

LOW COST LOOSELY COUPLED PARALLEL PROCESSING SYSTEM USING
STANDARD PROCESSOR

ZAHARI AWANG AHMAD

UNIVERSITI TEKNOLOGI MALAYSIA

ACKNOWLEDGEMENT

It is my pleasure to acknowledge my supervisor for this project Dr Muhammad Nasir Ibrahim who precisely guided and facilitated well in the process of research, understanding and anticipating the subject of interest throughout the project. I am grateful to the comments and suggestions given by Dr Muhammad Nasir Ibrahim during the closed project presentation. I also would like to acknowledge Dr Mariani Idroas who assisted in this thesis reporting. Also, I would like express my appreciation on her previous works which has inspired me in proceeding with this project.

I would like also to acknowledge Dr Shaikh Nasir Shaikh Husin and Dr Muhammad Nadzir Marsono who are being fabulous project presentation panel assessors. Not to forget friends from Intel Penang/Kulim, Altera, Jabil and other institutions for being helpful college mates along the course in the UTM.

My studies in pursuing Master Degree in Computer and Micro-electronics were conducted properly in Intel U, PG9. Special thanks to Intel Penang management especially the management team in Intel U, PG9.

Lastly but not least to my wife Suraya Ishak and kids, Zarif Afnan, Syamimimi Amni and Syazia Aina who are being very supportive and understanding throughout the period of my studies in UTM.

ABSTRACT

The experiment is to use low cost processors and integrate them to work in parallel. Along the way to achieve the objective, many fields of knowledge need to be studied. As in this project, knowledge of matrix operation and how it can be broken down for the purpose of parallel processing are studied. Special task is identified to focus in this project that is matrix operation for image processing. Eventually, the ultimate objective is to see how much speed of processing that can be improved when parallel processing is implemented. The speedup is the keywords. There are three main parts in this development. Firstly, the parallel processing system requires a hardware integration whereby two ARM processors are integrated. Secondly, the system requires firmware to run on both processors. The firmware is to develop based on C ARM compiler. This firmware will handle the image processing, data transfer between processors and data transfers to Laptop PC. Thirdly, the whole system requires a program on the laptop PC to display data on LCD monitor and capture the processing time. Upon all the system in operation, processing time of single processor and dual parallel processor can be compared and analyzed. A significant improvement in term of speedup is observed and there seems to be many ways to further develop the system to benefit wide range of applications.

ABSTRAK

Analisa ini menggunakan unit pemprosesan kos rendah dan pemproses diintegrasikan untuk membuat kerja-kerja pemprosesan secara selari. Untuk mencapai objektif tersebut, banyak bidang pengetahuan harus didalami. Dalam projek ini, pengetahuan berkenaan dengan operasi matrik dan bagaimana ia boleh dipecahkan untuk pemprosesan selari diuji. Operasi matrik yang diketengahkan adalah operasi untuk pemprosesan imej. Objektif utama ujikaji ini adalah untuk melihat sejauh mana kelajuan pemprosesan meningkat apabila konsep selari digunakan. Untuk membangunkan sistem ini, terdapat tiga peringkat utama harus dijalankan. Pertama, sistem pemprosesan selari memerlukan integrasi perkakasan dimana dua pemproses ARM diintegrasikan. Kedua, sistem memerlukan pengaturcaraan *firmware* untuk operasi serentak kedua-dua pemproses. Aturcara tersebut dibangunkan menggunakan *C ARM compiler*. Hasilnya digunakan untuk memproses imej dan juga penghantaran data pergi dan balik antara kedua-dua pemproses ARM dan juga antara pemproses ARM dan komputer riba. Ketiga, satu program diperlukan untuk komputer riba bagi memaparkan data di LCD dan merekod masa pemprosesan. Dengan itu masa pemprosesan oleh sistem pemproses tunggal dan sistem selari dengan dua pemproses boleh di banding dan dianalisa. Hasil dari projek ini agak baik dan nampaknya banyak lagi ruang terbuka untuk menaiktaraf sistem ini seterusnya supaya dapat membantu pelbagai aplikasi pemprosesan.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	III
	ACKNOWLEDGEMENT	IV
	ABSTRACT	V
	ABSTRAK	VI
	TABLE OF CONTENTS	VII
	TABLE OF FIGURES	IX
	LIST OF APPENDICES	XI
	ACRONYMS	XII
1	INTRODUCTION	1
	1.1 Parallel Processing Overview	1
	1.2 Parallel Processing Architecture	2
	1.3 Terminology Of Parallelism	3
	1.4 Parallel Processing Trend In Computing	5
2	LITERATURE REVIEW	7
	2.1 Research Objective	7
	2.2 Scope Of Research	8
	2.3 Important Of Research	9
	2.4 Problem Statement	10
	2.5 Approach	11
	2.6 Methodology	12
	2.7 Implementation Plan	17

