8-POINT RADIX-8 PIPELINED FFT USING SDF

P.Geetha
ECE Department (M.E-EST), Karpaga Vinayaga College of Engineering and Technology,
G.S.T Road, Mathuranthagam, Kancheepuram District, Tamil Nadu, India.
Kannanramgb@gmail.com

A.Manimaran
Associative Professor (ECE Dept),
Karpaga Vinayaga College of Engineering and Technology, G.S.T Road, Mathuranthagam,
Kancheepuram District, Tamil Nadu, India.Email: lims0181@gmail.com

Abstract--Fast Fourier Transform (FFT) operation is one of the most important fundamental operations in the digital signal processing systems. A new pipelined Radix-8 based Fast Fourier Transformation (FFT) architecture is designed for performing frequency transformation techniques. The objective of this design is to improve the speed and to reduce the area, delay and power. Radix-8 FFT, which is used to improve the speed of functioning by reducing the computational path. In the proposed new architecture named as "Radix-8 SDF FFT". In this architecture, the numbers of stages are reduced to 75%. The SDF FFT is to increasing the processing speed of architecture. The performance evaluation of Radix-8 SDF FFT architecture is determined through Very Large Scale Integration (VLSI) system design environment. In the VLSI system design, less area utilization, low power consumption and high speed are the main parameters. Hence, the main goal of proposed architecture is to reduce hardware complexity, power consumption and increasing both speed and throughput of the system. Applications: Mobile Ad-hoc Network (MANET), Orthogonal Frequency Division Multiplexing (OFDM) System.

Keywords-- Fast Fourier Transform (FFT), Single-path Delay Feedback (SDF) FFT, Very Large Scale Integration (VLSI)

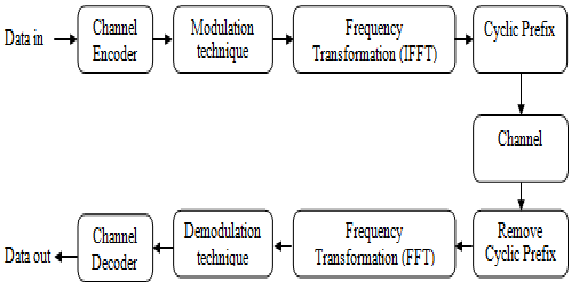
1. Introduction

In digital signal processing FFT (Fast Fourier Transform)plays a crucial role in various applications like spectralanalysis, filter simulation, auto correlation and patternrecognition for the computation of DFT (Discrete FourierTransform) . FFT also plays a critical role in modern digitalcommunications such as digital video broadcasting andorthogonal frequency division multiplexing (OFDM) systems. Though FFT reduces the computation time and improvesthe performance largely over the direct evaluation of DFT ,theFFT core is one of the modules having high computationalcomplexity in the physical layer of these communicationsystems. Therefore, FFT architecture is presented forimplementing the FFT computation efficiently. Variousalgorithms like radix4 , split radix are based on radix2FFT approach. FFT architectures likeR2MDC (Radix2 Multipath Delay Commutator), R2SDF(Radix 2 Single path Delay Feedback), R4MDC (Radix4Multipath Delay Commutator), R4SDF (Radix 4 Single pathDelay Feedback) are designed for Radix2 and

Radix4algorithm earlier. In these architecture, hardwares requiredare not well established and also the complexity is high. Toachieve the efficient implementation of the FFT with reducedhardware complexity and power consumption.

2. OFDM

OFDM is a Frequency Division Multiplexing scheme used as a digital multi-carrier modulation method. A large number of closely spaced orthogonal sub-carrier signals are used to carry data on several parallel data streams or channels. Each sub-carrier is modulated with a conventional modulation scheme (such as phase-shift keying) at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth



2.1 OFDM Block Diagram

OFDM is a specialized FDM, the additional constraint being all the carrier signals are orthogonal to each other. The sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not required. This greatly simplifies the design of both the transmitter and the receiver, unlike conventional FDM, a separate filter for each sub-channel is not required.

