# ASSEMBLY 1 PROJECT
# Computer Organization & Programming

Class: CS 550 A

Name: Akshatha Vasant Hegde

CWID: 20009287

**1) What is the 68000 Processor? (Mentioning about 68000 addressing modes: inherent, immediate, relative, extended, indexed)**

Motorola 68000 is the first member of 680x0 line of microprocessors. The 68000 is Motorola's first 16-bit microprocessor. Its address and data registers are all 32 bits wide, and its ALU is 16 bits wide. [1]

The processor provides 6 different addressing modes, each of which has multiple versions, for a grand total of 14.

- Register Direct

  Reads or writes data in one of the microprocessor's registers. There are two versions of it - Data Register Direct and Address Register Direct Mode

- Register Indirect

  The addressing is performed using a register, but the data is accessed indirectly. The register contains the address in memory of the data. The different types are - Register Indirect, Register Indirect with Postincrement, Register Indirect with Predecrement, Register Indirect with Displacement, Register Indirect with Index.

- Absolute Data

  This mode provides a version of Register Indirect mode where the address is specified directly in the instruction and not through a register. The subtype is Absolute Long Data

- Program Counter Relative

  The addressing modes relative to the Program Counter (PC) are used in relocatable programs, as the effective address is computed as a displacement from the address of the current instruction being executed. The subtypes are -Program Counter Relative with Displacement and Program Counter Relative with Index

- Immediate Data

  Immediate data uses the plain data written in the extension words instead of referring to the system memory. A type of Immediate data is Quick Immediate.

- Implied

  This is another mode that is available only for some instructions. Those are bound to specific registers and are thus not really allowing any generic effective address to be used.[2]

**2) What is the 68000 Assembly Language?**

For programming instruction codes, we assign a name to each instruction code. The instruction code name is called a “mnemonic” or memory jogger. The instruction mnemonic should describe, in a minimum number of characters, what the instruction does. They are standard for a given microprocessor, and therefore understood by all users. There is a proposed standard set of assembly language mnemonics. Such a program is an assembly language program.[1]

Assemblers use pseudo instructions or directives to make the formatting of the edited text easier. These directives are not translated directly into machine language instructions.[3]

## 3) Why are we using Assemblers?

The assembler program translates a user program, or source program written with mnemonics, into a machine language program, or object program, which is then executed. The assembler's input is a source program, and its output is an object program.[1]

Thus, the assembly language program is translated into binary. The assembler program reads each assembly instruction of a program as ASCII characters and translates them into the respective binary opcodes.[3]

Assembly is a task that we assign to the microprocessor as it never makes any mistakes when translating codes; it always knows how many words and what format each instruction requires.[1]

## 4) What is the 68000 Simulator?

A simulator is a program that allows the user to observe an operation through simulation without performing that operation.[4] The simulator goes through the operating cycle of a computer, keeping track of the contents of all the registers, flags, and memory locations.[1]

Simulation software is used widely to design equipment so that the final product will be as close to design specs as possible without expensive in process modification.[4]

The Motorola 68000 simulator included in the BSVC distribution simulates the 68000 at only the software level. This means the simulator does not understand what goes on in the 68000 at the hardware level. Instead, the simulator performs a set of actions for each instruction that gives the same result.[5]

## Bibliography

[1] Kane, G., Hawkins, D., & Leventhal, L. A. (1981). In 68000 assembly language programming. essay, Osborne/McGraw-Hill.

[2] Motorola 68000: Addressing modes. The Digital Cat. (n.d.). Retrieved March 4, 2022, from https://www.thedigitalcatonline.com/blog/2019/03/04/motorola-68000-addressing-modes/

[3] Wiley-Interscience. (n.d.). Microprocessor theory and applications with 68000/68020 and pentium. O'Reilly Online Learning. Retrieved March 4, 2022, from https://learning.oreilly.com/library/view/microprocessor-theory-and/9780470380314/13_ch06.html#ch006-sec004

[4] Wikimedia Foundation. (2022, February 5). Simulation software. Wikipedia. Retrieved March 4, 2022, from https://en.wikipedia.org/wiki/Simulation_software

[5] Mott, B. W. (n.d.). Chapter 3 Motorola 68000 simulator and Assembler. chap3.htm. Retrieved March 2, 2022, from https://www.csc.kth.se/hacks/doc/bsvc/chap3.htm