FFT and FIR Filter implementations for the DSL MODEMS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

VLSI and Embedded Systems

By

C. Chandra Mohan

Department of Electronics and Communication Engineering

National Institute Of Technology

Rourkela

2008

FFT and FIR Filter implementations for the DSL MODEMS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

VLSI and Embedded Systems

By

C. Chandra Mohan

Under the Guidance of

Prof. S. K. PATRA

Department of Electronics and Communication Engineering

National Institute Of Technology

Rourkela

2008

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**FFT and FIR implementation for the DSL MODEMS**" submitted by **C.ChandraMohan** in partial fulfillment of the requirements for the award of Master of Technology Degree in **Electronics & communication Engineering** with specialization in "**VLSI & Embedded Systems**" at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. S.K. Patra
Dept. of Electronics & Communication Engg.
National Institute of Technology
Rourkela-769008

Date:

ACKNOWLEDGEMENTS

This project is by far the most significant accomplishment in my life and it would be impossible without people who supported me and believed in me.

I would like to extend my gratitude & my sincere thanks to my honorable, esteemed supervisor Professor **Dr. S. K. Patra,** Department of Electronics and Communication Engineering. He is not only a great lecturer with deep vision but also and most importantly a kind person. His trust and support inspired me in the most important moments of making right decisions and I am glad to work with him.

I want to thank all my teachers **Dr. G.S. Rath Sir**, **Dr. G. Panda Sir, Dr. K.K. Mahapatra Sir and Dr. S.K. Meher Sir** for providing a solid background for my studies and research thereafter. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to take this privilege to express my deep sense of gratitude to my project guide, **Mr. Praveen K.S** for his valuable help and inspiring guidance that facilitated me to carry out this work satisfactorily. I sincerely thank for his exemplary guidance and encouragement. I would like to thank all the team members of VLSI Design at Ikanos Communications for the support in the project.

I would like to thank all those who made my stay in Rourkela an unforgettable and rewarding experience.

Last but not least I would like to thank my parents, who taught me the value of hard work by their own example. They rendered me enormous support during the whole tenure of my stay in NIT Rourkela.

C.CHANDRA MOHAN

CONTENTS

Abstract

Broad band digital communication that operates over a standard copper wires. It requires the DSL modems which splits the transmissions into 2 frequency bands. The lower frequencies for voice and the higher frequencies for digital data (internet) in order to transmit the data to larger distances through a copper cable we need modulation techniques. Generally in this DSL modems modulation used is QAM technique. The output of the QAM is complex data this complex data we cannot transfer directly through a copper cable because the data should be in time domain or otherwise the phase of the data which is in frequency domain can be lost, in copper cable so this data should be converted in time domain by using IDFT technique. As IDFT requires more number of complex multiplications and more number of complex additions in comparison to IFFT so to reduce the additions and multiplications IFFT technique is used. At the receiver side we can retrieve the same data by using FFT technique. In this section the implemented FFT architecture is fully efficient and this architecture will require less area. And before we have to transmit through the copper line we have to do interpolation or decimation by using the Filtering operation. The implemented poly phase architecture for the filtering is fully efficient, symmetrical and it requires less number of multipliers.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS USED

ASIC ---------- Application-Specific Integrated Circuit

DSL ---------- Digital Subscriber Line

VDSL ---------- Very High data Digital Subscriber Line

ADSL ---------- Asymmetric Digital Subscriber Line

DMT ---------- Discrete Multi Tone

RTL ---------- Register Transfer Level

FFT ---------- Fast Fourier Transform

FIR ---------- Finite Impulse Response

DFT ---------- Discrete Fourier Transform

RAM ---------- Random Access Memory

ROM ---------- Read Only Memory

FPGA ---------- Field Programmable Gated Array

DFT ---------- Design for Testability

VCS ---------- Verilog Computing Simulator

ONU ----------Optical Network Units

ISDN ---------- Integrated Services Digital Network

SONET ---------- Synchronous Optical Network

Chapter 1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION:

Digital subscriber line (DSL*)* technology is a modem technology that uses existing twisted-pair telephone lines to transport high-bandwidth data, such as multimedia and video, to service subscribers. The term XDSL covers a number of similar yet competing forms of DSL, including ADSL, SDSL, HDSL, RADSL, and VDSL. XDSL is drawing significant attention from implementers and service providers because it promises to deliver high-bandwidth data rates to dispersed locations with relatively small changes to the existing Telco infrastructure. xDSL services are dedicated, point-to-point, public network access over twisted-pair copper wire on the local loop between a network service provider (NSP's) central office and the customer site, or on local loops created either intra-building or intra-campus. Currently the primary focus in xDSL is the development and deployment of ADSL and VDSL technologies and architectures

1.2 MOTIVATION:

Suppose we need to launch a new product or device in the field of electronics, there is an important need of VLSI implementation which is the main focus in the present industry. The communication functionality also needs specifically VLSI implementation. In this the effective and accurate writing of code is done by EDA tool which gives more about the design and the functionality of the chip. In this EDA tool Coding style is RTL which is either verilog or vhdl. The efficient writing of RTL CODE results in avoiding the possible minor mistakes which makes the design user friendly &flexible and which gives the efficient gate level implementation. As the EDA tools have been automated the implementation methods have become sophisticated in order to make a real time chip design easier and enables us to know the number of gates required for the design. As the transistor sizes have been shrinking and shrinking, a less number of problems are addressed. The result in the shrinking of transistor size lead to major problems like interconnects delays and power leakage. So the design engineer's focus has been shifted from logic design to physical design. This thesis aims to design and write the RTL code for the one of digital signal processing block in DSL modems.

1.3 THESIS CONTRIBUTION:

The front-end (RTL) design flow and the concepts behind each step of the process have been studied. Then I had written the RTL code for the architectures for the FFT and filter. This RTL code synthesized through one of EDA tool compiler. by synthesis we can know the gate count, power consumption and the timing analysis for the architectures of those blocks.

1.4 THESIS OUTLINE:

Following this introduction the remaining part of the thesis is organized as under, Chapter 2 provides the introduction of the DSL technology and various forms of the DSL. Chapter 3 provides the basic operation of the FFT algorithm and the FIR filter. Chapter 4 discusses the design and architecture of the FFT and filter Chapter 5 discusses the simulation report of the design implementation. Chapter 6 summarizes the work undertaken in this thesis and points to possible directions for future work.

Chapter 2

OVER VIEW OF THE DSL TECHNOLOGY

2. OVERVIEW OF DSL

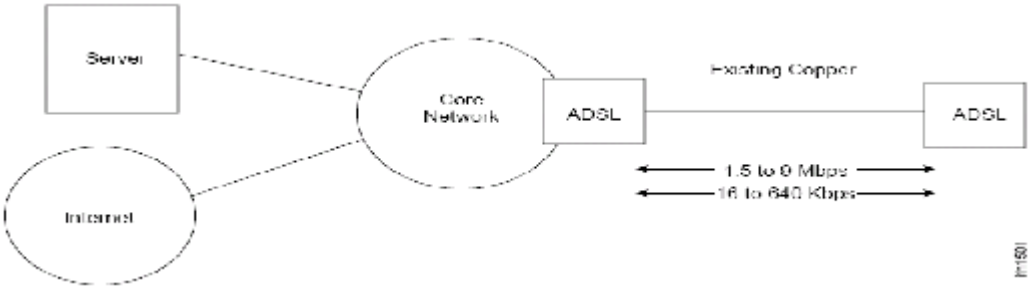
2.1 INTRODUCTION:

Digital subscriber line (DSL) technology is a modem technology that uses existing twisted-pair telephone lines to transport high-bandwidth data, such as multimedia and video, to service subscribers. The term xDSL covers a number of similar yet competing forms of DSL, including ADSL, SDSL, HDSL, RADSL, and VDSL. xDSL is drawing significant attention from implementers and service providers because it promises to deliver high-bandwidth data rates to dispersed locations with relatively small changes to the existing Telco infrastructure. xDSL services are dedicated, point-to-point, public network access over twisted-pair copper wire on the local loop between a network service provider (NSP's) central office and the customer site, or on local loops created either intra-building or intra-campus. Currently the primary focus in xDSL is the development and deployment of ADSL and VDSL technologies and architectures.

2.2 Asymmetric Digital Subscriber Line (ADSL) Technology:

ADSL technology is asymmetric. It allows more bandwidth downstream—from an NSP's central office to the customer site—than upstream from the subscriber to the central office. This asymmetry, combined with always-on access (which eliminates call setup), makes ADSL ideal for Internet/intranet surfing, video-on-demand, and remote LAN access. Users of these applications typically download much more information than they send.

ADSL transmits more than 6 Mbps to a subscriber, and as much as 640 kbps more in both directions (shown in Figure 1). Such rates expand existing access capacity by a factor of 50 or more without new cabling. ADSL can literally transform the existing public information network from one limited to voice, text, and low-resolution graphics to a powerful, ubiquitous system capable of bringing multimedia, including full motion video, to every home this century.



2.1. The Components of ADSL Network including a Telco and a CPE

2.2.1 ADSL Capabilities:

An ADSL circuit connects an ADSL modem on each end of a twisted-pair telephone line, creating three information channels—a high-speed downstream channel, a medium-speed duplex channel, and a basic telephone service channel. The basic telephone service channel is split of from the digital modem by filters, thus guaranteeing uninterrupted basic telephone service, even if ADSL fails. The high-speed channel ranges from 1.5 to 6.1 Mbps, and duplex rates range from 16 to 640 kbps. Each channel can be sub multiplexed to form multiple lower-rate channels.

ADSL modems provide data rates consistent with North American T1 1.544 Mbps and European E1 2.048 Mbps digital hierarchies (see Table 1) and can be purchased with various speed ranges and capabilities. The minimum configuration provides 1.5 or 2.0 Mbps downstream and a 16 kbps duplex channel; others provide rates of 6.1 Mbps and 64 kbps duplex. Products with downstream rates up to 8 Mbps and duplex rates up to 640 kbps are available today ADSL modems accommodate Asynchronous Transfer Mode (ATM) transport with variable rates and compensation for ATM overhead, as well as IP protocols.

Downstream data rates depend on a number of factors, including the length of the copper line, its wire gauge, presence of bridged taps, and cross-coupled interference. Line attenuation increases with line length and frequency and decreases as wire diameter increases. Ignoring bridged taps ADSL performs as shown in Table 1.

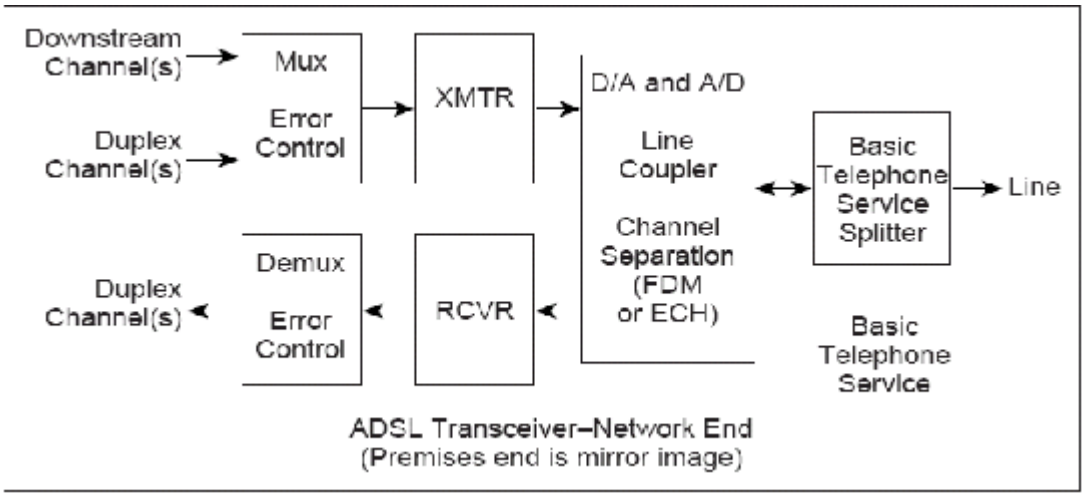
1. Table for the downstream and duplex bearer channels

Downstream Bearer Channels	
n x 1.536 Mbps	1.536 Mbps
	3.072 Mbps
	4.608 Mbps
	6.144 Mbps
n x 2.048 Mbps	2.048 Mbps
	4.096 Mbps
Duplex Bearer Channels	
C Channel	16 Kbps
	64 Kbps
Optional Channels	160 Kbps
	384 Kbps
	544 Kbps
	576 Kbps

Many applications envisioned for ADSL involve digital compressed video. As a real-time signal, digital video cannot use link- or network-level error control procedures commonly found in data communications systems. ADSL modems therefore incorporate forward error correction that dramatically reduces errors caused by impulse noise. Error correction on a symbol-by-symbol basis also reduces errors caused by continuous noise coupled into a line.

