CUSTOMIZABLE ASSEMBLER DESIGN FOR DYNAMIC INSTRUCTION SET ARCHITECTURE (ISA)

MOHAMAD HAZARI BIN ZINAL IBIADIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CUSTOMIZABLE ASSEMBLER DESIGN FOR DYNAMIC INSTRUCTION SET ARCHITECTURE (ISA)

MOHAMAD HAZARI BIN ZINAL IBIADIN

This report is submitted in partial fulfillment of requirements for the Bachelor Degree of Electronic Engineering (Computer Engineering)

Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka

June 2014

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSTI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : CUSTOMIZABLE ASSEMBLER FOR DYNAMIC DESIGN INSTRUCTION SET ARCHITECTURE (ISA)

Sesi Pengajian : 1 3 / 1 4

Saya MOHAMAD HAZARI BIN ZINAL IBIADIN **(HURUF BESAR).** Mengaku membenarkan Laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (√) :

☐	**SULIT***	*(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD****	**(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	**TIDAK TERHAD**	

Disahkan oleh:

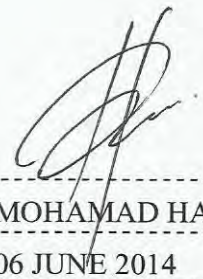
(TANDATANGAN PENULIS)

Tarikh: 06/06/2014

(COP DAN TANDATANGAN PENYELIA)
NUR FATIHAH BINTI AZMI
Pensyarah
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya,
76100 Durian Tunggal, Melaka

Tarikh: 6/6/2014

"I hereby declare that this report is the result of my own work except for quotes as cited in the references."



Signature : .....

Author : MOHAMAD HAZARI BIN ZINAL IBIADIN

Date : 06 JUNE 2014

"I hereby declare that I have read this report and in my opinion this report is sufficient in terms of the scope and quality for the award of Bachelor of Electronic Engineering (Computer Engineering) With Honors'."

Signature	:	
Supervisor's Name	:	PUAN. NUR FATIHAH BT AZMI
Date	:	06 JUNE 2014

Dedicated to my beloved family especially father & mother, lecturer and to all my beloved friends.

ACKNOWLEDGEMENT

Thank God, because of His grace, I have been able to implement this project successfully and I want to take this opportunity to express my appreciation to all those who helped me a lot in completing this FYP. I would like to express appreciate to my family because support me to continue my study in computer engineering. They also support me in term of moral, fund and transportation in my university life.

I appreciate to Mr. Sani Irwan bin Md Salim, because he created and introduced this project title to me. I appreciate to Pn. Nur Fatihah bt Azmi, because she was willing to supervise and give comments in my project.

Finally, I would like to thank to my friends, who always ready to serve and help me. I express my gratitude and hope that God will repay you well.

ABSTRACT

In the development of processor architecture, one of the crucial parts is the creation of a compatible assembler to the processor's instruction set architecture (ISA). Reconfigurable processor such as UTeMRISC03 requires a flexible assembler design in order to accommodate the modification being made to its ISA. The objectives of this project are to design a customizable assembler for dynamic instruction set architecture and to verify the assembler is compatible with the UTeMRISC03 processor architecture. In order to meet the objectives, a new assembler has been designed using Visual Basic (VB). The designed software is user friendly because it is able to modify according to any design requirement. The new assembler is capable in converting an assembly language program to its instruction word dictated by the processor's opcode file. The correct object file is also generated in line with the selected ISA width determined by the users. The object file is successfully loaded to the processor architecture in the FPGA platform in order to verify its compatibility. With the customizable feature achieved in this assembler design, the assembler would be beneficial as the main tool in the development of a complete package in a reconfigurable processor development in the future.

ABSTRAK

Dalam pembangunan seni bina pemproses, salah satu bahagian penting ialah penciptaan satu penghimpun serasi bagi seni bina set suruhan pemproses. Pemproses boleh dikonfigurasi semula seperti UTeMRISC03 memerlukan satu penghimpun fleksibel mereka supaya menampung pengubahsuaian dibuat ke senibina set arahan nya (ISA). Objektif projek ini akan mereka satu penghimpun yang menyesuaikan untuk suruhan dinamik seni bina tetap dan mengesahkan penghimpun serasi dengan seni bina pemproses UTeMRISC03. Untuk mencapai objektif, satu penghimpun baru telah direka bentuk menggunakan Visual Basic (VB). Perisian bercorak ramah pengguna kerana ia mampu mengubah suai menurut mana-mana keperluan reka bentuk. Penghimpun baru berkebolehan dalam menukar satu bahasa perhimpunan merancang kepada kata suruhannya diperintah oleh fail opkod pemproses. Fail objek betul juga dijana seiring dengan kelebaran ISA terpilih berazam oleh pengguna. Fail objek berjaya memuatkan kepada seni bina pemproses di platform FPGA supaya mengesahkan keserasiannya. Dengan ciri yang menyesuaikan dicapai dalam mereka penghimpun ini, penghimpun akan menguntungkan sebagai alat penting di pembangunan satu pakej lengkap di pembangunan pemproses boleh dikonfigurasi semula pada masa akan datang.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURE

LIST OF ABBREVIATIONS

FPGA	–	Field Programmable Gate Array.
ROM	–	Random Access Memory
FKEKK	–	Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer.
IEEE	–	Institute of Electrical and Electronics Engineers
ISA	–	Instruction Set Architecture.
RISC	–	Reduce Instruction Set Computer.
UTeM	–	Universiti Teknikal Malaysia Melaka
COE	–	Coefficient
LST	–	Listing
INT	–	Intermediate
FYP	–	Final Year Project

LIST OF APPENDIX

CHAPTER I

INTRODUCTION

1.1 Overview

A reconfigurable processor is a processor core that with configurable internal architecture in order to improve performance and efficiency. Within the internal architecture, the instruction set architecture act as the intermediate module between user program code (software) and hardware implementation. When instruction set architecture was modified, the processor's assembler must adhere the changes. The assembler is design using visual basic with two-pass assembly method and applies the tokenization and lexical analysis procedure to perform assembling process. The assembler‟s output is a listing file formatted as coefficient file (.coe) will used to instantiate the ROM module during the FPGA implementation of the processor core.

1.2 Objectives

The objectives of this project are as follows:

i. To design a customizable assembler for dynamic instruction set architecture.

ii. To verify the assembler‟s object file is compatible with the UTeMRISC03 processor architecture.

1.3 Problem Statement

Processor‟s assembler is tied up to its architecture. In case of reconfigurable assembler, the modification of the architecture will causes the existing assembler not compatible with the processor. Any changes on processor‟s architecture such as new instruction set, bus modification and memory expansion would alter the system configuration. Hence, the machine code needs to comply with the latest version of the architecture. Having a customizable assembler that capable in producing object files that matched the processor architecture could solve this problem.

1.4 Scope

The assembler is developed using two-pass assembler approach. The dynamic instruction set architecture is to be mode optional 16-bit and 22-bit.

1.5 Report Structure

This thesis delivery the concept applied, method used, problem solving, finding analysis and result of visual basic. This thesis consists of five chapters and the following briefly describe what contents each chapter has. But for this PSM 1 two last chapter it not required to submit because the not yet the result.

In chapter 1, the purpose is to give reader an overall picture about what is actually this project doing. Introduction, objectives, scope of project, problem statement and summary of methodology are able to introduce this project to reader.

In chapter 2, the literature review of project is explained in detail. Study was done for existing assembler and the disadvantages of existing assembler were found. Then, the theoretical concept that applied in this project based on the Visual Basic programming technique and target hardware also state in this chapter.

In chapter 3, the methodology of project is described. The processes of design are shown step by step. There are four main functions need to be designed which are load, tokenization, lexical analysis and decode to get the output. The assembler also will explain on how the main functions of existing assembler working. Then, the programming using the Visual Basic also will explain on how to design the new interface assembler.

In chapter 4, all results from project are included. The results are majority focus on assembler are using Visual Basic. That‟s mean the result assembler will shown as Visual Basic interface. The COE file and LST file will generate after the assembler is done. The file also will test at FPGA processor to verify the file is creating well. That‟s the result will shown.

In chapter 5, a conclusion is made for the project that carried out in final year. The conclusion included project discovering, analysis achieved and future enhancement. Besides, the accuracy of project results will be concluded by comparing with objectives and problem statement. Finally, the important of this project to the target user will also describe

.

CHAPTER II

LITERATURE REVIEW

Based on the literature review, the assembler and visual basic is already complete on study about the basic function and the step to use the software. For creating the new assembler need to know the previous information like what language and type are used. From the finding, the previous assembler is used Perl language to create TINYASM [1] and also using eclipse. IEEE is the better website to find the information journal and that is a more useful for this project information. Based on the journal, the assembler is a translator that translates source instructions in symbolic language into target instructions in machine language [2], on a one to one basis. The reason is operation of an assembler reflects the architecture of the computer. The assembler language depends heavily on the internal organization of the computer. The features architectures such as memory word size, number formats, internal character codes, index registers, and general purpose registers, affect the way assembler instructions are written and the way the assembler handles instructions and directives [2]. For visual basic part is used to design the interface of the assembler and should know the all functions on the software.

2.1 Reconfigurable Processor or Soft-Core Processor

Reconfigurable processor or soft core processor is a programmable processor already implement in to embedded computing systems. Example a soft core processor is field programmable gate arrays (FPGAs) processor. A soft core FPGA processor is a synthesizable processor mapped onto the FPGA fabric in contrast to a hard core processor that is laid out next to the FPGA fabric[3]. A single instruction set computer, a processor with a certain function itself can be reconfigured to another processor with another function[4].

