FPGA IMPLEMENTATION OF IMAGE PROCESSING 2D CONVOLUTION FOR SPATIAL FILTER

NG BEE YEE

UNIVERSITI TEKNOLOGI MALAYSIA

FPGA IMPLEMENTATION OF IMAGE PROCESSING 2D CONVOLUTION FOR SPATIAL FILTER

NG BEE YEE

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 family, lecturers, fellow friends and those who have guided and inspired me throughout my journey of education

ACKNOWLEDGEMENT

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 encouragement, guidance and sharing of knowledge throughout the process of completing this project.

I would like to extend the appreciation to Intel Microelectronics (M) Sdn. Bhd. for funding my studies. I would like to thank my manager as well as my colleagues who had provided me with help and support throughout the duration of my studies.

Last but not least, I would like to thank my family for giving me the support and encouragement as well as for being understanding throughout my studies.

ABSTRACT

Computer manipulation of images is generally defined as digital image processing (DIP). DIP is used in variety of applications, including video surveillance, target recognition, and image enhancement. Some of the many algorithms used in image processing include convolution, edge detection and contrast enhancement. These are usually implemented in software but may use special purpose hardware for speed. With advances in the VLSI technology hardware implementation has become an attractive alternative. Assigning complex computation tasks to hardware and exploiting the parallelism and pipelining in algorithms yield significant speedup in running times. The main objective of this project is to develop an image processing algorithm, 2D convolution. The algorithm is designed and implemented in synthesizable Verilog HDL. Upon completion of the coding, its functionality and timing are then verified thoroughly. Subsequently, the performance of the 2D convolution is analyzed. The designed 2D convolution applies pipeline and parallel architecture for speedup and real-time applications. The entire design process starts with architecture definition and design. Once the required modules and functionalities such as DU and CU are defined, they are then coded and integrated. Verification is done from bottoms up starting from individual sub-modules. In addition, the design is further verified with real image pixels and compared the output pixels with that obtained from software (MATLAB). Altera Quartus II compilation report shows the 2D convolution design acheieves *fmax* as high as 394MHz using off-chip RAM. The performance is slighly degraded, to about 322MHz with on-chip RAM.

ABSTRAK

Komputer manipulasi imej secara amnya ditakrifkan sebagai pemprosesan imej digital (DIP). DIP digunakan dalam pelbagai aplikasi, termasuk pengawasan video, pengesahan sasaran, dan peningkatan kualiti imej. Banyak algoritma digunakan dalam pemprosesan imej termasuk convolution, pengesahan pinggir imej dan peningkatan kontras imej. Biasanya algoritma ini dilaksanakan dalam perisian (sofware), malah ia juga dilaksanakan dalam perkakasan (hardware) dengan tujuan untuk mencapai kelajuan dalam perlaksanaan algoritma. Dengan kemajuan dalam teknologi VLSI, perlaksanaan perkakasan telah menjadi satu alternatif yang amat menarik. Melaksanakan tugas-tugas pengiraan yang kompleks dalam perkakasan, dan mengeksploitasi parallelism dan pipeline dalam algoritma memberikan kecepatan ketara. Objektif utama projek ini adalah untuk membina satu algoritma pemprosesan imej, iaitu convolution dua-dimensi (2D). Algoritma tersebut telah direkabentuk dan dilaksanakan dalam bahasa Verilog HDL. Setelah proses rekaan selesai, fungsinya akan diuji dan prestasiny turut dianalisis. 2D convolution ini menggunakan pipeline dan seni-bina selari untuk mencapai kecepatan dan real-time aplikasi. Proses pembangunan bermula dengan definisi seni-bina dan reka-bentuk. Selepas itu, fungsi dan modul-modul utama seperti DU dan CU akan ditakrifkan dan direka-bentuk. Kemudian, merekan telah digabungkan untuk membina 2D convolution. Ujian terhadap rekaan akan dilakukan dari bawah, iaitu bermula dengan modul asas. Di samping itu, system tersebut juga diuji and disahkan fungsinya dengan menggunakan piksel imej, seterusnya membandingkan piksel output yang diperolehi dengan output daripada perisian (MATLAB). Laporan kompilasi daripada Altera Quartus II menunjukkan bahawa reka-bentuk 2D convolution ini mencapai *fmax* sebanyak 394MHz menggunakan off-chip RAM. Walaubagaimanapun, prestasinya akan turun kepada 322MHz dengan on-chip RAM.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xiv
	LIST OF APPENDICES	xv
1	**INTRODUCTION**	1
	1.1 Background	1
	1.2 Motivation and Problem Statement	3
	1.3 Objectives	4
	1.4 Scope of Work	4
	1.5 Report Outline	5
2	**LITERATURE REVIEW AND THEORY**	7
	2.1 Literature Review	7
	2.2 Convolution Operation	14
	2.2.1 1D Convolution	15
	2.2.2 2D Convolution	16

