DESIGN AND IMPLEMENTATION OF RADIX-4 FAST FOURIER TRANSFORM IN ASIC CHIP WITH 0.18 µm STANDARD CMOS TECHNOLOGY

by

SIVA KUMAR PALANIAPPAN

Thesis submitted in fulfillment of the requirements

for the degree of

Master of Science

June 2008

ACKNOWLEDGEMENTS

As for the most projects, this research work would not have been successful without a proper help and guidance. I would like to extend sincere gratitude to those who have supported me to complete this project and my years in this postgraduate field of studies.

First of all, I would like to thank Dr. Tun Zainal Azni Zulkifli, my principle advisor in this Masters project. His constant encouragement support provided me the confidence to do this thesis work efficiently. Additionally, I also thank Dr. Tun Zainal and Dr. Othman Sidek for providing me financial support by offering me research officer position. I will also like to thank Madam Norlaili Mohd Noh for the moral supports in these 2 years of program. I am grateful to the assistant, guidance and support given by Mr. Zulfiqar Ali Abdul Aziz and Mr. Arjuna Marzuki. I am thankful to Silterra Malaysia Sdn. Bhd. for providing the fabrication of the FFT chip and Artisan for the standard cell libraries. I would like to thank Harikrishnan Ramiah, Nik Ismaliza Nik Idris, Shukri Korakkottil Kunhi Mohd, Ruhaifi Abdullah Zawawi and Sit Yoke Leen as member of postgraduate studies for the moral and technical support.

On a personal level, my deepest thanks to my grandmother and my parents for their unwavering love and encouragement. I am thankful too for special people I've blessed to have in my life; among them are: uncle (Selvarajoo Ramasamy and Subramaniam Ramasamy), aunty (Chitra Sellapan and Subhatra Devi), Pramila Gunasekaran, Sheela Devi Palaniappan, Prakash Babu Balakrishnan, Vicknatheeban Tharmalingam, Ronald Reegan Raipen, Pragash Gunasegaran, Mohamad Habib Mohamad Haniffah, Jega Divan, Nitiabalan Karrupayah and Rizah Kamaruddin.

CHAPTER FOUR : ASIC DESIGN METHODOLOGY

CHAPTER FIVE : 16-POINT RADIX-4 PROPOSED FFT DESIGN

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuits
BPSK	Binary Phase Shift Keying
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
CTS	Clock Tree Synthesis
DAC	Digital to Analog Converter
DFT	Discrete Fourier Transform
DIF	Decimation in Frequency
DIT	Decimation in Time
DRC	Design Rule Check
DSP	Digital Signal Processing
DUT	Device Under Test
ESD	Electrostatic Discharge
FFT	Fast Fourier Transform
FSM	Finite State Machine
HDL	Hardware Description Language
IC	Integrated Circuit
ICI	Inter-channel Interference
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
I/O	Input Output
ISI	Inter-symbol Interference
LAN	Local Area Network
LSB	Least Significant Bit

LEF	Library Exchange Format
LVS	Layout Versus Schematic
MCM	Multi Carrier Modulation
MSB	Most Significant Bit
OFDM	Orthogonal Frequency Division Multiplexing
P/S	Parallel To Serial Converter
QAM	Quadrature Amplitude Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register Transfer Level
SCM	Single Carrier Modulation
SDC	Synopsys Design Constraint
SDF	Standard Delay Format
SMU	Signal Monitor Unit
SNR	Signal to Noise Ratio
S/P	Serial To Parallel Converter
VHDL	Very High Description Language
VLSI	Very Large Integrated Circuit

DESIGN AND IMPLEMENTATION OF RADIX-4 FAST FOURIER TRANSFORM IN ASIC CHIP WITH 0.18 µm STANDARD CMOS TECHNOLOGY

by

SIVA KUMAR PALANIAPPAN

Thesis submitted in fulfillment of the requirements

for the degree of

Master of Science

June 2008

REKABENTUK DAN PERLAKSANAAN RADIKS-4 JELMAAN FOURIER PANTAS DALAM CIP ASIC DENGAN TEKNOLOGI 0.18 µm PIAWAIAN CMOS

ABSTRAK

Jelmaan Fourier pantas (FFT) merupakan blok yang penting dan digunakan secara meluas dalam algoritma pemprosesan isyarat digital. Dengan kemunculan teknologi pemprosesan semikonduktor dalam sistem VLSI, pencapaian rekaan FFT telah meningkat dengan drastik dan digunakan dalam rekaan aplikasi mudah alih. Walau bagaimanapun, dengan perkembangan teknologi semikonduktor yang lebih ke arah saiz dan geometri kecil, rekaan FFT turut menghadapi cabaran dalam peningkatan kuasa dalam rekaan tersebut. Permintaan terhadap sistem mudah alih memerlukan kuasa dan keluasan yang rendah. Justeru itu, projek ini mencadangkan algoritma untuk mengurangkan penggunaan kuasa dan keluasan dengan meminimumkan pendaraban dalam algorithma FFT rama-rama. Maka, nilai pengiraan yang berulang daripada algorithma dikira awal dan disimpan dalam ingatan daftar untuk pengiraan seterusnya. Cara ini membolehkan penurunan yang drastik dalam kuasa dan keluasan aktif pada rekaan asas FFT. Kuasa cip bagi rakabentuk projek ini adalah 18 mW dimana pegurangan sebanyak 1.47 % berbanding rekabentuk yang sedia ada. Keluasan aktif cip adalah sebanyak 0.46 mm^2 dimana penurunan sebanyak 11.36 % diperhatikan berbanding rekabentuk yang sedia ada. Rekaan 16-titik radix-4 0.18 µm piawaian CMOS diproses dengan frekuensi masa 40 MHz. Cip ini beroperasi pada penjana kuasa 1.8 V. Pengenalan ini sangat berguna dalam aplikasi pendaraban frekuensi ortogon sisi berasaskan sistem tanpa wayar LAN yang memerlukan sub–blok FFT untuk menukar domain masa isyarat ke domain frekuensi.

DESIGN AND IMPLEMENTATION OF RADIX-4 FAST FOURIER TRANSFORM IN ASIC CHIP WITH 0.18 µm STANDARD CMOS TECHNOLOGY

ABSTRACT

The Fast Fourier Transform (FFT) is a critical block and widely used in digital signal processing algorithm. With the advent of semiconductor processing technology in VLSI system, it has enabled the performance of FFT design to increase steadily and applied in portable application design. However, as semiconductor technologies move toward finer size and geometries FFT design has also faced challenges in power increment in the design. The portability requirement of the system is to have a less power and area consumption. The reported conventional radix-2 architecture observed to consume large chip area in conjunction with the increased number of hardware and power consumption corresponding to the in efficiency of the algorithm. The highlighted criterion increases significantly contributes to the complexity of the radix-4 implementation. This work proposed a method to reduce the power and area consumption by reducing number of multiplication in FFT radix-4 butterfly algorithm. Hence, repeated values of pre-calculated algorithms are kept in the memory register for next calculation. This method can drastically reduce both power and active area of the FFT core design. The proposed architecture dissipates 1.47 % less power in comparison with the conventional architecture resulting in 18 mW power dissipation. Active area of the chip is reduced 11.36 % measuring 0.46 mm^2 of chip area. 16-point radix-4 has been design in a standard 0.18 µm CMOS process with clock frequency of 40 MHz. This chip operates with the voltage headroom of 1.8 V. This approach is very useful for Orthogonal Frequency Division Multiplexing (OFDM) based wireless LAN system which required FFT sub-block to change time domain signal to frequency domain.

