

STUDY ON FPGA BASED IIR FILTER USING QUANTITATIVE APPROACH

OW SHEN WAH

UNIVERSITI TEKNOLOGI MALAYSIA

STUDY ON FPGA BASED IIR FILTER USING QUANTITATIVE APPROACH

OW SHEN WAH

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer and Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2012

Specially dedicated to my beloved family, lecturers and friends
for the guidance, encouragement and inspiration
throughout my journey of education

ACKNOWLEDGEMENT

First and foremost, I would like to take this opportunity to express my deepest gratitude to my project supervisor, Prof. Dr. Mohamed Khalil bin Hj Mohd Hani for his great encouragement, guidance and sharing of knowledge during the process of completing this project. Without his constant motivation and supervision with valuable suggestion, this project would not been a success.

Besides, I wish to thank my postgraduate course-mates for their cooperation and information sharing in completing this project. Yet, not to forget my fellow friends for their care and moral support when it was most required.

Lastly, yet importantly, my highest appreciation goes to my beloved family for their understanding and blessing from the beginning up to now. Special thanks to my partner, Mr. Lim Hsiu Fuh, who always been there and stood by me through the good and bad times. The precious support and encouragement will be fondly remembered.

ABSTRACT

The main goal of this project is to design a digital filter which is compatible between simulation tool (software) and hardware implementation using Matlab and Quartus II. The filter is realized in Direct Form II biquad architecture to achieve scalable and expandable design which can be cascaded if necessary. Filter quantization procedure is presented based on the finite word-length arithmetic to determine the bit length of the filter's digital parameters as accurate as possible. With the resulting bit-true model, hardware design implementation using Verilog RTL for Altera FPGA is then performed. A biquad filter in FPGA, using numerous hardware realization methods, namely fully combinational, combinational-sequential and bit serial are designed and performance analysis is carried out by comparing their efficiency and area. The design is then optimized further to be more cost-effective by implementing the bit-serial arithmetic architecture where multipliers are replaced with lookup table (LUT). According to the simulation result, fully-combinational is the fastest and the most expensive approach with unconstrained resource utilization while combinational-sequential compromises between area and speed with limited resources. On the other hand, bit-serial model achieves highest maximum frequency with lowest propagation delay between registers. Optimization with LUT usage is a hybrid model of fully-combinational and bit-serial which it balances up the pros and cons by improving the maximum frequency of fully combinational and reducing the total execution time of bit-serial approach.

ABSTRAK

Kajian ini dilakukan bertujuan mengkaji cara-cara untuk mereka-bentuk digit penapis moden yang serasi antara alat simulasi dan pelaksanaan perkakasan dengan menggunakan MATLAB dan Quartus II. Penapis itu akan direalisasikan dalam biquad seni bina untuk mencapai reka bentuk berskala yang dapat diperkembangkan secara latta sekiranya ada keperluan. Prosedur pengkuantuman penapis akan dicadang berdasarkan panjang-perkataan terhad aritmetik untuk menentukan panjang-oktet parameter digital penapis setepat mungkin. Dengan model-bit yang benar, reka bentuk perkakasan akan dilaksanakan dengan menggunakan Verilog RTL untuk Altera FPGA. Selain itu, penapis modular biquad akan dilaksanakan dengan menggunakan beberapa kaedah untuk merealisasikan perkakasan, iaitu logik gabungan, gabungan-berjjukan and bit-siri dengan membandingkan kecekapan dan luas permukaan mereka. Reka bentuk akan dioptimumkan supaya lebih kos efektif dengan melaksanakan bit-siri berseni aritmetik di mana pengganda akan digantikan dengan jadual lookup (LUT). Menurut hasil simulasi, logik gabungan adalah keadah yang paling cepat dan paling mahal dengan penggunaan sumber yang tidak dikekang manakala gabungan-berjjukan kompromi antara keluasan dan kelajuan dengan sumber yang terhad. Sebaliknya, bit-siri model mencapai frekuensi maksimum tertinggi dengan lengah perambatan yang terendah. Pengoptimuman dengan penggunaan LUT adalah satu model hybrid antara logik gabungan dan bit-siri dengan meningkatkan kekerapan maksimum daripada logic gabungan dan mengurangkan jumlah masa pelaksanaan daripada bit-siri.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xii
	LIST OF FIGURES	xiii
	LIST OF ABBREVIATIONS	xvii
	LIST OF APPENDICES	xviii
1	INTRODUCTION	1
	1.1 Introduction to Digital Filter	1
	1.2 Introduction to FPGA	4
	1.3 Project Motivations and Objectives	5
	1.4 Scope of Work	6
2	LITERATURE RESEARCH	8
	2.1 Principle of Filter Operations	8
	2.2 Quantization and Finite Wordlength Effects	11
	2.3 FPGA Implementations	16
	2.3.1 Baugh-Wooley Multiplier	19
	2.4 Biquad Block of a Digital Filter	20
	2.5 Biquad Architecture Implementations	22

