International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||10||Pages||1473-1476||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Design a Parallel Pipeline Radix-4 FFT Architecture

[1]Amaresh Kumar,[2]Dr. Manish Mishra , [3]Roopak Kumar Verma

1,2,3 Departments of Electronics DDU Gorakhpur University Gorakhpur
Email: mmanishm@yahoo.com, amaresh.704@rediffmail.com, roopakdeepak@gmail.com

Abstract

This paper presents a novel design to develop parallel pipelined architectures for the fast Fourier transform (FFT). The architecture is based on the radix-4 algorithm. By exploiting the regularity of the algorithm, butterfly operation and multiplier modules were designed. The architecture adopts four butterflies, and the pipeline stage is optimized to balance the processing speed and the area. Novel parallel pipelined architectures for the computation of complex and real valued fast Fourier transform are derived.

Introduction

The fast Fourier transform (FFT) class of algorithms is widely used in communication and digital signal processing. The FFT algorithm is considered one of the basic algorithms in many DSP projects. Nowadays, FFT is the key building block for the mobile communications; especially for the orthogonal frequency division multiplexing (OFDM) transceiver systems. Implementation of FFT of different architectures, for fast and efficient computational schemes, has attracted many researchers. The methodology of FFT simulation, implementation, and verification plays a key role in the industrial or consumer electronics areas, for example, the FFT image or acoustic processing, encoding and decoding, harmonic analysis in renewable energy and so on. The FFT is a typical computation where the memory access intensively and the high parallelism is needed. VLSI realization of FFT algorithm, should have pipelined architecture and/or parallelism, be regular and modular. At algorithm level, it should achieve the multiplicative complexity as low as possible. At the architecture level, use the delay-feedback buffering strategy to minimize the memory size. It should have modular and regular modules, local routing, and low control complexity.

FFT ALGORITHM

The Discrete Fourier Transform (DFT) for an N point input signal $x(n)$ is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W^{kn}_{N}$$

Where k=0, 1,…..N-1 and the twiddle or rotation or phase factor is given by

$$W^{kn}_{N} = e^{-j2\pi k n/N}$$

Here *X(k)* represents the frequency domain values obtained from the Fourier transform on *x(n)*. DFT computes X(K) from x(n).The direct computation of DFT is very complex and because of this FFT algorithm is used. To reduce the number of computations FFT algorithm uses a divide-and-conquer Approach recursively. Radix-2 is an FFT algorithm in which divide by 2 approach is recursively used. Apart from radix-2 algorithm there are other algorithms like radix-4, radix-8 and other higher radix algorithms where there are only less intermediate stages when compared to radix-2 algorithm. Radix-4 is an FFT algorithm which is used when N in DFT is a power of 4.This algorithm helps in reducing the number of multiplications but increases the number of additions.

Radix-4 FFT Algorithms- N - point input signal are decompose

X(k) = $\sum_{n=0}^{N/4-1}[x(n) + x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) + x(n+3N/4)]W^{0}_{N}\, W^{kn}_{N/4}$

X(k+1) = $\sum_{n=0}^{N/4-1}[x(n) - j\,x\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + jx(n+3N/4)]W^{n}_{N}\, W^{kn}_{N/4}$

X(k+2) = $\sum_{n=0}^{N/4-1}[x(n) - x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) - x(n+3N/4)]W^{2n}_{N}\, W^{kn}_{N/4}$

X(k+3) = $\sum_{n=0}^{N/4-1}[x(n) + j\,x\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - jx(n+3N/4)]W^{3n}_{N}\, W^{kn}_{N/4}$

Where k = 0,1 ----N/4 – 1 and used the property $W^{4kn}_{N} = W^{kn}_{N/4}$

FFT Architecture

FFT processors can be divided into three main classes:

I. Pipeline FFTs. They utilize concurrent processing of different stages to achieve high throughput.

II. Column FFTs. Each stage in the FFT is computed with a set of processing elements and the result is fed back to the same processing elements for the computation of the next stage.

III. Fully parallel FFTs. The operations in the signal-flow graph are mapped isomorphically to a hardware structure. The implementations are hardware intensive and are with current technology not practical for large FFTs.

