Efficient VLSI Architecture for DIT and DIF Fast Fourier Transform using Real Valued Data

Neeraj Kumar Pandey
M. Tech. Scholar
Technocrats Institute of Technology – Advance
Bhopal (M.P.)

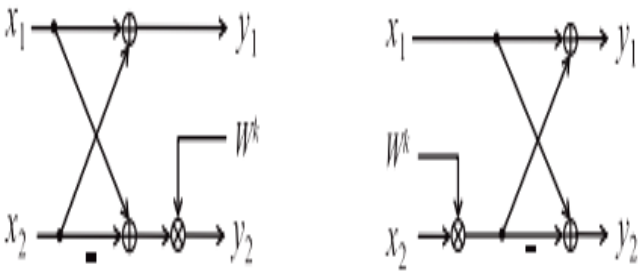
Pankaj Soni
Professor
Technocrats Institute of Technology – Advance
Bhopal (M.P.)

ABSTRACT
With the upcoming of new innovation in the fields of VLSI and correspondence, there is additionally a perpetually developing interest for fast preparing and low territory outline. It is additionally a verifiable truth that the chip range and most propagation time unit shapes a necessary piece of processor outline. Because of this respect, rapid and low zone designs turn into the need of the day. A fast fourier transform (FFT) is any quick calculation for figuring the DFT. The advancement of FFT calculations tremendously affected computational parts of flag handling and connected science. The decimation in-time (DIT) fast Fourier transform (FFT) all the time has advantage over the decimation in-frequency (DIF) FFT for most genuine esteemed applications, similar to discourse/picture/video handling, biomedical flag preparing, and time-arrangement examination, and so forth., since it doesn't require any yield reordering.

Keywords
FFT, MCPD, LUT, Decimation in Time, Decimation in Frequency, real Value data.

1. INTRODUCTION
The discrete fourier transform (DFT) and fast fourier transform (FFT) algorithm is frequently used in digital signal processing, wireless communication and speech recognition. Many types of techniques are implemented FFT algorithm. In recent year parallel and pipelined technique are used but consume more delay. Some collapsed pipeline models have been proposed for the calculation of RFFT where butterfly operations are multiplexed into a little rationale unit. The structures in and could give satisfactory throughput to a few applications yet the capacity multifaceted nature of those structures keeps on being high. Couples set up structures have additionally been proposed for RFFT utilizing particular pressing calculations. Memory-struggle for read/compose operation is observed to be the real test in the plan of calculations and structures for set up calculation. As of late, a set up design and strife free memory tending to conspire have been proposed for persistent preparing of RFFT. FFT calculations are grouped into two general classifications, to be specific, the obliteration in-time (DIT) and the devastation uncommonness (DIF) calculations.

The key contrasts between the two are appeared in Fig. 1. In the event of DIF calculation (Fig. 1(a)), the information tests are nourished to the processing structure in their regular request, while the yield is produced in bit-switched arrange. Then again, if there should arise an occurrence of DIT calculation (Fig. 1(b)), the info tests require bit-inversion reordering before being prepared, while the yield FFT coefficients are produced in characteristic request. In various RFFT applications, for example, picture and video preparing, biomedical flag handling, and time-arrangement investigation and so on., the entire information grouping is for the most part accessible together in the meantime for the FFT calculation. The DIT RFFT has favorable position over the DIF frame for these applications, since a DIT RFFT structure require not sit tight for the landing of info tests but rather can create the yields when those are figured.



(a) (b)

Figure 1: (a) DIF FFT butterfly (b) DIT FFT butterfly

2. PROBLEM FORMULATION
If an N – point DFT is implemented directly, the requirement of arithmetic operations is of the order of $O(N_2)$ that is N_2 multiplications and N (N-1) additions. The order of arithmetic operations for FFT is O (N log N) that is N log N additions and (N/2) log N multiplications.

- It efficiently computes inner products of vectors, which is a key requirement in many DSP systems. One of the key computational blocks in DSP is multiply/accumulate (MAC), which is implemented by a standard adder unit and a multiplier.
- Using DA, MAC unit can be implemented by pre computing all possible products and using a ROM to store them. The FFT of using DA is in its exponential increase of the size of ROM with increase in internal precision and number of inputs.

3. FFT ALGORITHM
A fast Fourier transform (FFT) is a calculation to process the discrete Fourier change (DFT) and it's opposite. Fourier examination changes over time (or space) to recurrence and the other way around; a FFT quickly registers such changes by factorizing the DFT network into a result of scanty (for the most part zero) elements.

