

RECONFIGURABLE ADDRESS GENERATION UNIT FOR 2D CORRELATION
IN FPGA

HENG AI HOON

UNIVERSITI TEKNOLOGI MALAYSIA

RECONFIGURABLE ADDRESS GENERATION UNIT FOR 2D CORRELATION
IN FPGA

HENG AI HOON

A project report submitted in partial fulfilment 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 2012

To my beloved father and mother

ACKNOWLEDGEMENT

First of all, I would like to put on record my indebtedness to my supervisor Prof. Dr. Mohamed Khalil Hani for his guidance and teachings throughout the progress of this project proposal. He has provided me with many opportunities to learn more and gain experiences in this field of study. This report would not have been successful without his advice.

I would also like to express my gratitude to all authors and experts whose research results and findings that I have been referring to that provides crucial knowledge and clarifications in completing this project.

Special thanks to my friends and all those who have helped me in one way or another throughout this project.

ABSTRACT

2D correlation has been commonly used in image processing. In general, performance of the 2D correlation function depends on its processing speed, memory speed as well as address calculation speed. As the processing and memory speed increase, the address calculation speed becomes bottleneck for overall performance. It is thus necessary to accelerate the address calculation or generation by implementing it in hardware like FPGA rather than depending on software to calculate the addresses; such hardware is known as address generation unit (AGU). Prior arts of reconfigurable AGU can be reconfigured to generate address for different digital signal processing (DSP) functions including 2D correlation; however, they don't support address generation for different designs of 2D correlation circuits. None of the prior arts of AGU able to handle image edge condition while considering data reuse in 2D correlation circuit. Furthermore, prior arts of AGU have never been implemented in FPGA. In this paper, a reconfigurable AGU for different designs of 2D correlation in FPGA, which takes care of image edge condition while considering data reuse, is presented. The proposed reconfigurable AGU is targeted for two different architectures of 2D correlation circuit. The two architectures of 2D correlation circuit, which work together with the reconfigurable AGU, are also designed. The proposed reconfigurable AGU reduces the circuit area by sharing or reusing the common components such as adder, comparator, register and etc. In general, the reconfigurable AGU reduces circuit area by 30% as compared to integrating two dedicated AGUs for two different architectures of 2D correlation circuit. The maximum speed of the reconfigurable AGU is 125MHz for Cyclone III device targeting FPGA.

ABSTRAK

Korelasi 2D telah biasa digunakan dalam pemrosesan imej. Secara amnya, prestasi fungsi korelasi 2D bergantung kepada kelajuan pemrosesan, kelajuan memori serta kelajuan untuk alamat pengiraan. Selaras dengan peningkatan kelajuan pemrosesan dan memori, kelajuan alamat pengiraan menjadi kejejalan untuk prestasi keseluruhan. Oleh itu, adalah perlu untuk mempercepatkan pengiraan alamat dengan menggunakan perkakasan seperti FPGA dan bukannya bergantung kepada perisian untuk mengira alamat; perkakasan itu dikenali sebagai “unit pengenerasi alamat” (AGU). Generasi AGU yang lama boleh diatur semula untuk menjana alamat untuk fungsi pemrosesan isyarat digital (DSP) yang berbeza termasuk korelasi 2D; bagaimanapun, mereka tidak menyokong pengiraan alamat untuk litar korelasi 2D dengan reka bentuk yang berbeza. Tambahan pula, generasi AGU terlebih dahulu tidak pernah dilaksanakan di FPGA. Dalam kertas ini, AGU yang boleh dikonfigur semula untuk korelasi 2D dengan reka bentuk yang berbeza di FPGA, yang mengambil peduli keadaan tepi imej sambil mempertimbangkan penggunaan semula data, dibentangkan. AGU yang dicadangkan mensasarkan untuk dua litar korelasi 2D dengan reka bentuk yang berbeza. Kedua-dua litar korelasi 2D, yang bekerja bersama-sama dengan AGU, juga direka. AGU cadangan mengurangkan kawasan litar dengan berkongsi atau menggunakan semula komponen yang sama seperti penambah, komparator, “register” dan sebagainya. Secara amnya, AGU yang boleh dikonfigur semula mengurangkan kawasan litar sebanyak 30% berbanding dengan mengintegrasikan dua AGU yang berlainan untuk dua litar korelasi 2D yang berbeza. Kelajuan maksimum untuk AGU yang boleh dikonfigur adalah 125MHz untuk “Cyclone III” yang mensasarkan FPGA.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	ix
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xviii
	LIST OF APPENDICES	xix
1	INTRODUCTION	1
	1.1 Problem Background	1
	1.2 Problem Statement	2
	1.3 Objectives	3
	1.4 Scope of Study	4
2	LITERATURE REVIEW	5
3	THEORY AND METHODOLOGY	11
	3.1 2D Correlation	11
	3.2 Project Approach	14
	3.2.1 Project Flow	14
	3.2.2 Project Planning	16
	3.2.3 Tool involved in the project	16

