

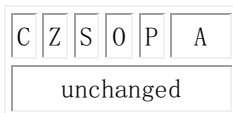
JMP label
4-byte address

Unconditional Jump. Transfers control to another part of the program. 4-byte address may be entered in this form: 1234h:5678h, first value is a segment second value is an offset.

Algorithm:
always jump

Example:
include 'emu8086.inc'

ORG 100h
MOV AL, 5
JMP labell ;jump over 2 lines!
PRINT 'Not Jumped!'
MOV AL, 0
labell:
PRINT 'Got Here!'
RET



JE label

Short Jump if first operand is Equal to second operand (as set by CMP instruction). Signed/Unsigned.

Algorithm:
if ZF = 1 then jump

Example:
include 'emu8086.inc'

ORG 100h
MOV AL, 5
CMP AL, 5
JE labell
PRINT 'AL is not equal to 5.'
JMP exit
labell:
PRINT 'AL is equal to 5.'
exit:
RET



JGE label

Short Jump if first operand is Greater or Equal to second operand (as set by CMP instruction). Signed.

Algorithm:

if SF = OF then jump

Example:

```
include 'emu8086.inc'

ORG 100h
MOV AL, 2
CMP AL, -5
JGE labell
PRINT 'AL < -5'
JMP exit
labell:
PRINT 'AL >= -5'
exit:
RET
```

C	Z	S	O	P	A
unchanged					

JNS label

Short Jump if Not Signed (if positive). Set by CMP, SUB, ADD, TEST, AND, OR, XOR instructions.

Algorithm:

if SF = 0 then jump

Example:

```
include 'emu8086.inc'

ORG 100h
MOV AL, 00000111b ; AL = 7
OR AL, 0 ; just set flags.
JNS labell
PRINT 'signed.'
JMP exit
labell:
PRINT 'not signed.'
exit:
RET
```

C	Z	S	O	P	A
unchanged					

JNZ label

Short Jump if Not Zero (not equal). Set by CMP, SUB, ADD, TEST, AND, OR, XOR instructions.

Algorithm:

if ZF = 0 then jump

Example:

```
include 'emu8086.inc'

ORG 100h
MOV AL, 00000111b ; AL = 7
OR AL, 0 ; just set flags.
JNZ label1
PRINT 'zero.'
JMP exit
label1:
PRINT 'not zero.'
exit:
RET
```

C	Z	S	O	P	A
unchanged					

JZ label

Short Jump if Zero (equal). Set by CMP, SUB, ADD, TEST, AND, OR, XOR instructions.

Algorithm:

if ZF = 1 then jump

Example:

```
include 'emu8086.inc'

ORG 100h
MOV AL, 5
CMP AL, 5
JZ label1
PRINT 'AL is not equal to 5.'
JMP exit
label1:
PRINT 'AL is equal to 5.'
exit:
RET
```

C	Z	S	O	P	A
unchanged					

LOOP label

Decrease CX, jump to label if CX not zero.

Algorithm:

- $CX = CX - 1$
- if $CX \neq 0$ then
 - jump
- else
 - no jump, continue

Example:

```
include 'emu8086.inc'

ORG 100h
MOV CX, 5
labell:
PRINTN 'loop!'
LOOP labell
RET
```

C	Z	S	O	P	A
unchanged					

LOOPNE label

Decrease CX, jump to label if CX not zero and Not Equal ($ZF = 0$).

Algorithm:

- $CX = CX - 1$
- if $(CX \neq 0)$ and $(ZF = 0)$ then
 - jump
- else
 - no jump, continue

Example:

```
; Loop until '7' is found or 5 times.
include 'emu8086.inc'

ORG 100h
MOV SI, 0
MOV CX, 5
labell:
PUTC '*'
MOV AL, v1[SI]
INC SI ; next byte (SI=SI+1).
CMP AL, 7
LOOPNE labell
RET
v1 db 9, 8, 7, 6, 5
```

C	Z	S	O	P	A
unchanged					

LOOPNZ label

Decrease CX, jump to label if CX not zero and ZF = 0.

Algorithm:

- $CX = CX - 1$
- if $(CX \neq 0)$ and $(ZF = 0)$ then
 - jump
- else
 - no jump, continue

Example:

; Loop until '7' is found,
; or 5 times.

```
include 'emu8086.inc'  
  
ORG 100h  
MOV SI, 0  
MOV CX, 5  
label1:  
  PUTC '*'  
  MOV AL, v1[SI]  
  INC SI ; next byte (SI=SI+1).  
  CMP AL, 7  
  LOOPNZ label1  
  RET  
v1 db 9, 8, 7, 6, 5
```

C	Z	S	O	P	A
unchanged					