

HARDWARE IMPLEMENTATION OF RGB TO HSL  
CONVERTER USING FPGA

SOON VOON SIEW

UNIVERSITI MALAYSIA PERLIS

2016



# HARDWARE IMPLEMENTATION OF RGB TO HSL CONVERTER USING FPGA

by

SOON VOON SIEW  
(1430211288)

A thesis submitted  
in fulfilment of the requirements for the degree of  
Master of Science in Computer Engineering

School of Computer and Communication Engineering  
UNIVERSITI MALAYSIA PERLIS

2016

# UNIVERSITI MALAYSIA PERLIS

## DECLARATION OF THESIS

Author's full name : SOON VOON SIEW  
Date of Birth : 26-10-1989  
Title : HARDWARE IMPLEMENTATION OF RGB TO HSL  
CONVERTER USING FPGA  
Academic Session : Semester 1 – 2014/2015

I hereby declare that the thesis becomes the property of Universiti Malaysia Perlis (UniMAP) and to be placed at the library of UniMAP. This thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1972)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that the thesis is to be made immediately available as hard copy or on-line open access (full text)

I, the author, give permission to the UniMAP to produce this thesis in whole or in part for the purpose of research or academic exchange only (except during a period of \_\_\_\_\_ years, of so requested above).

Certified by:

\_\_\_\_\_  
SIGNATURE

\_\_\_\_\_  
SIGNATURE OF SUPERVISOR

891026-07-5230  
NEW IC NO. / PASSPORT NO.)

DR. PHAKLEN EHKAN  
NAME OF SUPERVISOR

Date: \_\_\_\_\_

Date: \_\_\_\_\_

NOTES: \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.

In the name of God, Most Gracious, Most Merciful

©This item is protected by original copyright

Dedicated to

My beloved parent

## ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this thesis. A special gratitude I give to my supervisor, Dr. Phaklen Ehkan, whose contribution in simulating suggestions and encouragement, helped me to coordinate my project.

I wish to thank the Malaysia Ministry of Higher Education for providing the research grant under myBrain15 scholarship (MyMaster) from February 2014 until February 2016.

Thank you.

## TABLE OF CONTENTS

	<b>Page</b>
<b>DEDICATION</b>	i
<b>ACKNOWLEDGMENTS</b>	ii
<b>LIST OF TABLE</b>	viii
<b>LIST OF FIGURE</b>	ix
<b>LIST OF ABBREVIATION</b>	xiii
<b>ABSTRAK</b>	xiv
<b>ABSTRACT</b>	xv
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope of Works	
1.5 Thesis Organization	4
<b>2 LITERATURE REVIEW</b>	<b>6</b>
2.1 Introduction	6
2.2 Field Programmable Gate Arrays (FPGA)	6

2.3	Hue Saturation Luminance (HSL) Colour Space	10
2.3.1	RGB to HSL Conversion	12
2.4	Application of Hue-Saturation Colour Space	16
2.4.1	Object Detection	16
2.4.2	Image Segmentation	17
2.4.3	Industrial Application	18
2.5	Implementation Platform	19
2.6	Hardware Architecture	23
2.6.1	Parallel Architecture	23
2.6.2	Pipeline Architecture	25
2.6.3	Super-pipeline Architecture	26
2.6.4	Superscalar Architecture	27
2.7	Critical Review	28
<b>3</b>	<b>METHODOLOGY</b>	<b>31</b>
3.1	Introduction	31
3.1.1	Research Flow	32
3.2	RGB to HSL Conversion Algorithm	33
3.2.1	Simplification Derivation	35
3.2.2	Arithmetic Complexity Comparison	39
3.3	C Code Programming Conversion Implementation	41

3.4	Hardware Conversion Architecture	41
3.4.1	Parallel Architecture for Hardware Converter	41
3.4.2	Pipeline Architecture for Hardware Converter	43
3.5	Hardware Implementation	44
3.5.1	Description of the RGB to HSL System Architecture	45
3.5.2	Testing Speed and Accuracy Architecture	46
3.5.3	Testing Flow Chart	46
<b>4</b>	<b>RESULT AND DISCUSSION</b>	<b>48</b>
4.1	Introduction	48
4.2	C Code Conversion Algorithm	48
4.2.1	Generation of Reference Result	49
4.3	Parallel Architecture of Hardware Converter	50
4.3.1	Internal Architecture of Min_Max_Selector Module	51
4.3.2	Internal Architecture of Hue Module	53
4.3.3	Internal Architecture of Saturation Module	54
4.3.4	Internal Architecture of Luminance Module	55
4.4	Pipeline Internal Architecture	55
4.4.1	Pipeline Architecture – Stage 1	58
4.4.2	Pipeline Architecture – Stage 2	59
4.4.3	Pipeline Architecture – Stage 3	60