2.5.1	Fully Combinational	23
2.5.2	Combinational-Sequential Implementation	24
2.5.3	Word-serial Implementation	25
2.5.4	Bit-serial Implementation	26
2.6	Optimization using Bit Serial Arithmetic	29
3	METHODOLOGY AND IMPLEMENTATION PLAN	31
3.1	IIR Digital Filter Design Considerations	31
3.2	Bit-true Model Realization	33
3.2.1	Filter Response Simulation	34
3.2.2	Coefficient Quantization	35
3.2.3	Intermediate Wordlength	36
3.2.4	Bit true Filter Modelling	38
3.3	FPGA Implementations	41
3.3.1	Biquad Architecture Implementation in FPGA	42
3.3.2	Lookup Table Implementation in FPGA	42
3.4	Design Optimization	44
3.5	IIR Bandstop Filter Specification	45
4	DESIGN AND MODELING	48
4.1	High Level Design of Biquad IIR Filter	48
4.1.1	Synchronous Dataflow Graph (SDFG) of IIR Filter	49
4.2	Fully-Combinational Biquad Model	50
4.2.1	Scheduling and RTL Code of Fully Combinational Model	50
4.2.2	Algorithm and ASM-Chart of Fully Combinational Model	51
4.2.3	Datapath Unit (DU) Design using General Multiplier	52
4.2.4	Baugh-Wooley Multiplier	53
4.2.5	Datapath Unit (DU) Design using Baugh-Wooley Multiplier	56
4.2.6	Verilog Coding of Fully Combinational Model	56

4.3	Combinational-Sequential Biquad Model	57
4.3.1	Scheduling and RTL code of Combinational-Sequential Model	57
4.3.2	ASM-Chart of Combinational-Sequential Model	58
4.3.3	Datapath Unit (DU) Design	59
4.3.4	Control Unit (CU) Design	61
4.3.5	Top level Design of Combinational-Sequential Model	63
4.3.6	Verilog Coding of Combinational-Sequential Model	63
4.4	Bit-Serial Biquad Model	64
4.4.1	ASM Chart of Bit-Serial Model	64
4.4.2	Bit-Serial Building Blocks	66
4.4.2.1	Bit-Serial Adder	66
4.4.2.2	Bit-Serial Two's Complementor	67
4.4.2.3	Serial by Parallel Multiplier (SPM)	67
4.4.2.4	Bit-Serial Subtractor	69
4.4.3	Datapath Unit Design	69
4.4.4	Control Unit Design	71
4.4.5	Top level Design	72
4.4.6	Verilog Coding of Bit-Serial Model	73
4.5	Bit-Serial Arithmetic Model using Lookup Tables	73
4.5.1	Algorithm of Lookup Tables	74
4.5.2	ASM-Chart of Bit-Serial Arithmetic using Lookup Tables	75
4.5.3	Datapath Unit Design	76
4.5.4	Verilog Coding of Bit-Serial Arithmetic using Lookup Table	76
5	RESULTS AND DISCUSSION	77
5.1	Quantization Effect of Digital Filter	77
5.1.1	Simulation Results of Quantization Comparison	78

5.1.2	Discussion on Quantization Effect of Digital Filter	80
5.2	Filter Verification Using MATLAB simulation	80
5.3	Verification Using TestBench	82
5.3.1	Verilog Coding of TestBench for Biquad IIR Bandstop Filter	83
5.4	Simulation Results of Fully-Combinational Biquad Model	84
5.4.1	Simulation Result of Fully-Combinational Model using General Multiplier	84
5.4.2	Simulation Result of Baugh Wooley Multiplier	85
5.4.3	Simulation Result of Fully Combinational Model using Baugh Wooley Multiplier	87
5.4.4	Discussion on Fully-Combinational Model	88
5.5	Simulation Results of Combinational-Sequential Biquad Model	89
5.5.1	Simulation Result of Datapath Unit in combinational-sequential Biquad Model	89
5.5.2	Simulation Result of Control Unit in Combinational-Sequential Model	90
5.5.3	Simulation Results of Top-Level of Combinational-Sequential Model	90
5.5.4	Discussion on Combinational-Sequential Model	91
5.6	Simulation Results of Bit-Serial Biquad Model	92
5.6.1	Simulation Result of Bit-serial Adder	92
5.6.2	Simulation Result of Bit-Serial Two's Complementor	93
5.6.3	Simulation Result of Serial Parallel Multiplier (SPM)	94
5.6.4	Datapath Unit Simulation Results	95
5.6.5	Simulation Result of Control Unit	95
5.6.6	Top Level Simulation Results	96
5.6.7	Discussion on Bit-Serial Model	96