2.2 Relationship ISA into Microprocessor

Instruction set architecture is the structure of a computer that a machine language programmer (or a compiler) must understand to write a correct (timing independent) program for that machine[5]. The Microprocessor is a processor using small instruction set to process the application in their control processing unit. The instruction set architecture almost related into microprocessor because the instruction set architecture is the part of the processor that is visible to the programmer or compiler writer. The ISA serves as the boundary between software and hardware.

2.3 UTeMRISC03 Processor

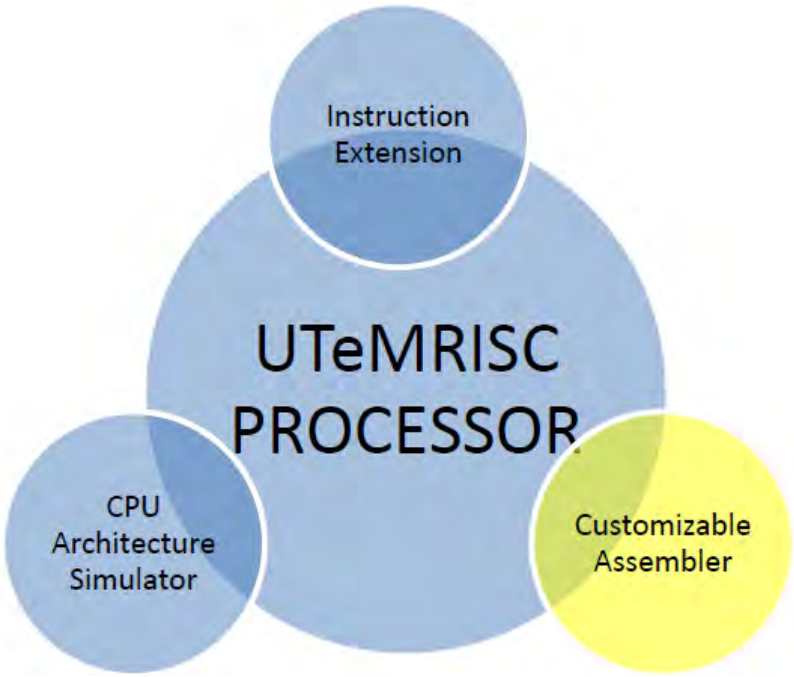


Figure 2.1 UTeMRISC03 Processor Development[6]

In general, UTEMRISC03 processor is an inititive from researchers in UTeM to develop a complete suite of reconfigurable processor design for specific applications. The project involved several key aspect of processor design including the architecture, instruction set extension, customizable assembler and the architecture simulator.

As part of the UTeMRISC03 processor project, the customizable assembler design is essential in order to accommodate any changes being made to the processor architecture and also during the extension of the instruction set list. Being tightly-coupled with the processor's ISA, the assembler design need to be flexible to generate the correct object files based on the current ISA configuration.

2.4 What is an Assembler

An assembler is a translator that translates machines oriented languages into machine languages[2]. The assembler consist are many block in process such as table search procedure, opcode table, symbol table, directive table, source file, source line buffer, lexical scan routine, error process, object code assembly area object file, pass indicator and location counter. This block function is needed to run a main program in an assembler. The type of an assembler is a one-pass assembler, a two-pass assembler, a resident assembler, a macro-assembler, a cross-assembler, a meta-assembler, a disassemble, a high level assembler, and a micro-assembler [2]. There are many type of assembler, but the focus assembler type is two-pass assembler. A two pass assembler also needs a one pass assembler to complete the sequence of process.

2.4.1 Main component and operation of assembler

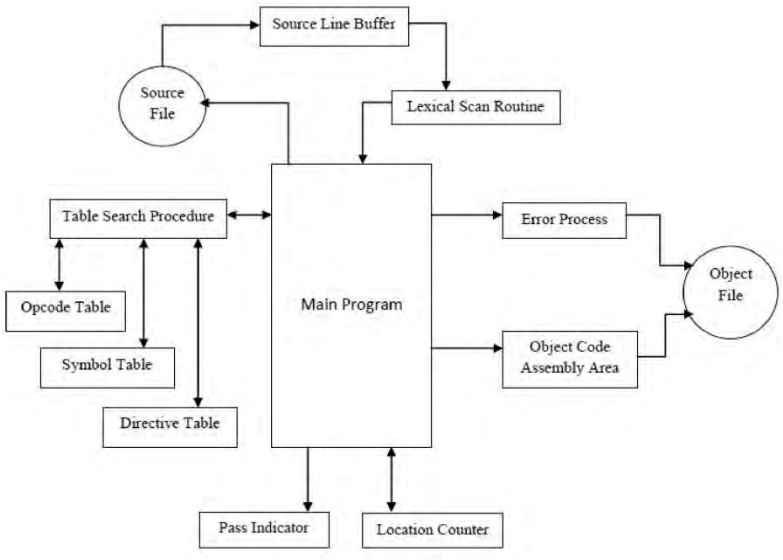


Figure 2.2 Main Component and Operation of Assembler[2]