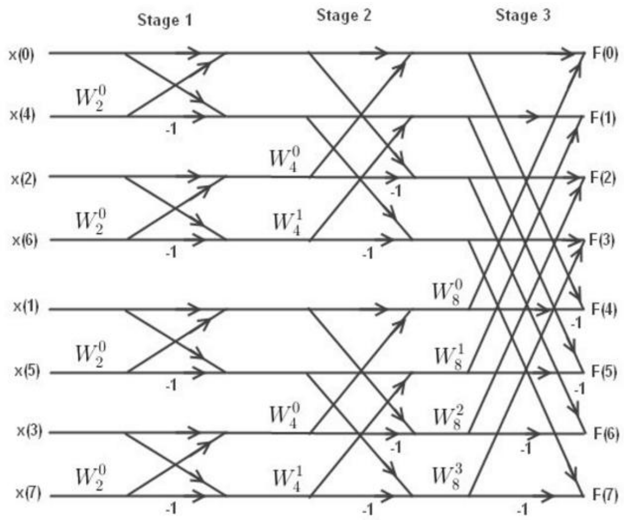


Figure 2: Radix-2 Decimation in Time Domain FFT Algorithm

As a result, fast Fourier transforms are widely used for many applications in engineering, science, and mathematics. Show the butterfly operations for radix-2 DIT FFT & DIF FFT in figure 2 and figure 3 respectively. The radix-2 algorithms are the simplest FFT butterfly algorithm.

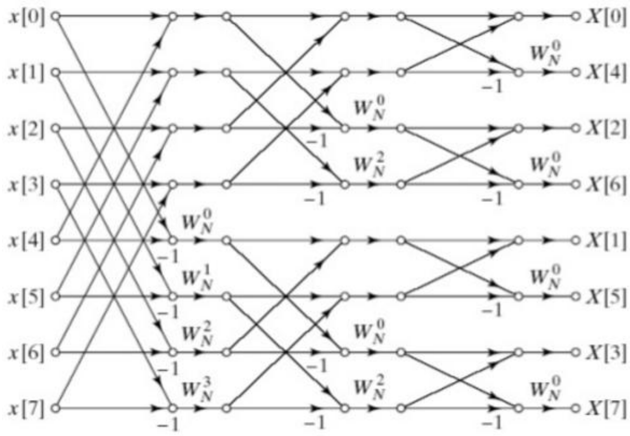


Figure 3: Radix-2 Decimation in Frequency Domain FFT Algorithm

4. PROPOSED METHODOLOGY

Complex augmentation is a more troublesome operation to comprehend from either a logarithmic or a geometric perspective. How about we do it mathematically to start with and we should take particular complex numbers to duplicate, say

$$A + j\,B = (A_1 + j\,B_1).\,(A_2 + j\,B_2) \qquad (1)$$

However the complex multiplication can be simplified:

$$A = (A_1 - B_2)\,.A_2 + Z \qquad (2)$$

$$B = (A_2 + B_2)\,.A_2 - Z \qquad (3)$$

With: $$Z = A_2.\,(A_1 - B_1) \qquad (4)$$

Signed Multiplier:-

Fig 4 shows the sign multiplier below:

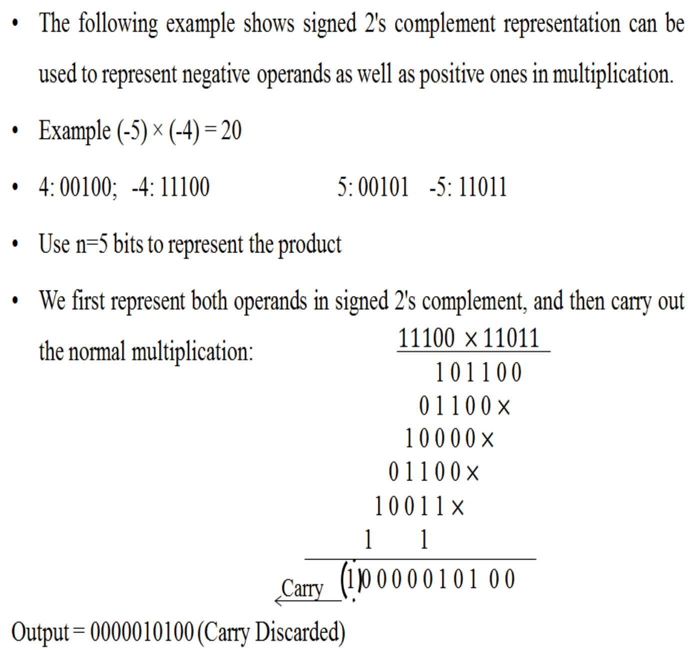


Figure 4: Signed Multiplier

5. SIMULATION RESULT

All the outlining and trial in regards to calculation that we have said in this paper is being created on Xilinx 6.2i refreshed rendition. The most recent arrival of ISETM (Integrated Synthesis Environment) plan instrument gives the low memory prerequisite estimated 27 rate low. ISE 6.2i that gives propelled devices like brilliant aggregate innovation with better use of their figuring equipment gives speedier planning conclusion and higher nature of results for a superior time to outlining arrangement. In the result fig 5, 6, 7, 8, 9 shows the Synthesis report, Timing summary, RTL view, output waveform of 8 bit multiplier addition operation respectively. Fig 10, 11, 12, 13 shows the Synthesis report, Timing summary, RTL view and output waveform of 8 bit addition multiplier operation respectively.