5.7.	Simulation Result of Bit-Serial Arithmetic using Lookup Tables	97
5.7.1	Simulation Results	97
5.7.2	Discussion on Bit-Serial Arithmetic using LUT	98
5.8	Summary of Comparison	104
5.9	Design Challenges	105
5.9.1	Rounding Issue	105
5.9.2	Bit-order in Bit-Serial Implementation	106
5.9.3	Delay Element in Bit-Serial Arithmetic using LUT	106
6	CONCLUSION AND FUTURE RECOMMENDATIONS	107
6.1	Conclusion	107
6.2	Future Recommendations	108
	REFERENCES	109
	Appendices A - F	112 - 146

LIST OF TABLES

TABLE NO.	TITLE	PAGE
3.1	Bandstop Filter Quantized Coefficient Values Floating-point to Fixed point conversion	46
4.1	Two's complement and one's complement representations	53
4.2	Datapath Unit Assigned Register Arithmetic Operations	60
4.3	RTL-CS for the Combinational Sequential Control Unit	62
4.4	RTL-CS for the Bit Serial Control Unit	71
5.1	Performance Analysis of Fully-Combinational using General Multipliers	85
5.2	Performance Analysis of Fully-Combinational using BW Multipliers	87
5.3	Comparison of performance of fully combinational model using general purpose multiplier and Baugh Wooley multiplier	88
5.4	Performance Analysis of Combinational-Sequential Model	91
5.5	Bit-wise operation of bit-serial adder	92
5.6	Performance Analysis of Combinational-Sequential Model	96
5.7	LUT Values Used During Bit Serial Filter Operation	99
5.8	Summary of Comparison of Different Models of Design	104

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Block Diagram of a Digital Filter	2
1.2	Internal structure of a generic FPGA (courtesy Xilinx, Inc.)	5
2.1	Time and frequency Domain Response of a Bandpass Filter	9
2.2	Template of a Band-stop Filter	10
2.3	An ADC-based digital conversion system	12
2.4	A saturating arithmetic operational behavior	13
2.5	Truncation vs. rounding in fixed-point representation	13
2.6	Arithmetic Error in a fixed-point MAC unit	14
2.7	Frequency of a FIR with coefficients determined using SA	15
2.8	(a) SFG & (b) DFG of a DSP recursion $y(n) = ay(n - 1) + x(n)$	17
2.9	Example of (a) before (b) after pipelining	17
2.10	IIR filter with look-ahead arithmetic	18
2.11	The parametric biquad (second-order) building block	22
2.12	Fully combinational implementation of a biquad system	23
2.13	Scheduling of combinational implementation	24
2.14	Combinational-sequential implementation of a biquad system	24
2.15	Scheduling diagram of combinational-sequential implementation	25
2.16	Word-serial Implementation	26
2.17	Bit-serial Implementation of a biquad section	26
2.18	Scheduling of bit-serial implementation with 16 bit coefficient length and 20-bit internal data path	27
2.19	Bit Serial Adder	27
2.20	Bit-serial multiplier for three-bit numbers	28
2.21	IIR Digital Filter using Bit-Serial Arithmetic Architecture	30
3.1	A fixed-point filter design procedure	32

