Design and Implementation of a Group Demultiplexer for Satellite Applications

by

Arati Chandrasekhara, B.Eng, India

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the
requirements for the degree of

Master of Engineering

Ottawa-Carleton Institute of Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Ontario, Canada
April 1998

@ Copyright 1998
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-32394-3
Abstract

Signal regeneration on-board the satellite makes it possible to use different multiple access schemes for the uplink and the downlink. The uplink and the downlink can be separately optimized for greater power efficiency and bandwidth utilization. A multiple access scheme using frequency division multiple access (FDMA) for the uplink and time division multiplex (TDM) for the downlink has been considered for regenerative satellite communications. This method has been found more efficient and cost effective as compared to using either FDMA or TDMA for both the uplink and downlink transmission. The conversion from frequency to time division multiplex format requires the uplink signal to be demultiplexed and each individual carrier to be demodulated.

The demultiplexing part of the process can be performed by a number of different structures. Among those the one that has received the most attention for satellite applications is the polyphase filter network followed by a fast fourier transform (FFT) processor. Also, the environment in which the satellite operates necessitates the use of radiation-hardened technology for on-board processing (OBP) hardware. Where volumes and economic factors warrant it, an application specific integrated circuit (ASIC) implementation based on radiation-hardened gate arrays could be the preferred technology. However, in view of the cost and time constraints associated with such an implementation, it is of interest to consider the viability of implementing OBP hardware on radiation-hardened field programmable gate arrays (FPGAs).
This thesis addresses the design and implementation of a polyphase-FFT 8 channel, T1 rate, group demultiplexer, to test the viability of OBP hardware realization with radiation-hardened FPGA technology. The design of the group demultiplexer for implementation on FPGAs with modest gate counts involves identifying suitable architectures for the building blocks of the group demultiplexer. These architectures allow a trade-off between the required number of gates (device utilization) and speed of operation. The architectures developed ensure that the device utilization is low and that the requirements of speed are met.

An implementation which has been optimized in terms of speed and device utilization is presented in this work. Also, addressed here is the possibility of extension of the implementation to incorporate larger number of channels. The implementation presented here utilizes the largest available radiation-hardened devices from Actel (8K gates) and its non-hardened counterpart (8K gates). Also, considered here are the implications of the implementation of the design on larger radiation-hardened devices when they become available. The practicality of applying FPGA technology to the implementation of OBP circuits will increase considerably once radiation-hardened FPGAs with a higher gate equivalence become available.
Acknowledgments

I would like to thank my supervisors, Dr. Tad Kwasniewski and Dr. Valek Szwarc, for their guidance and support throughout my research at Carleton University and Communications Research Centre (CRC).

Carleton University, Canadian Institute for Telecommunications Research (CITR) and Communications Research Centre (CRC) are gratefully acknowledged for providing the financial support. I thank my colleagues at CRC for their help and support.

Finally, to my family and friends: thank you for your love and support.
CONTENTS

List of Tables................................................................................................................. ix
List of Figures.................................................................................................................. x
List of Abbreviations.......................................................................................................... xii

Chapter 1: Introduction ...................................................................................................... 1
1.1 Background.................................................................................................................. 1
1.2 Thesis Motivation......................................................................................................... 6
1.3 Thesis Objectives......................................................................................................... 7
1.4 Thesis Outline.............................................................................................................. 8

Chapter 2: On-Board Processing Demultiplexer Architectures ........................................ 9
2.1 Introduction.................................................................................................................. 9
2.2 Per-Channel Method.................................................................................................. 10
2.3 Block Method............................................................................................................. 13
   2.3.1 The FFT/IFFT Frequency Domain Filtering Approach........................................ 13
   2.3.2 Polyphase-FFT..................................................................................................... 15
2.4 Multistage Method..................................................................................................... 16
2.5 Comparison of the Demultiplexer Architectures....................................................... 20
2.6 Conclusion.................................................................................................................. 22

Chapter 3: Polyphase-FFT Demultiplexer ....................................................................... 23
3.1 Introduction.................................................................................................................. 23
3.2 On-Board Receiver Structure.................................................................................... 24
3.3 Analysis of the Polyphase-FFT Digital Group Demultiplexer.................................... 27
   3.3.1 The Input FDMA Signals....................................................................................... 27
   3.3.2 The Polyphase-FFT Network............................................................................... 29
   3.3.3 Interpolation in the Channel Processor for Rate Conversion................................. 34
   3.3.4 The Final Output.................................................................................................. 37
3.4 Conclusion.................................................................................................................. 39
Chapter 7: Summary and Suggested Further Research and Development

7.1 Further Research and Development

References

Appendix A

Timing Data

Appendix B

List of Coefficients for Polyphase and Rate Conversion Filters and Phase Shift Multipliers

Appendix C

Raised-Cosine Filter
List of Tables

Table 4.1: Characteristics of Actel FPGA families .................................................. 46
Table 4.2: Radiation-hardened target specifications ................................................. 48
Table 4.3: Features of Actel RH1280 and A1280XL devices ..................................... 49
Table 4.4: Specifications for the operating conditions (commercial applications) .. 50
Table 5.1: System parameters of the implemented group demultiplexer ................. 59
Table 5.2: Parameters for the building blocks of the implemented group demultiplexer .......................................................... 60
Table 5.3: Permissible interconnect delays between devices .................................. 69
Table 6.1: Parameters for the polyphase filters ......................................................... 75
Table 6.2a: Device utilization and delay for two subfilters (real and quadrature) with the phase shifter for that channel for channels 1, 2, 3 and 4 ........ 84
Table 6.2b: Device utilization and delay for two subfilters (real and quadrature) with the phase shifter for that channel for channels 5, 6, 7 and 8 .......... 85
Table 6.3: Device utilization and delay for the 8-point, 12 bits input/output (complex) DIT-FXT ............................................................................. 98
Table 6.4: Parameters for the rate conversion filters ................................................. 100
Table 6.5: Timing and device utilization for the two rate conversion filters (real and quadrature) on a single device ................................................. 103
Table A1: Timing information for the input and output for the devices p1 to p8 .... 112
Table A2: Timing information for the input and output for the FFT devices f1a, f1b, f2 and f3 .......................................................... 113
Table A3: Timing information for the input and output for the eight devices of the rate conversion filters ..................................................... 114
Table B1: Coefficients for the polyphase filter ......................................................... 115
Table B2: List of phase shifter multipliers ............................................................... 116
Table B3: Coefficients for the rate conversion filter ................................................ 117
List of Figures

Figure 1.1: Block diagram of group demodulator .................................................. 4
Figure 2.1: "Per-Channel" demultiplexer (Analytic signal approach) ......................... 11
Figure 2.2: Frequency demultiplexing in the AS method ............................................ 12
Figure 2.3: FFT/IFFT frequency domain filtering method for channel k ....................... 15
Figure 2.4: "Block" demultiplexer (polyphase-FFT) .............................................. 17
Figure 2.5: "Multistage" demultiplexer (4 channels) ............................................. 18
Figure 2.6: Multistage spectra ............................................................................ 19
Figure 3.1: Frequency spectrum of the transmitted FDMA signal ................................ 24
Figure 3.2a: On-board receiver structure ................................................................... 25
Figure 3.2b: Frequency spectrum at points A,B,C,D,0,1,N-1 .................................... 26
Figure 3.3: Baseband model for N=8 channels as input to the demultiplexer .......... 28
Figure 3.4: Model for demultiplexing of a single channel ........................................ 29
Figure 3.5: Polyphase network/ FFT processor used to demultiplex N channel...... 32
Figure 3.6: Timing relationship between input and output samples in the rate conversion filter ................................................................. 35
Figure 4.1: Basic Actel FPGA architecture ............................................................... 42
Figure 4.2: Standard flow for FPGA design based on the Actel and Mentor Graphics tools ................................................................. 53
Figure 5.1: Polyphase-FFT demultiplexer with rate conversion filters .................... 58
Figure 5.2: BER vs. SNR curve for the implemented group demultiplexer ............ 61
Figure 5.3: Interfacing between the 12 devices used in the implementation of the polyphase-FFT demultiplexer ................................................. 63
Figure 5.4: Timing relationship between the clocks for the 8 channels for the polyphase subfilters ........................................................................... 66
Figure 5.5: Input and output delays ................................ .............................................. 67
Figure 6.1: Decomposition of $h_{RC_{1/4}}(n)$ with N=4 ........................................ 73
Figure 6.2: Direct form realization for filter for channel 1 ........................................ 77
Figure 6.3: Block diagram for the implementation of the polyphase subfilters ... 78
Figure 6.4: Binary fractional fixed format representation ......................................... 79
Figure 6.5: Block diagram for the 8 point radix-2 DIT FFT ....................................... 89
Figure 6.6: Eight point DIT FFT algorithm .............................................................. 90
Figure 6.7: Complex butterfly for DIT FFT ............................................................ 91
Figure 6.8: Flow diagram of the 8 point DIT FFT ...................................................... 94
Figure 6.9: Implementation of the radix-2, 8 point DIT FFT ..................................... 95
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td>Analytic Signal</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CITR</td>
<td>Canadian Institute for Telecommunications Research</td>
</tr>
<tr>
<td>CLB</td>
<td>Configurable Logic Block</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>CRC</td>
<td>Communication Research Centre</td>
</tr>
<tr>
<td>CSA</td>
<td>Carry Save Adder</td>
</tr>
<tr>
<td>DEMOD</td>
<td>Demodulation</td>
</tr>
<tr>
<td>DEMUX</td>
<td>Demultiplexing</td>
</tr>
<tr>
<td>DIT</td>
<td>Decimation-in-Time</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>EPROM</td>
<td>Electrically Programmable Read Only Memory</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read Only Memory</td>
</tr>
<tr>
<td>FDM</td>
<td>Frequency Division Multiplex</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>GaAS</td>
<td>Gallium Arsenide</td>
</tr>
<tr>
<td>HPA</td>
<td>High Power Amplifier</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>MCD</td>
<td>Multicarrier Demodulation</td>
</tr>
<tr>
<td>MCDD</td>
<td>Multicarrier Demultiplexer Demodulation</td>
</tr>
<tr>
<td>MCM</td>
<td>Multi-Chip Module</td>
</tr>
<tr>
<td>OBP</td>
<td>On-Board Processing</td>
</tr>
<tr>
<td>ONO</td>
<td>Oxide-Nitride-Oxide</td>
</tr>
<tr>
<td>PLICE</td>
<td>Programmable Low Impedance Circuit Element</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>SAW</td>
<td>Surface Acoustic Wave</td>
</tr>
<tr>
<td>SCPC</td>
<td>Single Channel Per Carrier</td>
</tr>
</tbody>
</table>
Si  Silicon
SOS  Silicon On Sapphire
SRAM  Static Random Access Memory
SSB  Single SideBand
TDM  Time Division Multiplex
TDMA  Time Division Multiple Access
Tmux  Transmultiplexer
TTL  Transistor Transistor Logic
TWTA  Travelling Wave Tube Amplifier
VHDL  Very High Speed Integrated Circuit Hardware Description Language
VLSI  Very Large Scale Integration
1.1 Background

To examine the feasibility of field programmable gate array (FPGA) design of a group demultiplexer, a representative subsystem which simultaneously demultiplexes 8 QPSK channels, each with a complex data rate of 0.722 Msamples per second is considered here. The work presented here is in the framework of the on-going research on on-board processing (OBP) techniques at Communication Research Centre (CRC) [1] and Canadian Institute for Telecommunications Research (CITR). OBP enables signal regeneration on board the satellite and makes it possible to use different multiple access
schemes on the uplink and downlink.

For the OBP system frequency-division multiple access using single channel per carrier (SCPC-FDMA) on the uplink and time-division multiplex (TDM) on the downlink is assumed. With SCPC-FDMA, each earth station in the satellite network transmits a radio frequency (RF) carrier to the satellite transponder. Each uplink carrier occupies a specific frequency band location within the uplink bandwidth. In addition, the data rate is typically constant, for all channels, so that each station has the same bandwidth and the network is uniform. With TDMA systems, each station transmits one data frame at a time which is globally synchronized in time to avoid collision. For both FDMA and TDMA the uplink signal is amplified by the satellite’s travelling wave tube amplifier (TWTA) and retransmitted in the downlink beam. As compared to the TDMA method, the SCPC-FDMA method on the uplink reduces the cost and complexity of the earth terminals considerably, because global synchronization is not needed. However, amplification of the multiple carriers by the TWTA operated at saturation on the downlink produces intermodulation distortion which significantly degrades individual channel performance. On the other hand, with TDMA since there is only one carrier at a time, the intermodulation distortion is eliminated [2].

The SCPC-FDMA/TDM link combines the advantages of SCPC-FDMA on the uplink and the benefits of the TDMA on the downlink. Thus, this method reduces the cost and complexity of the earth stations and allows the satellite amplifiers to operate at satura-
tion for maximum efficiency. It enables high power amplifier (HPA) power and antenna size at the earth terminal to be reduced [3]. Also, splitting of the satellite link into two distinct sections prevents noise and other interference to be accumulated and transferred from the uplink to the downlink, which improves performance.

The above discussions indicate the benefits of using FDMA on the uplink and TDM on the downlink in satellite communications systems. The FDMA/TDM configuration necessitates on-board signal regeneration in which the FDMA uplinks are frequency demultiplexed and then the individual carriers demodulated, routed and recombined into TDM signals for retransmission.

The digital signal processing (DSP) operation which simultaneously downconverts multiple FDM signals to baseband is referred to as multicarrier demodulation and the device which accomplishes this is called a multicarrier demodulator (MCD). Multicarrier demodulation includes frequency demultiplexing of the FDM carriers and subsequent demodulation of each carrier to recover the individual channel data. Figure 1.1 depicts the block diagram for a group demodulator. It consists of two main blocks: a digital group demultiplexer and a bank of channel processors. The digital group demultiplexer separates out the individual channels in the received signal. The function of the channel processors is then to regenerate each signal by performing timing and carrier recovery and making decisions on the QPSK signal.

Several digital approaches to demultiplexing have been proposed. A very good
FDMA/QPSK signals → LNA → DN → X → LPF → A/D → Digital Group Demultiplexer → Channel Processor → Channel Processor

\[ \sin 2\pi f_s t \]

\[ \cos 2\pi f_s t \]

LNA: Low noise amplifier
DN: Down converter
A/D: Analog to digital converter

Figure 1.1: Block diagram of group demodulator
description of digital methods to convert FDM to TDM and vice-versa (transmultiplexing methods) is available in [4]. Of the various transmultiplexing designs the one that has received the most attention for on-board processing consists of a digital polyphase network followed by an FFT processor. The basic structure of the digital transmultiplexer on which the group demultiplexer is based was first proposed by Bellanger [5]. The structure proposed by Bellanger was aimed at telecommunication networks, for direct translation from TDM to FDM format or vice-versa. The first polyphase-FFT structure for on-board processing was proposed by Takahata et al. [3].

Several authors have also considered a demultiplexer design based on a series of half-band filters arranged in a binary tree. This multistage demultiplexer was first proposed by Tsuda et al. [6] for use in switched telephone networks. Gockler and Eyssele [7] have provided a design that is suited for on-board processing.

The frequency domain approach [8], [9] is another method for performing demultiplexing. This method becomes attractive for configurations with large number of channels with carriers of different bit-rates and bandwidth requirements. Another approach, the per-channel method [10], [11], [12], performs the demultiplexing operation by individual bandpass filters, each assigned to an SCPC channel.

