

OPENCL BASED FPGA IMPLEMENTATION OF OBJECT
DETECTION USING CONVOLUTIONAL NEURAL NETWORK

LEE ZHAO LUN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**OPENCL BASED FPGA IMPLEMENTATION OF OBJECT
DETECTION USING CONVOLUTIONAL NEURAL
NETWORK**

LEE ZHAO LUN

**This report is submitted in partial fulfilment of the requirements
for the degree of Bachelor of Electronic Engineering with Honours**

**Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka**

JUNE 2018

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : OpenCL based FPGA implementation of Object
Detection using Convolutional Neural Network
Sesi Pengajian : 2017/2018

Saya LEE ZHAO LUN mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

SULIT*

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD*

(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan).

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: 6,Lebuh Rasi Jaya,
Tmn Rasi Jaya
31450 Menglembu
Ipoh Perak

Tarikh : 21 Mei 2018

Tarikh : 21 Mei 2018

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “OpenCL based FPGA implementation of Object Detection using Convolutional Neural Network” is the result of my own work except for quotes as cited in the references.

Signature :

Author :

Date :

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name :

Date :

DEDICATION

Dedicated to my beloved family member for the support and encouragement. A special thanks to Yap June Wai and Supervisor Professor Dr. Zulkalnain who gave me support, insight, guidance and advice throughout the whole journey of my final year project.

ABSTRACT

Deep Convolutional Neural Network (CNN) algorithm has recently gained popularity in many applications such as image classification, video analytic, object recognition and segmentation. Being compute-intensive and memory expensive, CNN computations are common accelerated by GPUs with high power dissipations. Recent studies show implementation of CNN on FPGA and it gain higher advantage in term of energy-efficient and flexibility over Software-configurable-GPUs. However, unlike high-end GPU which have large memory on chip, FPGA on the other hand, has limited memory on chip and could have fatal bottleneck if the kernel was not properly pipelined. Thus, in this work, a FPGA accelerator with a new architecture of deeply pipelined OpenCL kernel is proposed to optimize the accelerator throughput under FPGA constraints such as memory capacity and clock frequency. The proposed framework is verified by implement Tiny-Yolo-V2 on the DE1-SoC. The design development in this project is HLS approach to ease programming effort from writing complex RTL codes and provide fast verification through emulation and profiling tools provided in the OpenCL SDK v16.1.

ABSTRAK

Algoritma Deep Convolutional Neural Network (CNN) baru-baru ini telah mendapat populariti dalam pelbagai aplikasi seperti klasifikasi imej, analitik video, pengiktirafan objek dan segmentasi. Algoritma begini memerlukan pengiraan intensif dan memori yang besar untuk berfungsi. Algoritma ini biasa digunakan dalam GPUs dengan pelepasan kuasa tinggi. Dalam kajian baharu menunjukkan bahawa FPGA memiliki kelebihan dalam kecekapan tenaga dan program fleksibiliti berbanding kepada GPUs. Tetapi, GPU memiliki saiz memori yang besar berbanding kepada FPGA dan ia akan membawa issue kesesakan kepada kernel jika kernel tidak dioptimumkan. Jadi, projek ini memperkenalkan seni bina kernel baharu bagi menyelesaikan masalah memori dan kekerapan jam dalam FPGA. Project ini menggunakan rangka kerja Tiny-Yolo V2 pada DE1-SoC. Projek ini juga menggunakan methodologi HLS bagi mempermudah usaha pengaturcaraan program daripada methodologi RTL yang kompleks. Methodologi HLS juga memberi pengesahan pantas dengan emulasi dan alat profil dalam OpenCL SDK v16.1.

ACKNOWLEDGEMENTS

I would like to thank the Faculty of Electronic and Computer Engineering for the support on my final year project. I want to thank Wei Tank and June Wai for all the help they have gave me. Additionally, I want to express my gratitude to Innovate Malaysia DreamCatcher for loaning the DE1-SoC development Kit. It is a genuine pleasure to express the deep sense of appreciation to my supervisor Professor Dr. Zulkalnain and my core-supervisor Dr Wong yan chew for the encouragement and insight through the whole journey.