3.2	GUI of FDA tool	34
3.3	Quantization of Filter Coefficient in FDA Tool	36
3.4	Nodes at the biquad to find maximum swing shown by arrow	37
3.5	Coefficient scaling in FDA tool	37
3.6	Filter intermediate word length specification in FDA tool	38
3.7	Direct-Form II, Second Order Sections filter structure	39
3.8	Bit-True model of the filter in Simulink	40
3.9	Look up (LUT) implementation of a digital filter	43
3.10	Filter Designed in FDA Tool	45
3.11	FDA Tool Generated Coefficients	45
3.12	FDA Tool Generated Quantized Coefficients	46
3.13	Fixed Point Coefficients stored in RF.mif	47
4.1	SDFG of biquad bandstop IIR filter	49
4.2	Scheduling and RTL code of fully combinational model	50
4.3	ASM Chart for Fully Combinational Model	51
4.4	Functional Block Diagram of Fully Combinational Model	52
4.5	Tabular form of bit-level Baugh Wooley Multiplication	54
4.6	Functional block diagram of 16-bits Baugh Wooley Multiplier	55
4.7	Functional Block Diagram of Fully Combinational Model using Baugh-Wooley Multipliers	56
4.8	Scheduling and RTL code of Combinational-Sequential	57
4.9	ASM Chart for Combinational-Sequential Model	58
4.10	Datapath functional block diagram for Combinational-Sequential Model	59
4.11	Top Level Unit of Combinational-Sequential Biquad Model	63
4.12	ASM chart of Bit-Serial Model	65
4.13	Functional block diagram of bit-serial adder	66
4.14	Bit-Serial Two's Complementor	67
4.15	Functional block diagram of Serial Parallel Multiplier (SPM)	68
4.16	Bit Serial Subtractor	69
4.17	Functional Block diagram of DU	70
4.18	Top Level Unit of Bit-Serial Biquad Model	72
4.19	ASM Chart for Bit Serial Biquad using LUT	75

4.20	Functional block diagram for Bit-Serial Biquad Using Lookup Table	76
5.1	Magnitude response of a bandstop IIR Filter (reference filter)	78
5.2	Magnitude and phase responses of reference IIR filter and quantized filter with 4 bits word length for filter coefficients	78
5.3	Magnitude and phase responses of reference IIR filter and quantized filter with 7 bits word length for filter coefficients	79
5.4	Magnitude and phase responses of reference IIR filter and quantized filter with 8 bits word length for filter coefficients	79
5.5	Magnitude and phase responses of reference IIR filter and quantized filter with 16 bits word length for filter coefficients	79
5.6	Magnitude Response of Biquad IIR Bandstop Filter	81
5.7	Input Signal of 10kHz and 12kHz Sinusoids	81
5.8	Filtered Bandstop Signal	82
5.9	TestBench Simulation in ModelSim	83
5.10	TestBench Expected Input and Output Values (hex)	83
5.11	Simulation results of fully-combinational biquad model using general purpose multipliers	84
5.12	Simulation result of 16-bits BW multiplier	86
5.13	Simulation result of 8-bits BW multiplier	86
5.14	Simulation results of fully-combinational biquad model using Baugh-Wooley Multipliers	87
5.15	Simulation Results of Datapath Unit of Combinational-Sequential Biquad Model	89
5.16	Simulation Results of Control Unit of Combinational-Sequential Biquad Model	90
5.17	Top Level Simulation Results of Combinational-Sequential Biquad Model	90
5.18	Simulation result of bit-serial adder	92
5.19	Simulation result of bit-serial two's complementor	93
5.20	Simulation result of serial-parallel multiplier	94
5.21	Simulation Result of Datapath Unit of Bit-Serial	95
5.22	Simulation Result of CU of Bit-Serial model	95
5.23	Simulation result of top-level of bit-serial model	96

5.24	Simulation Results of Bit-Serial Arithmetic using Lookup Table	97
5.25	Simulation result of first stage in bit-serial arithmetic using lookup table	100
5.26	Simulation result of second stage of bit-serial arithmetic using lookup table	101
5.27	Simulation result of third stage of bit-serial arithmetic using lookup table	102
5.28	Simulation result of negative numbers of bit-serial arithmetic using lookup table	103
5.29	Difference of one between expected and simulated value	105

LIST OF ABBREVIATIONS

ADC	-	Analog-to-Digital Converter
ASIC	-	Application-specific Integrated Circuit
BW	-	Baugh-Wooley
CU	-	Control Unit
DU	-	Datapath Unit
DAC	-	Digital-to-Analog Converter
DSP	-	Digital Signal Processing
FIR	-	Finite Impulse Response
FPGA	-	Field-programmable Gate Array
FT	-	Fourier Transform
GUI	-	Graphical User Interface
HDL	-	Hardware Descriptive Language
IIR	-	Infinite Impulse Response
LTI	-	Linear Time Invariant
LSB	-	Least Significant Bit
LUT	-	Look-up
MAC	-	Multiplication-Accumulation Unit
MSB	-	Most Significant Bit
RAM	-	Random Access Memory
RTL	-	Register-transfer Level
SA	-	Simulated Annealing
SNR	-	Signal-to-noise Ratio
VLSI	-	Very-Large-Scale Integration

