FINITE IMPULSE RESPONSE FILTER DESIGN ON DISTRIBUTED ARITHMETIC ARCHITECTURE

MUHAMAD IQBAL BIN ABU ZAHARIN

A project report submitted in partial fulfilment of the
requirement for the award of the degree of
Master of Engineering (Electrical – Electronics & Telecommunications)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2012

ACKNOWLEDGEMENT

In the name of Allah, the Most Beneficent and Merciful, with the deepest gratefulness to Allah who has given me the strength and ability to complete this project report.

First and foremost, I wish to express my sincere appreciation to my project supervisor, Professor Dr. Mohamed Khalil Hani, for encouragement, guidance and support. I would also like to thank Puan Nordinah Ismail for her helpful suggestions and ideas. Their knowledge and support throughout this project has made this work possible.

I would also like to thank my family member for their advice, prayer, encouragement and continuous moral support for the completion of this project report. Last but not least, thank you to my lecturers, friends and supporting staffs who has helped me to complete this thesis in various ways.

ABSTRACT

In signal and image processing application, highly repetitive operations and intense multiplication computation that exist in Digital Signal Processing systems makes it challenging to achieve less hardware requirement (area) and less latency (speed) performance using software based platforms, thus a lot of hardware design architecture becomes available to fill the gap. A close examination of the algorithms used in these, and related, applications reveals that many of these fundamental actions involve calculation of sum of products, vector dot product, inner product or multiply and accumulate (MAC). A study on the FIR filter involves multiply and accumulates operation where inner product forms the basis of the algorithms of the core. It is the aim of this paper to develop efficient architecture of a FIR filter on Distributed Arithmetic (DA) through ROM based ideally suited for efficient computation in order to achieve better performances in terms of speed and area by providing an effective methodology to implement MAC using a simple combination of memory elements, adders and shifters instead of lumped multipliers. The design is an 8-tap low pass filter based on 16-bit input samples and 16-bit signed coefficients at sampling frequency of 16MHz. All the coefficients are stored in the ROM 8x16 words size. The behavioural model of the FIR filter is structured in Verilog code and synthesized using Altera QuartusII version 11. The timing simulation is verified by running testbench codes written in Verilog using ModelSim 6.6d. The results from the simulation show that the FIR filter can be implemented using Distributed Architecture rather than using lumped multipliers.

ABSTRAK

Di dalam applikasi isyarat dan imej pemprosesan, operasi penggandaan yang terlalu mendalam dan berulang yang wujud di dalam sistem Pemprosesan Isyarat Digital adalah mencabar terutamanya dalam memenuhi syarat kurang perkakasan dan mempercepatkan kelajuan pemprosesan dalam melaksanakan pemprosesan melalui perisian menyebabkan wujudnya banyak rekacipta senibina perkakasan baru dalam memenuhi kekurangan tersebut. Penyelidikan yang terperinci terhadap algorithma yang digunakan banyak menjurus kepada asas hasil tambah penggandaan, penggandaan vektor dan hasil darab dan penambahan. Pengkajian terhadap '*finite impulse response filter*' (FIR) banyak melibatkan kepada asas hasil darab dan penambahan di mana penggandaaan vektor menjadi asas kepada algorithma teras. Kertas ini bertujuan untuk membangunkan rekabentuk filter menggunakan '*Distributed Arithmetic*' (DA) melalui pengenalan memori ROM untuk pengiraan yang lebih berkesan dalam memenuhi keputusan yang lebih baik dari segi kelajuan pemprosesan dan jumlah perkakasan yang digunakan dengan menggunakan kaedah hasil darab dan penambahan yang lebih berkesan dengan menggunakan kombinasi memori, penambah dan pengalih dari menggunakan longgokan pendarab. Rekacipta ini adalah '*8-tap low pass filter*' yang mempunyai 16-bit sampel input dan 16-bit pekali pada kadar frekuensi sampel 16MHz. Kesemua pekali disimpan di dalam memori ROM bersaiz 8x16. Model pemprosesan filter dibina dengan menggunakan kod Verilog dan disintesis menggunakan perisian Altera Quartus versi11. Simulasi pemasaan diperiksa menggunakan kod ditulis menggunakan kod Verilog dan perisian Modelsim 6.6d. Keputusan ujian menunjukkan '*FIR filter'* boleh direkabentuk menggunakan '*Distributed Arithmetic Architecture*' berbanding menggunakan longgokan pendarab.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	DECLARATION		**ii**
	ACKNOWLEDGEMENT		**iii**
	ABSTRACT		**iv**
	ABSTRAK		**v**
	TABLE OF CONTENTS		**vi**
	LIST OF TABLES		**ix**
	LIST OF FIGURES		**x**
	LIST OF ABBREVIATIONS		**xii**
1	**INTRODUCTION**		
	1.1	Motivation and Rational of the Work	1
	1.2	Objectives	2
	1.3	Scope of Work	3
	1.4	Organization of Project Report	3
2	**BACKGROUND OF DISTRIBUTED ARITHMETIC**		
	2.1	Distributed Arithmetic	5
	2.2	Mathematical Background	6
	2.3	Algorithm Test	8
	2.4	DA Memory Construction	8
	2.5	DA Memory Organization	10
3	**RESEARCH METHODOLOGY**		
	3.1	Distributed Arithmetic Transformation	14

