ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 4, October 2014

64 Point Radix-4 FFT Butterfly Realization using FPGA

Amaresh Kumar, U.N. Tripathi, Roopak Kumar Verma, Manish Mishra
Department of Electronics, D.D.U. Gorakhpur University, Gorakhpur
Department of Computer Science, D.D.U. Gorakhpur University, Gorakhpur

Abstract-The Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) play vital role in signal processing. It is often used in many communication systems. The proposed paper produces realization of N–bit FFT processor using Radix-4 algorithm. Radix-4 FFT processors have 3N/4 log_4N complex multiplications and 3N log_4N complex additions. N-bit processor contain N/4 butterfly for single stage. The single butterfly needs 12 complex adders and 3 complex multipliers. The proposed Radix-4 FFT processor is realized on VHDL platform using vertex FPGA.

Keywords: **Fast Fourier Transform (FFT), FPGA.**

I. INTRODUCTION

The Fast Fourier Transform (FFT) Algorithm plays an important role in operation of digital signal processor. In the recent years, FFT and IFFT have been frequently applied in the modern communication systems. The FFT is an efficient class of computational algorithms of the DFT. Original computation of DFT with N samples input requires N^2 complex computation. The FFT algorithms are based on the fundamental principle of decomposing the computation of the DFTs, all with comparable improvements in computational speed. Cooley and Tukey [1] introduced the concept of the FFT to demonstrate a significant computational reduction of O (N^2) to O (NlogN) by making efficient use of symmetry and periodicity properties of twiddle factors. Followed by pioneering work of Cooley and Tukey algorithm, several algorithms have been developed to further reduce the computational complexity. Other researchers proposed numerous techniques such as Radix-4, radix-8 and split radix [2] to avoid radix to structure in order to reduced the complexity of FFT algorithm. These architecture are either based on the decimation-in-time (DIT) or on the decimation-in-frequency (DIF) [3]. In all FFT processors, the basic building blocks are the "Butterfly" which depends on radix of FFT Processor. The radix-4 FFT algorithm is more suitable for digital signal processor which has minimal complex computation than radix-2, radix-8 and structural architecture is also more suitable than other radix FFT algorithms. In this paper an attempt has been made for the efficient implementation of N-point radix-4 butterfly FFT algorithms on FPGA platform.

FFT Algorithm: The N-point DFT is defined as

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

$$0 \le n \le N\text{-}1,\ 0 \le k \le N\text{-}1 \qquad (1)$$

Where X(K) is the frequency domain and x(n) is time domain of sequence and W_N^{nk} is called Twiddle factors ,where:

$$W_N^{nk} = [\cos(\tfrac{2\pi n}{N}) - j\sin(\tfrac{2\pi n}{N})]^k \qquad (2)$$

The development of computationally efficient algorithms for the DFT is made possible if we adopt a divide-and-conquer approach [4], the approach which is based on the decomposition of N-point DFT in to successively smaller DFTs. $N = r^v$ where v is called number of stage of FFT and r is called radix of FFT algorithms.

II. RADIX-4 FFT ALGORITHM

Further, the N-point discrete Fourier Transform X (K) of sequence x(n) is defined as:

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad (3)$$

For radix-4 $N = r^v = 4^v$ where v is number of stage and r is the radix of FFT.

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

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

$$X(K) = \sum_{n=0}^{\frac{N}{4}-1} x(n) W_N^{nk} + W_N^{Nk/4} \sum_{n=0}^{\frac{N}{4}-1} x(n+N/4) W_N^{nk} + W_N^{Nk/2} \sum_{n=0}^{\frac{N}{4}-1} x(n+N/2) W_N^{nk} + W_N^{3Nk/4} \sum_{n=0}^{\frac{N}{4}-1} x(n+3N/4) W_N^{nk},$$

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 4, October 2014

$X(K) = \sum_{n=0}^{\frac{N}{4}-1}\{x(n) + W_N^{\frac{Nk}{4}}(x(n+N/4)) + W_N^{\frac{Nk}{2}}(x(n+N/2)) + W_N^{\frac{3Nk}{4}}(x(n+3N/4))\}W_N^{nk}$,

