# Design 8-Bit RISC Processor Base on X86 Architecture using VHDL

Industrial Project Thesis is presented in partial fulfillment for the award of the
Bachelor of Electrical Engineering (Honors)
UNIVERSITI TEKNOLOGI MARA

MUSNAZRIL BIN MUSTAQ KHAN
Faculty of Electrical Engineering
UNIVERSITI TEKNOLOGI MARA
40450 Shah Alam, Selangor

# ACKNOWLEDGEMENT

I would like to convey my million thanks to my most respected supervisor Puan Wan Fazlida Bte Hanim, for her priceless assistance, guidance and support throughout the development of the project. Without her help, this project may not be materialized. I would also like to express my utmost gratitude to all who have been involved directly or indirectly. May Almighty Allaw swt bless and reward them for their generosity.

# ABSTRACT

This paper presents the process of building a basic x86 RISC processor using VHDL. The RISC conceptual design had been taken out from the complex CISC X86-Architecture. The development process is performed using VHDL programming language and optimized using Xilinx schematic editor. The modules designed are the Arithmetic Logic Unit, Control Unit and Register Set. The tests performed are based on a special test program, which is build inside the processor Read Only Memory.

# TABLES OF CONTENT

CHAPTER PAGE

# CHAPTER 1
# INTRODUCTION

## 1.1. VHDL for Processor Design Summary

Historically, families of increasingly complex processor have dominated the evaluation of computer architecture. Under market pressure to preserve existing software, Complex Instruction Set Computer (CISC) architecture involved by the gradual accretion of microcode and increasingly elaborate operation. The intent was to supply more support for languages and operations systems, as semiconductor advances made it possible to fabricate integrated circuits that are more complex. It seemed self-evident that architecture should become more complex as these technological advance made it possible to hold more complexity VLSI devices.

In recent years, however, Reduces Instruction Set Computer (RISC) architecture has implemented a much more sophisticated handling of the complex iteration between hardware, firmware, and software. RISC concepts emerge from statistically analysis of how software actually uses the resource of the processor. Dynamic measurement of system kernels and object module generated by optimizing compilers show an overwhelming predominance of simplest instructions, even in the code for CISC machines. Complex instruction are often ignored because of single way of performing a complex operation rarely matches the precise needs of high-level language and system environments. RICS architecture widely accepted for its capability in performing single complex operation in various ways.