4	MODELLING AND DESIGN	18
4.1	Arch1	19
4.2	AGU1	25
4.3	Arch2	28
4.4	AGU2	35
4.5	Reconfigurable AGU	38
5	RESULT AND DISCUSSION	45
5.1	AGU1 for Arch1	46
5.2	AGU2 for Arch2	51
5.3	Reconfigurable AGU for Arch1 and Arch2	58
5.4	Performace of Design	70
6	CONCLUSION AND FUTURE WORK	71
6.1	Conclusion	71
6.2	Future Work	72
	REFERENCES	74
	Appendix A	75

LIST OF TABLES

TABLE NO.	TITLE	PAGE
3.1	Project Gantt chart	16
4.1	RTL-CS table of Arch1	24
4.2	RTL-CS table of Arch2	33
5.1	Comparing the output of Arch1 using AGU1 with Matlab output for 192x128 image and 5x5 kernel	49
5.2	Comparing the output of Arch1 using AGU1 with Matlab output for 128x200 image and 3x3 kernel	50
5.3	Comparing the output of Arch2 using AGU2 with Matlab output for 192x128 image and 5x5 kernel	55
5.4	Comparing the output of Arch2 using AGU2 with Matlab output for 128x200 image and 3x3 kernel	57
5.5	Comparing the output of Arch1 using reconfigurable AGU with Matlab output for 192x128 image and 5x5 kernel	63
5.6	Comparing the output of Arch1 using reconfigurable AGU with Matlab output for 128x200 image and 3x3 kernel	65
5.7	Comparing the output of Arch2 using reconfigurable AGU with Matlab output for 192x128 image and 5x5 kernel	67
5.8	Comparing the ouput of Arch2 using reconfigurable AGU with Matlab output for 128x200 image and 3x3 kernel	69

5.9	Resource utilization	70
-----	----------------------	----

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Image read and store for 2D correlation.	2
2.1	Address calculation from coordinate x and y	6
2.2	The detailed circuit diagram of address generation circuit for matrix addressing sequence	7
2.3	The detailed circuit diagram of clipping function in Figure 2.2	8
2.4	Overall function diagram of the memory reconfiguring unit	9
2.5	Functional diagram of primitive generation unit	9
2.6	Hardware schematic of AGU for data fetch of convolution kernel	10
3.1	Operation of spatial filtering	12
3.2	Illustration of kernel movement in convolution using a 3x3 kernel	12
3.3	Simplest averaging filter or box filter	12
3.4	Another implementation of simplest averaging filter	12
3.5	A sample image	13
3.6	Image padded with edge pixels.	13
3.7	Illustration of data reuse for next processing.	13

3.8	Flow Chart of Project	15
4.1	Arch1 vs. Arch2	19
4.2	DFG and schedule of Arch1 for a 3x3 kernel	20
4.3	ASM chart of Arch1 for a 3x3 kernel	21
4.4	High level block diagram of Arch1 and AGU1	22
4.5	Functional block diagram of datapath unit of Arch1 for 8x4 image and 3x3 kernel	23
4.6	Functional block diagram of control unit of Arch1	24
4.7	Address sequence and the corresponding sequence of coordinate X and Y for Arch1	26
4.8	ASM chart of AGU1	27
4.9	Functional block diagram of AGU1	28
4.10	SFG and schedule of Arch2 for a 3x3 kernel	29
4.11	ASM chart of Arch2	30
4.12	High level block diagram of Arch2 and AGU2.	31
4.13	Functional block diagram of datapath unit for Arch2 for a 3x3 kernel.	32
4.14	Functional block diagram of CU for Arch2	33
4.15	Address sequence and the corresponding sequence of coordinate X and Y for Arch2	36
4.16	ASM chart of AGU2	37
4.17	Functional block diagram of AGU2	38