TABLE OF CONTENTS

Declaration	
Approval	
Dedication	
Abstract	i
Abstrak	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	ix
List of Tables	xii
List of Symbols and Abbreviations	xiii
List of Appendices	xv
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.1.1 Problem Statement	3
1.1.2 Research Objective	4
1.1.3 Scope of study	5

1.1.4	Thesis Structure	5
CHAPTER 2 Literature review		7
2.1	Machine Learning	7
2.1.1	Deep learning	9
2.1.1.1	Convolutional Neural Network	10
2.1.1.2	Convolutional layers	10
2.1.1.3	Max-pool layers	12
2.1.2	Deep learning-based Object Detection Model	12
2.1.3	YOLO Object Detection Framework	13
2.1.4	YOLOv2 Improvement	15
2.2	FPGA	16
2.2.1	History of FPGA	16
2.2.2	Development kit DE1-SoC	16
2.3	Heterogenous computing	17
2.3.1	OpenCL	20
2.3.2	Intel FPGA for OpenCL SDK v16.1	23
2.4	Previous Project and Research	25
2.4.1	FPGA-based Object Detection using non-deep learning approach	25
2.4.1.1	FPGA-based Real-time moving Target Detection System for unmanned Aerial Vehicle Application	25

2.4.1.2	Real time vision-based object identification and tracking for manufacturing application	26
2.4.2	FPGA-based Object Detection using deep learning approach	27
2.4.2.1	An OpenCL Deep Learning Accelerator on Arria10 FPGA	27
2.4.2.2	PipeCNN: An OpenCL-based FPGA Accelerator for Large-scale convolution neuron networks	28
2.6	Summary	29
CHAPTER 3 METHODOLOGY		30
3.1	Overview of the system	30
3.2	Description of the Project Implementation	33
3.3	Stage 1: Development Environment and program setup	34
3.3.1	Tools Setup	35
3.3.1.1	Intel FPGA for OpenCL SDK v16.1	35
3.3.1.2	Utility tool setup	37
3.3.2	Hardware Integration	38
3.3.2.1	Board Support Packages	38
3.3.2.2	Drivers Installation	38
3.3.2.3	Server-Client TCP/IP for GUI	40
3.4	Phase 2: Algorithm Development	42
3.4.1	CNN Architecture	42
3.4.2	Host Code	43

3.4.3	Kernel Code	48
3.4.3.1	Data Mover Kernel	48
3.4.3.2	Convolutional Kernel	50
3.4.4	Emulation	52
3.4.5	Kernel Profiling	52
3.5	Stage 2: Testing and validation	53
3.5.1	Code Compilation	53
3.5.2	Deployment	54
3.5.3	Performance analysis	55
3.6	Summary	55
CHAPTER 4 RESULTS AND DISCUSSION		56
4.1	The prototype Overview	56
4.2	Resource Utilization	58
4.3	Performance	60
4.4	HTML and JavaScript GUI	64
4.5	Summary	65
CHAPTER 5 CONCLUSION AND FUTURE WORKS		67
5.1	Conclusion	67
5.2	Future Development	68
REFERENCES		69

APPENDICES

LIST OF FIGURES

NO	Title	Page
2.1	Mathematics representation of neuron	8
2.2	Neurology representation of single neuron	8
2.3	Deep learning classification	9
2.4	CNN layers	10
2.5	Convolutional layer	11
2.6	YOLO CNN Architecture	15
2.7	DE1-SoC	17
2.8	Flynn's taxonomy	18
2.9	Fine-grain parallelism	19
2.10	Coarse-grain parallelism	19
2.11	2D range work group	21
2.12	OpenCL-FPGA framework	24
2.13	Algorithm flowchart	25
2.14	Overall DLA architecture	27
2.15	PipeCNN architecture	28
3.1	Overview of the project	31
3.2	OpenCL-based FPGA design flow for CNN accelerator	32
3.3	Block Diagram of the project	33

3.4	Flowchart of Stage1	34
3.5	Download Intel FPGA for OpenCL SDK	35
3.6	Example of Successful installation of SDK	36
3.7	SDK Installation flow diagram	37
3.8	Disk Imager	37
3.9	Content of OpenCL BSP	38
3.10	Enable UVC support	39
3.11	Build AOCL Driver v16.1	40
3.12	Server and client flowchart	41
3.13	Tiny-Yolov2 Architecture	43
3.14	OpenCL initialization	44
3.15	Data Rearrangement	45
3.16	Shared memory pseudocode	47
3.17	Data mapping of the data mover kernel	49
3.18	Pseudo-code of Data-mover	50
3.19	Data Rearrangement	51
3.20	Convolution kernel writing to Global memory	51
3.21	Stage 2 flowchart	53
3.22	OpenCL script	54
4.1	Prototype setup	57
4.2	Graph of Resource utilization	60
4.3	Host program clocking	61
4.4	Kernel Profiling	61
4.5	Memory Analysis	62
4.6	Graph of execution time versus number of compute unit	63

4.7 Demo GUI

65

LIST OF TABLES

NO	Title	Page
2.1	The spec of Deep learning-based Object detection model	13
3.1	Tiny YOLOv2 CNN Architecture	42
4.1	Resource utilization	59
4.2	Configuration version	63
4.3	Best runtime 4CU 8VEC	64