The high-speed, low-power and high levels of integration required determines the technologies which can be used for on-board processors. Furthermore, the space environment in which these devices operate dictates the use of radiation-hardened devices. Both
Silicon (Si) and Gallium Arsenide (GaAs) based technologies are promising for on-board processing applications [8].

1.2 Thesis Motivation

The Communications Research Centre (CRC), Ottawa in collaboration with Canadian industry, Government agencies and Universities is involved in carrying out research and development in hardware and software technologies applicable to the design of high performance application specific integrated circuit (ASIC) and FPGA based digital circuits and subsystems for communication and image processing applications.

The on-going research at CRC addresses the development of techniques and methodology of implementing OBP subsystems for future communication satellites operated at Ka band (20-30 GHz) [1]. Multicarrier demultiplexer demodulation (MCDD) is one of the major subsystems of OBP for communications satellites. The MCDD has two main functions: demultiplexing (DEMUX) and demodulation (DEMOD). The implementation of the group demultiplexer is a component of the current research work at CRC.

This work reports the design and implementation of a representative demultiplexer using radiation-hardened space qualified antifuse FPGAs. In the design, a single unified structure can demultiplex 8 FDMA channels. The channels are uniformly spaced and have the same filtering requirements.

The FPGA technology offers the potential for rapid and economical implemen-
tion of digital circuitry for OBP applications. The limited gate count and throughput of currently available FPGA devices presents a challenge for the implementation of the OBP hardware such as group demultiplexers.

1.3 Thesis Objectives

The main objective of this thesis is the development of a design methodology for the implementation of an 8 channel, T1 rate, group demultiplexer on radiation hardened FPGAs. This incorporates:

- Investigation of the building blocks for the group demultiplexer and exploration of suitable architectures for the building blocks, and the arithmetic modules incorporated in them.

- The study of the input/output pin requirements and interfaces for the building blocks.

- Optimization of the design, for minimum gate count (device utilization) and throughput, and consequently power, to enable the implementation of the design on radiation hardened and space qualified small antifuse FPGAs,

- Optimization of the interfacing between the building blocks and the devices used for implementation and to investigate clock distributions internal to the device.

Evaluation of the FPGA technology and its limitations, with the view of implementing group demultiplexers with a larger number of channels and greater data throughput, are also discussed in this work. The possibility of implementation of the design on
larger Actel devices is also explored.

It is indicated by the implementation of the group demultiplexer, that it is possible to implement DSP systems on small antifuse FPGA's and to optimize their performance.

1.4 Thesis Outline

This thesis is organized in the following manner. Chapter 2 provides an overview and comparison of OBP demultiplexer architectures. In Chapter 3, an analysis of the polyphase-FFT demultiplexer and its architecture is presented. Chapter 4 reviews the Actel FPGA technology and describes the computer aided design (CAD) tools used in the implementation of the group demultiplexer. In Chapter 5, the system architecture is presented. Chapter 6 gives a description of the implementation of the building blocks of the polyphase-FFT demultiplexer and the implementation results for them. Chapter 7 summarizes the thesis and provides recommendations for further research in this area.
CHAPTER 2

On-Board Processing Demultiplexer Architectures

2.1 Introduction

The use of uplink FDMA techniques (with low cost earth stations) and downlink TDM techniques (that can fully exploit the satellite transponder output power without intermodulation distortion), is an attractive option for satellite systems. The feasibility of this approach, however, depends on an efficient means of translating between the two formats on board the satellite. The complexity of the on-board system (including the VLSI design) and power consumption are obviously of primary concern. The on-board processing system receives an input FDMA signal and supplies an output TDM signal; therefore
it must be able to separate each radio channel, perform demodulation and correctly switch to the appropriate downlink channel.

The function of the demultiplexer is to separate the individual input FDMA channels and to route each of them to a demodulator for the appropriate demodulation and decoding at baseband. Hence, in principle the operation of the demultiplexer corresponds to a bank of bandpass filters followed by a bank of downconverters. This section deals with efficient demultiplexer architectures and a comparison of the various architectures.

2.2 Per-Channel Method

This method performs the demultiplexing operation by individual bandpass filters each assigned to an SCPC channel and is similar to analog signal filtering. Selection of each input signal and its translation to a low-frequency band is achieved by the operation of digital filtering and decimation (i.e. decrease of the signal sampling rate).

Within the class of per-channel methods, an effective solution is represented by the Analytic Signal (AS) approach [10], [11], [12]. This method is basically concerned with the generation of a complex SSB baseband signal (AS) and its successive allocation to one of the channels by complex bandpass filtering and digital decimation.

The implementation structure of the AS method envisages a cascade of complex bandpass filters separated by a decimator (by a factor of $N$, the number of channels). This implementation is depicted in Fig. 2.1, where $H_i$ is a complex bandpass filter performing
2.1: "Per-channel" demultiplexer (Analytic signal approach)

A rough selection of $i$th channel (out of $N$ demultiplexer input channels), having passband and transition bandwidths approximately equal to the channel spacing. The filter $H_i$ operates at the input (high) sampling rate and supplies the output at a low sampling rate after decimation of the signal by a factor of $N$. The $G_i$ complex filters, operating at the low sampling frequency, carry out the required additional filtering to select the channels and to translate them to the proper low-frequency bands. Linear phase finite impulse response (FIR) filters are conveniently utilized in this kind of approach to avoid phase distortions and to obtain high implementation efficiency. The frequency demultiplexing performed by the AS method for four channels is depicted in Fig. 2.2.
Figure 2.2: Frequency demultiplexing in the AS method

- **a**: FDM input signal
- **b, c**: frequency response of high rate channel filters $H_i$
- **d, e**: spectra of the filtered FDM signal
- **f, g**: spectra of the complex signal obtained by decimation over $N$
- **h, i**: frequency response of the low rate channel filters $G_i$
- **j**: frequency response of the low pass prototype
- **k, l**: spectra of the complex demultiplexed signal
- **m, n**: recovered baseband spectra
The required number of multiplications-per-second and per-channel required by the Analytical Signal method is given by [12]:

\[ M_{AS} = K_{AS} \times W^2 \left[ W(N + 4) - 2B(N + 2) \right] \div \left[ (W - 2B)(W - B) \right] \]  

(2.1)

where

\[ K_{AS} = \left( \frac{2}{3} \right) \log \left( \frac{2}{10\delta_1 \delta_2} \right) \]

\[ N = \text{number of demultiplexer input channels} \]

\[ W = \text{channel spacing in Hz} \]

\[ B = \text{one-side bandwidth of the signal spectrum in Hz} \]

\[ \delta_1 = \text{pass band amplitude ripple} \]

\[ \delta_2 = \text{stop band amplitude ripple} \]

2.3 Block Method

When demultiplexing a large number of carriers the use of a block approach is preferred to keep the computational rate to a lower level. These methods capitalize on the computational savings of the FFT and result in structures with lower computational complexity than the per-channel methods. Two approaches of the block method: the FFT/IFFT (inverse FFT) frequency domain filtering method and the polyphase-FFT are discussed in the following sections.

2.3.1 The FFT/IFFT Frequency Domain Filtering Method

The FFT/IFFT frequency domain filtering method consists of convolving the frequency multiplexed signal with a bank of filters using an overlap and save technique [13], [14]. The desired linear convolutions are computed in terms of circular convolutions. The
circular convolutions are computed by transforming the time domain quantities to be convolved to the frequency domain, multiplying the resulting frequency coefficients across the overall spectrum by the desired filter functions, and transferring back to the time domain. The main computational requirement is thus to perform Fourier transformations on the input signal blocks and to perform inverse Fourier transformations on a carrier-by-carrier basis. In an actual implementation the filter frequency coefficients are precomputed and stored in the memory. The size of the FFT is $N+L$ where $N$ is the number of channels and $L$ is the overlap used between blocks. To obtain the individual baseband signals, the number of inverse transforms to be performed equals the number of carriers $N$. Since the inverse transforms are performed on an individual carrier basis, the case of carriers with different bandwidths is readily handled by performing inverse transforms of larger sizes for the wideband carrier.

Referring to Fig. 2.3, to obtain signal $k$ from the frequency multiplexed signal ($S_{in}$), the signal is transferred to the frequency domain by an FFT ($X(k)$). It is then applied to the input of the frequency response of the desired filter ($H(k)$), typically a square root Nyquist transfer function, that serves the double purpose of demultiplexing and matched filtering. The product is then transferred back with an IFFT to recover the time domain waveform ($S_{outk}$).

This approach is very flexible in handling any mix of carriers having different bit rates, bandwidths, and center frequencies. Changes in the frequency assignment plan
CHAPTER 2: On-Board Demultiplexer Architectures

Time domain
\[ S_{in} \]
Overlapping data blocks
FFT
\[ X(k) \]
X
\[ H(k) \]
Y(k)
IFFT
\[ S_{outk} \]
Frequency domain
Time domain

\[ N = \text{total number of channels} \]
\[ Y(k) = X(k)H(k) \]

Figure 2.3: FFT/IFFT frequency domain filtering method for channel \( k \)

may be easily accommodated by reprogramming via ground control. Only the frequency filter coefficients stored in random access memory (RAM) and a recording buffer that controls the size of the inverse transforms to be performed need to be reprogrammed [15], [8].

2.3.2 Polyphase-FFT

The polyphase-FFT block consists of a bank of FIR filters followed by an FFT processor. Both the filters and the FFT operate at a lower sampling frequency, thus reducing the overall complexity. The number of channels (\( N \)) to be demultiplexed should be a
power of two, in order to fully exploit the advantages of the FFT techniques based on algorithms such as the Cooley Turkey algorithm. The number of FIR filters and the size of the transform is equal to the number of carriers being demultiplexed.

When all the carriers have the same bit rate, the polyphase-FFT method represents the most efficient technique for demultiplexing (particularly when the number of carriers is large) [8]. However, when the carrier bit rates are not equal, this method cannot be used. Figure 2.4 depicts the polyphase-FFT demultiplexer with its various building blocks. $P_0$ to $P_{N-1}$ represent the polyphase subfilters which are followed by the FFT processor.

In this case the overall number of real multiplications per-second per-channel is given by [12]:

$$M_{FFT} = 2W[L_{FT}/N + 4\log_2 N]$$

(2.2)

where

$L_{FT} = (4/3)NW\log[1/(10\delta_1 \delta_2)]/(W - 2B)$

$N$ = number of demultiplexer input channels

$W$ = channel spacing in Hz

$B$ = one-side bandwidth of the signal spectrum in Hz

$\delta_1$ = pass band amplitude ripple

$\delta_2$ = stop band amplitude ripple

2.4 Multistage Method

The multistage method can be considered as a binary tree of two-channel demultiplexers. Each demultiplexing stage performs lowpass and highpass filtering with sub-
sequent decimation by a factor of two.

The multistage method can be used whenever the number \( N \) of the demultiplexer channels is a power of two. The signal spectrum is split down into two subbands by halfband filters and decimated by a factor of two. Both filter outputs are again split by two fil-

Figure 2.4: "Block" demultiplexer (polyphase-FFT)
ters and decimated leading to a division into four bands. After $L (\log_2 N)$ stages, $N$ channels are obtained. The multistage method for four channels is depicted in Fig. 2.5 and the spectra in Fig. 2.6.

![Multistage demultiplexer diagram]

Figure 2.5: "Multistage" demultiplexer (4 channels)

The number of multiplications per-second per-channel [11] required by this method is given by

$$M_{MS} = (64/3) \times W (\log [1/(10\delta_1 \delta_2)]) [\log_2 N + (2B)/(W - 2B)]$$

(2.3)

where

- $N$ = number of demultiplexer input channels
- $W$ = channel spacing in Hz
- $B$ = one-side bandwidth of the signal spectrum in Hz
- $\delta_1$ = pass band amplitude ripple
- $\delta_2$ = stop band amplitude ripple
Figure 2.6: Multistage spectra
a  input signal
b  first-stage filter (complex)
c  first stage spectra (lower branch)
d  first stage spectra (higher branch)
e  first stage spectra (lower branch) after decimation by two
f  first stage spectra (higher branch) after decimation by two
g  second stage filtered spectra for channel 2
h  second stage filtered spectra for channel 2 after decimation by 2
2.5 Comparison of the Demultiplexer Architectures

There are several ways to compare the various demultiplexing architectures. Here, the computational complexities of the different demultiplexing methods are compared by the number of multiplications per-channel per-second. Also the flexibility of the different architectures to accommodate variations in the channel transmission rates are compared.

Equations 2.1, 2.2 and 2.3 give the number of multiplications required by the three methods. Once the other parameters have been set, for small values on N the difference in complexity is a function of W. The difference in computational complexity for low values of N between the three methods is not significant, however, for larger values on N, the polyphase-FFT method has the lowest computational complexity.

Keeping all the parameters constant, for N equal to 8, the ratio of multiplications per-second per-channel for the per-channel method and polyphase-FFT method is 1.19, while for N equal to 32 it is 2.20. Similarly, for the multistage method and the polyphase-FFT method, the ratios are 7.3 and 7.73 respectively. These numbers indicate that the polyphase-FFT method does indeed require the least number of multiplications.

From an implementation point of view, the considerations and conclusions which can be drawn about the three approaches based on studies in [11], [12] are:

- Per-channel method generally has higher computational complexity, smaller finite precision arithmetic sensitivity and smaller control circuit complexity. Per-channel structures have greater flexibility and are well suited to variations of transmission rates and
number of processed channels $N$. The per-channel method allows processing of channels with different transmission rates within the same demultiplexer, as the $N$ paths are substantially independent.

-Polyphase-FFT method has lower computational complexity, higher finite precision arithmetic sensitivity and greater control circuit complexity. The Polyphase-FFT method has limited flexibility as it is able to operate only at a fixed value of $N$ (i.e. number of points of the FFT processor). The FFT/IFFT frequency domain filtering method, however, has a high degree of flexibility and can support a mix of different signal bandwidths.

-Multistage method has computational complexity comparable with that of block methods, finite precision arithmetic sensitivity and control circuit complexity comparable to per-channel methods. Multistage structures have an intermediate degree of flexibility, as they allow variations of the transmission data rates, although limited to powers of two.

A comparison made among the methods in [16] shows that in terms of the computational complexity (number of operations per-channel per-second) the polyphase-FFT method is the most efficient in all cases. Also the storage requirements per-channel follow a similar trend to the computational rate. The multistage architecture is more suitable where a small number of low bit rate channels require high bandwidth utilization. Also, if all carriers to be demultiplexed have the same bit rate (and bandwidth) then the polyphase-FFT architecture is preferred.
2.6 Conclusion

In this chapter three main architectures for OBP demultiplexer were presented: per-channel, block and multistage. A comparison of the architectures and their suitability for different applications was also discussed. From the comparison it can be concluded that for a large number of channels, the polyphase-FFT is very suitable as its computational complexity is lowest. However, the per-channel method and the FFT/IFFT frequency domain filtering method have a large degree of flexibility and are suited for demultiplexing channels with different transmission rates.

