

Design and Implementation of 8 point FFT using Verilog HDL

Sonali Kangarkar
 Student
 VLSI Design & Embedded System
 KLE Dr.M.S.Sheshgiri College of Engineering &
 Technology

Rajashri Khanai, PhD
 Professor
 Department of Electronics and Communication
 KLE Dr.M.S.Sheshgiri College of Engineering &
 Technology

ABSTRACT

The importance of Digital Signal Processing (DSP) algorithms have increased drastically in recent times, the two important techniques of DSP are the Discrete Fourier Transform(DFT) and the Fast Fourier Transform(FFT). DFT is broadly used in the applications such as convolution, linear filtering etc. Another algorithm to compute DFT efficiently is the Fast Fourier Transform (FFT). Fast Fourier Transform processor has an important role in the field of communication system such as audio broadcasting and digital video etc. This paper deals with the designing of an 8 point FFT using radix-2 DIT FFT algorithm. This 8 point FFT design is implemented using Verilog HDL in Xilinx ISE Software.

General Terms

Discrete Fourier Transform, Fast Fourier Transform, Decimation in Time, Decimation in Frequency, algorithm

Keywords

Digital Signal Processing (DSP), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Split-Radix FFT (SRFFT), Decimation in Time FFT(DIT-FFT), Decimation in Frequency(DIF-FFT)

1. INTRODUCTION

Digital signal processing is one of the frequently used techniques for video and audio applications. Many techniques are available in the DSP domain to analyze the video or audio signals. Discrete Fourier Transform (DFT) is widely used algorithm in digital signal processing applications such as linear filtering, convolution, spectrum analysis and correlation. DFT is said to be a frequency domain representation of original Sequence.

$$X(k) = \sum_0^{N-1} x(n)e^{-j2\pi nk/N}, k = 0, 1, \dots, N-1$$

The equation shown above can be written as

$$X(k) = \sum_0^{N-1} x(n)W_N^{kn}, k = 0, 1, \dots, N-1$$

The relationship between a time-domain signal and its frequency-domain representation is specified using DFT. The twiddle factor properties i.e. periodicity and symmetry are not exploited by DFT hence direct computation of this algorithm is inefficient.

Cooley-Tukey proposed a new algorithm called Fast Fourier Transform (FFT), which is faster than DFT. If the samples in the signal is a power of two, the FFT algorithm can be adopted. The computation of FFT takes $(N/2) \times \log_2 N$ multiplications and $N \times \log_2 N$ additions. When comparing DFT, FFT takes fewer numbers of computations.

Two different forms are available in FFT. They are, Decimation in Time (DIT) and Decimation in Frequency (DIF). Both DIT and DIF use the butterfly structure to compute FFT.

2. LITERATURE SURVEY

2.1 Designing Pipelined Structure from Radix-2:

The efficient adder compressors using the DIT-FFT algorithm. The different structures which are also dedicated for the 16 bit-width radix-2 pipelined DIT butter-fly are implemented, running at 100Mhz. Reducing the real multipliers is the main aim of this paper. This is achieved by modifying the complex multiplier structures & incorporating them into the butterfly structure. The low power and fast multiplier architectures consists of the adder compressors structures. The disadvantages of Parallel FFT processor is the speed of processing & hardware utilization [2].

2.2 Split Radix FFT Algorithm

Split-Radix FFT (SRFFT) algorithm is a modification of the Cooley-Tukey algorithm which makes use of both Radix-4 & Radix-2 decomposition in the same algorithm. In Radix-2 algorithm, the odd points and the even numbered points of the DFT can be calculated separately. Thus, there is likelihood of adopting various methods for individual parts of the algorithm, to reduce the total number of arithmetic operations involved. This idea is exploited by Split Radix FFT by making use of both decompositions in the same algorithm i.e. Radix-4 and Radix-2 decompositions. It represents an N point DFT in terms of one N/2-point DFT and two N/4 point DFTs. This algorithm combines the lesser complexity of computation associated with Radix 4 and the ease of Radix 2 algorithm so as to achieve the lowest number of arithmetic operations. Thus, the even numbered samples of the N-point DFT are computed using Radix-2 algorithm [1].

2.3 Radix 4 Pipelined FFT

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

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

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

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

For the 16-point FFT, 16 equations are obtained K = 0, ..., 3 & for 64-point FFT, K = 0, 1, 2, ..., 15 [7].

3. IMPLEMENTATION

In comparison with the DFT, FFT algorithm initiated a new trend in DSP by minimizing the order of complexity of DFT multiplication by a factor of (N log N).

