DESIGN OF MULTI-MODULATION BASEBAND MODULATOR AND DEMODULATOR FOR SOFTWARE DEFINED RADIO

CHYE YIN HUI

UNIVERSITI SAINS MALAYSIA

2011

DESIGN OF MULTI-MODULATION BASEBAND MODULATOR AND DEMODULATOR FOR SOFTWARE DEFINED RADIO

By

CHYE YIN HUI

**Thesis submitted in fulfillment of the requirements
for the degree of
Master of Science**

DECEMBER 2011

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

UNIVERSITI SAINS MALAYSIA

ACKNOWLEDGEMENTS

There are lots of people whom I wish to thank for contributing to the completion of this thesis. First of all, my greatest gratitude goes to my supervisor, Associate Professor Dr. Mohd Fadzil Ain, at School of Electrical and Electronic Engineering, USM, for his valuable advices and suggestions, continuous supports and encouragement in helping me solve all the problems during this research.

I would like to thank Dean, Professor Dr. Mohd Zaid Abdullah; Deputy Dean of Postgraduate and Research, Associate Professor Dr. Kamal Zuhairi Zamli; and all the staffs and technicians at School of Electrical and Electronic Engineering, USM, for providing supports, good services and environment during my research in laboratories.

Special thanks to PhD student, Mr. Majid Salal Naghmash, who is working on IF processing of SDR for frequently exchanging points of view with me that always result in useful inspirations to my works.

I would also like to thank to Mr. Mazlaini Yahaya from TMRD (Telekom Malaysia Research and Development) for giving precious guidance and information related to Xilinx FPGA implementation.

My appreciation also goes to my friends and family for their emotional caring and supports in helping me relieve the stress and restore self-confidence to face and overcome all the difficulties throughout the period of this research.

The work in this thesis was supported in part by Malaysia Communication and Multimedia Commission (MCMC) SCIENCE FUND grant and USM Fellowship.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS ii

TABLE OF CONTENTS iii

LIST OF TABLES vii

LIST OF FIGURES ix

LIST OF ABBREVIATIONS xii

LIST OF SYMBOLS xvii

ABSTRAK xxiv

ABSTRACT xxv

CHAPTER 1 INTRODUCTION 1

1.1 General Overview 1

1.2 Problem Statements 3

1.3 Objectives 4

1.4 Scope of Works 5

1.5 Contribution of Research 6

1.6 Thesis Outline 7

CHAPTER 2 LITERATURE REVIEW 9

2.1 Introduction 9

2.2 Software Defined Radio (SDR) Overview 10

2.2.1 Functional Architecture of SDR 11

2.2.2 Baseband Processing Segment in SDR 12

2.2.3 Suitability of FPGA as Implementation Platform of SDR 13

2.3 Digital Linear Modulations 15

2.3.1 Pulse Amplitude Modulation (PAM) 16

2.3.2 Phase Shift Keying (PSK) 18

2.3.3 Quadrature Amplitude Modulation (QAM) 22

2.4 Digital Pulse Shaping Filter (PSF) 24

2.4.1 Raised Cosine (RC) Pulse Shaping Filter (PSF) 24

2.4.2 Digital Implementation of RC Filter 29

2.5 Symbol Timing Recovery (STR) 30

2.5.1 Schemes of STR 30

2.5.2 Configurable STR 33

CHAPTER 3 DESIGN CONCEPT 35

3.1 Introduction 35

3.2 Finite Impulse Response (FIR) Filter 35

3.2.1 Odd-Tap Symmetric Direct Form (DF) FIR Filter 37

3.2.2 Odd-Tap Symmetric Systolic FIR Filter 39

3.3 Sample Rate Conversion (SRC) 42

3.3.1 Interpolation Filter 43

3.3.2 Polyphase Interpolator 45

3.4 Gardner Timing Error Detector (TED) 47

3.5 Pseudo-Random (PR) Binary Number Generator 49

CHAPTER 4 METHODOLOGY 51

4.1 Introduction 51

4.2 Installation of Software Tools 53

4.3 High Level DSP Modeling 54

4.3.1 Signal Delay 56

4.3.2 Multi-Modulation Baseband Modulator (MMBM) 59

4.3.2.1 Auxiliary Blocks 60

4.3.2.2 Input and Output Blocks 60

4.3.2.3 PR Bit Generator Subsystem 62

4.3.2.4 Symbol Mapper Subsystem 63

4.3.2.5 Interpolation Filter Subsystem 64

4.3.2.5.1 2xMA FIR Multi Interpolator Subsystem 65

4.3.3 Multi-Modulation Baseband Demodulator (MMBD) 67

4.3.3.1 Auxiliary Blocks 68

4.3.3.2 Input and Output Blocks 68

4.3.3.3 RRC Filter Subsystem 70

4.3.3.3.1 Multi Odd-Tap Systolic FIR Filter Subsystem 72

4.3.3.3.2 Reset Delay Subsystem 73

4.3.3.4 Timing Recovery Subsystem 73

4.3.3.4.1 TED Subsystem 74

4.3.3.4.2 Search Min Error Subsystem 76

4.3.3.4.3 Data and Timing Updates Subsystem 77

4.3.3.4.4 Data Capture Subsystem 78

4.3.3.5 Symbol Demapper Subsystem 79

4.3.3.6 Pulse Shaping Subsystem 80

4.4 Compilation of HDL Netlists of DSP Models 82

4.5 HDL Design of Setup Configuration Module 84

4.5.1 Configurations of ADC and DAC 84

4.5.2 Configuration of Clock Synthesizer 87

4.6 HDL Integration of DSP Models with Setup Configuration Module..........................88

4.7 Synthesis of HDL Integrated Modules 89

4.8 FPGA Implementation of Synthesized Integrated Modules 89

4.9 Hardware Implementation and Verification Using FPGA, ADC and DAC 91

CHAPTER 5 RESULTS AND DISCUSSIONS 94

5.1 Introduction 94

5.2 Simulation Results in Xilinx System Generator/Simulink Environment 94

5.2.1 Top Level MMBM 95

5.2.2 Top Level MMBD...97

5.2.3 Ideal MMBMD 100

5.2.4 Unmapped FPGA Utilization...103

5.3 Simulation Results in ModelSim Environment....105

5.3.1 HDL Simulations of MMBM and MMBD Netlists....105

5.3.2 HDL Simulation of Setup Configuration Module....112

5.3.3 HDL Simulation of Integrated Modules....114

5.4 Synthesis Results in Synplify Pro Environment....120

5.5 Reports Generated in Xilinx ISE Environment....121

5.5.1 FPGA Device Utilization....121

5.5.2 Post PAR Static Timing Report....124

5.5.3 Power Estimation Report....126

5.6 Hardware Outputs during Program Running in FPGA....127

5.6.1 Baseband Modulated Signals....127

5.6.2 Smoothed Bit....129

5.7 Efficiency Evaluation of FPGA Utilization....131

5.7.1 Symbol Mapper and Demapper....131

5.7.2 Pulse Shaping Filter (PSF) and Matched Filter (MF)....133

5.7.3 Symbol Timing Recovery (STR)....135

5.7.4 Baseband Modulator (BM) and Demodulator (BD)....139

CHAPTER 6 CONCLUSION AND FUTURE WORKS....141

6.1 Conclusion....141

6.2 Future Works....143

REFERENCES....144

LIST OF PUBLICATIONS....149

APPENDIX A In Depth DSP Models and Parameter Settings....A1

APPENDIX B Determination of Optimum Binary Points for DAC and ADC....A66

APPENDIX C Serial Programming Interface (SPI) Timing....A74

APPENDIX D Verilog Codes of Setup Configuration....A75

APPENDIX E Verilog Codes of Integrated Module and Test-Fixture....A81

LIST OF TABLES

Page

Table 1.1: Wireless Standards with Modulations and Data Rates (Rappaport, 2002) 4

Table 4.1: Selection Values for Digital Modulation Scheme 55

Table 4.2: Delay Samples for Input Message Bit 57

Table 4.3: Delay Samples for Symbol Integer..........58

Table 4.4: Delay Samples for I/Q Symbols 58

Table 4.5: Latency of *MMBM*..........60

Table 4.6: Latency of *Symbol Mapper* Subsystem..........64

Table 4.7: Design Parameters of Multi-Rate Interpolation Filter (MRIF)..........65

Table 4.8: Design Parameters of Polyphase Interpolator of MRIF..........66

Table 4.9: Latency of *MMBD* 68

Table 4.10: Latency of *RRC Filter* Subsystem 71

Table 4.11: Design Parameters of MOSSF..........72

Table 4.12: Latency of *Timing Recovery* Subsystem 74

Table 4.13: Latency of *Symbol Demapper* Subsystem..........80

Table 4.14: Design Parameters of RC Interpolation Filter (RCIF) 81

Table 4.15: Required Files Generated by *System Generator* Block..........83

Table 4.16: Required Pins of ADS5500 ADC (Avnet, 2006; Texas Instruments, 2007)..........85

Table 4.17: Required Pins of DAC5687 DAC (Avnet, 2006; Texas Instruments, 2005) 85