CONTRIBUTIONS OF THE THESIS

The principal contributions of this research are:

i) Development of an enhanced FFT radix-4 algorithm. Radix-4 algorithm is more attractive since it requires less number of multiplication operations for FFT processor which reduces the complexity of computation. Modification of algorithm in term of mathematical simplification and the reducing of repeated computation using register memory has enhanced the power and active area consumption of the FFT design. Attention is given on improvement of twiddle factor usage in radix-4 FFT design which reduces the complexity of conventional radix-4 architecture. Area and power saving of enhanced radix-4 have been achieved without imposing any additional computational or structural complexity in the radix-4 FFT algorithm.

ii) Comparison between the architecture of the conventional radix-2, radix-4 and the enhanced radix-4 had been carried out in a study of number of stages and number of computations in the contents of algorithm optimization.

iii) Design of a single 16-point radix-4 FFT chip on a platform of 0.18 µm standard CMOS technology in ASIC. The proposed FFT processor has a better power and area consumption compared to the conventional radix-4 design with a 1.8 V supply voltage.

iv) An 8-point radix-2 FFT ASIC chip was successfully fabricated in utilizing in 0.18 µm standard CMOS technology. This chip was packaged in Quikpak for measurement. Measurement strategies were clearly described in details.

CHAPTER 1
INTRODUCTION

1.0 Overview

Application using frequency analysis of discrete-time signals in digital signal processor is the most convenient method especially in general-purpose digital computer or specially designed digital hardware (Proakis and Manolakis, 1996 and Schoukens *et. al*, 2004). Frequency analysis is performed on a discrete-time signal $\{x(n)\}$ by converting the time-domain sequence to an equivalent frequency-domain presentation. A straightforward representation of the Fourier transform is illustrated in Figure 1.1 (Brigham, 1974). From the figure, it shows the essence of the Fourier transform of a waveform is to decompose or separate the waveform into a sum of sinusoids of different frequency. Thus, the pictorial representation of the Fourier transforms is a diagram which displays the amplitude and frequency of each of the determined sinusoids.

The Fourier Transform identifies or distinguishes the different frequency sinusoids and their respective amplitudes which combine to form an arbitrary waveform. Mathematically, this relationship is stated as (Bracewell, 1978)

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt \,. \tag{1.1}$$

It operates on the function $s(t)$ and produces $S(f)$, which is referred to as the Fourier transform of $s(t)$, where $s(t)$ is the waveform to be decomposed into sum of sinusoids, $S(f)$ is the Fourier transform of $s(t)$. The constant j represents the imaginary quantity $\sqrt{-1}$. In general the function $s(t)$ and $S(f)$ are complex values.

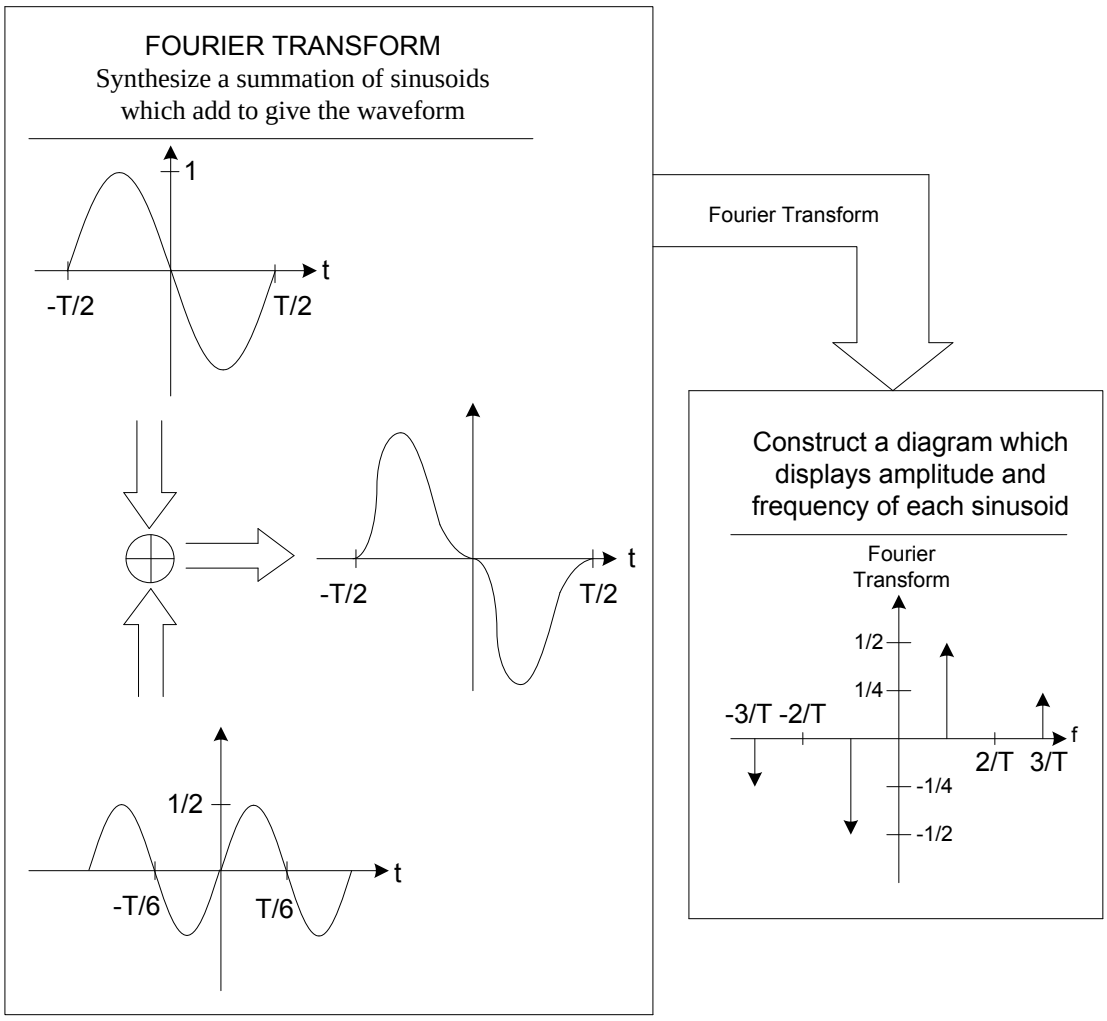

Figure 1.1: Interpretation of the Fourier Transform (Brigham, 1974).