FFT algorithm is divided into two parts i.e. Decimation in Time (DIT-FFT) & Decimation in Frequency (DIF-FFT). In this paper decimation in time approach is used to design and implement 8 point FFT.

DIT approach is one that uses the divide and conquer approach, this approach breaks an N point transform into two N/2 point transforms, again breaking down each N/2-point transform into two N/4 point and this continues until two point DFT are obtained.

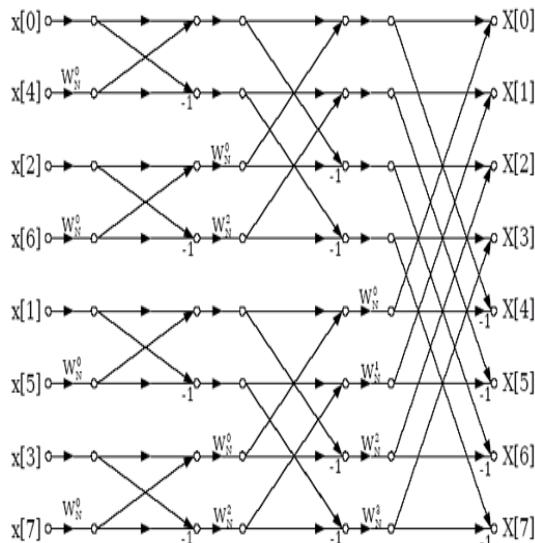


Fig 1, 8 Point DIT-FFT

FFT algorithm involves complex computations, in complex notation, the frequency domain and the time domain consists of one signal each which is made up of N complex points. Every complex point consists of two numbers, i.e. the real number and the complex number. For instance, if there is a complex number Z [20], it is a combination of both the real number ReZ [20] and the imaginary number ImZ [20]. In short, a complex number consists of two numbers. Suppose if we multiply two complex numbers, we have four components which are to be combined so as to finally obtain two components of the product.

The fig 1.2, illustrates an example of the time domain signal decomposition, this approach is followed by the FFT algorithm. As shown in the example, four stages are required to decompose a 16-point signal. Breaking of a 16-point signal into two signals, each of 8-point, this takes place in the first stage. Now each of the 8-point signals are further decomposed into two signals each of 4-point. Therefore, we obtain 4 signals of 4-point. This consists of the second stage. This process continues until a single point is obtained as shown in the figure below. During the decomposition the signal is separated into its odd and even samples.

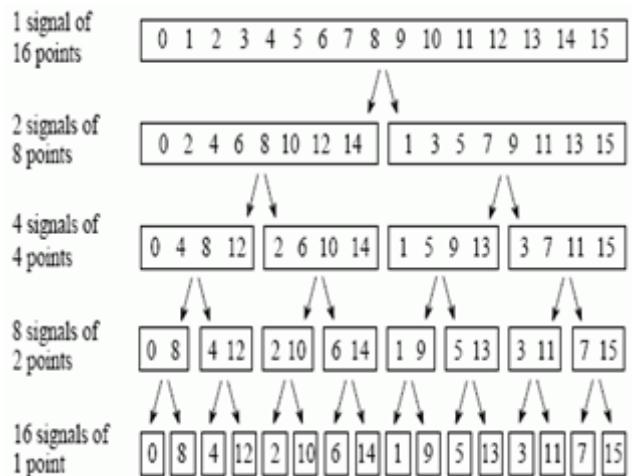


Fig 2, FFT Decomposition

The decomposition discussed above is simply, a *reordering* of the samples for a particular signal. The rearrangement pattern required for FFT is as shown in the Fig 1.3, below. On the left hand side is the number of the original signal with its binary equivalent, whereas on the right hand side is the bit reversed pattern of original signal required for FFT algorithm with its binary equivalent.

Index	Binary	Bit-Reversed	
		Binary	Index
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Fig 3, Bit Reversal

As shown below, for synthesizing a frequency domain signal three loops are necessary. The first or the outmost loop operates over the $\log_2 N$ stages, whereas the second loop operates over frequency spectra one by one in this stage. Finally, the inner loop uses the butterfly diagram computation to determine the frequency spectra points. The blocks shown as overhead in figure 1.4, below determine the sinusoids required in the butterflies, also it determines the indexes required for the loop at the beginning and at the end.

