ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

Implementation of 16-Point Radix-4 FFT Algorithm

M.S.Chavan[1], Anshita Agarwal [2], Richa Jha[3], Monika Kumbhare[4]

Assistant Professor, Dept. of Electronics, B.V.D.U College of Engineering, Pune, Maharashtra, India[1]

UG Student [Electronics], Dept. of Electronics, B.V.D.U College of Engineering, Pune, Maharashtra, India [2]

UG Student [Electronics], Dept. of Electronics, B.V.D.U College of Engineering, Pune, Maharashtra, India [3]

UG Student [Electronics], Dept. of Electronics, B.V.D.U College of Engineering, Pune, Maharashtra, India[4]

ABSTRACT: 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 N2 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 Turkey introduced the concept of the FFT to demonstrate a significant computational reduction of O (N 2) to O (N log N) by making efficient use of symmetry and periodicity properties of twiddle factors. Followed by pioneering work of Cooley and Turkey 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 to avoid radix to structure in order to reduce the complexity of FFT algorithm. This architecture are either based on the decimation-in-time (DIT) or on the decimation-in-frequency (DIF). 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 16 point radix-4 butterfly FFT algorithms on FPGA platform

KEYWORDS: FFT Algorithm, DIT, Radix 4, Butterfly structure, FPGA Implementation.

I.INTRODUCTION

A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and it’s inverse. A Fourier transform converts time (or space) to frequency and vice versa; FFT rapidly computes such transformations. As a result, fast Fourier transforms are widely used for many applications in engineering, science, and mathematics. The basic ideas were popularized in 1965, but some FFTs had been previously known as early as 1805. Fast Fourier transforms have been described as "the most important numerical algorithm of our lifetime". There are many different FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory; this article gives an overview of the available techniques and some of their general properties, while the specific algorithms are described in subsidiary articles linked below. The discrete Fourier transform is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields (see DFT for properties and applications of the transform) but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing the DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while a FFT can compute the same DFT in only O(N log N) operations. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log(N). This huge improvement made the calculation of the DFT practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers. The best-known FFT algorithms depend upon the factorization of N, but there are FFTs with O(N log N) complexity for all N, even for prime N. Many

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

FFT algorithms only depend on the fact that N-the primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it. An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the only difference is that an FFT is much faster. Let x0... xN1 be complex numbers.

II.SYSTEM MODEL AND COMPUTATION

The radix-4 16-point FFT was designed using verilog code and simulated in NcVerilog Cadence in order to verify its functionality. The design is synthesized utilizing 0.18µm technology provided by Artisan Library. Timing constraint is set with operating frequency 50MHz. FFT architecture is divided into three main process blocks. The block diagram of process block is shown in Fig. 4. This block consist of data input, butterfly computation and data output. The data is read in every rising edge of clock and stored in the memory register. Butterfly computation block compute the stored data before going to data output process. The data is kept in the register before it is read out. The FFT radix-4 processor architecture consist of a butterfly architecture, memory register, control circuit, serial to parallel and parallel to serial converter. Twiddle factor are stored as 16-bit two's complement signed fixed point word.

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 (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

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

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

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

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

Now ,

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

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).

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

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%.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

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., ((Nlog4N)/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 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.

