Department of
Computer Science and Engineering

Title: Implementation of conditional statement using assembly language.

Microprocessors and microcontrollers
CSE 304

Green University of Bangladesh

1 Objective(s)

- The main objective of this topic is to implement basic conditional statements in assembly language.

2 Problem analysis

To execute conditional operations, CPU checks the state of flag registers. Flag registers basically reflect what was the result of the last thing that CPU did. Decision is being made based on the status of various flag registers. Based on the flag register we can execute a certain portion or can execute different portion of our program. Just like if-else condition in high level language. This jump can be either conditional or unconditional. Unconditional jump may take place by JMP instruction. For conditional branching, CMP instruction can be used.

Syntax: CMP destination, source
Example:
CMP DX, 00 ; Compare the DX value with zero
JE L7 ; If yes, then jump to label L7
.
.
L7:

For conditional branching, various instructions are used other than JE. JNE/JNZ stands for Jump not Equal or Jump Not Zero, JG/JNLE stands for Jump Greater or Jump Not Less/Equal, JGE/JNL stands for Jump Greater/Equal or Jump Not Less etc.Follow your reference book for more branching instructions. Table 1. shows different jump instructions that are used in assembly language. It also depicts description of instructions and corresponding flag registers.

In high-level languages, branching structures enable a program to take different paths, depending on conditions. In this section, we'll look at three structures.

IF-THEN:

The IF-THEN structure may be expressed in pseudocode as follows:

```
IF condition is true
THEN
        execute true-branch statements
END_IF
```

Example:

```
IF AX < 0
THEN
        replace AX by -AX
END IF
```

IF-THEN-ELSE:

The IF-THEN-ELSE structure may be expressed in pseudocode as follows:

```
IF condition is true
THEN
        execute true-branch statements
ELSE
        execute false-branch statements
END_IF
```

Example:

```
IF AL < BL
THEN
        display the character in AL
ELSE
        display the character in BL
END IF
```

CASE:

Table 1: Jump Instructions

Opcode	Description	Flag
JA	Jump IF above	CF=0 and ZF=0
JAE	Jump IF above or equal	CF=0
JB	Jump IF below	CF
JBE	Jump IF below or equal	CF or ZF
JC	Jump IF carry	CF
JE	Jump IF equal	ZF
JG/JGE	Jump IF greater/Greater than equal	ZF=0 and SF=OF
JL/JLE	Jump IF Less/Less than equal	SF!=OF

A CASE is a multiway branch structure that tests a register, variable, or expression for particular values or a range of values. The general form is as follows:

CASE expression:
values l: statements_l
values_2: statements_2
values_n: statements_ n
END_CASE

Example:

CASE AX
<0: put -1 in BX
=0: put 0 in BX
>O: put 1 in BX
END_CASE

3 Problems

3.1 Printing two characters in alphabetic order using assembly language

```
1  .MODEL SMALL
2   .STACK 100H
3
4   .DATA
5     msg_1   DB  'Enter the first capital letter : $';message 1
6     msg_2   DB  'Enter the second capital letter : $';message 2
7     msg_3   DB  'The given capital letters in alphabetical order are : $';message
          3
8     NEXT  DB  0DH,0AH,"$"
9
10  .CODE
11    MAIN PROC
12      MOV AX, @DATA
13      MOV DS, AX
14
15      MOV AH, 9                    ; set string output function
16
17      LEA DX, NEXT                 ; Next line
18      INT 21H
19
20      LEA DX, msg_1                ; display message 1
21      INT 21H
22
23      MOV AH, 1                    ; set input function
24      INT 21H                      ; read first character
25
26      MOV BL, AL                   ; save first character into BL
```

```
27
28      MOV AH, 9                       ; set string output function
29
30      LEA DX, NEXT                    ; new line
31      INT 21H
32
33      LEA DX, msg_2                   ; message 2
34      INT 21H
35
36      MOV AH, 1                       ; set input function
37      INT 21H                         ; read second character
38
39      MOV BH, AL                      ; save second character into BH
40
41      MOV AH, 9                       ; set string output function
42
43      LEA DX, NEXT                    ; next line
44      INT 21H
45
46      LEA DX, msg_3                    ; message3
47      INT 21H
48
49      MOV AH, 2                       ; set output function
50
51      CMP BL, BH
52
53      JAE Larger_
54        MOV DL, BL
55        INT 21H
56
57        MOV DL, BH
58        INT 21H
59
60        JMP _END
61
62      Larger_:
63        MOV DL, BH
64        INT 21H
65
66        MOV DL, BL
67        INT 21H
68
69      _END:
70
71      MOV AH, 4CH
72      INT 21H
73    MAIN ENDP
74  END MAIN
```

Output of the program is given below.

```
Enter the first capital letter: H
Enter the second capital letter: A
The given capital letters in alphabetical order
are :AH
```

4 Discussion & Conclusion

Based on the focused objective(s) to understand about the conditional statements in assembly language and the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

5 Lab Task (Please implement yourself and show the output to the instructor)

1. Print two characters in reverse alphabetic order using assembly language.
2. Take a number from user, print whether the given number is odd or even.
3. Find out the largest number between two numbers using assembly language.
4. Write an assembly program that checks if a given number is positive, negative, or zero and displays an appropriate message.

6 Lab Exercise (Submit as a report)

- Take a character input from user, check whether the given character is vowel or not (a,e,i,o,u).
- Take input from user, you have to find out whether the given input is alphabet or digit.
- Take a number input from user, check whether the given number is divisible by 5 or not.
- Write an assembly program that converts a hexadecimal digit to its decimal equivalent. The program prompts the user to enter a hexadecimal digit and then displays its decimal equivalent. If the user enters an illegal character (not '0' to '9' or 'A' to 'F'), the program provides an error message and allows the user to try again. The user can choose to continue or exit the program after each successful conversion.

7 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.