Assembly 1 Project

Due on June 16th, 2019

Computer Organization & Programming

CS555WS—Summer I

Ed Banduk

Daniel Kadyrov

The MC68000, or 68000, processor was introduced by Motorolla in 1979. It was one of the first widely available processors with a 32-bit instruction set and was used in personal computers manufactured by Apple, Commodore, Atari, and IBM in the 1980s. Its architecture consists of an address bus, internal registers, status register, instruction set, privilege levels, and interrupts (Zabidi).

An addressing mode is the way an operand of an instruction is specified. The addressing modes specify the source and destination of the data. The 68000 processor has 14 addressing modes that fall into several categories including inherent, immediate, relative, extended, and indexed.

The inherent, also known as implied, addressing mode has no operation and no operand. The address is designated by the `INH` command and its function is to increase the register.

In immediate mode, the operand is an immediate, or literal, value that is stored explicity in the instruction. In the syntax, the memory address immediately follows the instruction word and a # is in front of the source operand. The following example uses the `MOVE` command to copy the literal value of 7 to data register `D1`. Immediate addressing is used for constants that do not change and can only be used as a source addressing mode.

```
    MOVE.B #7, D1
```

In absolute mode, the effective address is defined. In relative mode, the effective address is generated by adding a constant value to the content of the program counter. Extended address mode has a 16-bit address of the 2nd operand and can process variables in any location of memory space.

In basic index mode, the effective address is generated by adding a constant value to the address register content. In full index mode, the constant value is added to teh contents of two registers.

In the example below, the value of 75 is added to A2 and to D4, and then stored in the D2 register. Below it, the value of 75 is subtracted from both the values in A2 and D4 and stored in D2 (MC68000 Manual, 1989)

```
    ADD $75(A2), D4
    SUB 5(A2, D4), D2
```

68000 Assembly Language is a low level programming language for the operation of a 68K CPU. Assembly language, and high level languages than stem from it, are mnemonic so humans are able to read and develop the code easier. Assembler programs convert this code into executable, binary, machine code (Englander, 2014).

The 68000 simulators are programs written for modern machines to emulate running programs on the Motorola 68000 without any 68000 hardware. The simulator performs a set of instructions as though it was a 68000 processor. There are many available 68000 simulators available online, many with an open-source license, and are able to run on Windows, MacOS, and UNIX systems (Crossware). The most popular are from Crossware, XRMX, and Tripod.

References

[1] Zabidi, M. M. (n.d.). *68000 Architecture*. Lecture. Retrieved June 15, 2019, from `http://ocw.utm.my/pluginfile.php/1305/mod_resource/content/0/02-68kArchitecture.ppt.pdf`

[2] *M68000 family programmers reference manual*. (1989). Phoenix, AZ: Motorola.

[3] Englander, I. (2014). *The architecture of computer hardware, systems software, & networking: An information technology approach* (Fifth ed.). Hoboken, NJ: John Wiley & Sons.

[4] (n.d.). Retrieved from `https://www.crossware.com/m68xxx/Simulator`