

# Testfragen und Aufgaben zu Transportbefehlen

1. Entdecken Sie im folgenden Codeabschnitt die fehlerhaften Befehle:

```
1:  mov al,50h           ; ok
2:  mov al,100h          ; Fehler, Maximalwert bei 8-Bit-Registern ist FFh
3:  mov 22,bh            ; Fehler, Ziel kann keine Konstante sein
4:  mov cx,70000o        ; ok
5:  mov cx,70000         ; Fehler, Maximalwert 16-Bit-Reg.: FFFFh=65535d
6:  mov bx, 10001111000000b; ok
7:  mov eax,177FFA001h   ; Fehler, Maximalwert 32-Bit-Reg.: FFFFFFFFh
8:  mov edx, 02A4h       ; ok, Konstante wird erweitert zu 000002A4h
9:  xchg cx,10h          ; Fehler, Austausch mit Konstante nicht möglich
10: mov eax,-1           ; ok, -1 wird vom Assembler im Zweierkomplement eingesetzt
11: mov eax,edi          ; ok
12: mov ah,bl            ; ok
13: mov bx,bl            ; Fehler, Operanden versch. Bitbreite: BX: 16 Bit, BL: 8 Bit
14: xchg eax,bp          ; Fehler, Operanden versch. Bitbreite: EAX: 32 Bit, BP: 16 Bit
15: xchg dx,dx           ; ok, bewirkt aber nichts!
16: mov dl,di            ; Fehler, Operanden versch. Bitbreite: DL: 8 Bit, DI: 16 Bit
17: mov bp,bh            ; Fehler, Operanden versch. Bitbreite: BP: 16 Bit, BH: 8 Bit
18: xchg edi,dl          ; Fehler, Operanden versch. Bitbreite: EDI: 32 Bit, DL: 8 Bit
19: mov esi,dx           ; Fehler, Operanden versch. Bitbreite: ESI: 32 Bit, DX: 16 Bit
20: xchg esi,ebx         ; ok
21: xchg ch,cx           ; Fehler, Operanden versch. Bitbreite: CH: 8 Bit, CX: 16 Bit
22: mov ch,cl            ; ok
```

2. Bestimmen Sie den Inhalt des Registers EAX nach der folgenden Befehlssequenz:

```
mov bx, 7856h
xchg bl,bh
mov ax, 3412h
xchg al,ah
shl eax,16      ; Inhalt von eax um 16 Bit nach links schieben
                ; rechts werden Null-Bits nachgezogen
mov ax,bx
```

Es ist dann EAX=12345678h.

3. Vereinfachen Sie den folgenden Codeabschnitt:

```
1:  mov al,0
2:  mov ah,1      ;Diese beiden Befehle können ersetzt werden durch mov ax,100h

3:  mov ebx,0
4:  mov bx,2800h ; Die beiden Befehle können ersetzt werden durch mov ebx,2800h

5:  mov eax,0
6:  mov al,dl      ; Diese beiden Befehle können ersetzt werden durch movzx eax,dl

7:  xchg ax,ax     ; entspricht No Operation (NOP), ändert kein Register u. kein Flag

8:  mov ax,si
9:  mov si,di
10: mov di,ax      ; soll eine Vertauschung von DI und SI bewirken, besser xchg di,si
```

4. Es soll folgende Aufgabe bewältigt werden:

- Inhalt von AX nach BX bringen
- Inhalt von BX nach CX bringen
- Inhalt von CX nach AX bringen

(Ein Ringtausch). Schreiben sie Befehlssequenzen um die Aufgabe zu lösen:

a) mit mov-Befehlen

```
mov DX,BX      ; Inhalt von BX parken (retten)
                ; geht nur, wenn DX frei
mov BX,AX
mov AX,CX
mov CX,DX
```

b) kürzer! (Wie?)

```
xchg ax,bx
xchg ax,cx
```

5. Schreiben sie jeweils eine Befehlssequenz um folgendes zu bewirken:

- a) höherwertiges Wort von EAX nach DI bringen und niederwertiges Wort von EAX nach SI bringen
- b) CX ins niederwertige Wort von EAX bringen und DX ins höherwertige Wort von EAX bringen
- c) CL ins niederwertige Byte von DX bringen und CH ins höherwertige Byte von DX bringen

Hierbei müssen auch die shift-Befehle shl und shr benutzt werden.

```
; a)
mov si,ax
shr eax,16
mov di,ax
```

```
; b)
mov ax,dx
shl eax,16
mov ax,cx
```

```
; c)
mov dl,cl
mov dh,ch
```

```
; oder besser und kürzer
mov dx,cx
```

6. Setzen Sie mit *einem* Assemblerbefehl das höherwertige Wort von EAX gleich Null, ohne das niederwertige Wort zu verändern!

```
movzx eax,ax
```