$W_N^{\frac{Nk}{4}} = [\cos(\frac{2\pi}{4}) - j\sin(\frac{2\pi}{4})]^k = (-j)^k$,

$W_N^{\frac{Nk}{2}} = \{\cos(\frac{2\pi}{2}) - j\sin(\frac{2\pi}{2})\}^k = (-1)^k$,

$W_N^{\frac{3Nk}{4}} = \{\cos(\frac{2\pi*3}{4}) - j\sin(\frac{2\pi*3}{4})\}^k = (j)^k$,

Now ,

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

The equation (4) is the N-point DFT because phase factor depends on N-point but not on N/4. To convert into N/4 DFT, we subdivide the DFT into four N/4 point sequence, X(4K), X(4K+1), X(4K+2) and X(4K+3).

$X(4K) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) + x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) + x(n+3N/4)]\,W_N^0 W_{\frac{N}{4}}^{nk}$

,

$X(4K+1) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) - jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + jx(n+3N/4)]\,W_N^n W_{\frac{N}{4}}^{nk}$

,

$X(4K+2) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) - x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) - x(n+3N/4)]\,W_N^{2n} W_{\frac{N}{4}}^{nk}$

$X(4K+3) = \sum_{n=0}^{\frac{N}{4}-1}[x(n) + jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - jx(n+3N/4)]\,W_N^{3n} W_{\frac{N}{4}}^{nk}$

With property $W_N^{4nk} = W_{\frac{N}{4}}^{nk}$

Where $0 \le k \le N/4 - 1$

$0 \le n \le N/4 - 1$

These are the expression of Radix-4 FFT algorithms. The radix-4 Butterfly contains 3 complex multiplications and 12 complex additions .N/4 butterfly involves in each stage and number of stage is $\log_4 N$ for N-point sequence. Therefore, the number of complex multiplications is $3N/4\log_4 N$ and number of complex additions is $12N/\log_4 N$. In comparisonof radix-2 FFT, number of complex multiplications are reduce by 25% but number of complex additions are increased by 50%. Fig 2 shows a signal flow graph of Radix-4 butterfly decimation-in-frequency algorithm and signal flow graph for 64-point DIF FFT.

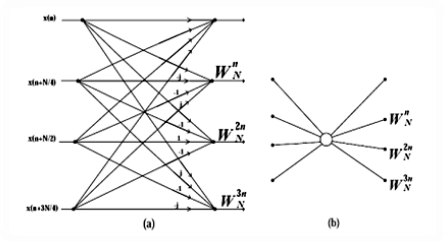

Fig 1 (a) and Fig (b) signal flow graph of radix-4 butterfly DIF FFT algorithm.

III. ARCHITECTURE OF RADIX-4 FFT BUTTERFLY

For N-point sequence, the radix-4 FFT algorithm consist of taking number of 4 data points at a time from memory, performing the butterfly computation and returning the result to memory. This procedure repeated many times, i.e., $((N\log_4 N)/4)$ times in the computation of N-point data DFT. Therefore, memory requirement is essential factor for FFT processor design. The requirement of memory size is 2N for the input sequence which is complex number and 2N for the output sequence. So the capacity of memory for N-point FFT processor is 4N. Bevan M. Baas [5] described about different type memory architecture like single memory architecture, Dual memory architecture, Array architecture and cached memory architecture. These type memory architectures are not suitable for FFT processor.

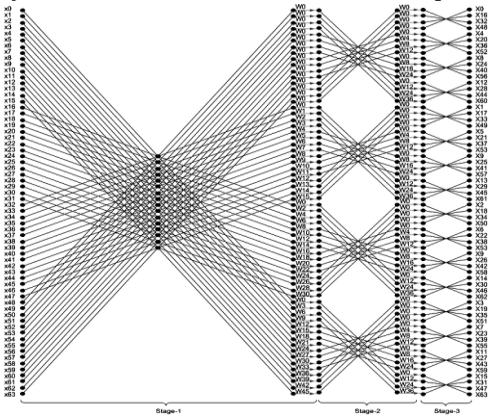

Fig 2: signal flow graph of 64-point radix-4 butterfly FFT.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 4, October 2014

IV. HARDWARE REQUIREMENT FOR DIFFERENT FFT ARCHITECTURE

Table1: Hardware complexity comparison of different FFT Processor.

