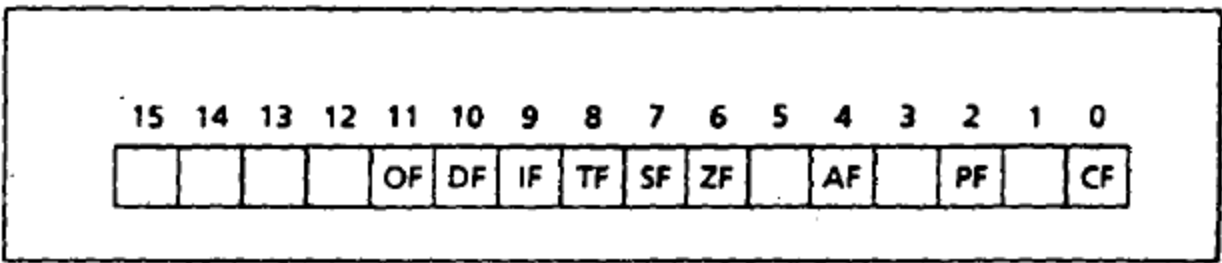




Assembly : Control Flow (Jump, If Else, Loops & Logical)

Author	Md. Zarif Ul Alam
Date Created	@March 8, 2021 9:46 PM
Tag	CSE315 Microprocessors Microcontrollers and Embedded Systems

Flags Register



Flags Register flags

Flag	Intel Mnemonic
Overflow	OF
Sign	SF
Zero	ZF
Auxiliary Carry	AF
Parity	PF
Carry	CF

OF

- 1 → signed overflow
 - add 2 numbers of same sign , result different sign
 - carry in and carry out of MSB not same

CF

- 1 → unsigned overflow
- not affected by *INC* or *DEC*

ZF

- 1 → result 0

SF

- 1 → MSB 1

PF

- 1 → low byte of result has even number of 1 (even parity)

LOOP (using CX)

```

.MODEL SMALL
.CODE
.STARTUP
MOV AH, 2 ; display char function
MOV CX, 256 ; no. of chars to display
MOV DL, 0 ; DL has ASCII code for null char
PRINT_LOOP:
INT 21H ; display a char
INC DL ; increment ASCII code
DEC CX ; decrement counter
JNZ PRINT_LOOP ; keep going if CX not 0
.EXIT
END
```

JNZ → jump if Z = 0

FOR LOOP

Write a program to display a row of 80 stars ‘*’

```

FOR 80 times DO
    display ‘*’
END_FOR
```

```

MOV CX, 80 ; number of ‘*’ to display
MOV AH, 2 ; char display function
MOV DL, ‘*’ ; char to display
TOP:
INT 21h ; display a star
LOOP TOP ; repeat 80 times
```

Corner Case

Caution!

- ‘For loop’ implemented using LOOP is executed at least once
- If CX contains 0 when loop is entered, the LOOP instruction will decrement CX to FFFFh
- The loop will be executed 65535 more times
- JCXZ (jump if CX is zero) may be used

```
TOP:      JCXZ    SKIP
          ; body of the loop
          LOOP   TOP
SKIP:
```

WHILE LOOP

Write a program to count the characters in an input line

```
Initialize count to 0
Read a character
WHILE character <> carriage_return DO
    count = count + 1
    read a character
END_WHILE
```

```
MOV     DX, 0      ; DX counts the characters
MOV     AH, 1      ; read char function
INT     21h        ; read a char in AL
WHILE_:
CMP     AL, 0DH    ; CR?
JE      END_WHILE
INC     DX
INT     21h
JMP     WHILE_
END_WHILE:
```

REPEAT LOOP

Write a program to read characters until a blank/space is read

```
REPEAT
    read a character
UNTIL character is a blank
```

```
MOV    AH, 1    ; read char function
REPEAT:
    INT    21h    ; read a char in AL
    CMP    AL, ' ' ; a blank?
    JNE    REPEAT ; no, keep reading
```

Conditional JUMP [+126 byte || -127 byte]

- Signed Jump

Signed Conditional Jumps

Symbol	Description	Condition for jumps
JG/JNLE	Jump if greater than Jump if not less than or equal to	ZF = 0 and SF = OF
JGE/JNL	Jump if greater than or equal to Jump if not less than	SF = OF
JL/JNGE	Jump if less than Jump if not greater than or equal to	SF <> OF
JLE/JNG	Jump if less than or equal Jump if not greater than	ZF = 1 or SF <> OF