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A-1	Verilog program of Fully Combinational (General multiplier)	112
A-2	Verilog program of Fully Combinational (BW multiplier)	114
A-3	Verilog program of 16-bits BW Multiplier	116
B-1	Verilog program of Datapath Unit (DU) for Combinational- Sequential Model	121
B-2	Verilog program of Control Unit (CU) for Combinational- Sequential Model	123
B-3	Verilog program of Top-Level for Combinational- Sequential Model	124
C-1	Verilog program of Datapath Unit (DU) for Bit- Serial Model	125
C-2	Verilog program of Control Unit (CU) for Bit-Serial Model	127
C-3	Verilog program of Top-Level for Bit-Serial Model	128
C-4	Verilog program of Bit-Serial Adder	129
C-5	Verilog program of Bit-Serial Two's Complementor	130
C-6	Verilog program of Serial-Parallel Multiplier	131
C-7	Verilog program of Bit-Serial Subtractor	132

C-8	Verilog program of Bit-Serial Delay	133
C-9	Verilog program of Shift-Registers	134
D-1	Verilog program of Bit-Serial Arithmetic using LUT	135
D-2	Perl-Script of LUT Generator	138
E-1	Matlab Code for Biquad IIR Bandstop Filter	140
F-1	Testbench for Biquad IIR Bandstop Filter	141

CHAPTER 1

INTRODUCTION

Digital filter has been a subject of interest for Digital Signal Processing (DSP) systems due to its enormous technology impacts and limitless applications. With the advancement of silicon scaling and digital architecture, Field Programmable Grid Array (FPGA) is also no longer a stranger in the electronics field. In this chapter, an overview of digital filter and FPGA is presented. This is followed by the motivation, objectives and the scope of work that would be achieved in the project.

1.1 Introduction to Digital Filter

A digital filter is a *Linear Time Invariant (LTI)* system, if it satisfies the properties below [1] :

- 1) Linearity – If a scaled input $Kx(n)$ produces an output $Ky(n)$ (where K is any arbitrary constant), the system satisfies the condition of homogeneity. If the output is $K_1y_1(n) + K_2y_2(n)$ when the input is $K_1x_1(n) + K_2x_2(n)$, then the system satisfies the superposition property. The system is said to be linear if it fulfills both homogeneity and superposition properties.

- 2) Time Invariant - If the output is $y(n-M)$ when the input is delayed by M samples, that is, when the input is $x(n-M)$, the system is said to be time-invariant.

LTI digital filter system translates input-to-output relationship by performing numerical calculations on discretely sampled signals. A block diagram of a digital filter system is shown in Figure 1.1. In order for the signals to work in the digital domain, first, the analog input signal $x(t)$ must be sampled and digitized using Analog-to-Digital Converter (ADC). This digitized form is a binary representation of the input voltage at the instant of sampling, n . The model for digital filter system can then be described by a circuit diagram showing the interconnection of its components, which are the delay elements, multipliers, and accumulators. The digital filter will be capable of performing numerical calculations on resulting binary numbers, such as multiplying the input values by constants (coefficients) and sum up the products together to produce the output $y(n)$. The output is finally converted back to analog via a Digital-to-Analog Converter (DAC).

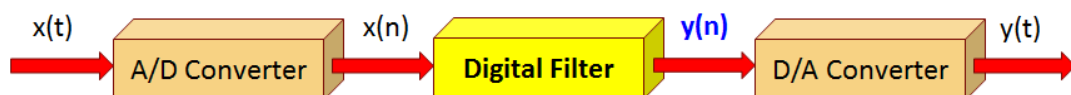


Figure 1.1 Block Diagram of a Digital Filter

Digital filters can be divided into two broad categories, namely FIR and IIR filters. For FIR filters, the filter output depends on present and previous input samples x^n to x^{n-p} . The inputs are delayed using delay elements in the circuit, multiplied by coefficients a_0 to a_p and added together. Characteristic equation of a typical FIR filter is

$$y^n = a_0x^n + a_1x^{n-1} + \dots + a_px^{n-p} \quad (1-1)$$

Where p is the filter order, $a_0 \dots a_p$ are coefficients, x^n is the filter input at the time step n , and y^n is the filter output at the time step n .

Meanwhile, for the IIR filter, output depends not just on a set of input samples, x^n to x^{n-p} , but also on a set of previous output, y^{n-1} to y^{n-p} . These are multiplied by coefficients $a_0 \dots a_p$ and $b_1 \dots b_p$ before being added together. In other words, it is recursive as the output needs to feedback into the input for computation. This can be described by its characteristic equation as given by

$$y^n = a_0 x^n + a_1 x^{n-1} + \dots + a_p x^{n-p} + b_1 y^{n-1} + \dots + b_p y^{n-p} \quad (1-2)$$

Compared to FIR, phase response of an IIR filter is non-linear, and the hardware implementation will be more complex. Most IIR filters can be designed using an analog filter model, such as Butterworth, Chebyshev, Elliptic. In this paper, main focus will be on IIR type digital filter.