Table 4.18: SPI Codes for ADS5500 ADC (Texas Instruments, 2007) 86

Table 4.19: SPI Codes for DAC5687 DAC (Texas Instruments, 2005)..........86

Table 4.20: Required Pins of ICS8442 Clock Synthesizer (Avnet Memec, 2005) 87

Table 5.1: Unmapped FPGA Utilization of MMBM 103

Table 5.2: Unmapped FPGA Utilization of MMBD..........103

Table 5.3: Additional Files Required by Test-bench File..........105

Table 5.4: Simulation Timing of MMBM in ModelSim vs. Sys Gen 110

Table 5.5: Simulation Timing of MMBD in ModelSim vs. Sys Gen..........111

Table 5.6: ADC/DAC Timing of Setup Configuration in ModelSim 113

Table 5.7: Timing of Integrated Modules in ModelSim..119

Table 5.8: Estimated Timing Report of Synthesis 120

Table 5.9: Device Utilization Summary of Integrated Module of MMBM.......................121

Table 5.10: Device Utilization Summary of Integrated Module of MMBD 122

Table 5.11: FPGA Utilization of Configurable MMBM..123

Table 5.12: FPGA Utilization of Configurable MMBD..123

Table 5.13: Post-PAR Static Timing Report for Integrated Module of MMBM.................125

Table 5.14: Post-PAR Static Timing Report for Integrated Module of MMBD 125

Table 5.15: Estimated Power Report for Integrated Modules 126

Table 5.16: Comparison of Real-time and Simulation Results for MMBM........................128

Table 5.17: Comparison of Real-time and Simulation Results for MMBD 130

Table 5.18: Unmapped FPGA Utilization of Symbol Mapper 131

Table 5.19: Unmapped FPGA Utilization of Symbol Demapper 132

Table 5.20: Unmapped FPGA Utilization of PSF...133

Table 5.21: Unmapped FPGA Utilization of MF 133

Table 5.22: Unmapped FPGA Utilization of STR..136

Table 5.23: FPGA Utilizations of Proposed STR vs. Existing STR...................................137

Table 5.24: Unmapped FPGA Utilization of BM 139

Table 5.25: Unmapped FPGA Utilization of BD..139

LIST OF FIGURES

Page

Figure 2.1: Functional Architecture of SDR (Mitola, 1995) 11

Figure 2.2: Simple BPSK Modem (Ahamed and Scarpino, 2005) 19

Figure 2.3: Simple QPSK Modem (Song and Yao, 2010)....20

Figure 2.4: Spectrum of Rectangular Symbol Pulse (Gentile, 2002) 25

Figure 2.5: Frequency Responses of RC Filter and Rectangular Pulse (Gentile, 2002)....26

Figure 2.6: Time Responses of RC Filter and Rectangular Pulse (Gentile, 2002)....27

Figure 2.7: Architectures of Feedback STR. A: Synchronous (Sciagura et al., 2007); B: Asynchronous (Jian et al., 2005). 31

Figure 2.8: Architecture of Feedforward STR (Zhu et al., 2005) 32

Figure 2.7: Configurable Multi-Symbol-Rate STR (Tachwali et al., 2009)....34

Figure 3.1: Direct Form (DF) FIR Filter Structure (Oppenheim et al., 1999)....36

Figure 3.2: Odd-Tap Symmetric FIR Filter Structure (Oppenheim et al., 1999)....38

Figure 3.3: Odd-Tap Transposed Symmetric FIR Filter Structure 39

Figure 3.4: Odd-Tap Symmetric Systolic FIR Filter Structure (adapted from Xilinx, 2008)....39

Figure 3.5: Interpolation (Hentschel, 2002)....43

Figure 3.6: Polyphase Interpolator (adapted from Hentschel, 2002) 46

Figure 3.7: Gardner TED (Sciagura et al., 2007)....47

Figure 3.8: Linear-Feedback Shift Register (adapted from Cagigal and Bracho, 1986)....50

Figure 4.1: General Flow of Design and Implementation 51

Figure 4.2: Development Flow of Design and Implementation....53

Figure 4.3: DSP Model of MMBMD in Sys Gen/Simulink 54

Figure 4.4: DSP Model of *Signal Delay* Block....56

Figure 4.5: Settings of *Signal Delay* Block 56

Figure 4.6: DSP Model of MMBM in Sys Gen/Simulink....59

Figure 4.7: DSP Model of *PR Bit Generator* Subsystem 62

Figure 4.8: DSP Model of *Symbol Mapper* Subsystem....63

Figure 4.9: DSP Model of *Interpolation Filter* Subsystem 64

Figure 4.10: DSP Model of *2xMA FIR Multi Interpolator* Subsystem 66

Figure 4.11: DSP Model of MMBD in Sys Gen/Simulink..67

Figure 4.12: DSP Model of *RRC Filter* Subsystem 70

Figure 4.13: DSP Model of *Multi Odd-Tap Systolic FIR Filter* Subsystem.........................72

Figure 4.14: DSP Model of *Reset Delay* Subsystem...73

Figure 4.15: DSP Model of *Timing Recovery* Subsystem 73

Figure 4.16: DSP Model of *TED* Subsystem..75

Figure 4.17: DSP Model of *Search Min Error* Subsystem...76

Figure 4.18: DSP Model of *Data and Timing Updates* Subsystem 77

Figure 4.19: DSP Model of *Data Capture* Subsystem 78

Figure 4.20: DSP Model of *Symbol Demapper* Subsystem...79

Figure 4.21: DSP Model of *Pulse Shaping* Subsystem 80

Figure 4.22: Settings of *System Generator* Block. A: MMBM; B: MMBD.........................82

Figure 4.23: Hardware Implementation of FPGA and P240 91

Figure 4.24: Hardware Assembly for Testing and Measurement 93

Figure 5.1: Simulation of *MMBM* in Sys Gen for BPSK 95

Figure 5.2: Simulation of *MMBM* in Sys Gen for 4-PAM 96

Figure 5.3: Simulation of *MMBM* in Sys Gen for QPSK 96

Figure 5.4: Simulation of *MMBM* in Sys Gen for 16-QAM...97

Figure 5.5: Simulation of *MMBD* in Sys Gen for BPSK..98

Figure 5.6: Simulation of *MMBD* in Sys Gen for 4-PAM..99

Figure 5.7: Simulation of *MMBD* in Sys Gen for QPSK 99

Figure 5.8: Simulation of *MMBD* in Sys Gen for 16-QAM 100

Figure 5.9: Simulation of Ideal MMBMD in Sys Gen for BPSK 101

Figure 5.10: Simulation of Ideal MMBMD in Sys Gen for 4-PAM 101

Figure 5.11: Simulation of Ideal MMBMD in Sys Gen for QPSK....................................102

Figure 5.12: Simulation of Ideal MMBMD in Sys Gen for 16-QAM................................102

Figure 5.13: HDL Simulation of MMBM for BPSK 106

Figure 5.14: HDL Simulation of MMBM for 4-PAM 106

Figure 5.15: HDL Simulation of MMBM for QPSK..107

Figure 5.16: HDL Simulation of MMBM for 16-QAM..107

Figure 5.17: HDL Simulation of MMBD for BPSK...108

Figure 5.18: HDL Simulation of MMBD for 4-PAM...108

Figure 5.19: HDL Simulation of MMBD for QPSK...109

Figure 5.20: HDL Simulation of MMBD for 16-QAM 109

Figure 5.21: HDL Simulation of Setup Configuration Module..112

Figure 5.22: HDL Simulation of Integrated Module of MMBM for BPSK........................115

Figure 5.23: HDL Simulation of Integrated Module of MMBM for 4-PAM......................115

Figure 5.24: HDL Simulation of Integrated Module of MMBM for QPSK........................116

Figure 5.25: HDL Simulation of Integrated Module of MMBM for 16-QAM 116

Figure 5.26: HDL Simulation of Integrated Module of MMBD for BPSK.........................117

Figure 5.27: HDL Simulation of Integrated Module of MMBD for 4-PAM.......................117

Figure 5.28: HDL Simulation of Integrated Module of MMBD for QPSK 118

Figure 5.29: HDL Simulation of Integrated Module of MMBD for 16-QAM 118

Figure 5.30: Real-time Results of MMBM. A: BPSK; B: 4-PAM; C: QPSK;
D: 16-QAM...127

Figure 5.31: Real-time Results of MMBD. A: BPSK; B: 4-PAM; C: QPSK;
D: 16-QAM...129

LIST OF ABBREVIATIONS

AAF	Anti-Aliasing Filter
A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter
Addr	Address
AIF	Anti-Imaging Filter
AMS	Add-Multiply-Sum
ASICs	Application Specific Integrated Circuits
ASK	Amplitude Shift Keying
ASP	Analog Signal Processing
ASR	Addressable Shift Register
BD	Baseband Demodulator
BM	Baseband Modulator
BPF	Bandpass Filter
BPSK	Binary Phase Shift Keying
BRAMs	Block Random Access Memories
C	Current (sample)
CDMA	Code Division Multiple Access
Coef	Coefficient
Concat	Concatenate
DA	Distributed Arithmetic
DAC	Digital-to-Analog Converter
DDC	Digital Down Converter
DDS	Direct Digital Synthesizer
DF	Direct Form