2.2.2ADSL Technology:

ADSL depends on advanced digital signal processing and creative algorithms to squeeze so much information through twisted-pair telephone lines. In addition, many advances have been required in transformers, analog filters, and analog/digital (A/D) converters. Long telephone lines may attenuate signals at 1 MHz (the outer edge of the band used by ADSL) by as much as 90 dB, forcing analog sections of ADSL modems to work very hard to realize large dynamic ranges, separate channels, and maintain low noise figures. On the outside, ADSL looks simple—transparent synchronous data pipes at various data rates over ordinary telephone lines. The inside, where all the transistors work, is a miracle of modern technology. Figure **2** displays the ADSL transceiver-network end.



2.2. Overview of the ADSL Transceiver-Network

To create multiple channels, ADSL modems divide the available bandwidth of a telephone line in one of two ways—frequency-division multiplexing (FDM) or echo cancellation. FDM assigns one band for upstream data and another band for downstream data. The downstream path is then divided by time-division multiplexing into one or more high-speed channels and one or more low-speed channels. The upstream path is also multiplexed into corresponding low-speed channels. Echo cancellation assigns the upstream band to overlap the downstream,

and separates the two by means of local echo cancellation, a technique well known in V.32 and V.34 modems. With either technique, ADSL splits off a 4 kHz region for basic telephone service at the DC end of the band.

An ADSL modem organizes the aggregate data stream created by multiplexing downstream channels, duplex channels, and maintenance channels together into blocks, and attaches an error correction code to each block. The receiver then corrects errors that occur during transmission up to the limits implied by the code and the block length. The unit may, at the user's option, also create super blocks by interleaving data within sub blocks; this allows the receiver to correct any combination of errors within a specific span of bits. This in turn allows for effective transmission of both data and video signals.

2.2.3 ADSL standards:

The American National Standards Institute (ANSI) Working Group T1E1.4 recently approved an ADSL standard at rates up to 6.1 Mbps (ANSI Standard T1.413). The European Technical Standards Institute (ETSI) contributed an annex to T1.413 to reflect European requirements. T1.413 currently embodies a single terminal interface at the premises end. Issue II, now under study by T1E1.4, will expand the standard to include a multiplexed interface at the premises end, protocols for configuration and network management, and other improvements.

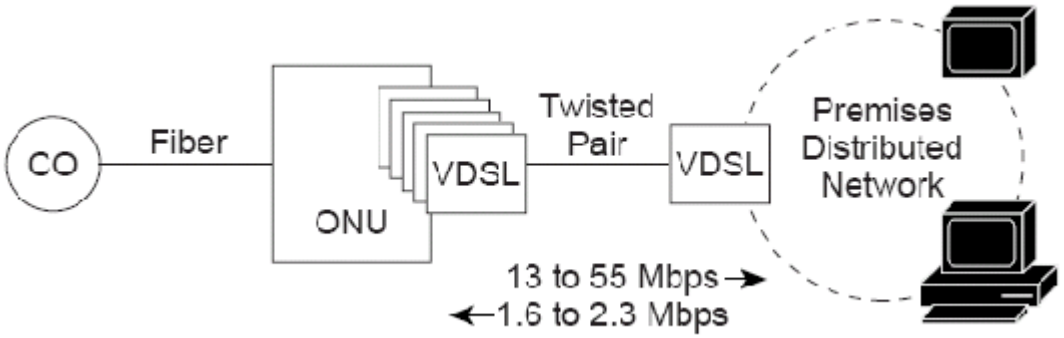
2.3 Very high data rate Digital Subscriber Line (VDSL):

It is becoming increasingly clear that telephone companies around the world are making decisions to include existing twisted-pair loops in their next-generation broadband access networks. Hybrid fiber coax (HFC), a shared-access medium well suited to analog and digital broadcast, comes up somewhat short when used to carry voice telephony, interactive video, and high-speed data communications at the same time. Fiber all the way to the home (FTTH) is still prohibitively expensive in a marketplace soon to be driven by competition rather than cost. An attractive alternative, soon to be commercially practical, is a combination of fiber cables feeding neighborhood optical network units (ONUs) and last-leg-premises connections by existing or new copper. This topology, which is often called fiber to the neighborhood (FTTN), encompasses fiber to the curb (FTTC) with short drops and fiber to the basement (FTTB), serving tall buildings with vertical drops.

One of the enabling technologies for FTTN is VDSL. In simple terms, VDSL transmits high-speed data over short reaches of twisted-pair copper telephone lines, with a range of speeds

depending on actual line length. The maximum downstream rate under consideration is between 51 and 55 Mbps over lines up to 1000 feet (300 m) in length. Downstream speeds as low as 13 Mbps over lengths beyond 4000 feet (1500 m) are also common. Upstream rates in early models will be asymmetric, just like ADSL, at speeds from 1.6 to 2.3 Mbps. Both data channels will be separated in frequency from bands used for basic telephone service and Integrated Services Digital Network (ISDN), enabling service providers to overlay VDSL on existing services. At present the two high-speed channels are also separated in frequency. As needs arise for higher-speed upstream channels or symmetric rates, VDSL systems may need to use echo cancellation.

Figure 2.3: The overview of devices in VDSL Network



2.3.1 VDSL capabilities:

Although VDSL has not achieved ADSL's degree of definition, it has advanced far enough that we can discuss realizable goals, beginning with data rate and range. Downstream rates derive from submultiples of the SONET (Synchronous Optical Network) and SDH (Synchronous Digital Hierarchy) canonical speed of 155.52 Mbps, namely 51.84 Mbps, 25.92 Mbps, and 12.96 Mbps. Each rate has a corresponding target range

2.4 DSL (DMT) Transceiver Model

A DMT transmitted bits undergo a lot of transformation as highlighted in the figure.

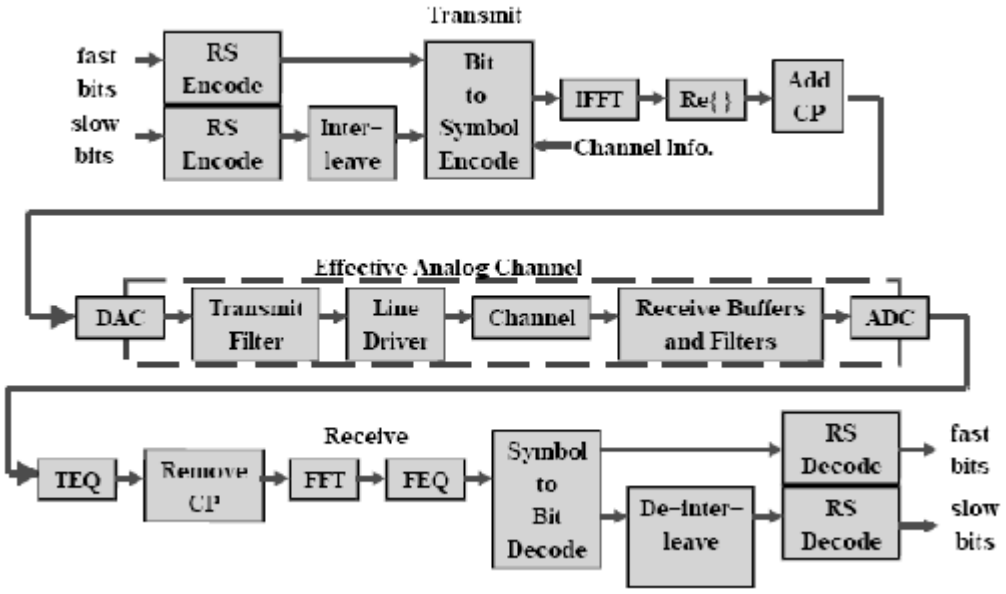


Figure 2.4 DMT (DSL) Transceiver Model

In a DSL connection each modem (CO & CPE side) both transmits & receive at the same time, hence the two end-to-end connection will co-exist on the line one for transmission from the home to the central office & the other from central office to the home. In general, the up-streams & down-streams transmission are frequency division multiplexed, but some spectral overlap is allowed for modems in echo cancellation mode.

Initialization Process: DSL modems are channel adaptive. The initialization process requires 6 seconds of real time & consists of four phases. The first phase is activation & acknowledgement, during which the modems perform initial synchronization. The second phase, transceiver training allows each modem to train its channel equalization & if necessary its echo canceller. The third phase is channel analysis, during which the modems measure channel SNR & exchange rate options. The final phase, exchange phase is used to determine loop attenuation & performance margin information & to confirm parameters.

Fast & Slow bits & interleaving: ADSL was targeted for the video-on-demand market. Video transmission created the need for a data interleaver coupled with Reed-Solomon forward error correction(FEC) to make the system more robust to impulse noise. With the

interleaver, however, the latency of the transmitted data is approximately 17ms which can hurt TCP/IP performance. Thus current ADSL modems support both video & Internet access through two parallel framers, each implementing FEC. Because of its latency, the interleaved framer is referred to as the "slow" buffer, & the non-interleaved framer is called the "fast" buffer.

Time-Domain Equalization(TEQ) & the Cyclic Prefix(CP): there are two forms of equalization that occur in a DSL modem transceiver. After the received analog signal has been digitized, it is first modified by a TEQ followed by FEQ(frequency domain equalizer), which is applied after the FFT. The purpose of TEQ is to "shorten" the effective channel length such that it falls within the cyclic prefix length. It also prevents inter-symbol interference.

Reed-Solomon(RS) encode & Interleave: The bit stream is split into "fast" & "slow" bits & Reed-Solomon encoded. The slow bits are interleaved for improved robustness to burst noise at the cost of increased latency.

Bit to Symbol Encode: Bits are divided up among frequency bins by capacity & mapped to appropriate QAM constellations. The bit capacity & QAM size of each frequency bin is based upon channel SNR & determined during modem training.

IFFT & Re{}: The complex N-Length frequency vector (made up of N QAM constellation) is IFFT'd to produce a 2N-length time domain signal. The real part of the output of the IFFT is passed on to the next block.

Add Cyclic Prefix(CP): The cyclic prefix is added to the transmit vector such that when the vector is convolved with the channel, the convolution is cyclic. This is necessary for frequency equalization in the receiver.

Transmit Filter & Line Driver: These blocks constitute the analog front end(AFE). The AFE shapes the output such that it will not interfere with the received signal, & such that it meets the power spectral density(PSD) requirements of the specifications.

Remove Cyclic Prefix: The CP is no longer necessary prior to the FFT.

FFT: Transform the time domain data vector into a vector of QAM symbols.

Frequency Equalization (FEQ): Applies an inverted frequency-domain approximation of the channel response to the received data vector. The CP & TEQ validate the assumption of cyclic convolution with the channel.

Symbol to Bit Encode: Maps QAM symbols back to bits.

2.5 Block Diagram of DSL CO side:

General block diagram of DSL Modem chip at the Central Office side is as shown. It uses the Discrete Multi Tone(DMT) technique. The DMT line code sends multiple tones of data over the line allocating more data to the lower frequencies where there are less analog impairments.

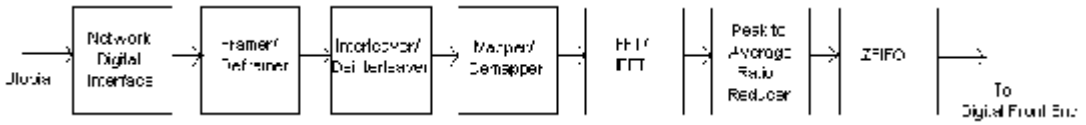


Figure : Block Diagram of DSL Modem Chip at CO side.

The Network and Digital Interface block implements network interface functionality such as ATM. ATM(Asynchronous Transfer Mode) is a cell relay (cell relay refers to a method of statistically multiplexing fixed length packets i.e. cells to transport data between computers or kinds of network equipment. It is unreliable connection oriented packet switched data communication protocols.) , packet switching network (packets are formatted block of data) and data link layer protocol which encodes data traffic into small fixed size(53 bytes : 48 bytes of data & 5 bytes of header information) cells. ATM is a connection oriented technology in which a logical connection is established between the two end points before the actual data exchange. This accepts input data from the network side through ATM interface, and stores the data in separate fifos for each channel.

Framer reads the data from the input fifo periodically at the symbol rate, and constructs the frame. If enough data is not available, it inserts idle cells. There are two paths available for data from ATM, fast and interleaved. Interleaved data processed after the fast data. For each fast and interleaved path, CRC (Cyclic Redundancy Check for burst error correction), scrambling and FEC encoding tasks are performed.

Scrambler uses following algorithm to scramble separate output from both the fast and interleaved buffers.

$$dn' = dn \text{ XOR } dn\text{-}18' \text{ XOR } dn\text{-}23'$$

where dn is the n-th output from the fast or interleaved buffer (scrambler input), and dn' is the n-th output from the corresponding scrambler. Both the scrambler and descrambler are self-synchronizing. FEC Encoder corrects occasional errors in the data.

Interleaver gets the data from the Framer and performs the interleaving. Interleaving is used in digital data transmission technology to protect the transmission against burst errors. These errors overwrite a lot of bits in a row, so a typical error correction scheme that expects errors to be more uniformly distributed can be overwhelmed. Interleaving is used to help stop this from happening

Mapper performs tone ordering, constellation mapping, gain scaling. Subsequently, constellation mapping (with the option of trellis encoding for ADSL) in performed in the constellation/trellis encoder block. Its outputs are complex numbers, which are gain scaled to make uniform the distribution of the values entering the Fourier Transform Engine (FTE or FFT/IFFT) block. The RX and TX paths share the FTE block. An IDFT operation is performed to convert frequency domain tones into a time domain signal to which a cyclic prefix is added before sending it to the Digital Front End.