	Complex multiplier	Complex adder	Memory size	Control logic	Comp. Utilization of adder	Comp. Utilization of multiplier
R2SDF	$\log_2 N - 1$	$2\log_2 N$	$N - 1$	Simple	50%	50%
R4SDF	$\log_4 N - 1$	$8\log_4 N$	$N - 1$	Medium	25%	75%
R4SDC	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$	Complex	100%	75%
R2^{2}SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	Simple	75%	75%
R2MDC	$\text{Log}_2 N - 2$	$2\log_2 N$	$3N/2 - 2$	Simple	50%	50%
R4MDC	$3(\log 4N - 1)$	$8\log_4 N$	$5N/2 - 4$	Medium	25%	25%

V. RESULT

In this section we conduct the experiment to realize 64 point, radix-4 butterfly on matlab and same configuration is modeled on hardware description language VHDL to physically realize the proposed butterfly. The size of FFTs under test ranges from 8 points, 16 points, 32 points to 64 points. Since the trend of results from different FFT sizes is similar, for the sake of the space, here we only report the results collected from 64 point FFTs as shown in Fig. 3 and Fig 4 .

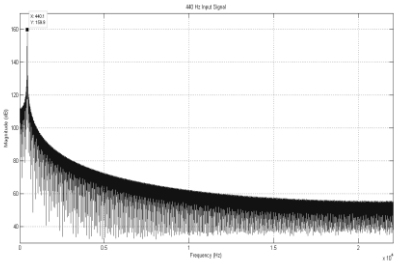

Fig 3: Frequency vs. Magnitude plot of Matlab coded 64 point radix-4 FFT for 16 bit input data stream.

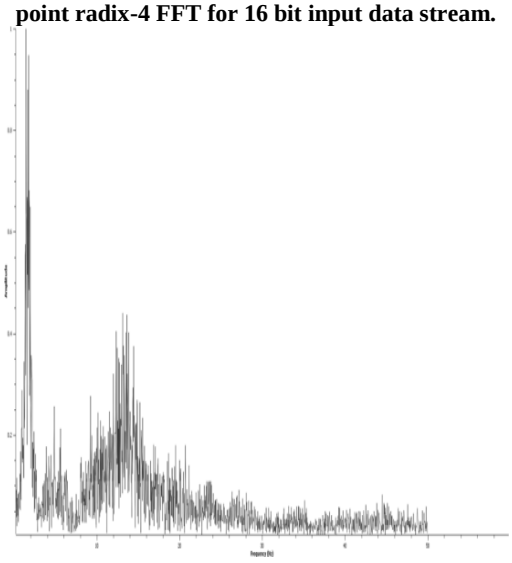

Fig4:: Frequency vs. Magnitude plot of VHDL coded 64 point radix-4 FFT for 16 bit input data stream .

The top level design is shown in figure 5, where x(n) is 16 bit input data in the form of x_i (n) and x_r(n) for imaginary and real part of data, synchronous reset (rst), clock, chip enable, start, busy, and finish.

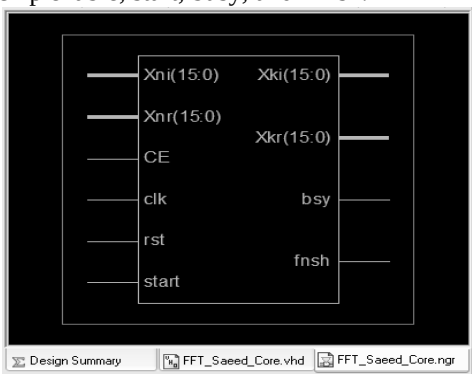

Fig 5: Top Level Design on FPGA

The result of synthesis tool and the timing analysis using the Xilinx simulator indicate a maximum operating frequency of 450 MHz; this provides an execution time of a 64 complex data point transform in 0.135 µs. A comparison is done between Xilinx FFT core, Zarlink^{TM} FFT processor and our proposed processor. Our 64 point radix-4 FFT processor achieves highest operating frequency of all the processors listed in table II.