Because of the way the Fourier integral is defined, not every function $s(t)$ can be transformed to $S(f)$. Therefore, two sufficient conditions are considered for the existence of Fourier transforms (Brigham, 1974) which are:

a. Condition 1

The integral of $s(t)$ from -∞ to ∞ exists. That is,

$$\int_{-\infty}^{\infty} |s(t)| dt < \infty \tag{1.2}$$

b. Condition 2

Any discontinuities in $s(t)$ are bounded. Figure 1.2 illustrates a discontinuous at $s = s_o$, but only over a finite distance, so this functions meets the second condition.

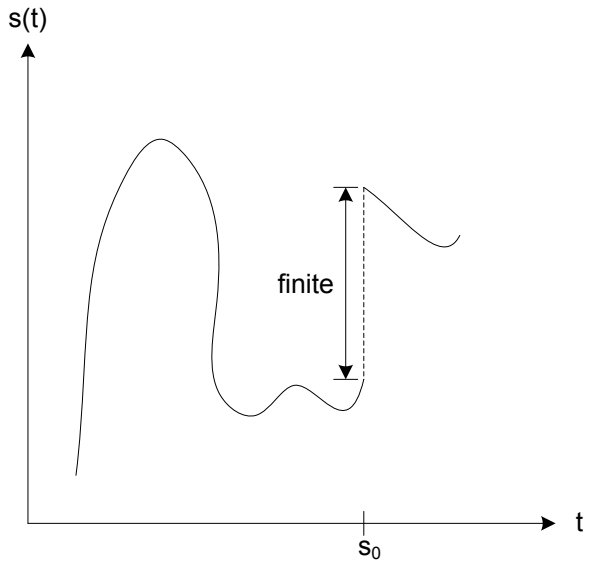

Figure 1.2: Example function of finite discontinuity

Because these conditions are only sufficient, many functions which do not meet these conditions nevertheless have Fourier transforms. In fact, such useful functions as $\sin(t)$, step function $H(t)$ fall into this category.

While the generality of the continuous Fourier transform works well for the study of Fourier transform theory, it has limited direct practical use. One of the reasons for the limited use is because any signal $s(t)$ that exists over a finite interval has a spectrum $S(f)$ which extends to infinity. Since digital computers have finite memory, they can neither store nor process the infinite number of data points that would be needed to describe an arbitrary, infinite function. Thus, for practical functions, a special case of the Fourier transform known as the discrete Fourier transform is used. This transform is defined for finite length $s(t)$ and $S(f)$ described by a finite number of samples, and is discussed in chapter 3.

The representation of finite Fourier Transform is given in $\{X(\omega)\}$ of the sequence of $\{x(n)\}$. Since $\{X(\omega)\}$ is continuous function in frequency domain Fourier Transform, it is not computationally convenient to be represented by the sequence of $\{x(n)\}$. However, the sequence of $\{x(n)\}$ can be represented by sampling the spectrum $\{X(\omega)\}$. This frequency domain representation leads to the discrete Fourier Transform (DFT), which is an important algorithm for performing frequency analysis of discrete-time signal.

Discrete Fourier Transform (DFT) based signal processing is widely used and plays a significant role in digital signal processing algorithm. Generally, DFTs are rarely computed directly, but instead are calculated using the Fast Fourier Transform (FFT), which consumes a collection of algorithm that efficiently calculates the DFT of a sequence (Cheng *et. al*, 2005). Fast Fourier transform is a concept of expressing the discrete representation of a Fourier series or a Fourier integral at equally spaced points N point product, into a sub series product. Hence, this would reduce the number of multiplication and additions for complex data (Cooley and Tukey, 1993). With these advantages, the fast Fourier transform has become well known as a very efficient algorithm for calculating the discrete Fourier Transform (DFT) of a sequence N numbers (Heideman *et.al*, 1984). In the modern year, motivated by the emerging technology applications it has been applied in a wide range of fields such as communications, signal processing, instrumentation, biomedical engineering, numeric methods and applied mechanics (Chang and Parhi, 2003).

In recent development of communication system, OFDM is proposed as the primary modulation method (Lin *et. al*, 2005). The FFT and IFFT (Inverse Fast Fourier Transform) of OFDM sub-blocks are the key components of the system. Thus, FFT plays a

significant role in OFDM system in converting time domain signal to frequency domain representation. The effectiveness of FFT processor contributes to the optimization of the OFDM system (Zhong *et. al*, 2006). Consequently, FFT implementation in embedded communication system requires complex computation power and low power consumption. Therefore, ASIC (Application Specific Integrated Circuit) solutions have been widely used to implement communication systems (Lee *et. al*, 2004).

With the advent of this DSP application, the study of high performance VLSI (Very Large Integrated Circuit) architecture is increasing in importance (Chang and Nquyen, 2006). As semiconductor technology had moved towards finer size and geometries, both the available performance and functionality per die increases drastically. Unfortunately, the power and area consumption of the processors fabricated in advancing technologies also continues to grow with inefficient algorithm design. The increasing demand of portable and embedded applications has contributed significantly to this growing number of power-limited opportunities. Thus, this technology increase has resulted in the current situation, in which potential FFT application is limited by power and area consumption. The implementation of Fast Fourier Transform with the modification of the existing algorithm is brought up as the VLSI technologies improve in ASIC implementation.

1.1 Research goal

The main goal of this research is to enhance the algorithm and architectures necessary for high performance FFT processors implemented in VLSI semi-custom platform. The implementation of the enhanced architecture is realized in an ASIC (Application Specific Integrated Circuit) platform. The highlighted enhancement is supported by a strong foundation of improvement in power consumption and active chip area reduction. A suitable solution achieves the described goal.

1.1.1 Low power and area efficiency

Power consumption increase in a standard CMOS VLSI design for FFT application is due to the complex multiplication within the butterfly design. There are several methods of computing FFT algorithm in the signal processing in literature which involves different number of computation. This dissertation compares the conventional radix-4 architecture and enhanced radix-4 architecture in term of power and area by modification algorithm in FFT processor to reduce the number of mathematical computation which decreases the number of hardware. Concentration is given to FFT computation method where FFT algorithm encompasses many complex multiplication and addition within the butterfly processing unit which significantly consumes power in the FFT processor (Hasan *et. al*, 2003) with large active chip area consumption.

1.1.2 VLSI implementation

Alternate methodologies and technologies that offers hardware and software solution in designing Fast Fourier Transform are FPGA (Field Programmable Gate Array) implementation, general DSP processor implementation and CPLD (Complex Programmable Logic Device) programming (Mondwurf, 2002). In this work, a FFT having a specific application is designed in ASIC using top-down semi-custom approach utilizing 0.18 μm standard 1 poly 6 metal CMOS technology. Semi-custom design approach permits straightforward implementation compared to full-custom design. This method directly transforms a behavioral RTL description to a structural gate level netlist using a standard cell library furnished by the foundry. Thus, this method requires less design effort and typically contains fewer errors, compared to transistor level design (Brown and Vranesic, 2003). Therefore, in commercial settings, simpler design methodologies reduce

the time-to-market. Due to an absence of design complexity effort, further optimization of performance, power and area is concentrated in a time limited realm.