Along the RX path, the incoming sample stream is processed by a time-domain equalizer (TEQ) to normalize the impact of the channel response. A DFT operation performed by the FTE block converts the time-domain signal into the frequency domain. A frequency equalizer (FEQ) is used to compensate for the carrier-specific channel distortion (phase and amplitude).

The PAR(Peak to Average Ratio Reducer Block) module takes the IFFT output, and based on the packet header, either performs the PAR operation or bypasses it. The PAR engine output is then sent to the Transmit Window (filtering) and Cyclic Extension module through the ZFIFO block. The PAR module has a kernel memory, which contains a time-domain signal consisting of tones which are not carrying data. The PAR module scans the IFFT output for peaks which are greater than the specified threshold value, and chooses the highest peak. It then scales the kernel, aligns the kernel to the peak, and subtracts the kernel from the IFFT output to reduce the peak. This process of removing a peak (consisting of detecting highest peak, scaling kernel, and subtracting kernel) is called an iteration. The PAR module can be configured to do desired number of iterations. The ZFIFO blocks saves the output of PAR & then data is sent to the AFE (Analog Front End – DSL modem chip at the Customer Premises Equipment Side) through Digital front End.

Chapter 3

OVERVIEW OF FFT AND FIR FUNCTIONS

3.1 INTRODUCTION OF FFT ALGORITHM:

3.1.1 DFT:

In view of the importance of the DFT in various digital signal processing applications, such as linear filtering, correlation analysis, and spectrum analysis, its efficient computation is a topic that has received considerable attention by many mathematicians, engineers, and applied scientists

Basically, the computational problem for the DFT is to compute the sequence $\{X(k)\}$ of N complex-valued numbers given another sequence of data $\{x(n)\}$ of length N, according to the formula

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad 0 \le k \le N-1$$

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

In general, the data sequence $x(n)$ is also assumed to be complex valued. Similarly, The IDFT becomes

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(k) W_N^{nk}, \qquad 0 \le n \le N-1$$

Since DFT and IDFT involve basically the same type of computations, our discussion of efficient computational algorithms for the DFT applies as well to the efficient computation of the IDFT We observe that for each value of k, direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and N-1 complex additions ($4N$-2 real additions). Consequently, to compute all N values of the DFT requires N^2 complex multiplications and N^2-N complex additions. Direct computation of the DFT is basically inefficient primarily because it does not exploit the symmetry and periodicity properties of the phase factor W_N. In particular, these two properties are

$$\text{Symmetry property}: W_N^{k+N/2} = -W_N^k$$

$$\text{Periodicity property}: W_N^{k+N} = W_N^k$$

3.1.2 FFT:

A fast Fourier transform (FFT) is an efficient algorithm used to calculate the Discrete Fourier Transform (DFT) and it's inverse. FFT's are widely used in application areas like

Digital Signal Processing, solving Partial Differential Equations and multiplication of large integers and complex numbers. Generally it is known as Cooley-Tukey FFT algorithm, and these algorithms are based on divide and conquer approach. In this approach N-point DFT is successfully decomposed into smaller DFTs. Because of this decomposition the number of computations is reduced. The other types of FFT algorithms are

- Prime-factor FFT algorithm
- Bruun's FFT algorithm
- Rader's FFT algorithm
- Bluestein's FFT algorithm

3.1.3 Computational efficient algorithms of FFT:

3.1.3.1 Radix-2 FFT:

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and conquer approach. We split the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$,corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1), \qquad n = 0,1,\ldots,\frac{N}{2}-1$$

Thus $f_1(n)$ and $f_2(n)$ are obtained by decimating x(n) by a factor of 2, and hence the resulting FFT algorithm is called decimation-in-time algorithm.

Now the N-point DFT can be expressed in terms of the DFT's of the decimated sequences as follows

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0,1,\ldots,N-1$$
$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn}$$
$$= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)}$$

But $W_N^2 = W_{N/2}$. With this substitution the equation can be expressed as

$$X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$
$$= F_1(k) + W_N^k F_2(k), \qquad k = 0,1,\ldots,N-1$$

Where $F_1(K)$ and $F_2(K)$ are the N/2-point DFTs of the sequences of f1(m) and $f_2(m)$, respectively

Since $F_1(K)$ and $F_2(K)$ are periodic, with period N/2, we have $F_1(K+N/2)= F_1(K)$ and $F_2(K+N/2)= F_2(K)$ in addition the factor $W_N^{k+N/2}=-W_N^K$. Hence it can expressed as

$$X(k) = F_1(k) + W_N^k F_2(k), \qquad k = 0,1,\ldots,\frac{N}{2}-1$$

$$X(k+\frac{N}{2}) = F_1(k) - W_N^k F_2(k), \qquad k = 0,1,\ldots,\frac{N}{2}-1$$

We observe that the direct computation of $F_1(k)$ requires $(N/2)^2$ complex multiplications. The same applies to the computation $F_2(k)$. Further more, there are N/2 additional complex multiplications required to compute $W_N^k F_2(k)$. Hence the computation of X(k) requires $2(N/2)^2+N/2=N^2/2+N/2$ complex multiplications. This first step results in a reduction of the number of multiplications from N^2 to $N^2/2+N/2$, which is about a factor of 2 for N large

By computing N/4-point DFTs, we would obtain the N/2-point DFTs $F_1(k)$ and $F_2(k)$ from the relations

$$F_1(k) = F\{f_1(2n)\} + W_{N/2}^k F\{f_1(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1, \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$$F_1\left(k+\frac{N}{4}\right) = F\{f_1(2n)\} - W_{N/2}^k F\{f_1(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

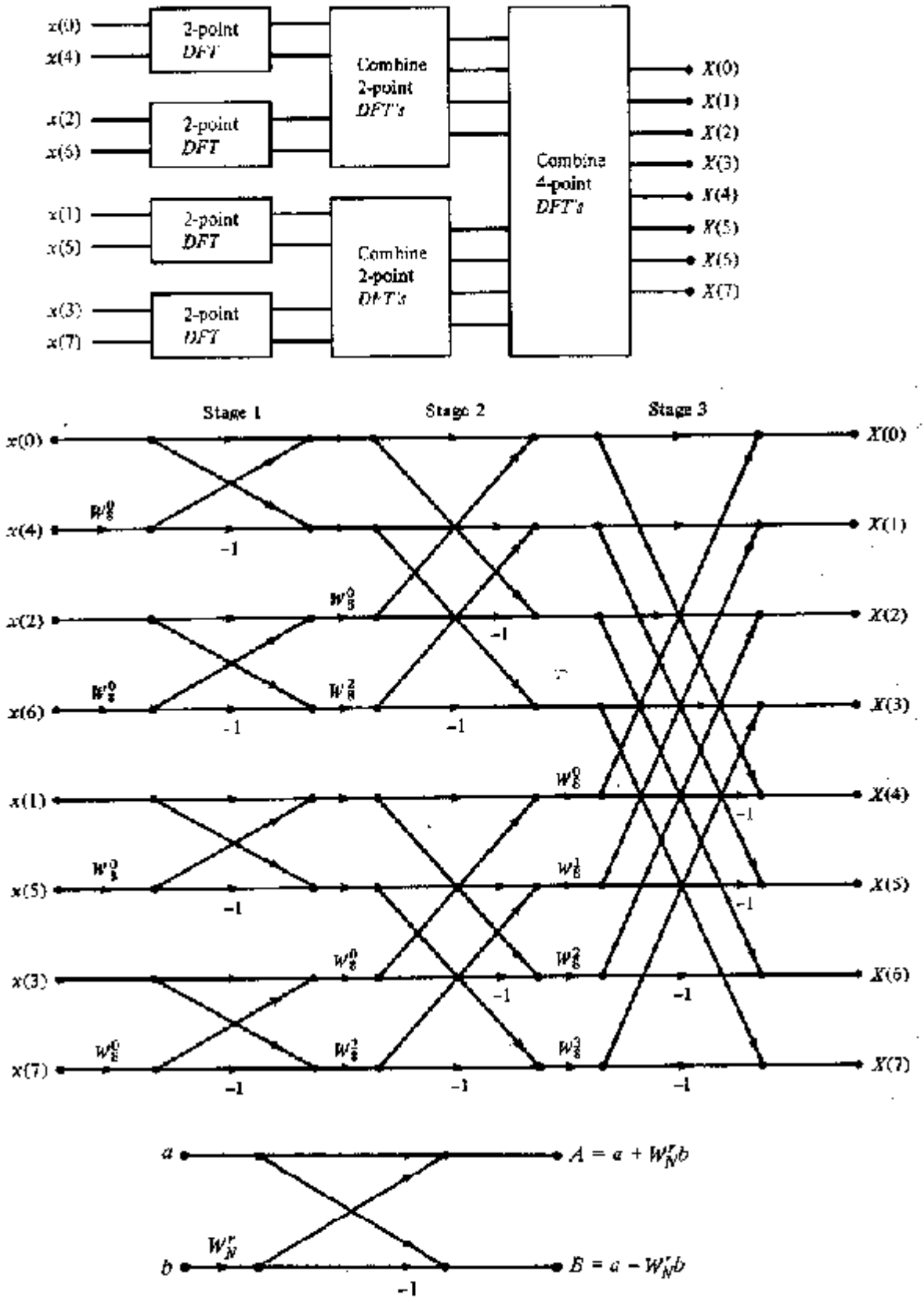
$$F_2(k) = F\{f_2(2n)\} + W_{N/2}^k F\{f_2(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$$F_2\left(k+\frac{N}{4}\right) = F\{f_2(2n)\} - W_{N/2}^k F\{f_2(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1, \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$F\{*\}$ represents Fourier transform

The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to one-point sequences. For $N=2^v$, this decimation can be performed $v=\log_2 N$ times. Thus the total number of complex multiplications is reduced to $(N/2)\log_2 N$. The number of complex additions is $N\log_2 N$.

For illustrative purposes, Figure1 depicts the computation of N=8 point DFT. We observe that the computation is done in 3 stages, begins with the four 2-point DFTs, then two 4-point DFTs, and finally one 8-point DFT. The combination for the smaller DFTs to form the larger DFT is illustrated as follows for N=8.



3.1 Block Diagram And Butterfly Diagram of 8-point FFT

Here the important observation is input is bit reversed sequence and the output is normal sequence. Another radix-2 FFT algorithm called decimation in frequency it is also similar to decimation in time except here the input is normal sequence and output is bit reversed sequence.

3.1.3.2Radix-4 FFT:

When the number of data points N in the DFT is a power of 4 (N=4v), we can, of course, always use a radix-2 algorithm for the computation. However, for this case, it is more efficient computationally to employ a radix-r FFT algorithm. Let us begin by describing radix-4 decimation in time FFT algorithm briefly. We split or decimate the N-point input sequence into four sub sequences, x(4n), x(4n+1), x(4n+2), x(4n+3), n=0,1 ….N/4-1.

