Disassembler project

By Martin (Xiao) Huang, Tien Huynh, Jeremy Luxon

Date: 12/13/2017

# Program description

This program is the final project for our CSS 422 Hardware and architecture design class. We are creating the disassembler that is to revert the machine code stored in memory back into human readable format. Over the past four weeks we have been working on the project and produce a qualified deliverable.

## Project model

We choose to use waterfall model to design our program. Here is the four phases of our project.

### Designing and Modeling

The disassemble can be divided into three sub sections: UI, opcode, and EA handling.

UI: The UI is the section that controlling user input and output. The program takes input from keyboard by users and outputs specific strings to react to users. For example, we designed our program to has a welcome page that welcome the users when they first click on the program. Then it would ask users type in the starting address and the ending address to be decoded. After decoding finished. It would display the result.



opCode: The opcode part is the section that identifying what operation code it is when reading the instruction from memory. The Motorola instruction code is 16 bit. Most of the opcode has their identifier bits, mode bits(3bits), and register bits(3bits). If there is effective addressing mode happens we will discuss in the next section.

EA: The EA part is the section that dealing with effective addressing mode.

1. Data Register Direct
2. Address Register Direct
3. Address Register Indirect
4. Immediate Data
5. Address Register Indirect with Post incrementing
6. Address Register indirect with Pre decrementing
7. Absolute Long Address
8. Absolute Word Address

Week 2: Researching on Motorola M68K manual

Week 3: Building the disassembler

Week 4: Testing and Wrapping