Design And Implementation Of Fast Fourier Transform With Reduced Complexity In Real-Time Systems

Naganaik Mudhavathu, Dr.P.Karpagavalli

Research Scholar, Associate Professor

Department of Electronics and Communication Engineering

Sri Satya Sai University of Technology & Medical Sciences, Bhopal Indore Highway, Pachama, Sehore (M.P.)-466001

Email : naganaik505@gmail.com

Abstract. The VLSI and Digital Signal Processing are the two aspiring and emergent domains for the development in technological understanding in the present world of electronics. Design of high performance, low area and low power VLSI circuits are needed for the DSP applications. This paper is dealing with FFT which is one of the most used algorithms in digital signal processing, FFT based on radix-2, radix-4 and split radix are most widely used for practical applications due to their simple architecture. This paper presents the 16, 32, 64 and 128 point radix-4 FFT algorithm. In FFT algorithm radix-4 can be used for any number of parallel samples which is a power of 2, furthermore, both decimation in frequency (DIF) and decimation in time (DIT) decompositions can be used. In addition to this, the designs can achieve a very high throughput, which makes them suitable for the most demanding applications.

Keywords: VLSI, FFT, decimation in frequency (DIF) and decimation in time (DIT), High Throughput

1. Introduction

The Discrete Fourier Transform (DFT) plays a significantly important role in many applications of digital signal processing. Basically, it has been applied in a wide range of fields such as linear filtering, spectrum analysis, digital video broadcasting and orthogonal frequency demodulation multiplexing (OFDM). The rapidly increasing demand of OFDM based applications, including modern wireless telecommunication such as LAN, needs real-time high speed computation in Fast Fourier Transform algorithm. This has made the design of FFT processor a critical requirement for the up-coming wireless technology [1]. With the advent of this requirement, the study of high performance VLSI FFT architecture is likewise of increasing importance. Many different hardware architectures have been proposed for the implementation of FFT algorithms. The main concern of the design approach will be power and architectural size. Among various FFT algorithms, radix-2 FFT with Cooley-Turkey algorithm, is very popular because it makes efficient use of symmetry and periodicity properties of the twiddle factor/coefficient for the exponential term is

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

Which reduce the computational complexity from (N^2) to ($N\log_2 N$)[2]. Several architectures have been proposed based on Cooley Turkey algorithm to further reduce the computation complexity, including radix-4, radix-2, and split-radix. Basically, this Fast Fourier Transform algorithm use Divide-and-Conquer approach to divide the computation recursively and then extract as many common twiddle factors as possible. The number of required real additions and multiplications is usually used to compare the efficiency of different FFT algorithms. In terms of the multiplicative comparison, the split-radix FFT is computationally better to all the other algorithms because it has most trivial multiplications [3]. Eventually, this algorithm has a drawback because of irregular structure that leads this algorithm not suitable for implementation on digital signal processors. Structural regularity is also important in implementation of FFT algorithms on dedicated chips such as in ASIC (Application Specific Integrated Chip). Hence, radix-2 and radix-4 FFT algorithm are preferable in terms of speed and accuracy. This paper presents an area and power efficient 16- point radix-4 Fast Fourier Transform. The approach in re-utilizing the stored identical component enhances the physical finger print of the architecture.

The 16 bit imaginary and 16 bit real input-output is realized at 1.8V with operating frequency of 50MHz. The chip is designed for fixed-point data format. Great care had been taken into account to overcome the overflow issue in fixed-point data format [4]. During the FFT computation, results at a particular stage are rounded and stored in the register memory. Since the FFT computation is an iterative process, the successive rounding errors at each output of butterfly accumulate over the FFT stages. The issue is solved by maintaining the error at the successive butterfly small. Twiddle factor/coefficient value are pre-calculated and stored in the register memory as 16-bit two’s complement signed fixed-point words. Fast computational schemes for the implementation of FFT architectures have fascinated many researchers.

