ENHANCED IMPLEMENTATION OF EMBEDDED CONCURRENT PROCESSOR USING NIOSII AND SHARED MEMORY ON FPGA FOR BETTER WORKLOADBALANCE

By

MAYS QAHTAN SEDEEQ
(1532321573)

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science (Embedded System Design Engineering)

School of Computer and Communication Engineering
UNIVERSITI MALAYSIA PERLIS
2016

ACKNOWLEDGMENT

At the beginning and in the name of ALLAH the most greatest and merciful, where this work would never been achieved without the blessing, health and knowledge granted by ALLAH the almighty.

I would like to express my gratitude profoundly from my heart for the fruitful assistance, guidance and encouragement of my supervisor Dr. Muataz Hameed Salih Al Doori, whom support, trust and valuable advice was continuous till the end of this work.

I gratefully acknowledge the continuous oversight provided by Professor Dr. R. Badlishah Ahmad (Dean of the School) where the granted facilities to all students equally, made overcoming difficulties easier and made this short journey as one of the most exiting experience of my life.

I cannot but extend my grateful thanks to University Malaysia Perlis and the School of Computer and Communication Engineering presented by its outstanding lecturers whom brotherly guidance and share of knowledge were the bases of whatever noticed success.

Finally I would thank my mother, father and husband whom were always beside me helping me and enduring all my breakdowns. I dedicate this work to my children whom I apologise for any kind of negligence through this time.

TABLE OF CONTENTS

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULTS AND DISCUSSION

CHAPTER 5 CONCLUSION AND FUTURE WORK

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

ALU	Arithmetic Logic Unit
ASIC	Application Specific Integrated Circuit
CAD	Computer-Aided Design
CPU	Central Processing Unit
DSP	Digital Signal Processing
EDA	Electric Design Automation
EECP	Enhanced Embedded Concurrent Processor
FIFO	First-In-First-Out
FPFA	Field Programmable Gate Array
GPU	Graphic Processing Unit
HPRC	High Performance Reconfigurable Computing
IP	Intellectual Property
LAB	Logic Array Block
LUT	Look Up Table
MIMD	Multiple Instruction stream Multiple Data stream
MISD	Multiple Instruction stream Single Data stream
MPSoC	Multi-Processor System on Chip
NCD	Native Data Description
NoC	Network on Chip
NUMA	Non Uniform Memory Access
PLL	Phase-Locked Loop
RTL	Register Transfer Level
SBC	Single Board Computer

SBCM	Single Board Computer Multiprocessor
SBT	Scala Built Tool
SIMD	Single Instruction stream Multiple Data stream
SISD	Single Instruction stream Single Data stream
SMP	Symmetric Multi-Processor
SoC	System on Chip
SOPC	System On Programmable Chip
TI	Texas Instrument
UART	Universal Asynchronous Receiver/Transmitter
UEECP	Unbalanced Enhanced Embedded Concurrent Processor
UMA	Uniform Memory Access
VGA	Video Graphic Array
VHDL	VHSIC High Description Language
VHSIC	Very High Scaled Integrated Circuits
VLIW	Very Long Instruction Word
VLSI	Very Large Scaled Integration

Peningkatan Pelaksanaan Pemproses Sewaktu Tertanam Menggunakan NiosII Dan Pengkongsian Ingatanpada Field Programmable Gate Array (FPGA) Untuk Mengimbangkan Beban Kerja Yang Lebih Baik

