Branching Structures and Algorithm conversion to assembly language

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:**	
Lecturer:	*Name & email*				

Lab Outline

1. **Branching structures enable a program to take different paths, depending on conditions.**
2. **Here, We will look at three structures.**
   - **IF-THEN**
   - **IF-THEN-ELSE**
   - **CASE**
3. **Algorithm conversion to assembly language**

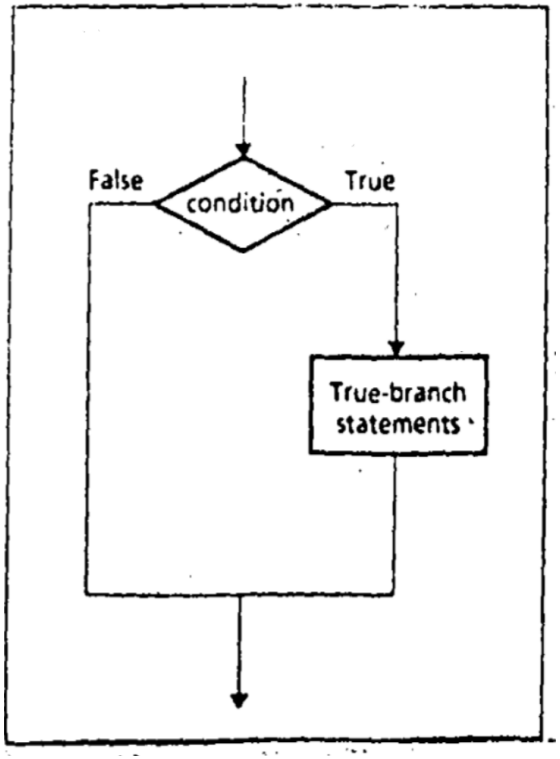
IF-THEN

IF condition is true.

THEN

execute true-branch statements

END_IF



A Pseudo Code , Algorithm and Code for IF-THEN

- The condition is an expression that is either **true** or **false**.
- If It is **true**, the **true-branch** statements are executed.
- If It is **false**, nothing is done, and the program goes on to whatever follows.
- **Example**: to Replace a number in AX by its absolute value…

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

```
        CMP AX, 0
        JNL END_IF
        NEG AX
END_IF:
```

IF-THEN-ELSE

IF condition is true

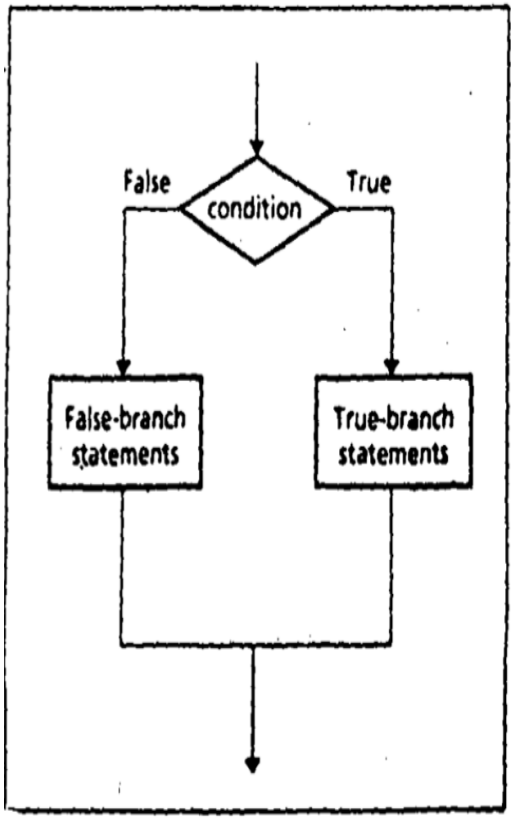
THEN

execute true-branch statements

ELSE

execute false-branch statements

END_IF



IF-THEN-ELSE

Example: Suppose AL and BL contain extended ASCII characters. Display the one that comes