3	**METHODOLOGY AND DESIGN TOOL**		18
	3.1	Design Approach and Implementation Flow	18
	3.2	MATLAB	21
	3.3	Verilog HDL	22
	3.4	FPGA Synthesis	24
	3.5	Altera Quartus II 9.1	25
	3.6	ModelSim-Altera 6.6d Starter Edition	26
4	**1D CONVOLUTION MODELING AND DESIGN**		28
	4.1	1D Convolution Design	28
	4.2	Design Specification	30
	4.3	Algorithmic Modeling	30
	4.4	RTL Modeling	31
	4.5	RTL Design	34
		4.5.1 DU Design	34
		4.5.2 CU Design	36
		4.5.3 Top-Level Design and HDL Coding	38
	4.6	Logic Synthesis and Simulation	40
	4.7	Design Optimization and Performance Analysis	40
		4.7.1 Resource Utilization	41
		4.7.2 Maximum Operating Frequency	43
		4.7.3 Conclusion	44
5	**2D CONVOLUTION MODELING AND DESIGN**		46
	5.1	2D Convolution Design	46
	5.2	Design Specification and Constraint	48
	5.3	Algorithmic Modeling	51
	5.4	RTL Modeling	52
	5.5	RTL Design	54
		5.5.1 DU Design	55
		5.5.2 CU Design	58
		5.5.3 Top-Level Design and HDL Coding	60
	5.6	Logic Synthesis and Simulation	62

6	**DESIGN VERIFICATION AND PERFORMANCE ANALYSIS**		63
	6.1	1D Convolution Quartus II Simulation	63
	6.2	2D Convolution Quartus II Simulation	65
		6.2.1 DU Module Timing Simulation	65
		6.2.2 CU Module Timing Simulation	66
		6.2.3 Top-Level Timing Simulation	68
	6.3	2D Convolution System Verification Methodology	69
		6.3.1 Algorithm Development in MATLAB	70
		6.3.2 Pixel Extraction from MATLAB	70
		6.3.3 ModelSim-Altera Simulation	71
		6.3.4 Output Pixel Extraction	74
		6.3.5 Output Image Visualization	74
	6.4	Performance Analysis	76
7	**CONCLUSION AND FUTURE WORK**		79
	7.1	Conclusion	79
	7.2	Future Work	80
REFERENCES			82
Appendices A – E			84 - 110

LIST OF TABLES

TABLE NO.	TITLE	PAGE
4.1	RTL-CS Table for Control Unit with Multiplier in DU	36
4.2	RTL-CS Table for Control Unit with Barrel Shifter in DU	37
4.3	CU and DU Modules *fmax*	43
4.4	1D Convolution Design System *fmax*	44
5.1	RTL-CS Table for 2D Convolution Control Unit	59
6.1	Input Pixels with the Expected Output Pixels	64
6.2	2D Convolution Top-Level and Sub-Modules *fmax*	77
6.3	Clock Cycle Count per FSM State	78
6.4	2D Convolution System Performance Measure Summary	78

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Basic Building Block [3]	8
2.2	Memory Structure for Real-time Convolution [4]	9
2.3	Multi-FPGA Architectural Scheme [4]	10
2.4	Block Diagram of an RxS Complete Convolver [1]	11
2.5	Block Diagram of a 3x3 Elementary Convolver [1]	12
2.6	5x5 Convolution using a 2x2 Grid of 3x3 Elementary Convolvers [1]	13
2.7	Architecture of 1D Convolution Module and Assembly of R 1D Convolution Modules to form a RxS Convolver with R=5 and S=5 [1]	14
2.8	3x3 Pixel Window and Origin	15
2.9	2D Convolution	17
3.1	2D Convolution Design Flow	20
3.2	Multiple Abstractions for Digital System Design	23
3.3	General Architecture of an FPGA	24
3.4	Quartus II Design Flow	26
4.1	1D Convolution Mask	30
4.2	I/O Block Diagram of 1D Convolution	32
4.3	ASM-chart of 1D Convolution	32
4.4	DFG of 1D Convolution using Multiplier Operator	33
4.5	DFG of 1D Convolution using Shift Operator	33
4.6	Functional Block Diagram of Datapath Unit using Multiplier	34
4.7	Functional Block Diagram of Datapath Unit using Barrel Shifter	35

