HARDWARE AND SOFTWARE CO-SIMULATION PLATFORM FOR CONVOLUTION OR CORRELATION BASED IMAGE PROCESSING ALGORITHMS

SAYED OMID AYAT

UNIVERSITI TEKNOLOGI MALAYSIA

HARDWARE AND SOFTWARE CO-SIMULATION PLATFORM FOR CONVOLUTION OR CORRELATION BASED IMAGE PROCESSING ALGORITHMS

SAYED OMID AYAT

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

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2014

My genuine dedications to,
My beloved wife, father and mother;
My enduring and experienced supervisor;
Who are always there for me,
Every step of the way,
Thanks!

ACKNOWLEDGEMENT

First and foremost, I would like to convey my deepest gratitude to my supervisor, Prof. Dr. Mohamed Khalil, who has been the most influential person who always guide, motivate, support, and advise me throughout my thesis completion with his inspiring and constructive comments and valuable feedbacks.

Furthermore, I would like to extend my deepest and sincere appreciation to my beloved family, especially to my wife, who always support me, encourage me, and listen to me along my research. They are my source of inspiration and my greatest refuge.

Finally, I would like to pay special tribute to my seniors, Miss Lee, Mr. Sia, Mr. Liew and my cherished friends, who are willing to guide me with patience and keenness. Thank you so much for deserving a lot of time and passion to guide, teach, motivate and share their experiences and knowledge with me.

ABSTRACT

Software implementation of image processing algorithms in which convolution or correlation is applied is too slow to be real-time. As long as the system design gets larger, it should be partitioned into two parts: software and hardware. In order to achieve real time performance, it is essential to map the fast convolution or correlation module, which is the heaviest computation intensive part, in hardware instead of software. Our test case is “generic image pre-processing algorithm” which includes resizing, noise filtering and normalization. In noise filtering part of the preprocessing algorithm in which convolution is used should be implemented in hardware while the rest of the preprocessing algorithm stays in software. Next, to verify our hardware/design software we can deploy it on FPGA board, but it is very time consuming and involves a lot of technical complexities. In that case, this design used hardware/software co-simulation and direct programming interface (DPI-C) whereas it allows System Verilog calls C functions and vice versa. The proposed work has overcome the problems faced when running a co-simulation based on Modelsim simulated using direct programming interface (DPI) technique.

ABSTRAK

Pelaksanaan perisian algoritma pemprosesan imej di mana kekusutan atau korelasi digunakan adalah terlalu lambat untuk dilaksanakan dalam masa nyata. Apabila reka bentuk sistem bertambah besar , proses berkenaan harus dibahagikan kepada dua bahagian: perisian dan perkakasan. Untuk mencapai prestasi masa nyata, ia adalah penting untuk memetakan modul kekusutan atau modul korelasi, yang merupakan bahagian pengiraan yang intensif, dalam perkakasan dan bukannya perisian. Kes ujian kami adalah algoritma pra-pemprosesan imej yang generik termasuk penyelerasan saiz, penapisan hingar dan normalisasi. Bahagian kekusutan dalam modul penapisan hingar telah direkakan sebagai perkakasan manakala modul-modul lain dalam algoritma pra-pemprosesan imej kekal sebagai perisian. Seterusnya, untuk mengesahkan rekabentuk perkakasan / perisian berkenaan, papan *Field-Programmable Gate Arrays* boleh digunakan tetapi proses tersebut mengambil masa yang panjang dan melibatkan banyak kerumitan teknikal. Dalam kes itu, reka bentuk ini menggunakan simulasi perkakasan / perisian dengan teknik antara muka pengaturcaraan langsung (DPI-C) yang membolehkan panggilan fungsi C dari Sistem Verilog dan sebaliknya. Kerja yang dicadangkan telah mengatasi masalah yang dihadapi apabila melaksanakan simulasi bersama berdasarkan simulator Modelsim yang menggunakan antara muka pengaturcaraan langsung teknik (DPI).