The polyphase-FFT method is the most efficient for the present implementation, since a representative group demultiplexer which simultaneously demultiplexes 8 channels with the same bit rate and bandwidth, is to be implemented. Also, the channels being a power of two and being equally spaced make the polyphase-FFT method the preferred method, in view of our interest in developing an architecture that can be extended to a larger number of channels.
CHAPTER 3

Polyphase-FFT Demultiplexer

3.1 Introduction

The advantages of using the polyphase-FFT digital group demultiplexer for on-board processing have been discussed in Chapter 2. This chapter provides an overview of the receiver on-board the satellite for multicarrier demodulation. An analysis of the polyphase-FFT digital group demultiplexer is provided in Section 3.3 to show how the architecture of the polyphase-FFT digital group demultiplexer is arrived at.
3.2 On-Board Receiver Structure

Each earth station transmits on its own carrier frequency. Assuming that there are N earth stations, the resulting frequency spectrum of the FDMA signal in the RF band is shown in Fig. 3.1

![Figure 3.1: Frequency spectrum of the transmitted FDMA signal](image)

The receiver structure on-board the satellite for group demodulation is shown in Fig. 3.2a and the spectrum at points A, B, C, D, 0, 1, N-1 is shown in Fig. 3.2b. The FDMA signal in the RF band is amplified by the LNA (low-noise amplifier) and is translated to the IF band by a down-convertor. Then, the IF signal is frequency-converted to baseband signal with in-phase (I) and quadrature phase (Q) components. Before the I and Q components are sampled by the analog to digital (A/D) convertors, a low pass filter (i.e. anti-aliasing filter) is required to remove the out-of-band frequency components to avoid alias-
ing. Then, the discrete-time samples are routed to the digital group demultiplexer (DEMUX). The function of the DEMUX is to simultaneously filter and down convert to baseband the composite carriers. In addition, filtering or pulse shaping of all carriers is achieved concurrently in the DEMUX. Symbol timing and carrier recovery for each output is required before data can be detected. Symbol timing adjustment enables the data to be detected at the maximum 'eye-opening' point while carrier recovery ensures that any carrier phase and frequency error is compensated for coherent demodulation. Finally, the recovered data from each channel can be combined into a serial TDM format for downlink transmission.

Figure 3.2a: On-board receiver structure
Figure 3.2b: Frequency spectrum at points A, B, C, D, 0, 1, N-1
3.3 Analysis of the Polyphase-FFT Digital Group Demultiplexer

3.3.1 The Input FDMA Signals

The analysis here follows that in [17] and is provided to clarify the functionality and relationships of the group demultiplexer's building blocks. The input to the multi-carrier demodulator $y(t)$ is a baseband signal consisting of $N$ FDMA signals $\sum_{k=0}^{N-1} s_k(t)$ and noise $z(t)$, and is given by

$$y(t) = \sum_{k=0}^{N-1} s_k(t) + z(t)$$  (3.1)

For QPSK modulation, the $k$th FDMA signal takes the form

$$s_k(t) = \sum_{i=-\infty}^{\infty} A_k e^{j(2\pi f_k t + \alpha_{k,i} + \phi_k)} h(t - i T_b - \gamma_k)$$  (3.2)

where $A_k$ is the carrier amplitude, $f_k$ is the frequency offset associated with the channel, $\alpha_{k,i}$ is the data phase in the $i$th symbol interval, and $h(t)$ is the transmitted pulse shape, the impulse response of the transmit filter. The parameters $\gamma_k$ and $\phi_k$ represent random timing and phase offsets, respectively. Figure 3.3 gives a model for the baseband signal $y(t)$ when the number of channels is 8. The eight channels are arranged such that

$$f_k = \left(k + \frac{1}{2}\right) f_c,$$

where $f_c$ is the frequency spacing between channels. Normally, one
would represent the baseband signals as being symmetrically distributed about the zero frequency mark, but the representation of Fig. 3.3 proves convenient as the total bandwidth occupied by the signals is $f_s = Nf_c$ which is also the sampling rate.

![Figure 3.3: Baseband model for N=8 channels as input to the demultiplexer](image)

It is assumed that all the channels use the same transmit filter $h_s(t)$, which in general will have a root-Nyquist characteristic. Furthermore, it is assumed that the filtering requirements are split between the transmitter and receiver. If the transmit filter takes on a root-raised cosine characteristic with a rolloff factor $\beta$ (Appendix C), then the signal will occupy a bandwidth of $(1 + \beta)f_b$, where $f_b = 1/T_s$ is the symbol rate. The channel separation $f_c$ must then be greater than or equal to $(1 + \beta)f_b$ [17], [18].
3.3.2 The Polyphase-FFT Network

For the derivation of the multi-carrier demultiplexer structure based on the polyphase-FFT network, we begin with the model of Fig. 3.4 for the demultiplexing of a single channel.

\[ y(nT_s) \xrightarrow{X} y'(nT_s) \xrightarrow{h(nT_s)} \downarrow N \xrightarrow{x_k(mT_c)} \]

\[ -j2\pi f_k nT_s \]

Figure 3.4: Model for demultiplexing of a single channel

The process of isolating the signal in the \( k \)th channel involves shifting the signal to baseband and filtering with a low-pass filter. Since the original signal was sampled at \( N \) times the rate of a single channel, we can also decimate the output of the low-pass filter by a factor of \( N \). The sampled impulse response of the low-pass filter is denoted by \( h(nT_s) \), and it is assumed that the same filter characteristic is used in the demultiplexing of all the channels, \( 0 \leq k \leq N - 1 \). It is also assumed that \( h(nT_s) \) is a finite impulse response filter.
with $NL$ taps. The equation for the combined frequency shifting and filtering operations is then

$$\sum_{n=1}^{NL} h(nT_s) y'[mT_c - nT_s]$$

where

$$y'(nT_s) = y(nT_s)e^{-j2\pi f_s nT_s}$$

Therefore,

$$x_k(mT_c) = \sum_{n=1}^{NL} h(nT_s) y'[mT_c - nT_s]e^{-j2\pi f_s (mT_c - nT_s)}$$

Substituting $NT_s = 1/f_c$,

$$e^{-j2\pi km} = 1 \text{ when } k \text{ and } m \text{ are integers and }$$

$$f_k = (k + 1/2)f_c$$

$$x_k(mT_c) = \sum_{n=1}^{NL} h(nT_s) y[(mN-n)T_s]e^{-j2\pi f_s (mN-n)T_s}$$

$$= e^{-j\pi n} \sum_{n=1}^{NL} h(nT_s) y[(mN-n)T_s]e^{j2\pi f_s (mN-n)T_s/N}$$

(3.3)

If we now express $n$ as

$$n = vN - \rho$$

and sum over both $v$ and $\rho$. Equation 3.3 can be written as
For each value of \( \rho \), and across all values of \( \nu \), each set of filter coefficients \( h[(\nu N - \rho)T_s] \) can be associated with a separate filter

\[
\bar{p}_\rho(\nu T_c) = \bar{p}_\rho(\nu NT_s) = h[(\nu N - \rho)T_s] e^{j\pi \nu} = (-1)^\nu h[(\nu N - \rho)T_s] \tag{3.5}
\]

In a similar fashion, the input signal can be written as \( N \) separate, decimated signals

\[
\tilde{y}_\rho[(m - \nu)T_c] = \bar{y}_\rho[(m - \nu)NT_s] = y[(m - \nu)NT_s + \rho T_s] \tag{3.6}
\]

If we combine Eq. 3.5 and Eq. 3.6, and let

\[
W_N = e^{\frac{j\pi}{N}} \tag{3.7}
\]

then we obtain the final equation for the output

\[
x_k(mT_c) = (-1)^m \sum_{\rho = 0}^{N-1} W_N^{-2kp} W_N^{-\rho} \sum_{\nu = 1}^{L} \bar{p}_\rho(\nu T_c) \tilde{y}_\rho[(m - \nu)T_c]
\]

\[
= (-1)^m \sum_{\rho = 0}^{N-1} W_N^{-2kp} W_N^{-\rho} [\bar{p}_\rho(mT_c)^* \tilde{y}_\rho(mT_c)] \tag{3.8}
\]

where

- "*" denotes digital convolution
- "\( L \)" denotes the number of taps per filter.
Figure 3.5: Polyphase network/FFT processor used to demultiplex N channels
CHAPTER 3: Polyphase-FFT demultiplexer

Working from right to left then, the set of operations defined by Eq. 3.8 is: a digital filtering operation $\tilde{p}_p(mT_c) \ast \tilde{x}_p(mT_c)$, multiplication by a set of phase offsets $[W_N^{-p}]$, followed by a discrete Fourier transform $\left[ \sum_{p=0}^{N-1} W_N^{-2kp} \right]$, and finally a set of alternating sign changes on the output $[(-1)^m]$. The filtering operation is performed at the lower sampling rate $f_c = 1/T_c$ rather than the higher input sampling rate $f_s = Nf_c$. Figure 3.5 gives a block diagram of the overall structure defined by Eq. 3.8. The Fourier transform is performed on the signal $z_k(nT_s)$ which is the output of the polyphase filter multiplied by the phase offset.

The filters $\tilde{p}_p(vT_c)$ are called polyphase filters, as the decimation process of Eq. 3.5 gives a set of filters with the same amplitude response but different phase responses. Assuming that the original filter $h(NT_c)$ has a linear phase response, the filter $\tilde{p}_p(vT_c)$ will have a linear phase response whose slope is a constant times $p/N$.

The fact that the polyphase filters have finite impulse response means that there cannot be perfect separation between the channels and some interchannel interference or crosstalk will occur. The design problem for the multi-carrier demultiplexer is thus a trade-off between minimizing the number of taps $L$ in the filters and reducing the degradation caused by the crosstalk to an acceptable level.
3.3.3 Interpolation in the Channel Processor for Rate Conversion

Following the multi-carrier demultiplexer is the channel processor where symbol timing and carrier phase synchronization are performed. An important part of the channel processor is the rate conversion stage where the input samples at rate \( f_c = 1/T_c \) are converted to a set of samples at the symbol rate \( f_b = 1/T_b \). This rate conversion can be written as a simple digital filtering operation [19].

\[
r_k(lT_b) = \sum_m x_k(mT_c)g(lT_b - mT_c)
\]  

(3.9)

where \( g(t) \) is the impulse response of the interpolating filter.

The relationship between the timing of the input samples \( x_k(mT_c) \) and the output samples \( r_k(lT_b) \) is illustrated on the time line of Fig. 3.6. Here it is seen that there are several input samples for every output sample, but the timing of the two sets does not coincide. Several new indices need to be defined. The first involves converting the signal index \( m \) to a filter index \( \xi \), where

\[
\xi = \text{int} \left[ \frac{lT_b}{T_c} \right] - m
\]  

(3.10)

where \( \text{int} \lfloor z \rfloor \) is the largest integer not exceeding \( z \).

Next, since the objective is to interpolate between the input samples to produce an output sample at time \( lT_b \), we define the basepoint index \( \beta_l \) as the index of the input
sample just preceding the $l$th input sample.

$$\beta_l = \text{int} \left[ \frac{lT_b}{T_c} \right] \quad (3.11)$$

The normalized time difference between the input sample at time $\beta_l T_c$ and the output sample at $lT_b$ is referred to as the fractional interval

$$\mu_l = \frac{lT_b}{T_c} - \beta_l \quad (3.12)$$

Then using the indices of Eq. 3.10, Eq. 3.11 and Eq. 3.12, the rate conversion filtering can be rewritten as
If $g(t)$ is a finite impulse response filter then $I_1$ and $I_2$ are fixed integers and

$$ I = I_2 - I_1 + 1 $$

is the number of filter taps.

The value of $\beta_i$ and $\mu_i$ must be known to obtain the proper filter coefficients $g[(\xi + \mu_i)T_b]$. These two parameters are estimated by the timing recovery algorithm. In general, a finite number of quantization steps has to be used for $\mu_i$ so that the filter coefficients can be stored in memory. In the initial stages of analysis, it is usually assumed that $\mu_i$ and $\beta_i$ are precisely known and then an attempt to determine the number of taps, $L$ for the polyphase filter and $I$ for the rate conversion filters, to minimize the probability of error is made. This is a reasonable assumption given that symbol synchronous transmission is being considered for many on-board processing systems. If symbol synchronous transmission is used, then timing recovery is simplified and the rate conversion filter works its way through a set pattern of filter coefficients. This would also imply that the timing offset $\gamma_k$ in the transmitted signal response for channel $k$ is zero (perfect symbol synchronization) or a random quantity of small magnitude.

In the rate conversion process, there is the choice of performing some of the receiver filtering or it can be used simply as a delay to interpolate between the input sam-
ples \( x_k(mT_c) \). In the latter case, the ideal choice for the impulse response \( g(t) \) is a sinc function. However, the receiver filter can be split between the polyphase filters \( \bar{p}_p(rT_c) \) and the rate conversion filter. If the overall receiver filter characteristic is to be a root Nyquist response, then the polyphase filters and the rate conversion filter will each approximate a fourth-root Nyquist characteristic.

### 3.3.4 The Final Output

In order to analyze the system performance, the expression for the output of the multi-carrier demultiplexer has to be arrived at. It should clearly show that for the \( k \)th channel the desired symbol is the \( i \)th symbol interval plus any unwanted interference terms. To arrive at such an expression, the FDMA input of Eq. 3.1 and Eq. 3.2 are combined with the demultiplexing Eq. 3.3 or Eq. 3.8 and the rate conversion process of Eq. 3.13. In terms of analysis it is easier to work with Eq. 3.3. To begin, the sampled input signal is given as

\[
y[(mN-n)T_s] = e^{j\pi m} \sum_{q=0}^{N-1} A_q e^{j\alpha_q} \sum_{i=-\infty}^{\infty} e^{j\alpha_q,i} h_s[(mN-n)T_s - iT_b - \gamma_q] e^{-j2\pi[q + \frac{1}{2}]n/N}
\]

\[
+ z[(mN-n)T_s] \quad (3.14)
\]

substituting Eq. 3.14 into Eq. 3.3, the demultiplexer output for the \( k \)th channel is
\[
x_k(mT_c) = A_k e^{j\phi_k} \sum_{i=-\infty}^{\infty} e^{j\alpha_{k,i}} \sum_{n=1}^{\infty} h(nT_s)h_s((mN-n)T_s - iT_b - \gamma_k) \\
+ \sum_{q=0}^{N-1} A_q e^{j\phi_q} \sum_{i=-\infty}^{\infty} e^{j\alpha_{q,i}} \sum_{n=1}^{\infty} h(nT_s)h_s((mN-n)T_s - iT_b - \gamma_q)e^{j2\pi(k-q)\frac{n}{N}} \\
+ e^{-j\pi n} \sum_{n=1}^{\infty} z[(mN-n)T_s]h(nT_s)e^{j2\pi(k+\frac{1}{2})\frac{n}{N}}
\]  

(3.15)

Next, proceeding through the rate conversion stage, the equation for the final output can be given by

\[
r_k(lT_b) = A_k e^{j\phi_k} \left[ e^{j\alpha_{k,l}}u(0) + \sum_{i=-\infty}^{\infty} e^{j\alpha_{k,i}}u(lT_b - iT_b) \right] \\
+ \sum_{q=0}^{N-1} A_q e^{j\phi_q} \sum_{i=-\infty}^{\infty} e^{j\alpha_{q,i}}v(lT_b - iT_b) + z_k(lT_b)
\]  

(3.16)

where

\[
u(lT_b - iT_b) = \sum_{\xi = l_1}^{l_2} \sum_{\xi = 1}^{NL} h(nT_s)h_s((\beta_i - \xi)T_c - nT_s - iT_b - \gamma_q)g((\xi + \mu_i)T_c) \\
u^{j2\pi(k-q)\frac{\xi}{N}}
\]  

(3.17)

\[
u^{j2\pi(k-q)\frac{\xi}{N}}
\]  

(3.18)
and

\[
z_k(t) = \sum_{\xi = 1}^{l_2} \sum_{n = 1}^{N_2} z[(\beta_l - \xi)T_c - nT_s]h(nT_s)g[(\xi + \mu)s] e^{-j\pi(\beta_l - \xi)} \left( e^{j2\pi k + \frac{1}{2}} \right)_{N_2} e^{-j\pi k e^{-j\pi k N_2}}
\]  

(3.19)

The first term inside the square brackets on the right of Eq. 3.16, \( e^{j\alpha_k} u(0) \), is the desired symbol for the \( k \)th channel and the \( l \)th symbol interval. The second term in the brackets represents intersymbol interference, while the final summation is the interchannel interference or crosstalk from the \( N - l \) other channels.

