Jump

`JMP reg` → absoluter, Register indirekter Sprung

Line	Address	Code	Assembler Code
20	0011	BB 0018r	MOV BX, OFFSET jmpaddr
21	0014	FF E3	JMP BX
22	0016	90	NOP
23	0017	90	NOP
24	0018	E4 04	jmpaddr: IN AL,[0004h]
25	001A	90	NOP

jmpaddr → BX
BX → IP

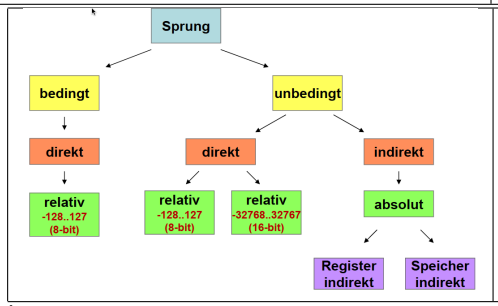
`JMP Speicheroperand` → absoluter Speicher-indirekter Sprung

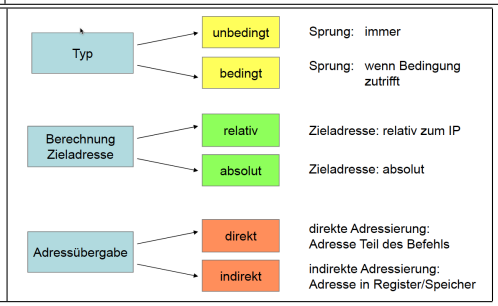
Line	Address	Code	Assembler Code
28	001C	0029r	casetable DW OFFSET case0
29	001E	002Dr	DW OFFSET case1
30	0020	0031r	DW OFFSET case2
31			
32			;BX contains index 0,1 or 2
33	0022	D1 E3	SHL BX,1 ;index * 2
34	0024	2E: FF A7001Cr	JMP casetable[BX]
35	0029	B0 00	case0: MOV AL,0
36	002B	EB 06	JMP endcase
37	002D	B0 64	case1: MOV AL,100
38	002F	EB 02	JMP endcase
39	0031	B0 C8	case2: MOV AL,200
40	0033	EB FE	endcase:

`JMP label` → direkter, relativer Sprung

Line	Address	Code	Assembler Code
13	0005	E4 04	back: IN AL,[0004h]
14	0007	E6 06	OUT [0006h],AL
15	0009	EB 02	JMP forward
16	000B	24 56	AND AL,56h
17	000D	E6 08	forward: OUT [0008h],AL
18	000F	EB F4	JMP back
19	0011	....	

- Forward IP = 000Bh + 0002h = 000Dh
- Back IP = 0011h + FFF4h = 0005h
  (IP, displacement)





Sprungbefehle:	
unbedingt, relativ, direkt	JMP IP = [IP -128; IP + 128] oder IP = [IP – 32'768; IP + 32'767]
unbedingt, absolut, indirekt	JMP IP = [0; 65'535] → reg oder mem
bedingt, relativ, direkt	Jxx IP = [IP -128; IP + 128]

Vergleichsinstruktionen:		Signed	Unsigned	Flagabhängig
CMP	SUB ohne Resultat, aber Flags werden gesetzt	greater	above	J<flag1Letter>
TEST	AND ohne Resultat, aber Flags werden gesetzt	less	below	JN<f1L>