TABLE Of CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABREVIATIONS

AGU	-	Address generation unit
ALU	-	Arithmetic logic unit
ASM	-	Algorithmic sate machine
C	-	C programming language
CU	-	Control Unit
DFG	-	Data flow graph
DPI	-	Direct programming interface
DU	-	Dtatpath unit
FPGA	-	Field programmable gate array
HDL	-	Hardware description language
HDVL	-	Hardware description and verification language
IDE	-	Integrated development environment
ISS	-	Instruction set simulator
MATLAB	-	Matrix laboratory
PLI	-	Programming language interface
RAM	-	Random access memory
RGB	-	Red Green Blue
RTL	-	Register transfer level
DFG	-	data flow graph
SoC	-	System-on-chip

SV	-	SystemVerilog
VLI	-	Verilog Procedural Interface
2D	-	2-dimensional

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

This thesis proposes a co-simulation platform using the state-of-art System Verilog HDVL via direct programming interface (DPI) technique applied on image processing sub algorithms. This chapter gives an overview of the proposed title and, problem statement, objectives, scope of work, expected contribution and thesis outline.

In the recent years, as the technology grows system-on-chip (SOC) designs become bigger and bigger. Having all the design in hardware makes our design bulky and expensive. On the other hand implementing into the pure software will consequence very slow design since we are not able to use parallel processing as we are in the hardware design. The hardware design is done only when performance is taken into account. This is because hardware design will speed up the whole system or an application.

There is a report shows that functional verification has become the biggest bottleneck as it consumes roughly 70% of the chip development time and efforts [1]. To verify our hardware/software design we can deploy it on FPGA board, but it is

very time consuming and involves a lot of technical complexities. In that case, this design used hardware/software co-simulation and direct programming interface (DPI-C) whereas it allows System Verilog calls C functions and vice versa. The proposed work has overcome the problems faced when running a co-simulation based on Modelsim simulator using direct programming interface (DPI) technique.

Image processing becomes very useful and important these days as the other technologies grows. It is widely used in different area such as military, astronomy, security systems, robotics and etc. The image itself can be declared and defined by its pixel values and these values can be assigned to matrix elements. In the case of color images which are consist of a combination of 3 main colors, i.e. red, green and blue. Each color can separately assign to a particular matrix. In the case of the N by M pixel gray level image it can only defined by one matrix that is N by M. Because of this definition of image, we are dealing with 2 dimensions matrixes.

There are a lot of operations that can operate on 2D matrix or in other word our input image. For instance simple operations like addition, multiplication, subtraction and so on. But when we want to process the real image using some operation such as edge detecting, blurring, sharpening and even logical operation such as dilation and erosion we have to use more advanced operation called “convolution” and “correlation”. In this work convolution and correlation module is implemented in hardware.

1.2 Problem Statements

In this segment at first the problems are stated and then, according to each statement related solution is proposed.

Software implementation of convolution or correlation algorithm is too slow to be real-time. It is because in software we are only restricted to execute one operation

at a same time. Since both convolution and correlation share most part of their algorithm, it can be merged together in one module. Another difficulty is that in the real world applications input image size is not fixed. On the other hand filtering is different from one application to another and kernel size also depends on that specific application.

The whole image processing algorithms in which convolution or correlation is applied should not be implemented in hardware because it will be costly. Besides, when one algorithm is working efficiently on software it is not appropriate to implement it in hardware. The hardware implementation is advisable only when parallel processing is possible so that we will be able to speed up the design. We should find a way for co-designing the hardware/software algorithm. As long as we are using the free version of ModelSim we may face some limitations. The biggest difficulty faced is that software part of algorithm doesn't consume time simulation.

Verification via deploying into FPGA board is very time consuming and involves a lot of technical complexities.

1.3 Objectives

Design and implement a 2D Convolution or 2D Correlation hardware module targeting for FPGA. And parameterizing the whole code to handle any kernel and Image size.

To partitioned into two parts in order to achieve real time performance: software and hardware. Software and hardware parts should be able to communicate with each other through DPI-C. We have to establish a platform to overcome Modelsim limitations in such a way that can be applied to more HW/SW co-simulation cases.

It is good to know whether the hardware/design software works fine or not. But as it is mentioned Verification via deploying into FPGA board is very time consuming and involves a lot of technical complexities. Therefore, to verify our design, we use HW/SW co-simulation using DPI-C.

1.4 Scope of the Research

The scopes of the work for this thesis are:

I. Our test case is “generic image pre-processing algorithm”, which is the part of pattern recognition algorithm and includes 3 parts: resizing, filtering and normalization.