1.2 Chapters overview

This thesis is organized as follow. Chapter 2 describes the application of Fast Fourier Transform block in OFDM system. The integration blocks of the OFDM transceiver is explained in this chapter. Chapter 3 begins with an introduction of the Discrete Fourier Transform and the Fast Fourier Transform algorithm. A derivation of the FFT is also given in chapter 3, following to an overview of the major existing FFT algorithm. Radix-2 and radix-4 FFT radix calculation is elaborated in chapter 3.

Chapter 4 focuses on the design methodology of this project. Semi-custom design approach is explained in detail from the verilog verification to the GDSII file that is ready for fabrication. Chapter 5 describes the reduction in computation technique which leads to the enhancement of power and active area of the design. The remainder of chapter 5 focuses on the proposed FFT architecture and the control circuit design. The performance of the design is evaluated in the context algorithmic, state machine, timing diagram and architectural form. Conventional radix-4 and proposed radix-4 FFT 16-point is cross compared in terms of power and area of the design in chapter 6, as results and discussion. Output of simulation results are presented in this chapter. Chapter 7 describes the test methodology of FFT block in lab environment. Finally, chapter 8 summarizes the work presented and suggests areas for future work.

CHAPTER 2
SYSTEM OVERVIEW

2.0 Introduction

The basic concept of the Orthogonal Frequency Division Multiplexing (OFDM) is explained in this chapter. The OFDM transceiver is emphasized with the sections elaborating OFDM transceiver sublocks, on OFDM the FFT processor.

2.1 Basic theory of OFDM

The Orthogonal Frequency Division Multiplexing (OFDM) is a form of multi-carrier modulation technique that was first introduced more than four decades ago. This system transmits a single data stream over a number of lower rate sub-carriers (Hara and Prasad, 2003). The OFDM technique finds a vital place in consumer electronic products with the advances of digital signal processing (DSP) and very large integrated circuit (VLSI) technologies (Lin *et. al*, 2005).

Initially, before the introduction of the multi-carrier modulation technique, single carrier modulation is used in data transmission. This serial data transmission method transmits the information bearing sequentional symbols with the frequency spectrum of each symbol occupying the entire available bandwidth B_{SCM}. The single carrier modulation is substantial to fail due to the occurrence of interference (Intini, 2000). Figure 2.1 shows the comparison between single carrier and multi-carrier modulation. Subsequently, for multi-carrier modulation the bandwidth are divided into Δf, where the bandwidth of each Δf are sufficiently narrow with the sampling frequency of T_s (Hara and Prasad, 2003). A from the Figure 2.1, each sub-channel is associated with each carrier frequency (f_1, f_2, f_3, f_4) and the frequency sub-channels are not overlapping each other.

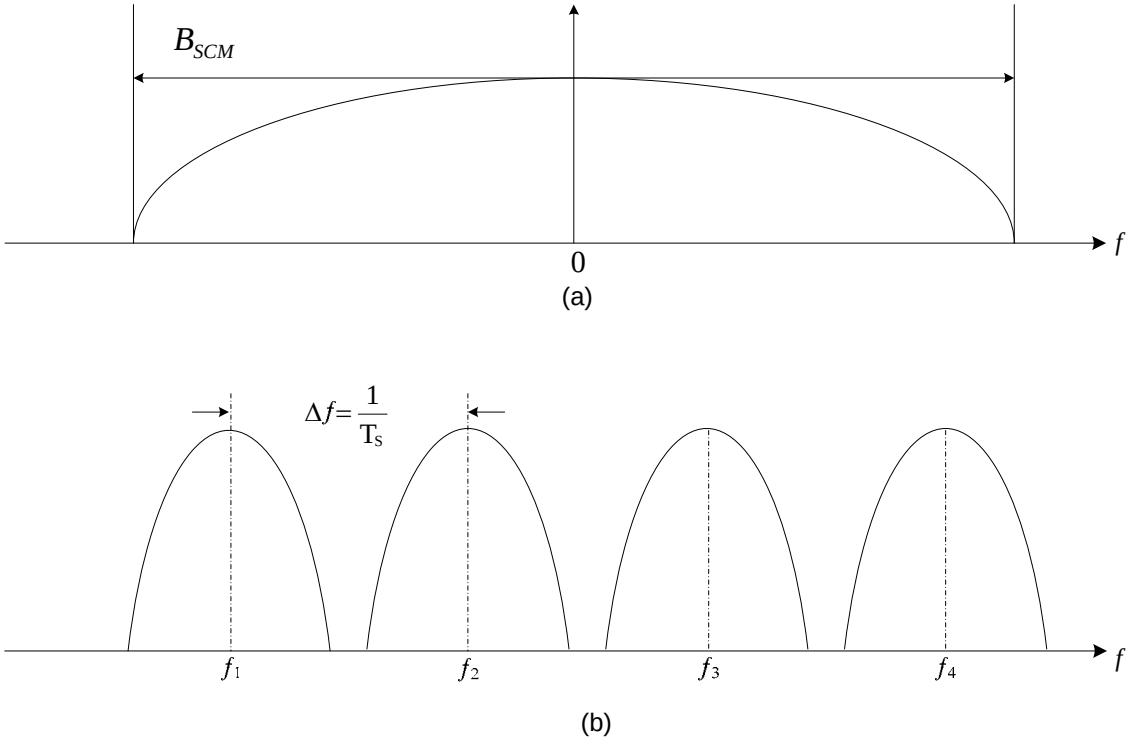

Figure 2.1: Frequency spectra of (a) single carrier transmitted signals (b) multi-carrier transmitted signals

However, this method leads to inefficient use of available spectrum. To avoid this problem, ideas are proposed from the mid-1960s to use multi-carrier data with overlapping sub-channels which are called OFDM. Figure 2.2 illustrates OFDM signal waveform. Overlapping multi-carrier modulation technique can lead to bandwidth optimization. To do this, the carriers must be mathematically orthogonal. Figure 2.2 shows that at the center frequency of (f_1, f_2, f_3) each sub-carrier, there is no crosstalk from other sub-channels.