DLL	Delay Locked Loop
DQPSK	Differentially-encoded Quadrature Phase Shift Keying
DRC	Design Rule Check
DSP	Digital Signal Processing
DSPs	Digital Signal Processors
DTFT	Discrete-Time Fourier Transform
DUC	Digital Up Converter
E	Early (sample)
EDIF	Electronic Design Interchange Format
EMs	Embedded Multipliers
FEC	Forward Error Correction
FFs	Flip-Flops
FFT	Fast Fourier Transform
FIFO	First-In-First-Out
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
Gen	Generator
GF(2)	Galois Field of 2 elements
GMSK	Gaussian Minimum Shift Keying
GPPs	General Purpose Processors
GUI	Graphical User Interface
HDL	Hardware Description Language
HEX	HEXadecimal
HSDPA	High-Speed Downlink Packet Access

I	In-phase
IF	Intermediate Frequency
I/O	Input / Output
IOBs	Input / Output Blocks
ISE	Integrated Software Environment
ISI	Inter-Symbol Interference
L	Late (sample)
LF	Loop Filter
LFSR	Linear-Feedback Shift Register
LMS	Least-Mean Square
LNA	Low Noise Amplifier
LO	Local Oscillator
LP	Linear Phase
LPF	Lowpass Filter
LSB	Least Significant Bit
LUTs	Look-Up Tables
LVTTL	Low Voltage Transistor-Transistor Logic
MA	Multiply-Add
MAC	Multiply-ACcumulate
MF	Matched Filter
ML	Maximum Likelihood
MMBD	Multi-Modulation Baseband Demodulator
MMBM	Multi-Modulation Baseband Modulator
MMBMD	Multi-Modulation Baseband Modulator and Demodulator
MOSSF	Multi-Odd-tap Symmetric Systolic Filter

MRIF	Multi-Rate Interpolation Filter
MSB	Most Significant Bit
Mult	Multiplier
Mux	Multiplexer
NCD	Native Circuit Description
NCO	Numerically-Controlled Oscillator
NGD	Native Generic Database
OQPSK	Offset Quadrature Phase Shift Keying
OS	Operating System
PA	Power Amplifier
PACE	Pinout Area Constraints Editor
PAM	Pulse Amplitude Modulation
PAR	Place And Route
PC	Personal Computer
PLL	Phase-Locked Loop
PN	Pseudo-Noise
PR	Pseudo-Random
PSF	Pulse Shaping Filter
PSK	Phase Shift Keying
Q	Quadrature
QAM	Quadrature Amplitude Modulation
QMC	Quadrature Modulator Correction
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAMs	Random Access Memories

RC	Raised Cosine
RCF	Re-Construction Filter
RCIF	Raised Cosine Interpolation Filter
RF	Radio Frequency
ROM	Read-Only Memory
RRC	Root-Raised Cosine
SDR	Software Defined Radio
SER	Symbol Error Rate
SFDR	Spurious Free Dynamic Range
SNR	Signal-to-Noise Ratio
SPI	Serial Programming Interface
SR	Software Radio
SRC	Sample Rate Conversion
STR	Symbol Timing Recovery
Sys	System
TDM	Time-Division Multiplexing
TED	Timing Error Detector
TI	Texas Instruments
UCF	User Constraints File
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
XST	Xilinx Synthesis Technology

LIST OF SYMBOLS

A_i	Signal amplitude of index i
$A_{i,\,I}$	Signal amplitude in I channel
$A_{i,\,Q}$	Signal amplitude in Q channel
A_{sym}	Symbol amplitude
$A_{sym,\,I}$	Symbol amplitude in I channel
$A_{sym,\,Q}$	Symbol amplitude in Q channel
b_n	Binary sequence at instant n
$b_{n\,-\,\lambda}$	Delayed binary sequence at instant $n - \lambda$ with λ latency
c_λ	Coefficient associated with λ-latency binary sequence
C_σ	Filter coefficient for segment σ
C_τ	Filter coefficient at tap τ
$d_{\rho,\,i}$	Output data in level ρ for previous segment i
$d_{\rho,\,\sigma}$	Output data in level ρ for segment σ
$d_I\,(mT_{sym})$	Data samples in I channel at instant m, symbol period T_{sym}
$d_I\,[n]$	Data samples in I channel at instant n
$d_Q\,(mT_{sym})$	Data samples in Q channel at instant m, symbol period T_{sym}
$d_Q\,[n]$	Data samples in Q channel at instant n
$D_{\text{1-2-}split}$	Latency of *1-to-2 Splitter* subsystem
$D_{\text{1-4-}split}$	Latency of *1-to-4 Splitter* subsystem
$D_{\text{2-1-}comb}$	Latency of *2-to-1 Combiner* subsystem
D_{2xMA}	Latency of *2xMultiply-Add* subsystem
$D_{\text{4-1-}comb}$	Latency of *4-to-1 Combiner* subsystem
D_{4xMA}	Latency of *4xMultiply-Add* subsystem
D_{add}	Latency of *Add* block (or adder)

$D_{add\text{-}mult}$	Latency of *Add-Multiply* subsystem
$D_{AMS\text{-}A}$	Latency of *Add-Multiply-Sum A* subsystem
$D_{AMS\text{-}B}$	Latency of *Add-Multiply-Sum B* subsystem
D_{bit}	Latency for synchronizing input message and recovered bits
$D_{bit\text{-}comb}$	Latency of *Bit Combiner* subsystem
D_{capt}	Latency of *Data Capture* subsystem
D_{ch}	Channel delay
D_{ch1}	Channel delay in modulo symbol period T_{sym}
D_{compen}	Compensated channel delay
D_{ctrl}	Latency of *Control* subsystem
D_{ctrl1}	Latency of *Control1* subsystem
D_{CB}	Latency of *Coef Buffer* subsystem
D_{CB1}	Latency of *Coef Buffer1* subsystem
$D_{data\text{-}ctrl}$	Latency of *Data Control* subsystem
D_{delay}	Latency of *Delay* block
D_{demap}	Latency of *Symbol Demapper* subsystem (overall)
D_{demap1}	Latency of *Symbol Demapper* subsystem (without TDM)
D_{DB}	Latency of *Data Buffer* subsystems
D_{DB1}	Latency of *Data Buffer1* subsystem
D_{DTU}	Latency of *Data and Timing Updates* subsystem
D_{gr}	Group delay of RRC filter
D_{gr1}	Group delay of RC filter
D_{int}	Latency for synchronizing original and recovered symbol integers
$D_{I\text{-}demap}$	Latency of *I Demap* subsystem
$D_{I\text{-}map}$	Latency of *I Map* subsystem

D_{IPF}	Latency of *Interpolation Filter* subsystem
$D_{IQ\text{-}in}$	Initial delay of peak of I/Q input signal
D_{map}	Latency of *Symbol Mapper* subsystem
D_{mult}	Latency of *Mult* block (or multiplier)
$D_{mult\text{-}sum}$	Latency of *Multiply-Sum* subsystem
D_{MOSSF}	Latency of *Multi Odd-Tap Systolic FIR Filter* subsystem (or MOSSF)
D_{MRIF}	Latency of *2xMA FIR Multi Interpolator* subsystem (or MRIF)
D_{PRBG}	Latency of *PR Bit Generator* subsystem
D_{PS}	Latency of *Pulse Shaping* subsystem
$D_{Q\text{-}demap}$	Latency of *Q Demap* subsystem
$D_{Q\text{-}map}$	Latency of *Q Map* subsystem
D_{recov}	Latency of *Timing Recovery* subsystem
D_{RCIF}	Latency of *4xMA FIR Interpolator* subsystem (or RCIF)
D_{RRC}	Latency of *RRC Filter* subsystem
$D_{set\text{-}0}$	Latency of *Set Zero* subsystem
D_{symb}	Latency for synchronizing original and recovered I/Q symbols
D_{SME}	Latency of *Search Min Error* subsystem
D_{TED}	Latency of *TED* subsystem
$D_I(z)$	Z-transform of data samples $d_I[n]$ in unit cycle z
$D_Q(z)$	Z-transform of data samples $d_Q[n]$ in unit cycle z
e	Timing error
e_{min}	Minimum timing error
$e(mT_{sym})$	Timing error at instant m, symbol period T_{sym}
$e[n]$	Timing error at instant n
$E(z)$	Z-transform of timing error $e[n]$ in unit cycle z

f	Frequency
f_c	Carrier frequency
f_{co}	Cut-off frequency
f_N	Nyquist frequency
f_s	Sampling frequency or sample rate
f_{s1}	Input (lower) sampling frequency or rate
f_{s2}	Output (higher) sampling frequency or rate
$f(x)$	Polynomial function in terms of x
G_{ch}	Channel gain
$h(nT_{s2})$	Impulse response at instant n, sampling period T_{s2}
$h[n]$	Impulse response at instant n
$h_{RC}(t)$	Impulse response of RC filter at time t
$h_{RRC}(t)$	Impulse response of RRC filter at time t
$H(e^{j2\pi fTs2})$	DTFT of signal $h(nT_{s2})$ in frequency f, sampling period T_{s2}
$H(z)$	System function in unit cycle z
$H_{DF}(z)$	System function of DF FIR filter in unit cycle z
$H_S(z)$	System function of systolic FIR filter in unit cycle z
$H_{RC}(\omega)$	Frequency response of RC filter in radian frequency ω
$H_{RRC}(\omega)$	Frequency response of RRC filter in radian frequency ω
Int	Symbol integers
k	Number of bits
K_i	Integral gain
K_p	Proportional gain
L	Upsampling (oversampling) factor or interpolation rate
m	Sampling instant at lower sampling rate; symbol instant