Digital filters are increasingly popular in digital processing applications as they offer numerous advantages [2] such as reproducible response, not temperature sensitive, and programmable, which is superior over the analog filters. Performance-wise, digital filters offer lower passband ripple, faster transition, higher stopband attenuation, linear phase in time domain over its analog counterpart. Besides, digital filters are able to take full advantage of the advanced submicron technology that IC chip makers are able to offer today. Nevertheless, the downside of digital filter is the aliasing of digital signal caused by sampling effects. Unlike analog filter, digital filter are unable to pass power and requires a power supply. It might also encounter interference where out-of-band signals are frequency shifted and appear in the passband. Applications of digital filtering are enormous, including noise suppression in consumer electronics, selectively filtering electrical signals such as brain, heart, neurological signals from human body in biomedical applications, image enhancements of high frequency image elements, bandwidth limiting of intended television and radio signals in communications and many more.

1.2 Introduction to FPGA

Field-programmable gate array (FPGA) consists of [3] field-programmable logic (FPL) that offers programmability interconnection or ‘glue logic’ that can be customized for specialized purposes. Unlike Application-specific Integrated Circuits (ASIC’s), FPGA’s are not hard-coded and can be freely programmed using the myriads of hardware and software platform available. It is a great candidate for digital filtering hardware development and implementation as FPGAs are well catered for datapath design.

An internal building block of a generic FPGA is shown in Figure 1.2 [4]. It consists largely of programmable logic blocks that contain arrays of combinatorial blocks and flip-flops to be cond by the designer. In addition, large amounts of static Random Access Memory (RAM) are integrated as FPGA logic is often used in conjunction with memory based on consumer trends. Clock conditioning in the forms of Delay Locked Loops (DLLs) and Phase Locked Loops (PLLs) are also supported inside the same silicon chip. Flexibility of the input/output (IO) blocks behind the chip pads is another feature of FPGA, which means that the IO ports can be freely cond as input, output, or both at the same time.

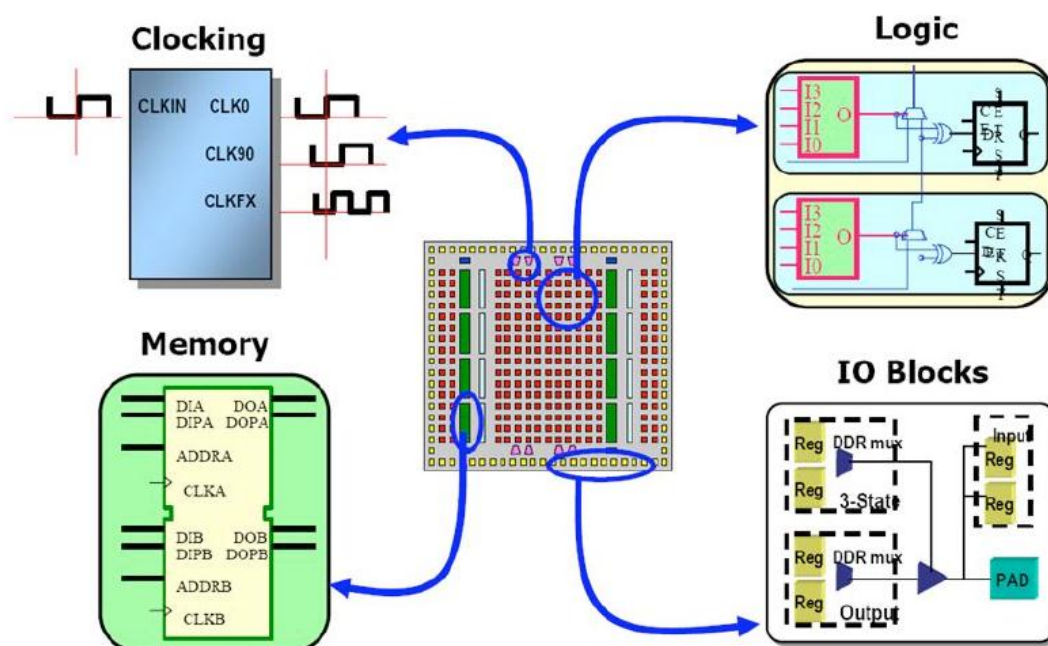


Figure 1.2 Internal structure of a generic FPGA (courtesy Xilinx, Inc.)