3.4 Conclusion

Through the analysis of the polyphase-FFT group demultiplexer, it is shown that the demultiplexing of the FDMA signals can be achieved by means of a combination of a polyphase network, phase shifters and an FFT processor.

Further, the sampling rate at the output of the demultiplexer is not suitable for the subsequent demodulation process, where an integer number of samples per data symbol is required. A sampling rate conversion filter is necessary for interfacing between the demultiplexed channels and the demodulator.
CHAPTER 4

Actel FPGA Technology, CAD tools, Design and Test Environment

4.1 Introduction

The introduction of advanced CAD tools has greatly simplified the design and implementation process, vastly reduced design cycle time and improved the quality of products. Starting from a given design specification, CAD tools allow designers to capture its schematic and perform extensive verification and simulation at different levels.

This chapter discusses the Actel FPGA technology and tools, describes the CAD tools, along with the design and test environment which were used for the implementation of the group demultiplexer.
4.2 Actel Antifuse FPGA Technology

Actel Corporation is a leading supplier of high performance field programmable gate arrays (FPGAs) and provides antifuse based architecture in the FPGA market [20]. Currently Actel is the only vendor manufacturing space qualified FPGA’s. The environment in which the satellite operates necessitates the use of space qualified devices. Hence, for the implementation of the group demultiplexer for satellite applications, the Actel radiation-hardened FPGA is an appropriate choice.

4.2.1 Actel Device Architecture

The underlaying architecture of an Actel FPGA is very similar to that of a conventional gate array. The core of the device consists of simple logic modules which are used to implement the required logic gates and storage elements. These logic modules are interconnected with an abundance of segmented routing tracks. The segmented lengths are pre-defined and can be connected with low-impedance switching elements to connect the logic modules. Surrounding the logic core is the interface to the I/O pads of the devices. This interface consists of I/O modules that translate and interconnect the logic signals from the core of the device to the FPGA output pads. A block diagram of a generic Actel FPGA is given in Fig. 4.1 [21]. The major elements of the Actel FPGA architecture are thus, the I/O modules, interconnect resources, clocking resources, and logic modules. Each Actel FPGA family has a slightly different mix of these resources, optimized for different cost.
Figure 4.1: Basic Actel FPGA architecture
performance, and density requirements.

The Actel FPGA architecture is based on the programmable low impedance circuit element (PLICE) antifuse technology. The PLICE antifuse is built using an oxide-nitride-oxide (ONO) dielectric between N+ diffusion and polysilicon. The antifuse is functionally opposite to a conventional fuse. An antifuse is a two terminal device with an unprogrammed state presenting a very high resistance between its terminals. When a high voltage (16 volts) is applied across its terminals, the antifuse will blow and create a low-resistance link, which is permanent [22].

A major advantage of the antifuse is its small size. This advantage is somewhat diminished by the size of the transistors, which must be large enough to be able to handle large currents, and the inclusion of isolation transistors that are needed to protect low voltage transistors from high programming voltages. A second major advantage of an antifuse is its relatively low series resistance. Also the parasitic capacitance of an unprogrammed amorphous antifuse is significantly lower than that of the SRAM, EPROM and EEPROM programming technologies [22].

The Actel logic block is based on the ability of a multiplexer to implement different logic functions by connecting each of its inputs to a constant or to a signal [22]. For example, consider a two-to-one multiplexer with inputs, \( s \) (select signal), \( a \) and \( b \) and output \( f = sa + \bar{s}b \) by setting signal \( b \) to logic 0, the multiplexer can implement the AND function \( f = sa \). Setting signal \( a \) to logic 1 provides the OR function \( f = s + b \). By con-
nnecting together a number of multiplexers and basic logic gates, a logic block can be constructed which can implement a large number of functions in this manner. In the ACT-1 device, the sequential logic is not explicitly present and so must be formed using programmable routing and the purely combinational logic blocks. In ACT-2/1200XL, ACT-3 and 3200DX devices there are two alternating types of logic blocks: the C-module which is purely combinatorial, and the S-module which has similar combinational functionality to the C-module but includes a D flip-flop. The C-module is a combinatorial module optimized to implement high fan-in macros such as 5-input AND, and 5-input OR. The S-module is designed to implement high-speed sequential functions within a single module. The S modules consist of a full C module driving a flip-flop, which allows an additional level of logic to be implemented without additional propagation delay.

Multiplexer-based logic blocks have the advantage of providing a large degree of functionality for a relatively small number of transistors. This is, however achieved at the expense of a large number of inputs which when utilized place high demands on the routing resources. Such blocks are, therefore, more suited to FPGAs that use programmable switches of small size such as antifuses.

The I/O modules provide the interface between the device pins and the logic array. Low-skew, high fanout clock distribution networks are provided in each device. The Actel architecture uses vertical and horizontal routing tracks to interconnect the various logic and I/O modules. These routing tracks are metal interconnects that may either be of con-
tinuous length or broken into pieces called segments. Varying segment lengths allow the interconnection of 90% of design tracks to occur with only two antifuse connections. Segments can be joined together at the ends, using antifuses to increase their length up to full length of the track. All interconnects can be accomplished with a maximum of four antifuses.

4.2.2 Actel FPGA Families and Macro Library

Actel offers ACT1 ACT2/1200XL ACT3 and 3200DX families of devices which represent different types of architectures. The user-definable I/O’s are capable of driving at both TTL and CMOS levels. Each I/O pin is available as an input, output, three-state, or bidirectional buffer. Table 4.1 shows the characteristics of Actel FPGA families [21].

When designing with Actel FPGAs, the appropriate library must be used. The elements available in each of the Actel Libraries are listed in the ACT Family Macro Library Guide. The ACT family Macro Library consists of six types of macros Hard, Soft, I/O, TTL, JTAG, and ACTgen parameterized macros.

Hard macros are logic elements constructed of either one or two combinatorial and/or sequential modules and cannot be modified. Soft macros are logic blocks constructed of multiple Hard macros, which can be modified, if necessary. I/O macros include input, output and bidirectional buffers, latches and registers. TTL macros offer a variety of logic functions which include adders, shifters, counters and registers. ACTgen parame-
trized macros are those created by the Actel ACTgen macro builder.

Table 4.1: Characteristics of Actel FPGA families.

<table>
<thead>
<tr>
<th>Family Resources</th>
<th>ACT 1</th>
<th>ACT 2/1200XL</th>
<th>3200DX</th>
<th>ACT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Module</td>
<td>Simple Logic Module</td>
<td>Combinatorial and Sequential Modules</td>
<td>Combinatorial and Sequential Modules Wide Decode Modules Embedded Dual-Port SRAM</td>
<td>Combinatorial and Enhanced Sequential Modules</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Channeled</td>
<td>Channeled</td>
<td>Channeled</td>
<td>Channeled</td>
</tr>
<tr>
<td>Clocking Resources</td>
<td>Routed Clock(1)</td>
<td>Routed Clocks(2)</td>
<td>Routed Clocks(2) Quad Clocks(2)</td>
<td>Routed Clocks(2) Dedicated Array Clock Dedicated I/O Clock</td>
</tr>
<tr>
<td>I/O Modules</td>
<td>Simple I/O Module</td>
<td>Latched I/O Module</td>
<td>Latched I/O Module</td>
<td>Registered I/O Module</td>
</tr>
</tbody>
</table>

4.2.3 Radiation-Hardened FPGAs

Actel Corporation has introduced radiation-hardened (rad-hard) versions of the A1280 (RH1280) and A1020 devices (RH1020), with gate densities of 8,000 and 2,000
array gates, respectively [23]. The features of these device include total radiation dose capability, low single event upset susceptibility, high dose rate survivability and latch-up immunity, as indicated in Table 4.2.

The target specifications for the Actel radiation-hardened devices [23] as compared to the typical values for radiation hardened technology [24], [25] and Harris radiation hardened CMOS/SOS family [26] are given in Table 4.2. The given values represent threshold values, beyond which the devices are likely to fail.

A description of the parameters included in Table 4.2 is provided here. Total dose is the measure of nuclear radiation energy absorbed. Low rates of ionizing radiation have negligible effect on circuit operation. The amount of total dose received is measured in rad. Single event is a single, highly ionizing particle travelling through a memory circuit. Single event upset (SEU) is a transient change in circuit biases which can cause transient errors in the circuit. Latchup refers to any stable or quasi-stable mode of device operation in which the postradiation relationship between device's electrical input and output differ significantly from the preradiation relationship for these parameters. Neutron fluence is a measure of the neutron irradiation which can damage circuits by knocking silicon atoms out of their place in the lattice structure. It is measured in neutrons/cm².

The RH1280 and RH1020 devices are latch-up immune, indicating that no catastrophic failures will occur. Single event upsets can occur in these devices as with other commercial semiconductor chips, but the rate of upset is low, as shown in the target speci-
Table 4.2: Radiation-hardened target specifications

<table>
<thead>
<tr>
<th>Feature</th>
<th>ACTEL specifications</th>
<th>Typical specifications</th>
<th>Harris specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Dose</td>
<td>&gt; $5 \times 10^5$ rad*</td>
<td>&gt; $2 \times 10^5$ to $10^6$</td>
<td>&gt; $10^6$</td>
</tr>
<tr>
<td>Single Event Upset (SEU)</td>
<td>&lt; $10^{-7}$ errors/bitday</td>
<td>&lt; $10^{-10}$ to $10^{-12}$</td>
<td>&lt; $10^{-10}$</td>
</tr>
<tr>
<td>Latchup</td>
<td>Immune</td>
<td>Immune</td>
<td>Immune</td>
</tr>
<tr>
<td>Dose Rate Upset</td>
<td>&gt; $10^9$ rad(Si)/sec</td>
<td>&gt;$3 \times 10^8$ to $4 \times 10^9$</td>
<td>&gt; $10^{11}$</td>
</tr>
<tr>
<td>Survivability</td>
<td>&gt;$10^{12}$ rad(Si)/sec</td>
<td>NA</td>
<td>&gt; $10^{12}$</td>
</tr>
<tr>
<td>Neutron Fluence</td>
<td>&gt; $10^{14}$ /cm$^2$</td>
<td>&gt;$3 \times 10^4$ to $2 \times 10^{15}$</td>
<td>&gt; $10^{14}$</td>
</tr>
</tbody>
</table>

* rad is a measure of absorbed energy, here in silicon.

The total dose, dose rate upset and neutron fluence are comparable with other radiation-hardened devices. The RH1280 and RH1020 are supported by Actel’s Designer and Designer Advantage Systems. These devices are targeted for use in military and space applications subject to radiation effects.

4.2.4 RH1280 and A1280XL device features

The implementation of the group demultiplexer was done on the RH1280 as well as the A1280XL devices. The basic features of these two devices are listed in Table 4.3
Table 4.3: Features of Actel RH1280 and A1280XL devices

<table>
<thead>
<tr>
<th>Features</th>
<th>A1280XL</th>
<th>RH1280</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package used</td>
<td>160PQFP(^{(1)})</td>
<td>172CQFP(^{(2)})</td>
</tr>
<tr>
<td>Total no. of pins</td>
<td>160</td>
<td>172</td>
</tr>
<tr>
<td>User pins</td>
<td>125</td>
<td>140</td>
</tr>
<tr>
<td>Total logic gates</td>
<td>8000</td>
<td>8000</td>
</tr>
<tr>
<td>Total logic modules</td>
<td>1232</td>
<td>1232</td>
</tr>
<tr>
<td>C-modules</td>
<td>608</td>
<td>608</td>
</tr>
<tr>
<td>S-modules</td>
<td>624</td>
<td>624</td>
</tr>
<tr>
<td>CMOS Process</td>
<td>0.6 micron</td>
<td>0.8 micron</td>
</tr>
<tr>
<td>Speed Grade</td>
<td>-2(^{(3)})</td>
<td>STD.300KR(^{(4)})</td>
</tr>
<tr>
<td>Application(^{(5)})</td>
<td>Commercial</td>
<td>Commercial</td>
</tr>
</tbody>
</table>

(1) Plastic Quad Flat Pack (PQFP)  
(2) Ceramic Quad Flat Pack (CQFP)  
(3) Speed Options available for the A1280XL are  
STD = standard speed  
-2 = approximately 25% faster than STD (-1 corresponds to 15%)  
(4) Speed Options available for the RH1280 are  
STD.0.KR = standard speed with 0 rad radiation dose level  
STD.300.KR = standard speed with \(3 \times 10^5\) rad radiation dose level  
(5) Temperature and voltage ranges

Both device types can be simulated under the best, typical and worst case operating conditions. The specifications for the three conditions for commercial applications are listed in Table 4.4
Table 4.4: Specifications for the operating conditions (commercial applications)

<table>
<thead>
<tr>
<th>Case</th>
<th>Temperature</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst</td>
<td>70 °C</td>
<td>4.75 V</td>
</tr>
<tr>
<td>Typical</td>
<td>25 °C</td>
<td>5.00 V</td>
</tr>
<tr>
<td>Best</td>
<td>0 °C</td>
<td>5.25 V</td>
</tr>
</tbody>
</table>

4.2.5 Actel FPGA CAD Tools

The Actel FPGA CAD tools are referred to as the Designer Series Development System. The Designer Series software is used in the design, optimization and programming of Actel FPGAs. The Designer Series software includes the following software programs:

- ACTgen macro builder: enables creation of macros. This tool supports the creation of a variety of macros such as adders, subtracters, accumulators, comparators, counters, decoders, multiplexers, shift registers, storage registers, latches and I/O macros. It allows the bit width and the set of control signals present on a macro to be defined. These macros optimize the Actel architecture with respect to performance and module count according to the required specifications.

- ACTmap VHDL synthesis: provides a simple user interface for synthesis, logic optimization, retargeting among FPGA families and importing of existing PAL designs,
- DESIGNER: provides an environment for checking and optimizing the netlist (Compile), assigning pin information (PinEdit), editing and placing macros (ChipEdit), assigning timing information (DTEdit), mapping the design into silicon (Layout), debugging timing problems (DTAnalyze), creating a file that is used to program the device (Fuse), and extracting delays for back annotation (Extract).

- APSW/APS2: enables the programming of Actel FPGA devices.