ABSTRAK

Kewujudan aplikasi moden yang mencabar dan memerlukan kuasa pengiraan yang tinggi telah mendorong pencipta untuk mencipta satu system yang sesuai bagi mengendalikan cabaran-cabaran yang membangkit. MPSoC muncul sebagai satu penyelesaian yang cerah untuk membekalkan prestasi masa sebenar disamping menangani "parameter" lain yang kritikal seperti kos, penggunaan tenaga dan kegunaan dikawasancip. Reka bentuk multipemproses yang berkongsi memori mengalami beberapa isu. Menyelaras akses pemproses berhubung dengan memori yang dikongsi adalah masalah pertama yang dihadapi bersama-sama dengan kemungkinan mempunyai keselamatan 'thread' yang boleh menyebabkan kesilapan pengiraan.Kemungkinan mempunyai 'deadlocks' adalah masalah kedua dan ianya ditangani dengan mereka bentuk sistem hierarki dengan kecekapan yang tinggi dalam selok-belok pengendalian.Sebagai pemproses berhubung mula menjalankan aplikasi tertentu, isu sama ada mempunyai pemproses terbiar atau muatan boleh mengurangkan prestasi keseluruhan memohon sedemikian beban mengimbangi algoritma merupakan satu langkah penting untuk mencapai peningkatan yang diminta.Lembaga yang dipilih dalam reka bentuk yang dicadangkan adalah Nios II terbenam Kit penilaian (NEEK) taufan III edisi di mana hasil yang akan dipaparkan di papan multimedia LCD yang dilampirkan dan juga pada VGA port. Rekabentuk ini telah mengharungidua fasa, dimana fasa pertama adalah untuk meningkatkan pelaksanaan Pemproses Sewaktu Tertanam menggunakan pemproses NiosII dan perkongsian memori. Fasa kedua adalah mengesahkan rekabentuk melalui pelaksanaan Mandelbrot- set sebagai aplikasi serta penggunaan algorithm pengimbang beban.Rekabentuk yang dicadangkan dibina menggunakan tiga pemproses NiosII yang dilampirkan melalui Qsys dalam bentuk 'hierarchical master-slave' pada satu cip.Pemampan cepat serta kod ruang margin berasingan telah digunakan untuk melengkapkan penambaikan prestasi yang dicadangkan rekabentuk.Keputusan yang diperolehi menunjukkan peningkatan dalam bilangan paparan dalam jangka masa sesaat menyebabkan kelajuan yang tinggi sekurang-kurangya 20 kali ganda berbanding dengan kelajuan multipemproses yang tidak seimbang.Peningkatan dalam kuasa komputer yang berkembang dengan sistem berkenaan perlu dipertimbangkan untuk memenuhi permintaan yang tinggi yang menuntut aplikasi tersebut pada masa hadapan.

Enhanced Implementation of Embedded Concurrent Processor using NiosII and Shared Memory on FPGA for Better Workload Balance

ABSTRACT

The existence of modern demanding applications and their need for high computational power motivate designers to look for an adaptive system that would handle the uprising challenges. MPSoC merges as a very promising solution that would provide real time performance and also addresses other critical parameters such as cost, power consumption and the utilisation of on chip area. The design of a multiprocessor that shares memory suffers from some issues. Synchronising the access of the connected processors to the shared memory is the first problem faced along with the possibility of having thread safety that may cause computing faults. The possibility of having deadlocks is the second problem and it was addressed by designing a hierarchical system with high efficiency in handling interrupts. As the connected processors start running a given application, the issue of having either an idle processor or overloaded one could decrease the overall performance, thus applying the load balancing algorithm is a significant step to achieve the demanded enhancement. Field Programmable Gate Array (FPGA) was found very efficient in constructing and enhancing the implementation of embedded concurrent processor. The selected board in the proposed design was NiosII Embedded Evaluation Kit (NEEK) CycloneIII edition where the result was to be displayed on the attached LCD multimedia board and on VGA port as well. The design went through two phases, first phase is enhancing the implementation of embedded concurrent processor utilising NiosII and shared memory. Second is validating the design through implementing the Mandelbrot-set as an application and applying the load balancing algorithm. The proposed design is constructed via three NiosII processors connected through the Qsys in a hierarchal master slave style all on one chip. Fast buffers and separate code margins were used to complete the enhancements on the performance of the proposed design. The obtained result showed an increase in the number of displayed frames in one second making speed enhancement up to 20 times the speed of unbalanced multiprocessor. The expanding of this type of system to gain more computing power must be considered to satisfy the uprising demanding applications in the future.

CHAPTER 1

INTRODUCTION

1.1 Overview

There are very important aspects that should be covered when dealing with whether designing or implementing a system. This is how to choose the best equipment, overcome limitations and make use of the latest provided solutions to achieve the demanded goals. The need for embedded systems came to surface when general-purpose computers are considered too costly for the most of products that integrate some kind of embedded system technology. The other cause is that general-purpose solution may also fail to deliver a number of functional or performance demands. Generally, there are two tight restrains in embedded computers which are, functionality and implementation. Particularly, insurance must be given to very important demands such as the reactive relation between real time operation and external events, supporting the limits in size and weight, providing a cost effective power and cooling consumption, meeting safety and reliability necessities and satisfying tight cost objectives.

In this aspect MPSoC (Multi-Processors System on Chip) is considered a solution to be utilised in a wide range of recent high performance embedded systems. The reason for these considerations is that MPSoC satisfies the computational demands of smart real time applications spread across multiple domains. The MPSoCs are recently engaged in a wide variety of products for their ability to provide beneficial balance between energy efficiency and performance (Kwon, Kim, Jeun, Ha, & Paek, 2008).

Parallelism is presented by multi-processor computer architectures, as the future of computing. In this thesis the suggested design is capable of performing different tasks and functions on different processors in real time embedded applications with real parallelism. In the proposed design the achieved enhancement in memory coherency was accomplished through reducing the hazard. Hazards prevent the execution of the next instruction in the following clock cycle reducing the performance. It is important to point out that the enhancements on the performance of the proposed system was achieved through implementing two specifications, first handling the interrupts separate exception stacks were only used and secondly fast and separate section of code where implemented. The proposed system handed also two enhancements in concern to the memory mechanism, which was first fixed low-latency read access to executable code and second was the fixed low-latency read, write, or read and write access to data. These enhancements made great improvement in the execution time which was obvious while implementing the chosen application.