M	Filter order; number of shift register stages; degree of binary sequence
M_{DIP}	9-bit value of the first black DIP switch
M_o	Modulation order
n	Sampling instant at higher or single sampling rate; timing index
n_{capt}	Capture instant
n_{lim1}	Lower limit of timing index
n_{lim2}	Upper limit of timing index
n_{lock}	Locking instant
n_{on}	On-state timing for valid triggering of update loading
n_{opt}	Optimum sampling instant
n_{reload}	Reload instant
n_{upd}	Updated timing index
N	Filter length or filter taps
N_{DIP}	2-bit value of the second black DIP switch
N_o	Optimized filter taps
N_s	Segmented filter taps
N_{sym}	Offset symbol period or interval
ov_{th}	Over-threshold state
O	Over filter phase taps
$p(t)$	Pulse waveform at time t
$p_{rect}(t)$	Rectangular pulse waveform at time t
$\|P_H\|$	Absolute value (or modulus) of higher pulse peak
$\|P_L\|$	Absolute value (or modulus) of lower pulse peak
q	Filter phase taps or number of taps in each phase sub-filter
q_o	Optimized filter phase taps

R	Sample rate-change factor
R_b	Bit rate
R_{sym}	Symbol rate
$s_i\,(t)$	Signal waveform of index i at time t
S_{sym}	Number of samples per symbol
S_{sync}	Synchronous state of stable optimum sampling instant n_{opt}
t	Time
t_p	Pulse interval
$t_{p(min)}$	Minimum pulse interval
T_b	Bit period or interval
T_{b1}	First bit interval
T_R	Period of linear recurring sequence b_n
T_s	Sampling period or interval, or unit sample time
T_{s1}	Input (higher) sampling period or interval
T_{s2}	Output (lower) sampling period or interval
T_{sym}	Symbol period or interval
U	Downsampling factor or decimation rate
v_{init}	Initial condition
$w(nT_{s2})$	Discrete-time intermediate signal at instant n, sampling period T_{s2}
$W(e^{\,j2\pi fTs2})$	DTFT of signal $w(nT_{s2})$ in frequency f, sampling period T_{s2}
$x(t)$	Analog or continuous-time input signal at time t
$x(mT_{s1})$	Discrete-time input signal at instant m, sampling period T_{s1}
$x[n]$	Discrete-time input signal at instant n
$X(e^{\,j2\pi fTs1})$	DTFT of signal $x(mT_{s1})$ in frequency f, sampling period T_{s1}
$y(t)$	Analog or continuous-time output signal at time t

$y(nT_{s2})$	Discrete-time output signal at instant n, sampling period T_{s2}
$y_{\phi}\ (mT_{s1})$	Discrete-time output signal of phase ϕ at instant m, sampling period T_{s1}
$y[n]$	Discrete-time output signal at instant n
$y_{\sigma}\ [n]$	Discrete-time output signal for segment σ at instant n
$Y(e^{j2\pi fTs2})$	DTFT of signal $y(nT_{s2})$ in frequency f, sampling period T_{s2}
z	Unit cycle dimension, equivalent to $e^{j\omega}$
α	Roll-off factor
β	Shape parameter for Kaiser window method
$\delta[n]$	Unit sample (or impulse) sequence at instant n
Δt	Time difference between cursors 1 and 2 in oscilloscope display
θ_0	Phase offset
λ	Index of shift register stage; latency in LFSR
μ	Scaling factor
ρ	Level index of filter structure
σ	Filter segment index
τ	Filter tap index
τ_p	Polyphase filter tap index
χ	Filter segments
ω	Radian frequency
ω_{co}	Cut-off radian frequency
ω_p	Passband radian frequency
ω_s	Stopband radian frequency
ϕ	Filter phase index
ϕ_s	Sampling phase

REKABENTUK PEMODULAT DAN PENYAHMODULAT JALUR-DASAR PELBAGAI-MODULASI BAGI RADIO TERTAKRIF PERISIAN

ABSTRAK

Berlainan daripada radio berasaskan perkakasan yang hanya menyampaikan satu perkhidmatan komunikasi menggunakan piawaian tertentu, radio tertakrif perisian (SDR) menawarkan pelantar yang amat boleh-dikonfigurasi-semula untuk menyepadukan pelbagai fungsi bagi sistem komunikasi wayarles yang pelbagai-modulasi, pelbagai-jalur dan pelbagai-piawaian. Tetapi, projek ini hanya berdasarkan SDR pelbagai-modulasi, iaitu 4-PAM, BPSK, QPSK dan 16-QAM. Pemodulat (MMBM) dan penyahmodulat (MMBD) jalur-dasar pelbagai-modulasi yang boleh-dikonfigurasi direkabentuk dengan menggunakan algoritma pemprosesan isyarat digit (DSP) berdasarkan ciri-ciri umum yang dikongsi oleh struktur-struktur satu-modulasi, dan seterusnya dilaksanakan dalam FPGA Virtex-4 Xilinx. Perbandingan keputusan masa nyata dan simulasi menunjukkan bahawa pemasaan adalah setara, dan perubahan tanda dan magnitud adalah ketara. Tambahan pula, sekurang-kurangnya 17% penggunaan FPGA telah dijimat bagi MMBM dan MMBD yang boleh-dikonfigurasi tersebut berbanding dengan pemodulat and penyahmodulat jalur-dasar berkumpulan-modulasi masing-masing, yang diubahsuai daripada kerja-kerja terdahulu. Di samping itu, penukar digit-ke-analog (DAC) dan penukar analog-ke-digit (ADC) boleh ditukar untuk aplikasi lain dengan menyepadukan konfigurasi persediaan bagi DAC dan ADC ke dalam MMBM dan MMBD yang boleh-dikonfigurasi tersebut masing-masing. Selain itu, MMBM dan MMBD yang boleh-dikonfigurasi tersebut boleh diubahsuai selanjutnya untuk merangkumi modulasi PSK, PAM dan QAM yang M-ary lebih tinggi.

DESIGN OF MULTI-MODULATION BASEBAND MODULATOR AND DEMODULATOR FOR SOFTWARE DEFINED RADIO

ABSTRACT

In contrast to hardware-based radio that only delivers single communication service using particular standard, the software defined radio (SDR) provides a highly reconfigurable platform to integrate various functions for multi-modulation, multi-band and multi-standard wireless communication systems. However, this project is only based on multi-modulation SDR, such as 4-PAM, BPSK, QPSK and 16-QAM. The configurable multi-modulation baseband modulator (MMBM) and demodulator (MMBD) are designed using digital signal processing (DSP) algorithms based on common features shared by single-modulation structures, and then implemented into Xilinx Virtex-4 FPGA. Comparing the real-time and simulation results shows that the timings are equivalent, and the sign and magnitude changes are significant. Furthermore, at least 17% FPGA utilizations have been saved for the configurable MMBM and MMBD as compared to the grouped-modulation baseband modulator and demodulator respectively, which are modified from the previous works. Moreover, digital-to-analog converter (DAC) and analog-to-digital converter (ADC) can be changed for different applications by integrating setup configuration for DAC and ADC into the configurable MMBM and MMBD respectively. Besides, the configurable MMBM and MMBD can be further modified to include higher M-ary modulations of PSK, PAM and QAM.

CHAPTER 1

INTRODUCTION

1.1 General Overview

Multi-modulation is one of the multi functionalities applicable to software defined radio (SDR) in wireless communication systems besides multi-standard and multi-band functionalities. The multi modulations used in the project of this thesis are pulse amplitude modulation (PAM), binary phase shift keying (BPSK), quadrature phase shift keying (QPSK) and quadrature amplitude modulation (QAM). The digital amplitude modulation is PAM, digital phase modulations are BPSK and QPSK, and digital amplitude-phase modulation is QAM. The exclusion of digital frequency modulation such as frequency shift keying (FSK) and Gaussian minimum shift keying (GMSK), is because of nonlinear modulation scheme whereas the PAM, PSK and QAM are linear modulation schemes (Proakis and Salehi, 2008).

The configurable multi-modulation baseband modulator (MMBM) and demodulator (MMBD) for SDR is developed by using digital signal processing (DSP) algorithms, and is highly dependent on the common features as stated below:

a) Schemes of digital amplitude, phase and amplitude-phase modulations for symbol mapper and demapper.

b) Filter design parameters, coefficients and structures for pulse shaping filter (PSF) and matched filter (MF).

c) Detection of timing error (*e*) and acquisition of optimum sampling instant (n_{opt}) for symbol timing recovery (STR).

The operations and implementations of symbol mapper and demapper, PSF, MF and STR in baseband modulation and demodulation, will be presented and described in detailed in Chapters 2 to 4.