4.4.4	Pipeline Architecture – Stage 4	61
4.4.5	Pipeline Architecture – Stage 5	62
4.4.6	Pipeline Architecture – Stage 6	63
4.4.7	Pipeline Architecture – Stage 7	64
4.5	Simulation Results of Parallel and Pipeline Converter	65
4.6	Performance Testing Architecture	66
4.6.1	Hardware Testing Architecture and Method	67
4.6.2	Accuracy for Parallel and Pipeline Architecture for 10 Hz Data Rate against Reference C Result	71
4.6.3	Accuracy of Parallel Architecture to Reference C	72
4.6.4	Accuracy of Pipeline Architecture to Reference C	75
4.6.5	Comparison of HSL Accuracy between Both of Parallel and Pipeline Architecture	78
4.7	Real Hardware Implementation	81
4.7.1	Hardware Implementation – Parallel RGB to HSL	86
4.7.2	Hardware Implementation – Pipeline RGB to HSL	88
4.7.3	Hardware Implementation – Image Comparison	89
4.8	FPGA Resources Utilization Summary	95
<b>5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>97</b>
5.1	Introduction	97

5.2	Conclusion	97
5.3	Future work	98
	<b>REFERENCES</b>	<b>100</b>
	<b>APPENDICES</b>	
	APPENDIX A	104
	APPENDIX B	108
	APPENDIX C	109
	<b>LIST OF PUBLICATION</b>	<b>111</b>

©This item is protected by original copyright

## LIST OF TABLES

<b>Table</b>	<b>Description</b>	<b>Page</b>
2.1	HSV and HSL Equations	15
2.2	HSI Equations	16
2.3	Summary of Conversion Equations Implementation Platform	23
3.1	Comparison between Agoston's Formulas with Simplification Formulas	40
3.2	Top Level Entity Functional Block Description	42
3.3	Description of the RGB to HSL System Architecture	45
4.1	Accuracy of 10 Hz Data Rate for 1024 RGB Samples	71
4.2	Summary of the FPGA Device Utilization	96

## LIST OF FIGURES

<b>Figure</b>	<b>Description</b>	<b>Page</b>
2.1	Illustration of Logic Cell	7
2.2	FPGA Block Structure	8
2.3	Designing Flow of FPGA	10
2.4	HSL Dual Cones in Cylinder Model	12
2.5	Generation of Hue Rainbow Colour	12
2.6	Parallel Architecture	24
2.7	Pipeline Architecture	25
2.8	Super-pipeline Architecture	27
2.9	Superscalar Architecture	28
3.1	Research Flow	32
3.2	Parallel Architecture for RGB to HSL Converter	42
3.3	Pipeline Architecture for RGB to HSL Converter	43
3.4	Top Level Architecture of Hardware Testing	44
3.5	Testing Architecture	46
3.6	Testing Flow of the RGB to HSL Converter	47
4.1	Reference Results from C Code	49
4.2	Input RGB Samples and HSL Output Results	50

4.3	Parallel Architecture of RGB to HSL Converter	51
4.4	RTL of Min_Max_Selector Module	52
4.5	RTL Structure of Hue Module	53
4.6	Internal Architecture of Saturation Functional Block	54
4.7	Internal Architecture of Luminance Functional Block	55
4.8	Pipeline Converter Internal Architecture	57
4.9	Pipeline Architecture Stage 1	58
4.10	Pipeline Architecture Stage 2	59
4.11	Pipeline Architecture Stage 3	60
4.12	Pipeline Architecture Stage 4	61
4.13	Pipeline Architecture Stage 5	62
4.14	Pipeline Architecture Stage 6	63
4.15	Pipeline Architecture Stage 7	64
4.16	Simulation Results for Parallel Architecture	65
4.17	Simulation Results for Pipeline Architecture	66
4.18	Converters Performance Test Module	67
4.19	Performance Testing Flow	69
4.20	Conversion Iteration Process Flow	70
4.21	Hue Accuracy for Parallel Architecture	72
4.22	Saturation Accuracy for Parallel Architecture	73
4.23	Luminance Accuracy for Parallel Architecture	74

4.24	Hue Accuracy for Pipeline Architecture	75
4.25	Saturation Accuracy for Pipeline Architecture	76
4.26	Luminance Accuracy for Pipeline Architecture	77
4.27	Parallel and Pipeline Hue Conversion Accuracy	78
	Comparison	
4.28	Saturation Accuracy for Parallel and Pipeline	79
	Comparison	
4.29	Frequency Respond of the Luminance Hardware	80
	Converter	
4.30	Hardware Implementation Equipment Setup	81
4.31	Captured Reference Red Channel Image	82
4.32	Captured Reference Green Channel Image	83
4.33	Captured Reference Blue Channel Image	83
4.34	Combined Colour Image	84
4.35	C Code Converter Hue Image	84
4.36	C Code Converter Saturation Image	85
4.37	C Code Converter Luminance Image	85
4.38	Parallel RGB to HSL Converter - Hue Output Image	86
4.39	Parallel RGB to HSL Converter - Saturation Output	87
	Image	

