

4 Assembler

Übung 12 Eine Null-Adressmaschine führt das nachstehende Assembler-Programm aus (ab dem Zeichen ; stehen Kommentare). Welchen Inhalt (Dezimalwert) hat der Stack nach jeder Instruktion (in Bild eintragen)?

```
push 3 ; 1.
push 5 ; 2.
mpy    ; 3. Multiplikation
push 4 ; 4.
push 2 ; 5.
mpy    ; 6. Multiplikation
add    ; 7. Addition
```

| | | | | | | |
|----|----|----|----|----|----|----|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| 1. | 2. | 3. | 4. | 5. | 6. | 7. |

Übung 13 Programm

Welchen Inhalt hat das Register AX bzw BX an den angegebenen Stellen des folgenden Programms?

```
mov ax,2      ; Wert in AX:
mov bx,1      ; Wert in BX:
add ax,bx     ; Wert in AX:      Wert in BX:
mov cx,3      ; zaehler fuer loop
n: add ax,2
loop n

          ; Wert in AX nach loop:
sub ax,p     ; Wert in AX:
and ax,maske ; Wert in AX:
ret
p         dw 3
maske    dw 4
```

Übung 14 *Anweisungen*

- *Erläutern Sie die folgenden Move-Anweisungen.*

```
mov ax,bx
mov ah,bl
mov ax,123
mov ax,[123]
```

- *Die Assembler-Sprache enthält die beiden Befehle*
 - *jmp Adresse ; jump*
 - *call Adresse ; call subroutine*

In beiden Fällen wird der nächste Befehl von der angegebenen Adresse geholt. Worin unterscheiden sich beide Befehle?

- *Was ist das Status- oder Flag-Register?*

Übung 15 *Fibonacci-Zahlen*

Die Fibonacci²-Zahlen sind durch die Startbedingung $i_1 = 1$, $i_2 = 1$ und die Rekursion $i_n = i_{n-2} + i_{n-1}$ definiert. Damit gilt

$$i_3 = i_1 + i_2 = 1 + 1 = 2$$

$$i_4 = i_2 + i_3 = 1 + 2 = 3$$

$$i_5 = i_3 + i_4 = 2 + 3 = 5$$

$$i_6 = i_4 + i_5 = 3 + 5 = 8$$

und so weiter. Schreiben Sie ein Assembler-Programm, das ausgehend von den Startwerten 1,1 weitere 20 Fibonacci-Zahlen berechnet. Die Werte der Fibonacci-Zahlen sollen nacheinander in der LED-Anzeige des Emulators (siehe zugehöriges Beispielpogramm LED_display_test.asm bei emu8086) angezeigt werden.

²Leonardo Pisano genannt Fibonacci, italienischer Mathematiker, ca. 1170-1250