3	LOOSELY COUPLED PARALLEL PROCESSING	
	SYSTEM	18
	3.1 Introduction To Loosely Coupled Parallel Processing System	18
	3.2 Advantages And Disadvantages Of The System	19
	3.3 Area Of Application	20
4	PARALLEL MATRIX OPERATION AND GRAY SCALE MORPHOLOGY	23
	4.1 Overview	23
	4.2 Partition Matrices For Parallel Processing.	23
	4.3 Other Matrix Operation	26
	4.3 Gray Scale Morphology	27
5	RESULT ANALYSIS	31
	5.1 Overview Of The Developed Demonstrator System	31
	5.2 System Hardware.	31
	5.3 System Firmware	35
	5.4 Result Summary	46
6	CONCLUSION AND FURTHER WORKS	49
	6.1 Conclusion	49
	REFERENCES	51
	APPENDIX A	53
	APPENDIX B	70
	APPENDIX C1	81
	APPENDIX C2	90

CHAPTER 1

INTRODUCTION

1.1 Parallel Processing Overview

Parallel processing has been a phenomenon in these days. The race against speed of processing in computing has been intensely studied by many researchers. The parallel processing is the use of multiple processors to execute different parts of the same program simultaneously [4]. The expected result of this implementation is to reduce the wall-clock time. The processors may operate at the speed they used to but they accomplish tasks faster than it used to. Traditionally, the approaches to get tasks accomplished faster have been generalized on the design of single processors. There are efforts to make a single processor design larger by increasing memory size so that addressing can be directly and faster. The other efforts are to make the processor more powerful by increasing basic word length and computational precision. The most popular approach is making the processor operate in high speed using micron-etching technology processor, putting more transistors in a chip and couple the processor with larger and faster communication pathways [4]. However those efforts are now reaching the limit. The smallest possible size of transistors has been produced; processor operating clock frequency is now about at the ceiling and cost of making a higher speed processor gone ridiculously high. The best alternative at the present time is to make use the existing matured processors which operate at a fairly fast clock frequency and design them to a suitable computing parallelism. The cost will be optimized and speed achievement is possible. The trend is obviously

observed by looking at the market players of processor maker like Intel and AMD where they are designing their processor to have multiple cores inside.

1.2 Parallel Processing Architecture

Computer architecture can be classified into different type according to its modes of execution. Based on the Flynn taxonomy classification of computer architecture as shown in figure 1-1, the two dimensions of computer architecture that are Instruction and Data, can take either Single or Multiple values [4]. Therefore, as per the realistic in computing world, there are basically three basic types of computer system. The three types of computer system are Single Instruction Single data (SISD), Single Instruction and Multiple Data (SIMD) and Multiple Instruction and Multiple Data(MIMD).