Mandelbrot-set (M-set) application is selected to validate the proposed design. The M-set plot shows the iterations that a point takes in the complex plane. The repeated process in this application and the displaying of each constructed M-set with all data needed to be dealt with, is a challenge to any computing system. The proposed enhanced embedded concurrent processor with shared memory on FPGA handed great improvement in number of frames displayed with a speed improvement that exceeded 20 times that in unbalanced multiprocessor. The results are shown on two displaying systems which are the attached LCD multimedia board and on Video Graphic Array (VGA) port as well.

What was taken in to consideration in the proposed design was how to minimise the relative high complexity of the end design, which was achieved through presenting

the design steps starting from the lowest stages heading up to the end design. This presentation was aiming to propose the design tasks in a very simple manner rendering the system as much as possible with real time respond. In the attempt to avoid dead locks, the numbers of processes were more than the number of processors. To prevent the shared memory from being corrupted by the action of another processor, the software was written so that each processor claims temporarily ownership of the memory through the mutex core ensuring mutually exclusive access.

At the present, FPGAs opened new horizons not just for implementing prototypes, but for the benefit of the final designs as well. The development of FPGA capacity permits designers to implement an entire multiprocessor system on one FPGA. The FPGA chief companies provide the possibility of utilising soft-core processors particularly designed to well match in the FPGA. At the same time, FPGAs permit the utilisation of hard-core processors. Moreover, FPGAs are fitted with on-chip memory blocks, peripherals, and interconnection circuitry. One of the FPGA based multiprocessor strengths is run-time reconfiguring ability. This characteristic provides multiprocessor systems an adaptation feature to a particular application, providing the designed system with much flexibility.

1.2 Problem statement

In the last decade the design and implementation of an embedded multiprocessor has been taken a lot of attention for their wide applications and need in different areas of humans lives. The continuous enhancements where mostly focused on the throughput and the ability to process large amount of data. Then new characteristics emerged with the great steps that have been achieved in the industry of silicon chips. Characteristics like the ability of these systems to process data in real-time respond and taking an

accurate decision with low complexity as much as possible. Other important factors like cost, power consumption and time to market also emerged as parameters to be considered.

Recently, embedded multiprocessor sharing peripherals that were implemented with different platforms suffered from significant issues. One of the faced issues is synchronisation techniques, which is how to coordinate the access of the connected processors to the peripheral they share, preventing the case of thread safety. Also having deadlocks where two processors waiting for the other to release a resource it holds. Load balancing problems also come to surface as the need to increase the number of connected processors where bad load balancing and workload distribution would cause waste of resources with the occurrence of an idle processor waiting for a process. For these fundamental causes and more, the utilization of platforms that does not hand the high requirements demanded to implement MPSoC will be only a waste of time.

The performance of the designed system is very much related with the use of an efficient programmable chip. FPGAs with all the supported tools have opened new horizons for the design of such demanding systems. In this proposed design the utilisation of FPGA board was the main reason in facilitating the design and achieving all objectives successfully.

1.3 Dissertation Objectives

The objectives of the proposed design are:

1. To enhance the implementation of embedded concurrent processor by the specifications afforded in the use of NiosII and shared memory on FPGA achieving better workload balance and providing even distribution of the handed tasks among

the connected processors by performing an adequate balancing algorithm also fulfilling the following specification:

i. Achieving efficient throughput with low level of complexity, high operating frequency and consuming as less on chip resources.

ii. Enhancing memory coherency which is an issue that affects the design of computer systems where two or more processors share a common area in the memory. This enhancement is achieved by reducing hazard and increasing memory hits by using multiple memory ports to allow concurrent operations.

2. To verify and evaluate architecture complexity performance in terms of high operating frequency, consume as less chip resources and efficient throughput by using CAD tool and FPGA board (NEEK) testing.

1.4 Dissertation Scope

This dissertation focused on the steps taken to make the enhancement demanded on the implementation of the embedded processor that operates in a concurrent manner. Consideration was also given to the concepts of choosing the right tools that would lead to achieve the goals of this research along with addressing the faces issues. Multiprocessor type, architecture style, communication style and transferring data methods were all discussed through intensive comparison. To provide the beholder of this research with a beneficial understanding all the subsequent results to the finals were discussed and analysed.

1.5 Contribution

The contributions of this dissertation are:

1. The processing time is significantly improved while implementing the same application and the on chip multiple I/O consumption is decreased.
2. Achieving lower level of design complexity, high operating frequency, less chip resources and scalable throughput.