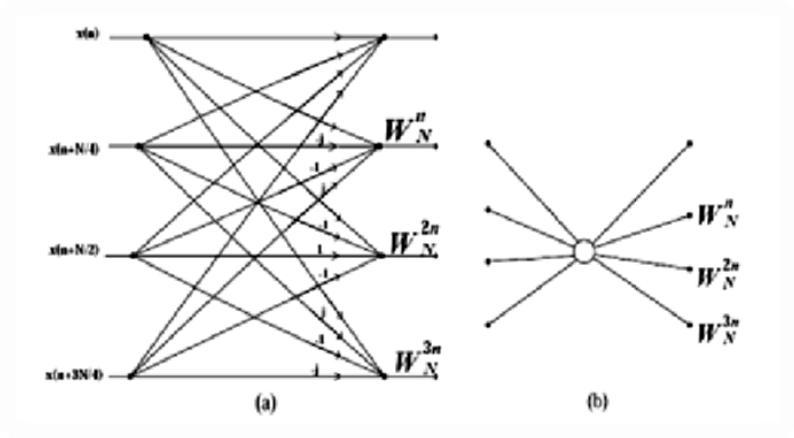
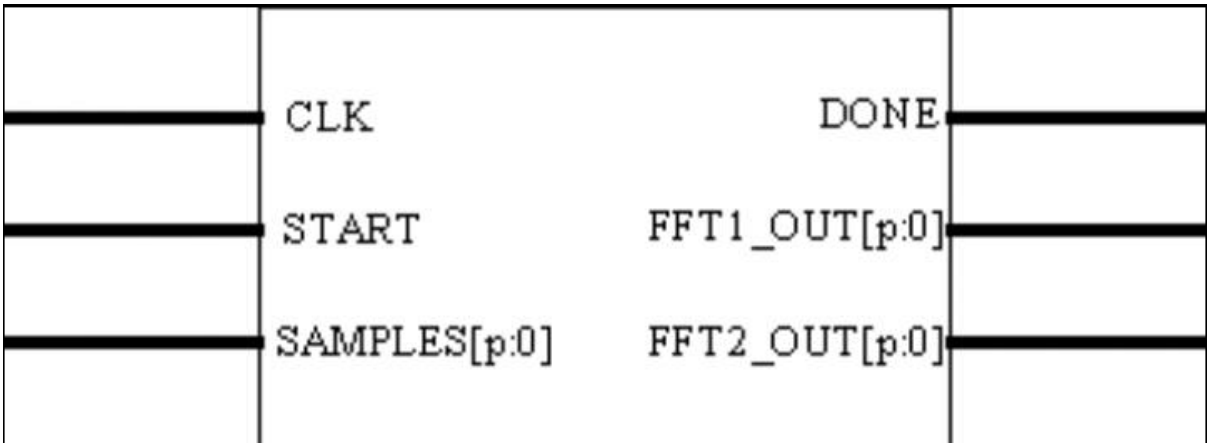


Fig 1 (a) and Fig (b) signal flow graph of radix-4 butterfly DIF FFT algorithm.

IV.HARDWARE AND SOFTWARE IMPLEMENTATION



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

In the hardware implementation part, the whole 16-point FFT is implemented in VHDL. For this design, the twiddle factors have been stored in the registers in the system instead of using ROM.
The two input signals that shown in the figure 3 are i data i and i data q, which represent the real part and imaginary part respectively

In this section we conduct the experiment to realize 16 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 16 point FFTs
x(n) is 16 bit input data in the form of xi (n) and xr(n) for imaginary and real part of data, synchronous reset (rst), clock, chip enable, start, busy, and finish.

V. APPLICATIONS

1. Television terrestrial broadcasting systems.
2. Phase correlation system.
3. Mobile receiver.
4. Implementation of digital communication system.
5. Fault characterization and classification.
6. Radar.
7. Medical Imaging

a. X-ray computed tomography (CT).
b. Magnetic Resonance Imaging (MRI)