$$X(p,q) = \sum_{l=0}^{3}\left[W_N^{lq} F(l,q)\right] W_4^{lp}$$

$$F(l,q) = \sum_{m=0}^{(N/4)-1} x(l,m)\, W_{N/4}^{mq}$$

$$p = 0,1,2,3; \quad l = 0,1,2,3; \quad q = 0,1,2,\ldots,\frac{N}{4}-1$$

and

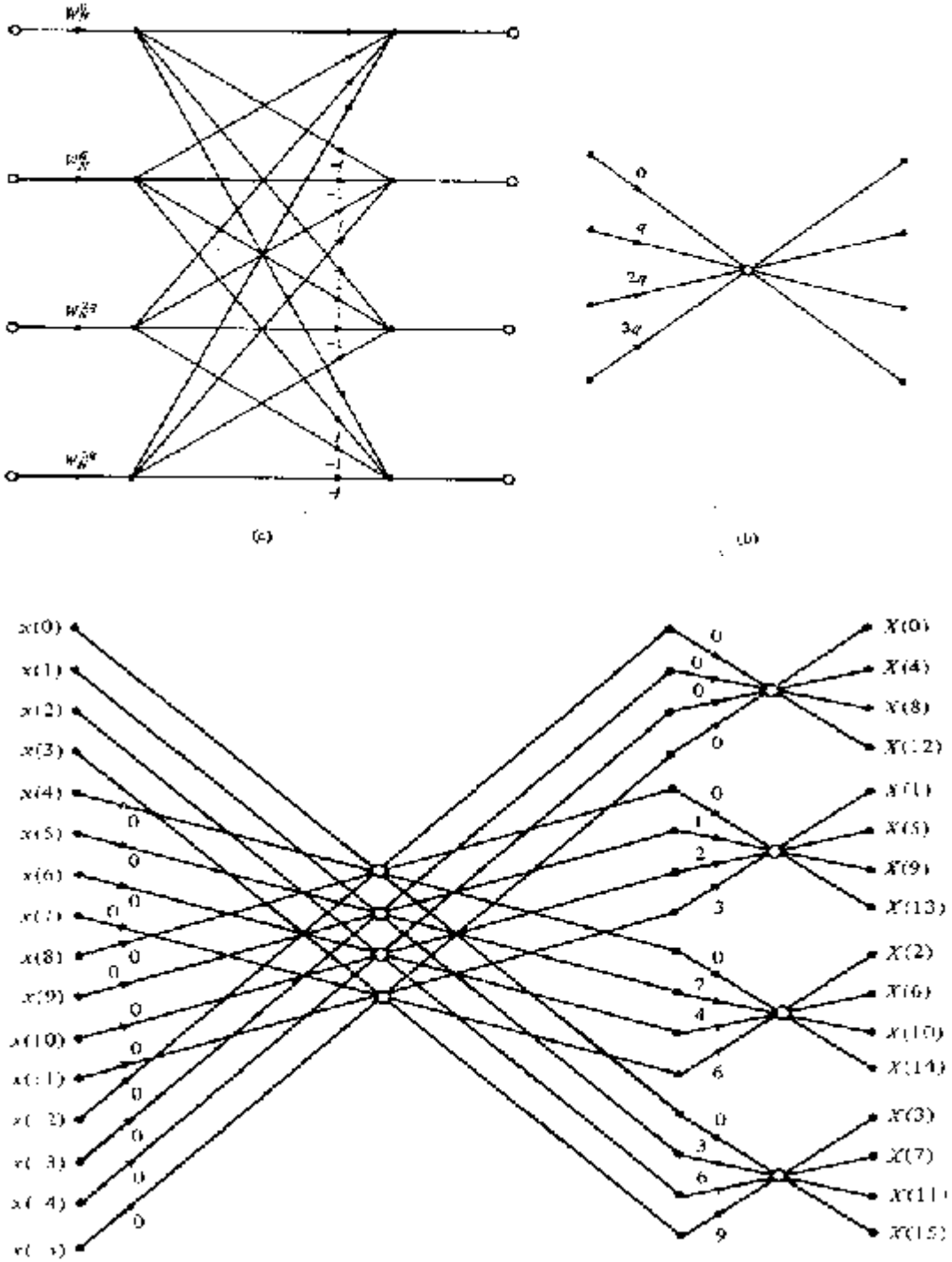
$$x(l,m) = x(4m+l)$$

$$X(p,q) = X\left(\frac{N}{4}p + q\right)$$

Thus the four N/4-point DFTs F(l,q) obtained from the above equitation are combined to yield the N-point DFT. The expression for combining the N/4-point DFTs defines a radix-4 decimation-in-time butterfly, which can be expressed in matrix form as

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix}$$

The radix-4 butterfly is depicted in figure below note that each butterfly involves 3 complex multiplications, since W_N^0=1, and 12 complex additions.



3.2. 16-point FFT using radix-4 algorithm

Radix-4 Decimation in frequency is also decimation in time with same computational efficiency.

For illustrative purposes, let us re-drive the radix-4 decimation-in-frequency algorithm by breaking the N-point DFT formula into four smaller DFTs. We have

$$\begin{aligned}X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{N/4-1} x(n) W_N^{kn} + \sum_{n=N/4}^{N/2-1} x(n) W_N^{kn} + \sum_{n=N/2}^{3N/4-1} x(n) W_N^{kn} + \sum_{n=3N/4}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{N/4-1} x(n) W_N^{kn} + W_N^{Nk/4} \sum_{n=0}^{N/4-1} x\left(n+\frac{N}{4}\right) W_N^{kn} + \\ &\quad W_N^{Nk/2} \sum_{n=0}^{N/4-1} x\left(n+\frac{N}{2}\right) W_N^{kn} + W_N^{3Nk/4} \sum_{n=0}^{N/4-1} x\left(n+\frac{3N}{4}\right) W_N^{kn}\end{aligned}$$

From the definition of the twiddle factors, we have

$$W_N^{kN/4} = (-j)^k, \qquad W_N^{kN/2} = (-1)^k, \qquad W_N^{3kN/4} = (j)^k$$

Thus

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

The relation is not an N/4-point DFT because the twiddle factor depends on N and not on N/4. To convert it into an N/4-point DFT we subdivide the DFT sequence into 4 N/4-point sub sequences, X(4k),X(4k+1),X(1k+2) and X(4k+3),k=0,1,..,N/4. Thus we obtain the radix-4 DIF DFT as

$$\begin{aligned}X(4k) &= \sum_{n=0}^{N/4-1} \left[x(n) + x\left(n+\frac{N}{4}\right) + x\left(n-\frac{N}{2}\right) + x\left(n+\frac{3N}{4}\right) \right] W_N^0 W_{N/4}^{kn} \\ X(4k+1) &= \sum_{n=0}^{N/4-1} \left[x(n) - jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + jx\left(n+\frac{3N}{4}\right) \right] W_N^n W_{N/4}^{kn} \\ X(4k+2) &= \sum_{n=0}^{N/4-1} \left[x(n) - x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) - x\left(n+\frac{3N}{4}\right) \right] W_N^{2n} W_{N/4}^{kn} \\ X(4k+3) &= \sum_{n=0}^{N/4-1} \left[x(n) + jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - jx\left(n+\frac{3N}{4}\right) \right] W_N^{3n} W_{N/4}^{kn}\end{aligned}$$

Where we have used the property $W_N^{4kn}=W_{N/4}^{kn}$. Note that the input to each N/4-point DFT is a linear combination of four signal samples scaled by a twiddle factor. This procedure is repeated v times where $v=\log_4 N$.

3.1 Real multiplications:

N	Radix-2	Radix-4
16	24	20
32	88	-
64	264	208
128	72	-
256	1800	1392
512	4360	-
1024	10248	7856

3.2 Real additions:

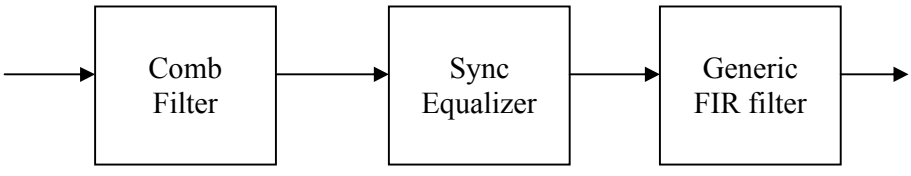
N	Radix-2	Radix-4
16	152	148
32	408	-
64	1032	976
128	2054	-
256	5896	5488
512	13566	-
1024	30728	28336

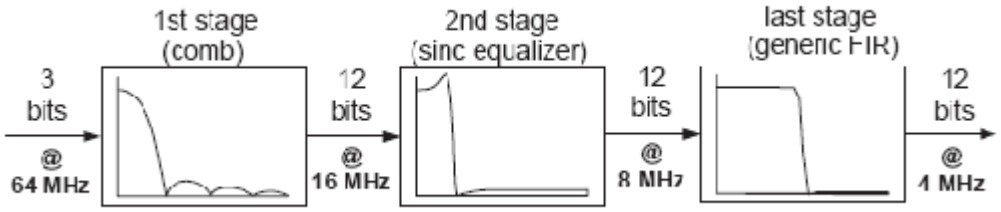
3.2 INTRODUCTION OF FIR FILTER:

The main function of the FIR filter design in DSL modems is for interpolating and decimation operations at Transmitter and Receiver of modems. In general the filtering is done here in three stages.

1. Comb filter
2. Sinc equalizer filter
3. generic filter

Block Diagram:



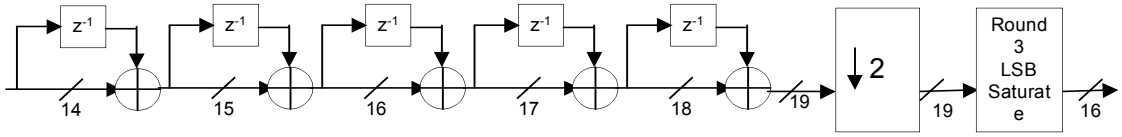


3.3 Block Diagram of 3-stage filtering in DSL modems

Comb filter stage:

The first stage of the filter is usually implemented with a simple comb filter. it is simple design and it must be operated at a very high sampling rate. The response of the filter shows in above figure.

Block diagram:



3.4. 5-Taps Comb Filter

Sync Equalizer stage:

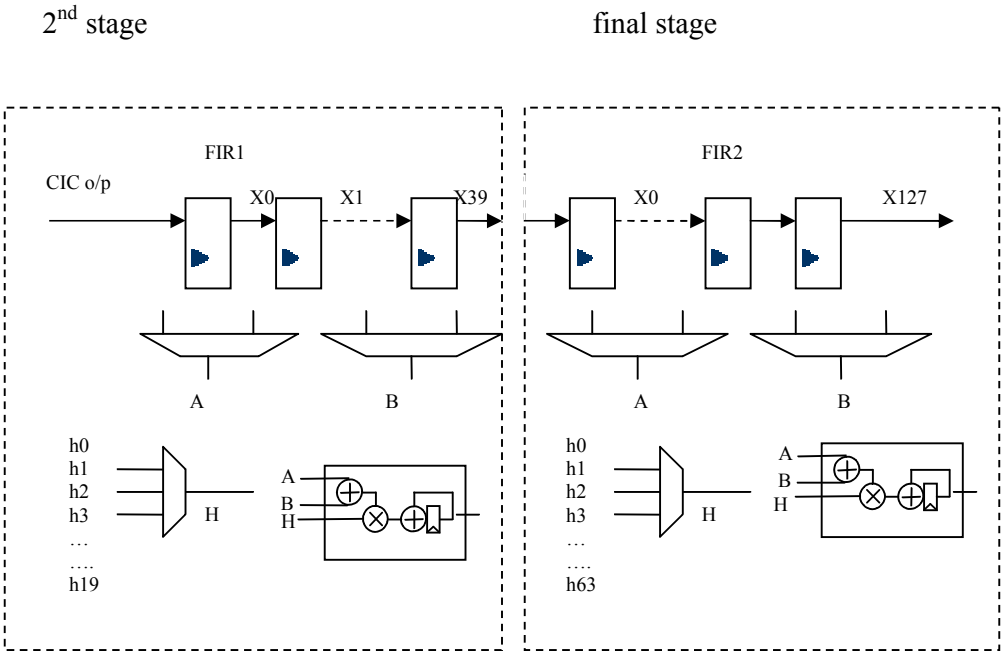
The second stage of the filter is sync equalizer here the main function is to compensate the in-band attenuation introduced by the comb stage. The response of this filter stage shown the above diagram.

Generic FIR stage:

The last stage of the filter is generic FIR filter here the response of the filter is steep transition. It is used to provide required stop band attenuation. The response of this stage filtering is shown in above diagram.

The designing architectures of 2nd and final stage of the filters are same

Block Diagram:



3.5 Sync and Generic stages of Filter

This high level deign of FIR filters can be done with MATLAB and its signal processing Toolbox which provides the Remez function to calculate the coefficients of the filter. This MATLAB provides filter coefficients with infinite precision, the next design step is the truncation of the coefficients to fixed number of bits. Then the filter can be realized easily.

Chapter 4

DESIGN OF FFT AND FIR

4.1 Literature survey:

1. THE HYBRID ARCHITECTURE PARALLEL FAST FOURIER TRANSFORM (HAPFFT)

The HAPFFT is an improved formulation for building Parallel FFT custom hardware modules. It provides improved performance, efficient resource utilization, and reduced design time. The HAPFFT is modular in nature. It includes a custom front-end parallel processing unit, which produces intermediate results. The intermediate results are sent to multiple, independent FFT modules. These independent modules form the back-end of the HAPFFT, and are generic, meaning that any pre-existing FFT architecture may be used. With *P* back-end modules a speedup of *P* will be achieved, in comparison to an FFT module composed solely of a single module. Furthermore, the HAPFFT defines the front-end processing unit as a function of *P*. It hides the high communication costs typically seen in Parallel FFTs. Reductions in control complexity, memory demands, and logical resources, are achieved.

An extraordinary result of the HAPFFT formulation is a sub linear area-time growth. This phenomenon is often also called super linear speedup. Sub linear area-time growth and super linear speedup are equivalent terms.

A further benefit resulting from the HAPFFT formulation is reduced design time. Because the HAPFFT defines only the front-end module, and because the back-end parallel modules may be composed of any preexisting FFT modules, total design time for a HAPFFT is greatly reduced.

