Implementation of 8-Point FFT with IEEE 754 Floating Format for OFDM System

Reconfigurable system able to generate output according to the input fed during runtime

Yogesh Sharma
Lecturer: Dept. of Electronics & Communication Engineering
DBA College of Engineering
Nagpur, India
yogeshsharma172@gmail.com

Yogesh Gaidhane
Lecturer: Dept. of Electronics & Communication Engineering
DBA College of Engineering
Nagpur, India
svss.yogesh@gmail.com

Preeti Mankar
Lecturer: Dept. of Electronics Engineering
St. Vincent Palloti College of Engineering
Nagpur, India
preetimankar414@gmail.com

Atul Borkar
Lecturer: Dept. of Electronics & Communication Engineering
DBA College of Engineering
Nagpur, India
borkar.atul@gmail.com

***Abstract*—** Development in technology is in debt to development in the field of communication as communication plays the important role. The process of communication involves various methods for modulation and demodulation. The most sought after type of modulation is considered to be Orthogonal Frequency Division Modulation (OFDM) for its advantages. The most important part of the OFDM system is the FFT and IFFT blocks.

The point of FFT & IFFT varies for various applications of OFDM and hence there is a need of a Reconfigurable system which can generate output according to the input fed to it. This is the main motto of this work to develop a reconfigurable FFT-IFFT system.

Keywords- OFDM, FFT/IFFT, Floating point representation, Complex Multiplier, Vedic Mathematics, Reconfigurable FFT/IFFT

*****

I. INTRODUCTION *(HEADING 1)*

OFDM can be seen as either a modulation technique or a multiplexing technique. One of the main reasons to use OFDM is to increase the robustness against frequency selective fading or narrowband interference. Error correction coding can then be used to correct for the few erroneous subcarriers. The concept of using parallel data transmission and frequency division multiplexing was published in the mid-1960s [1, 2]. Some early development is traced back to the 1950s.

OFDM has been adopted as a standard for various wireless communication systems such as wireless local area networks, wireless metropolitan area networks, digital audio broadcasting, and digital video broadcasting. It is widely known that OFDM is an attractive technique for achieving high data transmission rate in wireless communication systems and it is robust to the frequency selective fading channels.

The ability to define the signal in the frequency domain, in software on VLSI processors, and to generate the signal using the inverse Fourier transform is the key to its current popularity. The use of the reverse process in the receiver is essential if cheap and reliable receivers are to be readily available. Although the original proposals were made a long time ago, it has taken some time for technology to catch up.

At the transmitter, the signal is defined in the frequency domain. It is a sampled digital signal, and it is defined such that the discrete Fourier spectrum exists only at discrete frequencies. Each OFDM carrier corresponds to one element of this discrete Fourier spectrum. The amplitudes and phases of the carriers depend on the data to be transmitted. The data transitions are synchronized at the carriers, and can be processed together, symbol by symbol.

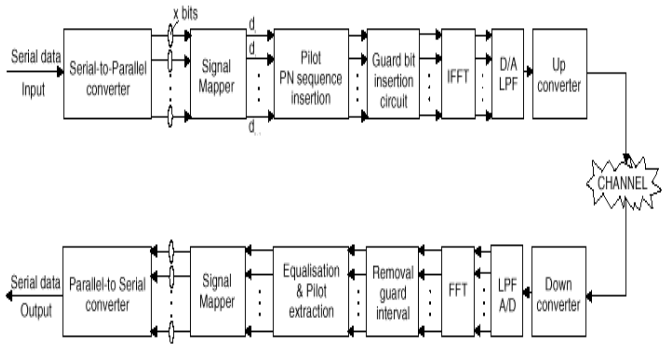


Figure 1. Example of a ONE-COLUMN figure caption.

There are many types of FFT architectures used in OFDM systems. They are mainly categorized into three types namely the parallel architecture, the pipeline architecture and the shared memory architecture. The high performance of a parallel and pipelined architecture is achieved by having more butterfly processing units but they consume larger area than the shared memory architecture. On the other hand, the shared memory architecture requires only one butterfly processing unit and has the advantage of area efficiency.