Table 1: Computational Delay of Multi-add And Add-multi Operations

Architecture	T_{MA}	T_{AM}	T_{AM}- T_{MA}	% Difference
		8-bit		
Previous Design	8.345	8.96	0.615	6.9%
Proposed Design	8.048	8.343	0.295	3.5%
		16-bit		
Previous Design	9.453	10.321	0.868	8.4%
Proposed Design	8.653	9.240	0.587	6.3%
		32-bit		
Previous Design	14.532	14.982	0.45	3.0%
Proposed Design	13.671	14.010	0.339	2.4%

```
=========================================================================
*                      Final Report                                     *
=========================================================================
Final Results
RTL Top Level Output File Name     : tma_8bit.ngr
Top Level Output File Name         : tma_8bit
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : NO

Design Statistics
# IOs                              : 32

Macro Statistics :
# Adders/Subtractors               : 1
#      8-bit adder                 : 1
# Multipliers                      : 1
#      8x8-bit multiplier          : 1

Cell Usage :
# BELS                             : 23
#      GND                         : 1
#      LUT2                        : 8
#      MUXCY                       : 7
#      XORCY                       : 7
# IO Buffers                       : 32
#      IBUF                        : 24
#      OBUF                        : 8
```

Figure 5: Synthesis Report of 8-bit multiplier addition Operation

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 8.048ns
```

Figure 6: Timing Summary of 8-bit multiplier addition Operation

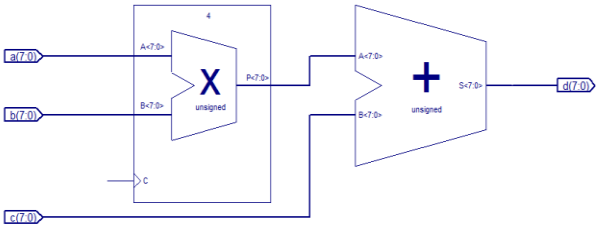


Figure 7: RTL view of 8-bit multiplier addition Operation

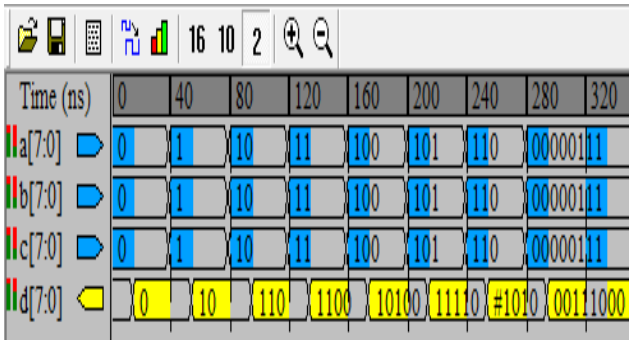


Figure 8: Output waveform of 8-bit multiplier addition Operation

```
=========================================================================
*                      Final Report                                     *
=========================================================================
Final Results
RTL Top Level Output File Name     : tam_8bit.ngr
Top Level Output File Name         : tam_8bit
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : NO

Design Statistics
# IOs                              : 32

Macro Statistics :
# Adders/Subtractors               : 1
#      8-bit adder                 : 1
# Multipliers                      : 1
#      8x8-bit multiplier          : 1

Cell Usage :
# BELS                             : 23
#      GND                         : 1
#      LUT2                        : 8
#      MUXCY                       : 7
#      XORCY                       : 7
# IO Buffers                       : 32
```

Figure 9: synthesis report of 8-bit multiplier addition Operation

```
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 2vp2fg256-7

 Number of Slices:                        4  out of   1408     0%
 Number of 4 input LUTs:                  8  out of   2816     0%
 Number of bonded IOBs:                  32  out of    140    22%
 Number of MULT18X18s:                    1  out of     12     8%

=========================================================================
```

Figure 10: Synthesis Report of 8-bit addition multiplication operation

