Leo De Silva

A Level Computer Science

DESIGNING & MAKING THE SOFTWARE SUITE

for a proprietary machine code specification.

2024, St Albans School

Contents

1 Analysis

1.1 Problem Defenition

The goal of this project is to design the hardware specification for a custom 16-bit single cycle CPU, and develop the suite of tools required to simulate such a processor, including an Emulator (1.2.2), Assembler (1.2.3), and Compiler (1.2.4). The project will detail the abstract design of the computer's Instruction Set Architecture (ISA) (1.2.1) and its implementation in hardware, considering the internal registers, system clock, main memory, and fetch execute cycle.

The project will compose three primary parts, an emulator capable of loading machine code 'catridges' and simulating the hardware, memory, peripherals and registers detailed by the ISA in order to execute programs with correct clock timing and behaviour. An assembler to translate programs written in an assembly specification into binary machine code (the assembler will handle higher level conveniences such as relative addressing through labels). And finally a compiler - to translate a higher level programming langauge into machine code. This will require compiler optimisations with relation to the produced object code; complex data structures such as arrays, objects and strings; conditional and interative expressions; and functions and procedures. The suite required to simulate such a computer should be capeable of writing and compiling complex programs such as pong or tetris, and emulating them with hardware correct timings - dealing with peripherals such as a keyboard or speaker.

1.2 Background to the Problem Area

This project will explore lower level systems software and look in detail at the fundemental architecture of modern computing systems and how they are designed from the ground up.

1.2.1 Instruction Set Architecture

The ISA acts as an interface between the hardware and software of a computing system, and contains crucial information regarding the capabilities of a processor, including: a functional defenition of storage locations (e.g. registers and memories) as well as a description of all instructions and operations supported.

An ISA can be classified according to its architectural comlpexity into a Complex Instruction Set Computer (CISC), or a Reduced Instruction Set Computer (RISC). A CISC processor implements a wide variety of specialized instructions in hardware, minimising the number of instructions per program at the cost of a more complex design, higher power consumption and slower execution as each instruction requires more processor cycles to complete. Joshi (2024) Thus optimisations in performance occur on the hardware level. This is historically the most common branch of processor and often results in processors with large instruction sets, for example the Intel x86's 1503 defined instructions Giesen (2016). A RISC processor however aims to simplify hardware using an instruction set containing a few basic instructions to load, evaluate and store data - instead placing optimisations on the software side unlike a CISC processor.

1.2.2 Emulator

An emulator is a software program that allows one computer to imitate another - and by simulating the hardware of the other – execute machine code programs written for a processor other than itself.

1.2.3 Assembler

An assembler is a program that translates assembly language (a low level programming language that uses mneumonics to directly represent machine code instructions) into object code that can be executed by the processor.

1.2.4 Compiler

A compiler is a program that translates high level program source code into a set of machine language instructions. Some compilers translate source code into an intermediate assembly language before using an assembler to produce the machine code instructions, whereas others compile into machine code directly.

1.3 Programming Language

1.4 Existing Systems

1.5 Prototyping

1.6 Client Proposal

1.6.1 Client Interview

1.7 Objectives

2 Design

3 Technical Solution

4 Testing

5 Evaluation

References

Giesen, Fabian (2016). *How many x86 instructions are there?* URL: `https://fgiesen.wordpress.com/2016/08/25/how-many-x86-instructions-are-there/`. (accessed: 14.04.2024).

Joshi, Vibha (2024). *RISC and CISC in Computer Organization.* URL: `https://www.geeksforgeeks.org/computer-organization-risc-and-cisc/`. (accessed: 14.04.2024).