DEVELOPMENT OF A FIXED PIPELINE GPU ON FPGA

KEVIN LEONG WEI CHUNG

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

MAY 2011

To my beloved father and mother

ACKNOWLEDGEMENT

This project would not have been successful without the support from many people. Firstly, I would like to express my highest gratitude to my project supervisor, Dr. Usman Ullah Sheikh for his continuous guidance, patience and support toward me throughout the project.

I would also like to take this opportunity to thank my wife, Lim Choon Mei, for her continuous support during my project development. Besides that, I would also like to thanks my peers, Kua Wee Soon, Lee Si Long and Ong Hui Yien, for their continuous encouragement when I was facing project obstacles, without them, this project will never be accomplished.

Lastly, thanks to all my friends especially my senior Alice Koh Chee, who was always there for support and encouragement throughout the project.

ABSTRACT

In recent years, a lot of research has been carried out to study on how feasible it is to implement GPU on FPGA. Besides that, fixed pipeline GPU has been the main choice for the graphic card vendor nowadays for example nVidia and AMD, and it has been a driving force for us to come out with the idea to have fixed pipeline GPU as a method of implementation into FPGA. In this project, a fixed pipeline GPU has been implemented on Altera Cyclone II FPGA. A fixed point of 12 coordinates undergo a series of fixed pipeline transformation to obtain a 3D Pyramid like polygon. One of the strong points for this project is on the rasterization module, where it does not require any buffer and hence extra buffer can be used to perform other tasks for daily usage. The floating point implementation is being done by using fixed point binary. Furthermore, we are able to prove that FPGA implemented GPU is able to outperform software rendered Direct3D in term of the FPS rate. The final result of the project is a rotation of 3D polygon where we are able to perform translation, rotation speed control, eye point control and look to point control.

ABSTRAK

Semenjak kebelakangan ini, terdapat banyak kajian untuk membuktikan keberkesanan pengaplikasian GPU ke dalam FPGA. Selain daripada itu, GPU berteknologi saluran paip tetap adalah teknologi yang digemari oleh pembekal-pembekal kad grafik yang ternama, seperti AMD dan nVidia. Fenomena ini telah menjadi pendorong utama untuk kita mendapat idea untuk memilih GPU berteknologi saluran paip tetap untuk diimplikasikan ke dalam FPGA. Dalam project ini, GPU bersaluran paip tetap diaplikasikan ke dalam Altera Cyclone II FPGA. 12 koordinat akan mengalami siri transformasi saluran paip tetap untuk memperolehi sebuah poligon 3 dimensi yang merupai piramid. Salah satu unsur yang paling penting dalam project ini adalah dalam modul raster. Modul raster FPGA ini tidak memerlukan buffer. Oleh itu, buffer yang ada dalam sistem ini boleh digunakan untuk tujuan lain. Dalam sistem ini, binari titik tetap digunakan untuk pelaksanaan arimetik floating point. Secara keseluruhannya, kami dapat membuktikan bahawa GPU FPGA mampu menghasilkan FPS dengan lebih cepat berbanding dengan teknologi pemprosesan 3 dimensi grafik perisai yang menggunakan Direct3D. Hasil di akhir projek ini adalah sebuah putaran poligon 3 dimensi di mana kita mampu melakukan penterjemahan, pengawalan kelajuan putaran, pengawalan ke titik penglihatan mata dan pengawalan melihat ke titik objek.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

T_{px}	-	Translation X
T_{py}	-	Translation Y
T_{pz}	-	Translation Z
S_x	-	Scaling X
S_y	-	Scaling Y
S_z	-	Scaling Z
R_x	-	Rotation X
R_y	-	Rotation Y
R_z	-	Rotation Z
S_z	-	Scaling Z
u	-	Right vector
v	-	Up vector
n	-	View direction vector
M_{view}	-	View Matrix
M_{proj}	-	Projective Matrix
M_{screen}	-	Screen-space Matrix
θ	-	Theta, degrees

LIST OF ABBREVIATIONS

3D	-	Three Dimension
ASM	-	Algorithmic State Machine
CRT	-	Cathode Ray Tube
FPGA	-	Field-Programmable Gate Array
GPU	-	Graphical Processing Unit
PC	-	Personal Computer
PLL	-	Phase-locked Loop
PS2	-	Play Station 2
RAM	-	Random-access memory
ROM	-	Random-only memory
RTL-CS	-	Register Transfer Level – Control Signal
VGA	-	Video Graphic Array

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

In this era, 3D technology is becoming increasingly important in all kind of aspect, as more and more applications are using 3D technology ranging from medical devices, entertainment, portable devices down to specific types of research application. In order to get a good 3D processing throughput, various methods of GPU has been developed for example, cell based GPU (Intel's Larabees and PS3) and parallel fixed pipeline GPU which is widely being used by NVIDIA and AMD.