Some of the approach to design FFT is memory-based, pipeline-based, and general-purpose DSP. Memory based is most area efficient but it needs many computation cycles. Pipeline-based architecture possesses small chip area and high throughput rate with a lower frequency. All fast Fourier transform designs can be isolated basically into three types of pipelined architectures—single-path delay commutator (SDC), multiple-path delay commutator (MDC), and single-path delay feedback (SDF). Among these architectures, the SDF architecture is more convenient as—(1) The SDF structure is more relevant for the fixed length FFT implementation and (2) SDF structure requires less number of delay elements as compared to SDC and MDC architectures.

2. Review Of Work

2.1 **Existing Algorithms:**

Radix 2 & Radix 4 Algorithms:
The N-point DFT of an input sequence is defined as:

$$X(K) = \sum_{n=0}^{N-1} x(n) * W_N^{nk} \qquad 0 \le K \le N-1$$

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

When N is a power of two, the FFT based on the Cooley –Tukey algorithm is most commonly used in order to compute the DFT efficiently. The Cooley – Tukey algorithm reduces the number of operations (N^2) for the DFT to ($N \log_2 N$) for the FFT. In accordance with this the FFT calculated in a series of $n= \log_p N$ stages where p is the base of the radix. The other popular algorithm is the radix-4 FFT, which is even more efficient than the radix-2 FFT.

The radix-4 FFT equation is listed below:

$$X(K) = \sum_{n=0}^{\frac{N}{4}-1} (x(n) + (-j)^k x\left(n+\frac{N}{4}\right) + (-1)^k x\left(n+\frac{N}{2}\right) + (j)^k x\left(n+\frac{3N}{4}\right)) W_N^{nk}$$

The radix-4 FFT equation essentially combines two stages of a radix-2 FFT into one, so that half as many stages are required (see Figure 1). Since the radix-4 FFT requires fewer stages and butterflies than the radix 2 FFT, the computations of FFT can be further improved. For example, to calculate a 16-point FFT, the radix-2 takes $\log_2 16=4$ stages but the radix-4 takes only $\log_4 16=2$ stages. Next, we discuss the numerical issue that arises from a finite length problem. Most people use a fixed-point DSP to perform the calculation in their embedded system because the fixed-point DSP is highly programmable and is cost efficient. The drawback is that the fixed-point DSP has limited dynamic range, which is worsened by the summation overflow problem that occurs all the time in FFT.

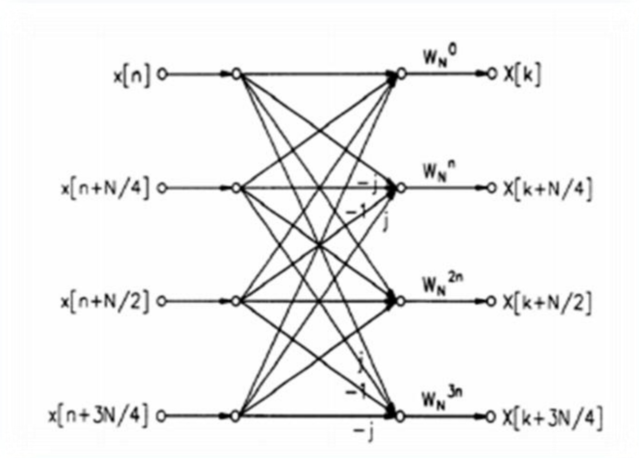


Figure 1: Basic Butterfly of a Radix-4 FFT algorithm

A scheme is needed to overcome this issue, hence which would lead to rearrangement of twiddle factors. Now to provide the rearrangement of twiddle factors we consider the mathematical deduction for DFT as shown below:

$$X(p,q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l,m) W_N^{(Mp+q)(NL+l)}$$

$$where\ W_N^{(Mp+q)(Np+q)} = W_N^{MLmp} W_N^{MLq} W_N^{MLp} W_N^{Lq}$$

$$however, W_N^{Nmp} = 1, W_N^{MqL} = W_{N/L}^{Mq} = W_M^{mq}\ and\ W_N^{Mpl} = W_{N/M}^{lp} = W_L^{lp},$$