4.8	Functional Block Diagram of Control Unit with Multiplier in DU	37
4.9	Functional Block Diagram of Control Unit with Barrel Shifter in DU	38
4.10	Top-Level Block Diagram of 1D Convolution with Multiplier in DU	39
4.11	Top-Level Block Diagram of 1D Convolution with Barrel Shifter in DU	39
4.12	DU Verilog Code Modification for 1D Convolution with Barrel shifter DU and Positive-Edge Triggered FF CU	41
4.13	DU Timing Waveform for Designs in Option (i) and (ii)	41
4.14	Quartus II Compilation Report Summary of 1D Convolution with Multiplier in DU	42
4.15	Quartus II Compilation Report Summary of 1D Convolution with Barrel Shifter in DU	42
5.1	2D Convolution System Level Architecture	46
5.2	Block Diagram of an RxS Complete Convolver [1]	47
5.3	3x3 Convolution Mask for Gaussian Filter in Spatial Domain	49
5.4	2D Convolution Output Pixel Response	50
5.5	I/O Block Diagram of 2D Convolution	52
5.6	ASM-chart of 2D Convolution	53
5.7	DFG of 1D Convolution with Kernel Coefficient [1 2 1]	54
5.8	DFG of 1D Convolution with Kernel Coefficient [2 4 2]	54
5.9	Functional Block Diagram of Sub-Module *shift_121*	55
5.10	Functional Block Diagram of Sub-Module *shift_242*	56
5.11	Functional Block Diagram of 2D Convolution Datapath Unit	57
5.12	Functional Block Diagram of 2D Convolution Control Unit	60
5.13	Top-Level Block Diagram of 2D Convolution	61
6.1	1D Convolution Top-Level Timing Simulation Waveform (Barrel Shifter in DU and Positive-Edge Triggered State Registers in CU)	64
6.2	5x10 Random Input Pixels	65
6.3	Expected Output Pixels	65

6.4	DU Module Timing Simulation Waveform	66
6.5	CU Module Timing Simulation Waveform	67
6.6	2D Convolution Top-Level Timing Simulation Waveform	68
6.7	MATLAB Code for 2D Convolution Algorithm	70
6.8	Input Image Pixels from MATLAB Variable Editor Terminal	71
6.9	2D Convolution Simulated Output Pixel File	73
6.10	2D Convolution Simulation Report File	73
6.11	MATLAB Command to Display FPGA Simulated Output Image	74
6.12	Original Image with Gaussian Noise	75
6.13	Gaussian Filtered Output Image: (a) MATLAB-version (b) FPGA-version	75
6.14	Quartus II Compilation Report Summary of 2D Convolution Design	76

LIST OF ABBREVIATIONS

ASIC	-	Application Specific Integrated Circuit
ASM	-	Algorithmic State Machine
CAD	-	Computer Aided Design
CB	-	Computation Block
CLB	-	Configurable Logic Block
CPU	-	Central Processing Unit
CU	-	Control Unit
DFG	-	Data Flow Graph
DIP	-	Digital Image Processing
DSP	-	Digital Signal Processor
DU	-	Datapath Unit
FIFO	-	First In First Out
FIR	-	Finite Impulse Response
FPGA	-	Field Programmable Gate Array
FSM	-	Finite State Machine
HDL	-	Hardware Description Language
IC	-	Integrated Circuit
LAB	-	Logic Array Block
RAM	-	Random Access Memory
RTL	-	Register Transfer Level
RTL-CS	-	RTL Control Sequence
Verilog	-	Verilog HDL
VHDL	-	Very High Speed Integrated Circuits HDL
VLSI	-	Very Large Scale Integration

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	1D Convolution Verilog Program	84
B	Timing Simulation of 1D Convolution Design	91
C	2D Convolution Verilog Program	95
D	PERL Script for Data Post-Processing	104
E	2D Convolution System ModelSim Testbench	109

CHAPTER 1

INTRODUCTION

This project is about the hardware architecture design of a 2D convolution for spatial domain filter. The 2D convolution algorithm is implemented on the Field Programmable Gate Array (FPGA). This chapter gives an overview of the whole project, starts with a brief introduction to the background, followed by the problem statement, project objectives, scope of work and report outline.

1.1 Background

Computer manipulation of images is generally defined as digital image processing (DIP). DIP is gaining widespread popularity nowadays; it is a dynamic area with applications widely used in our everyday life such as in medicine, space exploration, surveillance, authentication, automated industry inspection and etc. Applications such as these involve different processes and algorithms like image enhancement, noise reduction, feature recognition and edge/object detection.