	3.2	FIR Filter Design Procedure	17
		3.2.1 Filter Specification	17
		3.2.2 Coefficient Calculation	18
		3.2.3 Realization	18
		3.2.4 Implementation	18
	3.3	RTL Design Flow	19
	3.3.1	Algorithmic Modeling	20
		3.3.2 RTL Modeling	20
		3.3.3 Datapath (DU) and Control Unit (CU) Design	20
		3.3.4 Logic Synthesis	21
		3.3.5 Verification	21
4	**FIR FILTER DESIGN AND MATLAB ANALYSIS**		
	4.1	Introduction to DA based Finite Impulse Response Filter	22
	4.2	Why Implement FIR Filter using DA?	23
	4.3	Filter Design and Analysis using MATLAB	24
	4.4	Impulse Response	25
	4.5	Pole Zero Plot	26
	4.6	Realization and Implementation	27
5	**RTL DESIGN OF FIR FILTER**		
	5.1	RTL Design of FIR Filter	29
	5.2	Design Specification	29
	5.3	Algorithm Modelling	30
	5.4	RTL Modelling	31
	5.5	Datapath Unit Design	33
	5.6	Control Unit Design	34
	5.7	Design of Complete FIR Filter	35
	5.8	Simulation by Components	36

		5.8.1 ROM LUT	36
		5.8.2 REGR1 – Shift Right	36
		5.8.3 PSC Load and Shift Register	37
		5.8.4 PSC and ROM Integration	38
		5.8.5 ALU	39
		5.8.6 Accumulator Register C	39
	5.9	Simulation Results of DU, CU, Top Module	40
		5.9.1 Datapath Unit	40
		5.9.2 Control Unit	41
		5.9.3 Top Module of FIR	41
	5.10	Simulation Results Comparison between MATLAB and FPGA	42
6	**FUTURE WORK AND CONLUSION**		
	6.1	DA Limitation	43
	6.2	Conclusion	43
REFERENCES			45
APPENDIX			46 - 72

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	ROM LUT	9
2.2	ROM 0 for *I = 0*, row 0 of matrix A	12
2.3	ROM 1 for *I = 1*, row 1 of matrix A	12
2.4	ROM 2 for *I = 2* row 2 of matrix A	12
5.1	Table describes state and its operation	32
5.2	RTL Code derived from the ASM flowchart	33
5.3	RTL Control Signal Table for FIR filter	34
5.4	Operation table of the shift right register	37
5.5	PSC Load or Shift Operation Table	38
5.6	ALU operation	39