Due to continuous-time radio frequency (RF) signal still remains prominence in propagating through air interface (channel), the conversions between continuous-time and discrete-time domains using analog-to-digital converter (ADC) and digital-to-analog converter (DAC), have become an important implementation issue, in order to optimize the performance of data sampling in terms of bandwidth, signal-to-noise ratio (SNR), precision, and timing constraints (Friedman, 1990; Harris, 1998; Walden, 1999). Therefore, effective signal transformations are required before and after field programmable gate array (FPGA) by using ADC and DAC respectively. Ignoring the implementation issues of interfacing FPGA with ADC and DAC would possibly cause undesired data losses before and after processing of FPGA respectively, and adversely affect the overall system performance. Therefore, setup configuration module is also proposed for configuring ADC and DAC as well as FPGA onboard clock synthesizer, in order to avoid the undesired data losses.

The configurable MMBM and MMBD together with the setup configuration module are implemented using Xilinx Virtex-4 FPGA MB Board and P240 Analog Module (ADC and DAC). The model name of Xilinx Virtex-4 FPGA is xc4vsx35-10ff668. The ADC and DAC contained in P240 Analog Module are Texas Instruments (TI) ADS5500 (14-bit, 125 MSps) and TI dual-channel DAC5687 (16-bit, 500 MSps, 2x-8x interpolation) respectively.

1.2 Problem Statements

Wireless communications have become pervasive since the past two decades, such as universal mobile telecommunications system (UMTS) and wireless local area network (WLAN) (Rappaport, 2002). The rapid growth of wireless communication systems has led to the development of newer wireless systems and standards for high speed data services and voice calls. Traditional hardware-based radio deliver only single communication service using particular standard and cannot support multi-standard wireless communication system due to its limited cross-functionality. Moreover, its upgrade via physical intervention would result in high development and production costs. However, the emerging software radio (SR) and software defined radio (SDR) has overcome the problems by providing re-configurability and programmability of different physical layer functions in one piece of hardware with lower costs (Mitola, 1995; Software Defined Radio Forum, 2009).

The SDR concept employs intensively the DSP techniques to perform signal processing tasks which are traditionally performed by the analog components in transmitter and receiver (Mitola, 1995; Harris 1998). The DSP advantages of low complexity, high configurability, high speed, small size, low power consumption, low development time and low production cost, have become keys for substituting the sophisticated analog signal processing (ASP) (Mitola, 1995; Harris, 1998; Oppenheim and Schafer, 1999). However, the scalability, flexibility and multi-functionality of SDR would require complicated DSP-based signal processing tasks, thus leading to the issues of hardware and timing constraints in the implementation platform of SDR (Baines, 1995; Cummings, 2004).

Multi wireless standards are associated with different digital modulations and data rates (in chips or bits per second) as listed in Table 1.1 (Rappaport, 2002). Thus, a SDR platform with integration of different modulations and data rates can be reused for multi standards and would save considerable development time and costs (Hatai and Chakrabarti, 2010). However this would increase the complexity of SDR, thus larger size and higher power consumption. Therefore, a novel compression technique with optimization for the integration is proposed in this thesis.

Table 1.1: Wireless Standards with Modulations and Data Rates (Rappaport, 2002)

Standard	Modulation	Data Rate
CDMA2000	BPSK, QPSK	1.2288 Mcps
WCDMA	QPSK, 16-QAM	3.84 Mcps
HSDPA	QPSK, 16-QAM	3.6 Mbps
IEEE802.11 (WLAN)	QPSK, 16/64-QAM	12 Mbps

1.3 Objectives

The aim of this thesis is to design multi-modulation baseband modulator (MMBM) and demodulator (MMBD) for software defined ratio (SDR). To achieve the aim, several objectives have to be fulfilled such as stated below:

1. To design and simulate MMBM and MMBD with configurable-modulations of BPSK, 4-PAM, QPSK and 16-QAM.
2. To design and simulate setup configuration module for configuring ADC, DAC and FPGA onboard clock synthesizer.
3. To implement the integrated modules of MMBM and MMBD (with setup configuration) into FPGA devices with ADC and DAC.
4. To analyze system performances of configurable MMBM and MMBD in terms of timings, pulse peaks, and FPGA utilization.

1.4 Scope of Works

The works in this thesis focus on the designs of: digital signal processing (DSP) models of multi-modulation baseband modulator (MMBM) and demodulator (MMBD); setup configuration module for ADC, DAC and FPGA onboard clock synthesizer; and the integrated modules of MMBM and MMBD (with setup configuration).

The first design is to develop a combination of pseudo-random (PR) bit generator, symbol mapper and root-raised cosine (RRC) pulse shaping filter (PSF), and a combination of RRC matched filter (MF), symbol timing recovery (STR), symbol demapper and raised cosine (RC) PSF, as the DSP models of MMBM and MMBD respectively, with configurable-modulations of BPSK, 4-PAM, QPSK and 16-QAM. The design includes simulations of models and hardware description language (HDL) Verilog codes, in Xilinx System Generator/Simulink and ModelSim environments respectively, and comparison between both results in term of timing.

The second design is to develop a combination of configurations of ADC, DAC and FPGA onboard clock synthesizer as setup configuration module. The design includes simulation of HDL Verilog codes in ModelSim environment and verification of serial programming interface (SPI) timing characteristics.

The third design is to develop the integration of DSP model of MMBM (and MMBD) with setup configuration module as the integrated module of MMBM (and MMBD respectively). The design includes simulation of HDL Verilog codes, synthesis and implementation in ModelSim, Synplify Pro and Xilinx ISE environments respectively. The real-time implementation results are compared with the simulation results in terms of pulse interval and pulse peaks, in order to prove that the thesis objectives are satisfied.

1.5 Contribution of Research

The main contributions of this thesis are stated as below:

1. The development of baseband modulator (BM) and demodulator (BD) with configurability of multi digital amplitude and/with phase modulations such as BPSK, 4-PAM, QPSK and 16-QAM, which expands the software defined radio (SDR) applications.
2. The development of pulse-shaping filter (PSF) with configurability of multi interpolation rate (*L*) corresponding to the selected digital modulation, which reduce complexity of sample rate conversion (SRC) in multi-rate processing.
3. The improvement of the efficiencies of configurable multi-modulation baseband modulator (MMBM) and demodulator (MMBD) to reduce hardware size or FPGA utilization, thus leading to lower power consumption.
4. The development of setup configuration module for analog-to-digital converter (ADC) and digital-to-analog converter (DAC), which offers configurability and changeability of ADC and DAC in order to interface with various analog signals for various applications of communication systems.

1.6 Thesis Outline

Chapter 1: General overview of this thesis, problem statements, objectives, scope of works and contribution of thesis, are presented.

Chapter 2: Literature review starts with software defined radio (SDR) concept including functional architecture and implementation platform. The second section is basics of digital linear modulation (PAM, PSK and QAM) and the recent researches using FPGA. The third section is theory and implementation of raised cosine (RC) pulse shaping filter (PSF). The last section is schemes of symbol timing recovery (STR), and the recent researches of STR.

Chapter 3: Design concept elaborates implementation techniques, starting with theory and structures of finite impulse response (FIR) filter. The second section is theory and polyphase architecture of interpolation filter in sample rate conversion (SRC). The third section is theory and structure of Gardner timing error detector (TED). The last section is theory and structure of pseudo-random (PR) bit generator.

Chapter 4: The first section is development flow of design and implementation of multi-modulation baseband modulator (MMBM) and demodulator (MMBD) for SDR. The second section is the required software and versions. The third until eighth sections are: development of DSP models of *MMMM* and *MMBD* then compiled as HDL netlists using Xilinx System Generator/Simulink software; HDL design of setup configuration module for ADC, DAC and clock synthesizer, and its integration with the DSP models using ModelSim software; synthesis and implementation of the integrated module of MMBM (and MMBD) using Synplify Pro and Xilinx ISE software respectively. The last section is hardware implementation and verification of the integrated modules of MMBM and MMBD using FPGA, ADC and DAC.

Chapter 5: Initially, simulation results of DSP models run in System Generator (Sys Gen) are discussed and compared to verify functionality. In the second section, HDL simulation results run in ModelSim are analyzed and compared with the Sys Gen simulation results in term of timing for overall behavioral verification. The third section is verification of timing constraints with estimated timing after synthesis of the integrated modules using Synplify Pro. The fourth section is FPGA performances for implementing the integrated modules using ISE, in terms of device utilization, speed and power consumption. In the fifth section, real-time results are compared with the simulation results in terms of pulse interval and peaks for verifying system functionality. In the last section, efficiencies of symbol mapper, symbol demapper, PSF, matched filter (MF), STR, MMBM and MMBD are evaluated in term of FPGA utilization.

Chapter 6: Summary of the project in this thesis emphasizes on the fulfillment of the thesis objectives, brief development flow of design and implementation, and advantages of the proposed design compared to the previous researches and works. The next section is recommendations of future works to extend the purposes of the proposed design and new exploration for other researchers.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the concept of software defined radio (SDR) is described briefly to show its importance in wireless communication systems nowadays. The SDR architecture focusing on baseband processing is studied to emphasize the signal processing tasks of digital modulation and demodulation. Then, the reasons to choose FPGA as implementation platform of SDR are also presented.

The digital linear modulations such as PAM, PSK and QAM are commonly used to develop symbol mapper and demapper in SDR. The recent researches related to PAM, PSK and QAM are also presented.

The raised cosine (RC) filtering is presented in detailed to show its prominence in developing pulse shaping filter (PSF) and matched filter (MF) in SDR.