1.6 Dissertation Outline

This thesis is constructed of five chapters

Chapter 1 contains a brief presentation of the research approach on FPGA and some background information on the concepts that was dealt with. The aim and the demanded objectives of this research were listed. Pointing out the faced problems when dealing with the multiprocessing environment, scope of the research and finally listing some possible challenges that could be faced with the utilisation of FPGAs.

Chapter 2 and in details, highlighted the multiprocessing environment and the parallel concurrent processing with the benefits it hands over the old limited functionality of the singular processing. The development of MPSoC was also presented, with all the interest given to these systems over years. The benefits gained by the utilisation of FPGAs over other platforms were discussed in details. And for better credibility the challenges faced with FPGAs platform was also listed.

Chapter 3 discusses the implemented methodology and the construction of the utilised hardware tools with a brief on the implemented application. The design was divided into two phases, the first phase was designing an enhanced embedded concurrent processor with shared memory on FPGA utilising all supported tools with

the selected NEEK board like the Qsys and the SBT for Eclipse. The second phase was testing and selecting the implemented application which was Mandelbrot-set. A suitable load balancing algorithm was selected to reduce the effect of an idle processor and its effect on degrading the systems performance.

Chapter 4 discussed the reason for choosing the implemented application and mainly presented the achieved results. The improvement made in the throughput by the proposed embedded concurrent processor was pointed out and compared with that in a single processor.

Chapter 5 presented the conclusion of the entire work done in this design, suggested future work and way to expand multiprocessing systems and their computing power.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Computer hardware and software are usually the main components of embedded system and additionally in some cases, there are mechanical or other components which are designed to execute certain dedicated function. They could be a part of other large system or a product (Thiele, Bacivarov, Haid, & Huang, 2007). These systems recently became very popular and invaded human lives for their size, low cost and simple design. The embedded systems can be sorted out to four types which are stand-alone embedded systems, real time embedded systems, network information appliances and mobile devices. The classification is made according to the requirements of functionality and performance of those systems. Based on the performance of microcontroller embedded systems they can be classified in to (i) small scaled embedded systems (ii) medium scaled embedded systems (iii) large scaled embedded systems. The embedded systems are known for their limitations in resources, specially the memory. Usually, there is an absence of secondary storage devices like the CD-ROM or the floppy disk. Embedded systems also must function versus some deadlines.

2.2 Embedded System Architecture

Essentially embedded system comprises of a Central Processing Unit (CPU) and around it there is a custom-built hardware. A memory chip is contained in the hardware part where the software is loaded on it the loaded software is known as 'firmware'. Above the hardware runs the operating system, and above the residing operating system runs the application software.

To any computer (including a desktop computer) this architecture is relevant. Nevertheless, there are substantial differences in embedded systems such as it is not compulsory to have an operating system. For example, in remote control units or toys which are considered small appliances the operating system is not required, where just a software specific to the application is written. On the other hand, it is smart to have an operating system applications involve complex processing. The application software with the operating system should be integrated and the whole software is transferred to the memory chip. In Figure 2.1 depicts the layered architecture of an embedded system.

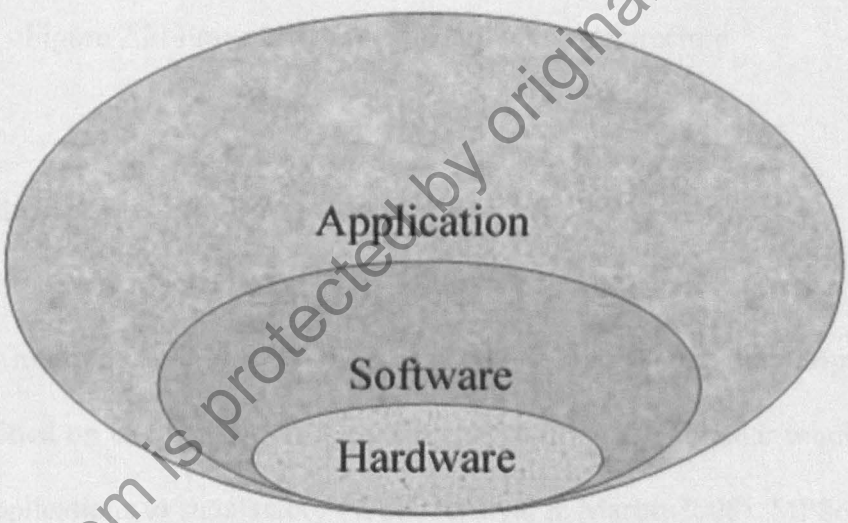

Figure 2.1: Layered Architecture of an Embedded System

The embedded systems hardware is constructed of the following parts:

- Central Processing Unit (CPU)
- Memory (read only memory and random access memory)
- Input/output devices
- Communication interfaces
- Application specific circuitry