

## **Design of Optimum FFT Architecture For OFDM Application**

**Rajesh Garikapati<sup>1</sup>, K.Hariharan<sup>2</sup>**

*<sup>1</sup>PG-Scholar, <sup>2</sup>Asistant Professor*

*School of Computing, SASTRA University,*

*Thanjavur – 613401, Tamilnadu, India*

*Email id:rajeshgarikapati16@gmail.com*

### **Abstract**

OFDM stands for Orthogonal Frequency Division Multiplexing. It is a multicarrier modulation technique used in wireless digital communication systems. Some of the well-known applications of the OFDM are wireless LAN IEEE 802.11a and Digital Audio Broadcasting. The OFDM translates the high rate serial data stream into several slower parallel streams such that the channel on each of slow parallel streams can be considered flat. This results in the overlap of subcarrier frequency there by increasing the bandwidth utilization up to 50%. This project focus on the implementation of FFT based Orthogonal Frequency Division Multiplexing applied in multicarrier scheme. It uses radix-2, based 8 point FFT architecture for efficient calculation of Complex Multiplications. The decisive Implementation of architecture is design of a FFT which can perform exponential factors without multiplier. The main focus is to eliminate the use of multipliers which reduces area and increases the speed.

**Keywords:** DIT-FFT, Radix-2, complex multiplication, CSD, Verilog

### **Introduction**

The basic principle of Fast Fourier Transform (FFT) algorithm is to break down the calculation of discrete Fourier transform of an arrangement of length N into consecutive smaller discrete Fourier transforms. There are fundamentally two classes of FFT algorithms. They are i) Decimation-in-time ii) Decimation-in-frequency. FFT is essential maneuver in the progressively widespread multi carrier modulation skills such as orthogonal frequency division multiplexing and evolving high speed data communication system. The vital design aspect of the hardware application of FFT is a procedure of execution of multiplication by intricate exponential factors. In FFT processor butterfly is the elementary unit and using sole butterfly total computation is

done. Butterfly unit consists of twiddle factors, complex addition and multiplication [2]. The execution of an FFT processor is one of the major difficult parts in order to achieve real time processing requirements while decreasing difficulty. The FFT hardware is mainly inhibited by the power and area necessities [3]. So the emphasis is to curtail these two bounds without surrendering the performance. FFT hardware insists vast FFT sizes to store twiddle factors, pipeline registers and they are normally stored in either storage or memory elements. The vast size of the area of FFT also includes special entities like complex multipliers and complex adders.

## Existing Method

### Highly Pipelined Calculations

The Winograd FFT algorithm is used to implement the 8 point DFT which is named as the base FFT operation. It provides a least number of additions and multiplications. The complex number is given in the input data buffer RAM and the complex result is obtained in the output buffer RAM in each clock cycle. All these calculations are computed by the small unit called FFT8. All calculations are done in the high pipelined mode. A higher clock frequency of greater than 250MHz is attained by dividing the 8-point DFT algorithm into several stages. A latent delay of 14 clock cycles is the delay of the FFT8 from input of the first data to get the output of the first result can be observed.

## Proposed Constant Based FFT

### Discrete Fourier Transform

Altering time representation of signal into frequency representation can be achieved by using Fourier transform.

The DFT of an input order can be calculated using below formulation

$$X(k) = \sum_{n=0}^{N-1} X(n)W_N^{nk} \quad 0 \leq k \leq N - 1 \quad (1)$$

$$W_N = e^{-j2\pi/N}$$

Where  $X(k)$  is the  $k^{\text{th}}$  harmonic

$X(n)$  is the input sequence

### IDFT

This is inverse discrete Fourier transform it is exactly reverse of DFT .the N-Point IDFT is given by equation (2)

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k)e^{j2\pi kn/N} \quad (2)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} x(k) W_n^{-kn} \quad (3)$$

The above equation is divided into two sums

$$X(n) = \frac{1}{N} \sum_{l=0}^{\frac{N}{2}-1} X(2l)W_N^{-2ln} + \frac{1}{N} \sum_{l=0}^{\frac{N}{2}-1} X(2m + 1)W_N^{-n(2m+1)} \tag{4}$$