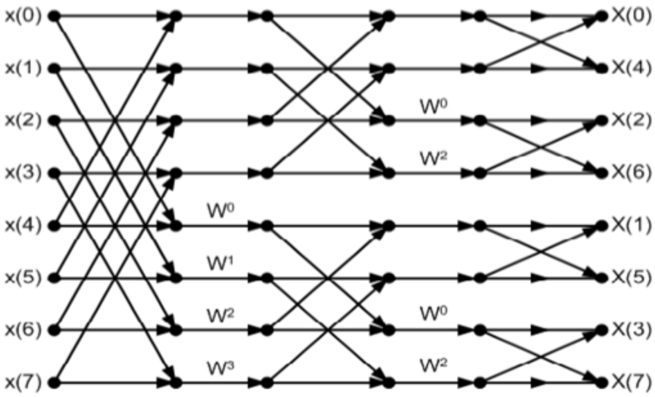
3. Basic FFT Structures

In general, FFT is used to convert the time domain representation of sampled signal into frequency domain representation of sampled one. Two types of FFT structures are available for performing those transformations are (1) Decimation in Time (DIT) - FFT and (2) Decimation in Frequency (DIF) - FFT. The Radix-2 FFT architecture is based on Cooley – Tukey algorithm and it's also called the butterfly diagram.

3.1 Butterfly diagram of 8 point FFT

In Radix-2 computation, N-point DFT (Discrete Fourier Transformation) is represented as follows:

$$X(k)=1/\mathrm{N} \sum_{n=0}^{N-1} \mathrm{x(n)}\ \mathrm{e}^{-\mathrm{j}2\pi\mathrm{n}/\mathrm{N}} \qquad \text{For} \quad \mathrm{n}=0,1,2\ldots\mathrm{N}-1$$



Similarly, N-point Inverse Discrete Fourier Transformation (IDFT) is represented as follows:

$x(n) = \sum_{k=0}^{N-1} X(K)e^{-j2\pi Kn/N}$, for n=0,1,2…N-1

The Fast Fourier Transform (FFT) was proposed by Cooley and Tukey to efficiently reduce the time Complexity to O (N log 2N), where N denotes the FFT size and is represented as

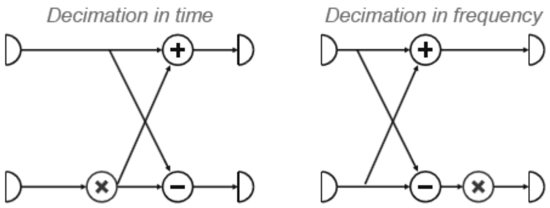
$X(k)= \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi n/N}$, for $0 \le k \le N-1$

The IFFT is represented as

$$x(n) = 1/N \sum_{k=0}^{N-1} X(K){W_N}^{-nk}, \text{for } n=0,1,2\ldots N-1$$

$$W_N = e^{-J2\pi/N}$$

FFT technique to determine the spectral characteristics of discrete sampled signals. Two types of FFTs (Decimation in Time (DIT) FFT and Decimation in Frequency (DIF) FFT) are involved to find the frequency transformation of sampled signals.



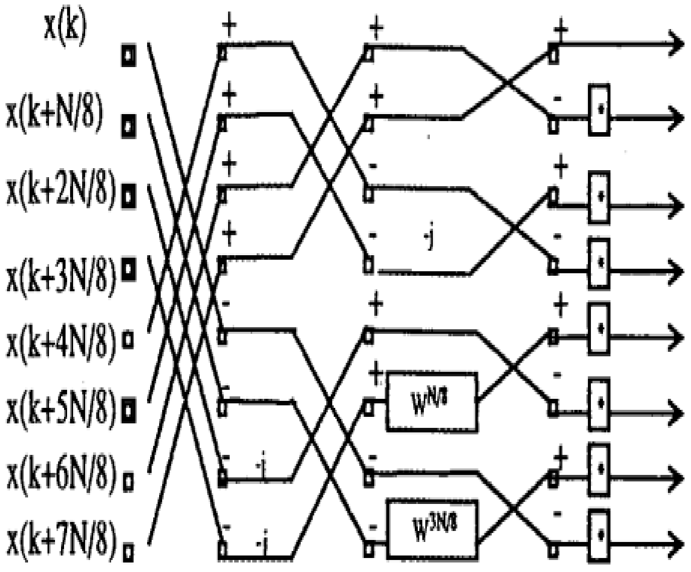
3.2 Decimation in time and frequency diagram

In Decimation-in-time(DIT) FFT, the butterfly is multiply before add/subtract and Decimation-in-frequency(DIF)FFT butterfly is multiply after add/subtract.