Hence from the above representation we have,

$$X(p,q) = \sum_{l=0}^{l-1} \{W_N^{lq} [\sum_{m=0}^{M-1} x(l,m) W_N^{(mq)}]\} W_L^{lp}$$

{Reduced Form For Twiddle Factors}

$$X(4p+q) = \sum_{l=0}^{3} F(l,q) W_4^{lp}$$

Where p = 0,1,2,3 and F(l,q) is given by:

$$F(l,q) = W_N^{lq} \sum_{l=0}^{3} x(4m+1) W_4^{lp}$$

Given for the consideration, l= 0,1,,2,3 and q = 0,1,2,… N/4-1.

A comparison of the number of complex multiplications required for direct evaluation of the DFT and the number needed for FFT is given in below table-1.From the table-1 we can find the FFT algorithm is greater than 100 times faster than the direct evaluation for a 512-point DFT.

Number of Stages M	Number of Points N	Number of Complex Multiplications Using		Speed Improvement Factor $\frac{N^2}{(N/2)\log_2 N}$
		Direct Evaluation N^2	FFT algorithm $(N/2)\log_2^N$	
2	4	16	4	4
3	8	64	12	5.333
4	16	256	32	8
5	32	1024	80	12.8
6	64	4096	192	21.33
7	128	16384	448	36.57
8	256	65536	1024	64
9	512	262144	2304	113.77
10	1024	1048576	5120	204.8

Table-1: comparison of number of complex multiplications for the direct evaluation of the DFT versus the FFT algorithm

3. Proposed Methodology

3.1 Design implementation process:

We propose complete novel concept of improving an image compression method based on Fast Fourier Transformation. This method provides lossy and lossless compression of images both in grayscale and color. We analyze the amount of compression by compression ratio with the analysis of quality of image using Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

Considering the facts and its implication we here by estimate and realize the importance of FFT in compression techniques and its implementation using HDL for better hardware analyzation in real time systems. Our method, consist of two phases as shown below:

Phase1:

1. We mathematically analyze and approach a model for designing the FFT & IFFT based compression techniques.
2. To provide the better performance of compression technique we estimate the complexity and also its futuristic model approach.
3. As per the different compression model we consider both cases lossy and lossless its performance estimation such as MSE, PSNR and compression ratio (CR).

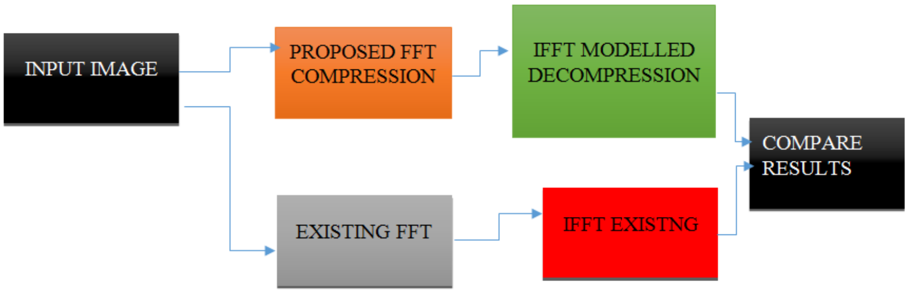


Figure 2 : Phase 1- Block Diagram For Proposed Application

Phase 2:

1. The proposed model for FFT, with reduced complexity is modeled and applied in Hardware description modeling for better characteristics related to performance apart from the MSE and PSNR.
2. In our design we analyze the FFT and IFFT for usage of Obfuscation of functional, coding and modeling.
3. Now, considering a specific application where most of the DSP design modules are being used and its design/functional security is key aspect.
4. We propose a black box which would provide basic operation in DSP as top module, considering the fact that the other design modules as obfuscation either by code, functionality or modeling.
5. Our aim is to provide a better and suitable way of securing the functionality and coding aspects of the design models so that these cannot be hacked or interfered by outsourcing.
6. Also we use digital secure algorithms for obfuscating the modules depending upon user interest. Requirements of the users would vary depending the type of application being used in the design. As per the proposed modeling criteria we utilize the DSP domain aspects such as filters, scaling, sampling, processor, controller etc…
7. Based on the above modules consideration we finally design and estimate its area, power and delay analysis on the respective hardware for better performance.

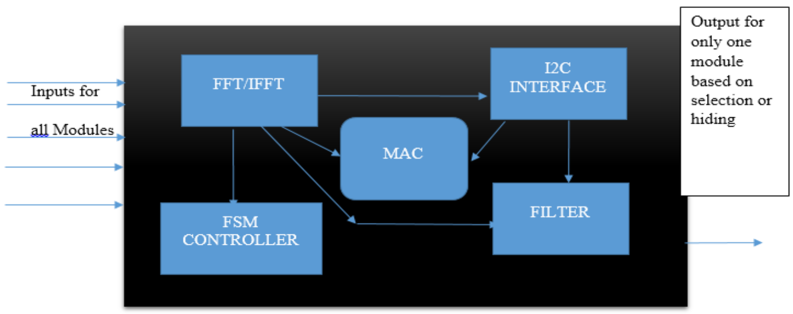


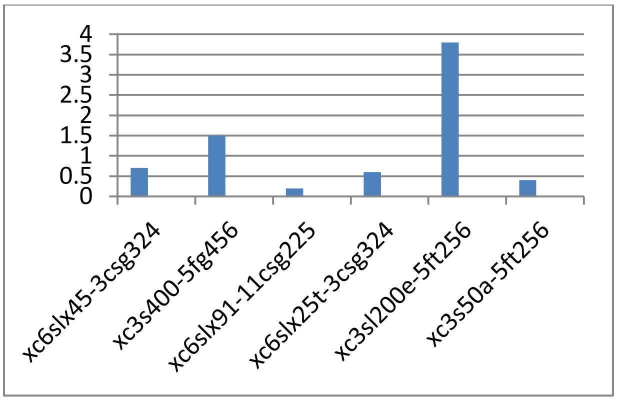
Figure 3 : Phase 2- Block Diagram For Proposed Design

4. Simulation Results

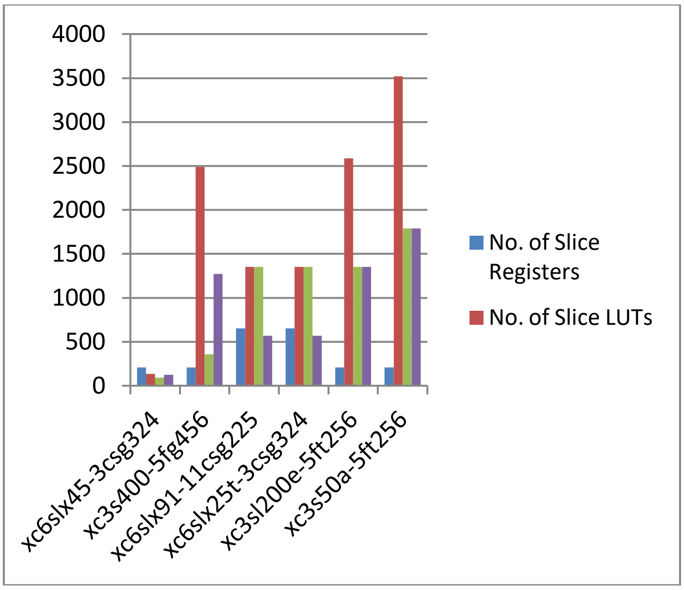
The presented architectures has been programmed for the use in FPGAs. The target FPGA is a Spartan6 FPGA graph 1 compares the power in different Spartan6 families as was shown in table 2 and graph 2 compares the area in different Spartan6 families as was shown in table 3