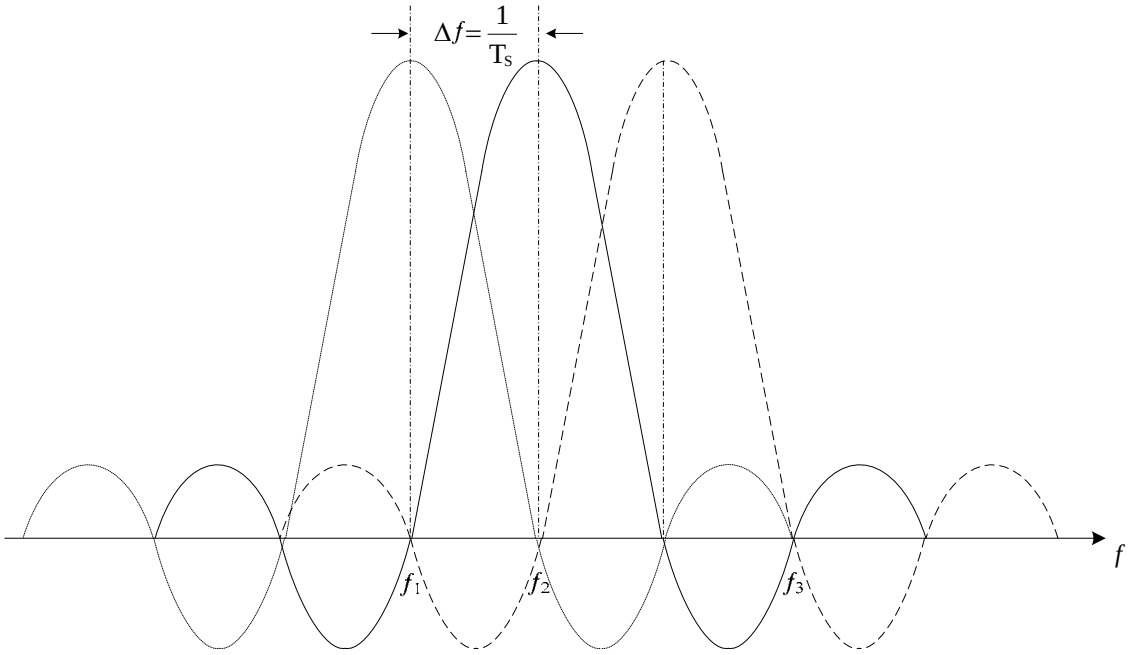

Figure 2.2: OFDM waveform signal

The basic idea of OFDM is to divide the available spectrum into N orthogonal sub-channels. The digital baseband blocks (Proakis and Salehi, 2002) of an OFDM transceiver, is illustrated in Figure 2.3.

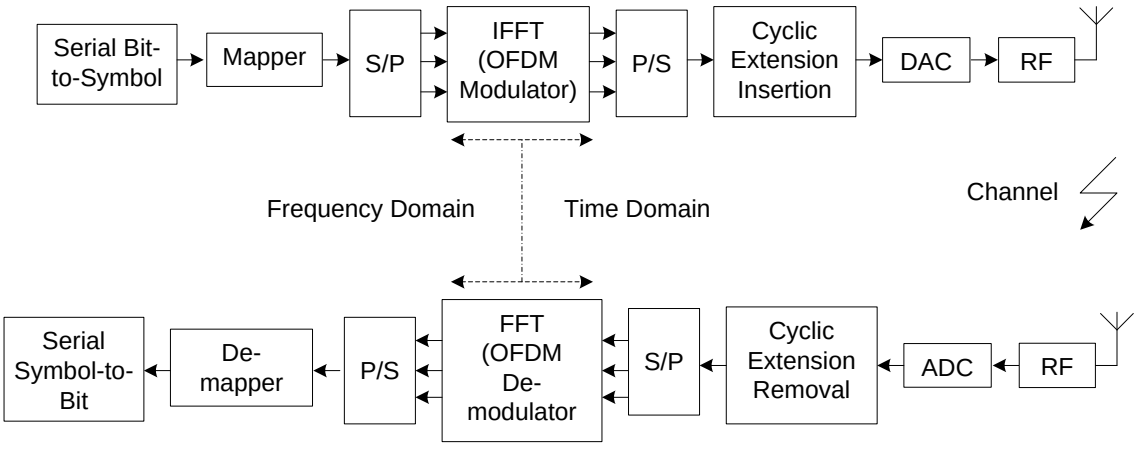

Figure 2.3: Simplified transceiver block of OFDM system

The OFDM transceiver block contains a mapper / demapper, a serial to parallel (S/P) converter / parallel to serial (P/S) converter, a digital to analog converter (DAC), an analog to digital converter (ADC), a cyclic insertion/removal and a FFT/IFFT. In the sub-block of OFDM transmitter, mapper block converts the data to signal located in the frequency domain where each sub-channel is assigned one signal. Next, these signals are transformed the time domain with Inverse Fast Fourier Transform (IFFT). Fast Fourier Transform algorithm is an efficient and fast method to implement the Discrete Fourier Transform which are based on divide and conquer method (Proakis and Manolakis, 1996). Divide and conquer approach is elaborated in chapter 3. The final block of OFDM transmitter is to insert a cyclic extension to remove the effect of inter-symbol interference (ISI) and inter-channel interference (ICI).

The OFDM receiver block is also described in Figure 2.3. After down-conversion and analog-to-digital conversion the cyclic extension is removed. The data is fed into FFT block as a parallel input after going through serial to parallel converter. FFT block is used to demodulate the N sub-carriers of the OFDM signal and transformed back to the frequency domain.

2.2 Mapping and De-mapping

Input data is converted to complex valued signal points by the mapper block. These data signals are converted to signal based on the constellation, e.g. BPSK or 64-QAM. Figure 2.4, represents the BPSK, QAM and 16-QAM constellation plot where the in-phase (I) axis corresponds to the real part and the quadrature (Q) axis corresponds to the imaginary part of the output signal. The amount of data transmitted on each sub-carrier is

dependant on the type of constellation. For example BPSK and 16-QAM transmit one and four data bits per sub carrier.

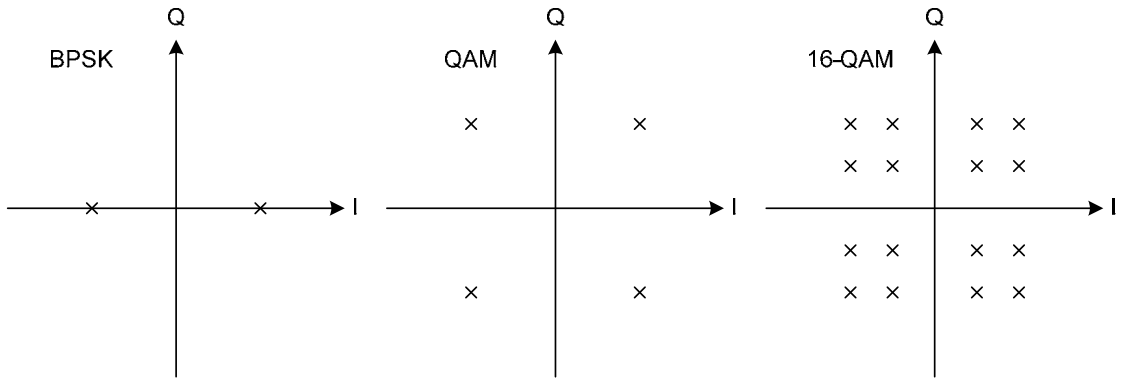

Figure 2.4: Typical constellations for wireless system application

From the Figure 2.4, 16-QAM has four possible values on both the (I) and (Q) axis and thus two bits are needed to specify the location on each axis. Channel with high interference need a smaller constellation as the BPSK, since the required signal to noise ratio (SNR) in the receiver is low (Intini, 2000). Larger constellations like 16-QAM are more sensitive to noise compared to BPSK because the distance between signal points decreases as the constellation increase with given average signal power are same for all the constellations. De-mapping is performed to return from signal points to data points.

2.3 IFFT and FFT block

The Fast Fourier Transform (FFT) converts the signals from time domain representation to frequency domain representation and consequently the IFFT performs the reverse operation. Signals are processed in frequency domain due to simplified computation as compared to the time domain computation, e.g. convolution in the time domain becomes multiplication in the frequency domain. These signals are converted back

to time domain using IFFT block in order to reduce the number of backend RF-oscillators and demodulators.