4.2.6 Actel FPGA Design Cycle

The design process for using Actel FPGAs requires the following steps:

1. Entering the design in the form of schematic, netlist, logic expressions, or hardware description languages.

2. Simulating the design for functional verification.

3. Mapping the design into the FPGA architecture.

4. Placing and routing the FPGA design.

5. Extracting delay parameters of the routed design.

6. Resimulating for timing verification.

7. Generating the FPGA device configuration format.

8. Configuring or programming the device.

Steps 1, 2 and 6 were done using Mentor Graphics CAD tools. The group demultiplexer was entered in the form of a schematic using Design Architect from the Mentor Graphics CAD tools. Macros from Actel ACT2 libraries and those generated using the Actgen tool were used in the schematic. Steps 2 and 6 were done using the Quicksim simulator tool from Mentor Graphics. The design was simulated for operation under the worst, typical and best operating conditions. Steps 3 and 4 involve several processes: logic minimization, technology mapping, placement and routing, which were done using the Actel Designer Series Software System. Logic minimization is done by the “combiner” which is integrated into the Compile function of the Designer series software system. It improves the design in terms of density, speed and routability by performing the functions of combinatorial module reduction, sequential remapping, unused logic removal, constant output reduction and fan-in reduction. Technology mapping binds the technology-independent description of the circuit to the basic entities of the Actel technology. Placement allocates these entities to the specific physical block on the device. Routing establishes the connections between different blocks. Placement and routing was done using the Layout tool. Step 5 was carried out using the DTAnalyze feature of the Actel Designer series software. This includes a static timer tool which was used to determine the delays through the circuit. Steps 7 and 8 were performed using the Fuse and APSW/APS2 tools. Figure 4.2 depicts the interface between the Mentor Graphics tools and the Actel tools.
Figure 4.2: Standard flow for FPGA design based on the Actel and Mentor Graphics tools.
4.3 Mentor Graphics Design Environment

Below is a brief description of the Mentor Graphics CAD tools which were used for the present work. The Falcon Framework is a powerful common environment in which all Mentor Graphics applications run. These applications use the Falcon Framework to provide a common user interface, text editor and decision support system. The Falcon Framework consists of the Design Manager and other applications. The Design Manager allows the user to view and navigate design data, move, copy and manage data, and invoke applications.

Design Architect is a tool box containing a collection of tools (graphic and test editors) that are used to create and modify the data that is modelling a design. The three primary tools in the Design Architect are the Symbol Editor, the Schematic Editor, and the VHDL Editor.

Quicksim II within the Mentor Graphics tool set simulates a digital schematic design from Design Architect. Simulation is the modelling and behavioral analysis of an electronic design without actual physical hardware implementation.

4.4 Conclusion

This chapter provided an overview of the Actel FPGA technology with emphasis on architecture, capabilities and design cycle. The CAD tools from Mentor Graphics and Actel were also discussed. The Actel tools are interfaced with the Mentor tools to facili-
tate design and implementation of the circuit.

Actel supplies the only radiation-hardened FPGA's available today. The RH1280 and RH1020 are radiation-hardened versions of the A1280A and A1020A devices with gate counts of 8000 and 2000 gates respectively. In addition to being radiation-hardened these devices incorporate all the other features of the A1280A and A1020A devices. The Actel radiation-hardened device specifications are comparable to those of other radiation hardened devices. These devices are suited for applications which are subject to the effects of radiation.
5.1 Introduction

In this chapter the building blocks for the polyphase-FFT group demultiplexer are identified and the architecture of the polyphase-FFT group demultiplexer is reviewed. This chapter addresses the interfacing between the building blocks of the polyphase-FFT group demultiplexer and discusses the input, output and clock requirements for the building blocks and the system. The delay introduced by the input and output pads for each device used in the implementation is also discussed.
5.2 Building Blocks and Architecture of the Polyphase-FFT Group Demultiplexer

The basic building blocks in the structure of the polyphase-FFT group demultiplexer are the commutator, a bank of polyphase subfilters, a set of phase shifters, FFT processor and a set of sign inverters. Figure 5.1 depicts the building blocks for the polyphase-FFT group demultiplexer.

To recover the signal in each of the $N$ channels, the received FDMA signal in the baseband is passed to a commutator which separates the input sampled data into $N$ different streams. The sampled data is processed in a parallel fashion by the polyphase subfilters, and the sampling rate is reduced by $N$ times due to decimation by the commutator. Each data stream goes into a digital subfilter. This set of subfilters is referred to as a polyphase filter bank.

The outputs from the polyphase filters are then multiplied by the phase shifts before being processed by the FFT processor. The signs of the FFT outputs are inverted for every other sample to yield the demultiplexed QPSK signals in the baseband. To facilitate processing by the demodulator, a sampling rate conversion filter is needed at the output of the demultiplexer for each channel.
5.3 System Specifications and Parameters

The basic system parameters of the implemented group demultiplexer are summarized in Table 5.1 and the parameters for the polyphase subfilters, FFT processor and a representative rate conversion filter are summarized in Table 5.2. These parameters are in accordance with the research work done at CRC [18], [1]. Two's complement arithmetic is used in the implementation of the building blocks.

Table 5.1: System parameters of the implemented group demultiplexer

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulation scheme used for the input</td>
<td>QPSK</td>
</tr>
<tr>
<td>Number of QPSK signals</td>
<td>8</td>
</tr>
<tr>
<td>Channel separation</td>
<td>1.544 MHz</td>
</tr>
<tr>
<td>Total bandwidth of the modulated signal</td>
<td>12.352 MHz</td>
</tr>
<tr>
<td>Data symbol rate (T1 rate)</td>
<td>0.772 Msamples per sec</td>
</tr>
<tr>
<td>Data symbol rate with 3/4 convolutional coding</td>
<td>1.029 Msamples per sec</td>
</tr>
<tr>
<td>Sampling frequency for real and quadrature</td>
<td>12.352 MHz</td>
</tr>
<tr>
<td>Input / Output resolution</td>
<td>8 bits/12 bits</td>
</tr>
<tr>
<td>Filtering</td>
<td>full raised cosine Nyquist</td>
</tr>
<tr>
<td>Roll off factor for the filter</td>
<td>$\beta = 0.4$</td>
</tr>
</tbody>
</table>

The results presented in [18] indicate that acceptable performance in terms of the
Table 5.2: Parameters for the building blocks of the implemented group demultiplexer

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Module</th>
<th>FIR polyphase filter</th>
<th>FIR rate conversion filter</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristic</td>
<td>Module</td>
<td>Root fourth Nyquist</td>
<td>Root fourth Nyquist</td>
<td>8 point</td>
</tr>
<tr>
<td>Number of subfilters</td>
<td>Module</td>
<td>8 complex</td>
<td>8 complex</td>
<td>NA</td>
</tr>
<tr>
<td>Input/Output resolution</td>
<td>Module</td>
<td>8 bits/12 bits</td>
<td>12 bits/12 bits</td>
<td>12 bits/12 bits</td>
</tr>
<tr>
<td>Coefficient quantization</td>
<td>Module</td>
<td>7 bits</td>
<td>7 bits</td>
<td>12 bits</td>
</tr>
<tr>
<td>No. of taps/filter</td>
<td>Module</td>
<td>7 taps</td>
<td>9 taps</td>
<td>NA</td>
</tr>
</tbody>
</table>

Bit error rate (BER) for a given signal to noise (SNR) ratio \(E_b/N_0\) can be obtained with either 7 or 9 taps per filter and 7 or 8 bits of quantization for each coefficient. The bit error rate curve for an 8-channel group demodulator for polyphase filter with 7 taps, rate conversion with 9 taps and 7 bit quantization is shown in Fig. 5.2. [18]. The rate conversion filter implemented here is a representative filter. The actual implementation will require an adaptive filter.

5.4 Interfacing between the Building Blocks

The 8 channel group demultiplexer is implemented on twelve RH1280 and also on twelve A1280XL devices. Two polyphase subfilters, one each for the real and quadrature
Figure 5.2: BER vs. SNR curve for the implemented group demultiplexer
inputs with the corresponding phase shifter is implemented on a single device. The implementation of the eight polyphase subfilters for the real inputs, eight subfilters for quadrature inputs with the corresponding phase shifters thus requires a total of eight devices. The 8-point radix-2, Decimation-in-Time (DIT) FFT is implemented on four devices. The sign inversion of every alternate output of the FFT is done in the devices on which the third stage of the FFT is implemented. The two rate conversion filters for each channel for the real and quadrature inputs are implemented on one device. Hence, eight complex rate conversion filters for all eight channels of the group demultiplexer require eight devices for realization.

Proper interfacing between the devices used for the implementation of the group demultiplexer ensures desired functionality. Figure 5.3 depicts the interfacing between the twelve devices used in the implementation of the polyphase-FFT group demultiplexer. Devices representing the polyphase subfilters and phase shifters are denoted by $p1$ to $p8$, the FFT processor by $f1a, f1b, f2$ and $f3$.

In this implementation, the address bus $al$ (3 bits) is used to route the input data samples from the A/D converter to the appropriate filter. This enables the implementation to be extended easily to a larger number of channels, simply by using more devices for the extra channels and increasing the number of bits for the address line. The limited number of user input/output pins available for each device dictates the manner in which the data is routed between the various building blocks. Also, the architecture of the building
Figure 5.3: Interfacing between the 12 devices used in the implementation of the polyphase-FFT demultiplexer
blocks and the interfacing between them is chosen to keep the gate count and the I/O count as low as possible, in order to keep the power dissipation of the device to a minimum. The interfacing between the devices is summarized here with reference to Fig. 5.3.

Two 8-bit data bus \((i1)\) each for the real and quadrature data samples from the A/D converter are routed to each of the devices \((p1\) to \(p8)\) of the polyphase subfilters. In each device a decoder circuit decodes the 3-bit address line \((a1)\) and routes the input data sample to the subfilters. The 12-bit complex data sample output from the subfilters for each channel is processed by the corresponding phase shifter. The 12-bit complex output data samples from the phase shifter are then processed by an FFT. The 12-bit complex data samples from channels 1, 2, 3 and 4 are routed on a 12-bit complex bus \((i2)\). Similarly, the 12-bit outputs from channel 5, 6, 7 and 8 are routed on bus \(i3\). The throughput capabilities achieved for the butterfly in the first stage enabled it to be reused. As a requirement for this, the data is routed on two buses to the FFT. Limited number of input/output pins on each device prevented a separate 12-bit complex bus to be used for each channel. The FFT's output data samples are routed on four 12-bit complex buses \((i4, i5, i6\) and \(i7)\), with the output for two channels available on one bus.

### 5.5 Clocking Scheme

Circuits operating at lower rates have lower power dissipation, and hence it is desirable that the logic module switching rate be as low as possible. For the polyphase-
CHAPTER 5: Polyphase-FFT Group Demultiplexer System Architecture

FFT group demultiplexer the building blocks operate at $1/N$ (N is the number of channels) times the input sampling rate. This enables the polyphase filters, FFT and the rate conversion filters to operate at a lower frequency resulting in a reduction in the power dissipation of the devices on which they are implemented.

Clock skew can cause setup and hold time violation problems, hence the system clock needs to be routed to all the twenty devices, through a low skew network. Two dedicated subnano second skew networks [20] with high fanout, available in the RH1280 and A1280XL devices are used to route the system clock and the lower frequency clock within each device. The low frequency clock may either be made available from an external circuitry or may be generated within each device. Since the number of logic blocks required to generate the clock within the device is very low, approximately 0.3% of the total blocks per device, and generating a clock inside each device reduces interfacing, routing and skew problems, this approach is used here.

To ensure minimum power dissipation, each of the polyphase subfilters is operated at the low frequency of $1/8$ the sampling frequency. The address line in synchronism with the system clock is decoded to generate the clock for each channel. The timing relationship between the clocks for the 8 channels is depicted in Fig. 5.4.

The low frequency clocks for the rate conversion filter are generated by decoding the address lines, in a manner similar to the one used for the polyphase subfilters.

The nature of the implementation of the FFT requires that the first stage of the FFT
operate at 1/2 the system clock, the second stage at 1/4 the system clock and the third stage at 1/8 the system clock. The clocks required for each stage are generated within the device using a clock divider network.

The clock generation requirements within the group demultiplexer dictates that the clock signal be available when the first data becomes available.

5.6 Timing Information

In a synchronous system, the clock-to-output delay is important for determining the setup conditions of the next device on the printed circuit board. The output delay is the time for the signal to be available at the output pads as indicated in Fig. 5.5. Similarly, the
input delay is the time for the signal to arrive on-chip from the input pads. This information is needed to provide proper delays for the interconnects between the devices on the printed circuit board. Delay through the interconnect between two devices should be less than the difference of the system clock and the sum of the input delay of the second device and the output delay of the first device.

![Figure 5.5: Input and output delays](image)

The post layout timing delays for the inputs and outputs for the devices used in the implementation of the group demultiplexer are provided in Appendix A. From the data presented, it is observed that for the worst and typical case operating conditions, the A1280XL devices have 20-40% lower delays than the RH1280. However, for the best
case operating condition the delays of the two devices are comparable. A similar trend in the performance is observed for the logic modules and is presented in a later chapter. The reason for this difference can be attributed to two factors. First, the A1280XL devices are processed in 0.6 μ technology while the RH1280 are processed in 0.8 μ technology making the A1200XL faster. Second, the A1280XL devices used for the implementation are speed grade -2 which is about 25% faster than the standard speed that is used for the RH1280 devices.

Another point of interest is the fact that tristate output buffers used for the polyphase subfilters exhibit longer delay (10%) as compared to the output buffers without tristate logic. This observation is of consequence for high speed circuits where the use of tristate output buffers might slow down performance. For high speed circuits, manual input/output pin placement may be used to reduce the delays through the input/output pads.

The input and output delays also provide information regarding the maximum delays permissible between devices. From the computed data, for worst case operating conditions, the delay permissible between the devices is presented in Table 5.3. These values do not incorporate the delay through the clock buffer and the effect of the clock skew. If the delay through the clock buffer, which is approximately 10ns for worst case operating conditions and 3 ns for best case operating conditions, is taken into account, the values in Table 5.3 should be augmented by at least 3ns.
Table 5.3: Permissible interconnect delays between devices

<table>
<thead>
<tr>
<th>Between devices</th>
<th>Permissible delay RH1280</th>
<th>Permissible delay A1280XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1 to p8 and f1a &amp; f1b</td>
<td>42.4 ns</td>
<td>55 ns</td>
</tr>
<tr>
<td>f1a &amp; f1b and f2 &amp; f3</td>
<td>52 ns</td>
<td>62.3 ns</td>
</tr>
</tbody>
</table>

The polyphase subfilters and the rate conversion filters can be reset using the clear signal, whose characteristics are such that it must be active low for 75ns or more.

5.7 Conclusion

This chapter described in detail the interfacing between the building blocks and the devices used in the implementation of the polyphase-FFT demultiplexer. The timing information which is very critical to the functioning of the system was also provided.

For low power dissipation and increased reliability it is desirable to keep the I/O pins to a minimum and to operate the logic modules at the lowest possible frequencies. The dedicated low-skew networks available on the A1280XL and RH1280 devices, used to route the clocks, keep the clock skews within each device to subnano second values.

Proper setup and hold times between the devices need to be provided to enable the data samples to be correctly transferred and latched. The input and output delays provide the information necessary to ensure that.
6.1 Introduction

There are a number of ways in which each building block of the polyphase-FFT group demultiplexer can be implemented on an FPGA. The optimum design requires a consideration of the routing and logic block architecture of the FPGA being used.

The building blocks hardware design is based on the functional, fixed word mathematical representation. This chapter presents the manipulation of the original equations (Chapter 3) leading to a final hardware representation for each of the building blocks. A detailed description of the implementation of the building blocks on the Actel RH1280.
and A1280XL devices is given. Results for the implementations describing their performance in terms of speed and device utilization are also provided.