4.40	Parallel RGB to HSL Converter - Luminance Output Image	87
4.41	Pipeline RGB to HSL Converter - Hue Output Image	88
4.42	Pipeline RGB to HSL Converter - Saturation Output Image	88
4.43	Pipeline RGB to HSL Converter - Luminance Output Image	89
4.44	Reference Hue Image	90
4.45	Parallel Hue Image	90
4.46	Pipeline Hue Image	91
4.47	Reference Saturation Image	92
4.48	Parallel Saturation Image	93
4.49	Pipeline Saturation Image	93
4.50	Reference Luminance Image	94
4.51	Parallel Luminance Image	94
4.52	Pipeline Luminance Image	95

## LIST OF ABBREVIATIONS

<b>CCTV</b>	Closed Circuit TV
<b>RGB</b>	Red, Green, Blue
<b>HSL</b>	Hue, Saturation, Luminance
<b>HSV</b>	Hue, Saturation, Value
<b>YCbCr</b>	Y(luma), Cb(Chroma Blue), Cr(Chroma Red)
<b>FPGA</b>	Field Programmable Gate Array
<b>VGA</b>	Video Graphics Array
<b>VHDL</b>	VHSIC (Very High Speed Integrated Circuit) Hardware
<b>HSI</b>	Hue, Saturation, Intensity
<b>ROM</b>	Read Only Memory
<b>VLSI</b>	Very Large Scale Integration



## PERLAKSANAAN PERKAKASAN UNTUK PENUKARAN RGB KE HSL MENGUNAKAN FPGA

### ABSTRAK

Model warna RGB (Merah, Hijau, Biru) adalah asas kepada model warna dan dicantumkan bersama untuk menghasilkan julat warna penuh. RGB tidak dapat menghasilkan maklumat yang mencukupi untuk menganalisa imej digital. HSL (Hue, Saturation, and Luminance) adalah model warna lain untuk RGB. HSL berkebolehan untuk menyediakan maklumat yang boleh digunakan sepenuhnya seperti sudut warna, ketepuan warna dan kecerahan warna. Maklumat ini digunakan sepenuhnya untuk menganalisis imej. Dalam kerja ini, pelaksanaan algoritma penukaran matematik RGB ke HSL menggunakan bahasa VHDL. FPGA membolehkan keupayaan keselarian untuk mempercepatkan proses penukaran daripada langkah-langkah memproses dan hanya menggunakan satu kitaran sahaja. Persamaan RGB ke HSL dilaksanakan dengan menggunakan dua kaedah iaitu senibina '*parallel*' dan '*7-stages pipeline*' dengan menggunakan bahasa VHDL. Senibina '*parallel*' hanya mempunyai satu tempoh masa kelengahan data. Kedua-dua kaedah ini boleh menghasilkan nilai HSL bagi setiap pixel tanpa kelengahan data daripada 10Hz hingga 150MHz. Senibina-senibina '*parallel*' dan '*pipeline*' untuk penukar RGB ke HSL telah mencapai ketepatan dengan pengesahan perkakasan masing-masing sehingga 99% dan 98%.

# **HARDWARE IMPLEMENTATION OF RGB TO HSL CONVERTER USING FPGA**

## **ABSTRACT**

The RGB (Red, Green, Blue) colour model is the basic colour model and add together to produce full colour range. RGB is unable to produce sufficient information for digital image analysis. The HSL (Hue, Saturation, and Luminance) is another colour model for RGB. HSL is capable to provide other useful information such as colour in degree, saturation of the colour and brightness of colour. These information are useful for image analysis. In this work, the implementation of RGB to HSL mathematic conversion algorithm in FPGA is using VHDL language. FPGA enables parallelism and pipelining capabilities to speed up conversion process of the processing steps and consumes only one cycle. The RGB to HSL equation is implemented by using two methods which are parallel and 7-stages pipeline architectures using VHDL language. Parallel architecture has only one clock period of data latency. These two methods can produce HSL value for each pixel without data latency from 10Hz to 150MHz. The parallel and pipeline architectures for RGB to HSL converter have achieved rate of accuracy with the hardware verification up to 99% and 98%, respectively.

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Nowadays digital image is the most popular image form in the world. With its highly portable capability, mathematical operation friendly and small memory storage size has caused digital image to empower the imaging technology in the world. The digital image can be produced by digital camera and close circuit television (CCTV). The pixel colour in digital image is represented by several colour spaces such as Red-Green-Blue (RGB), Hue-Saturation-Luminance (HSL), Hue-Saturation-Value (HSV), Hue-Saturation-Intensity (HSI), and Luma-Chroma (YCbCr).