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Output from the Algorithm test	8
2.2	Top level functional block diagram of a DA	10
3.1	An 8-bit multiplier (AND gates) with adder to produce output, Y	15
3.2	DA is a bit serial technique	15
3.3	Substitute the Scaling Accumulator into the original design	16
3.4	Pre-computed sums stored in LUT for the bitwise addition	16
3.5	Design flow of a filter	17
3.6	RTL flow design	19
4.1	Direct form FIR implementation derived from Matlab's Simulink model	23
4.2	A lowpass filter frequency response	24
4.3	The impulse response of the FIR filter	26
4.4	The Pole / Zero Plot from Matlab analysis	27
4.5	MATLAB simulation output as Discrete Impulse input forced	28
5.1	ASM Flowchart of the FIR filter	32
5.2	Functional block diagram of DU of an 8-order FIR filter	33
5.3	Functional block diagram of the Control Unit	35
5.4	Top level block diagram of FIR filter	35
5.5	The contents of the ROM are the coefficients fetched based on the input data B as the address	36

5.6	Shift right register output	36
5.7	Serial-In Parallel Out	37
5.8	Parallel-in Serial Out	37
5.9	Integration between PSC and ROM. The input addresses successfully call the ROM memory content as expected	38
5.10	Timing simulation of ALU operation	39
5.11	Register C is a positive-edge triggered D-type	39
5.12	Timing simulation of Datapath Unit	40
5.13	Timing simulation of Datapath Unit when filter coefficients are applied	40
5.14	Timing simulation of Control Unit	41
5.15	Timing simulation of Top Module of FIR filter	41
5.16	MATLAB Simulation when input is 6	42
5.17	Timing simulation of Top Module when input 6 is applied	42

LIST OF ABBREVIATIONS

ASM	-	Algorithmic State Machine
CU	-	Control Unit
DA	-	Distributed Arithmetic Architecture
DU	-	Datapath Unit
FBD	-	Functional Block Diagram
FIR	-	Finite Impulse Response
FSM	-	Finite State Machines
RTL	-	Register Transfer Operation
MAC	-	Multiply and Accumulate

CHAPTER 1

INTRODUCTION

This project report describes about the design of efficient architecture of Finite Impulse Response (FIR) filter implemented on Distributed Arithmetic (DA) through ROM based for a fast computation of multiply and accumulate (MAC) in order to achieve better performances in terms of area and speed.

1.1 Motivation and Rational of the Work

Finite Impulse Response filter is widely used in signal and image processing applications. FIR filter has impulse response to any finite length of input. The filter settles to zero in finite time. The output of this linear time invariant system is determined by convolving its input signal with impulse response which is a weighted sum of the current and finite number of previous values of the input.

The standard form of FIR filter is realized by a shifting register a loop in which the filter coefficients are multiplied by the shifting register values. The sum of these multiplications will determine the output value. This operation is also called multiply and accumulates (MAC) operation which is the core of FIR filter

implementations. As the number of filter order increases the hardware requirement (eg: more multipliers, delays thus increasing area) become more important and crucial especially for developing system with hardware and cost constraints.

Multiplication is convolution. Multiplication of integers and discrete time convolution are same operations. The bit level description of multiplication can be mixed with the convolution. The most-often encountered form of computation in digital signal processing is inner products. Inner product computations can execute efficiently by DA by taking advantage of pre-computed data stored into ROM memory.

Our motivation for using DA in this FIR filter design is because of its computational efficiency. The advantages are best exploited in circuit design. By proper design one may reduce the total gate count in a signal processing arithmetic unit by a number seldom smaller than 50 and often as large as 80 percent (Stanley A. White,1989).

1.2 Objectives

In every FIR filter, multiply and accumulate is the core of FIR implementation. A FIR filter simply perform convolution as represented in the below equation:

$$y(n) = \sum_{k=0}^{N-1} A_k . b(n-k) \tag{1.1}$$

where: b(n-k) – input signal

Ak – is the coefficients/impulse response representation

N- filter order

By taking advantage of DA architecture, this project report will discuss on the FIR implementation using DA and how it can reduce the resource requirements for the inner product computations.