The radix-4 DIF algorithm is similar to the radix-2 DIF algorithm. The derivation uses the same approach, by decimating the DFT output in the frequency domain. They differ in that the atomic computational unit is a radix-4 butterfly. The advantage of using the radix-4 butterfly is that it can be computed without any twiddle factor multiplications, while the total number of butterflies required is half that of the radix-2 algorithm. Thus, the total number of complex twiddle factor multiplications for a radix-4 FFT is half that of the radix-2 FFT. The disadvantage is that twice the number of complex additions is needed, and the size of the input data set is limited to a power of four lengths. Nevertheless, when using fixed-point computer arithmetic, because complex multiplications are very often more expensive than complex additions, a radix-4 FFT may be cheaper to implement. [1]

2. MULTIPLIER-LESS BASED PARALLEL-PIPELINED FFT ARCHITECTURES FOR WIRELESS COMMUNICATION APPLICATIONS [2]

The FFT processor is widely used in DSP and communication applications. It is a critical block in OFDM based communication systems, such as WLAN (IEEE 802.11) and MC-CDMA receiver. Recently, both high data processing and high power efficiency assumes more and more importance in wireless systems. Due to the nature of non-stop processing at the sample rate, the pipelined FFT appears to be the leading architecture for high performance applications. However, only one processor element (PE) in each column makes a bottleneck on the throughput of pipelined FFTs. For example, for N-point radix-4 FFT, the radix-4 PE, consisting of a radix-4 butterfly element and a complex multiplier, has to calculate N/4 times to complete all computation in one stage.

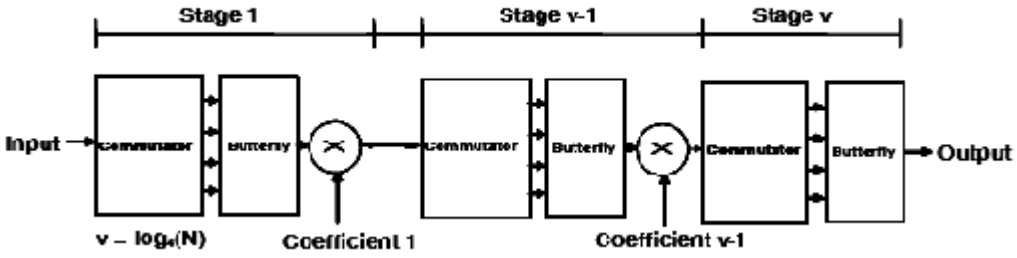


Figure 4.1 N-Points RADIX – 4 Pipelined FFT Processor Architecture [2]

The DFT of N complex data points x(n) is defined bv

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

---------- Eq-(2.2)

Where $W_N = e{-}j2\pi/N$. W_N is twiddle factor or coefficient.

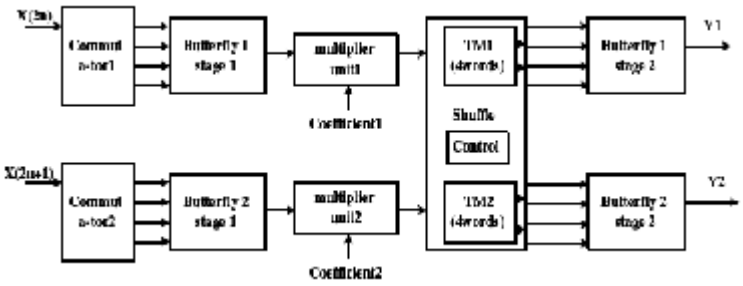


Figure 4.2 16-Points 2-Parallel-Pipelined FFT Architecture [2]

We explore to allocate two, Radix-4 PEs in each stage of the 16-point pipelined FFT. It can achieve double throughput, compared to the pipelined FFT at the same operation frequency. The input data are separated into two streams. Two commutators in stage1, each has half storage units of the commutator in pipelined FFTs, transform odd and even sequential input data to parallel data respectively. The coefficients are divided into two responding sections, in terms of even and odd. Two coefficient sections are fed into two complex multipliers, respectively. Due to the separate processing on odd and even data, a shuffle unit is needed in stage 2 to implement the inter stage data shuffle. The shuffle unit is composed of two triple port SRAMs, each sized 4 words, and an addressing control unit. The 4 outputs from two triple port SRAMs are fed into each simplified butterfly unit. Each simplified butterfly unit only yields the half of all outputs. The coefficients are partitioned into 4 sections, each having 4 coefficients. Three of these sections are fed into 3 complex multiplier units respectively.

4.2 Design of FFT:

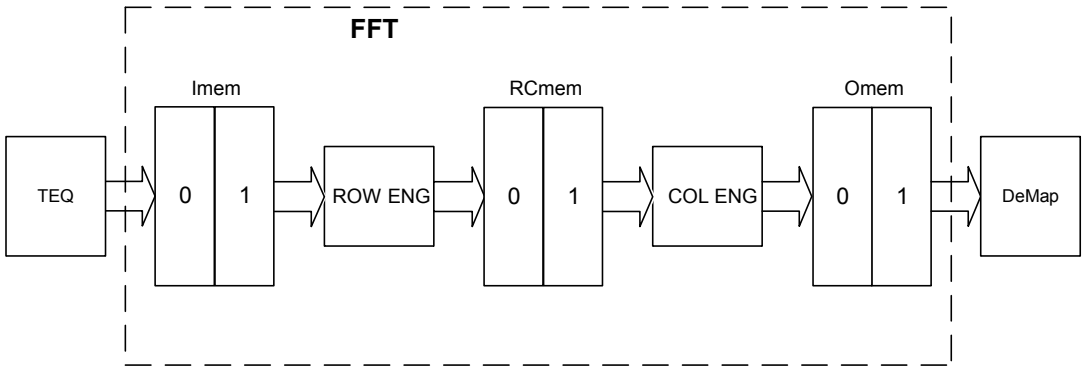
The FFT engine of DSL modem is designed to support fast Fourier transform computation at line speeds with fixed latency. This FFT engine works for multi channels and high speed data processing.

This FFT engine has following enhancements

It supports full 4k tone function with 4KHZ spacing or 8KHZ spacing mode

Optimize throughput of channel –base processing.

This FFT unit supports the transform configurations of 4k tones, 2k tones, 256 tones and 128 tones. It consists of input memory(Imem), row engine, intermediate data memory(RCmem), column engine, and output memory(Omem) it interfaces the TEQ and Demapper block as shown in figure.



4.3 Block Diagram of FFT

For DFT computation, a real sequence of size 2N is used to compose a complex sequence of length N on which the DFT is performed and finally a post-processing step is used extract the DFT is performed and finally a post-processing step is used to extract the DFT of 2N sequence.

Here assume FFT inputs 2N real sequences x(n), n=0,1,2,……,2N-1.

Let $x_1(n) = x(2n)$ and $x_2(n) = x(2n+1)$

$x_c(k) = x_1(k)+jx_2(k)$ applying DFT we will get $x_c(k) = x_1(k)+jx_2(k)$

Hence, $X_1(k) = [X_c(k) +X_c*(N-k)]/2$ and $X_2(k) = -j[X_c(k) – X_c*(N-k)]/2$ and finally,

$X(k) = X_1(k) + W_{2N}^{k} X_2(k)$, k=0,1,2,…,N-1

$X(k+N) = X_1(k) – W_{2N}^{k} X_2(k)$, k=0,1,2,…,N-1

For k=0, $X(0) = [Rel\{X_c(0)\} + Img\{X_c(0)\}] + j0$.

For k=N/2, $X(N/2) = Rel\{X_c(N/2)\} – jImg\{X_c(N/2)\}$.

In the above block diagram shown first FFT decodes the flags field when receiving a packet from TEQ, and then performs FFT operation according to the commands. When finishes the commands, the FFT copies the packet header to Demapper.

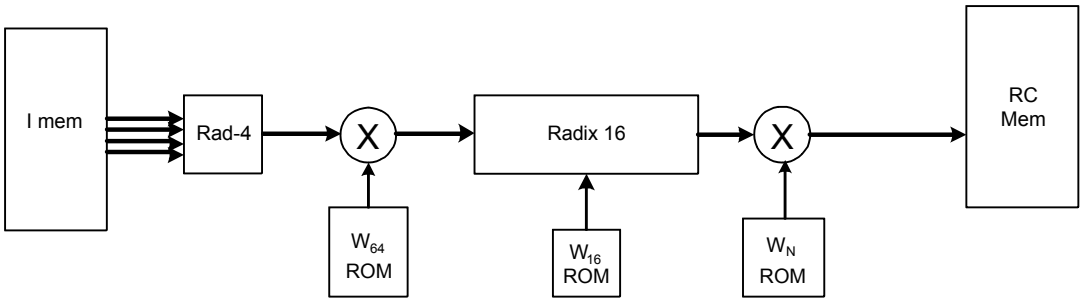
Function:

The FFT engine decomposes the size N complex DFT computation into a 2-D map of R rows and C columns (where N = R*C), then performs a post-processing after DFT computation. In order to obtain the desired performance, the FFT is pipelined at block level and unit level. Which is broken into row and column pipelines as shown in Figure 1? The RCmem ping-pong memory (0/1) supports row DFT and column DFT block level pipelining, while the Imem and Omem ping-pong memories (0/1) supports DFT unit level pipelining.

Row engine:

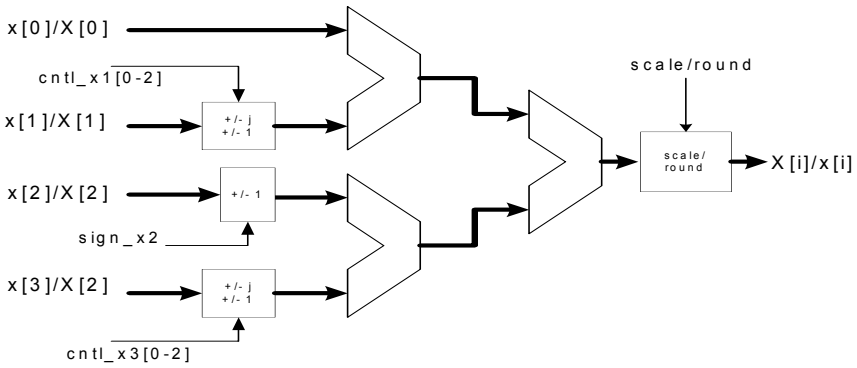
This row engine is designed to process up to 64 columns at each row. As a result each row can be transformed by 4x16 decomposition. The figure shown in down is block diagram of row engine, which consists of Imem, radix-4 block, W_{64} ROM, radix-16 block, W_{16} ROM, W_N ROM, complex multipliers and RCmem. The radix-4 block produces one output per cycle, which is fed to the radix-16 block after multiplying W_{64} ROM twiddle factors. A complex multiplier is required at the output of the radix-16 block for multiplication with the

W_N row column twiddle factors where N is the size of the DFT being processed. The DFT computation requires 3 multipliers in row engine.

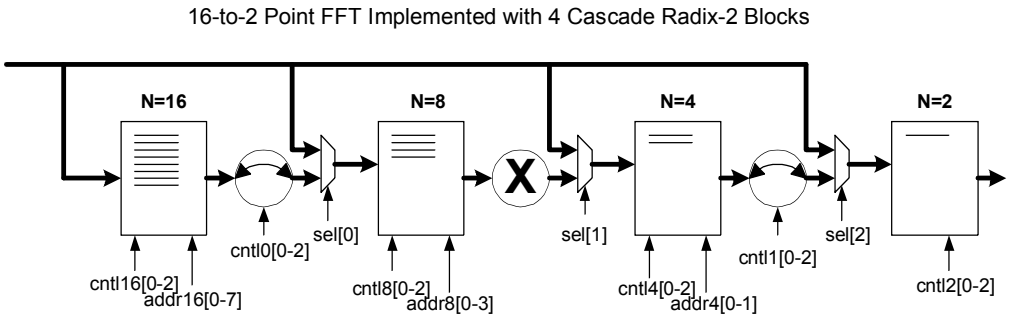


4.4. Row Engine diagram

The implementation of radix-4 and radix-16(using radix-2) block diagrams shown in below

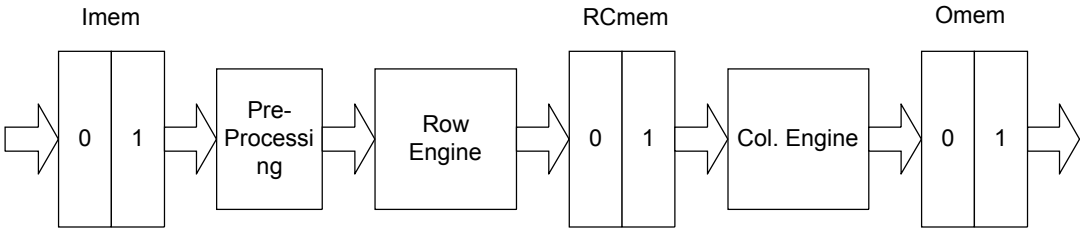


4.5 Radix-4 block diagram



Column engine of FFT block diagram is also similar to the row engine. And the similar architecture is used fir IFFT also except the input is giving to the IFFT block is complex form and the output of IFFT is real generally this IFFT block is used in the TX end of the DSL

modem and the FFT is used in the RX end of the DSL modem. The IFFT block diagram implemented as shown in below



4.6 Block Diagram of IFFT

4.1 Decomposition table for the FFT computation:

DFT Size = 2N	N = R * C Decomposition	Row Decomposition	Column Decomposition	Complex Multipliers
256	16 x 16	4 x 4 (16 rows)	4 x 4 (16 columns)	5
512	32 x 16	4 x 4 (32 rows)	4 x 8 (16 columns)	5
4096	64 x 32	4 x 8 (64 rows)	4 x 16 (32 columns)	7
8192	64 x 64	4 x 16 (64 rows)	4 x 16 (64 columns)	7

4.3 Design of FIR filter:

Most currently used digital filter designs fall into one of two groups: parallel architectures implemented on an ASIC, and serial architectures implemented on a generic DSP. The rapid progress in ASIC technology has made parallel architectures wasteful for all but the very high-speed designs. Most parallel ASIC designs are also quite inflexible. On the other hand, Asics allow great control over the implementation details, including data precision and arithmetic algorithms. This control is needed to design low power systems. In general, DSPs are not suitable for low power design. They are, by nature, large and complex devices with fixed (and often large) bus widths. The arithmetic operations are performed using fast but power hungry blocks. Few DSPs have sufficient on-chip memory resources to implement a large filter and thus require power draining board-level memory accesses. Moreover, most

DSPs cannot compute both the filter output and memory addressing for a non-trivial addressing scheme in a single cycle. Additional cycles require faster clock rates and burn yet more power. For all of these reasons, only an ASIC based solution will be considered in this design.