FFT Point	Time of Proposed FFT (µs)	Time of Xilinx FFT core (µs)	Time of Zarlink FFT (µs)
08	0.039	0.048	0.042
16	0.059	0.068	0.062
32	0.098	0.106	0.103
64	0.135	0.148	0.143

Table II: Comparative study of realized processor.

VI. CONCLUSION

In this paper, we have presented a novel 64 point radix-4 FFT processor, which is verified on matlab. The description was made by VHDL in Xilinx Virtex-7 FPGA chip which is high performance signal processing logic cells, RAM and DSP slices to accommodate the complete

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 4, October 2014

design. The output from the VHDL described architecture are validated (Fig 3 and Fig 4) against the standard matlab realized FFT. The multipliers, adders/subtractors units, control unit and their pipelining were implemented by efficient use of DSP blocks in order to obtain faster and low power design. The data and twiddle factor word length were chosen to achieve an acceptable signal-to-noise ratio. The implemented design gives an easy way to increase the number of points of FTT as well as IFFT by imposing simple modification. Future work includes the design of FFT with unconventional number system like RBSD, HSD, Logarithmic, and Residue number systems. These number systems provide speed-area optimization for specific application.

ACKNOWLEDGEMENT

The Author's are thankful to Dr. Manish Mishra (Assist. Prof.) Department of Electronics DDU Gorakhpur University, Gorakhpur. The Author's also thanks Roopak Kumar Verma and U.N. Tripathi DDU Gorakhpur University, Gorakhpur.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series, Math.Computat" vol. 19, pp. 297-301, 1965.

[2] P. Duhamel and H. Hollmann. "Split-Radix FFT Algorithm". Electronic Letters, 20(1):14–16, Jan 1984.

[3] A. Saidi, "Decimation-in-Time-Frequency FFT Algorithm," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. III: 453-456, April 19-22 1994.

[4] John G. Proakis, DimitrisG. Manolokis "Digital signal processing" Pearson prentice Hall, Inc 2008

[5] Baas, B. M., "A low-power, high-performance 1024-point FFT processor", IEEE Journal of Solid State Circuits. pp. 380-387, 1999.

[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] Bin Zhou, Yingning Peng and David Hwang "Pipeline FFT Architectures Optimized for FPGAs" Hindawi Publishing Corporation International Journal of Reconfigurable Computing Volume 2009, ArticleID 219140, 9 pagesd oi:10.1155/2009/219140

AUTHOR'S PROFILE

Amaresh Kumar was born in 1988. He received B.Sc and M.Sc degrees from D.D.U. Gorakhpur University, Gorakhpur, India in 2007 and 2009 respectively, where he is currently Research Scholar in Electronic Science. His research interested in Digital Signal Processor.

Roopak Kumar Verma was born in 1990. He received B.Sc and M.Sc degrees from D.D.U. Gorakhpur University, Gorakhpur, India in 2009 and 2011 respectively, where he is Research Scholar in Electronic Science. His research interested in Digital Signal Processor.

U.N. Tripathi is currently Assistant Professor in Computer Science, D.D.U. Gorakhpur University, Gorakhpur, India. He has about 13 years teaching experience. His research interested in Digital Signal Processor.

Manish Mishra was born in 1977. He received PhD degree from Lucknow University, Lucknow, India. He also received bachelor and master degrees D.D.U. Gorakhpur University, Gorakhpur, India, where he is currently Assistant Professor in Electronics Department. He has teaching experience about 13 years in UG and PG label. His research interested included in Digital Signal Processor and VLSI architecture.