The rest of the paper is organized as follows. In section II, the FFT algorithm is reviewed. Section III includes comparative study of various methods and architectures available for reconfiguring FFTs for wireless systems. Section IV gives a tabular comparison of the all the methods reviewed. Finally a conclusion is given in section V.

II. FFT ALGORITHM

Fast Fourier Transform (FFT) has already been widely applied to area of signal analysis, frequency spectrum estimate, and OFDM-based communication systems. In OFDM systems, the required FFT sizes vary according to different operation standards or system parameters. FFT size required in various communication standards are show in Table 1.1. In portable applications, much effort should be paid to the power consumption. Hence, it's desirable to design a domain specific, power-scalable, high speed and area-efficient variable-length FFT processor. [3]

There are generally three kinds of variable-length FFT architecture: memory-based, pipeline and general-purpose DSP. General-purpose DSP is the most flexible one to address the requirement of various FFT size. But it is neither power efficient nor area-efficient in some specific systems. Besides, it require the most computation cycles. Memory-based FFT is considered most area-efficient, but it also needs many computation cycles. Pipeline FFT processor has the highest throughout. To meet the real-time requirement, pipeline architecture is chosen.[4]

TABLE I. FFT size for OFDM applications

Application	System	FFT size
Wireless networks	WLAN(802.11a)	64
	WLAN(802.16e)	256,512,1024
Wired Broadband	ADSL	512
	VDSL	256,512,1024,2048,4096, 8192
Digital Terrestrial Broadcasting	DAB	256,512,1024,2048
	DVB-T	2048.8192
	ISDBT-T	2048,4096,8192
	DMB-T/H	4096

Fast Fourier transform (FFT) has been playing an important role in digital signal processing and wireless communication systems. The choice of FFT sizes is decided by different operation standards. It is desirable to make the FFT size changeable according to the operation environment. The Fourier transform is a very useful operator for image or signal processing.

The reduction of complexity is not sufficient for the large FFT sizes that are used in many digital communications standards. A great reduction of this complexity can be achieved by using an efficient algorithm. The most known and used of them is unmistakably the Radix algorithm. However other algorithms like the WFTA can be very convenient for specific FFT sizes. In this section, we focus on the development of the Radix algorithm. The Mixed Radix algorithm will then be introduced. Also two advantages of the WFTA and a way to combine the Radix and WFTA algorithms is proposed. [5]

FFT and IFFT methods have three types. One is fixed radix FFT, Mixed radix FFT and Split Radix FFT. Fixed radix decompositions are algorithms in which the same decomposition is applied repeatedly to the OFT equation. The most common decompositions are radix-2, radix-4, radix-8 and radix-16. An algorithm of radix-r can reduce the order of computational complexity to O(N logr(N)). Mixed-radix refers to using a variety of radices in succession. One application of this method is to calculate FFTs of irregular sizes.

The radix algorithm

In 1965 Cooley and Tukey proposed a new way to compute the FFT for sizes that are power of 2, this is the birth of the Radix algorithms. The basic principle is based on the decomposition of a FFT of size N in two FFTs of size N2.

Decimation in time / Decimation in frequency

This decomposition can be achieved by two means which are called Decimation in Time (DIT) and Decimation in Frequency (DIF) depending on the regrouped terms.

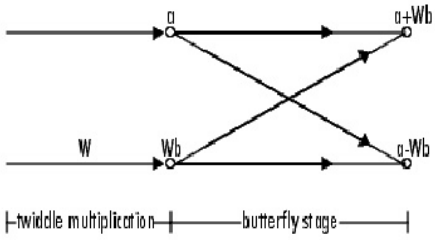


Figure 2. Decimation in Time

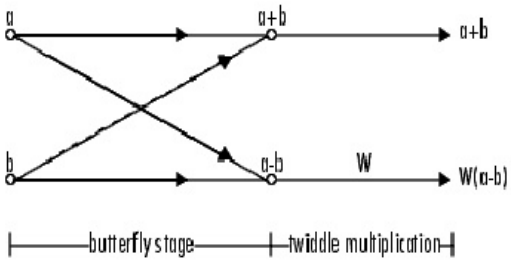


Figure 3. Decimation in Frequency

Radix-R algorithm

As the Radix-2 butterfly computes a FFT of size 2, the Radix-4 butterfly does the same for FFTs of size 4. The four outputs of this butterfly only need trivial multiplications by {−1, 1,−j, j}.