FPGA Families	Power
xc6slx45-3csg324	0.859
xc3s400-5fg456	1.48
xc6slx91-11csg225	0.273
xc6slx25t-3csg324	0.673
xc3sl200e-5ft256	3.789
xc3s50a-5ft256	0.258

Table 2: Power Analysis



Graph 1: Comparison of Power



Graph 2: Comparison of Area

Parameters	No. of Slice Registers	No. of Slice LUTs	No. of Used as Logic Gates	No. of LUT Flip Flop Pairs used
xc6slx45-3csg324	210	137	90	125
xc3s400-5fg456	210	2489	356	1272
xc6slx91-11csg225	654	1354	1354	570
xc6slx25t-3csg324	654	1354	1354	570
xc3sl200e-5ft256	210	2589	1352	1352
xc3s50a-5ft256	210	3520	1790	1790

Table 3: Area Analysis

5. Conclusion

Simulation results shows proposed FFT radix-4 represents a better and efficient architecture for computing FFT. This design facilitates the efficient computation of long FFT which usually require a huge architecture. In this design process, many identical components are being reused in which it reduces the gate count of the design. This is due to simplification of the mathematic algorithm in FFT structure. The comparison shows that the chip can reach low cost and low power for this system applications.

References

1. Manish Bansal and Sangeeta Nakhate, 2018. High-Performance Pipelined FFT Processor Based on Radix-2^2 for OFDM Applications : Intelligent Computing and Information and Communication, Advances in Intelligent Systems and Computing : Page No. 143-151.

2. James B. Birdsong and Nicholas I. Rummelt ,2016. The Hexagonal Fast Fourier Transform : IEEE: Page No. 1809-1812.

3. V.S. Tutatchikov, 2016. Two-dimensional fast Fourier transform, Butterfly in analog of Cooley-Tukey algorithm, IEEE: Information and Communication Technologies: Page No.: 495-498.

4. E. Cetin, R. C. S. Morling, and I. Kale, 1997. An integrated 256-point complex FFT processor for real-time spectrum analysis and measurement: IEEE Proc. of Instrumentation and Measurement Technology Conference'97: Page No. 96-101.

5. M. El-Khashab, and E. E. SwartzlanderWegmuller, 2003. An architecture for a radix-4 modular pipelined Fast Fourier Transform: Proc. of the Application Specific Systems, Architectures and Processors (ASAP'03).

6. M. Hasan, T. Arshan, and J.S. Thomson, 2003. A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN applications: IEEE Trans. of Consumer Electronics: Page No.128-134.

7. S. Xin, Z. Tiejun, and H. Chaohuan, 2004. A Highperformance power-efficient structure of FFT (Fast Fourier Transform) processor: International Conference of Signal Processing: Page No. 555-558.

8. C. Meletis, P. Bougas, G. Economakos, P. Kalivas, and K. Pekmestzi, 2004. High-speed pipelined implementation of radix-2 DIF algorithm: Trans. of Engineering, Computing and Technology - ENFORMATIKA'04: Page No. 305-308.

9. V. J. Alberto, R. R. Jesus, and O. Alejandro, 2005. VHDL core for 1024-point radix-4 FFT computation: Proc. of the 2005 International Conference on Reconfigurable Computing and FPGAs -ReConFig 2005.

10. N. Weste, M. Bickerstaff, T. Arivoli, P. J. Ryan, J. W.Dalton, D. J.Skellern and T. M. Percival, 1997. A 50MHz 16-point FFT processor for WLAN application: IEEE 1997 Custom Integrated Circuits Conference: Page No. 457-460.