FIR filter architectures fall into one of two general groups: namely, fully parallel and word-serial. A fully parallel implementation dedicates a MAC and a register for every filter tap and runs at the data rate. A word-serial implementation shares a single MAC for all the taps and runs at N times the data rate. A parallel implementation is large but fast and usually power efficient. A word-serial implementation takes up very little area but is usually slower and consumes more power. The major problem with most realizations of a word-serial architecture is the need to shift data and coefficients to the MAC, requiring a total of N 2 data transfers per output and consuming a lot of power. An alternative approach is to store both the data and the coefficients in RAM and ROM {, respectively, and simulate the shifting of data using an efficient memory addressing scheme [5]. However, most of the reported RAM based implementations do not take advantage of the coefficient symmetry [5]. One of the reasons for this oversight is the requirement of a rather complex addressing scheme.

An FIR filter or order N is defined by a set of coefficients {Ck} has a transfer function, H, given in below Equations. If the coefficients are symmetric about the middle, H(z) has linear phase. Linear phase is very desirable in a large number of applications. This property can be exploited to reduce the number of multiplications per input sample by a factor of two by combining the symmetric coefficients (3). If N is odd, the middle coefficient does not have a symmetric counterpart, and this special case must be taken into account in last Equation.

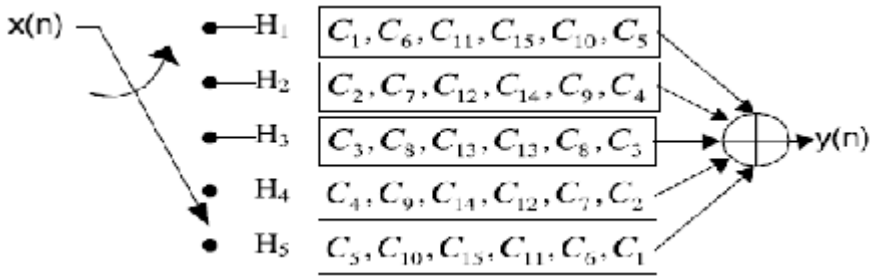
$$H(z) = \sum_{k=0}^{N-1} C_k z^{-k}$$

$$C_k = C_{N-k-1} \quad \text{where } 0 \le k \le \lfloor \frac{N}{2} \rfloor$$

$$H(z) = \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor - 1} C_k \left(z^{-k} + z^{N-k-1} \right) + \left\{ C_{\lfloor \frac{N}{2} \rfloor} z^{\lfloor \frac{-N}{2} \rfloor} \right.$$

An important feature of multi rate processing (decimation) is the poly phase representation, since it leads to computationally efficient implementations of decimation filters [7]. Poly phase decomposition splits the original N coefficients into D groups of M. N=D taps each, where D is the decimation ratio.

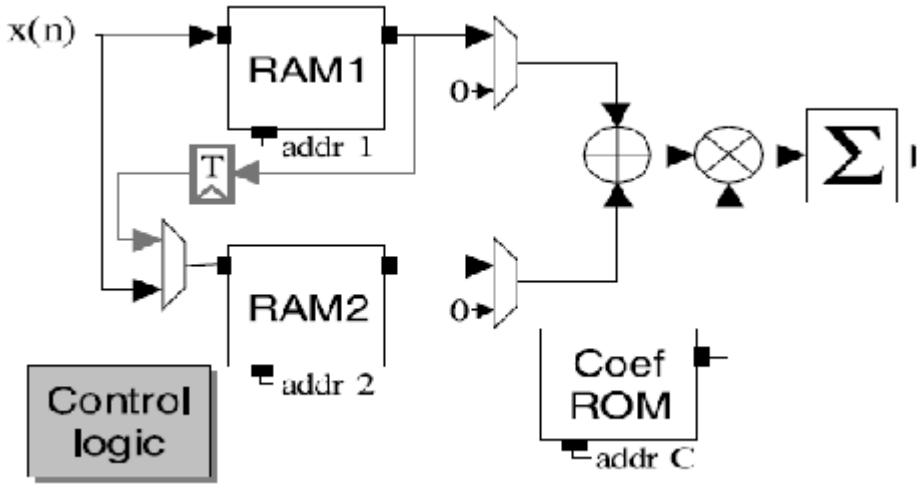
H = {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}, C_{17}, C_{18}, C_{19}, C_{20}, C_{21}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}, C_{27}, C_{28}, C_{29}, C_{30}}



4.7 A Poly Phase 30-Tap decimate by-5 filter

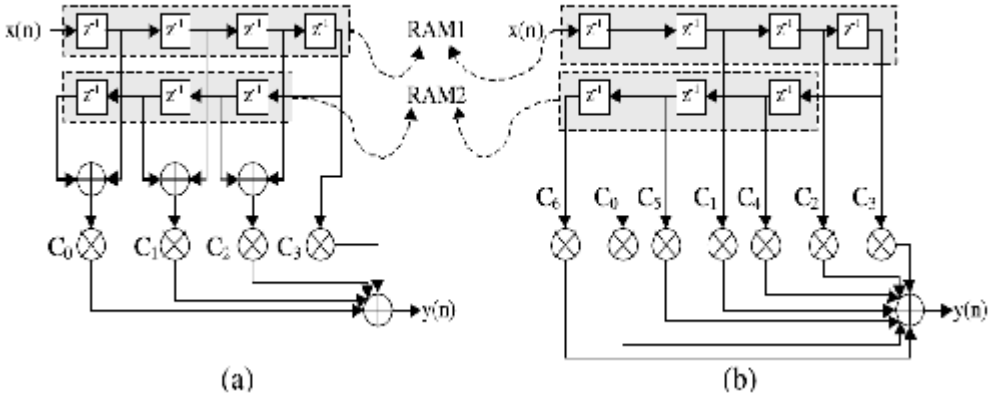
4.3.1 Architecture of the FIR:

The basic advantage of coefficient symmetry is that two input samples can be processed in one cycle using just one MAC. Since two new values are needed for each cycle, two RAMs (or one 2-read/1-write RAM) must be used. The control logic generates appropriate addresses for the two RAMs and the coefficient ROM and writes the input data to the appropriate RAM for every new input sample. The contents of the accumulator is dumped and reset for every new output sample is shown in below figure. The system operates at M/2 times the input data rate since the MAC has to process M filter taps for every input sample (factor of two savings are a result of coefficient symmetry). The memory addressing scheme implemented in the control logic is the key to this architecture and will be discussed in the next three sections. Although a number of different addressing schemes can be developed, they are usually expensive to implement in hardware and are frequently inefficient. An effective addressing scheme must map easily onto simple hardware. It must not waste any cycles just on memory access, insuring that read and write operations occur in the order needed for the computational unit. It must be flexible, but strive to use the same hardware in all modes of operation.



4.8 Block diagram of the FIR

In the above design control logic for regular FIR is for both linear phase (LP) and nonlinear phase (NLP) cases must be supported. As well as both even and odd filter lengths. Here the start by observing that the memory addresses and write signals must be generated such that newly arriving sample replaces the oldest sample already in memory. Further, the addresses to the two RAMs and to the coefficient ROM must be synchronized such that outputs of each correspond to the coefficients. If the filter is LP, the coefficient symmetry can be exploited by folding the delay line array and if it is in NLP is also folded delay line array only but bottom and top row of array computed separately is shown in figure below



4.9 Delay Line Array for LP and NLP FIR

Chapter 5

SIMULATION AND SYNTHESIS OF FFT AND FIR

6. SIMULATION AND SYNTHESIS REPORT

5.1 SIMULATION REPORT OF FFT:

Verilog HDL was chosen as the hardware description language with which to build the FFT. The choice was made based mostly on the ready availability of tools to compile and simulate Verilog HDL designs.

The 256(16X16)-point Radix – 4 FFT is verified using the Verilog HDL. The waveforms are verified using test bench. For the 16-inputs we are getting 16 real output values and 16 imaginary output values. The results are verified for the two different sequences. They are

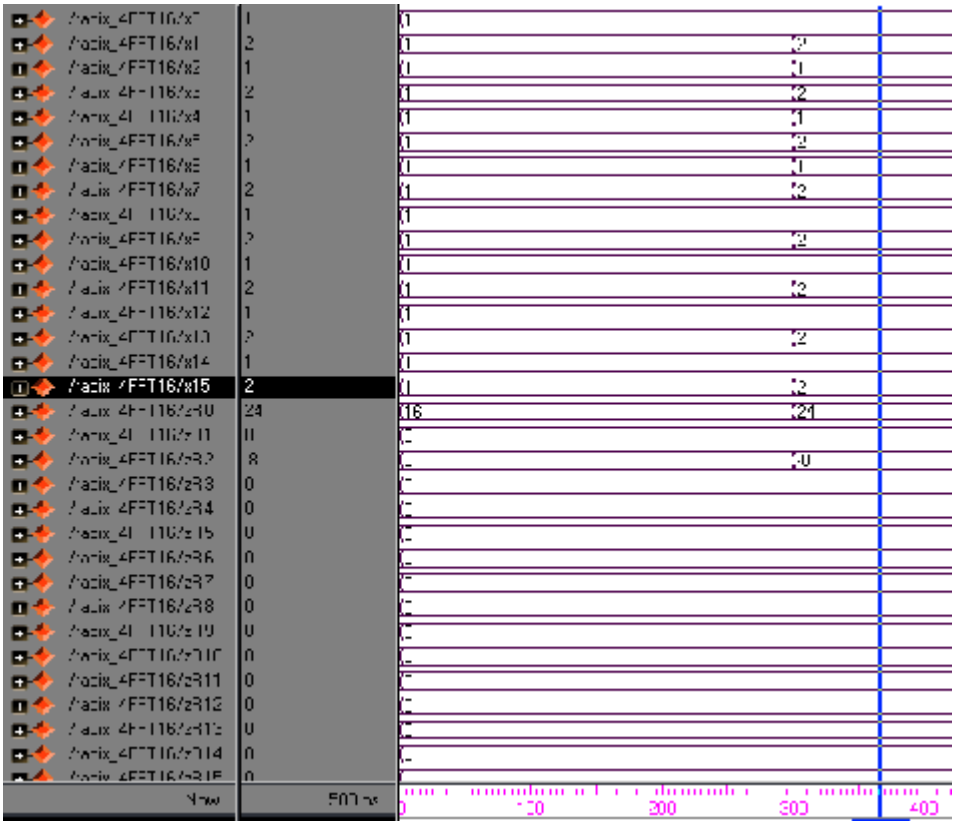
A = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

B = [1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2]

