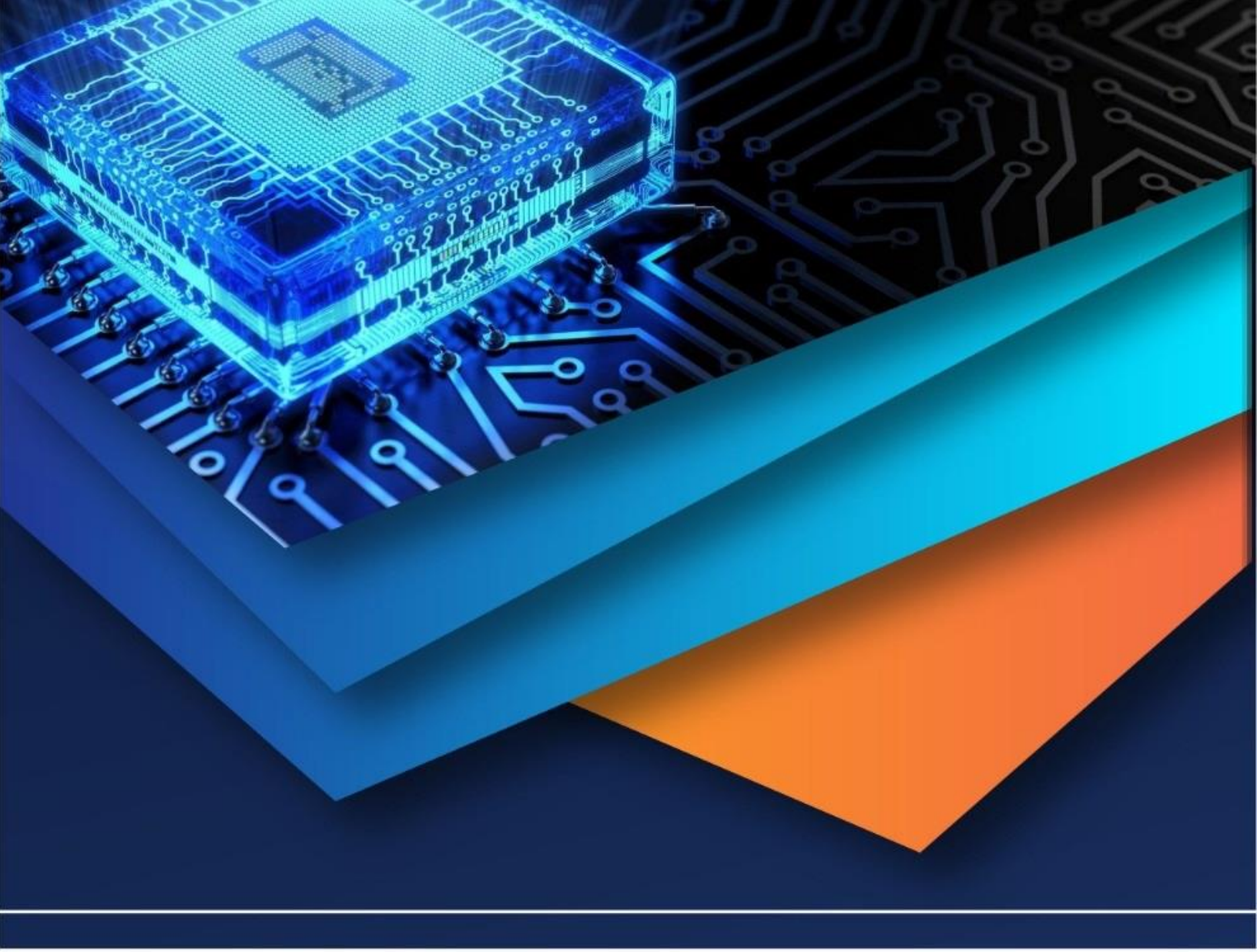
In this paper, some algorithmic transformations are proposed on a radix 2^2 architecture so that the hardware resources can be efficiently mapped. A radix 2^2 algorithm has the advantages of having a simpler structure as that of radix 2 but the same multiplicative complexity as radix 4 and the complex multiplication occurs only after two butterfly units which is a great structural advantage. The proposed optimization method gives the advantages of efficient usage of hardware resources with an increase in maximum frequency and a decrease in minimum period. In this work, only 8 twiddle factors are needed to be calculated and the work explains how the calculated twiddle factors can be used to find the remaining twiddle factors. As per the advantages of the proposed design, it can be efficiently used in wireless applications.

VI. AKNOWLEDGEMENT

The authors thank the faculties at TKM Institute of Technology for technical discussion and simulation support. The authors would also like to acknowledge the anonymous reviewers for their recommendations that helped in enhancing the presentation of this work.

REFERENCES

[1] Ramesha M, Dr. T. Venkata Ramana, “Design and Implementation of fully pipelined 64-point FFT Processor in a FPGA,” International Journal of Applied Engineering Research ISSN 0973-4562, Volume 11, Number 6(2016), pp 3940-3943
[2] K. Sarath Chandra Varma, Ch. Kranthi Kumar, K. Sravan Kumar, D. Sharat Chandra Varma, N. Rajasekhar, “Design and Simulation of 64-Point FFT Using Radix-4 Algorithm for OFDM Applications,” IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Volume 10, Issue 2, Ver. III (Mar – Apr.2015), PP 35-40.
[3] E. H. Wold and A. M. Despain, “Pipeline and parallel FFT processors for VLSI implementations,” IEEE Trans. Comput., vol. C-33, no.5. pp. 414-426, May 1984
[4] S. He and M. Torkelson, “A new approach to pipeline FFT processor,” in Proc. Int. Conf. Parallel Process., Apr. 1996, pp. 766-770
[5] B. Zhou, Y. Peng, and D. Hwang, “Pipeline FFT architectures optimized for FPGAs,” Int. J. Reconfigurable Comp., vol. 2009, pp. 1-9, Jun. 2009, Art. No. 219140
[6] H. Y. Lee and I. C. nPark, “Balanced binary-tree decomposition for area-efficient pipelined FFT processing,” IEEE Trans. Circuits. Syst. I, Reg. Papers, vol.54, no. 4, pp. 889-900, Apr. 2007
[7] M. Garrido, F. Qureshi, and O. Gustafson, “Low-complexity multiplierless constant rotators based on combined coefficient selection and shift-and-add implementation (CCSSI),” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 7, pp. 2002-
[8] Virtex-6 FPGA DSP48E1 Slice, User Guide (UG369), Xilinx, San Jose, CA, USA, Feb. 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY