Flow Control Instructions

Course Code: COE 3205

Course Title: Computer Organization & Architecture

Dept. of Computer Science
Faculty of Science and Technology

Lab No:	**7**	**Week No:**	**8**	**Semester:**	**Spring 21-22**
Lecturer:	*sohidul@aiub.edu*				

Lab Outline

1. Decision making and repeating statement
2. Level

Jump

- Jump instructions transfers control to another program
- The transfers can be unconditional or condition
- Depends on a particular combination of status flags settings

Unconditional Jump

- **Does not depend on any condition**

- **Syntax**
- Jump destination_level
- Example : jmp level1

Conditional Jump Conti…

- **Depends on a particular combination of status flags settings**

- **Syntax**
- Jump destination_level
- Example : **jnz** level1

Conditional Jump

- There are three types of conditional jumps
  - Signed Conditional Jumps
  - Unsigned Conditional Jumps
  - Single-Flag Jumps

Signed Conditional Jump

JG or JNLE	• Jump if Greater than • Jump if Not Less than or Equal to	ZF = 0 and SF = OF
JGE or JNL	• Jump if Greater than or Equal to • Jump if Not less than or Equal to	SF = OF
JL or JNGE	• Jump if less than • Jump if not greater than or equal	SF<>OF
JLE or JNG	• Jump if less than or Equal • Jump if not greater than	ZF = 1 or SF<> OF

Unsigned Conditional Jump

JA or JNBE	• Jump if Above • Jump if Not Below or Equal to	ZF = 0 and CF = 0
JAE or JNB	• Jump if Above or Equal to • Jump if Not Below	CF = 0
JB or JNAE	• Jump if Below • Jump if not Above or Equal	CF = 1
JBE or JNA	• Jump if Below or Equal • Jump if Not Above	CF=1 or ZF = 1

Single-Flag Conditional Jump

JE or JZ	• Jump if Equal • Jump if equal to Zero	ZF = 1
JNE or JNZ	• Jump if Not Equal • Jump if Not Zero	ZF = 0
JC	• Jump if Carry	CF = 1
JNC	• Jump if no Carry	CF=0
JO	• Jump if Overflow	CF=1 or ZF = 1
JNO	• Jump if No Overflow	OF=1
JS	Jump if Sign Negative	SF = 1
JNS	Jump if Non-Negative Sign	SF =0
JP/JPE	Jump if Parity Even	PF=1
JNP/JPO	Jump if parity Odd	PF=0

Label

- **Jump** instruction has a general format **jxx label** where **label** is a facility offered by the assembler
- **Labels are used with jump and loop statements to refer another instruction**
- Labels are needed to refer another instruction

Label

- These labels are converted by the assembler to exact address where the program is to continue.
  - Labels must start with a letter and can contain thereafter letters, numbers and underscores (_).
  - Spaces and punctuation marks are not permitted
  - Avoid using keywords in labels
  - Once_again, Next, Name34, this_37 are permitted as labels
  - 3rdday, tues+wed and semi;colons are not permitted as labels.

Label

Example

```
    Jmp Exit

Exit:
    Mov ah, 4ch
    Int 21h
```

Lab Tasks

Task: 1

- Write an assembly program that non-stop prints Hello World.
  Hints: Use unconditional **jmp** and **level** instructions.

Sample Output

Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
.....
.....
Hello world

Lab Tasks

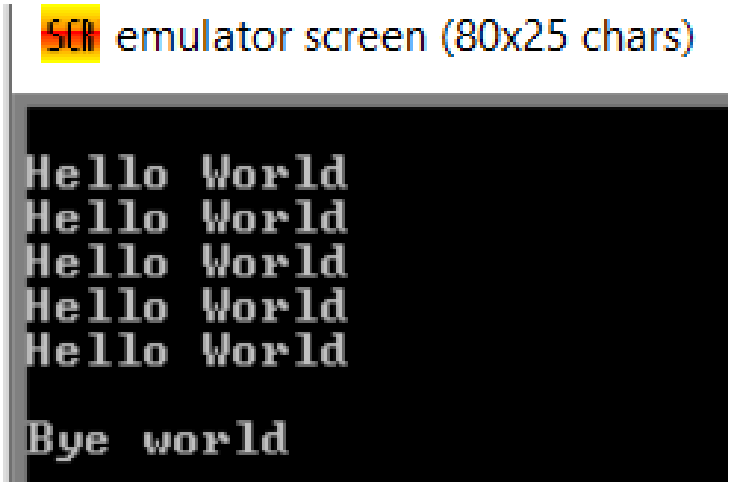
Task: 2

- Write an assembly program that prints Hello World five times and then prints Bye world. **Hints**: Use unconditional **CMP,** conditional **JE, JNE** instruction**.**

Sample Output

Hello world
Hello world
Hello world
Hello world
Hello world

Bye world



Lab Tasks

Task: 3

Read an integer from user. Check whether the number is positive or negative. Hints: **JMP, JL, JG** instructions

Sample output

Enter a number: 1

Positive

Enter a number: -1

Negative

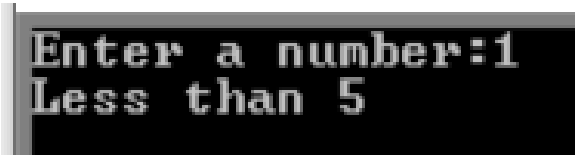
Lab Tasks

Task: 4

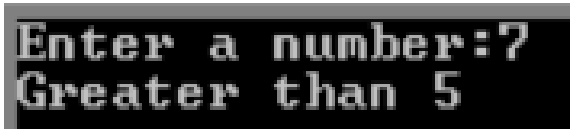
- Suppose that **CL** contains the value of **5**. Take an integer from user. Compare the value with **CL**. And show whether the user input is less than, greater than and equal to CL. **Hints:** use CMP, JL, JG, JE

Sample output

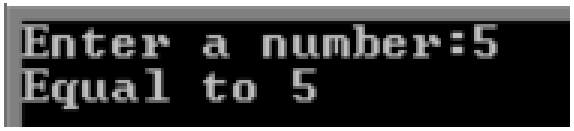
Enter a number: 1
Less than 5



Enter a number: 7
Greater than 5



Enter a number: 5
Equal to 5



Lab Tasks

Task: 5

Read a character and display it **50** times on the next line. **Hints**: use **DEC and JNZ** instructions and

Sample Output

Enter a character: d

ddddddddddddddddddddddddddddddddddddddddddddd
ddddd

Thank you.

Lab Tasks

Task: 6

Write a program to check password using Assembly Programming.
Suppose the password is **mypassword**

Sample output

Enter your password: **mypassword**
Password Matched

Enter your password: password
Password Not Matched

Books

- Assembly Language Programing and Organization of the IBM PC

Ytha Yu
Charles Marut

References