4.18	Functional block diagram of AGU1 with similar components highlighted	40
4.19	Functional block diagram of AGU2 with common components highlighted	41
4.20	Functional block diagram of reconfigurable AGU	42
4.21	High level block diagram of module unit which shows the interconnection between reconfigurable AGU and Arch1	43
4.22	High level block diagram of module unit which shows the interconnection between reconfigurable AGU and Arch2	44
5.1	Original bean image of dimension 192x128	46
5.2	Original bone image of dimension 128x200	46
5.3	Snapshot of ModelSim waveform showing AGU1 generates correct input and output addresses for 8x4 image and 5x5 kernel	47
5.4	Modelsim timing fragment showing AGU1 generates correct input and output addresses for 8x4 image and 5x5 kernel.	47
5.5	Snapshot of ModelSim waveform showing AGU1 generates correct input and output addresses for 10x5 image and 3x3 kernel	47
5.6	Modelsim timing fragment showing AGU1 generates correct input and output addresses for 10x5 image and 3x3 kernel	47
5.7	Snapshot of ModelSim waveform showing AGU1 generates correct input and output addresses for 192x128 image and 5x5 kernel for Arch1	48
5.8	Snapshot of Modelsim waveform showing the output generated by Arch1 using the address generated by AGU1 for 192x128 image and 5x5 kernel	48

5.9	Comparing the output image of Arch1 using AGU1 with Matlab output for 192x128 image and 5x5 kernel	49
5.10	Snapshot of ModelSim waveform showing AGU1 generates correct input and output addresses for 128x200 image and 3x3 kernel for Arch1	49
5.11	Snapshot of ModelSim waveform showing the output generated by Arch1 using the address generated by AGU1 for 128x200 image and 3x3 kernel	49
5.12	Comparing the output image of Arch1 using AGU1 with Matlab output for 128x200 image and 3x3 kernel	50
5.13	Snapshot of ModelSim waveform showing AGU2 generates correct input and output addresses for 8x4 image and 5x5 kernel	52
5.14	Modelsim timing fragment showing AGU2 generates correct input and output addresses for 8x4 image and 5x5 kernel	52
5.15	Snapshot of ModelSim waveform showing AGU2 generates correct input and output addresses for 10x5 image and 3x3 kernel	53
5.16	Modelsim timing fragment showing AGU2 generates correct input and output addresses for 10x5 image and 3x3 kernel	53
5.17	Snapshot of ModelSim waveform showing AGU2 generates correct input and output addresses for 192x128 image and 5x5 kernel for Arch2	54
5.18	Snapshot of ModelSim waveform showing the output generated by Arch2 using the address generated by AGU2 for 192x128 image and 5x5 kernel	54
5.19	Comparing the output image of Arch2 using AGU2 with Matlab output for 192x128 image and 5x5 kernel	55

5.20	Snapshot of ModelSim waveform showing AGU2 generates correct input and output addresses for 128x200 image and 3x3 kernel for Arch2	56
5.21	Snapshot of ModelSim waveform showing the output generated by Arch2 using the address generated by AGU2 for 128x200 image and 3x3 kernel	56
5.22	Comparing the output image of Arch2 using AGU2 with Matlab output for 128x200 image and 3x3 kernel	57
5.23	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch1 for 8x4 image and 5x5 kernel	59
5.24	ModelSim timing fragment showing reconfigurable AGU generates correct input and output addresses for Arch1 for 8x4 image and 5x5 kernel	59
5.25	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch1 for 10x5 image and 3x3 kernel	59
5.26	Modelsim timing fragment showing reconfigurable AGU generates correct input and output addresses for Arch1 for 10x5 image and 3x3 kernel	60
5.27	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch2 for 8x4 image and 5x5 kernel	60
5.28	ModelSim timing fragment showing reconfigurable AGU generates correct input and output addresses for Arch2 for 8x4 image and 5x5 kernel	60