1.3 Scope of Work

Prior the design work, the DA theoretical study and background understanding is developed in order to get the idea how DA manipulates the bit level description. The scope of design work for the FIR design can be divided into two aspects. One is FIR filter design using MATLAB in order to obtained the coefficients and analyze the filter performance.

The second aspect is the RTL design of the FIR filter. The computed coefficients from the MATLAB are stored into the ROM memory of DA architecture. The logic synthesize of the Verilog HDL codes is run using Altera Quartus II version 11 with simulation run on Modelsim version 6.6 with testbenches generated in order to verify the design correctness.

The comparison between the MATLAB output and designed FIR Filter simulation outputs will be discuss in terms of its performance and hardware requirements.

1.4 Organization of Project Report

This project report comprise of six main chapters; Introduction, Background of Distributed Arithmetic, Research Methodology, FIR Filter and Matlab Analysis, RTL Design of FIR Filter and Future Development and Conclusion.

Chapter 1 discusses the Motivation and Rational of the Work, Scope of Work and Organization of this Project Report.

Chapter 2 discusses about the Background of Distributed Arithmetic. This chapter focuses on DA and its mathematical background where algorithm was also tested using C-program. The bit serial technique is discussed essentially contributes to the memory construction and organization.

Chapter 3 discusses the Research Methodology applied throughout the development of this project. Three aspects covered including the Distributed Arithmetic architecture study, FIR filter design flow and RTL design flow.

Chapter 4 discusses the FIR Filter Design and Matlab Analysis that covers the filter design work flow, filter theory and understanding and how the coefficients are computed.

Chapter 5 covers the RTL Design of the FIR Filter. Describes the step by step flow from Algorithm modelling, RTL modelling, Datapath Unit (DU) design and Control Unit (CU) design and the integration of both DU and CU. Simulation by components of each module and output comparison with MATLAB are also discussed.

Chapter 6 discusses Future Work and Conclusion. The current DA architecture still have outstanding issue specifically when the filter order become increasingly large the memory size will expand exponentially with 2^k. Several techniques and proposed modified architecture will also be discussed.

REFERENCES

1. A. Amira and F.Bensaali, "*An FPGA based Parameterisable System for Matrix Product Implementation",*in Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on 17 December 2003.
2. S. Chandrasekaran and A. Amira, "*Novel Sparse OBC based Distributed Arithmetic Architecture for Matrix Transforms*" in Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on 27-30 May 2007.
3. Mohamed Khalil Hani,"*Starter's Guide to Digital Systems VHDL & Verilog Design*"
4. Application Note "*The Role of FPGA – based Signal Processing*", URL:http://www.xilinx.com
5. Stanley A. White, "*Applications of Distributed Arithmetic to Digital Signal Processing*: A Tutorial Review," IEEE ASSP Magazine, July, 1989
6. H.C. Chen, J.I. Guo, C.W. Jen and T.S Chang, "*Distributed Arithmetic realization of cyclic convolution and its DFT application,"* IEEE Proc.-Circuits Devices Syst., Vol. 152, No. 6, December 2005
7. Bo Hong, Haibin Yin, Xiumin Wang, Ying Xiao, "I*mplementation of FIR Filter on FPGA Using DA-OBC Algorithm*," IEEE International Conference, December 2010.
8. Pramod Kumar Maher, "*New Approach to Look-Up-Table Design and Memory-Based Realization of FIR Digital Filter*," IEEE Transcations on Circuits and Systems, Vol57, No. 3, March 2010

9. Heejong Yoo and David V.Anderson, "*Hardware Efficient Distributed Arithmetic Architecture for High Order Digital Filters*," IEEE International Conference, March 2005.

10. Tsutomu Sasao, Yukihiro Iguchi, Takahiro Suzuki, "*On LUT Cascade Realization of FIR Filters*," Proceedings of the 2005 8th Euromicro conference on Digital System Design.