Each colour model has its specific application and features. RGB colour space is one of the most common use colour space in digital image. This is because RGB colour space is simple and compatible with binary machine. Some application of digital image required to use other colour space for processing such as HSL space. HSL colour space is a popular space used in machine vision technology for object recognition process. Many of the object recognition process are developed base on object colour recognition to detect the target. Therefore the HSL space used to enhance the image colour during image processing on the image. Hue is the colour plane of the RGB, saturation is the colour intensity of the RGB in the image and video while luminance is the colour brightness of the RGB.

By using Altera FPGA DE2 board as hardware platform, it helps to increase the conversion yield. RGB to HSL conversion algorithm are implemented using software and

hardware languages. The VHDL is in order to simplify the conversion steps and speed up the process.

## **1.2 Problem Statement**

Currently, there are many existing RGB to HSL converter method used in image processing available but most of them are based on software. The problem with the software is that its implementation is not suitable in practically used if compared to hardware based. It consumes multiple clock cycle to complete 1 operation. Besides, the RGB colour model is an additive colour model in which red, green, and blue light are added together in numerous ways to reproduce a broad array of colours but there is limitation for RGB colour model to provide sufficient information for image analysis. HSL is a colour model that different from RGB. HSL provides the information such as colour in degree, the saturation of the colour and the brightness of the colour. This information is much more suitable for image analysis. Parallelism and pipelining capabilities of FPGA has simplified the processing steps and only consumes one clock cycles to produce the HSL value for each pixel without data latency. Hence, the embedded FPGA-based RGB to HSL converter with less computational time and capable to work on images analysis should be developed.

### **1.3 Objectives**

The main objective of this research project is to fully implement RGB to HSL hardware FPGA-based converter. In order to do that there are few steps need to be taken.

These include

- i. to develop of parallelism and pipeline architectures of RGB to HSL hardware converter.
- ii. to analyse hardware conversion output accuracy for both converter as well as characterizing the implementation issues that limit the achievable throughput.
- iii. to investigate and analysis a maximum operation frequency for both converter to achieve a high performance, evaluating the interaction between algorithm and hardware efficiency to determine which variants are most suitable for hardware implementation.
- iv. to compare hardware resources usage for both converter.

### **1.4 Scope of Works**

This research is bound to the following scopes:

- i. The selection of a formula used RGB to HSL conversion and development of C code and VHDL for it.
- ii. Using a serial RS232 to collect data in text file and interfacing a CCTV and VGA to the Altera FPGA DE2 Board.

- iii. Implementing simulation and applied on the FPGA hardware board as well as capturing the image into SD card for analysis and performance.
- iv. Performances measurement is evaluated in terms of frequency used and percentage of accuracy rate.

## 1.5 Thesis Organization

This thesis consists of 5 chapters as following:

Chapter 1 briefly introduces about digital images and how it's represented by several colour spaces such as RGB, HSL, HSV and YCbCr. The explanation of FPGA and how it works is also presented. Problem statement, objectives, and scope of work are stated clearly.

Chapter 2 conducts a review to previous works. It also covers the theories and conversion of RGB to HSL converter. Besides, an explanation on the application, implementation platform and method of implementation on hardware architecture also presented.

Chapter 3 describes the methodology of coding and hardware implementing in the Altera FPGA DE2 Board. This covers the work from software development, hardware implementation and hardware real time performance. The details of hardware conversion architectures are also mentioned in this chapter

Chapter 4 presents the results and discussions based on designing VHDL code in the board by using Altera Quartus II software and data analysis on the image results. It also demonstrates the stages of pipelining architecture and performances of testing architecture.

Chapter 5 summarizes about the findings, conclusions and future work recommendations for the RGB to HSL converter. Following these are the lists of references, and sample of programming and some testing data/results.

©This item is protected by original copyright

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

This chapter discusses the theories and principles of Hue, Saturation, and Luminance (HSL) colour space. It also reviews some of the related work in related field and includes the concept, implementation platform, design structure, and applications of the HSL colour space.

#### 2.2 Field Programmable Gate Array (FPGA)

FPGA is a type of semiconductor device that contain programmable logic and interconnections which mostly used in logic or digital electronic circuits. The programmable logic components or logic blocks as they are known may consist of anything from logic gates, through to memory elements or blocks of memories, or almost any element. FPGA supports thousands of gates and popular for prototyping integrated circuit (IC) designs. Once a design is set, hardwired chips will be produced to faster performance. FPGA chip is programmable and reprogrammable which is considered as an advantage of it. In this way, it becomes a large logic circuit that can be configured according to a design, but if changes are required it can be reprogrammed with an update. FPGA contains many identical logic cells that can be viewed as standard components. Each design is