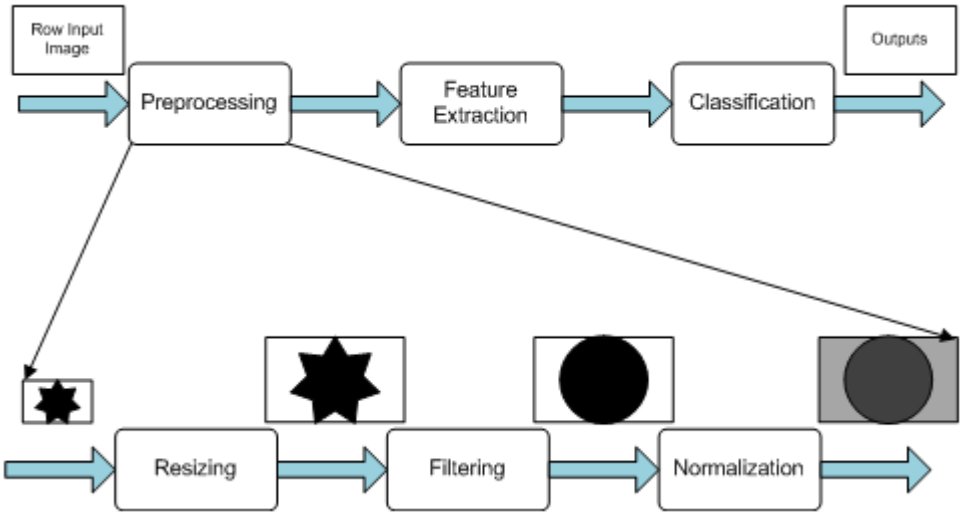

Figure 2-1 Typical Pattern Recognition Algorithm

II. HW/SW design is written in System Verilog and C, verification will be possible by employing MODELSIM free simulator and DPI-C.

III. After getting results, it will be compared with pure software

implementation in MATLAB or Dev-C++.

IV. In this paper Open source software tools such as Quartus II 13.0 and Altera-ModelSim 10.1d are used to ensure this work can be repeated and extended in the future.

1.5 Expected Contribution

The expected contribution of this work is to establish a platform to be able to simulate hardware-software designs before deploying into the FPGA board. The free version of Modelsim is being used and the proposed methodology should overcome its limitations to perform co-simulation.

REFERENCES

[1] "Modi D., Sitapara H., Shah R., Mehul, E., (2011). Integrating MATLAB with Verification HDLs for Functional Verification of Image and VideoProcessing ASIC, 2(2), 258–265.."

[2] E. SALIBA and A. DIPANDA, "An overview of Pattern Recognition," *wikiprogress. org*, 2013.

[3] B. Baba, "A HIGH SPEED 2D CONVOLUTION HARDWARE MODULE FOR IMAGE PROCESSING APPLICATIONS IN HARDWARE," UNIVERSITY TECHNOLOGY MALAYSIA, 2013.

[4] A. H. Heng, "Reconfigurable address generation unit for 2D correlation in FPGA," Master, Universiti Teknologi Malaysia, 2012.

[5] K. K. HORNG, "FPGA IMPLEMENTATION OF A RECONFIGURABLE ADDRESS GENERATION UNIT FOR IMAGE PROCESSING APPLICATIONS," MASTER, UNIVERSITY TECHNOLOGY MALAYSIA, 2013.

[6] A. Freitas, *Hardware-Software Co-verification Using the SystemVerilog DPI*: Techn. Univ. Chemnitz, Fakultät für Informatik, 2007.

[7] *http://www.wikipedia.org/*.

[8] *http://www.testbench.in*.

[9] *http://www.doulos.com/knowhow/sysverilog/tutorial/dpi/*.

[10] N. A. a. F. T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting," 2001.

[11] R. E. W. Rafael C. Gonzalez, *Digital Image Processing*, Second Edition ed., 2002.

[12] "Khalil-Hani, M. Digital Systems: VHDL & Verilog Design. UTMSkudai: Prentice Hall. 2012.."

[13] N. P. Escudero, "PERFORMANCE ESTIMATION FOR THE DESIGN SPACE EXPLORATION OF SYSTEM-ON-CHIP SOLUTIONS," PHD, 2003.

[14] J. K. J. KAN, "SOFTWARE AND HARDWARE CO-SIMULATION PLATFORM FOR IMAGE PROCESSING," Master, UTM, 2014.