Grundaufgabe a: Funktionen ermitteln

```
; Programmbereich:
anf:    MOV   EDX,400000H  ;Groessee der Verzoegerung
        MOV   [verzoe],EDX ;Verzoegerung speichern

m1:     MOV   EDI,10    ;EDI=10
        MOV   ESI,OFFSET ziff   ;Adresse von ziff in ESI

m2:     MOV   AL,[ESI+EDI-1]    ;AL=ziff+9
        OUT   0B0H,AL   ;SiebenSegment schreibt AL
        CALL  zeit      ;warten
        DEC   EDI       ;EDI=EDI-1
        JNZ   m2        ;if(EDI!=0) goto m2

        MOV   AL,0FFH   ;AL=255 (dec)
m3:     OUT   5CH,AL    ;LED Reihe links schreiben
        NOT   AL        ;AL negieren
        OUT   5DH,AL    ;LED Reihe rechts schreiben
        CALL  zeit      ;warten
        MOV   BL,AL     ;Inhalt von AL wird noch gebraucht
        IN    AL,59H    ;Tastenreihe rechts lesen auf AL
        BT    EAX,7     ;Bit 7 von EAX in Carry Flag
        MOV   AL,BL     ;AL bekommt alten Wert zurueck
        JC    m1        ;if(m1==0) goto m1
        JMP   m3        ;goto m3 (Loop)

;zeit ist ein Unterprogramm, welches nur Zeit verbrauchen soll:
zeit:   MOV   ECX,[verzoe] ;Lade wartezeit
z1:     DEC   ECX       ;ECX=ECX-1
        JNZ   z1        ;if(ECX!=0) goto z1
        RET             ;zurueck zum Hauptprogramm

; Datenbereich:
verzoe  DD    ?         ;Eine Speicherzelle (Doppelwort)
ziff    DB    3FH,03H,6DH,67H,53H,76H,7EH,23H,7FH,77H
```

anf setzt die Länge der Wartezeit

m1 Lädt Register

m2 Zählt auf Sieben Segment Anzeige

m3 schreibt auf LED Reihe links und invertierend rechts

zeit Verbraucht Zeit nach "verzoe"

Grundaufgabe b: Programmentwurf

einfaches Lauflicht

auf der rechten LED-Reihe soll ein sichtbarer Lichtpunkt von links nach rechts laufen und immer wieder von links beginnen

```
anf:    MOV     EDX,400000H
        MOV     [verzoe],EDX

        MOV     AL, 80H     ;Startwert fuer LED Reihe
lauf:   OUT     5CH, AL     ;Wert auf LED Reihe schreiben
        CALL    zeit        ;warten
        ROR     AL, 1       ;Bits um 1 nach rechts
        JMP     lauf        ;Schleife wiederholen

zeit:   MOV     ECX,[verzoe]
z1:     DEC     ECX
        JNZ     z1
        RET
```

Lauflicht mit Geschwindigkeitsumschalter

das Lauflicht soll durch den linken Schalter zwischen ßchnell"(Schalter oben) und "langsam"(Schalter unten) umschalten

```
anf:    MOV     Al, 80H

lauf:   MOV     EDX, 400000H    ; Wert fuer "langsam"
        MOV     [verzoe], EDX   ;"langsam" in Speicher
        OUT     5CH, AL         ;LED Reihe schreiben
        MOV     BL, AL          ;AL speichern
        IN      AL, 58H         ;Schalter einlesen
        BT      AL, 7           ;7. Bit von AL in Carry Flag
        JNC     langsam         ;Carry Flag = 0, schalter unten
        MOV     EDX, 200000H    ; Wert fuer "schnell"
        MOV     [verzoe], EDX   ;"schnell" in Speicher
        CMC                     ;Carry Flag umschalten (0)

langsam: CALL   zeit            ;warten
        MOV     AL, BL          ;AL aus speicher zurueck
        ROR     AL,1            ;Bits um 1 nach rechts
        JMP     anf             ;Schleife wiederholen

zeit:   MOV     ECX,[verzoe]
z1:     DEC     ECX
        JNZ     z1
        RET
```

Lauflicht verändert Richtung

zusätzlich zum oben implementierten soll die Bewegungsrichtung des Lichtpunktes durch den rechten Schalter der Schalterreihe zwischen "nach linksünd "nach rechts"wechseln.

```
anf:     MOV     Al, 80H
lauf:    MOV     EDX, 400000H    ; Wert fuer "langsam"
         MOV     [verzoe], EDX   ;"langsam" in Speicher
         OUT     5CH, AL         ;LED Reihe schreiben
         MOV     BL, AL          ;AL speichern
         IN      AL, 58H         ;Schalter einlesen
         BT      AL, 7           ;7. Bit von AL in Carry Flag
         JNC     langsam         ;Carry Flag = 0, Schalter unten
         MOV     EDX, 200000H    ; Wert fuer "schnell"
         MOV     [verzoe], EDX   ;"schnell" in Speicher
         CMC                     ;Carry Flag umschalten
langsam: CALL    zeit            ;warten
         MOV     AL, BL          ;AL aus speicher zurueck
         BT      AL, 0           ;0. Bit von AL in Carry Flag
         JNC     rechts          ;Carry Flag = 1; Schalter oben
         ROL     AL,1            ;Bits um 1 nach links
         CMC                     ;Carry Flag umschalten (0)
         JMP     anf             ;Schleife wiederholen
rechts:  ROR     AL, 1           ;Bits um 1 nach rechts
         JMP     anf             ;Schleife wiederholen
zeit:    MOV     ECX,[verzoe]
z1:      DEC     ECX
         JNZ     z1
         RET
```