- Unsigned Jump

Unsigned Conditional Jumps

Symbol	Description	Condition for jumps
JA/JNBE	Jump if above Jump if not below or equal	CF = 0 and ZF = 0
JAE/JNB	Jump if above or equal Jump if not below	CF = 0
JB/JNAE	Jump if below Jump if not above or equal	CF = 1
JBE/JNA	Jump if below or equal Jump if not above	CF = 1 or ZF = 1

- Single Flag Jump

Single-Flag Jumps

Symbol	Description	Condition for jumps
JE/JZ	Jump if equal Jump if equal to zero	ZF = 1
JNE/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	OF = 1
JNO	Jump if no overflow	OF = 0
JS	Jump if sign negative	SF = 1
JNS	Jump if nonnegative sign	SF = 0
JP/JPE	Jump if parity even	PF = 1
JNP/JPO	Jump if parity odd	PF = 0

Unconditional JUMP

no range restriction

IF

Replace the number in AX by its absolute value.

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

```

        CMP    AX, 0    ; AX < 0?
        JNL    END_IF
        NEG    AX
END_IF:
```

IF ELSE

Suppose AL and BL contains ASCII characters.
Display the one that comes first in the character sequence

```
IF AL <= BL THEN
    display the character in AL
ELSE
    display the character in BL
END_IF
```

```
MOV     AH, 2      ; prepare to display
CMP     AL, BL     ; AL <= BL?
JNBE    ELSE_

MOV     DL, AL
JMP     DISPLAY ←
ELSE_:
MOV     DL, BL
DISPLAY:
INT     21h
END_IF:
```

CASE

Example 1

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

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

```

CMP     AX, 0      ; test AX
JL      NEGATIVE   ; AX < 0
JE      ZERO       ; AX = 0
JG      POSITIVE   ; AX > 0
NEGATIVE:
MOV     BX, -1     ; put -1 in BX
JMP     END_CASE   ; and exit
ZERO:
MOV     BX, 0      ; put 0 in BX
JMP     END_CASE   ; and exit
POSITIVE:
MOV     BX, 1      ; put 1 in BX
END_CASE:
```

Example 2

If AL contains 1 or 3, display “o” for odd;
If AL contains 2 or 4, display “e” for even;

```

    CMP     AL, 1      ; AL = 1?
    JE      ODD        ; yes, display 'o'
    CMP     AL, 3      ; AL = 3?
    JE      ODD        ; yes, display 'o'
    CMP     AL, 2      ; AL = 2?
    JE      EVEN       ; yes, display 'e'
    CMP     AL, 4      ; AL = 4?
    JE      EVEN       ; yes, display 'e'
    JMP     END_CASE

ODD:
    MOV     DL, 'o'     ; get 'o'
    JMP     DISPLAY    ; go to display

EVEN:
    MOV     DL, 'e'     ; get 'e'

DISPLAY:
    MOV     AH, 2       ; char display function
    INT     21h         ; display character

END_CASE
```

```

CASE  AL
      1, 3: display 'o'
      2, 4: display 'e'
END_CASE
```

AND

Read a character, and if it’s an uppercase letter, display it.

```

Read a character into AL
IF ('A' <= character ) and (character <= 'Z') THEN
    display the character
END_IF
```

```

    MOV     AH, 1       ; read character function
    INT     21h         ; char in AL

    CMP     AL, 'A'     ; char >= 'A'
    JNGE    END_IF      ; no, exit
    CMP     AL, 'Z'     ; char <= 'Z'
    JNLE    END_IF      ; no, exit

    MOV     DL, AL      ; get char
    MOV     AH, 2       ; display character function
    INT     21h         ; display the character

END_IF:
```

OR

Read a character, and if it's 'y' or 'Y', display it; otherwise, terminate the program

```
Read a character into AL
IF (character = 'y') or (character = 'Y') THEN
    display the character
ELSE
    terminate the program
END_IF
```

```
MOV    AH, 1    ; read character function
INT     21h     ; char in AL

CMP     AL, 'Y' ; char = 'Y'
JE      THEN    ; yes, display the char
CMP     AL, 'y' ; char = 'y'
JE      THEN    ; yes, display the char
JMP     ELSE_

THEN:
MOV     DL, AL  ; get the char
MOV     AH, 2   ; display character function
INT     21h     ; display the character

ELSE_:
```