Advantages of FPGA approach to digital filter implementation includes higher sampling rates than traditional DSP chips, lower cost than ASIC for moderate volume applications, and more flexibility than the alternate approaches, leading to a shorter time-to-market especially development time. FPGA also allows the use of underlying FPGA fabric of localized memory in the form of lookup tables (LUTs) and flip-flops along with the logic LUT resource, allowing the user the choice for design optimization. On the contrary, limitations of FPGA are related to overhead imposed by programmability and constraints imposed by the architecture. Also, there are limitations on the logic function which may be implemented in each logic block based on the architecture. FPGA will also introduce routing delays in the array.

1.3 Project Motivations and Objectives

Research on digital filter implementation over the years has concentrated on custom implementation using various ASIC technologies. Several potential shortcomings of custom Very-Large-Scale Integration (VLSI) ASIC approach

1. Lack of flexibility in custom approach – Custom devices are often suited only for use in particular application, not reconfigurable.
2. Forestall the cost effective evaluation – Only high volume applications or extremely low volume applications can justify the expense of developing a full custom solution.
3. Lack of adaptability – Does not allow user to modify the function of a device.

Although these problems can be overcome with sufficient forethought, the costs in performance, design complexity, and additional design time often preclude flexible solutions. Field Programmable Gate Arrays (FPGAs) can be used to alleviate some of the problems with custom approach as they are programmable logic devices. In-system programmable allows modification of the operation of the device through simple reprogramming.

Therefore, the objective of this project is to illustrate the approaches in designing modern digital filters in FPGA as listed below

- (i) Fully combinational
- (ii) Combinational-Sequential
- (iii) Bit-serial
- (iv) Bit-serial Arithmetic using LUT

All design will be modeled and synthesized using Quartus II and quantitative measurement will be provided in terms of usability, area, and speed. In order to further improve the design and make full use of the capability of FPGA, multipliers are replaced with look-up s and adder-subtractor to achieve cost effectiveness in the filter implementation.

1.4 Scope of Work

This project starts off with a brief introduction of digital filter and FPGA, followed by the motivations and objectives of this project in Chapter 1. In Chapter 2,

there will be literature research to understand the underlying principle of digital filtering and the limitations of digital quantization, before reviewing the proposed biquad hardware architecture methodologies and its FPGA implementation in this project. As progress into Chapter 3, the methodology and implementation plan to carry out the filter determination and hardware realizations will be discussed, at the same time explaining about the software tools, namely Matlab & Quartus II, which will be using extensively to aid the analysis and development. Based on the simulation, bit true model of sample digital filter will be extracted and the biquad hardware design will be implemented based on the proposed methodologies in Chapter 4. The methodologies are fully combinational, combinational-sequential, word serial, and bit serial. Besides, the filter design will be further optimized to be more cost-effective using bit-serial arithmetic approach to eliminate the usage of general purpose multipliers and substituting it with stored computed coefficients. Each hardware design will be explored by realizing its RTL designs. Subsequently, the results of the hardware implementation will be discussed, by comparing the performance and cost of each method in Chapter 5. Finally, the project is summarized with a conclusion and future recommendations in Chapter 6.

REFERENCES

- B. A. Sheno, Introduction to Digital Signal Processing and Filter Design, John Wiley & Sons, Inc., 2006.
- [1] S. Winder, Analog and Digital Filter Design(2nd Edition), United States of America Newnes Publications, 2002.
- [2] R. W. e. al., FPGA-based implementation of complex signal processing systems, John Wiley & Sons, Ltd, 2008.
- [3] J. Serrano, "Introduction to FPGA design," *CAS - CERN Accelerator School Course on Digital Signal Processing*, pp. 231-247, 2007.
- [4] C. R. a. M. S. E. Boutillon, "Probability-driven simulated annealing for optimizing digital FIR filters," *Adaptive and Multilevel Metaheuristics, SCI*, vol. 136, pp. 77-93, 2008.
- [5] F. T. Arthur Williams, Electronic Filter Design Handbook, Fourth Edition, McGraw-Hill Handbooks, July 10, 2006.
- [6] C. J. W. Alan V. Oppenheim, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 957 - 976, Aug. 1972.
- [7] H. J. Y. J. C. Byung Wook Jung, "Finite Wordlength Digital Filter Design using Simulated Annealing," *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, vol. 2, pp. 546 - 550, 26-29 Oct. 2008.
- [8] R. I. a. B. L. S. Chen, "Digital IIR Filter Design Using Adaptive Simulated Annealing," *Digital Signal Processing*, vol. 11, no. 3, pp. 241-251, July 2001.
- [9] D. Z. R. W. Mehler, "Architectural Synthesis of Finite Impulse Response Digital Filters," *15th Symposium on Integrated Circuits and Systems Design 2002*, pp. 20-25, Sep. 2002.
- [10] J. M. G. L. Y. Y. Roger Woods, FPGA-based Implementation of Signal Processing Systems, United Kingdom A John Wiley and Sons, Ltd., Publication, 2008.
- [11] K. a. K. P. Tracy C. Denk, "Exhaustive scheduling and retiming of digital

