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 Quartuss 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 perlaksanaan perkakasan dengan menggunakan MATLAB dan Quartus II. Penapis itu akan direalisasikan dalam biquad seni bina untuk mencapai reka bentuk berskala yang dapat diperkembangkan secara lata 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-berjujukan 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-berjujukan 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

LIST OF TABLES

LIST OF FIGURES

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

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 $K_1 y_1(n) + K_2 y_2(n)$ when the input is $K_1 x_1(n) + K_2 x_2(n)$, 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

$$y^n = a_0 x^n + a_1 x^{n-1} + \ldots + a_p x^{n-p} \tag{1-1}$$

Where *p* is the filter order, $a_0 \ldots a_p$ 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 $a_0 \ldots a_p$ and $b_1 \ldots a_p$ 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

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

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.

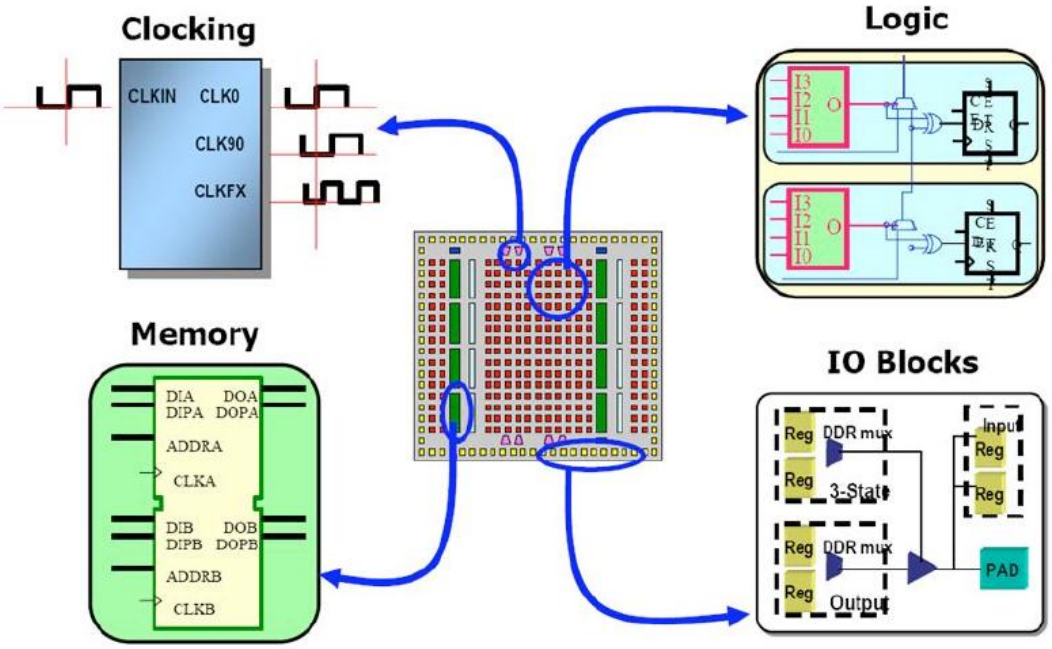

Figure 1.2 Internal structure of a generic FPGA (courtesy Xilinx, Inc.)

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

[1] B. A. Shenoi, Introduction to Digital Signal Processing and Filter Design, John Wiley & Sons, Inc., 2006.

[2] S. Winder, Analog and Digital Filter Design(2nd Edition), United States of America Newnes Publications, 2002.

[3] R. W. e. al., FPGA-based implementation of complex signal processing systems, John Wiley & Sons, Ltd, 2008.

[4] J. Serrano, "Introduction to FPGA design," *CAS - CERN Accelerator School Course on Digital Signal Processing*, pp. 231-247, 2007.

[5] 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.

[6] F. T. Arthur Williams, Electronic Filter Design Handbook, Fourth Edition, McGraw-Hill Handbooks, July 10, 2006.

[7] C. J. W. Alan V. Oppenheim, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proeedings of the IEEE*, vol. 60, no. 8, pp. 957 - 976, Aug. 1972.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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.

[13] A. R. H. S. a. J. M. Ravinder Kaur, "Design and Implementation of High Speed IIR and FIR Filter using Pipelining," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, April 2011.

[14] U. Meyer-Baese, Digital Signal Processing with Field Programmable Gate Arrays (3rd Edition), New York Springer Berlin Heidelberg, 2007.

[15] C. W. B. Baugh, "A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Transactions on Computers*, Vols. C-22, no. 12, pp. 1045-1047, 1973.

[16] S. Y. C. Pradabpet, "Design and implementation of biquad digital filter," *The 9th Asia-Pacific Conference on Communications 2003 (APCC 2003)*, vol. 3, Sep. 2003.

[17] S. R. a. S. M. F. N. Sedaghati-Mokhtari, "Hardware Implementation Analysis for Digital Filters," *Iranian Conference on Electrical Engineering (ICEE) 2006* , 2006.

[18] N. M. M. S. M. F. Hamed Hollisaz, "A Quantitative Approach to Digital Filter Implementation," *Microelectronics, 2005. ICM 2005. The 17th International Conference on* , pp. 160 - 164, 2005.

[19] B. Brandt, "A low-power area efficient digital filter for decimation and interpolation," *IEEE JSSC 29(6)*, 1994.

[20] A. G. Dempster, "Use of Minimum-Adder Multiplier Blocks in FIR Digital Filters," *IEEE Transactions on Circuits and Systems-II; Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 569-577, 1995.

[21] A. G. D. a. M. D. Macleod, "IIR Digital Filter Design Using Minimum Adder Multiplier Blocks," *IEEE Transactions on Circuits and Systems-II; Analog and Digital Signal Processing*, vol. 45, no. 6, pp. 761-763, 1998.

[22] J. A. A. G. Monica Arroyuelo, "FPGA-Based Digital Filters Using Bit-Serial Arithmetic," *CACIC*, 2007.

[23] "MATLAB - The Language Of Technical Computing," The MathWorks, Inc., [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 lIR filters in ASICs
[30] implementations and motivations," *ASIC Seminar and Exhibit*, pp. P7/4.1 - P7/4.4, Sep. 1990.