6.2 Polyphase Filter Network

The analysis here is based on [2]. Each data substream at the output of the commutator goes into a digital subfilter. These subfilters are derived from a prototype filter. Let \( h[n] \) be the prototype filter with sampling frequency \( 1/T_s \), then the digital subfilter impulse response is defined as

\[
h_k[n] = h[nN + k](-1)^n
\]

where \( h_k[n] \) has sampling interval \( T_c \) which is equal to \( NT_s \).

Let \( H_0(\omega) \) be the frequency spectrum of \( h_0[n] \), then

\[
H_0(\omega) = \sum_{n = -\infty}^{\infty} h_0[n]e^{-j\omega T_c}
\]

\[
= \sum_{n = -\infty}^{\infty} h_0(nT_c)e^{-j\omega nT_c}
\]

where \( \omega \) is the frequency in radians per second. Similarly, the frequency response of the \( k \)th filter \( h_k[n] \) is
\[ H_k(\omega) = \sum_{n=-\infty}^{\infty} h_k[n] e^{-j\omega n T_c} \]

\[ = \sum_{n=-\infty}^{\infty} h_0 \left( n + \frac{k}{N} \right) T_c e^{-j\omega n T_c} \]

\[ = \sum_{n=-\infty}^{\infty} h(n T_c) e^{-j\omega \left( n - \frac{k}{N} \right) T_c} \]

\[ = H_0(\omega) e^{j\omega k T_c / N} \quad (6.3) \]

The set of subfilters \( h_k[n] \) is called a polyphase filter bank since each path approximates a pure phase-shifter with the same frequency but different phase shifts. To minimize intersymbol interference, the system employs raised-cosine pulse shaping which is shared equally between the transmitter and the receiver. Thus, each received signal has been shaped by square-root raised cosine filter [Appendix C]. The prototype filter \( h[n] \) has frequency response of a fourth-root raised-cosine with baud period \( T \). That is,

\[ h(n) = h(n T_c) = h_{RC^{1/4}}(t) \bigg|_{t=n T_c}, \quad (6.4) \]

where \( h_{RC^{1/4}}(t) \bigg|_{t=n T_c} \) is a fourth-root raised cosine filter. The full raised-cosine response will be obtained by passing the output to another fourth-root raised cosine filter in the per-channel processor. Figure 6.1 depicts the decomposition of the \( h_{RC^{1/4}}[n] \) with \( N=4 \).
Figure 6.1: Decomposition of $h_{RC^{1/4}}[n]$ with $N=4$

(a) Before decomposition $h_{RC^{1/4}}[n]$
(b) $h_{RC^{1/4}}[4n + 0]$
(c) $h_{RC^{1/4}}[4n + 1]$
(d) $h_{RC^{1/4}}[4n + 2]$
(e) $h_{RC^{1/4}}[4n + 3]$
The filtering of the complex signal by the subfilter is performed by passing the real and imaginary parts of the signal into two identical subfilters since the coefficients of the subfilters are real.

6.2.1 Filter Coefficients for the Polyphase Network

The coefficients for the polyphase filter were derived using the Matlab* based software developed at CRC [18]. The design method for the filter is to build a prototype filter of 8192 taps starting in the frequency domain and using an inverse FFT to get the time domain response. The impulse response of the filter is then truncated to the desired number of taps. This is done to get better accuracy on the coefficients. The polyphase prototype filter gives the 56 taps used in the implementation of the 8 polyphase subfilters. The frequency response of the prototype is a fourth root raised cosine response to split the receiver filtering between the polyphase network and the rate conversion filter. The parameters chosen for the polyphase filters are listed in Table 6.1. A list of the coefficients for the polyphase filters is provided in Appendix B.

6.2.2 Implementation of the Polyphase Subfilters

An 8 channel group demultiplexer with complex inputs is implemented. Each

* Matrix Laboratory software package for technical computing from The Math Works Inc.
Table 6.1: Parameters for the polyphase filters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized frequency spacing (Fs)</td>
<td>1.5</td>
</tr>
<tr>
<td>Number of channels</td>
<td>8</td>
</tr>
<tr>
<td>Rolloff factor for the raised cosine filter</td>
<td>0.4</td>
</tr>
<tr>
<td>Window used</td>
<td>Kaiser</td>
</tr>
<tr>
<td>Beta factor for the Kaiser window</td>
<td>2</td>
</tr>
<tr>
<td>Number of taps per subfilter</td>
<td>7</td>
</tr>
<tr>
<td>Number of subfilters</td>
<td>8</td>
</tr>
<tr>
<td>Total number of taps</td>
<td>56</td>
</tr>
<tr>
<td>Number of bits per filter taps</td>
<td>7</td>
</tr>
</tbody>
</table>

A polyphase subfilter is implemented as an FIR filter with 7 taps, with 7 bits of quantization for the coefficients [18]. The inputs are complex but the coefficients are real, hence input for each channel needs two subfilters to generate the complex output. Two's complement fixed format arithmetic has been used in the implementation of the polyphase subfilters. The inputs to the polyphase subfilter are 8 bits integers, represented in binary fixed format.

The errors due to truncation in the polyphase filter ranges from \((-2^{-12} \text{ to } 0)\). The errors due to format conversion are not considered here. Assuming the errors to be uniformly distributed random variables, the variance due to these errors is given by \(2^{-24}/12\).
For the FIR filters with 7 taps the output is given by Eq. 6.5.

\[ y(n) = \sum_{k=1}^{7} h(k)x(n-k) . \]  

(6.5)

To implement the FIR filter, the six previous output samples are stored in registers. The previous samples and the current data sample multiplied by appropriate coefficients are added to compute each output sample. The subfilters are implemented by simple add and shift operations where every shift operation is hardwired. The architecture and implementation methodology for the subfilter for channel 1 is presented here. The architecture and implementation methodology for the other subfilters is identical to the one discussed.

The output for the subfilter for channel 1 corresponding to the direct form realization of the subfilter is given by Eq. 6.6 and is graphically represented in Fig. 6.2.

\[ y = (0 \times x) + (1 \times x1) + (-9 \times x2) + (46 \times x3) + (46 \times x4) + (-8 \times x5) + (0 \times x6) \]  

(6.6)

This equation is modified to result in powers-of-two multiplications. Power of two multiplications can be realized in dedicated hardware by simply shifting the data to the left or right by an appropriate number of bits. The resulting hardware is a small fraction of the hardware needed to implement a general purpose multiplier. Equation 6.7 shows the modified equation which results in powers-of-two multiplication.

\[ y = x1 + (8 \times -x2) + (-x2) + (32 \times x3) + (16 \times x3) + (2 \times -x3) + (32 \times x4) + (16 \times x4) + (2 \times -x4) + (8 \times -x5) \]  

(6.7)
Figure 6.2: Direct form realization for filter for channel 1
From Eq. 6.7 it is evident that all the multiplications can be performed by appropriate shifting of input signals. Thus, the operation of multiplication and accumulation for the FIR filter can be performed by shift and add operations. Equation 6.7 can be written as

\[
y = x_1 + (-x_2 \text{ shifted 3 left}) + (-x_2) + (x_3 \text{ shifted 5 left}) + (x_3 \text{ shifted 4 left}) \\
+ (-x_3 \text{ shifted 1 left}) + (x_4 \text{ shifted 5 left}) + (x_4 \text{ shifted 4 left}) \\
+ (-x_4 \text{ shifted 1 left}) + (-x_5 \text{ shifted 3 left})
\]  

(6.8)

---

**Figure 6.3:** Block diagram for the implementation of the polyphase subfilters.
The input samples are shifted to generate the partial products. The partial products are then added using carry save adders (CSAs) to generate the final output. Figure 6.3 shows the block diagram representation for the implementation of the polyphase subfilters.

The phase shifters which operate on the output of the polyphase subfilter, are implemented as complex multiplications by fractional numbers. The output of the polyphase subfilters are converted to binary fractional format and truncated to 12 bits to enable further processing by the phase shifter. Representing the output as a fixed-point fraction ensures that the product of two numbers remains a fraction and prevents overflow due to multiplication. Also, fixed word length can be maintained by truncating the least significant bits [27]. The binary fractional fixed format representation is depicted in Fig. 6.4.

---

*Example:*

\[ \langle 0.11000000000 \rangle_2 \text{ represents } \langle 0.75 \rangle_{10}, \text{ the decimal point is assumed} \]

\[ \langle 1.01000000000 \rangle_2 \text{ represents } \langle -0.75 \rangle_{10} \]

The first bit represents the sign bit. For a number \( x \) represented in this format the range is \( (-1 \leq x < 1) \)

*Figure 6.4: Binary fractional fixed format representation*
6.2.3 Comparison of Different Architectures for the Implementation of the Polyphase Subfilters

Two architectures for the polyphase subfilters were implemented on the Actel devices. The first implementation is discussed in Section 6.2.2. For the second implementation, Eq. 6.6. for channel 1 subfilter was modified to

\[ y = 46 \cdot (x3 + x4) + (-8) \cdot (x2 + x5) + 1 \cdot x1 + (-1) \cdot x2 \]  \hspace{1cm} (6.9)

The input data samples to be multiplied by the same coefficients were added and then add and shift multipliers were used to generate the partial products. The partial products were then added using CSA adders and combined in a final addition stage.

Comparison of the architectures showed that the implementation discussed in 6.2.2 used a lesser number of CLB's as compared to the implementation of Section 6.2.3. However the implementation of Section 6.2.3 was marginally faster than the implementation of section 6.6.2. Since both implementations were operational for the required speed, it was decided to use the implementation of Section 6.2.2 which resulted in the use of lower number of logic modules. Usage of lower number of logic modules enabled the subfilters and phase shifter for a channel to be implemented on a single device, thus reducing the numbers of devices required for the implementation of the group demultiplexer.
6.3 Phase Shifters

The phase shifting at the output of the subfilters is implemented as a multiplication by \( e^{-jn\pi/N} \) where \( n \) is the number ranging from 0 to \( N-1 \) and \( N \) is the number of channels in the group demultiplexer. The complex phase shifter operates on the output of the real and imaginary subfilter for a particular channel. The list of phase shifter multipliers for each of the channels is provided in Appendix B.

6.3.1 Implementation of the Phase Shifters

Twelve bit binary fractional fixed format is used in the implementation of the phase shifters as discussed in Section 6.2.2. The input to the phase shifter for each channel is the complex output of the polyphase subfilter for that channel. To maintain fixed word size the complex outputs from the phase shifters are truncated to 12 bits. Each complex multiplication requires four real multiplications. Truncation is used in the multiplier to limit the word size to 12 bits. The variance of truncation errors for each real multiplier is \( 2^{-24}/12 \), and for the complex multiplier it is \( 2^{-24}/3 \) [28].
6.4 Implementation Results for the Polyphase Subfilters and the Phase Shifters

The subfilters for the real and imaginary inputs with the corresponding phase shifter are implemented on a single RH1280 device and also on the A1280XL device. Each device has two FIR subfilters each with 7 taps and 7 bit coefficients for the taps and a phase shifter for that particular channel. The 8 channel filter bank of the group demultiplexer consists of 8 subfilters for the real inputs, 8 subfilters for the imaginary inputs and 8 phase shifters. A total of eight devices have been used to implement the polyphase filters and the phase shifters.

The results for the post-layout implementation on the RH1280 and A1280XL devices have been summarized in Tables 6.2a and 6.2b. The polyphase subfilters were tested at a system operating frequency of 24 MHz (41.6 ns clock period) and 3 MHz (332.8 ns clock period) for each subfilter. The functionality of the circuit was verified for the worst, typical and best case operating conditions through simulations. The delays are determined by the static timer tool from Actel. Static timing results show that the designs are capable of operating at a system clock rate of 38 MHz for the RH1280 device and 56 MHz for the A1280XL and the device operating internally at a clock rates of 4.75 MHz for the RH1280 device and 7 MHz for the A1280XL.

The device utilization (percentage of CLBs used) for both the RH1280 device and the A1280XL device is identical since they have the same number of CLBs. For ACT2 designs usually 100% of the CLB modules can be used without difficulty, but excessive
use of registers may make placement difficult [29]. For the polyphase subfilters and phase shifter implementation, designs with up to 99.19% device utilization were placed and routed successfully, and simulated to ensure proper functionality.

The maximum register to register delay is an indication of the maximum operating frequency for a design. In a data path architecture design, the maximum operating frequency is the inverse of the longest register to register delay. The register to register delay is defined here as the delay between two registers, from the clock edge at the first register to the input of the second register, including all combinatorial gate delays between the registers and the required setup time of the second register. For the designs, the A1280XL device is faster by 30% than the RH1280 device, for the worst and typical case operating conditions, and marginally slower by 3% for the best case conditions. This is in accordance with the statements made in Chapter 5.

The latency (number of clock cycles required for the first output to be available) through the polyphase subfilters is 24 system clock cycles.
Table 6.2a: Device utilization and delay for two subfilters (real and quadrature) with the phase shifter for that channel for channels 1, 2, 3 and 4

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Die</th>
<th>Device Utilization SEQUENTIAL</th>
<th>LOGIC SEQUENTIAL</th>
<th>Maximum Register to Register Delay in ns (frequency in MHz)</th>
<th>No. of IO's used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RH1280</td>
<td>218/624 (34.94%)</td>
<td>634/1232 (51.46%)</td>
<td>165.6 (6.03) 138.4 (7.22) 46.8 (21.36)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>117.3 (8.52) 98.7 (10.13) 48.6 (20.57)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RH1280</td>
<td>242/624 (38.78%)</td>
<td>1201/1232 (97.48%)</td>
<td>197.3 (5.06) 165.5 (6.04) 55.9 (17.88)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>141.5 (7.06) 118.9 (8.41) 58.9 (16.97)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RH1280</td>
<td>265/624 (42.47%)</td>
<td>1098/1232 (89.12%)</td>
<td>197.2 (5.07) 165.1 (6.05) 55.4 (18.05)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>137.8 (7.25) 116.3 (8.59) 57.2 (17.48)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RH1280</td>
<td>295/624 (47.28%)</td>
<td>1222/1232 (99.19%)</td>
<td>193.9 (5.15) 162.3 (6.16) 54.7 (18.28)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>139.1 (7.18) 117.1 (8.53) 57.9 (17.27)</td>
<td></td>
</tr>
</tbody>
</table>
Table 6.2b: Device utilization and delay for two subfilters (real and quadrature) with the phase shifter for that channel for channels 5, 6, 7 and 8

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Die</th>
<th>Device Utilization</th>
<th>Logic + Sequential</th>
<th>Maximum register to register delay in ns (frequency in MHz)</th>
<th>No of UO's used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sequential</td>
<td></td>
<td>worst case</td>
<td>typical case</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RH1280</td>
<td>235/624 (37.66%)</td>
<td>714/1232 (57.95%)</td>
<td>183.3 (5.45)</td>
<td>153.1 (6.53)</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>129.1 (7.74)</td>
<td>108.8 (9.19)</td>
</tr>
<tr>
<td>6</td>
<td>RH1280</td>
<td>236/624 (37.82%)</td>
<td>1163/1232 (94.40%)</td>
<td>191.4 (5.22)</td>
<td>160.3 (6.23)</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>136.1 (7.34)</td>
<td>115.0 (8.69)</td>
</tr>
<tr>
<td>7</td>
<td>RH1280</td>
<td>238/624 (38.14%)</td>
<td>1116/1232 (90.58%)</td>
<td>202.7 (4.93)</td>
<td>169.5 (5.89)</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>140.6 (7.11)</td>
<td>118.7 (8.42)</td>
</tr>
<tr>
<td>8</td>
<td>RH1280</td>
<td>225/624 (36.06%)</td>
<td>1209/1232 (98.13%)</td>
<td>192.8 (5.18)</td>
<td>161.7 (6.18)</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td></td>
<td></td>
<td>138.7 (7.20)</td>
<td>116.8 (8.56)</td>
</tr>
</tbody>
</table>
6.5 Fast Fourier Transform (FFT)

The FFT algorithm is an efficient method for computing the Discrete Fourier transform (DFT) of an input sequence when the size $N$ of the DFT is a power of 2. The DFT and the inverse DFT (IDFT) for an $N$-point sequence $x(n)$ are given by

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

(6.10)

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

(6.11)

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

6.5.1 Decimation in Time (DIT) FFT

The analysis given here is based on [28]. Let $x(n)$ be a sequence of data length $N$. The $N$-point DFT for $x(n)$ can be expressed as in Eq. 6.10. If we split the $N$-point 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 $x(n)$, we have

$$f_1(n) = x(2n)$$

(6.12)

$$f_2(n) = x(2n + 1), \quad n = 0, 1, \ldots, N/2 - 1$$

(6.13)

The $N$-point DFT of Eq. 6.10 can be expressed in terms of the DFT's of the
decimated sequences as follows

\[ X(k) = \sum_{n_{even}} x(k) W_N^{kn} + \sum_{n_{odd}} x(k) W_N^{kn} \]  

(6.14)

\[ X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)} \]  

(6.15)

But \( W_N^2 = W_{N/2} \), so substituting in Eq. 6.15

\[ X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km} \]  

(6.16)

\[ = F_1(k) + W_N^k F_2(k) , \quad k = 0,1, \ldots , N-1 \]  

(6.17)

where \( F_1(k) \) and \( F_2(k) \) are the \( N/2 \) point DFT's of the sequences \( f_1(m) \) and \( f_2(m) \).