- [12] signal processing systems," *IEEE Transactions on Circuits and Systems-II Analog and Digital Signal Processing*, vol. 45, no. 7, Jul. 1998.
- A. R. H. S. a. J. M. Ravinder Kaur, "Design and Implementation of High
- [13] Speed IIR and FIR Filter using Pipelining," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, April 2011.
- U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate*
- [14] *Arrays* (3rd Edition), New York Springer Berlin Heidelberg, 2007.
- C. W. B. Baugh, "A Two's Complement Parallel Array Multiplication
- [15] Algorithm," *IEEE Transactions on Computers*, Vols. C-22, no. 12, pp. 1045-1047, 1973.
- S. Y. C. Pradabpet, "Design and implementation of biquad digital filter," *The*
- [16] *9th Asia-Pacific Conference on Communications 2003 (APCC 2003)*, vol. 3, Sep. 2003.
- S. R. a. S. M. F. N. Sedaghati-Mokhtari, "Hardware Implementation Analysis
- [17] for Digital Filters," *Iranian Conference on Electrical Engineering (ICEE) 2006*, 2006.
- N. M. M. S. M. F. Hamed Hollisaz, "A Quantitative Approach to Digital Filter
- [18] Implementation," *Microelectronics, 2005. ICM 2005. The 17th International Conference on*, pp. 160 - 164, 2005.
- B. Brandt, "A low-power area efficient digital filter for decimation and
- [19] interpolation," *IEEE JSSC* 29(6), 1994.
- A. G. Dempster, "Use of Minimum-Adder Multiplier Blocks in FIR Digital
- [20] Filters," *IEEE Transactions on Circuits and Systems-II; Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 569-577, 1995.
- A. G. D. a. M. D. Macleod, "IIR Digital Filter Design Using Minimum Adder
- [21] Multiplier Blocks," *IEEE Transactions on Circuits and Systems-II; Analog and Digital Signal Processing*, vol. 45, no. 6, pp. 761-763, 1998.
- J. A. A. G. Monica Arroyuelo, "FPGA-Based Digital Filters Using Bit-Serial
- [22] Arithmetic," *CACIC*, 2007.
- "MATLAB - The Language Of Technical Computing," The MathWorks, Inc.,
- [23] [Online]. Available <http://www.mathworks.com/products/matlab/>. [Accessed 2 January 2012].

- "Quartus II Web Edition Software," Altera Corporation, [Online]. Available
[24] <http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html>. [Accessed 2 January 2012].
- "Use FDA Tool with DSP Systems," Mathworks, [Online]. Available
[25] <http://www.mathworks.com/help/toolbox/dsp/ug/bsva2f6.html>. [Accessed 2 Jan 2012].
- "Discrete-time, second-order section, direct-form II filter," Mathworks,
[26] [Online]. Available
<http://www.mathworks.com/help/toolbox/signal/ref/dfilt.df2sos.html>.
[Accessed 2 Jan 2012].
- "Biquad Filter," Mathworks, [Online]. Available
[27] <http://www.mathworks.com/help/toolbox/dsp/ref/biquadfilter.html>. [Accessed 2 Jan 2012].
- P. G. a. J. C. C.P. Lerouge, "A fast 16-bit NMPS parallel multiplier," *IEEE*
[28] *Journal of Solid-State Circuits*, vol. 19, no. 3, pp. 338-342, Mar. 1984.
- A. J. H. K. G. S. M. Sajjadi, "A new Implementation of DA-based IIR Filters
[29] on FPGA," *ICEE12*, 2010.
- J. H. C. Ditzen, "A parameterizable biquad block for IIR filters in ASICs
[30] implementations and motivations," *ASIC Seminar and Exhibit*, pp. P7/4.1 - P7/4.4, Sep. 1990.