

Flow Control Instructions

Anika Sayara
anikasayara@outlook.com

Conditional Jumps

Anika Sayara
anikasayara@outlook.com

Jump_instruction destination_label

Conditional Jump

- If the condition for the jump is true, *the next instruction to be executed is the one at destination_label*
- If the condition is false, *the instruction immediately following the jump is done next*

Range of Conditional
Jump

The destination label must *precede or follow the jump instruction no more than 126 bytes*

The way CPU implements a Conditional Jump

- If the conditions for the jump instruction, that is, the combination of status flag settings are *true*, the *CPU adjusts the IP to point to the destination label* so that the instruction at this label will be executed next
- If the jump condition is *false*, then *IP is not altered*

Types of Conditional Jump

There are three categories of conditional jumps-

- Signed conditional jumps: for *signed interpretation*
- Unsigned conditional jumps: for *unsigned interpretation*
- Single-flag jumps: operate on *settings of individual flags*

Signed Conditional Jump

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

Unsigned Conditional Jump

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

Single Flag Conditional Jump

Symbol	Description	Condition for Jumps
JE/JZ	Jump if equal	ZF=1
	Jump if equal to or zero	
JNE/JNZ	Jump if not equal	ZF=0
	Jump if not zero	
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

The CMP Instruction

Anika Sayara
anikasayara@outlook.com

CMP Instruction

CMP destination, source

- Does the *compare by subtracting the source from the destination*
- The result is *not stored*
- *Only the flags are affected*
- The operands of **CMP** *may not both be memory locations*
- Destination operand *may not be a constant*

EXAMPLE

Suppose AX and BX *contain signed numbers*. Write some code to put the biggest one in CX

Branching Structures

Anika Sayara
anikasayara@outlook.com

Replace the number in AX by its absolute value

Pseudocode Algorithm

IF-THEN

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

IF-THEN-ELSE

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

Pseudocode Algorithm

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

unsigned jumps should be used when comparing extended ASCII character codes (80H to FFH)

CASE

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

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

AND

Read a character and if it is an uppercase letter display it.

Pseudocode Algorithm
Read a character IF ('A'>=character) and (character<='Z') THEN display character END_IF

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

OR

Pseudocode Algorithm
<pre>Read a character IF (character='y') OR (character='Y') THEN display it ELSE terminate the program END_IF</pre>

Loop Instruction

Anika Sayara
anikasayara@outlook.com

LOOP destination_label

LOOP Instruction

- The counter for the loop is the register *CX which is initialized to loop_count*
- Execution of LOOP instruction *causes CX to be decremented automatically.*

FOR LOOP

Write a count controlled loop and display a row of 80 stars.

Pseudocode Algorithm
FOR 80 times DO display ‘*’ END_FOR

JCXZ (JUMP IF CX IS ZERO)

JCXZ destination_label

- FOR loop, as implemented with LOOP Instruction, is executed at least once. Actually, if CX contains 0 when the loop is entered, the LOOP instruction causes CX to be decremented to FFFFH. To prevent this, the *instruction JCXZ may be used before the loop*
- *If CX contains 0, control transfers to the destination label.*

JCXZ (JUMP IF CX IS ZERO)

JCXZ destination_label

JCXZ SKIP

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