Pipeline FFT/IFFT Processor Architecture:

The pipeline architecture for FFT/IFFT processors has been studied since 1970's. They have features like simplicity, modularity and high throughput. Those features are suitable for real-time applications. Pipeline FFTs are a class of parallel algorithms that contain an amount of parallelism equal to log $_r$ N where N is the number of points for an FFT and r is the radix. A Pipeline implementation of the FFT consisted of a series of computational blocks each composed of delay lines, coefficient storage, commutators, multipliers, and adders. Pipeline FFTs can be generally run at high-speeds and the amount of pipelining increased or decreased to meet timing. He and Torkelson have classified different pipeline approaches and put into functional blocks with unified terminology. Pipelined present smaller latency, with low power consumption which makes them suitable for most application. The pipelined architectures can be classified into two types: single-path architectures and multi-path architectures. Several single-path architectures have been proposed: Radix-2 single-path delay feedback, Radix-4 single-path delay feedback, Radix-2^2 single-path delay feedback, Radix-2^4 single-path delay feedback, Split-Radix single-path delay feedback, and Radix-4 single-path delay commutator. In single-path delay architecture have output are feedback to the input so it also called single-path delay feedback architecture. The multi-path architectures: Radix-2 multi path delay commutator, Radix-4 multi-path delay commutates, Split-Radix multi-path delay commutator, and Mixed-Radix multi-path delay commutator. The MDF is used one of the output is connected again forward to the next block of the input. So this type of architecture is called Feed forward architecture. This multipath delay feedback is also known as multipath delay commutator. Parallel processing is used to increase the sampling rate by replicating the hardware.

Radix-4 FFT Architecture:

Single-path delay feedback R4SDF: It is a radix-4 version of R2SDF. It is as efficient as R2SDF in terms of memory utilization and the utilization of multipliers increases from 50% to 75% at a cost of only 25% utilization of the butterfly element. The memory requirement for this architecture is N-1.

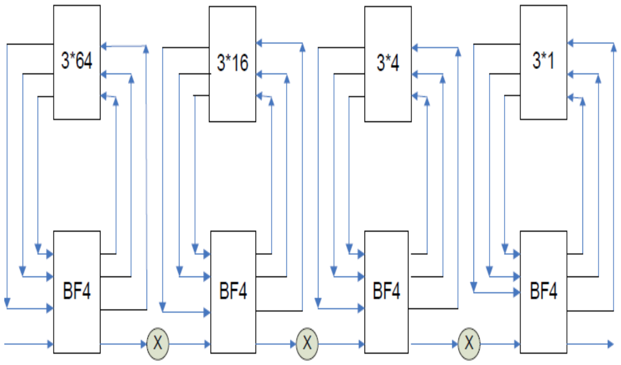

Fig -1 R4SDF(N=256)

Single-path delay Architecture R4SDC: S.He and M.Torkelson described that in Radix-4 Single path Delay Commutator a modified radix-4 algorithm is used.In this architecture the multipliers are utilized upto 75% using the programmable butterflies. The memory requirement can be reduced to 2N -2 using combined delay-commutator. The most complicated parts in this architecture are butterfly and delay-commutator. For low power applications a new commutator is introduced. This commutator is termed as IDR architecture which is far superior than single port, Dual port and Triple port RAM architectures interms of power and area for two stages. For the third stage single port RAM is the good choice. The RAM blocks in this IDR architecture are enabled only if the outputs are needed which helps in reducing switching activity which inturn reduces the power consumption.

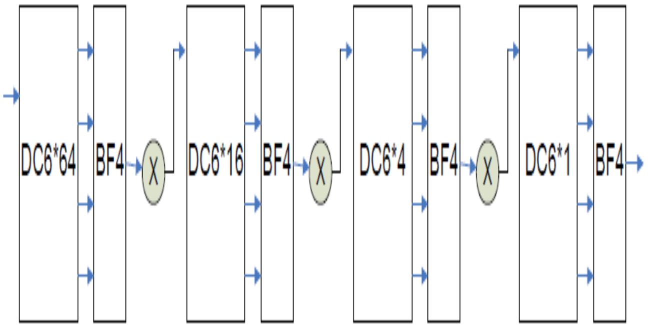

Fig -2 R4SDC(N=256)

Multi-path Delay commutator R4MDC: The Radix-4 Multi-path Delay Commutator as shown in Fig. is a radix-4 version of R2MDC. However, it suffers from low, 25%, utilization of all components. This can be compensated in applications like MIMO where 4 FFTs can be processed simultaneously. It requires 3logN multipliers, logN full radix-4 butterflies and 2.5N − 4 memory elements. The main drawbacks are high hardware cost and low utilization. However, if the four inputs are fed at the same time, computational elements are utilized perfectly which inturn reduce memory requirements.