 ISSN: 2321-8169
1979 – 1982

Twiddle Factor Calculations:

Figure 4 explains the cyclic features (twiddle factors) of the exponential function for W=8.

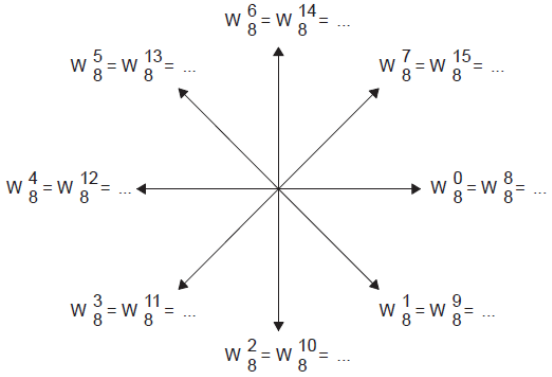


Figure 4. Twiddle Factors

Explicitly, the twiddle factors translate into the following:

W0/8 =W8/8 = cos (0°) - j sin (0°) = 1-j0
W1/8 = W9/8= cos (45°) - j sin (90°)= 0.7- j0.7
W2/8 =W10/8 = cos (90°) - j sin (90°) = 0 – j1
W3/8 = W11/8 = cos (135°) - j sin (135°) = 0.7 - j0.7
W4/8 = W12/8 = cos (180°) - j sin (180°) = -1 - j0
W5/8= W13/8 = cos (225°) - j sin (225°) = - 0.7- j0.7
W6/8 = W14/8 =cos (270°) - j sin (270°) = 0 + j1
W7/8 = W15/8 = cos (315°) - j sin (315°) = 0.7- j0.7

In similar manner twiddle factors for W=2, 4, 16,.. can be calculated.

The radix algorithm:

In 1965 Cooley and Tukey proposed a new way to compute the FFT for sizes that are power of 2, this is the birth of the Radix algorithms. The basic principle is based on the decomposition of a FFT of size N in two FFTs of size N2.

Decimation in time / Decimation in frequency:

This decomposition can be achieved by two means which are called Decimation In Time (DIT) and Decimation In Frequency (DIF) depending on the regrouped terms.

Radix-R algorithm:

As the Radix-2 butterfly computes a FFT of size 2, the Radix-4 butterfly does the same for FFTs of size 4. The four outputs of this butterfly only need trivial multiplications by {−1, 1,−j, j} as presented below,

$$X(0) = x(0) + x(1) + x(2) + x(3)$$
$$X(1) = x(0) + jx(1) - x(2) - jx(3)$$
$$X(2) = x(0) - x(1) + x(2) - x(3)$$
$$X(3) = x(0) -j x(1) - x(2) +j x(3)$$

Mixed-radix algorithm:

The Radix-2 algorithm proposed by Cooley and Tukey can be derived to provide the mixed-Radix algorithm (R. C. Singleton, 1969) (G. L. DeMuth, 1989). This last one employs the combination of different Radix-R algorithms allowing the computation of more FFT sizes. For example, to compute a 12-points FFT, the size N can be decomposed in N = 2 × 2 × 3. In this configuration, the FFT will be computed by a first and a second stage of six Radix-2 butterflies and a third stage of four Radix-3 butterflies. Nevertheless this FFT can also be computed by other decompositions such as N = 2 × 3 × 2, N = 3 × 2 × 2, N = 3 × 4 or N = 4 × 3. Due to the reduced number of computation stages and to the optimal complexity of the Radix-4 butterfly, the last two decompositions will require the lower complexity.

To summarize, the complexity of a Radix or Mixed-Radix algorithms will depend on the complexity of the Radix-R butterflies that are used and the number of twiddle factors necessary between the stages.

