DESIGN OF THE STREAMING PROCESSOR ARCHITECTURE FOR MICROKERNEL CONTROLLER AND ALU

NUHAIRI BIN ANUAR

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical – Computer and Microelectronic Systems)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2015

Specially dedicated to my wife, Emmy, my little boy, Umar, and my mother, Azizah.

ACKNOWLEDGEMENT

I am grateful to Allah for good health and wellbeing that were necessary in completing this project. I would like to give a special thanks to Prof. Dr. Mohamed Khalil bin Haji Mohd Hani for supervising me on this project. His input and directions since the inception greatly helped the outcome of the project. I also like to wish sincere thanks to Dr. Muhammad Nadzir Marsono, Dr. Rabia Bakhteri, and Dr. Usman Ullah Sheikh for the advised and feedback given on the project. I also would like to thank Ngo, Wai Loon for spending quality time with me discussing about the architecture of the Stream Processor. Last, but not least, I would like to thank my wife, Ermi Lihan for the support and understanding during the completion of the project.

ABSTRACT

Media application such as 3D graphic processing, image processing, video decode and encode requires high rates of arithmetic operation per second. As an outcome of a decade long research, Stream Processor architecture, which is designed to exploit the characteristic of media processing was proposed. A few architectures of stream processor had been proposed and among the popular streaming processor architecture was Imagine Stream Processor. This project is focusing on implementing Imagine-based stream processor architecture on Altera Cyclone IV GX FPGA specifically for ALU Clusters and Microkernel Controller modules. This project report presents the literature reviews of books, theses and papers regarding stream processor architecture and its related use cases. This report also documents the complete project methodology taken in order to design a stream processor on the FPGA where the design of the Stream Processor is using RTL design methodology and System Verilog Language. This report also detailed the architectural design of the stream processor in Chapter 3. Furthermore, result and discussion of the experimental work involve in this project also reported in Chapter 4. The project report concluded that an FPGA-based stream processor is successfully designed and tested with specific image processing use cases. This project report also described possible enhancements possible on the stream processor design.

ABSTRAK

Aplikasi media seperti pemprosesan grafik tiga dimensi, pemprosesan imej, penyahkodan dan pengkodan video memerlukan kadar operasi aritmetik persaat yang tinggi. Hasil daripada penyelidikan lebih satu dekad, Pemproses Aliran yang direka untuk mengekploitasi ciri-ciri pemprosesan media, telah dicadangkan. Antara beberapa reka bentuk Pemproses Aliran sudah dicadangkan, reka bentuk yang agak dikenali adalah Pemproses Aliran Imagine. Projek ini lebih fokus kepada mencipta Pemproses Aliran berasaskan seni reka Imagine di dalam Altera Cyclone IV GX FPGA terutamanya modul Kluster ALU dan Pengawal Mikrokernel. Laporan projek ini mempersembahkan kajian penulisan tentang buku-buku, tesis-tesis, dan kertas kerja berkenaan seni reka Pemproses Aliran dan pengunaannya. Laporan ini juga melaporkan methodologi penuh yang digunakan untuk mereka satu reka bentuk Pemproses Aliran di dalam satu FPGA, dimana rekaan RTL dan Bahasa System Verilog telah digunakan. Laporan ini juga menjelaskan reka bentuk Pemproses Aliran di dalam Bab ke-3. Juga, hasil dan perbincangan tentang kerja eksperimen yang terlibat dalam project ini dilaporkan dalam Bab ke-4. Laporan project ini membuat kesimpulan bahawa sebuah Pemprosesan Aliran berasaskan FPGA adalah berjaya direka and diuji dengan kes-kes ujian pemprosesan imej yang spesifik. Laporan projek ini juga menerangkan pembaikan yang boleh dilakukan keatas Pemproses Aliran tersebut.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

ALU	-	Arithmetic Logic Unit
FPGA	-	Field-Programmable Gate Array
GPP	-	General Purpose Processor
ISA	-	Instruction Set Architecture
RTL	-	Register Transfer Level
SIMD	-	Single Instruction Multiple Data
VLIW	-	Very Long Instructions Words

LIST OF SYMBOLS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background and Project Rationale

Stream Processor is a type of *programmable processor* which exploit the *compute intensity*, *parallelism* and *producer-consumer locality* in order to process image and media efficiently (Khailany, 2003).