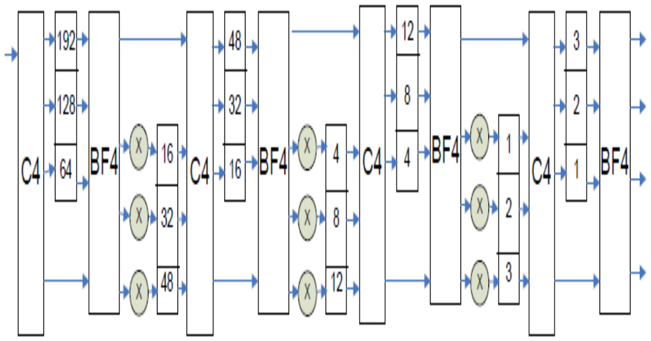

Fig- 3 R4MDC (256)

Requirement of hardware for Radix-4 FFT Pipeline Architecture

Table-1

Architecture	Complex Multiplies	Complex Additions	Memory size	Butterfly
R4SDF	(Log_4N-1)	$8Log_4N$	N – 1	Radix-4
R4SDC	$(Log_4N – 1)$	$3Log_4N$	2N - 2	Radix-4
R4MDC	$3(Log_4 N -1)$	$8Log_4N$	2.5N - 2	Radix-4

Advantage and Disadvantage of Pipeline Architecture: The two pipeline architectures i.e., single path delay feedback and multi-path delay commutator have both advantages and disadvantages. The advantage of multipath delay commutator over single-path dealy feedback is that throughput rate is much higher also the MDC scheme helps in achieving higher throughput rate, on the otherside the SDF scheme needs lesser memory and hardware cost. In UWB applications MDC architecture is preferred due to its high-throughput-rate. The disadvantages of MDC architecture are the number of data path required is large, the size of the FFT is large. The requirement of the memory cells and complex multiplier in the Multipath Delay Commutator architecture is comparatively higher than that of the Singlepath Delay Feedback structure. In the table, Pipeline Architecture of Radix-4 FFT Processor are given. The single-path delay feedback architecture have complex multiplies and minimum memory requirement than other architecture. So R4SDF are most suitable Architecture for processing.

Conclusion:

In this paper we present pipeline architecture Radix-4 FFT Processor. The pipeline Architecture are two type, single-path delay and Multipath delay commutator. This method introduce the minimum requirement of memory size and reducing power consumption also speedup FFT Process

ACKNOWLEDGEMENT:

The Authors are thankful to Dr. Manish Mishra (Assist. Prof.) Department of Electronics DDU Gorakhpur University Gorakhpur .The Authors also thanks Roopak Kumar Verma DDU Gorakhpur University Gorakhpur

References:

[1] J. W. Cooley and J. Tukey, “An algorithm for machine calculation of complex Fourier series,” Math. Comput., vol. 19, pp. 297–301, Apr.1965.

[2] G. Bi and E. V. Jones, “A Pipeline FFT Processor for Word-sequential Data,” IEEE Trans. Acoust. Speech, Signal Processing, Vol. 37(12), pp. 1982- 1985, 1989.

[3] S. He and M. Torkelson, “A New Approach to Pipeline FFT Processor,” The 10th International Parallel Processing Symposium(IPPS), pp. 766 - 770, 1996.

[4] Zeke Wang, Xue Liu, Bingsheng He, Feng Yu “A Combined SDC-SDF Architecture for Normal I/O Pipelined Radix-2 FFT” IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS 2013

[5] Weidong Li, Yutai Ma and Lars Wanhammar “Word Length Estimation for Memory Efficient Pipeline FFT/IFFT Processors”

[6] T.Angeline, D.Narain Ponraj “A Survey on FFT Processors” International Journal of Scientific & Engineering Research Volume 4, Issue3, March-2013 1 ISSN 2229-5518

[7] 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 [8] Bin Zhou, Yingning Peng, and David Hwang “Pipeline FFT Architectures Optimized for FPGAs” International Journal of Reconfigurable Computing Volume 2009, Article ID 219140, 9 pages doi:10.1155/2009/219140