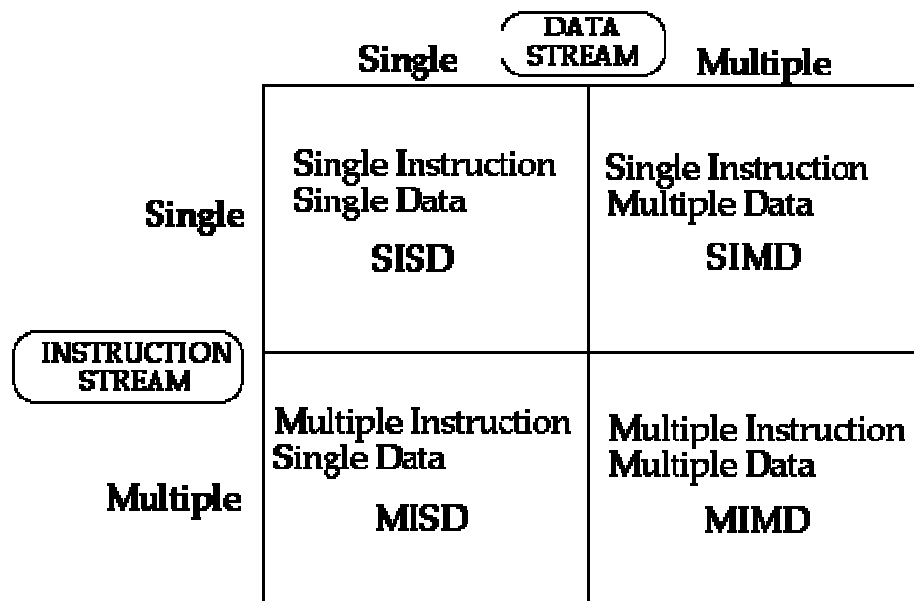


Figure 1-1 Flynn taxonomy classification of computer architecture

Parallelism can be exploited in Single Instruction Single data (SISD), Single Instruction and Multiple Data (SIMD) and Multiple Instruction and Multiple Data(MIMD) modes of execution based on two principles. Firstly, parallel

implementation by overlapping operation in time-temporal parallelism. Secondly, parallel implementation by replicating resources in space which is called spatial parallelism.

There are four major types of parallel architecture. Firstly, the pipelining that interleaves the successive steps of executing an instruction or operation across multiple stages of pipeline unit [2]. This way of executing the instruction can then be overlapped when the pipeline is filled up. Conceptually, this implementation uses temporal parallelism. Secondly, processor array where multiple processors are connected together in an array [2]. Generalizing the pipeline mode into an algorithm, then the algorithm can be broken into several steps where each step can be assign to a processor in the array. This model of computation also called algorithmic parallelism. One example of such computational model is systolic array. In systolic array, the algorithm is decomposed to identical steps and mapped onto identical processing units. Thirdly, array processor architecture where multiple processors are connected together usually in a two-dimensional array and controlled by a single control unit [2]. The final architecture is multi-processor or multi-computer architecture [2]. Multi-processor architecture is formed by connecting multiple processors together for sharing a common memory. On the other hand, multi-computer is formed by connecting multi-computer together but not sharing the memory. The computers are interconnected via a high speed communication network. Each computer remains a full function computer in the system where they control their own CPU, memory and I/O's including the communication.

1.3 Terminology of Parallelism

Parallel processing has its own lexicon of terms and phrases. They are used to address and emphasizing the concepts that are considered to be most important to its goals and the ways in which those goals may be achieved [4]. The following are some of the more commonly encountered ones:

Task – a logically discrete section of computational work. A simple example is a task of calculating average of 10 numbers. In computational, there are basically two distinct tasks which are the sum of all 10 numbers is calculated first followed by the division of the sum by the 10.

Parallel tasks – is the tasks that are independent each other in their computations. Therefore, the tasks can be performed concurrently without introducing any errors.

Serial Execution – referred as execution of a program tasks sequentially. In other word, execution of one statement at one time.

Parallelizable Problem – referred to a problem that can be broken into parallel tasks which may require changes in the code and/or the underlying algorithm.

A simple example of *parallelizable problem* described in the following:

Calculate the potential energy for each of several thousand independent conformations of a molecule; when done, find the minimum energy conformation. Each of the conformations is independently determinable;

therefore it can be done concurrently. Meanwhile, the calculation of the minimum such conformation is itself a *parallelizable problem*.

On the other hand, the *non-parallelizable problem* example described as follows:

In the calculation of the Fibonacci series (1,1,2,3,5,8,13,21,...), the following formula is used:

$$F(k + 2) = F(k + 1) + F(k)$$

In this calculation of the Fibonacci sequence above, it is clearly that the result of the formula is absolutely dependent. Notice how calculation of the $k + 2$ value uses those of both $k + 1$ and k , hence those three terms cannot be calculated independently, nor, therefore, in parallel.