Compute intensity is defined as the number of arithmetic operation per global memory reference or I/O (Khailany et al., 2008) where media processing application such *as image processing*, *video compression and three-dimensional graphics* requires high rates of arithmetic compared to conventional computing application. Streaming programmable media processor which is designed to meet these demands usually contains tens to thousands arithmetic unit. This increases the challenge of providing the sufficient data bandwidth (Rixner, 2001).

On the other hand, *data parallelism* exists in stream processing because the same function is applied to many records of a stream and each record can be processed simultaneously without dependency from other records (Khailany et al., 2008). It is very common for media processing to be greatly enhance with parallelism. Parallelism in media processing can be categorized into three, which are instruction-level parallelism (ILP), data-level parallelism (DLP), and task-level Parallelism (TLP) (Khailany, 2003).

Producer-consumer locality is defines as specific temporal locality, where data is produced once, read once or twice later, and never read again (Khailany et al., 2008). In media processing, locality of reference for data accesses is another important

characteristic. The locality in media processing can be classified into kernel locality and producer-consumer locality (Khailany, 2003).

Other than equipped with large number of arithmetic logics, streaming processor architecture also equipped with a data bandwidth hierarchy such as the three tiered data bandwidth hierarchy in Imagine Processor, which enabled the processor to efficiently provide data bandwidth continuity for tens to hundreds of arithmetic units (Rixner, 2001).

Even though a conventional load/store CPU or more commonly known as general purpose CPU is also capable of processing media applications, a dedicated stream media processor promise greater performance given same amount of resource in term of area, gate count, and power since stream processor is design specifically to exploit the three characteristic mentioned previously.

Since streaming processor architecture is significantly important in electronic world and still actively being researched, I proposed this project to gain knowledge of inner working of a streaming processor architecture and to find out how to improve current designs with a focus on the VLIW Controller Unit and the Streaming Arithmetic clusters designs.

1.2 Project Problem Statement

Recently, digital vision and imaging system have progressed greatly, but most of the algorithms such as 3D graphic, video compression, video decode and 2D image processing still requires high rates of arithmetic in the range of hundreds of billions operands per second (Rixner, 2001) making the integration with embedded system which are sensitive to power dissipation, physical size and cost very difficult. Historically, this application have been met with ASICs or ASSPS, but these solutions often lacking in flexibility. This project will propose a FPGA-based stream processor for vision and imaging purpose.

When processed on so-called conventional load-store CPU, most media processing algorithm take extremely large amount of CPU cycles and hence execution time which hindered real time media application on a digital system (Khailany et al. 2008). Due to the nature of conventional CPU memory architecture, media processing

also require large amount of memory bandwidth and suffer higher rates if cache miss since the cache are not optimized to take advantage of producer-consumer locality of image processing (Khailany, 2003). The execution of multiple stream operators aggravates the processing unit-external memory bandwidth bottlenecks (Alves et al., 2011).

While fixed-function processors such as the one based on ASICs have been able to match the high performance demanded by media processing algorithm, programmability proved to be a key requirement in many digital systems which processed complex and evolving algorithm (Khailany et al., 2008). Programmable solutions also inherently have cost advantage over fixed function processor since a single programmable chip can be used in many different systems. As image resolutions, scene complexity and algorithmic complexity continue to rise, the demand for real time performance will continue to increase.

1.3 Project Objectives

The objective of the project is to develop a programmable FPGA-based Stream Processor ALU and its Controller Unit for the purpose of common image processing algorithms.

The stream processor that is to be develop for the project will act as co-processor which will offload specific image processing operation from a general purpose host CPU. The intended stream processor is expected to work in a similar manner as a Graphical Processing Unit (GPU) of a conventional PC, where common simple operation will be done by the GPP and the intended circuit will only process certain image processing algorithm.

The stream processor ALU will be developed with a set of internal buffers registers to improve stream memory efficiency and to avoid memory operation bottlenecks by eliminating register-to-memory operations for intracluster data movement.