Lastly, symbol timing recovery (STR) schemes are presented in detailed. It shows that configurable multi-symbol-rate STR offers an attractive idea to support multi-modulation SDR.

2.2 Software Defined Radio (SDR) Overview

Software radio (SR) is a set of digital signal processing (DSP) primitives, a metalevel system for combining the DSP primitives into communication system's functions (transmitter, channel model, receiver), and a set of target processor on which SR is hosted for real-time communication (Mitola, 1993). An ideal SR receiver begins processing of incoming signal right after the antenna. Similarly, an ideal SR transmitter is located just before antenna. However, it is not practical due to constraints of ADC and DAC in terms of bandwidth and dynamic range. Therefore, a more practical version of SR called software defined radio (SDR) has emerged that performs bit-stream, baseband and intermediate frequency (IF) processing with ADC and DAC as closed to radio frequency (RF) conversion as possible (Mitola, 1995).

The SDR is simply defined as "radio in which some or all of the physical layer functions are software defined" by the SDR Forum (Software Defined Radio Forum, 2009). This implies that the architecture is flexible such that SDR may be configured in real time to adapt itself to various wireless standards and waveforms, frequency bands, bandwidths, and modes of operation (Rouphael, 2009). In commercial communication systems, the adoption of SDR concept in base-stations and mobile terminals is quite prevalent, because the SDR features of multi-band multi-standard functionality, programmability and upgradeability, offer an attractive solution to the problems of high cost, high power consumption and large size suffered by the Velcro approach which consists of multi radios employing multi chipsets and platforms to support various applications (Rouphael, 2009). Moreover, the flexibility of SDR also supports various quality-of-service (QoS) requirements dealing with numerous data, voice and multimedia applications during mode switching between different air-interface standards (Aghvami et al., 2001).

2.2.1 Functional Architecture of SDR

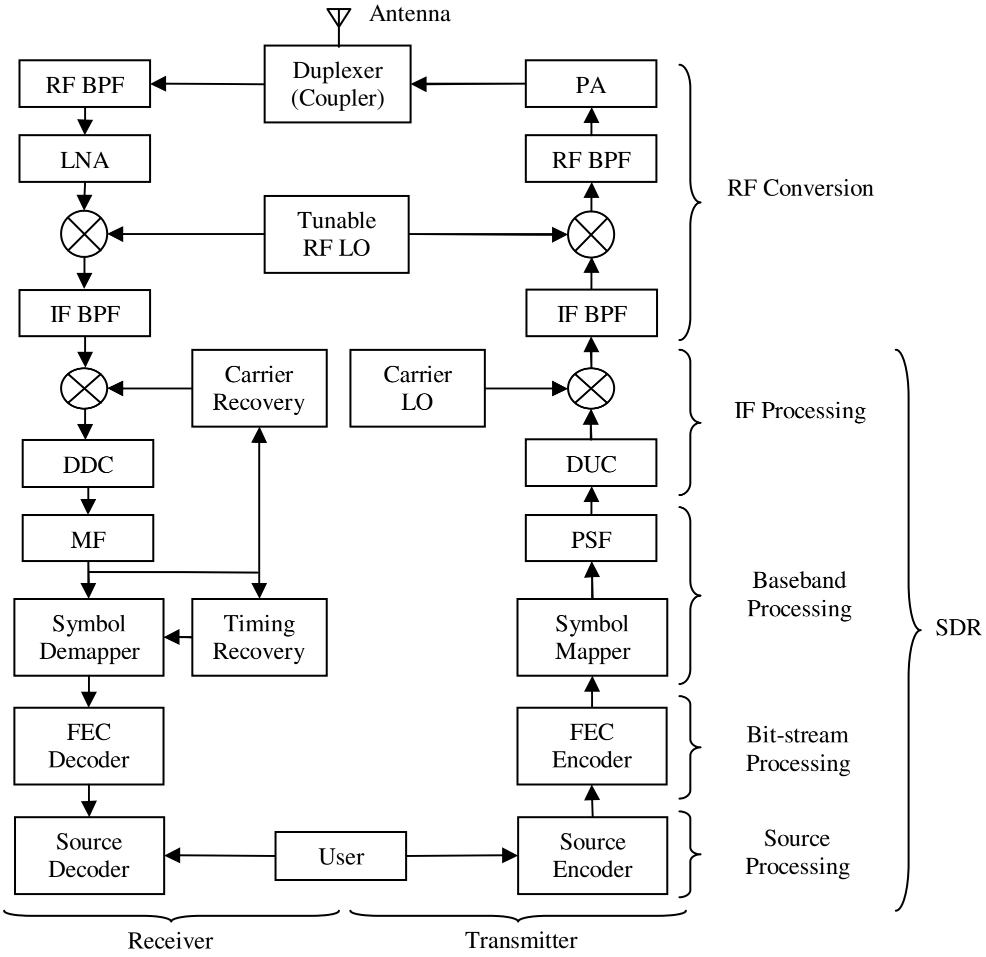

Figure 2.1: Functional Architecture of SDR (Mitola, 1995)

The functional architecture of SDR is illustrated in Figure 2.1 (Mitola, 1995). For transmitter part, the information source coming from the user is going through: source encoder, forward error correction (FEC) encoder, symbol mapper, pulse shaping filter (PSF) to become baseband signal, digital up converter (DUC), digital mixer with carrier local oscillator (LO) to become IF signal, IF bandpass filter (BPF), analog mixer with tunable RF LO to become RF signal, RF BPF, power amplifier (PA), duplexer, and finally antenna for transmission through air interface (channel). For receiver part, the RF signal received from antenna is going through: duplexer, RF BPF, low noise amplifier (LNA), analog mixer with tunable RF LO, IF BPF to become IF signal, digital mixer with carrier recovery, digital down converter (DDC) to become baseband signal, matched filter (MF), symbol demapper with symbol timing recovery (STR), FEC decoder, source decoder, and finally back to the user.

From this point of view, the DAC and ADC are located between the interfaces of RF conversion and IF processing for transmitter and receiver parts respectively. However, this thesis only focuses on the baseband processing segment of the SDR architecture, meaning that the ADC and DAC are located between the interfaces of IF processing and baseband processing.

2.2.2 Baseband Processing Segment in SDR

The baseband processing involves the first level of baseband modulation (in transmitter) and demodulation (in receiver), predistortion (in transmitter) for nonlinear channels, Trellis coding (in transmitter) and decoding (in receiver), and soft decision parameter estimation (in receiver) (Mitola, 1995). The complexity of baseband segment depends on the bandwidth at baseband, and complexity of channel waveform and the related processing such as soft decision in receiver (Mitola, 1995). The baseband modulator (BM) and demodulator (BD) are composed of symbol mapper and PSF; and MF, STR and symbol demapper, respectively.

In baseband modulation and demodulation, PSF and MF are respectively used to improve the symbol detection by reducing inter-symbol interference (ISI) that will degrade the performance of receiver (Gentile, 2002). However, the pulse shaping would require oversampling and filtering in order to reduce the infinity bandwidth of rectangular pulse of symbol after symbol mapping (in transmitter). This process is referred as interpolation in sample rate conversion (SRC), in which the sampling rate is increased from symbol rate by a certain oversampling factor. The concept of SRC and interpolation will be described in detailed in Subheading 3.3 in Chapter 3.

2.2.3 Suitability of FPGA as Implementation Platform of SDR

The commonly used processor devices for implementation of DSP-based SDR systems are general purpose processors (GPPs), digital signal processors (DSPs), application specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs) (Chou et al., 2004; Jaamour and Safadi, 2004; Safadi and Ndzi, 2006).

The GPPs function with fully deterministic execution time that are switched by operating systems (OS) using multi-threading concept, result in low speed, high power consumption and high cost (Cumming, 2004; Jaamour and Safadi, 2004; Safadi and Ndzi, 2006).

Increase in the number of instructions for executing computationally-intensive functions in the DSPs, causes low data throughput and high power consumption (Chou et al., 2004; Cummings and Haruyama, 1999; Cumming, 2004; Jaamour and Safadi, 2004; Safadi and Ndzi, 2006).

The ASICs suffer from low flexibility, high cost and long design cycle for system upgrades (Chou et al., 2004; Cummings and Haruyama, 1999; Cumming, 2004; Jaamour and Safadi, 2004; Safadi and Ndzi, 2006).

All these problematic issues have made the GPPs, DSPs and ASICs difficult to be adapted in multi-function and multi-rate DSP systems. However, the emerging technology of FPGAs offer a better solution for implementing SDR by employing dynamically reconfiguration and parallelism in pipelined structures, to build DSP functions such as multiply-accumulate (MAC), distributed arithmetic (DA) and Fast Fourier Transform (FFT), from the DSP primitive elements such as look-up tables (LUTs), multiplexers (Muxs), flip-flops (FFs), RAMs (Random Access Memories), dedicated multipliers and first-in-first-out (FIFO) (Cummings and Haruyama, 1999; Kramberger, 1999; Chou et al., 2004; Coulton and Carline, 2004; Cummings, 2004; Dick and Hwang, 2004; Safadi and Ndzi, 2006).