1.3 Parallel Processing Trend in Computing

In research paper of Oliver McBryan of Colorado University at Boulder, he states that in 1990 general consensus was developing that highly parallel computers are the only practical near-term route to the teraflops of computing power required by many scientific and engineering problems [7]. At that time, conventional supercomputers utilize low parallelism while highly parallel computers are still in the experimental stage. Processor manufacturers were battling in making a single processor that operates at clock speed of greater than 3 GHz in which the battle has become too complex. As a result, they turn to look for parallel processing computation in order to continue to improve the performance. Obviously, at present moment, that general consensus has become a factual. In 2005, dual core processor was introduced by both Intel and AMD. Later, Intel has introduced quad core processor and kept improving along the way. This multi-core processor contains multiple execution cores in the same chip, each of which can independently perform operations, thereby introducing new level of parallelism to desktop processors [6].

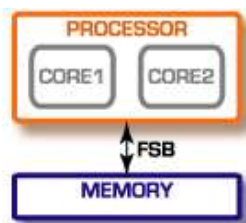


Figure 2-1 Representative diagram memory shared between both of the cores within the processor of a dual-core system.

Other type of parallel computing is the multiprocessor computer which simply means that multi processors are attached to the same motherboard. It has been classified that there are two type of multiprocessor system. The first one is Symmetric Multiprocessor (SMP) and the second one is Non-Uniform Memory

Access (NUMA). These systems share a single memory controller/system bus, and have performance characteristics similar to multi-core systems [6].

The other trend in parallel computing is the cluster. It composed of many PCs, workstation or servers connected together on the network. The processing units (PCs, workstation or servers) are called nodes in the cluster. Each node is a complete PC with its own processor(s) and memory. The cluster employs parallelism in memory access and the computation capability. The disadvantage of the clusters is that communication efficiency is reduced when large computation is exercised. This is of course due to communication is based on the network layer [6].

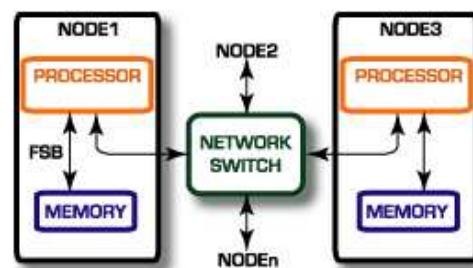


Figure 3-1 A representative clusters with scalable memory and computation capabilities.

However, in contrast to the multiprocessor system, the number of total nodes is not limited by hardware, and therefore the cluster provides a system that can scale to very large amounts of memory capacity while retaining high memory bandwidth. This project focuses the methodology that very much similar to the cluster. Instead of using PCs, ARM processors will be used as the nodes. Like a PC system, ARM processor is packaged with processing unit and memory resources in a single chip. The different will be the interconnection between nodes whereby it will use common communication I/Os like RS232, I2C or SPI.

REFERENCES

1. Matloff, Norman. (2008). *Introduction to Parallel Matrix Operation*. University California at Davis. May 20, 2008.
2. Jin, Lin. (2004). *Parallel Processing: Exploring the architectures' and algorithms' close relation*. IEEE. December 94/January 95.
3. R. Horan and M Lavelle. (2005). *Matrix Multiplication*. University of Plymouth. November 2, 2005.
4. Elmohamed, Saleh (2002) *Parallel Processing Concepts* Cornell University. October 25, 2002.
5. Novak Shannon et al. *Performance evaluation for a loosely coupled parallel Processing Environment*. Tulane University.
6. *Technology Trends: Parallel Processing and FDTD Solutions*. Retrieved on Jan 18, 2009, from http://www.lumerical.com/parallel_processing_ftdt_solution.php
7. McBryan, Oliver A. (1990) *Trends in Parallel Computing* Colorado University at Boulder. December 1990.
8. INEXGlobal. (2006). *JX-2148: LPC2148 ARM7-32 bit Microcontroller Education board*. [Manual]. Thailand: INEXGlobal.
9. Dietz, Hank. (1998). *Linux Parallel Processing HOWTO*. Purdue University. January 5, 1998.
10. P. Colella, (2004). *Defining Software Requirements for Scientific Computing*. [presentation]
11. Krste Asanovic et al. (2006). *The Landscape of Parallel Computing Research: A View from Berkeley*. University of California at Berkeley. December 18, 2006

12. *NAS Benchmarks*. Retrieved on Jan18, 2009, from
http://en.wikipedia.org/wiki/NAS_benchmarks
13. D Bailey et al. (1994). *RNR Technical Report*. RNR-94-007. March 1994.