DIP algorithms require high computation capabilities, especially when elaborate high resolution images for real-time systems. It is easier to implement such applications on a generable purpose computer. However, the implementation may not be very efficient in terms of speed due to the additional constraints put on memory and other peripheral device management. Usually, application specific hardware offers much greater speed than a software implementation.

Generally there are two types of technologies available for hardware design. First is Application Specific Integrated Circuit (ASIC), it is a full custom hardware design technique. Second is the semi-custom hardware device, which is also known as programmable device that includes Field Programmable Gate Array (FPGA) and Digital Signal Processor (DSP).

High performance can be achieved via full custom ASIC design. On the other hand, the complexity and the cost associated with the full custom hardware design are very high. The ASIC design is lack of flexibility in terms of design change and modification; time taken to design the hardware is also very high. In addition, if an error exist in the ASIC design, the product becomes useless once the design is fabricated. Therefore, ASIC design is normally used in high volume commercial application after the design is confirmed with its functionality and performance via FPGA prototype.

FPGA is programmable device. It is also known as reconfigurable device. Reconfigurable device is a processor used to program a design; and the design has the flexibility to change and modify anytime by reprogramming the device. Besides that, parallelism and pipelining hardware design techniques can be developed on an FPGA, which is not possible in dedicated DSP designs. Therefore FPGAs are ideal choice for implementation of real-time image processing algorithms.

Traditionally, hardware engineers use a Hardware Design Language (HDL) to configure the FPGA devices. The two primary languages include Verilog and VHDL. Verilog and VHDL are specialized design techniques that are not immediately accessible to software engineers, who have often been trained using imperative programming languages. Consequently, there are several efforts to translate and convert the algorithmic oriented programming languages directly into hardware descriptions over the past few years.

Classically, image processing algorithms are implemented on software. Advancement in the VLSI technology has made the hardware implementation to

become an attractive alternative. Complex computation tasks are assigned to hardware, and these allow the parallelism and pipelining design techniques to be developed in algorithms. All these yield significant speedup in running times. Generally, image processing algorithms are implemented on reconfigurable hardware to allow faster time-to-market, minimize the time-to-market cost, enable design flexibility and rapid prototyping of complex algorithms, and simplify debugging and verification works.

1.2 Motivation and Problem Statement

It is hard to enumerate aspects of electrical engineering where filtering is employed. Examples of filtering operations include noise suppression, enhancement of selected frequency range, bandwidth limiting, etc. Analog filters suffer from sensitivity to noise, nonlinearities, dynamic range limitations, inaccuracies due to variations in component values, lack of flexibility and imperfect repeatability. Consequently, digital filters (and FIR filters - convolers) are getting more and more attractive. The major drawback of digital filters is high computational requirements, especially for high frequency signals. Real-time image processing is an example of such a system. This project concentrates on image convolution (two-dimensional FIR filtering); however similar conclusions can be drawn for one-dimensional filters, matrix multiplication, or partially on artificial neural networks, etc.

The 2D convolution performance requirement in a DIP system is very crucial because it is the key building block for real-time system applications. DIP algorithms require high computation capability especially when elaborate high resolution images. Some applications even require more than 300 million multiplications and additions per second [1]. Therefore, 2D convolution system needs to operate in high frequency. However in most designs, multiplier operating frequency becomes the bottleneck. In addition, multiplies is also very costly in term of resource utilization. Besides that, in order to support real-time system, 2D convolution also requires an

effective memory access algorithm so that all neighborhood pixels can be accessed within one clock cycle.

1.3 Objectives

The objective of this project is to design hardware architecture of an image processing algorithm, 2D convolution for Gaussian filter in spatial domain. The 2D convolution algorithm is implemented on FPGA using Verilog HDL. This project designs an architecture that can perform fast computation and multiplication. The design assigns complex computation tasks to the hardware and exploiting the parallelism and pipelining in algorithm yield significant performance and speedup in execution times. Besides that, an efficient algorithm is implemented to access the entire neighborhood pixels defined by the convolution kernel within one clock cycle for real-time applications. In the project, several design options and architectures are explored for performance trade-off analysis.

1.4 Scope of Work

In this project, a 2D convolution algorithm is implemented on the Altera FPGA, using Altera Quartus II as a compiler and simulator. The 2D convolution hardware architecture has high performance and speedup, so that it is able to generate the outputs real-time. The design will be accomplished using Verilog HDL. A pipelined architecture is proposed to produce the output on every clock cycle.