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 \( W_N^k + N/2 = -W_N^k \). So

\[ X(k) = F_1(k) + W_N^k F_2(k) , \quad k = 0,1, \ldots , N/2-1 \]  

(6.18)

\[ X(k + N/2) = F_1(k) - W_N^k F_2(k) \]  

(6.19)

Let \( G_1(k) = F_1(k) \)

(6.20)

\[ G_2(k) = W_N^k F_2(k) , \quad k = 0,1, \ldots , N/2-1 \]  

(6.21)
Then the DFT $X(k)$ may be expressed as

$$X(k) = G_1(k) + G_2(k)$$ \hspace{1cm} (6.22)

$$X(k + N/2) = G_1(k) - G_2(k) , \hspace{1cm} k = 0, 1, \ldots, N/2 - 1$$ \hspace{1cm} (6.23)

The DIT can be repeated for each of the sequences $f_1(n)$ and $f_2(n)$. The decimation of the data sequences can be repeated again and again until the resulting sequences are reduced to one point sequences. For $N = 2^u$, this decimation can be performed $u = \log_2 N$ times.

The computation for the DIT algorithm for $N = 8$ is illustrated here. Figure 6.5 depicts the computation of $N = 8$ point DFT; the computation is performed in three stages, starting with the computations of four two-point DFT's, then two four point DFT's, and finally one eight point DFT.

The combination of the smaller DFTs to form the larger DFT is illustrated in Fig. 6.6 for $N = 8$. The base computation performed at each stage as illustrated in Fig. 6.6 is to take two complex numbers, say the pair $(a, b)$. multiply $b$ by $W_N^k$, and then add and subtract the product from $a$ to form two complex numbers $(A, B)$. This basic computation, which is shown in Fig. 6.7, is called a butterfly. Each butterfly involves one complex multiplication and two complex additions. The $W_N^k$ term in Fig. 6.7 of the butterfly is called the twiddle factor.
Figure 6.5: Block diagram for the 8 point radix-2 DIT FFT
Figure 6.6: Eight point DIT FFT algorithm
6.5.2 FFT Implementation

Twelve bit binary fixed fractional format and two's complement representation are used in the FFT. Use of fixed fractional format eliminates overflow due to multiplication as the product of the two numbers is less than either of the two numbers [27]. The addition of two complex numbers can however lead to overflow problems. To prevent overflow, the input sequence to the FFT needs to be scaled. The upper bound on $|X(k)|$ is

$$|X(k)| \leq \sum_{n=0}^{N-1} |x(n)|,$$

where $N$ is the dimension of the FFT \hfill (6.24)

For, two complement fixed fractional format the dynamic range is (-1, 1). Therefore, $|X(k)| \leq 1$ requires that

$$\sum_{n=0}^{N-1} |x(n)| < 1$$

(6.25)
For an FFT, there are two possible options to perform the scaling. In the first option, \( x(n) \) is initially scaled such that \( |x(n)| < 1 \) for all \( n \), then each point in the sequence can be scaled by \( N \) to ensure that Eq. 6.25 is satisfied. This scaling is extremely severe and in combination with the quantization errors gives a signal to noise ratio (SNR) proportional to \( N^2 \) [28]. In the second option the total scaling of \( I/N \) can be distributed among the stages of the FFT. By distributing the scaling of \( I/N \) uniformly throughout the FFT an SNR that is inversely proportional to \( I/N \) is obtained. The second option is used here to keep the SNR proportional to \( N \). The scaling by \( 1/8 \) is distributed throughout the FFT algorithm by scaling by \( 1/2 \) before each stage. The effects of the two methods of scaling is not very pronounced for an 8-point FFT, however, for a larger FFT it is advisable to use the second option for scaling.

In fixed-point arithmetic the addition of two numbers does not require any truncation, provided overflow is avoided. Multiplication, however, does require truncation. With an 8-point FFT, the multiplication in the first two stages is by 1 and \(-j\) which is trivial and does not require truncation. However, to maintain fixed word size, truncation is required in the third stage. The truncation errors occur in the final stages and range from \((-2^{-12} \text{ to } 0)\) for each real multiplication. The complex multipliers in the FFT use four real multipliers. The variance of truncation errors for each real multiplier is \( 2^{-24}/12 \) and for the complex multiplier it is \( 2^{-24}/3 \). Since these errors occur in the final stage they do not propagate
through the FFT. The multipliers are implemented by the add and shift method, where the shifts are hardwired. Simulation of the polyphase subfilters and phase shifters indicate that their output $|x(n)| < 0.82$. Furthermore, this condition ensures that there is no overflow in the complex butterflies in the final stage of the FFT.

DIT algorithm has been used for the implementation. For an 8-point FFT implementation both the DIT and the DIF (Decimation-in-frequency) algorithms use the same number of multipliers and adders. The interfacing requirements for both the algorithms, however, are different.

The flow diagram of the 8-point FFT corresponding to Fig. 6.6 but with scaling in each stage of the FFT is shown in Fig. 6.8. Because the scaling of 1/8 distributed throughout the FFT, the output of the FFT is 1/8 of its actual value. Referring to Fig. 6.8, it is observed that a total of 12 butterflies are used for the computation of the outputs. For the first stage of the FFT the butterflies are identical, a single butterfly is used for the first stage to perform all the operations, enabling reduction in number of logic modules. For the second stage two butterflies are used, and for the third stage since all the butterflies require different twiddle factors, four different butterflies are needed. Thus, by exploiting the throughput capabilities of the hardwired complex butterflies, it has been possible to reduce the overall number of butterflies in the actual implementation of the FFT.

Figure 6.9 depicts the actual implementation of the FFT where 7 butterflies instead of 12 are used. The FFT was implemented on four devices. Figure 6.10 shows the parti-
CHAPiER 6: Polyphase-FFT Group Demultiplexer Implementation

Figure 6.8: Flow diagram of the 8 point DIT FFT
Figure 6.9: Implementation of the radix-2, 8 point DIT FFT
Figure 6.10: Device partitioning for the FFT
tioning of the 8 point DIT FFT on four devices. Stages 1 and 2 are implemented on devices 1a and 1b and stage 3 is implemented on devices 2 and 3. Referring to Fig. 5.1, the outputs from the FFT are multiplied by \((-1)^n\).

6.5.3 FFT Implementation Results

The results for the implementation of the 8-point DIT FFT are summarized in Table 6.3, which depicts the device utilization and the maximum operating frequency for the implementation of the FFT on the RH1280 and A1280 devices. The frequency of operation required for the first stage is 1/2 the system clock frequency, for the second stage it is 1/4 system clock frequency and for the third stage it is 1/8 the system clock frequency. Consistent with these requirements, each input data sample is available for two clock cycles.

Similar to the polyphase subfilters, the functionality of the FFT was tested through simulations under worst, typical and best operating conditions with a system clock of 24 MHz. However, as indicated by the Actel static timing analysis, the FFT processor is capable of operating at system clock 38 MHz for the RH1280 devices and 53 MHz for the A1280XL devices for the worst case operating conditions. The latency through the FFT is 34 system clock cycles.
Table 6.3: Device utilization and delay for the 8-point, 12 bits input/output (complex) DIT-FFT

<table>
<thead>
<tr>
<th>Device #</th>
<th>Die</th>
<th>Device Utilization</th>
<th>Maximum register to register DELAY in ns (frequency in MHz)</th>
<th>No. of IOS used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>sequential logic</td>
<td>Worst case</td>
<td>Typical case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sequential</td>
<td>stage 1</td>
<td>stage 2</td>
</tr>
<tr>
<td>1a</td>
<td>RH1280</td>
<td>394/624 (63.14%)</td>
<td>40.9</td>
<td>77.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>606/1232 (49.19%)</td>
<td>(24.44)</td>
<td>(12.83)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30.0</td>
<td>52.9</td>
<td>25.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(33.33)</td>
<td>(18.90)</td>
<td>(39.52)</td>
</tr>
<tr>
<td>1b</td>
<td>RH1280</td>
<td>394/624 (63.14%)</td>
<td>47.6</td>
<td>76.5</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>606/1232 (49.19%)</td>
<td>(21.00)</td>
<td>(13.07)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31.4</td>
<td>54.3</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(31.84)</td>
<td>(18.41)</td>
<td>(37.87)</td>
</tr>
<tr>
<td>2</td>
<td>RH1280</td>
<td>448/624 (71.79%)</td>
<td>203.2</td>
<td>169.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>1205/1232 (97.81%)</td>
<td>(4.92)</td>
<td>(5.88)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.6</td>
<td>121.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6.92)</td>
<td>(8.21)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RH1280</td>
<td>448/624 (71.79%)</td>
<td>208.2</td>
<td>174.5</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>1205/1232 (97.81%)</td>
<td>(4.80)</td>
<td>(5.73)</td>
</tr>
</tbody>
</table>
6.6 Rate Conversion Filter

The output from the demultiplexer is at the rate $f_c$ and needs to be converted to the symbol rate $f_b$ in order to be processed by the demodulator. This function is performed by the rate conversion filter. The rate conversion filter implemented here is a representative filter. In an actual implementation, an adaptive rate control filter is used to adjust the timing sample sequence output by the demultiplexer and convert it to a baud rate sequence.

6.6.1 Rate Conversion Filter Coefficients

The coefficients for the rate conversion filter were derived using the Matlab based software developed at the CRC [18]. The design method is the same as that used for the polyphase subfilters. The parameters chosen for the rate conversion filter are listed in Table 6.4. A list of the filter coefficients is provided in the Appendix B.

6.6.2 Implementation of the Rate Conversion Filter

The rate conversion filter is implemented as an FIR filter with 9 taps and 7 bits of quantization for the filter coefficients. The filter coefficients are represented as integers and are in 2's complement format. The input to the rate conversion filter is 12 bits represented in binary fractional 2's complement format and the output is truncated to 12 bits.
Table 6.4: Parameters for the rate conversion filters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized frequency spacing (F_s)</td>
<td>1.5</td>
</tr>
<tr>
<td>Rolloff factor for the raised cosine filter</td>
<td>0.4</td>
</tr>
<tr>
<td>Window used</td>
<td>Kaiser</td>
</tr>
<tr>
<td>Beta factor for the kaiser window</td>
<td>2</td>
</tr>
<tr>
<td>Number of taps per filter</td>
<td>9</td>
</tr>
<tr>
<td>Number of bits per filter tap</td>
<td>7</td>
</tr>
</tbody>
</table>

and is also in binary fractional 2’s complement format. The truncation errors have a variance of $2^{-24}/12$. The inputs are complex and the filter coefficients are real, hence two filters are needed to generate the complex output.

The rate conversion filter is symmetric, and hence it is implemented by grouping the inputs which are multiplied by the same coefficients and then appropriately shifting them to form the partial products which are added by CSA adders. A final stage of addition is needed to obtain the output. Figures 6.11 and 6.12 represent the implementation of the rate conversion filter. Figure 6.11 gives the direct form realization of the filter which can be represented as in Eq. 6.26

$$
\begin{align*}
    y &= (-2 \times x) + (2 \times x_1) + (-8 \times x_2) + (12 \times x_3) + (63 \times x_4) \\
    &+ (12 \times x_5) + (-8 \times x_6) + (2 \times x_7) + (-2 \times x_8)
\end{align*}
$$  \hspace{1cm} (6.26)
Figure 6.11: Direct form realization of the rate conversion filter
which can be modified to

\[
y = (-2 \times (x + x^8)) + (2 \times (x_1 + x^7)) + (-8 \times (x^2 + x^6)) \\
+ ((12 \times (x^3 + x^5)) + (63 \times x^4))
\]

This then results in the implementation given by Eq. 6.28

\[
y = (-x + x^8) \text{ shifted 1 left } + (x_1 + x^7) \text{ shifted 1 left } + (-x^2 + x^6) \text{ shifted 3 left } \\
+ (x^3 + x^5) \text{ shifted 3 left } + (x^3 + x^5) \text{ shifted 2 left } + (x^4 \text{ shifted 6 left }) \\
+ (-x^4)
\]

(6.28)

---

Partial products generated by shifting the sums

Addition of partial products using Carry Save Adders

Final adder

Output

Figure 6.12: Block diagram for the implementation of the rate conversion filter
6.6.3 Rate Conversion Filter Implementation Results

The results for the implementation of the rate conversion filter in terms of the device utilization and the speed of operation are presented in Table 6.5. Each channel uses two rate conversion filters, one for the real input and another for the imaginary input. Hence, the total number of rate conversion filters used is 16 which were implemented on eight ACT2 devices. The functionality of the rate conversion filter is tested at 24 MHz system clock and 3 MHz for the filter. However, the devices are capable of operating at system clock rates of 38 MHz for the RH1280 and 56 MHz for the A1280XL device.

The rate conversion filters for all the channels are similar. However, a different decode circuit is used for each of the channels to correctly route in the data samples. This has resulted in slight variations in the device utilization. The latency through the rate conversion filter is 12 system clock cycles.

<table>
<thead>
<tr>
<th>Die</th>
<th>Device utilization</th>
<th>Maximum register to register delay in ns</th>
<th>No. of I/Os used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SEQUENTIAL</td>
<td>(frequency) MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOGIC</td>
<td>worst case</td>
<td>typical case</td>
</tr>
<tr>
<td>RH1280</td>
<td>328/624 (52.56%)</td>
<td>206.3 (4.85)</td>
<td>173.0 (5.78)</td>
</tr>
<tr>
<td></td>
<td>1018/1232 (82.63%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1280XL</td>
<td>140.6 (7.11)</td>
<td>118.6 (8.43)</td>
<td>58.0 (17.24)</td>
</tr>
</tbody>
</table>
6.7 Conclusion

This chapter discussed in detail the implementation of the building blocks of the group demultiplexer. Results for each implementation were also presented. From the results it can be deduced that it is feasible to implement the eight channel group demultiplexer using a reasonable number of FPGA devices and at the required system operating frequency.