LIST OF SYMBOLS AND ABBREVIATIONS

ALM	Adaptive Logic Module
ALUT	Adaptive LUT
AOC	Altera Offline Compiler
API	Application Programming Interface
BNN	Binarized Neural Network
BSP	Board Support Package
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CU	Compute Unit
DMA	Direct Memory Access
DNN	Deep Neural Network
DSP	Digital Signal Processing
FLOPS	Floating point operations per second
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
HLS	High Level Synthesis
LE	Logic Element
LUT	Look-up Table

MLAB	Memory Logic Array Block
OpenCL	Open Computing Language
OpenGL	Open Graphic Library
OPS	Operations per second
RTL	Register Transfer level
SDK	Software Development Kit
SOC	System on chip
UVC	USB Video device Class

LIST OF APPENDICES

Appendix A: Kernel code.....	74
------------------------------	----

CHAPTER 1

INTRODUCTION

This chapter will discuss the background of the project. Additionally, the problem statement and objectives of this project will be explained in this chapter. Last but not least, the scope and the thesis structure will be included in this chapter as well

1.1 Background

Computer science engineer and researcher have always dreamed of forging a machine that capable of having authentic and extraordinary intelligent. These machines are normally inferred as Artificial intelligence (AI), a machine that can have original thoughts and take actions based on these thoughts. One of the popular field in AI creation is the Machine learning. As the term suggest, it is study of computer learning ability in pattern and feature recognition without explicitly programmed.

Machine learning was discovered way back to 1943 when two early computer scientists named Warren McCulloch and Walter Pitts invented the first computational model of a neuron.

In the recent years, deep neural network has made a big breakthrough in image processing field, like image classification and object detection. In fact, last year AlphaGo won against Lee Sedol, a world-renowned champion in Go game. There is no way AlphaGo was hardcoded given the fact that there are more possible moves in Go than there are number of atoms in the universe. This breakthrough was shocking and later inspires many company to pursue the field of AI e.g. Deepmind and OpenAI. Today AI emerged in many field such as Biomedical, computer vision, model generation, text-bots and more. Yet, even with the state of the art deep learning-based object detection algorithm, modern hardware suffers expensive computation. Recent work [2], [5], [6] shows FPGA outperform GPGPU in performance per watt. In fact, with the recent release of new generation High level synthesis(HLS) OpenCL tool which highly reduces the hardware development time. As a result, FPGA become an attractive selection for CNN accelerator given it has utmost flexibility, short-time-to-market and energy efficiency.

Thus, this project focuses on developing an object detection algorithm that runs on FPGA accelerator, the DE1-SoC development kit. The DE1-SoC is an embedded device that has Hard processor system(ARM) and high-density soft logic fabrics FPGA (Cyclone V 22nm). The object detection algorithm used is based on Tiny Yolov2[1]. In fact, Tiny Yolov2 has a lightweight CNN architecture backbone that run 200fps on a Titan x GPU which is ideal to deploy in an embedded system. In the past, FPGA is commonly developed under RTL environment which takes a longer

development period. In this project, Intel FPGA for OpenCL SDK 16.1 is used as the development environment. Unlike RTL, OpenCL is a High-Level Synthesis(HLS) language that works on higher level of abstraction. The SDK offers ease to program complex algorithm and easy debugging. OpenCL is a parallel framework for accelerating algorithms on heterogeneous system. In fact, the algorithm divided into two parts. The c program is written and cross-compiled for ARM32, which is known as the host code. The computation focuses on the kernel code which is written in OpenCL, so it will be compiled as hardware image executable, aocx file into FPGA to reprogram its soft logics.

1.1.1 Problem Statement

Undoubtedly, one ultimate goal of computer vision field is the machine capability to understand what the image implies just like what human does effortlessly and perform decision and tasks based on what it precept. In fact, the recent breakthrough allows computer to recognize images better than human. However, these deep learning algorithms are computationally expensive and harder to implement in real time without enough computing power supplied. Inference is the part that comes to implement in real world application. Most deep learning training are conducted offline in GPUs due to the utmost performance they offered and availability. Additionally, Main vendor like Nvidia and AMD have made deep learning libraries support for their own devices. Despite the performance GPU offers, GPU are extremely power hunger computing devices which is not a good choice when there is energy constraint in the problem domain. FPGA on the other hand, offers more performance per watt and programmability. The advantages of energy efficiency, reconfigurable interconnections and massive processing elements provided by the Field Programmable Gate Array (FPGA) are naturally suitable to be implemented as