5.29	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch2 for 10x5 image and 3x3 kernel	61
5.30	ModelSim timing fragment showing reconfigurable AGU generates correct input and output addresses for Arch2 for 10x5 image and 3x3 kernel	61
5.31	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch1 for 192x128 image and 5x5 kernel	62
5.32	Snapshot of ModelSim waveform showing the output generated by Arch1 using the address generated by reconfigurable AGU for 192x128 image and 5x5 kernel	62
5.33	Comparing the output image of Arch1 using reconfigurable AGU with Matlab output for 192x128 image and 5x5 kernel	63
5.34	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch1 for 128x200 image and 3x3 kernel	64
5.35	Snapshot of ModelSim waveform showing the output generated by Arch1 using the address generated by reconfigurable AGU for 128x200 image and 3x3 kernel	64
5.36	Comparing the output image of Arch1 using reconfigurable AGU with Matlab output for 128x200 image and 3x3 kernel	65
5.37	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch2 for 192x128 image and 5x5 kernel	66
5.38	Snapshot of ModelSim waveform showing the output generated by Arch2 using the address generated by reconfigurable AGU for 192x128 image and 5x5 kernel	66

5.39	Comparing the output image of Arch2 using reconfigurable AGU with Matlab output for 192x128 image and 5x5 kernel	67
5.40	Snapshot of ModelSim waveform showing reconfigurable AGU generates correct input and output addresses for Arch2 for 128x200 image and 3x3 kernel	68
5.41	Snapshot of ModelSim waveform showing the output generated by Arch2 using the address generated by reconfigurable AGU for 128x200 image and 3x3 kernel	68
5.42	Comparing the output image of Arch2 using reconfigurable AGU with Matlab output for 128x200 image and 3x3 kernel	69

LIST OF ABBREVIATIONS

AGU	-	Address Generation Unit
ASM	-	Algorithmic State Machine
DSP	-	Digital Signal Processing
FPGA	-	Field-Programmable Gate Array
RTL-CS	-	Register Transfer Level – Control Signal
2D	-	Two Dimension

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Verilog code	75

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Nowadays, image processing has been used in a wide range of applications such as photography, medical imaging, forensics, transportation, military applications and etc. There are a number of digital signal processing (DSP) algorithms or functions available to be used in image processing, they are 2D convolution, 2D correlation, fast Fourier transform (FFT), filtering and etc. 2D correlation is one of the commonly used DSP functions in image processing. Depending on the types of kernel, 2D correlation can be used to serve different purposes in image processing such as smoothing, noise elimination and edge detection.

As image is normally stored in memory, it is necessary to read image from memory for 2D correlation processing and store the result into memory after processing as shown in Figure 1.1. For 2D correlation processing, it is required to feed the data path with the input data in certain order depending on the design of data path. Since image to be processed is stored in memory, it is necessary for data to be accessed in such order or sequence as well. In other words, addresses in such sequence are needed to supply to memory. Thus, it is necessary to calculate or generate address sequence for reading image data as well as for storing result. Some applications rely on software to calculate the address sequence which is inefficient and time consuming.

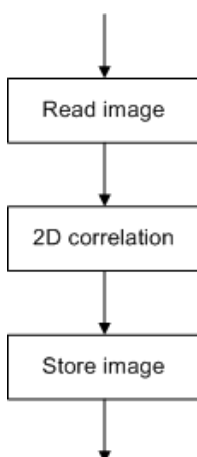


Figure 1.1: Image read and store for 2D correlation.

1.2 Problem Statement

For real time and faster application, 2D correlation is often implemented in hardware to accelerate the computation. Performance of the 2D correlation processing is not only depends on the speed of 2D correlation circuit itself, but also memory speed as well as address calculation speed. As the processing and memory speed increase, the address calculation speed becomes bottleneck for overall performance.

