



**KHULNA UNIVERSITY OF ENGINEERING AND TECHNOLOGY,
KUET**

SESSIONAL REPORT

Course No: CSE 2204

Department of: Computer Science and Engineering

Experiment No: 02

Name of the Experiment: Developing a program that perform
Different types of jump instruction in assembly language

Remarks

Date of Performance: 12.04.21

Name: Rifat Arefin

Date of Submission: 19.04.21

Roll: 1807117

Year: 2nd

Semester: 2nd

No. of experiments 2

Name of experiment: Developing a program that performs different types of jump instruction in assembly language

Objective:

1. To obtain about jump instructions
2. To get knowledge about various types of jump instructions and how it works.

Introduction:

There are two types of jump instruction. They are Conditional jump instruction and unconditional jump instruction.

JMP operator is used as unconditional jump instruction. On the other hand JE, JA, JB, JNE, JZ, JNZ, JAE, JBE etc. are used as Conditional jump instruction. Conditional jump instructions actually checks the carry flags and on the result of comp

comparing two operands, then the carry flag is set on reset.

If two register or memory stores same size data and then if they are equal then 'cmp' operator sets $ZF = 1$ and $CF = 0$ and 'je' and 'jz' uses the flags.

If source register is greater then $ZF = 0$, $CF = 0$ and 'ja' and 'jg' operators uses these flags.

And finally if source is smaller then the other then the $CF = 1$ and $SF = 1$ and the 'jb' or 'jbe' register uses the flags and does conditional jump to skip some lines.

Apparatus Required:

Emu 8086, laptop

Methodology:

Code:

org 100h

mov ax, 1000h ; ax is initialized with 1000h

mov bx, 2000h ; bx is initialized with 2000h

cmp ax, bx ; Comparing the values stored in
ax and bx

je Equal ; If the values are equal that
means $ZF = 1$, then this statement
executes and the program
control goes to "Equal".

jne Not-Equal ; If $ZF = 0$; then this statements
executes and jump to "Not-Equal".

Equal:

mov dx, 0000h ; If the values are equal

ret

then we simply make $dx = 0$
and then return the
program.

Not- Equal:

Ja Above ; If not equal then check the source destination is greater or not. If greater then goes to 'Above'.

Jb Below ; If the source is ^{greater} ~~smaller~~ than the destination then goes to 'Below'.

Above:

mov dx, 0002h; If the destination is greater
ret then dx is initialized/
assigned with 0002h.

Below:

mov dx, 0001h; If the destination is smaller
ret then dx is assigned with
0001h.

Result and discussion:

The program was performed perfectly. The experiment was done perfectly because we used various operands. Everytimes we performed experiments and found

expected values, so eventually we can ensure that ~~the~~ our experiment was perfect.

Conclusion: From the experiment, we learnt how to perform jump instruction in assembly language which are very necessary and basic operator for almost each and every experiments and program.

Reference:

1. Microprocessors and Interfacing - by D.V. Hall
2. emu 8086 / documentation / index. html.