Stevens Institue of Technology

Assembly 1 Project

Yash Avinash Patole
CWID: 10460520

Computer Organization and Programming
Prof. Edward Banduk
Date: 03/06/2021

Student name: Yash Avinash Patole
Assembly 1 Project

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

Answer:

The Motorola 68000 processor has 18 registers that are directly accessible by the user. Those are:

- 8 general-purpose data registers numbered from D0 to D7
- 8 address registers numbered from A0 to A7
- 1 Program Counter (the PC)
- 1 Status Register (the SR), which is divided in 2 parts: the System Byte, which we'll consider in later lectures, and the Condition Code Register (the CCR).

The 68000 user registers are 32 bits wide, i.e., they can accommodate 32 bit numbers. In order to be able to reference a particular bit in the register, each bit is given a number. The convention is to number the bits from 0 to 31, bit 0 being the least significant, i.e. the rightmost bit, and bit 31 being the most significant, i.e. the leftmost bit. In other words, the bit numbering starts at 0 at the right end and goes from right to left in increasing order.

Immediate Data Addressing

- a) Immediate
- b) Quick Immediate
- Any address register may be used for direct or indirect addressing, and any register may be used as an index register.

Student name: Yash Avinash Patole
Assembly 1 Project

Inherent:

- The locations of data are implied by the instruction name
    - Example: ABA, which adds register B to A, implies the registers in the instruction name
    - Note: no operands are needed to indicate memory to work with or constant value to load.

Extended: The address of the data is located in memory immediately following the instruction's opcode

- Format: the instruction name is followed by a label that is in the assembly program somewhere, or is followed by a constant value, without a # sign.

Relative: The value in the operand is computed from the relationship between where you are currently in the code in memory to where you are going to, the label.

Indexed: The value in an index register (X or Y) is added to an offset value, which provides the final effective address of the data.

2. What is the 68000 Assembly Language?

Answer:

- Instructions are listed by mnemonic in alphabetical order. The information provided about each instruction is: its assembler syntax, its attributes (i.e., whether it takes a byte, word, or longword operand), its description in words, the effect its execution has on the

condition codes, and the addressing modes it may take. The effect of an instruction on the CCR is specified by the following codes: U The state of the bit is undefined (i.e., its value cannot be predicted) - The bit remains unchanged by the execution of the instruction * The bit is set or cleared according to the outcome of the instruction.

Instruction	**RTL representation**	**Description in words**
MOVE #N,D1	[D1] <- N	Register D1 is loaded with the number N
MOVE D1,L	[M(L)]<- [D1]	The contents of register D1 are copied to memory location L.
MOVE D1,D2	[D2] <- [D1]	The contents of register D1 are copied to register D2
ADD D3,D7	[D7] <- [D3] + [D7]	The contents of register D3 are added to the contents of register D7, and the result is stored in register D7
ADD #N,D0	[D0] <- [D0] + N	The number N is added to the contents of register D0 and the result is stored in D0
SUB #N,D0	[D0] <- [D0] - N	The number N is subtracted from the contents of register D0 and the result is stored in D0
SUB D1,D5	[D5] <- [D5] - [D1]	The contents of register D1 are subtracted from the contents of register D5, and the result is stored in register D5
CMP #N,D2	[D2] - N	Subtract the number N from the contents of register D2. The result is discarded and the CCR is set up.
CMP D1,D2	[D2] - [D1]	Subtract the contents of D1 from the contents of D2. The result is discarded, and the CCR is set up.
BEQ X	IF CCR(Z) = 1 THEN [PC]<- X	Branch to location X if the Z bit of the CCR is set, i.e. if the previous operation yielded zero as result.

BNE X	IF CCR(Z) = 0 THEN [PC]<- X	Branch to location X if the Z bit of the CCR is cleared, i.e. if the previous operation didn't yield zero as result.

3. Why we are using Assemblers?

Answer:

Assembly language helps programmers to write human-readable code that is almost similar to machine language. Machine language is difficult to understand and read as it is just a series of numbers. Assembly language helps in providing full control of what tasks a computer is performing. The assemblers are used to translate the assembly language into machine language. There are two types of assembler are:

1)Single-pass Assembler: Single-pass assembler scans the program only once and creates equivalent binary program.

2)Multi-pass Assembler: In this more than one pass is used by an assembler.

Main advantages of using an assembler are:

1. It allows complex jobs to run in a simpler way.
2. It is memory efficient, as it requires less memory.
3. It is faster in speed, as its execution time is less.
4. It is mainly hardware oriented.
5. It requires less instruction to get the result.
6. It is used for critical jobs.
7. It is not required to keep track of memory locations.

8. It is a low-level embedded system.

4. What is the 68000 Simulator?

Answer:

The **68000 Simulator** is a simulator for the 68000 microprocessors. The simulator will run any 68000-program contained in a listing file with properly formatted. This type of file is usually saved in. LST format. The **68000 Editor** also was developed as complement to the **68000 Simulator**.

The **68000 Simulator** output has been tested on Windows XP, Windows 98 and Red Hat Linux 9 platforms.

The software only can read the listing file that have same format as listing file created by Flight68k Cross-Assembler.

Basic requirements for 68000 Simulator is Java 1.4. It also supports versions above 1.4 and other compatible JVM. 68000 Simulator is usually written in Java.

Student name: Yash Avinash Patole
Assembly 1 Project

References:

1] Savadjiev, P. (n.d.). Lecture 9: The registers and the memory of the 68000; intro To 68000 PROGRAMS. Retrieved March 07, 2021, from https://www.cs.mcgill.ca/~cs573/fall2002/notes/lec273/lecture9/#82

2] Basic addressing modes information. (n.d.). Retrieved March 07, 2021, from https://www.cs.nmsu.edu/~hdp/cs273/notes/addressing.html

3] What is Assembly Language?: Features: Advantages and disadvantages. (2021, March 05). Retrieved March 07, 2021, from https://www.educba.com/what-is-assembly-language/

4] Projects, C. (2021, March 02). 68000 assembly. Retrieved March 07, 2021, from https://en.wikibooks.org/wiki/68000_Assembly