It is thus necessary to speed up the address calculation or generation by implementing it in hardware like FPGA rather than depending on software to calculate the addresses; such hardware is known as address generation unit (AGU). To ensure that the AGU can accelerate, the AGU should have its own datapath instead of sharing with 2D correlation for address calculation.

Prior arts of reconfigurable AGU can be reconfigured to generate address for different digital signal processing (DSP) functions including 2D correlation; however, they don't support address generation for different types of designs of 2D correlation circuit. None of the prior arts of AGU able to handle image edge condition while considering data reuse in 2D correlation circuit. Prior arts of reconfigurable AGU normally have a number of processing units each for generating address dedicatedly

for different addressing sequence resulting in inefficient design and larger circuit area. Furthermore, prior arts of AGU have never been implemented in FPGA.

Therefore, it is necessary to design and implement a reconfigurable AGU for different designs of 2D correlation in FPGA with minimal circuit area, which takes care of image edge condition while considering data reuse.

Some challenges were encountered along the design and implementation phase of the project. One of the challenges is that there is not much implementation details can be obtained from the prior arts of AGU. Address calculation for Arch2 (refer to section 4.4), which is column by column basis, is quite challenging when image border is taken into account of consideration as none of the prior arts has done it before. Another challenge is the timing consideration of internal signals in AGU. In addition, interface and timing issue between AGU and 2D correlation circuits are the most time consuming part during the design phase. The circular buffer is designed in Arch2 (refer to section 4.3) for data reuse; in order to make Arch2 reconfigurable for different kernel sizes, parameterized coding is involved. The parameterized coding is quite tedious and time consuming.

1.3 Objectives

The objective of this project is to design a reconfigurable AGU for 2D correlation in FPGA. The 4 main objectives accomplished in this project are shown below.

1. The reconfigurable AGU can generate address sequence for reading out image data needed for 2D correlation processing from a memory; and can also generate address sequence for storing the result or output image into memory.
2. The AGU can be configured to generate address sequence for different designs of 2D correlation.

3. The AGU can also be configured to generate address sequence for different image dimensions and kernel sizes.
4. The reconfigurable AGU can take care of image border while considering data reuse in 2D correlation circuit.

The configuration data, consists of the type of design of 2D correlation circuit, image dimension and kernel size, is specified by the user.

1.4 Scope of Study

The project is targeted for implementation in FPGA. The project is implemented in Verilog code using Quartus II targeting FPGA. The device involved is Cyclone III targeting FPGA. Due to time constraint, the Verilog code is not loaded into FPGA board. Thus, it is limited to simulation based model. In other words, the project is targeted to deliver the simulation model or work of reconfigurable AGU and 2D correlation. The design is simulated on Altera Modelsim. Matlab simulated result is used as a reference to verify the outcome from Modelsim.

A reconfigurable AGU which is targeted to work for two different architectures of 2D correlation is designed. The AGU is also targeted to work for different image dimensions and kernel sizes. The two different architectures of 2D correlation circuit will be designed and implemented as well. Each design of 2D correlation is reconfigurable for different image dimensions and kernel sizes.

As the reconfigurable AGU is the focus of the project, we only aim to design the skeleton of 2D correlation which can work together with the reconfigurable AGU. Thus, the 2D correlation designs are limited to averaging filter and their performances are also limited in term of speed and accuracy. The reconfigurable AGU design is verified with Arch1 and Arch2, as well as sample images and kernels with different sizes.

REFERENCES

1. Tetsuo Kawano. *Reconfigurable address generation circuit for image processing, and reconfigurable LSI comprising the same*. U.S. Patent 7, 515, 159. 2009.
2. P. Hulina, L. Coraor, L. Kurian, and E. John. Design and VLSI implementation of an address generation coprocessor. *Proc. IEE Computers and Digital Techniques*, 1995. 142(2): 145-151.
3. Ramesh M. Kini, and S. David. Comprehensive address generator for digital signal processing. *International Conference on Industrial and Information Systems (ICIIS)*. December 28-31, 2009. Sri Lanka: IEEE. 2009. 325-330.
4. Ramesh M. Kini, and S. David. ASIC implementation of address generation unit for digital signal processing kernel-processor. *ICGST-PDCS*, 2011. 11(1): 1-9.