III. FLOATING POINT NUMBER FORMAT

Floating point numbers are one possible way of representing real numbers in binary format; the IEEE 754 standard presents two different floating point formats, Binary interchange format and Decimal interchange format. Fig. 2 shows the IEEE 754 single precision binary format representation; it consists of a one bit sign (S), an eight bit exponent (E), and a twenty three bit fraction (M or Mantissa). [5]If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significand then the number is said to be a normalized number; in this case the real number is represented by

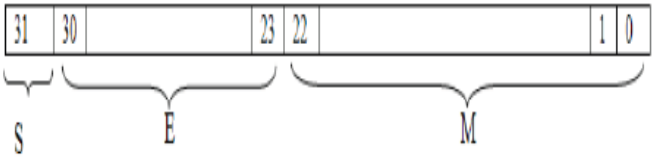


Figure 5. IEEE single precision floating point format

Sign Bit: This bit represents whether the number is positive or negative. 0 denotes a positive number and 1 denotes a negative number.

Exponent: This field represents both positive and negative exponents. This is done by adding a bias to the actual exponent in order to get the stored exponent. For IEEE 754 this value is 127.

Mantissa: This field is also known as the significant and represents the precision bits of the number. It comprises of implicit leading bits and the fraction bits.

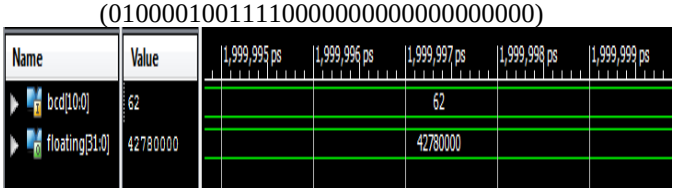
TABLE II. SINGLE AND DOUBLE PRECISION VALUES

Precision	Sign	Exponent	Fraction	Bias
Single Precision	1[32]	8[30-23]	23[22-00]	127
Double Precision	1[63]	11[65-52]	52[51-00]	1023

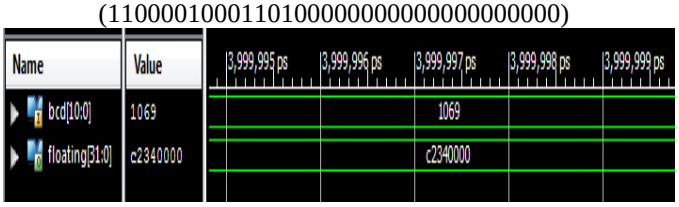
In proposed work, the BCD input is first converted into floating number format. The process of addition, subtraction and multiplication in the middle stage i.e. the complex multiplier takes place in floating point format only. At the end the floating output is again converted into BCD. The system can process signed, unsigned and decimal numbers thereby increasing the range.

IV. SIMULATION RESULTS

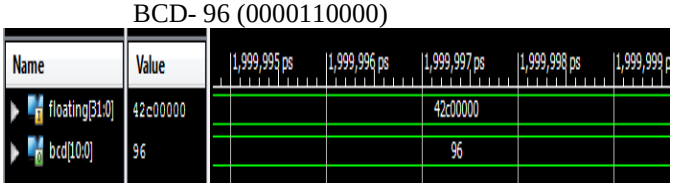
1. BCD to Floating Point Number
   Example- Positive BCD number – 62 Floating Representation – 42780000
   (01000010011110000000000000000000)



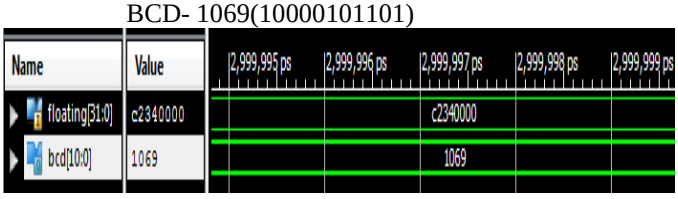
2. BCD to Floating Point Number(Negative Number)
   Example- BCD- 1069(10000101101)
   Float number- C2340000
   (11000010001101000000000000000000)