OFDM transmits a large number of narrowband carriers, closely spaced in the frequency domain. The available bandwidth in frequency is split into N sub-channels, one for each sub-carrier, where each has a power spectrum shape of a squared sinc pulse described in Figure 2.2.

The sub-carrier can be separated again with FFT algorithm even tough the spectrum is overlapped because IFFT is a linear operation. It is also the case for signals which had passed a multi-path channel, due to the cyclic extension in the OFDM transmitter. The property of orthogonality signals prevents the sub-carrier to be affected by other sub-carriers. This would ensure the signal is not corrupted. The orthogonality of sub-carriers in OFDM can be maintained and individual sub-channel is be completely separated by the FFT at the receiver when there are no inter-symbol interference (ISI) and inter-carrier interference (ICI) introduced by transmission channel distortion.

2.4 Cyclic extension and guard interval insertion

One way to prevent inter-symbol-interference in channel which caused by delay spread is to create an extended guard interval between each OFDM (Morrison *et.al*, 2001). The guard time is chosen larger than the expected delay spread such that multi-path components from one symbol cannot interfere with next symbol and could consist of null signal (Initini, 2000). However, with the existing delay spread the inter-carrier interference is unavoidable.

This problem is alleviated when the duration of cyclic prefix or suffix is extended respective to the length of the channel impulse response. There are two types of cyclic extensions, the cyclic prefix and the cyclic suffix. The cyclic prefix is a copy of the last M_1 sample from the IFFT, which are placed at the beginning of the OFDM frame, as shown in Figure 2.5. M_2 time samples are copied from the starting point of the original OFDM sequence block and is extended as suffix. There are two reasons to use a cyclic extension for a given guard interval at the transmitter while discarding it at receiver. Assuming that the cyclic prefix is longer than the channel impulse response, where L is the channel impulse response, the convolution between the data and the channel impulse response would resamble circular convolution and therefore no ICI will occur, for $M = (M_1 + M_2) \geq L$. Eventually, interference from the previous symbol will only affect the cyclic prefix and ISI is avoided. However, transmission rate would decrease if the cyclic extension is very large. The data rate efficiency would decrease in a factor of $N/(N+M)$.

Figure 2.5 and Figure 2.6 illustrates the cyclic extension procedure for the respective cyclic prefix and cyclic suffix. In the receiver the cyclic prefix is discarded before the FFT, but can be used to support synchronization due to the correlation with the last part of the OFDM frame (Morrison *et. al*, 2001). Thus, to ensure the efficiency of data rate, a guard interval of not more than 10% to 20% of the symbol duration is used. In some of the systems, only cyclic prefix are used as a guard interval (Morrison *et. al*, 2001).

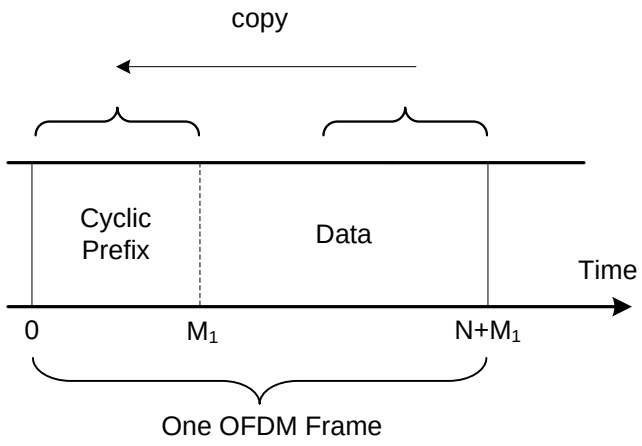

Figure 2.5: Cyclic Prefix Extension

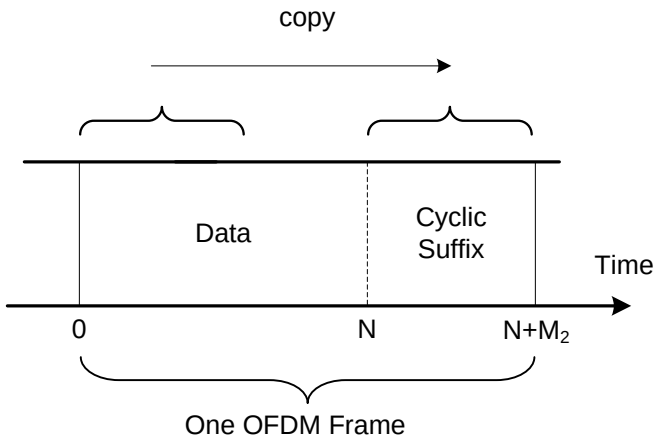

Figure 2.6: Cyclic Suffix Extension

CHAPTER 3
FAST FOURIER TRANSFORM ALGORITHM

3.0 Introduction

In this chapter, description of several various computation methods of Fast Fourier Transform (FFT) in determining the DFT is presented. In order to perform frequency analysis on a discrete time signal $\{x(n)\}$, the time-domain sequence is converted to an equivalent frequency-domain representation. Such a frequency-domain representation leads to the DFT computation, which is a powerful tool to perform time domain signal analysis in frequency domain (Chassaing, 2002).

3.1 The Discrete Fourier Transform

Due to the importance of Discrete Fourier Transform (DFT) in signal processing application, it is critical to have an efficient method to compute this algorithm (El-Khashab and Swartzlander, 2003). DFT operates on a N-point sequence of numbers, referred to as $x(n)$. The value $x(n)$ is presented in time domain data and usually can be taught as a uniformly sampled version of a finite period of a continuous function $f(x)$.

The DFT of $x(n)$ sequence is transformed to $X(k)$ in frequency domain representation employing by using Discrete Fourier Transform. The functions $x(n)$ and $X(k)$ is generally represented in complex signal form, given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e_N^{-j2\pi nk/N} \quad , \qquad k = 0,1,...,N-1 . \tag{3.1}$$

The DFT computation for a given sequence N complex-valued numbers is described in equation (3.1), where $x(n)$ is the input time domain representation and N is the number of input to the DFT. The value n represents the discrete time-domain index

and k is the normalized frequency domain index (Bi and Jones, 1989). The simplified equation is described by introducing W_N as given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad , \qquad k = 0,1,...,N-1 . \tag{3.2}$$

Frequency domain data can be changed to time domain employing Inverse Discrete Fourier Transform in, which the $X(k)$ is transform back to $x(n)$.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} \quad , \qquad n = 0,1,...,N-1$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} x(k) W_N^{-nk} \quad , \qquad n = 0,1,...,N-1 . \tag{3.3}$$

In this chapter, the description of efficient computation is discussed on DFT methods since the IDFT and DFT consumes the same type of computational algorithm. From the computation of each value of k, it is observed that direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and $N-1$ complex additions ($4N-2$ real additions). Eventually, to compute all N values of the DFT requires N^2 complex multiplications and $N^2 - N$ complex additions. It is noted from the Euler equation, W_N, is denoted by

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

$$= \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right). \tag{3.4}$$

For simplification the variable W_N is often known as the "N_{th} root of unity" since $\left(W_N\right)^N = e^{-j2\pi} = 1$. It is inefficient to compute the Fourier transform in DFT because this algorithm does not exploit the symmetry and periodicity properties of the phase factor W_N (Proakis and Manolakis, 1996).