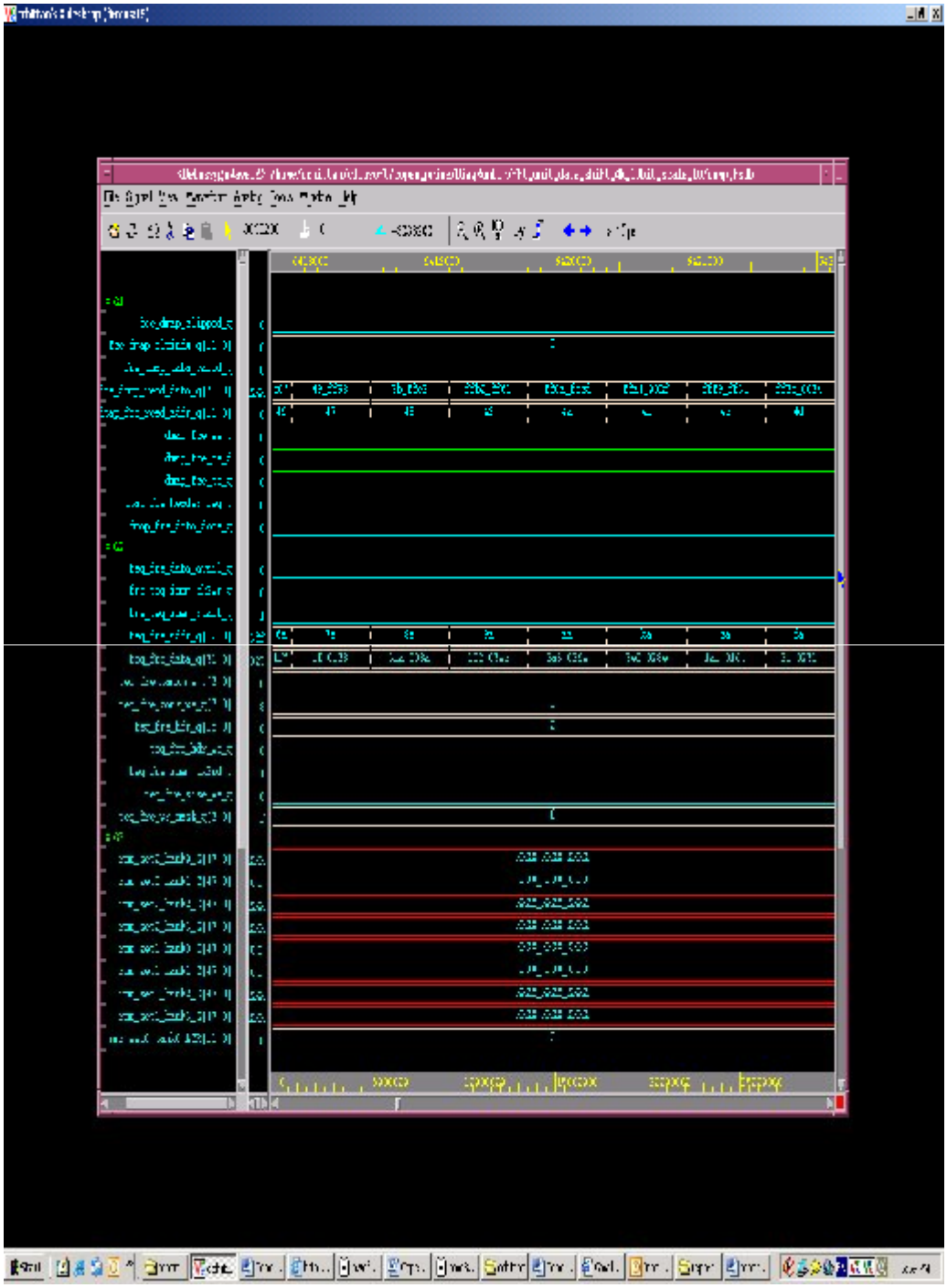
The corresponding outputs for the above sequences are

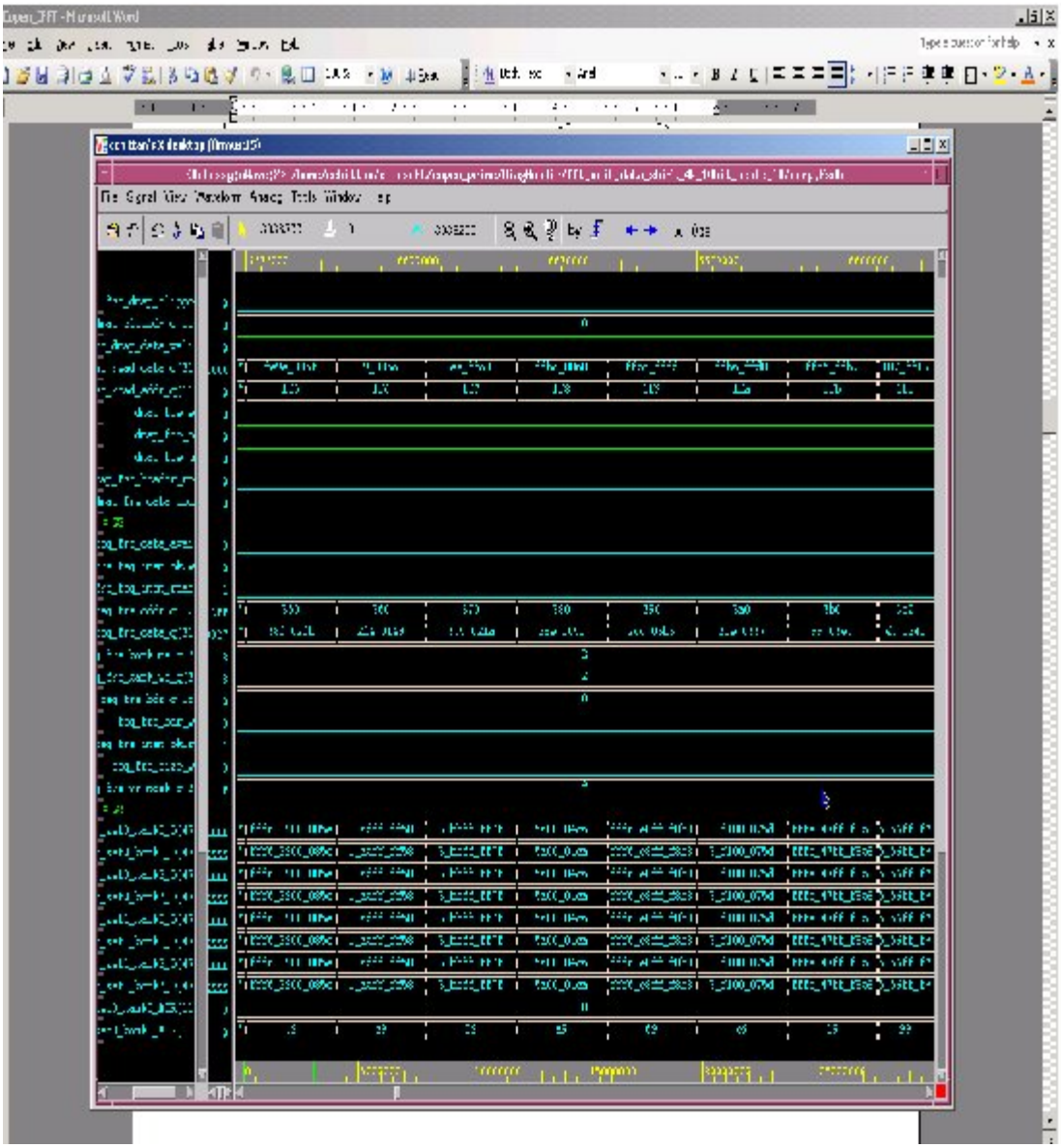
A’ = [16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

B’ = [24, 0, 0, 0, -8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]



5.1 Output of the FFT processor





5.2 Output waveforms for the input sequences

The outputs are crosschecked using MATLAB tool. The output of the VCS (SYNOPSYS) tool and MATLAB are one and the same. The same is shown in the above figure.

5.1.2 Synthesis Analysis:

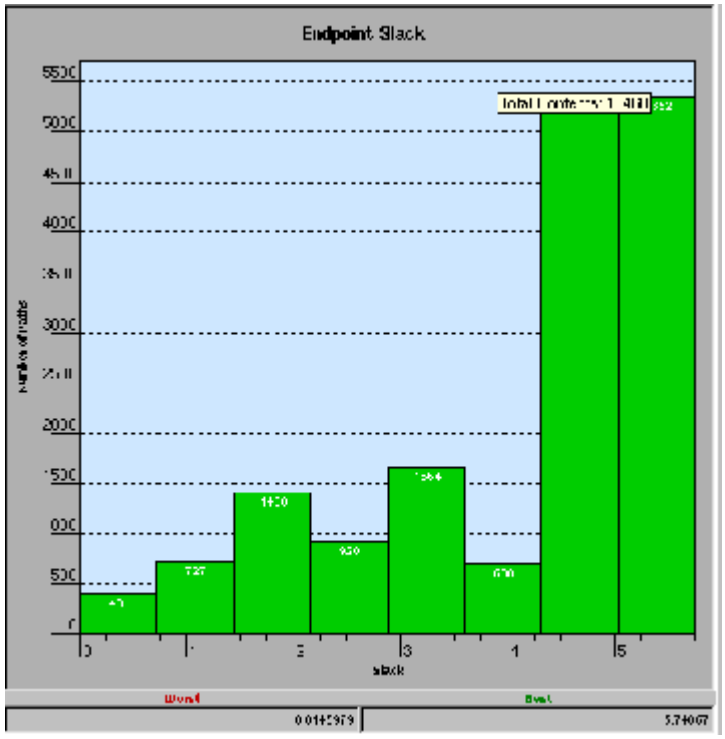
DC-Compiler is used to synthesize the RTL code by giving proper constraints to get optimized net list. The Timing, Area and Power report from the DC-Compiler has been extracted and presented below.

Report of area, timing, power:

The 256-point FFT is synthesized and optimized using DC-Compiler. The final timing, area and power reports are found to be

Power = 26.41 microwatt (from PRIME TIME)

This power calculation of the FFT design is found to be 26.41 mw. It uses VCD (Value Dump Change) file to generate power report. This report is more accurate when compared to DC-Compiler generated power report. Below Histogram shows that there is no violation of slack in the design. Number of paths in the design related to slack. After 4 units its best slack.



5.3 Histogram of FFT

Slack (MET) = 0.01 (Clock 6) (from PRIME TIME)

Fault coverage 97.2% (from DC-Compiler)

Total area:

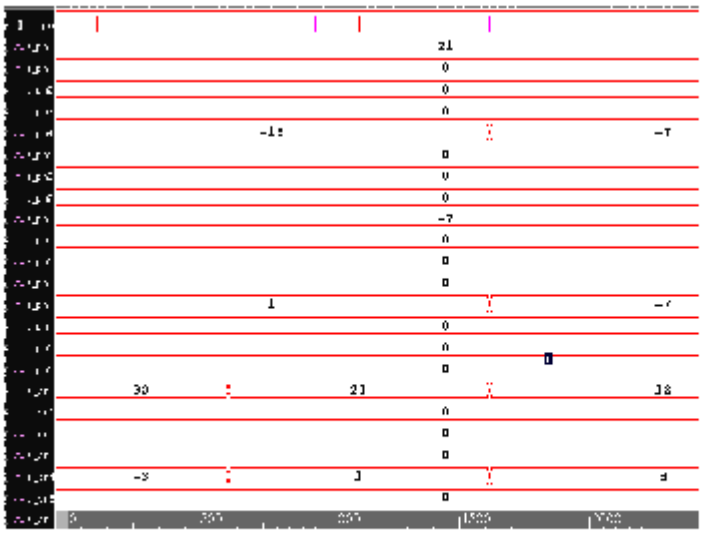
	Row engine	Column engine	Total area
Logic	100kgates	150kgates	250kgates
Memory (RAM)	Imem+RCmem:44kb	Omem+delaybuffe:40kb	84KB
Twiddle ROM	4kX36	4kX36	2X4KX36

The design is optimized; net list is obtained, and verified for the functionality in VCS successfully. The more area is due to more number of input/outputs. With out pads the area is found to be around 3 lakhs. Implementation of SIPO in FFT could have reduced the number of inputs there by reducing the area (due to pads) but initial delay will be more. And here 43k cycles required per four channel 4k-tone @ 8kHz spacing.

5.1.3 Post-Synthesis simulation:

Pre-synthesis simulation ensures the functionality of the design is proper and is verified using VCS. After generating the net list from DC-Compiler, the net list itself is simulated to ensure the proper functionality using VCS again. In normal mode operation test_se will be disabled (test_se=0)

The results obtained from the net list is almost matches the pre-synthesis simulation. The functionality is checked by using proper test vectors in VCS. The number of bits is used to represent the output value is 24-bits, which increases the precision of the output. The 256-point FFT Verilog code is optimized and results are shown in the below waveform.



5.4 Figure Post-Synthesis Simulation of FFT

5.2 Comparison of ASIC Tool Vs FPGA Tool:

ASIC tools consider wire load model, net list and for worst environmental condition to give timing, area and power reports. It is more accurate than FPGA synthesis tools. Optimization and DFT is not possible in FPGA tool. For more number of inputs and outputs mapping is difficult and may not be possible to dump on FPGA board. If we compare the results obtained from both the tools ASIC tool provides the clear picture than FPGA tool.

Advantages:

The combination of nearly cell-based density, speed, and power consumption, coupled with low nonrecurring engineering costs, short turnaround time, and compatibility, along with existing low-cost design tools, have made structured ASICs the logical choice for applications not demanding bleeding-edge performance.

Developing ASICs requires careful design and placement of I/O cells to support the latest complex I/O standards with good signal integrity on all pins. ASIC testing and fault coverage are important parts of the ASIC development process. Testing involves the desired design functionality and the design of the ASIC, and uses boundary scan insertion, built-in-self-test (BIST), signature analysis, Iddq, and automatic test pattern generation (ATPG) techniques.

Disadvantages:

High initial cost

Increase in area due to DFT

Design time increases

Performance degradation caused by DFT circuit

Script files are ordered collection of commands or constraints given to the design. The high complexity of modern design tools, together with an increasingly complicated design system directory structure, require customers to frequently document design efforts by creating scripts and configuration files. Script and configuration files define the ASIC net list and parameters in an executable log file that references file paths in the applicable ASIC design system. The net lists are technology dependent; that is, they represent the ASIC in terms of a specific technology system associated with an ASIC vendor. These scripts and files often make multiple references to various subdirectories and file paths within the library directory structure. [3]

5.3 SIMULATION REPORT OF FIR:

Verilog HDL was chosen as the hardware description language with which to build the FIR. The choice was made based mostly on the ready availability of tools to compile and simulate Verilog HDL designs.