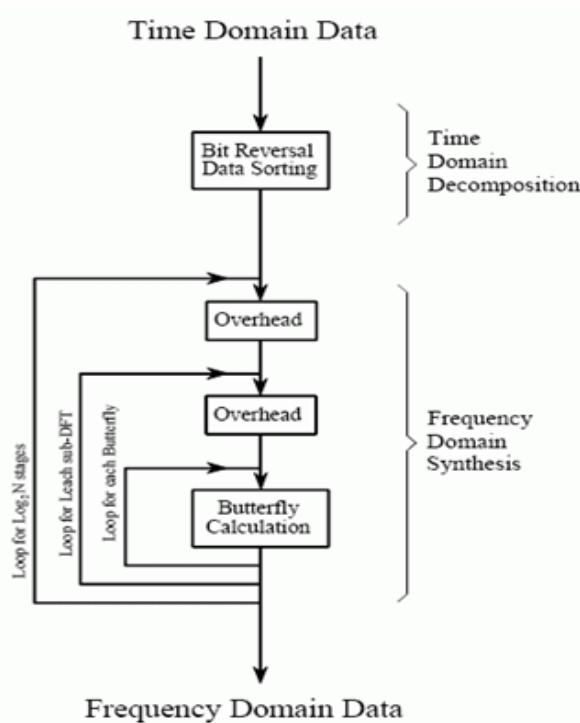


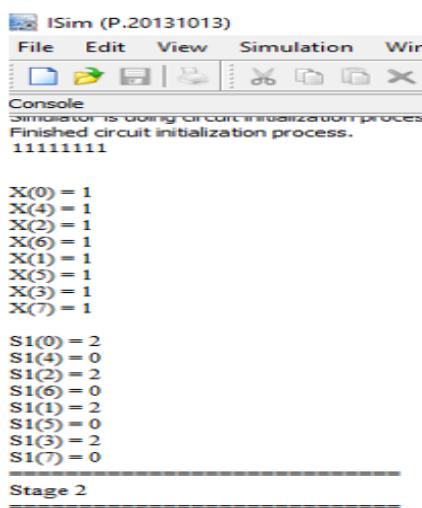
Fig 4, FFT Flow Diagram

The three steps for Flow diagram of FFT is based upon:

- 1) Decomposing a time domain N point signal one by one into N signals.
- 2) Determine spectrum for each N point signal.
- 3) Finally Synthesizing the N frequency spectra to form a single frequency spectrum.

4. EXPERIMENTAL RESULTS

The 8 point FFT designed was simulated using Xilinx ISE Software with Verilog HDL and the results obtained for the real inputs i.e. $x(n) = \{1,1,1,1,1,1,1,1\}$ are as shown below.



$S_2(0) = 4$
 $S_2(4) = 0$
 $S_2(2) = 0$
 $S_2(6) = 0$
 $S_2(1) = 4$
 $S_2(5) = 0$
 $S_2(3) = 0$
 $S_2(7) = 0$

Stage 3

$F(0) = 8$
 $F(1) = 0$
 $F(2) = 0$
 $F(3) = 0$
 $F(4) = 0$
 $F(5) = 0$
 $F(6) = 0$
 $F(7) = 0$

5. CONCLUSION

This paper makes use of simple design approach i.e. DIT-FFT approach which divides an N point transform into $N/2$ transform until two point DFT are obtained. The designed 8 point FFT block is simulated in Xilinx ISE Software with Verilog HDL for real inputs. As this FFT block is designed only for real inputs this can be extended to accept real as well as imaginary inputs, also it can be designed for higher point FFT.

6. REFERENCES

- [1] Arunkumar P. Chavan, Sowmya Nag K., "VLSI Implementation of Split-Radix FFT for High Speed Applications" International Journal of Computer Applications, January 2017
- [2] Muniandi Kannan and Srinivasa Srivatsa,"Hardware Implementation Low Power High Speed FFT Core" The International arab journal of Information Technology, Vol.6., January 2009
- [3] Marimuthu R* and P.S Mallick, "Design of Efficient Signed Multiplier Using Compressors for FFT Architecture" Journal of Engineering Science and Technology Review, May 2017
- [4] K.Baboji, Sriadibhatla.Sridevi, "FFT Implementation Using Floating Point Fused Multiplier with Four Term Adder" Dept. of Micro and Nano Electronics School of Electronics Engineering VIT University
- [5] Lamessa Dingeta, Gelaye Geresu, "Design of Pipelined Butterflies from Radix-2 FFT with Decimation in Time Algorithm using Efficient Adder Compressors" International Journal of VLSI System Design and Communication Systems, December 2016.