Periodicity property: $W_N^{k+N} = W_N^k$ (3.5)

Symmetry property: $W_N^{k+N/2} = -W_N^k$ (3.6)

3.2 Fast Fourier Transform

Fast Fourier Transform is a high efficient algorithm to compute the DFT. For the given round of error the FFT algorithm results in an equivalent data representation with the calculated DFT computation. The basic idea of this approach is to decompose the N-point DFT into successively smaller DFT. Eventually, this approach leads to a family of highly efficient computation of FFT algorithm.

Fast Fourier Transform is popularized by J. W. Cooley of IBM and John W. Tukey of Princeton University when they published a paper in 1965 which describes the fast computation of DFT (Cooley *et. al*, 1967). Several architectures have been proposed based on Cooley-Tukey algorithm to further reduce the computational complexity, including radix-2, radix-4 and split radix.

3.2.1 Divide and Conquer

Divide-and-Conquer approach is adopted in FFT algorithm to make the computation more efficient (Alberto *et. al*, 2005 and Proakis and Manolakis, 1996). To describe the Divide-and-Conquer approach, the value of N is factored as a product of two integers, which is

$$N = LM . \tag{3.7}$$

The sequence $x(n)$, given that $0 \le n \le N-1$ is stored in two-dimension by ℓ and m, where $0 \le \ell \le L-1$ and $0 \le m \le M-1$. A tabulation of representation is presented in Figure 3. to illustrate the value of ℓ and m in the divide and conquer approach. The value of ℓ is indexed in a row and m is in the column. Thus, the sequence $x(n)$ is stored in a rectangular array by mapping of index n to the indexes (ℓ, m) as follows:

$$n = \ell + mL . \tag{3.8}$$

From equation (3.8), first row of the arrangement consist first L elements of $x(n)$, the second row consists of the next L elements of $x(n)$, and so on. Thus, the stored sequence $x(n)$ is illustrated in Figure 3.1.

ℓ	m				
	0	1	2	…	M-1
0	x(0)	x(L)	x(2L)	…	x((M-1)L)
1	x(1)	x(L+1)	x(2L+1)	…	x((M-1)L+1)
2	x(2)	x(L+2)	x(2L+2)	…	x((M-1)L+2)
: :	: :	: :	: :	…	: :
L-1	x(L-1)	x(2L-1)	x(3L-1)	…	x(LM-1)

Figure 3.1: Divide and conquer data array arrangement

A similar arrangement is used to map index k to a pair of indices (p,q), where $0 \le p \le L-1$ and $0 \le q \le M-1$. Thus, the sequence $X(k)$ is stored in rectangular array by mapping index k to the indexes (p,q) as follows:

$$k = Mp + q\,. \tag{3.9}$$

$X(k)$ is mapped into the corresponding rectangular array $X(p,q)$ and $x(n)$ is mapped into the rectangular array $x(\ell,m)$. The DFT can be expressed as a double sum over the elements of $x(n)$ and $X(k)$ multiplied by the corresponding phase factors as follows:

$$X(p,q) = \sum_{m=0}^{M-1}\sum_{\ell=0}^{L-1} x(\ell,m) W_N^{(Mp+q)(mL+\ell)} \tag{3.10}$$

where

$$W_N^{(Mp+q)(mL+\ell)} = W_N^{MLmp} W_N^{mLq} W_N^{Mp\ell} W_N^{\ell q}\,, \tag{3.11}$$

$$W_N^{Nmp} = 1, W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq}\ , \tag{3.12}$$

and

$$W_N^{Mp\ell} = W_{N/M}^{p\ell} = W_L^{p\ell}\,. \tag{3.13}$$

For simplification, equation (3.10) can be expressed as

$$X(p,q) = \sum_{\ell=0}^{L-1} \left\{ W_N^{\ell q} \left[\sum_{m=0}^{M-1} x(\ell,m) W_M^{mq} \right] \right\} W_L^{\ell p} . \qquad (3.14)$$

From the equation (3.14), M -point is computed earlier,

$$F(\ell,q) = \sum_{m=0}^{M-1} x(\ell,m) W_M^{mq} , \qquad 0 \le q \le M-1 . \qquad (3.15)$$

with each rows of $\ell = 0,1,...,L-1$.

Next, new rectangular array $G(\ell,q)$ is defined as

$$G(\ell,q) = W_N^{\ell q} F(\ell,q) , \qquad 0 \le \ell \le L-1 \text{ and } 0 \le q \le M-1 . \qquad (3.16)$$

Finally, L -point is computed as

$$X(p,q) = \sum_{\ell=0}^{L-1} G(l,q) W_L^{\ell p} \qquad (3.17)$$

for the array $G(\ell,q)$ in each column of $q = 0,1,...,M-1$.

The development of Divide-and-Conquer methods influence the number of computation usage in term of additions and multiplications. From the equation (3.14), computation is performed for L DFTs for each M point. Thus, with this calculation it requires LM^2 complex multiplication and $LM(M-1)$ complex additions. The following equation (3.17) requires ML^2 complex multiplications and $ML(L-1)$ complex additions. Therefore, when $N = ML$, the computational complexity is

Complex multiplications: $N(M+L+1)$ (3.18)

and

Complex additions: $N(M+L-2)$. (3.19)

Referring to the DFT computation, number of multiplications has been reduced from N^2 to $N(M+L+1)$ and the number of additions has been reduced from $N(N-1)$ to $N(M+L-2)$.

3.2.2 Radix-2 FFT

The development of computationally efficient algorithms for the DFT using Divide-and-Conquer approach directly introduces a radix-2 method. This problem solving method is efficient when N is highly composite. In the case of, N can be factored as $N = r_1\ r_2\ r_3 ... r_v$. In particular to full fill the equation $N = r^v$, the r value are $r_1 = r_2 = ... r_v = r$. The number r is denoted as the radix of the FFT algorithm.