11. R. Sambasiva Nayak, MD.Javeed Ahammed "CMOS/VLSI Circuit for Power Optimization on Portable Devices,"Iinternational Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278 -3075, Volume-8, Issue-12, October-2019, and PP: 4300-4303. DOI: 10.35940/ijitee.L2716.1081219. (Scopus Journals).

12. W. Chang, and T. Nguyen, 2006. An OFDMSpecified Lossless FFT Architecture: IEEE Transaction on Circuits and Systems – I: Page No. 1235- 1243.

13. Ahmed Saeed, M. Elbably, G. Abdelfadeel and M. I. Eladawy, 2009, "FPGA implementation of Radix-22 Pipelined FFT Processor," Proceedings WAV'09 Proceedings of the 3rd WSEAS international symposium on Wavelets theory and applications in applied mathematics, signal processing & modern science, Page No. 109–114.

14. Chih-Peng Fan1, Mau-Shih Lee and Guo-An Su, Vol. 19, No.2, 2008 ,"Efficient low multiplier cost 256-point FFT design with Radix-24 SDF architecture," Journal of Engineering, National Chung Hsing university, Page No. 61–74.

15. Sivaprasad Godugunuri, Sambasiva Nayak.R, "Low-Power Differential SRAM design for SOC based on the 25-um Technology" published in the IOP Conference Series: Materials Science and Engineering, IOP Publishing Ltd, UK (Scopus Journals).

16. Jiang Wang and Leif Arne Ronningen, "An Implementation of Pipelined Radix-4 FFT Architecture on FPGAs," Journal of Clean Energy Technologies, Vol. 2, No. 1, January 2014, Page No. 101–103.

17. Dr. V. Krishnanaik, Purshotham,2015. Digital Signal Processing. Lap Lambard Publishing German.

18. Manuel, B. R., Konguvel, E., & Kannan, M., 2017. " An area efficient high speed optimized FFT algorithm ". 2017 Fourth International Conference on Signal Processing, Communication and Networking - ICSCN. Page No.

19. Chundi Vedasri, SambasivaNayak.R "Design of low power consumption of 8 bit multiplier Using NS Gate" Published in the International Journal of Advanced Scientific Technologies in Engineering and Management Sciences (IJASTEMS-ISSN: 2454- 356X) Volume.3, Issue.4, April.2017 www.ijastems.org Page 6 -8. UGC and ISSN Approved Journal.

20. M. Jiang, B. Yang, Y. Fu, A. Jiang, X. Wang, X. Gan, B. Zhao, and T. Zhang, 2004. Design of FFT processor with low power complex multiplier for OFDM-based high-speed wireless applications: Symp. ISCIT'04: Page No. 639-641.

21. J G. Proakis and D. G. Manolakis, 1996. Digital Signal Processing Principles, Algorithm, and Applications. Prentice Hall, Page No. 448-475.

22. Sivaprasad.G, Naveen.D, SambasivaNayak.R "Low-Power Differential SRAM design for SOC based on the 25-um Technology" published in the International Journal OF ECT (IJECT) VOL 7, Issue 4, Oct 2016-Dec 2016. ISSN: 2230-7109 & 2230-9543. www.iject.org Page 48-51. UGC and ISSN Approved Journal.

23. M. Kannan, E. Konguvel, J. Madhumitha, M. Mohammed Kamil Ashiq and K. Abhishek, " FPGA Implementation of FFT Architecture using Modified Radix-4 algorithm," in Asian Journal of Research in Social Sciences and Humanities, vol. 7, no. 2, Page No. 47-57, February 2017.

24. R. Samba Siva Nayak, Suman.J "FPGA Implementation of Convolutional Interleaver & De-interleaver", Published in the International Journal of Engineering and Social Science (IJESS), ISSN: 2249-9482 Volume-1, Issue-2, January 2012, pp: 96-102.