# 384.047 Digitale Systeme Übung – Lösungen

## Weiterführende Übungen 7

### Programm- und Stackanalyse

| Befehl | PC | C | SP | Stack |
|---|---|---|---|---|
| MOV C,3 | 0003h | 03h | 0000h | (leer) |
| CALL ICH | 0009h | | FFFEh | 1x Adresse, nämlich 0005h |
| DEC C | 000Ah | 02h | | |
| JPZ ENDE | 000Dh | | | |
| CALL ICH | 0009h | | FFFCh | 2x Adressen, letzte=0010h |
| DEC C | 000Ah | 01h | | |
| JPZ ENDE | 000Dh | | | |
| CALL ICH | 0009h | | FFFAh | 3x Adressen, letzte=0010h |
| DEC C | 000Ah | 00h | | |
| JPZ ENDE | 0010h | | | |
| RET | 0010h | | FFFCh | 2x Adressen, letzte=0010h |
| RET | 0010h | | FFFEh | 1x Adresse, nämlich 0005h |
| RET | 0005h | | 0000h | (leer) |
| MOV A,C | 0006h | | | |
| OUT 0,A | 0008h | | | |
| HALT | 0009h | | | |

## Programmierung

Eine mögliche (und ganz bestimmt nicht die einzige) Möglichkeit:

```
        (MOV HL,1234h)          ;zB
         ...
         MOV QUATSCH,HL         ;HL im RAM ablegen (LO-Byte zuerst)
         MOV HL,OFFSET QUATSCH  ;HL zeigt nun genau dorthin
         MOV A,[HL]             ;LO-Byte…
         MOV B,A                ;…ins B-Register
         INC HL                 ;eins weiter zeigen
         MOV A,[HL]             ;HI-Byte…
         MOV C,A                ;…ins C-Register
         MOV A,B                ;LO-Byte dorthin, wo…
         MOV [HL],A             ;…vorher das HI-Byte lag
         DEC HL                 ;ein zurueck
         MOV A,C                ;HI-Byte dorthin, wo...
         MOV [HL],A             ;…vorher das LO-Byte lag
         MOV HL,QUATSCH         ;und jetzt alles zurück ins HL!
         ...

ORG 2000h                       ;ab 2000h werden im RAM
QUATSCH: DB 0                   ;zwei Bytes (fuer LO/HI)
         DB 0                   ;reserviert
```

## Ablaufanalyse

LOOP1 wird genau acht Mal durchlaufen (weil beim achten Mal das 1-Bit aus dem Register A herausgeschoben und das Register A dadurch null und in weiterer Folge das ZF gesetzt wird).

LOOP2 wird endlos durchlaufen (das Register A wird niemals null, weil das 1-Bit im Kreis geschoben wird).

## Programmanalyse

Lassen Sie sich nicht dadurch verwirren, dass im Programm ein RET-Befehl ohne CALL-Befehl vorkommt. Das ist zwar extrem schlechter (!) Programmierstil, aber das Programm wird technisch gesehen weder im Assembler, noch im Prozessor (bzw. im Simulator) Probleme verursachen.

- In den Zeilen 2 und 3 wird der Wert 000Dh (=13 dezimal!) ab der Adresse 4000h im RAM abgelegt.
- In Zeile 4 wird der RET-Befehl genau diesen Wert als Rücksprungadresse vorfinden und verwenden (der Stackpointer wurde ja in Zeile 1 auf 4000h und damit auf den Beginn des Wertes gesetzt).
- Das Programm wird somit ab der Adresse 000Dh fortgesetzt. Dort steht der HALT-Befehl.
- Die Endlosschleife wird „übersprungen“.

Die Antwort lautet also: Das Programm endet beim HALT-Befehl.