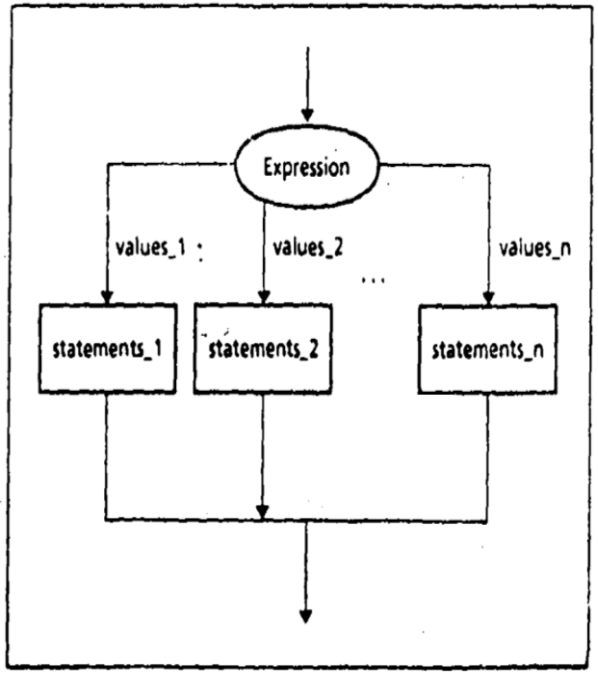
```
IF AL <= BL
        THEN
                Display the
character in AL
        ELSE
                Display the
character in BL
END IF
```

```
        MOV AH,2
        CMP AL,BL    ;AL<=BL
?
        JNBE ELSE_
        MOV DL,AL
        JMP DISPLAY
ELSE_:
        MOV DL,BL
DISPLAY:
        INT 2Ih
```

CASE

A CASE is a multi-way branch structure that tests a register, variable, or expression for particular values or a range of values.

CASE Expression
    Values_1: Statement_1
    Values_2: Statement_2
    …
    Values_n: Statement_n
END_CASE



CASE

Example: If AX contains a negative number, put -1 in BX; if AX contains 0, put 0 in BX; and if AX contains a positive number, put 1 in BX.

```
CASE AX
      <0 : put -1 in BX
      =0 : put  0 in BX
      >0 : put +l in BX
END_CASE
```

```
CMP AX,O
          JL NEGATIVE
          JE ZERO
          JG POSITIVE
NEGATIVE:
          MOV BX,-1
          JMP END_CASE
ZERO:
          MOV BX,0
          JMP END_CASE
POSITIVE:
          MOV BX, I
END_CASE:
```

Lab Tasks

Task: 1

Write a program to convert following code into assembly program.

```
IF AX < 0
THEN
PUT -1 IN BX
END IF
```

Lab Tasks

Task: 2

Write a program to convert following code into assembly progam.

```
IF "AL < 0
THEN
put FFh in AH'
ELSE
put 0 in AH

END_IF
```

Lab Tasks

Task: 3

Write a program to convert following code into assembly program.

Suppose DL contains the ASCII code of a character.

```
(IF DL >= "A") AND (DL <= 'Z')
 THEN
  display DL
END_IF
```

Lab Tasks

Task: 4

Write a program to convert following code into assembly program.

```
IF AX < BX
 THEN
  IF BX < CX
   THEN
    put 0 in AX
   ELSE
    put 0 in BX
  END_IF
END_IF
```

Lab Tasks

Task: 5

Write a program to convert following code into assembly program.

```
IF (AX < BX) OR (BX < CX)
 THEN
  put 0 in DX
 ELSE
  put 1 in DX
END_IF
```

Lab Tasks

Task: 6

Write a program to convert following code into assembly program.

```
IF AX < BX
 THEN
  put 0 in AX
 ELSE
  IF BX < CX
   THEN
    put 0 in BX
   ELSE
    put 0 in CX
  END_IF
END_IF
```

Lab Tasks

Task: 7

Write a program to display "**o**" if AL contains **I** or **3**; if AL contains **2** or 4, display "**e**"..

Books

- Assembly Language Programing and Organization of the IBM PC

Ytha Yu
Charles Marut

References