MATLAB is used to produce a software version of the algorithm. The design functionality and timing are verified thoroughly by performing the simulation with the real input image pixels, and comparing the FPGA outputs with the software version output from MATLAB. Lastly, the performance of the 2D convolution design is characterized via the performance measures.

1.5 Report Outline

The report is organized into 7 chapters namely the introduction, literature review and theory, methodology and design tool, 1D convolution modeling and design, 2D convolution modeling and design, design verification and performance analysis, and lastly the conclusion and future work.

Chapter 1 gives an overview of the project. It starts with a brief introduction to the background, followed by the problem statement, project objectives, scope of work and report outline.

Chapter 2 gives an overview on the prior work and literature review. Next, it introduces the theory of window operator and 2D convolution algorithm implemented in digital image processing.

Chapter 3 describes design methodology, implementation plan and design tools that used to develop and realize the 2D convolution hardware architecture on FPGA. This chapter presents the complete convoler design approach and implementation flow.

Chapter 4 discusses the 1D convolution modeling and design implementation. Different architectures have been explored for performance and architecture trade-off. Next, this chapter presents the 1D convolution design optimization and performance analysis, which is then used as the benchmark and guideline to determine the computation block (CB) architecture for 2D convolution design.

Chapter 5 is the core chapter for this project report which discusses the 2D convolution architecture and design implementation. This chapter starts with the 2D convolution design specifications and constraints. Next, this chapter presents the 2D convolution design, and the detailed implementation of every sub-module.

Chapter 6 presents the timing simulation diagrams of 1D and 2D convolution designs. In addition, this chapter also discusses the 2D convolution system design verification methodology and the results. The correctness of 2D convolution design and its sub-modules are verified and validated via Altera Quartus II timing simulations. Next, the entire 2D convolution system is further verified with real image pixels using MATLAB and ModelSim-Altera tools. Simulation waveforms, testbench and design performance analysis are presented in this chapter as well.

Lastly, Chapter 7 concludes the 2D convolution system design and proposes the future works for further improvement and enhancement of the project.

REFERENCES

[1] Bernard Bosi, Guy Bois, Yvon Savaria, "Reconfigurable Pipelined 2-D Convolvers for Fast Digital Signal Processing', *IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Vol. 7, No. 3,* September 1999.

[2] Ben Cope, *"Implementation of 2D Convolution on FPGA, GPI and CPU,"* Department of Electrical & Electronic Engineering, Imperial College London.

[3] Khader Mohammad, Sos Agaian, "*Efficient FPGA Implementation of Convolution,"* proceedings of the *2009 IEEE International Conference on Systems, Man, and Cybernetics,* October 2009.

[4] Arrigo Benedetti, Andrea Prati, Nello Scarabottolo, *"Image Convolution on FPGAs: the Implementation of a Multi-FPGA FIFO Structure,"* proceedings of the *24th Euromicro Conference (EUROMICRO 98),* pp. 123-130.

[5] Syed M. Qasim, Ahmed A. Telba, Abdulhameed Y. AlMazroo, *"FPGA Design and Implementation of Matrix Multiplier Architectures for Image and Signal Processing Applications," IJCSNS International Journal of Computer Science and Network Security, Vol. 10, No. 2,* February 2010.

[6] S. Perri, M. Lanuzza, P. Corsonello, G. Cocorullo, *"SIMD 2-D Convolver for Fast FPGA-based Image and Video Processors",* Perri at al., Paper D2.

[7] Ernest Jamro, Kazimierz Wiatr, *"Convolution Operation Implemented in FPGA Structures for Real-Time Image Processing,"* AGH Technical University, Institute of Electronics.

[8] Daggu Venkateshwar Rao, Shruti Patil, Naveen Anne Babu, V. Muthukumar, *"Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based hardware Descriptive Languages," International Journal of Theoretical and Applied Computer Sciences, Vol. 1, No. 1,* 2006, pp. 9-34.

[9] Fred Weinhaus, *"Digital Image Filtering,"* White Paper.

[10] Anthony Edward Nelson, *"Implementation of Image Processing Algorithms on FPGA Hardware,"* thesis submitted to the faculty of the Graduate School of Vanderbilt University, May 2000.

[11] S. Belkacemi, K. Benkrid, D. Crookes, A. Benkrid, *"Design and Implementation of a High Performance Matrix Multiplier Core for Xilinx Virtex FPGAs," IEEE International Workshop on Computer Architectures for Machine Perception (CAMP),* 2003.

[12] Mahendra Vucha, Arvind Rajawat, *"Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication," International Journal of Computer Applications (0975-8887), Vol. 26, No. 3,* July 2011.