Both are radix-2type: problem size is reduced by 2 at each stage. In this project the radix -2 64 point FFT is implemented in Decimation-in-Frequency.

4. Radix-8 FFT Architecture

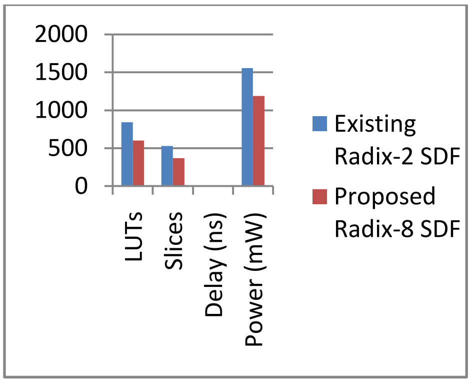
Radix-8 FFT algorithm which was surveyed to improve the speed of functioning by reducing the computation, it can be obtained by changing to base to 8. For a same number if base increases the power will reduce. FFT algorithms using higher radix can be designed by decomposition of the frequency domain samples into more groups at the cost of more complicated control.A radix-8 butterfly can also be realized by cascading three radix-2 stages, which is called radix-2^3 algorithm. Radix-r FFT can easily derived from DFT by decomposing the N point DFT into a set of recursively related r-point transform and x(n) is power of r. In Radix-8 algorithm the r is 8. The DIT Radix-8 FFT recursively partitions a DFT into eight quarter-length DFTs of groups of every eighth sample. The outputs of these shorter FFTs are reused to compute many outputs, which greatly reduce the total computational cost. The Radix-8 Decimation-In-Time and Decimation-In-Frequency Fast Fourier Transform (FFTs) gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs. Fig.1 shows the basic structure of Radix-8 butterfly unit. A radix-8 FFT algorithm can be constructed by following the same techniques used to convert the classical radix-2 and radix-4 algorithms



Signal flow graph for Radix-8 butterfly unit. The butterfly unit is designed to perform basically the radix-8 DIF FFT algorithm and also it can compute radix-4 or radix-2 DIF FFT algorithm. It computes the radix-8 FFT in all computational stage.

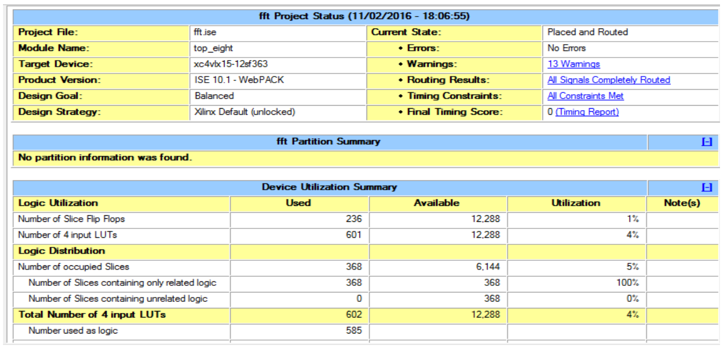
5. Comparison Table

Parameters	Existing Radix-2 SDF	Proposed Radix-8 SDF
LUTs	840	601
Slices	528	368
Delay	10.455ns	10.232ns
Power	1.556W	1.188W



6. Experiental Results

(A) Area

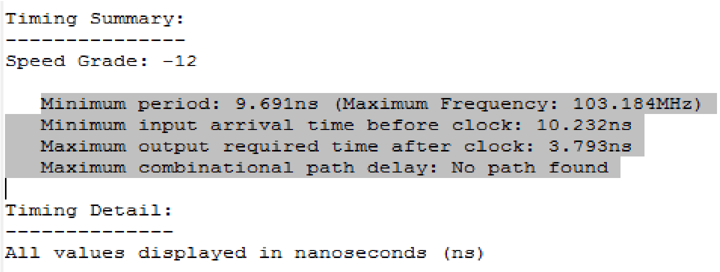


fft Project Status (11/02/2016 - 18:06:55)			
Project File:	fft.ise	**Current State:**	Placed and Routed
Module Name:	top_eight	• **Errors:**	No Errors
Target Device:	xc4vlx15-12sf363	• **Warnings:**	13 Warnings
Product Version:	ISE 10.1 - WebPACK	• **Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	• **Timing Constraints:**	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	• **Final Timing Score:**	0 (Timing Report)

fft Partition Summary	[-]
No partition information was found.	