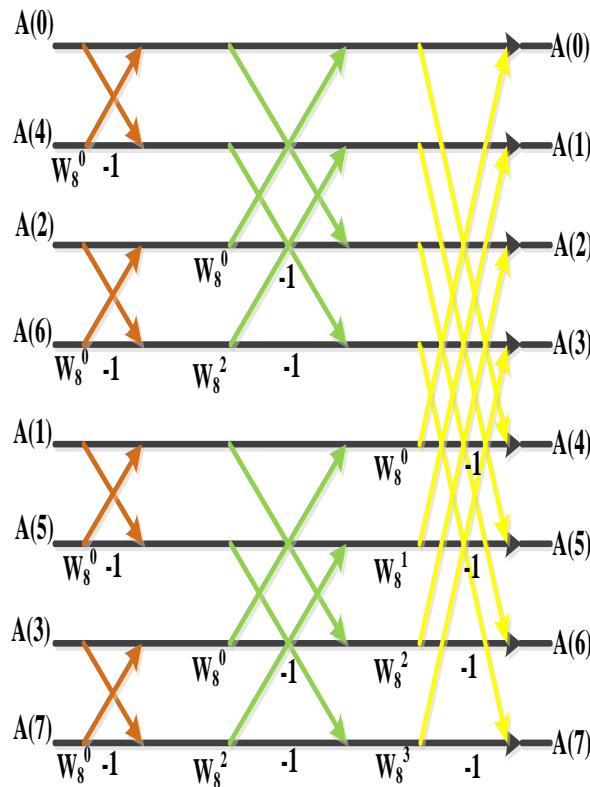
$$X(n) = \frac{1}{N} \sum_{l=0}^{\frac{N}{2}-1} G1(l)W_N^{-ln} + W_N^{-n} \frac{1}{N} \sum_{l=0}^{\frac{N}{2}-1} G2(l)W_{N/2}^{-ln} \tag{5}$$

$$X(n) = g1(n) + W_N^{-n} g2(n) \tag{6}$$

$g1(l)$  and  $g2(l)$  are Inverse Fourier transform of  $G1(l)$  , $G2(l)$  Respectively. It is also known as IFFT [1], [6].

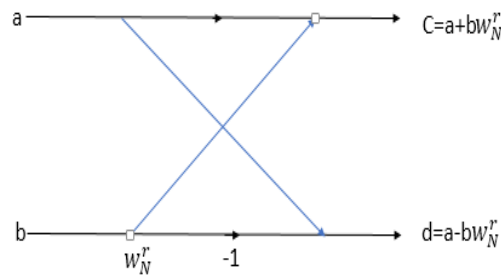
**Eight-Point FFT Units:**

The radix-2 DIT 8 point FFT has been chosen to build the algorithm. The Fig.1 represents the concept of the algorithm using radix-2 DIT-FFT [5], [6].



**Figure 1:** 8 point DIT-FFT [1]

Fig.2 The butterfly calculations in this case are mainly based on addition, subtraction, and shift operations. To minimize the latency allied with the addition of chain operands carry-save adder tree is implemented. The size of the internal unit is 16 bit.8 point FFT unit has been executed by decimation-in-time (DIT) Fast Fourier Transform algorithm.

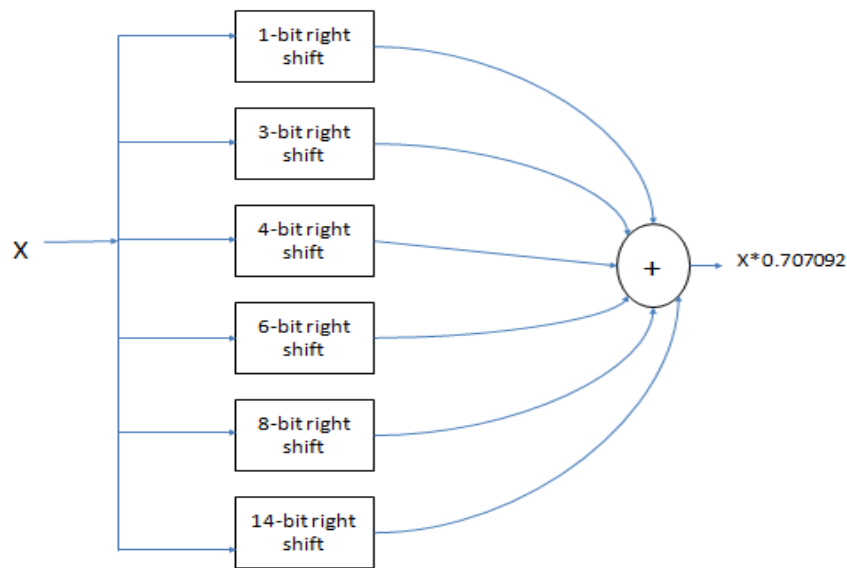


**Figure 2:** Butterfly Unit

The twiddle factors which we use in this FFT algorithm are of the fashion  $W_8^0=1$ ,  $W_8^1=0.707-j0.707$ ,  $W_8^2=-j$ ,  $W_8^3=-0.707-j0.707$ . These factors may be either trivial or non-trivial. If the factor is trivial and the twiddle factor is of the type then multiplication is much simple in which the real and imaginary part can be exchanged just by changing their sign [4] and if the factor is non-trivial more multifarious multiplications are required.