Fixed pipeline 3D GPU is a very common technique used by most current graphics cards today, although the number of pipeline stages varies among the graphics card, however, the fundamental concept remains the same. By definition, 3D pipeline is a series of processes that must be executed on a collection of model to generate 2D representation of the scene.

FPGA or field programmable Gate Array can be designed to behave just like a GPU. Due to the cost of FPGAs, it is still considerable expensive compared to common graphics cards. However, as the technology advances, the cost of the FPGA is becoming more affordable. By using FPGA, performance can be scaled up tremendously with the cost of more complex design and longer development time, not to mention that we can now get FPGAs that are relatively affordable.

In this project, we are targeting to design a 3D fixed pipeline transformation or in order word a simplified version of GPU into FPGA. By doing this, we are targeting to be able to speed up the 3D transformation speed as compared to a software rendered transformation by using a PC.

1.2 Problem Statement

3D transformation on software is found to be often slow and time consuming. Thus, hardware acceleration for 3D transformation is crucial. In this work, an accelerated 3D transformation hardware based on GPU will be developed to increase the performance of 3D rendering when compared to software rendering. The hardware design is to be implemented on FPGA so that it can be implemented on other system-on-chip.

1.3 Objectives

The objective of this project is to implement a fixed pipeline GPU on Altera DEII FPGA board, the end result is to be displayed on a display monitor. Below are the three main objectives of this project.

i. To implement 3D transformation capabilities on Altera DEII FPGA board.
ii. Final result is to be displayed on a display monitor.
iii. To speed up the 3D transformation algorithm using the Altera DEII board compared to the software rendering of Direct3D.

1.4 Scope of Study

This project requires an in-depth understanding on the 3D transformation pipeline and its process. It also requires a very good knowledge on verilog and FPGA code in order to implement it successfully on FPGA. Basically, the scope of this study can be categorized as below:

1. Design a fixed pipeline GPU consisting of world transformation -> view transformation -> projective transformation -> screen-space transformation -> rasterization,

2. Implement the 3D transformation algorithm using Verilog that consists of the unit module and scheduler.

3. To be able to display the end result of the 3D transformation polygon on a display monitor, in order to archive this, a VGA controller needs to be created and implemented on the FPGA.

REFERENCES

Niklas Knutsson (2005), AN FPGA-based 3D Graphics System, *Master's thesis completed in Electronics Systems, LinkoPing,*

Manisha Singh and Sahil Aror (2007), 3D WireMesh Generator, *ECE576 Final Project, Cornell University*

Prasanna Sundararajan (2010), High Performance Computing Using FPGAs, *WP375 (v1.0) September 10, XILINX*

Keegan McAllister and Shirley (2007), Triangle rasterization using barycentric coodinate, October 23, 2007

F.Bensaali A.Amira A.Bouridane (2003), An FPGA Based Coprocessor for 3D Affine Transformations, Field-Programmable Technology (FPT), *Proceedings. 2003 IEEE International Conference, page 288-p291*

Shuli Gao, Dhamin Al-Khalili, Noureddine Chabini (2009), Two Level Decomposition Based Matrix Multiplication for FPGAs, *Electronics, Circuits, and Systems, ICECS 2009. 16th IEEE International Conference, p427-p430*

Lee W. Howes, Paul Price, Oskar Mencer, Olav Beckmann, Oliver Pell (2006), Comparing FPGAS to Graphics Accelerators and the Playstation 2 using a Unified Source Desription, *Field, Programmable Logic and Applications,. FPL '06. International Conference, pg1-p6*

Colin Flanagan, DDR algorithm and barsenham line drawing algorithm, *http://www.cs.helsinki.fi/group/goa/mallinnus/lines/*

Dr John Loomis, TDE2 Digital Lab Project (VGALAP3), *www.johnloomis.org*

directxtutorial, The Ultimate DirectX tutorial, *www.directxtutorial.com*

Appendix A
List of Files included in the attached CD

1. Verilog Top level entity “MASTER_FPGA”
   a. master_fpga.qpf – Master FPGA project File
   b. master_fpga.sof – sof file to be programmed on FPGA
   c. Presentation slides – Project II presentation slides and video.