Radix-2 algorithm with N point length are split into M and L, whereby $M = N/2$ and $L = 2$ corresponding to decimation-in-time method. The signal $x(n)$ is partitioned according to its sample numbers, where $y(n)$ is the even samples and $z(n)$ is the odd samples, resulting in

$$y(n) = x(2n), \qquad n = 0,1,..., \frac{N}{2} - 1, \tag{3.20}$$

and

$$z(n) = x(2n+1), \qquad n = 0,1,..., \frac{N}{2} - 1. \tag{3.21}$$

From equation (3.2), the partitioned N point can be expressed as follow:

$$\begin{aligned} X(k) &= \sum_{n\ even} x(n) W_N^{kn} + \sum_{n\ odd} x(n) W_N^{kn} \\ &= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)}. \end{aligned} \tag{3.22}$$

For simplification, W_N^2 is represented as $W_{N/2}$. This can be expressed as

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} y(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} z(m) W_{N/2}^{km} \\ &= F_1(k) + W_N^k F_2(k), \qquad k = 0,1,..., N-1. \end{aligned} \tag{3.23}$$

Since $F_1(k)$ and $F_2(k)$ are periodic, it can be represented as $F_1(k+N/2)=F_1(k)$ and $F_2(k+N/2)=F_2(k)$ with period of $N/2$-point. Subsequently, the symmetrical property of the design results $W_N^{k+N/2}=-W_N^k$. Thus, equation (3.23) can be expressed as

$$X(k)=F_1(k)+W_N^kF_2(k)\,,\qquad k=0,1,...,\frac{N}{2}-1 \tag{3.24}$$

and

$$X(k+\frac{N}{2})=F_1(k)-W_N^kF_2(k)\,,\qquad k=0,1,...,\frac{N}{2}-1\,. \tag{3.25}$$

From the direct computation of $F_1(k)$ and $F_2(k)$, the number of complex multiplication required is $(N/2)^2$. Hence, there are $N/2$ additional complex multiplications in computing $W_N^kF_2(k)$. Therefore, it can be concluded that $X(k)$ requires $2(N/2)^2+N/2=N^2/2+N/2$ complex multiplications. It is observed that the number of multiplication is reduced from N^2 to $N^2/2+N/2$ compare to the DFT algorithm. To simplify the equation (3.24) and (3.25), the value are defined as

$$G_1(k)=F_1(k)\,,\qquad k=0,1,...,\frac{N}{2}-1\,, \tag{3.26}$$

and

$$G_2(k)=W_N^kF_2(k)\,,\qquad k=0,1,...,\frac{N}{2}-1\,. \tag{3.27}$$

With reference to equation (3.26) and (3.27), the DFT $X(k)$ can be expressed as

$$X(k)=G_1(k)+G_2(k)\,,\qquad k=0,1,...,\frac{N}{2}-1\,, \tag{3.28}$$

and

$$X(k+\frac{N}{2})=G_1(k)-G_2(k)\,,\qquad k=0,1,...,\frac{N}{2}-1\,. \tag{3.29}$$

The computation of DFT is described in equation (3.28) and (3.29) with the correspond illustration given in Figure 3.2.

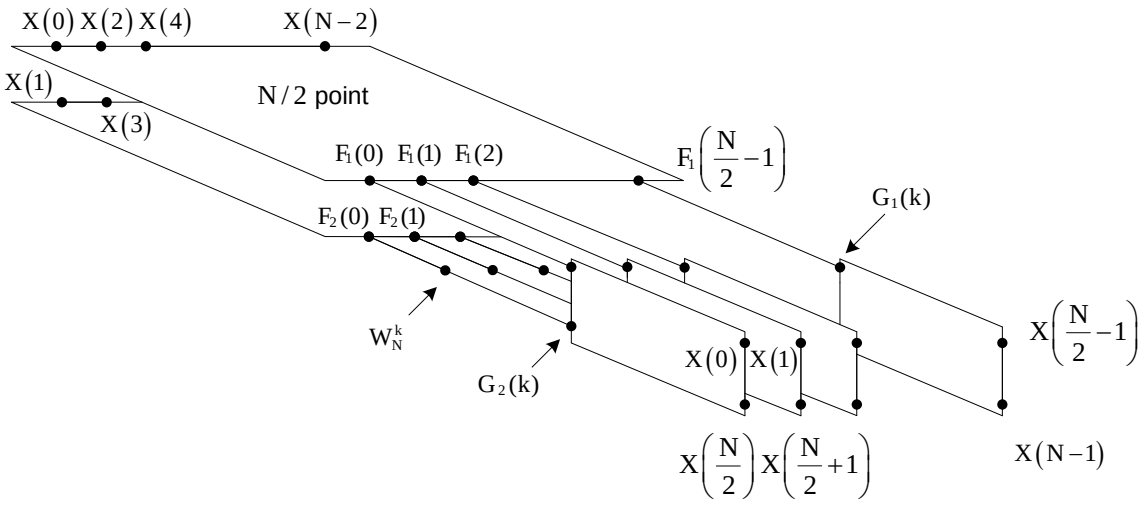

Figure 3.2: Radix-2 decimation in time algorithm

This process of decimation in time is repeated for each sequence of $y(n)$ and $z(n)$. Since the N point is divided into two part, $y(n)$ would results in $N/4$-point sequences

$$v_{11}(n) = y(2n) \quad , \quad n = 0,1,...,\frac{N}{4}-1, \tag{3.30}$$

$$v_{12}(n) = y(2n+1) \quad , \quad n = 0,1,...,\frac{N}{4}-1 \tag{3.31}$$

and $z(n)$ is represented with

$$v_{21}(n) = z(2n) \quad , \quad n = 0,1,...,\frac{N}{4}-1, \tag{3.32}$$

$$v_{22}(n) = z(2n+1) \quad , \quad n = 0,1,...,\frac{N}{4}-1. \tag{3.33}$$

After completion of $N/4$-point DFTs, the next step is to compute $N/2$-point DFTs $F_1(k)$ and $F_2(k)$ from the described equation:

$$F_1(k) = V_{11}(k) + W_{N/2}^{k} V_{12}(k) \quad , \quad k = 0,1,...,\frac{N}{4}-1, \tag{3.34}$$

$$F_1(k+\frac{N}{4}) = V_{11}(k) - W_{N/2}^{k} V_{12}(k) \quad , \quad k = 0,1,...,\frac{N}{4}-1, \tag{3.35}$$

$$F_2(k) = V_{21}(k) + W_{N/2}^{k} V_{22}(k) \quad , \quad k = 0,1,...,\frac{N}{4} - 1 \tag{3.36}$$

and

$$F_2(k + \frac{N}{4}) = V_{21}(k) - W_{N/2}^{k} V_{22}(k) \quad , \quad k = 0,1,...,\frac{N}{4} - 1. \tag{3.37}$$

The $\{V_{ij}(k)\}$ are the $N/4$-point DFTs of sequence $\{v_{ij}(n)\}$. From the equation it is observed that the computation of $\{V_{ij}(k)\}$ requires $4(N/4)^2$ multiplications. Thus, the computation of $F_1(k)$ and $F_2(k)$ can be done with $N^2/4 + N/2$ complex multiplication. To compute the $X(k)$ from $F_1(k)$ and $F_2(k)$, an additional $N/2$ complex multiplication is needed. Subsequently, the total number of multiplication is reduced to $N^2/4 + N$. For a radix-2 computation, this decimation can be performed $v = \log_2 N$ times. Hence, the total number of complex multiplication is reduced to $(N/2)\log_2 N$ and for the complex addition is $N \log_2 N$.

It is efficient and easier to use butterfly architecture in implementing the FFT computation. This butterfly computation can be done with two different approach which are the decimation-in-time (DIT) and decimation-in-frequency (DIF). The difference between these architectures is that, DIT butterfly structure requires a multiplication process before addition is done. However, both algorithms require the same number of complex multiplication and addition operations. Figure 3.3 shows the 8-point decimation-in-time in butterfly representation.