**Multiplier Unit**

If the twiddle factors are of the fashion  $W_8^1=0.707-j0.707$ ,  $W_8^3=-0.707-j0.707$ . Then they have 0.707 numbers in similar which reduce the complexity of multiplication. A number is symbolized as  $2^{-1}$  in a canonical signed digit representation. The value of 0.7071068 can be represented as  $2^{-1}+2^{-3}+2^{-4}+2^{-6}+2^{-8}+2^{-14}$  with an accuracy of 26-bit. Thus, with this decomposition, multiplying x with this constant turns this into an addition or subtraction of a series of right shifted values of x. This is shown in Fig. 3. These are just shift operations in binary arithmetic calculation. So with the help of only adders and shifters the multiplication of a number is carried out.



**Figure 3:** Complex Multiplier Unit

This significant remark combined with the illustration of each constant in Canonical Signed Digit representation provides a multiplier free calculation. A single clock cycle is enough to compute an 8 point FFT.

**Results and Discussions**

The simulation of this Verilog code has been magnificently verified in Modelsim ALTERA starter edition 10.1d. It is also synthesized in Xilinx ISE 12.1. This is also verified in MATLAB code for FFT. There are total 8 inputs of design which are real and imaginary i.e., real for real and imag for imaginary Fig 4. The outputs of the design are represented as out\_real for real and out\_imag for imaginary.

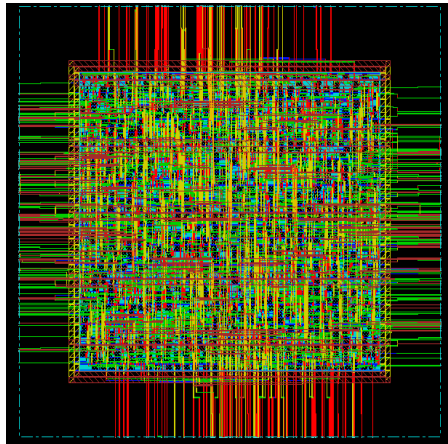
/tb8/fft8out0_real	36	36
/tb8/fft8out1_real	-4	-4
/tb8/fft8out2_real	-4	-4
/tb8/fft8out3_real	-4	-4
/tb8/fft8out4_real	-4	-4
/tb8/fft8out5_real	-4	-4
/tb8/fft8out6_real	-4	-4
/tb8/fft8out7_real	-4	-4
/tb8/fft8out0_imag	0	0
/tb8/fft8out1_imag	10	10
/tb8/fft8out2_imag	4	4
/tb8/fft8out3_imag	4	4
/tb8/fft8out4_imag	0	0
/tb8/fft8out5_imag	-4	-4
/tb8/fft8out6_imag	-4	-4
/tb8/fft8out7_imag	-12	-12

**Figure 4:** Test bench output for FFT

**Table 4.1:** Tabulation of Proposed Work

Area (No of occupied Slice )	Timing (Max Combinational Delay)	Power (Total Thermal Power Dissipation)
742 out of 16640	25.177ns	163.51mw

The above tabular form shows area, timing and power dissipation of proposed work which is done in tools like Altera Quartus and Xilinx ISE. The following Fig 6 shows the SoC system layout done in cadence tool.



**Figure 6:** Layout Design For FFT

### Conclusion and Future Work

The radix-2 DIT 8 point FFT architecture for efficient computation of complex multiplications is used. The design of FFT architecture which can perform exponential factors without multiplier is done. The area is reduced and the speed is increased by eliminating the multipliers which is the vital key. This work can be further extends to 64 point FFT architecture.

### References

- [1] Michael Bernhard, Joachim Speidel "Implementation of an IFFT for an Optical OFDM Transmitter with 12.1 Gbit/s" Universit"at Stuttgart, Institut fur Nachrichtenertragung, 70569 Stuttgart.
- [2] Cheng-Ying Yu, Sau-Gee Chen and Jen-Chuan Chih" EFFICIENT CORDIC DESIGNS FOR MULTI-MODE OFDM FFT" Department of Electronics Engineering and Institute of Electronics National Chiao Tung University Hsinchu, Taiwan, ROC
- [3] Koushik Maharatna, Eckhard Grass, and Ulrich Jagdhold" A 64-Point Fourier Transform Chip for High Speed Wireless LAN Application Using OFDM" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 39, NO. 3, MARCH 2004
- [4] Mounir Arioua, Said Belkouch, Mohamed Agdad, "VHDL implementation of an optimized 8-point FFT/IFFT processor in pipeline architecture for OFDM" Morocco. IEEE 978-1-61284-732-0.
- [5] R. Bhatia, M. Furuta, and J. Ponce, "A quasi radix-16 FFT VLSI processor," in Proc. IEEE Acoustics, Speech, and Signal Processing (ICASSP), Apr. 1991, p.
- [6] James w. Cooley, John W. Tukey "An algorithm for machine calculation of complex fourier series", JSTOR.