1.4 Project Scope

The circuit module to be designed in this project will be part of a larger high speed FPGA-based embedded visual recognition system. This project will focus on Stream ALU (ALU Clusters) design and ALU Controller Unit. Stream memory components, except half buffer within the ALU Cluster, are out of scope of this project. Following diagram shows the scope of this project within the dashed red box.

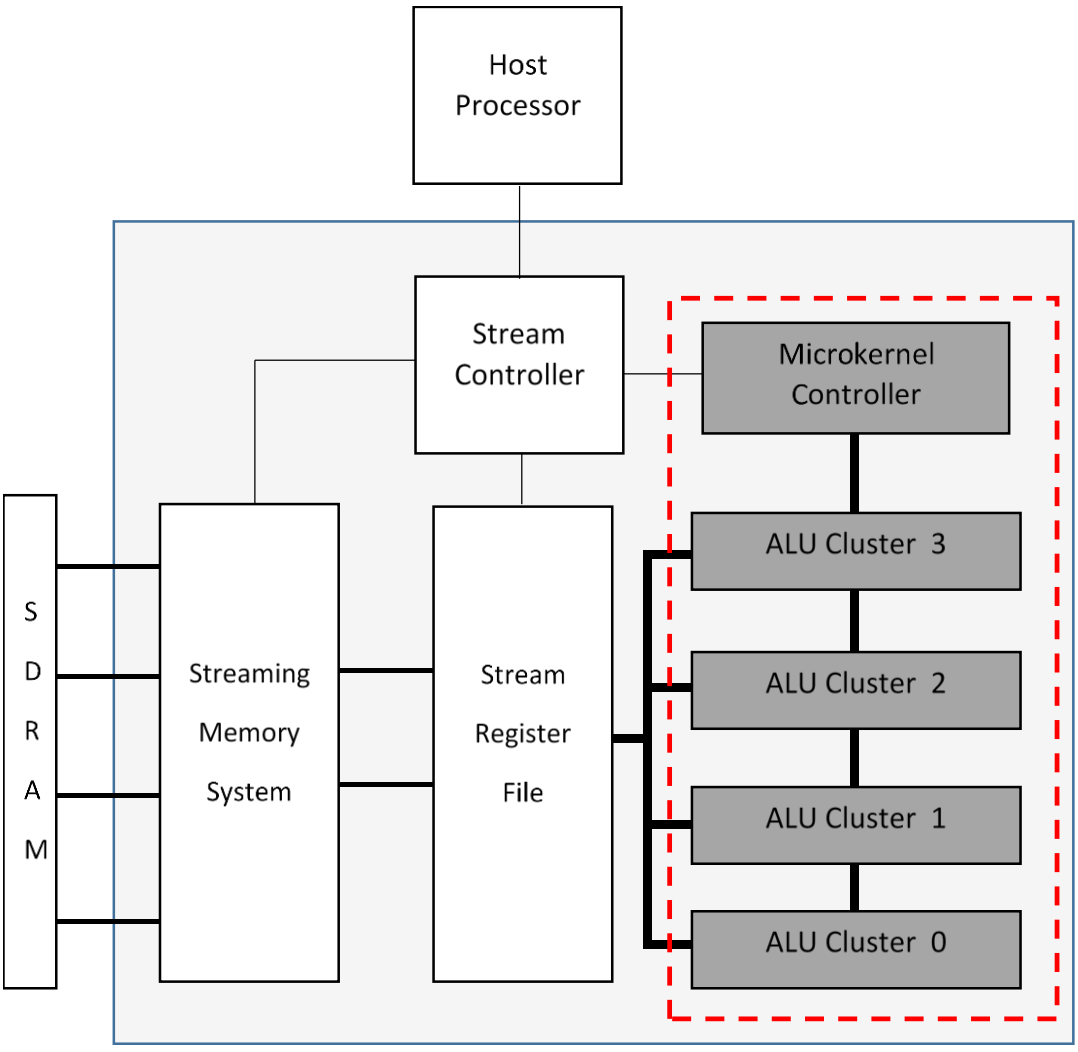

Figure 1.1: Top level diagram of the stream processor. The scope of this project is highlighted inside the red dashed box.