The 30-Tap Symmetric Poly Phase FIR filter is verified using the Verilog HDL. The waveforms (output and input) are verified using test bench. For the inputs we are getting real output values and. The results are verified for the sequences. They are

EX:

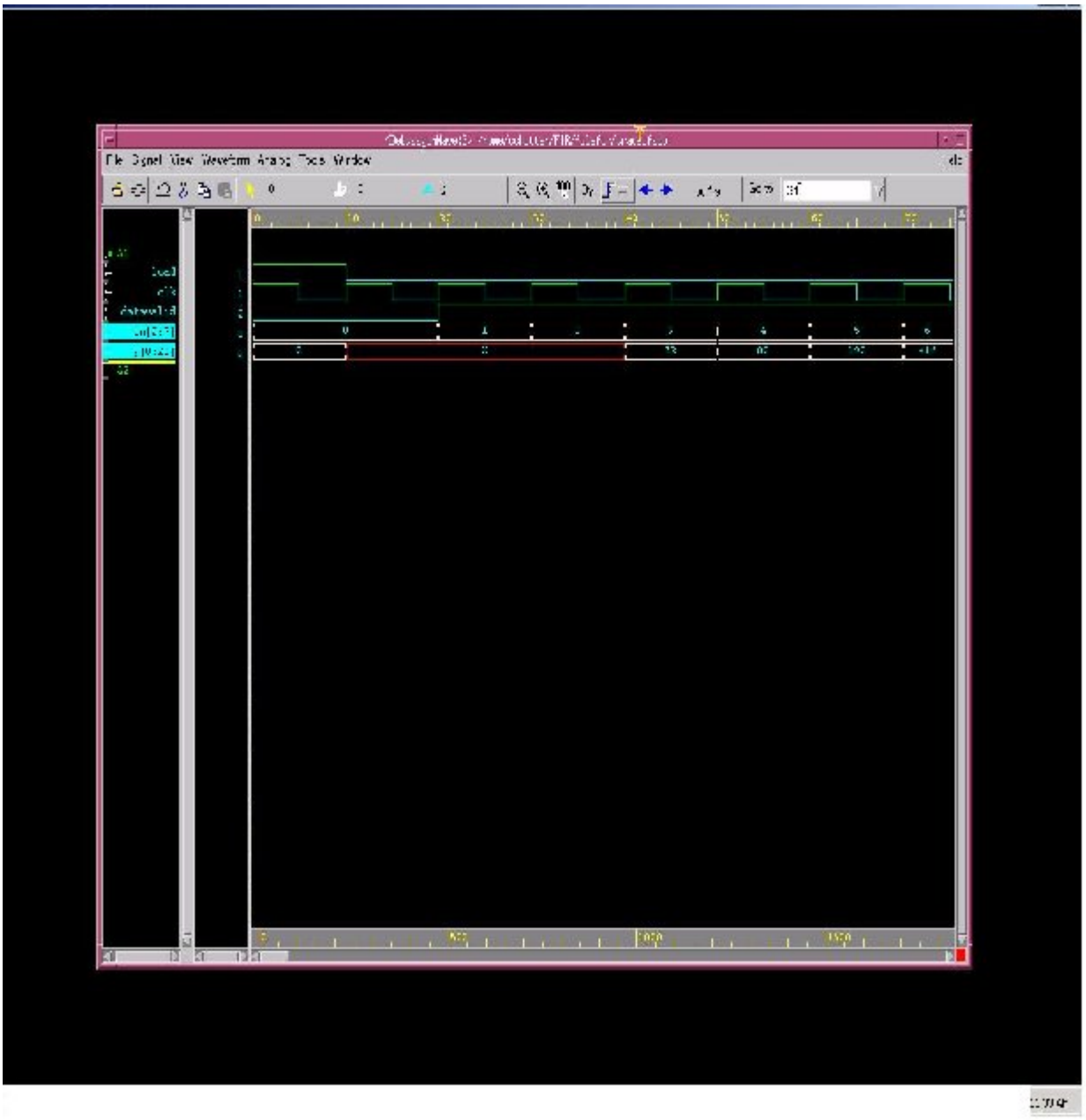
Matlab:

```
Input x = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30]

Coefficients h = [0.125   0.0625   0.25   .5   .4375   .3125   .875   .1875
.03125  .4897  .3673  .765  .876  .00123  .6543]

Output y = [0.1250      0.3125      0.7500      1.6875      3.0625      4.7500
7.3125     10.0625     12.8438     16.1147     19.7529     24.1562     29.4354
34.7159   40.6507   47.2398   53.830161.2964   69.5277   78.1263   87.2146
96.3342  105.6413  115.8233  126.3179  137.2499  148.6820  160.3641  172.1086
183.9782]
```

VCS simulation:

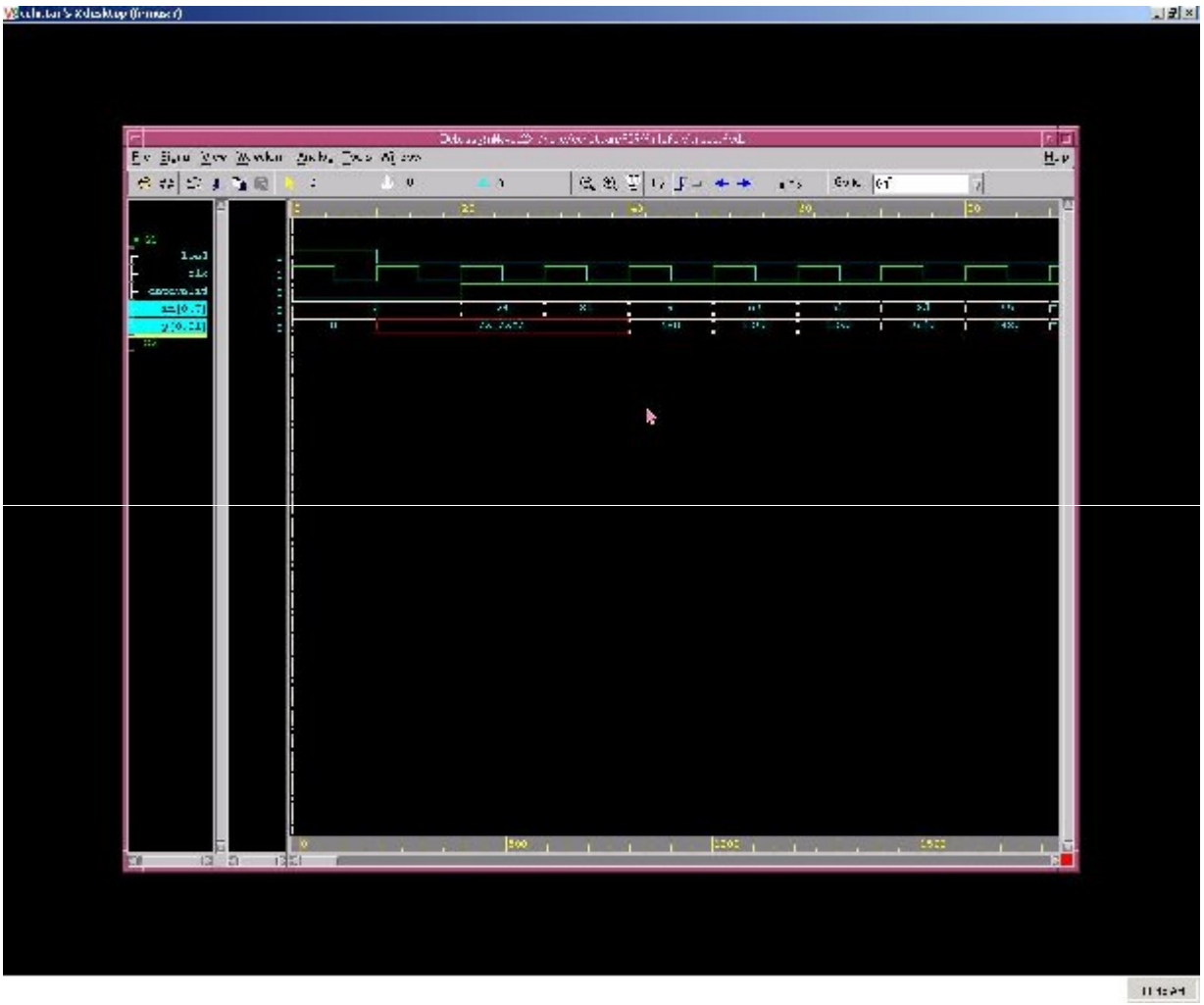


.5 = 100000000 = 128

.25 = 010000000 = 64

.125 = 001000000 = 32

.0625 = 000100000 = 16



5.5 input and output waveforms of the Filter

The outputs are crosschecked using MATLAB tool. The output of the VCS (SYNOPSYS) tool and MATLAB are one and the same. The same is shown in the above figure.

5.3.1 Synthesis Analysis:

DC-Compiler is used to synthesize the RTL code by giving proper constraints to get optimized net list. The Timing, Area and Power report from the DC-Compiler has been extracted and presented below.

Report of area, timing, power:

Total area required for the design of filter is 35k gates per channel and the power consumption is 10 micro watts.

Chapter 7

CONCLUSION

7. CONCLUSION

7.1 ACHIEVEMENT OF THE THESIS:

The RTL code for FFT and poly phase FIR filtering was developed and successfully simulated using VCS Simulator .We have verified with various test vectors to determine that system is working in a proper way. The Design passed the DFT successfully and the design is testable. The result of Post Synthesis is an Optimized Gate Level net list from which a net list code is extracted and it is simulated using VCS Simulator and it is verified that design is working properly. The FFT design was applied with defined constraints for which design works 0.51ns slack (Slack met) .Various reports of the design are obtained that is timing, area, and power. All reports are generated using the Design Compiler and Power Report (which reads VCD file) are also obtained from Primetime tool. The challenge is in meeting the specifications for all the parameters and within the constraints of the design from various aspects in making it the most optimized one.

The results of the various experiments carried out were summarized below,

- The power calculation of the FFT design is found to be 26.41 mw. It uses VCD (Value Dump Change) file to generate power report. This report is more accurate when compared to DC-Compiler generated power report
- The total area of the FFT is 250k gates and the memory required is 84KB
- The power calculation of the FIR filter design found to be 10 microwatts. It uses VCD file generate the power report
- The area of the filter design is around 35k gates.

7.2 SCOPE OF FUTURE WORK

REFERENCES

1. Samir Palnitkar. Digital Design and Synthesis. Prentice Hall PTR 2003

2. Joseph McRae Palmer "**The Hybrid architecture parallel fast Fourier transform**" http://contentdm.lib.byu.edu/ETD/image/etd855.pdf

3. www.patentstorm.us/patents/6951017-description.html

4. Marco Brambilla, Daniele Guide. "**High Speed FIR filters for Digital Decimation",** in proe.Int.Conf on Electronics circuits and systems, September , 2001

5. Brandt B (1994) "**A low power area efficient digital filter for Decimation or Interpolation**", IEEE JSSC 29(6)

6. VCS user guide
www.ecs.syr.edu/organizations/grove/data/vcs.ug.pdf

7. Vaidyanathan, P.P. (1998) "**Multi rate systems and filter banks**" (Prentice Hall, Englewood cliffs, NT)

8. Eugene Grayver, Babak Daneshrad, "**Word-serial Architectures for filtering and variable rate decimation**", wireless integrated systems laboratory, electrical engineering, January 2001.

9. Khalid H. Abed, Vivek Venugopal, "**High speed digital filter design using minimal signed representation**" proceedings of 60th IEEE Midwest Symposium on circuits and systems, 2005

10. L.Naviner, J.F.Naviner "**Design and Prototyping of a Decimator Filter for DECT Standard**", Proceedings of 43rd IEEE Midwest Symposium on Circuits and Systems, 2000

11. Design Compiler user guide
www.csg.csail.mit.edu/6.375/handouts/tutorials/tut4-dc.pdf

12. In-Cheol Park, Hyeong-Ju Kang, "**Digital filter synthesis based on Minimal Signed Digit representation**", Proceedings of 45th IEEE international Midwest Symposium on Circuits and Systems, 2002

13. Matlab user's manual. The Mathworks, INC., Natick, MA, USA, 1983

14. Nobuyuki Nishiguchi. "**An advance RTL to GDS2 design methodology for 90 nm and below system LSIs to solve timing closure, signal integrity and design for manufacturing.**" Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on 23-26 May 2005 Page(s):5938 - 5941 Vol. 6

15. Overview of DSL
www.dslforum.org/mktwork/download/gyoung_ITU-T_0206.pdf

16. E. B. Hogenauer, "**An economical class of digital filters for decimation and interpolation**", IEEE Trans. Acoust., Speech and Sign. Proc., vol.29, 1998

17. J. P. Cornil, J. Sevenhans, P. Spruyt, M. Mielants, and S. Braet, "**DMT ADSL and VDSL circuits and systems**", in Proc. Workshop an advances in Analog Circuit Design (AACD'99), Opio, France, Mar. 1999.

18. Documents provided by Ikanos Communications (India) Pvt Ltd