The results of the static timer tool from Actel indicate that the group demultiplexer is operational for a system clock of 38 MHz for the RH1280 and 53 MHz for the A1280XL devices. Higher throughputs can be achieved by utilizing arithmetic cells whose architectures inherently provide better performance through the use of carry look ahead circuitry and pipelining. Also, identifying the critical slow nets and rerouting the design to ensure minimum routing delays in the critical paths will result in higher throughputs. However, such changes can only be achieved at the cost of higher gate counts and possibly a larger number of devices.
CHAPTER 7

Summary and Suggested Further Research and Development

An 8-channel, T1 rate, group demultiplexer was implemented on the radiation-hardened Actel FPGAs. It is the first time reported in the literature, that an entire subsystem such as the group demultiplexer has been implemented on radiation-hardened Actel FPGAs. The design is fully operational for T1 data rates and can also operate at E1 data rates. The work presented in this thesis is in the framework of the research being carried out at the CRC and CITR, on OBP technologies.

Architectures for the building blocks of the polyphase-FFT group demultiplexer: polyphase subfilters and the FFT processor were optimized and implemented on the Actel
ACT2 devices. For DSP applications the speed of the arithmetic module is of prime importance. Various architectures for the arithmetic modules, which form an integral part of the building blocks of the demultiplexer, were considered. To limit the implementation of the group demultiplexer to a minimum number of devices, a trade-off between the speed of operation and the gate count was made. Working with the desired speed, an implementation which results in the minimum number of logic modules was used. For multiplication, the add and shift multiplier was found to be the most efficient. The choice of adders depended on their desired speed of operation. Wherever possible, low speed ripple adders were used to keep the number of logic modules to a minimum.

The polyphase-FFT group demultiplexer with 8 bit input resolution and 12 bit output resolution was implemented on twelve Actel RH1280 and A1280XL devices and the rate conversion filters on eight devices. The circuit was tested for functionality under worst, typical and best case operating conditions using Quicksim simulator tool from Mentor Graphics. The maximum frequency of operation and the device utilization for the devices were computed using Actel tools. The input and output delay information necessary for proper interfacing between the devices was also provided. Static timer analysis indicates that the group demultiplexer is operational for a system clock of up to 38 MHz for RH1280 devices and 53 MHz for the A1280XL devices. The present design can operate with channel separations of up to 4.75 MHz for RH1280 devices and 6.6 MHz for the A1280XL devices, which indicates that the design is also capable of operating at E1 rates.
The designs were mapped on to the A1280XL devices. The results from the testing of the devices indicate that the current consumption per device is approximately 27 mA at 3.3 V and 43 mA at 5 V for the polyphase subfilters and phase shifters. Also, if multichip modules (MCMs) are used, the power consumption will drop by approximately 15%.

7.1 Further Research and Development

In this thesis an 8-channel group demultiplexer was implemented on Actel RH1280 and A1280XL FPGAs. In the future it may be necessary to implement a group demultiplexer with a larger number of channels. The increase in the number of channels will require an extension of the current design and some modifications to it. The polyphase network can be modified to accommodate a larger number of channels by using extra devices for the channels which are added. Also, using the address to route the input data to the polyphase subfilters from the A/D converter makes it possible to extend this architecture to incorporate larger number of channels. The number of devices required for the implementation will increase as the number of channels increase.

Increasing the number of channels will require an FFT processor with a larger number of points, since the size of the FFT processor is proportional to the numbers of channels in the group demultiplexer. As the order of the FFT becomes larger, the twiddle factors for the butterflies will becomes more complex, thus placing a lot of emphasis on the design of the multipliers. Pipelining within the multiplier to ensure desired speed of
operation would be a viable solution. An interesting option or possibility would be to combine 8-point FFT's to yield a larger FFT. For larger FFTs, quantization errors introduced in the initial stages propagate to the final butterflies; hence their effects become more severe. Techniques to keep the usage of CLB's as low as possible will have to be considered.

Much work needs to be done on the implementation of the adaptive rate conversion filters. The present implementation is a representative filter indicating the complexity of the filter coefficients. In an actual implementation, a set of filters needs to be stored and decisions made by the timing and recovery algorithm will indicate the filter to be used.

The present implementation was on the Actel RH1280 and A1280XL device. The results of the work presented here indicate the potential and the limitations of implementing DSP hardware, such as the group demultiplexer, with the currently available 8k radiation-hardened devices. In view of the fact that twelve devices were required to achieve this, it is clear that a greater level of integration would be desirable. In the near future rad-hard FPGA's with 20k gates are to be introduced. Importing the design onto larger devices will enable the use of lesser number of devices to implement the same design, reducing considerably the interfacing requirements between the devices. As an example, the entire 8-point DIT FFT can be implemented on a single Actel A32400DX device with 40k gates. The viability of the FPGA technology will improve as the levels of integration are enhanced.
References


Appendix A

Table A1: Timing information for the input and output for the devices p1 to p8

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Die</th>
<th>Input delay (ns)</th>
<th>Output delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>worst</td>
<td>typical</td>
</tr>
<tr>
<td>1</td>
<td>RH1280</td>
<td>12.6</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.1</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>RH1280</td>
<td>9.5</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.4</td>
<td>7.0</td>
</tr>
<tr>
<td>3</td>
<td>RH1280</td>
<td>9.5</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.1</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>RH1280</td>
<td>9.6</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.7</td>
<td>6.5</td>
</tr>
<tr>
<td>5</td>
<td>RH1280</td>
<td>9.5</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.4</td>
<td>6.2</td>
</tr>
<tr>
<td>6</td>
<td>RH1280</td>
<td>9.5</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.4</td>
<td>7.0</td>
</tr>
<tr>
<td>7</td>
<td>RH1280</td>
<td>10.8</td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.7</td>
<td>6.5</td>
</tr>
<tr>
<td>8</td>
<td>RH1280</td>
<td>9.7</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.4</td>
<td>7.0</td>
</tr>
</tbody>
</table>

(1) Macro INBUF (input buffer)
(2) Macro TBHS (tristate output buffer)
Table A2: Timing information for the input and output for the FFT devices f1a, f1b, f2 and f3

<table>
<thead>
<tr>
<th>Device #</th>
<th>Device</th>
<th>Input delay (ns)(^{(1)})</th>
<th>Output delay (ns)(^{(2)})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>worst</td>
<td>typical</td>
</tr>
<tr>
<td>1a</td>
<td>RH1280</td>
<td>10.4</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.5</td>
<td>6.3</td>
</tr>
<tr>
<td>1b</td>
<td>RH1280</td>
<td>10.7</td>
<td>8.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>6.7</td>
<td>5.7</td>
</tr>
<tr>
<td>2</td>
<td>RH1280</td>
<td>9.9</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.7</td>
<td>6.5</td>
</tr>
<tr>
<td>3</td>
<td>RH1280</td>
<td>9.7</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.7</td>
<td>6.5</td>
</tr>
</tbody>
</table>

(1) Macro INBUF (input buffer)
(2) Macro OUTBUF (output buffer)
Table A3: Timing information for the input and output for the eight devices of the rate conversion filters

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Device</th>
<th>Input delay (ns)</th>
<th>Output delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>worst typical best case</td>
<td>worst typical best case</td>
</tr>
<tr>
<td>1</td>
<td>RH1280</td>
<td>14.5 12.1 4.1</td>
<td>20.9 17.5 5.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.3  6.9  3.3</td>
<td>13.5 11.5 5.7</td>
</tr>
<tr>
<td>2</td>
<td>RH1280</td>
<td>9.7   8.1  2.7</td>
<td>24.5 20.6 7.0</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.1   6.8  3.3</td>
<td>15.5 13.1 6.5</td>
</tr>
<tr>
<td>3</td>
<td>RH1280</td>
<td>10.1  8.5  2.8</td>
<td>21.7 18.2 6.1</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>7.7   6.6  3.1</td>
<td>13.7 11.6 5.7</td>
</tr>
<tr>
<td>4</td>
<td>RH1280</td>
<td>9.6   8.0  2.7</td>
<td>21.6 18.1 6.1</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.0   6.7  3.2</td>
<td>13.9 11.8 5.9</td>
</tr>
<tr>
<td>5</td>
<td>RH1280</td>
<td>9.3   7.8  2.6</td>
<td>20.6 17.3 5.8</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.3   6.9  3.3</td>
<td>13.0 11.0 5.5</td>
</tr>
<tr>
<td>6</td>
<td>RH1280</td>
<td>9.6   8.0  2.7</td>
<td>21.6 18.1 6.1</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>8.0   6.7  3.2</td>
<td>13.9 11.8 5.9</td>
</tr>
<tr>
<td>7</td>
<td>RH1280</td>
<td>10.3  8.6  2.9</td>
<td>22.1 18.5 6.3</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>9.7   8.1  3.9</td>
<td>14.4 12.2 6.1</td>
</tr>
<tr>
<td>8</td>
<td>RH1280</td>
<td>10.4  8.7  2.9</td>
<td>20.9 17.5 5.9</td>
</tr>
<tr>
<td></td>
<td>A1280XL</td>
<td>9.7   8.1  3.9</td>
<td>13.6 11.5 5.7</td>
</tr>
</tbody>
</table>

(1) Macro INBUF (input buffer)
(2) Macro OUTBUF (output buffer)
### Appendix B

#### Table B1: Coefficients for the Polyphase Filter

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>p1</td>
<td>-9</td>
<td>-4</td>
<td>-3</td>
<td>-7</td>
<td>-5</td>
<td>-12</td>
<td>-20</td>
<td>-26</td>
<td>-28</td>
<td>-22</td>
<td>-14</td>
</tr>
<tr>
<td>p2</td>
<td>-10</td>
<td>-8</td>
<td>-5</td>
<td>-9</td>
<td>-6</td>
<td>-20</td>
<td>-28</td>
<td>-36</td>
<td>-42</td>
<td>-30</td>
<td>-20</td>
</tr>
<tr>
<td>p3</td>
<td>18</td>
<td>-6</td>
<td>26</td>
<td>34</td>
<td>-10</td>
<td>42</td>
<td>53</td>
<td>46</td>
<td>41</td>
<td>44</td>
<td>52</td>
</tr>
<tr>
<td>p4</td>
<td>21</td>
<td>36</td>
<td>62</td>
<td>38</td>
<td>45</td>
<td>63</td>
<td>29</td>
<td>35</td>
<td>20</td>
<td>44</td>
<td>49</td>
</tr>
<tr>
<td>p5</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>7</td>
<td>12</td>
<td>37</td>
<td>37</td>
<td>12</td>
<td>37</td>
<td>37</td>
<td>45</td>
</tr>
<tr>
<td>p6</td>
<td>15</td>
<td>20</td>
<td>30</td>
<td>30</td>
<td>23</td>
<td>29</td>
<td>31</td>
<td>39</td>
<td>39</td>
<td>54</td>
<td>52</td>
</tr>
<tr>
<td>p7</td>
<td>16</td>
<td>24</td>
<td>24</td>
<td>32</td>
<td>48</td>
<td>60</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
</tbody>
</table>

115
Table B2: List of phase shifter multipliers

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Multiplier</th>
<th>((a + jb) \times \text{multiplier})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(e^{0j\pi/8})</td>
<td>((a+jb))</td>
</tr>
<tr>
<td>2</td>
<td>(e^{-j\pi/8})</td>
<td>((\cos \pi/8 + j\sin \pi/8) + j(b\cos \pi/8 - \sin \pi/8))</td>
</tr>
<tr>
<td>3</td>
<td>(e^{-j2\pi/8})</td>
<td>((\cos \pi/4 + j\sin \pi/4) + j(b\cos \pi/4 - \sin \pi/4))</td>
</tr>
<tr>
<td>4</td>
<td>(e^{-j3\pi/8})</td>
<td>((\sin \pi/8 + b\cos \pi/8) + j(b\sin \pi/8 - a\cos \pi/8))</td>
</tr>
<tr>
<td>5</td>
<td>(e^{-j\pi/2})</td>
<td>((b-j)a)</td>
</tr>
<tr>
<td>6</td>
<td>(e^{-j5\pi/8})</td>
<td>((-\sin \pi/8 + b\cos \pi/8) + j(-b\sin \pi/8 - a\cos \pi/8))</td>
</tr>
<tr>
<td>7</td>
<td>(e^{-j6\pi/8})</td>
<td>((-\cos \pi/4 + j\sin \pi/4) + j(-b\cos \pi/4 - a\sin \pi/4))</td>
</tr>
<tr>
<td>8</td>
<td>(e^{-j7\pi/8})</td>
<td>((-\cos \pi/8 + b\sin \pi/8) + j(-b\cos \pi/8 - a\sin \pi/8))</td>
</tr>
</tbody>
</table>

where \(\cos3\pi/8 = \sin\pi/8\), \(\sin3\pi/8 = \cos\pi/8\)

\(\cos5\pi/8 = -\sin\pi/8\), \(\sin5\pi/8 = \cos\pi/8\)

\(\cos3\pi/4 = -\cos\pi/4\), \(\sin3\pi/4 = \sin\pi/4\)

\(\cos7\pi/8 = -\cos\pi/8\), \(\sin7\pi/8 = \sin\pi/8\)
Table B3: Coefficients for the rate conversion filter

<table>
<thead>
<tr>
<th>Tap no.</th>
<th>Coeff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>-8</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>-8</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>-2</td>
</tr>
</tbody>
</table>
Appendix C

Raised-Cosine Filter

The raised-cosine filter has a frequency response $H_{RC}(\omega)$ given by

$$
H_{RC}(\omega) = \begin{cases} 
T & . \quad 0 \leq |\omega| \leq (1 - \beta)W \\
\frac{T}{2} \left[ 1 - \sin \left( \frac{\pi}{2\beta W} (|\omega| - W) \right) \right] & . \quad (1 - \beta)W \leq |\omega| \leq (1 + \beta)W \\
0 & . \quad |\omega| \geq (1 + \beta)W
\end{cases}
$$

(C.1)

where $\omega$ is the frequency in rad/s, $T$ is the baud period and $W = \pi/T$ is the minimum Nyquist bandwidth. The parameter $\beta$ is called the roll-off factor which is the excess bandwidth divided by the minimum Nyquist bandwidth. Figure C.1 shows the frequency response of the raised-cosine filter for different values of $\beta$. For $\beta = 0$, the filter characteristic corresponds to the ideal low-pass filter. For $\beta = 1$, the response is known as full-cosine roll-off characteristic which has bandwidth equal to twice the Nyquist bandwidth.

The impulse response of the raised-cosine filter is

$$
h_{RC}(t) = \left( \frac{\sin Wt}{Wt} \right) \left( \frac{\cos \beta Wt}{1 - (2\beta Wt / \pi)^2} \right) 
$$

(C.2)
which is depicted in Fig. C.2. From Fig. C.2, it is clear that the amplitudes of the oscillatory tails of $h_{RC}(t)$ are smaller when $\beta$ is large. This results in lower intersymbol interference, but higher excess bandwidth.

Figure C.1: Frequency response of raised-cosine filter for different roll-off factors

Figure C.2: Impulse response of raised-cosine filter for different roll-off factors