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

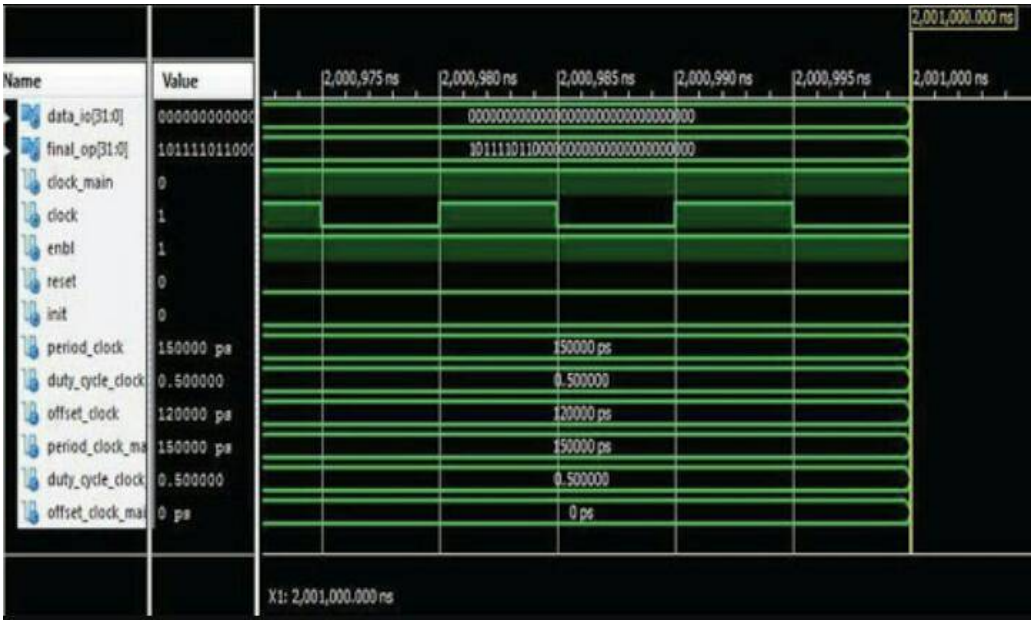
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

VI.RESULTS



The above figure shows the test bench waveforms of the final result obtained after implementing the program on Xilinx 8.1.The simulation result can also be observed on the CRO by converting the program in ucb file.

VII.CONCLUSION

This thesis is about the radix-4 16-point DIF FFT which has been implemented in 65nm CMOS technology. Based on the radix-4 algorithm, it presents the superiority of low area consumption and longer throughput. More specifically, the usage of the adders and multipliers are shrinking significantly. Since the hardware architecture of the design includes 4 stages with their twiddle factors separately. For each stage, one multiplier and two adders are being consumed. The number of the registers are the half of the input signals for each stage. The area depletion would decrease because of the less adders and the multipliers are being used in the implementation. For the proper trade-o↵ between the accuracy and the area, the word length of the twiddle factors for all five stages have been truncated to 18 bits, the word length of the final output of the design has been truncated to 16 bits. After being synthesized, the clock frequency of the design is measured, which is 1MHz, the critical path is 7.84ns. After the Place& Route, the die size of the design is 480 ↤ 400um2. By going through the Prime Time,the total 32 Conclusion & Future Work power of the design is 0.0323mW. The critical paths for the setup time and hold time are 3.12ns and 0.85ns separately

REFERENCES

1. http://ieeexplore.ieee.org/document/1052151/
2. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=BUTTERFLY+ARCHITECTURE&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=BUTTERFLY%20ARCHITECTURE%20for%20fft
3. https://hal.archives-ouvertes.fr/hal-00083992/document
4. https://www.researchgate.net/publication/242297044_DESIGN_OF_AN_FFT_PROCESSOR
5. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=FFT+PROCESSOR&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=FFT%20PROCESSOR
6. http://www.slideshare.net/RohitSingh227/fft-processor-35328738

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

7. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=FFT+PROCESSOR&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=FFT%20PROCESSOR%20IMAGE
8. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=BUTTERFLY+ARCHITECTURE&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=BUTTERFLY%20ARCHITECTURE&gsc.page=1
9. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=FFT+PROCESSOR&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=FFT%20PROCESSOR&gsc.page=1
10. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=FFT+PROCESSOR&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=FFT%20PROCESSOR&gsc.page=1
11. http://www.ijeit.com/Vol%204/Issue%204/IJEIT1412201410_10.pdf
12. https://cse.google.com/cse?cx=partner-pub-8036109189802438%3A7790813904&ie=UTF-8&q=16+point+radix+4+fft+algorithm+using+vhdl&sa=Search&siteurl=yourtv.link%2F#gsc.tab=0&gsc.q=16%20point%20radix%204%20fft%20algorithm%20using%20vhdl&gsc.page=1
13. http://www.eit.lth.se/sprapport.php?uid=856
14. http://www.sersc.org/journals/IJAST/vol61/6.pdf