```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 8.343ns
```

Figure 11: Timing Summary of 8-bit addition multiplication operation

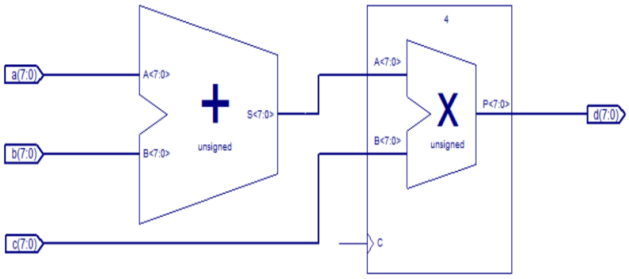


Figure 11: RTL view of 8-bit addition multiplier operation

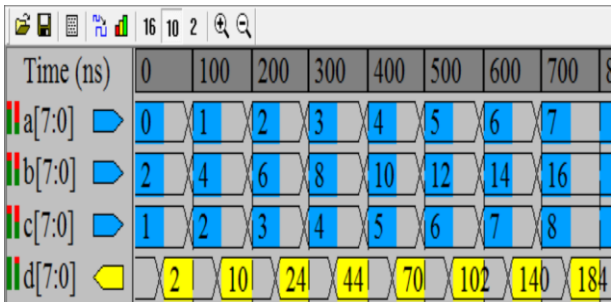


Figure 12: Output waveform of 8-bit addition multiplication operation

6. CONCLUSION

After the simulation work and implemented, we have found higher performance than the previous algorithm because less power consumed the previous algorithm in this paper all design all are implemented Xilinx 6.2i vertex2p device family and calculate the new parameter Slice, LUT (look up table), IOB (input output bounded), MPCD (maximum combinational path delay).

7. FUTURE SCOPE

In future, we can extend the work to variable length of 256 / 512 / 1024 / 2048 points etc. to improve the requirements of high throughput, consumption of low power and low area. Higher Mixed Radix number or Iterative Radix 2 can be employed to improve the percentage of utilization of hardware.

8. REFERENCES

[1] Pramod Kumar Meher, Basant Kumar Mohanty, Sujit Kumar Patel, Soumya Ganguly, and Thambipillai Srikanthan, *"*Efficient VLSI Architecture for Decimation-in-Time Fast Fourier Transform of Real-Valued Data*"*, IEEE Transactions on Circuits And Systems—I: Regular Papers, Vol. 62 , No. 12, December 2015.

[2] Kathir chandrabose K. and C. Paramasivam, "High throughput feed forward pipelined parallel architecture for FFT and IFFT", 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).

[3] Guohui Wang; Bei Yin; Inkeun Cho; Joseph R. Cavallaro; Shuvra Bhattacharyya and Jarmo Takala, "Efficient architecture mapping of FFT/IFFT for cognitive radio networks", 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[4] Antony Xavier Glittas; G. Lakshminarayanan, "Pipelined FFT architectures for real-time signal processing and wireless communication applications", 18th International Symposium on VLSI Design and Test Year: 2014.

[5] M. Ayinala, Y. Lao, and K. K. Parhi, "An in-place FFT architecture for real-valued signals," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 10, pp. 652–656, Oct. 2013.

[6] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda", pp. 01-06, Delhi (2011).

[7] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, "Implementation of Vedic multiplier for Digital Signal Processing", International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Joural of Computer Applications® (IJCA), pp.1-6.

[8] Sumit Vaidya and Depak Dandekar. "Delay-power performance comparison of multipliers in VLSI circuit design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.

[9] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced-bit Multiplication Algorithm for Digital Arithmetic", International Journal of Computational and Mathematical Sciences, Febrauary 2008, pp.64-69.

[10] Shashank Mittal, Md. Zafar Ali Khan and M.B. Srinivas, "Area Efficient High Speed Architecture of Bruun's FFT for Software Defined Radio", pp. 01-6, 2007 IEEE.

[11] B. G. Jo and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.

[12] Charles. Roth Jr., "Digital Systems Design using VHDL", Thomson Brooks/Cole, 7th reprint, 2005.

[13] Himanshu Thapaliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad, India 2004.

9. AUTHOR PROFILE

Mr. Neeraj Kumar Pandey has received his Engineering degree in Electronics & Communication on June 2012 from Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV), Bhopal, (M.P.) India and currently pursuing Master of Technology degree in Digital Communication from Technocrats Institute of Technology- Advance under Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV), Bhopal, (M.P.) India.

Prof. Pankaj Soni has received his Engineering degree in June 2008 and Master of Technology degree in Dec 2012 from Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV), Bhopal, (M.P.) India. He is currently working as Professor in department of Electronics & Communication in Technocrats Institute of Technology- Advance, Bhopal, (M.P.) India. He has five years teaching experience. He has published seven international research papers. His research interest is in Digital Communication, Wireless Communication and VLSI Design.