The expected contribution of this project is a simulation model of the hardware ALU computation core and a micro kernel controller unit on Modelsim with a set of

testbenches replicating the input of ALU and its controller. Items to be designed as part of the ALU Clusters consist of:

1. A set of functional units (FUs) for addition/subtract, multiply, divide, and logical shift left and right. It also support immediate operands and three input functional units.
2. ALU internal buffers or also known as Local Registers Buffers (Half Buffers) for all inputs of each functional units.
3. Interconnect between internal local buffers and FU outputs. This reduces memory bandwidth.

On the other hand, items to be designed as part of the Microkernel Controller consist of:

1. Very long instruction word (VLIW) control engines.
2. Microkernel storage to store the 'Kernels' (instructions).
3. Instruction fetch mechanism.

Signals from Stream Controller and data from SRF will be replicated within testbenches and generated with the help of Matlab code. The design for the stream processor is based on simplified Imagine Stream Processor.

1.5 Project Achievement

This project successfully modelled Stanford's Imagine Processor, which was a stream processor designed for custom VLSI, on a simulation with Altera Cyclone IV as the intended target. Imagine Stream processor is the state of the art stream processor that is designed for media processing application and still being actively researched and enhanced.

The design created in this project fulfil the resource limitation within Altera Cyclone IV including numbers of IO, number of LABs and dedicated memories. The design of the stream processor were successfully simplified while stream processing flow were maintained.

From reviewing the existing literature on stream processor, it is found that this project could be the first project with complete documentation on the design and methodology which simulated the Imagine Processor using System Verilog on a FPGA.

The result of the image processing kernel from the stream processor were compared to the output of similar algorithm in Matlab and it is proven to be correct. Among the kernels tested are RGB to YUV conversion, image difference/subtract image inversion and power law transformation.

REFERENCES

Alves, J. and Diniz, P. (2011). Custom FPGA-based micro-architecture for streaming computing. *2011 VII Southern Conference on Programmable Logic (SPL)*.

Dessai, S. and Vutukuru, K. (2012). Design and Development of Stream Processor Architecture for GPU Application Using Reconfigurable Computing. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, [online] 2(1). Available at: http://iaesjournal.com/online/index.php/IJRES/article/view/1309 [Accessed 17 Dec. 2014].

Farabet, C., Poulet, C. and LeCun, Y. (2009). An FPGA-based stream processor for embedded real-time vision with Convolutional Networks. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*.

Gonzalez, R. and Woods, R. (2007). Digital image processing. Reading, Massachusetts: Addison-Wesley Publishing Company.

Kapasi, U., Dally, W., Rixner, S., Mattson, P., Owens, J. and Khailany, B. (2000). Efficient conditional operations for data-parallel architectures. *Proceedings 33rd Annual IEEE/ACM International Symposium on Microarchitecture. MICRO-33 2000*.

Kapasi, U., Dally, W., Rixner, S., Owens, J. and Khailany, B. (2002). The Imagine Stream Processor. *Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors*.

Khailany, B. (2003). *The VLSI Implementation and Evaluation of Area- And Energy-Efficient Streaming Media Processors*. Ph.D. Stanford University.

Khailany, B., Williams, T., Lin, J., Long, E., Rygh, M., Tovey, D. and Dally, W.

(2008). A Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing. *IEEE J. Solid-State Circuits,* 43(1), pp.202-213.

Kyrkou, C. (2014). Stream Processors and GPUs: Architectures for High Performance Computing. [online] Available at: http://sokryk.tripod.com/Stream_Processors_and_GPUs_-_Architectures_for_High_Performance_Computing.pdf [Accessed 17 Dec. 2014].

Msdn.microsoft.com, (2015). Converting Between YUV and RGB (Windows CE .NET 4.2). [online] Available at: https://msdn.microsoft.com/en-us/library/ms893078.aspx [Accessed 23 May 2015].

Rixner, S. (2001). *Stream Processor Architecture*. Boston, MA: Kluwer Academic Publishers.