

Opcode Aufbau

Prinzipieller Aufbau

1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte
opcode	addr-mode	displacement/addr		immediate-data	
xxxxxxdw	mod reg r/m	addr-low	addr-high	data-low	data-high

1. Byte

- xxxxxx = Opcode (i. d. R. 6bit)
- Register ist Destination: **d=1** / Register ist Source: **d=0**
- 16bit-Operation: **w=1** / 8bit-Operation: **w=0**

2. Byte

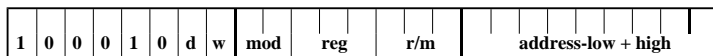
- **mod** Legt fest, ob r/m ein Speicher- oder ein Registeroperand ist
- **reg** Register gemäss Tabelle (Spalten w=0/w=1)
- **r/m** Operand gemäss Tabelle

r/m	mod = 00		mod = 01		mod = 10		mod = 11	
							w=0	w=1
000	[BX+SI]		[BX+SI+disp8]		[BX+SI+disp16]		AL	AX
001	[BX+DI]		[BX+DI+disp8]		[BX+DI+disp16]		CL	CX
010	[BP+SI]		[BP+SI+disp8]		[BP+SI+disp16]		DL	DX
011	[BP+DI]		[BP+DI+disp8]		[BP+DI+disp16]		BL	BX
100	[SI]		[SI+disp8]		[SI+disp16]		AH	SP
101	[DI]		[DI+disp8]		[DI+disp16]		CH	BP
110	direkt		[BP+disp8]		[BP+disp16]		DH	SI
111	[BX]		[BX+disp8]		[BX+disp16]		BH	DI

- Falls in der Adressierung **BP** vorkommt, wird über das Segmentregister **SS** adressiert, in allen anderen Fällen über **DS**.

Aufbau der Move-Befehle

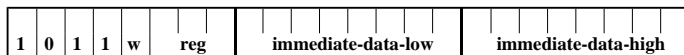
Allgemeine Adressierung



Immediate-Adressierung

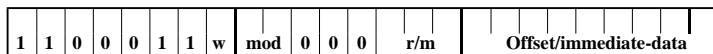
Register <- Konstante (2-3 Byte)

- MOV reg,data



Speicher <- Konstante (3-6 Byte)

- MOV mem,data



Akkumulator-Move-Befehle mit direkter Adressierung

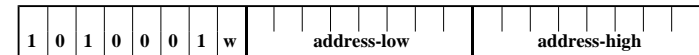
Laden des Akkumulators AL oder AX (4/3 Byte)

- MOV AX,word-var
- MOV AL,byte-var



Speichern des Akkumulators AX oder AL (4/3 Byte)

- MOV word-var,AX
- MOV byte-var,AL



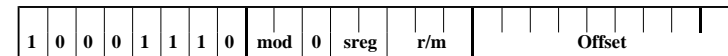
Move-Befehle für Segmentregister

Codierung der Segmentregister

Register	ES	CS	SS	DS
sreg	00	01	10	11

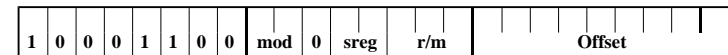
Segmentregister laden (2-4 Byte)

- MOV sreg,reg
- MOV sreg,mem



Segmentregister speichern (2-4 Byte)

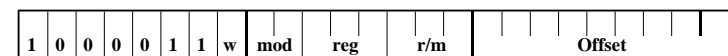
- MOV reg,sreg
- MOV mem,sreg



Aufbau des Exchange-Befehls

Allgemeine Adressierung (2-4 Byte)

- XCHG reg,reg
- XCHG reg,mem
- XCHG mem,reg



Exchange-Befehl für Akkumulator und 16-Bit-Register (1 Byte)

- XCHG AX,reg16
- XCHG reg16,AX