2.3 Digital Linear Modulations

The digital modulation or signaling is defined as the process of mapping digital data in the bit-stream form (a sequence of 0s and 1s) to signals that will match the characteristics (e.g. bandwidth) of communication channel and resist most of the channel impairments (e.g. noise, attenuation, distortion, fading, interference, etc.) for transmission over the communication channel (Proakis and Salehi, 2008).

In memoryless digital modulation scheme, the bit-stream is parsed into sets of *k* number of bits or k-bit symbol integers:

$$Int = 0, 1, 2, \ldots, M_o - 1, \tag{2.1}$$

where the modulation order M_o is given by

$$M_o = 2^k. \tag{2.2}$$

Each set of *k* number of bits (or symbol integer) is mapped into one of the M_o possible signal waveforms, i.e. $s_i(t)$, for $0 \le i \le M_o - 1$ (Proakis and Salehi, 2008).

The symbol period is T_{sym} such that the symbols are transmitted at every $t = T_{sym}$ seconds (Proakis and Salehi, 2008). While the symbol rate is given by:

$$R_{sym} = 1/T_{sym}. \tag{2.3}$$

Since each symbol contains *k* bits of information, the bit period is given by:

$$T_b = T_{sym}/k = T_{sym}/\log_2 M_o. \tag{2.4}$$

So, the bit rate is given by:

$$R_b = kR_{sym} = R_{sym}\log_2 M_o. \tag{2.5}$$

The preferred mapping of symbol integers into symbols is using Gray coding, in which the adjacent symbol integers (sets of k number of bits) differ by one bit (Proakis and Salehi, 2008). This is because only a single bit error will happen when the symbol demapper in demodulation selects the adjacent symbol of the expected (transmitted) symbol due to the effect of ISI.

The digital modulation is classified as linear if the principle of superposition can be applied in the mapping of the digital sequence into successive waveforms (Proakis and Salehi, 2008). The digital modulation schemes that satisfy this property are pulse amplitude modulation (PAM), phase shift keying (PSK) and quadrature amplitude modulation (QAM), which will be presented in the following subheadings.

2.3.1 Pulse Amplitude Modulation (PAM)

Assume that $p(t)$ is a rectangular pulse of duration T_{sym} with amplitude of 1, and modulation order is M_o (= 2^k), the signal waveforms of PAM can be represented as (Proakis and Salehi, 2008):

$$s_i(t) = A_i p(t), 0 \le i \le M_o - 1, \tag{2.6}$$

where the signal amplitude is given by:

$$\begin{aligned} A_i &= (2i + 1 - M_o), 0 \le i \le M_o - 1 \\ A_i &= \pm 1, \pm 3, \pm 5, \ldots, \pm (M_o - 1) \end{aligned} \tag{2.7}$$

However, the signal amplitude A_i can be scaled by a factor μ to become the desired symbol amplitude, as given by:

$$\begin{aligned} A_{sym} &= \mu A_i \\ A_{sym} &= \pm \mu, \pm 3\mu, \pm 5\mu, \ldots, \pm (M_o - 1)\mu \end{aligned} \tag{2.8}$$

The PAM is one-dimensional (1-D) digital modulation since all symbols are amplitude multiples of the same basic signals, as given by equation (2.6). Therefore, only in-phase (I) channel is involved in symbol mapping of PAM.

The recent researches related to PAM are designs and implementations of baseband PAM (Asif et al., 2006) and bandpass PAM or amplitude shift keying (ASK) modem involving FPGA platform (Saha and Sinha, 2009).

The former proposed a platform composed of microcontroller, DSP processor and FPGA, which perform processing of data from and to real world traffic, signal processing of PAM modulation and demodulation, and buffering of data interfaced with RF conversion and driving of clocking interface respectively (Asif et al., 2006). The DSP processor is actually the heart of communication system but the FPGA with FIFO buffer is mainly used to conserve input/output bandwidth required for the main computational load of the DSP processor (Asif et al., 2006).

As SDR complexity keeps increasing nowadays, the DSP processor can no longer support the computational speed due to increase of the number of instructions for executing computationally-intensive DSP functions in the SDR. Therefore, the PAM (or ASK) modulation and demodulation are recommended to be implemented using FPGA with full parallelism in running the DSP functions, thus leading to high data throughput and low power consumption (Saha and Sinha, 2009). Besides, the dynamically reconfiguration of FPGA also allows configurability of PAM modem to support for various applications of SDR.

2.3.2 Phase Shift Keying (PSK)

Assume that $p(t)$ is a rectangular pulse of duration T_{sym} with amplitude of 1, and modulation order is M_o (= 2^k), the signal waveforms of PSK can be represented as (Proakis and Salehi, 2008):

$$\begin{aligned} s_i(t) &= p(t)\sin\left[2\pi f_c t + \frac{2\pi i}{M_o} + \theta_0\right], 0 \le i \le M_o - 1 \\ &= p(t)\left[\cos\left(\frac{2\pi i}{M_o}\right)\sin\left(2\pi f_c t + \theta_0\right) + \sin\left(\frac{2\pi i}{M_o}\right)\cos\left(2\pi f_c t + \theta_0\right)\right] \end{aligned} \qquad (2.9)$$

where f_c is carrier frequency in Hertz (Hz) and θ_0 is phase offset in radian (rad).

The first and second components of equation (2.9) in terms of sin ($2\pi f_c t$ + θ_0) and cos ($2\pi f_c t$ + θ_0) can be represented as symbol amplitudes of I and Q channels:

$$A_{sym,I} = \mu\cos\left(2\pi i / M_o\right), 0 \le i \le M_o - 1 \qquad (2.10)$$

$$A_{sym,Q} = \mu\sin\left(2\pi i / M_o\right), 0 \le i \le M_o - 1 \qquad (2.11)$$

where μ is a scaling factor.

For binary PSK (BPSK): M_o = 2, $A_{sym,\ Q}$ = 0 for i = 0, 1; thus Q channel is useless, therefore BPSK is 1-D digital modulation (Proakis and Salehi, 2008). However for $M_o \ge 4$, $A_{sym,\ I} \neq 0$ and $A_{sym,\ Q} \neq 0$ for $0 \le i \le M_o - 1$; thus both I and Q channels are involved, therefore M_o ($\ge$ 4)-PSK is 2-D digital modulation.

The recent researches related to PSK are designs and FPGA implementations of BPSK modem (Ahamed and Scarpino, 2005; Popescu et al., 2010) and QPSK modem (Song and Yao, 2010; Hatai and Chakrabarti, 2010; Bhandari et al., 2009).

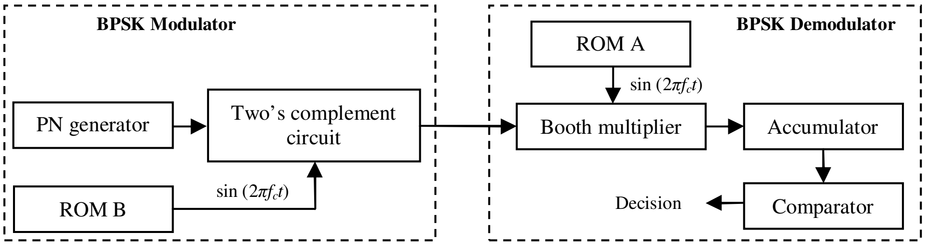

Figure 2.2: Simple BPSK Modem (Ahamed and Scarpino, 2005)

The FPGA implementation of a simple BPSK modem is illustrated in Figure 2.2 (Ahamed and Scarpino, 2005). The BPSK modulator is composed of pseudo-noise (PN) generator, read-only memory (ROM) and 2's complement circuit which are used for generating pseudo-random data pattern as input bits, generating sampled data stream as carrier signal, and inverting the phase of carrier sampled data only if the PN signal is '0', respectively. Whereas the BPSK demodulator is composed of ROM, Booth multiplier, accumulator and comparator which are used for generating sampled data stream as reference carrier signal, multiplying the transmitted and reference carrier signals, repeatedly storing 8 consecutive product samples from multiplier, and making decision of either '1' or '0' if the accumulated value is either positive or negative, respectively. For further improving the BPSK modem, the linear-feedback shift register (LFSR) in the PN generator is modified from 4-stage to 6-stage for better randomness, and the serial Booth multiplier is replaced by a parallel dedicated multiplier (Popescu et al., 2010).

The FPGA implementation of a simple QPSK modem is illustrated in Figure 2.3 (Song and Yao, 2010) below. In the QPSK modulator: the input bit is generated from data generator (PN generator), converted into a pair of bits and mapped to be the I and Q symbols $A_{sym,\,I}$ and $A_{sym,\,Q}$ which are then multiplied with two orthogonal carrier signals sin $(2\pi f_c t)$ and cos $(2\pi f_c t)$ respectively and summed together to form the QPSK signal, as referring to equations (2.9). The direct digital synthesizer (DDS) is used to generate the carrier signals sin $(2\pi f_c t)$ and cos $(2\pi f_c t)$ instead of using ROM (referring to Figure 2.2) because the DDS can achieve higher spurious free dynamic range (SFDR) by using phase dithering and Taylor series correction (Xilinx, 2005). On the other hand, in the QPSK demodulator: the received QPSK signal is multiplied with the same carrier signals sin $(2\pi f_c t)$ and cos $(2\pi f_c t)$ as in the modulator to form I and Q mixed signals, then high frequency components in the I and Q mixed signals are removed by using 2 separate finite impulse response (FIR) lowpass filter (LPF) and decision is made upon the filtered signal whether it is positive or negative to determine the recovered I and Q symbols as either '1' or '0' respectively, lastly demapped and converted back to bit sequence. In actual practice, DDS in the receiver should be replaced by a carrier recovery or synchronization, and a symbol timing recovery (STR) should be inserted between the LPF and the decision circuit (Dick et al., 2000).