Lauflicht mit Invertierung

durch drücken einer beliebigen Taste der blauen Tastenreihe wird die Anzeige invertiert, d.h. der Lichtpunkt ist dunkel etc. Invertierung nur solange die Taste gedrückt wird.

```
anf:     MOV     Al, 80H
lauf:    MOV     EDX, 400000H    ; Wert fuer "langsam"
         MOV     [verzoe], EDX   ;"langsam" in Speicher
         MOV     BL, AL          ;Kopie von AL anlegen
         IN      AL, 59H         ;Tastenreihe einlesen
         AND     AL, FFH         ;UND Operation mit FF
         JZ      nopress         ;kein Schalter gedrueckt
         NOT     BL              ;BL invertieren
         MOV     AL, BL          ;AL ueberschreiben
nopress: OUT     5CH, AL         ;LED Reihe schreiben
         IN      AL, 58H         ;Schalter einlesen
         BT      AL, 7           ;7. Bit von AL in Carry Flag
         JNC     langsam         ;Carry Flag = 0, Schalter unten
         MOV     EDX, 200000H    ; Wert fuer "schnell"
         MOV     [verzoe], EDX   ;"schnell" in Speicher
         CMC                     ;Carry Flag umschalten
langsam: CALL    zeit            ;warten
         MOV     AL, BL          ;AL aus speicher zurueck
         BT      AL, 0           ;0. Bit von AL in Carry Flag
         JNC     rechts          ;Carry Flag = 1; Schalter oben
         ROL     AL,1            ;Bits um 1 nach links
         CMC                     ;Carry Flag umschalten (0)
         JMP     anf             ;Schleife wiederholen
rechts:  ROR     AL, 1           ;Bits um 1 nach rechts
```

```
25          JMP     anf             ;Schleife wiederholen
26  zeit:   MOV     ECX,[verzoe]
27  z1:     DEC     ECX
28          JNZ     z1
29          RET
```

Zusatzaufgabe

Erweiterungen des Programms nach eigenen Ideen:

- symetrische LED Reihe zur Mitte
- Sieben Segment zählt 9 Schritte mit

```
1   anf:    MOV     Al, 80H
2           MOV     EDI, 0
3           MOV     ESI, OFFSET ziff
4   lauf:   MOV     EDX, 400000H    ; Wert fuer "langsam"
5           MOV     [verzoe], EDX   ;"langsam" in Speicher
6           MOV     BL, AL          ;Kopie von AL anlegen
7           IN      AL, 59H         ;Tastenreihe einlesen
8           AND     AL, FFH         ;UND Operation mit FF
9           JZ      nopress         ;kein Schalter gedrueckt
10          NOT     BL              ;BL invertieren
11          MOV     AL, BL          ;AL ueberschreiben
12  nopress: OUT    5CH,AL           ;LED Reihe links schreiben
13          NOT     AL              ;AL negieren
14          OUT     5DH,AL          ;LED Reihe rechts schreiben
15          MOV     BH,[ESI+EDI-1]  ;Sieben Segment berechnen
16          OUT     0B0H,BH         ;Sieben Segment schreiben
17          DEC     EDI             ;Sieben Segment runterzaehlen
18          JZ      timer           ;Timer auf 0 setzen
19          IN      AL, 58H         ;Schalter einlesen
20          BT      AL, 7           ;7. Bit von AL in Carry Flag
21          JNC     langsam         ;Carry Flag = 0, Schalter unten
22          MOV     EDX, 200000H    ; Wert fuer "schnell"
23          MOV     [verzoe], EDX   ;"schnell" in Speicher
24          CMC                     ;Carry Flag umschalten
25  langsam: CALL   zeit            ;warten
26          MOV     AL, BL          ;AL aus speicher zurueck
27          BT      AL, 0           ;0. Bit von AL in Carry Flag
28          JNC     rechts          ;Carry Flag = 1; Schalter oben
29          ROL     AL,1            ;Bits um 1 nach links
30          CMC                     ;Carry Flag umschalten (0)
31          JMP     anf             ;Schleife wiederholen
32  rechts: ROR     AL, 1           ;Bits um 1 nach rechts
33          JMP     anf             ;Schleife wiederholen
34  timer:  MOV     BH, 0FFH
35          RET
36  zeit:   MOV     ECX,[verzoe]
37  z1:     DEC     ECX
38          JNZ     z1
39          RET
```