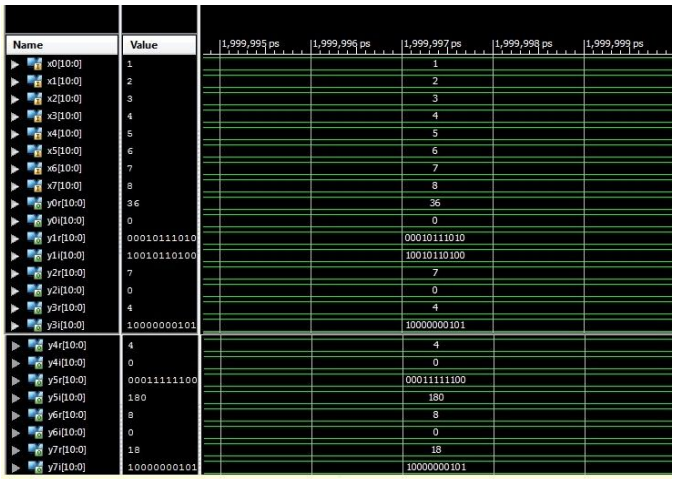
3. Float to BCD
   Example- Float number- 42C00000
   (01000010110000000000000000000)
   BCD- 96 (0000110000)



4. Float to BCD(Negative Number)
   Example- Float number- C2340000
   (11000010001101000000000000000000)
   BCD- 1069(10000101101)



5. Eight Point FFT



CONCLUSION

This paper represents a reconfigurable FFT/IFFT architecture which is capable of processing positive, negative and fractional numbers by making the use of floating point format. The results obtained in this thesis work turn up to the conclusion that making the use of floating point format helps to increase the range of the system. The system is capable of processing signed, unsigned, decimal, and fractional numbers. The twiddle factor can be incorporated with decimal points which results into appropriate answers.

REFERENCES

[1] BaigI,Jeoti V 'DCTprecoded SLM technique for PAPR Reduction' Intelligent and Advanced System international Conference,15-17 June 2010
[2] S.P.Vimal ,K.R.Shankar Kumar 'A New SLM Technique for PAPR Reduction in OFDM Systems', European Journal for scientific Research ISSN 1450-216X vol.65,No.2,2011,pp.221-230
[3] Yutian Zhao, Ahmet T. Erdogan, and Tughrul Arslan, "A Novel Low-Power Reconfigurable FFT Processor" IEEE 2005
[4] Qingwang Lu, Xin'an Wang and JiuChong Niu ShenZhen, China "A Low-power Variable-length FFT Processor Base on Radix-24 Algorithm" IEEE 2009
[5] Md. Nooruzzaman Khan M. Mohamed Ismail Dr. P .K. Jawahar "An Efficient FFT IIFFT Architecture for Wireless communication" ICCSP 2012 IEEE
[6] OFDM Simulation using Matlab 'Smart Research Laboratory'faculty advisor Dr.Mary Ann Ingram ,Guillermo Acosta ,Aug2000
[7] Md Nooruzzaman Khan M.Mohamed Ismail Dr.P .K.J awahar II M.Tech(VLSI & Embedded Systems) Associate Professor Professor "An Efficient FFT IIFFT Architecture for Wireless communication" ICCSP-'12
[8] Preethi Sudha Gollamudi, M. Kamaraju Dept. of Electronics & Communications Engineering Gudlavalleru Engineering College, Andhra Pradesh, India-769008 "Design Of High Performance IEEE- 754 Single Precision (32 bit) Floating Point Adder Using VHDL" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 7, July – 2013 IJERTIJERT ISSN: 2278-0181
[9] Sharon Thomas & 2V Sarada 1 Dept. of VLSI Design, 2Department of ECE, 1&2 SRM University " Design of Reconfigurable FFT Processor With Reduced Area And Power "ISSN (PRINT) : 2320 – 8945, Volume -1, Issue -4, 2013
[10] Konstantinos E. MANOLOPOULOS, Konstantinos G. NAKOS, Dionysios I. REISIS and Nikolaos G. VLASSOPOULOS Electronics Laboratory, Department of Physics National and Capodistrian University of Athens "Reconfigurable Fast Fourier Transform Architecture for Orthogonal Frequency Division Multiplexing Systems"