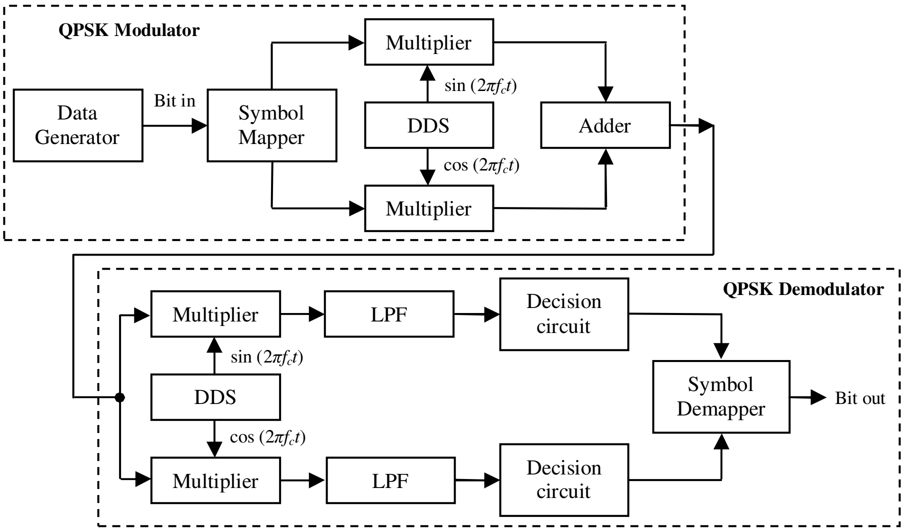

Figure 2.3: Simple QPSK Modem (Song and Yao, 2010)

The reconfigurable baseband modulator proposed by (Hatai and Chakrabarti, 2010) is similar to the upper part of Figure 2.3, with the difference that 2 root-raised cosine (RRC) pulse shaping filters (PSFs) are inserted between the symbol mapper and the multipliers separately in the I and Q arms, in order to reduce the inter-symbol interference (ISI). The RRC PSF is implemented as a FIR filter by using distributed arithmetic (DA) technique to eliminate the use of multipliers for power optimization and enhanced speed. The interesting feature of this baseband modulator is parameterization such that modulation scheme can be selected as either QPSK or its other 3 variants of Differentially-encoded QPSK (DQPSK), $\pi/4$ DQPSK and Offset QPSK (OQPSK), and roll-off factor α of RRC filter can be selected as either 0.22, 0.35, 0.5 or 0.9 according to the need of different standards such as IS-95, UMTS and WCDMA. This can be done by integrating all the aforementioned modulation schemes and RRC filters (with different α) into a single symbol mapper and PSF respectively with an appropiate hardware multiplexing. In addition, this reconfigurable baseband modulator would be advantageous if the future transmitter requires small size, low power consumption and high data rate multi-standard SDR.

Another interesting method of FPGA implementation of QPSK modulator is partial reconfiguration (Bhandari et al., 2009). In the modulator proposed by (Hatai and Chakrabarti, 2010), only one modulation is used at any point of time therefore resource utilized by all other modulations becomes overhead. However the partial reconfiguration can reduce this overhead by providing download of new symbol mapper (and PSF) altered for different modulation scheme. Besides, the resulted download time is smaller than the time for entire FPGA reconfiguration.

2.3.3 Quadrature Amplitude Modulation (QAM)

Assume that $p(t)$ is a rectangular pulse of duration T_{sym} with amplitude of 1, and modulation order is M_o (= 2^k, for even k = 2, 4, 6, …), the signal waveforms of rectangular QAM can be represented as (Proakis and Salehi, 2008):

$$s_i(t) = p(t)\left[A_{i,I}\sin\left(2\pi f_c t + \theta_0\right) + A_{i,Q}\cos\left(2\pi f_c t + \theta_0\right)\right], 0 \le i \le M_o - 1 \quad (2.12)$$

where f_c is carrier frequency in Hertz (Hz) and θ_0 is phase offset in radian (rad).

The first and second components of equation (2.12) in terms of sin ($2\pi f_c t + \theta_0$) and cos ($2\pi f_c t + \theta_0$) can be represented as signal amplitudes of I and Q channels:

$$A_{i,I} = (2i + 1 - M_o) = \pm 1, \pm 3, \pm 5, \ldots, \pm(M_o - 1), 0 \le i \le M_o - 1 \quad (2.13)$$

$$A_{i,Q} = (2i + 1 - M_o) = \pm 1, \pm 3, \pm 5, \ldots, \pm(M_o - 1), 0 \le i \le M_o - 1 \quad (2.14)$$

However, the signal amplitudes of $A_{i,\,I}$ and $A_{i,\,Q}$ can be scaled by a factor μ to become the desired symbol amplitudes, as given by:

$$A_{sym,I} = \mu(2i + 1 - M_o) = \pm\mu, \pm 3\mu, \pm 5\mu, \ldots, \pm(M_o - 1)\mu, 0 \le i \le M_o - 1 \quad (2.15)$$

$$A_{sym,Q} = \mu(2i + 1 - M_o) = \pm\mu, \pm 3\mu, \pm 5\mu, \ldots, \pm(M_o - 1)\mu, 0 \le i \le M_o - 1 \quad (2.16)$$

For $M_o \geq 4$, $A_{sym,\,I} \neq 0$ and $A_{sym,\,Q} \neq 0$ for $0 \leq i \leq M_o - 1$; thus both I and Q channels are involved, therefore M_o ($\geq$ 4)-QAM is 2-D digital modulation.

The recent researches related to QAM are designs and implementations of QAM modem involving FPGA platform (Asif et al., 2006; Vu et al., 2010).

The I and Q symbols of QAM are indeed two PAM symbols, but independent to each other due to the orthogonality of two carrier signals sin ($2\pi f_c t$) and cos ($2\pi f_c t$).

Therefore the combinational platform of microcontroller, DSP processor and FPGA proposed by (Asif et al., 2006) can also be used to implement QAM modem. Indeed, the QAM modem system is mainly implemented using DSP processor, and FPGA is used to buffer data interfaced with RF conversion and to drive clocking interface. The total DSP computation time for 4-QAM transmitter and receiver is 16600 CPU cycles where each CPU cycle is referred as system clock period of 5 ns (Asif et al., 2006). This DSP computation time is quite high (although not the most optimized one) thus the supported data rate is limited. Therefore the DSP processor is not recommended for QAM modem with high data rate as required by SDR nowadays.

The FPGA implementation of 16-QAM modem proposed by (Vu et al., 2010) is also similar to Figure 2.3, with the differences: 2 RRC PSFs (each in I and Q arms) are inserted between symbol mapper and multipliers in modulator, the multipliers are replaced by a complex multiplier (for both modulator and demodulator), and 2 RRC matched filters (MFs) (each in I and Q arms), a timing synchronization, an adaptive equalizer and a phase recovery (also feedback to the adaptive equalizer) are inserted between LPFs and decision circuits in demodulator. The pair of RRC PSF and MF (each in I and Q arms) reduces the ISI effect and provides detection of optimum sampling point. The timing synchronization is based on the minimum differential detection in which the differential signal controls a sampler to trigger at optimum sampling point corresponding to peaks of the matched-filtered I/Q baseband signals. The adaptive equalizer employs least-mean square (LMS) algorithm to minimize the difference between input and output of decision circuit. Whereas the phase recovery de-rotates the I/Q baseband signals to form the recovered I/Q symbols.

2.4 Digital Pulse Shaping Filter (PSF)

The communication channel is indeed a bandwidth-limited environment that must be matched with the bandwidth of the transmitted signal for effective and efficient data transmission through the channel (Rouphael, 2009). Thus, pulse shaping techniques are employed on the mapped symbols in order to reduce the infinity bandwidth down to the level that is under constraint of the channel bandwidth (Gentile, 2002).

The common pulse shaping techniques are the convolutions of symbols with rectangular pulse, sinc pulse, raised cosine (RC) pulse, cosine pulse, triangular pulse, and Gaussian pulse (Rouphael, 2009). However, RC pulse shaping is more prevalent because the RC pulse can be square-rooted to become root-raised cosine (RRC) pulse which can be realized identically in both of the transmitter and receiver as pulse shaping filter (PSF) and matched filter (MF) respectively (Gentile, 2002). Therefore, the RRC pulse shaping technique is more efficient in terms of cost and power. The RC (and RRC) PSF will be presented in detailed in the following subheadings.

2.4.1 Raised Cosine (RC) Pulse Shaping Filter (PSF)

Assume that $p_{rect}(t)$ is a rectangular symbol pulse of period T_{sym} , as given by:

$$p_{rect}(t) = \begin{cases} A_{sym}, |t| \leq T_{sym}/2 \\ 0, |t| > T_{sym}/2 \end{cases} \tag{2.17}$$