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	236	12,288	1%	
Number of 4 input LUTs	601	12,288	4%	
Logic Distribution				
Number of occupied Slices	368	6,144	5%	
Number of Slices containing only related logic	368	368	100%	
Number of Slices containing unrelated logic	0	368	0%	
Total Number of 4 input LUTs	602	12,288	4%	
Number used as logic	585			

(B) Delay

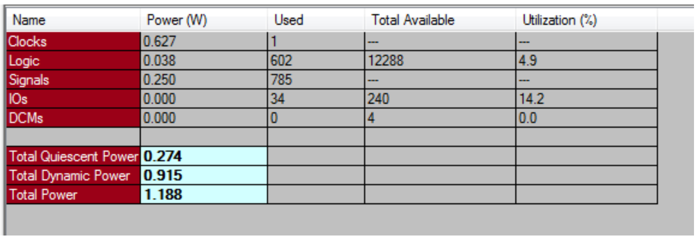


```
Timing Summary:
---------------
Speed Grade: -12

   Minimum period: 9.691ns (Maximum Frequency: 103.184MHz)
   Minimum input arrival time before clock: 10.232ns
   Maximum output required time after clock: 3.793ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)
```

(C) Power



Name	Power (W)	Used	Total Available	Utilization (%)
Clocks	0.627	1	---	---
Logic	0.038	602	12288	4.9
Signals	0.250	785	---	---
IOs	0.000	34	240	14.2
DCMs	0.000	0	4	0.0
Total Quiescent Power	**0.274**			
Total Dynamic Power	**0.915**			
Total Power	**1.188**			

7. Conclusion

The conclusion of this design is to improve the speed and to reduce the area, delay and power. Radix-8 FFT, which is used to improve the speed of functioning by reducing the computational path. In the proposed new architecture named as “Radix-8 SDF FFT”. In this architecture, the numbers of stages are reduced to 75%. The SDF FFT is to increasing the processing speed of architecture. The performance evaluation of Radix-8 SDF FFT architecture is determined through Very Large Scale Integration (VLSI) system design environment. In the VLSI system design, less area utilization, low power consumption and high speed are the main parameters. Hence, the main goal of proposed architecture is to reduce hardware complexity, power consumption and increasing both speed and throughput of the system. Applications: Mobile Ad-hoc Network (MANET), Orthogonal Frequency Division Multiplexing (OFDM) System.

8. References

[1]. Sudha Kiran G , Brundavani P, “FPGA Implementation of 256-Bit, 64-Point DIT-FFT Using Radix-4 Algorithm ” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013.

[2]. K.Sowjanya,Leela Kumari Balivada, , “ Design and Performance Analysis of 32 and 64 Point FFT using Multiple Radix Algorithms”, International Journal of Computer Applications (0975 – 8887) Volume78– No.1, September 2013.

[3]. C.Vennila, G.Lakshminarayanan, Seok-Bum Ko, “Dynamic PartialReconfigurable FFT for OFDM based Communication Systems”,*Springer Circuit System Signal processing*, p 1-18, October 2011.

[4]. Y.-N. Chang, "An efficient VLSI architecture for normal I/O orderpipeline FFT design", *IEEE Trans. Circuits Syst. II*, *Exp. Briefs*, Vol.55, No. 12, pp. 1234-1238, Dec. 2008.

[5]. Lenart and Viktor Öwall. Architectures for dynamic data scaling in2/4/8k pipeline FFT cores. IEEE transactions on Very Large ScaleIntegration (VLSI) systems, vol. 14, no. 11, Pages: 1286-1290,November 2006.

[6]. Mian Sijjad Minallah, Gulistan Raja. Real Time FFT ProcessorImplementation. IEEE—ICET 2006 2nd International Conference onEmerging Technologies.Peshawar, Pakistan 13-14, Pages:192-195, November 2006.