



Übungsblatt 10

ab 02. 07. 2007

Aufgabe 1 [5 Pkt.]

- a) [2 Pkt.] Was bezeichnen wir als RTL und wofür wird es verwendet?
- b) [3 Pkt.] Beschreibe den Befehl JSR mit der RTL!

Aufgabe 2 [6 Pkt.]

Ein Single Instruction Processor *SIP* kennt nur einen Befehl: *subtract and branch if negative* = *sbn a, b, c*. Der Befehl soll folgendes leisten:

```
M[a] = M[a] - M[b];  
if( M[a] < 0 ) goto c else goto next instruction;
```

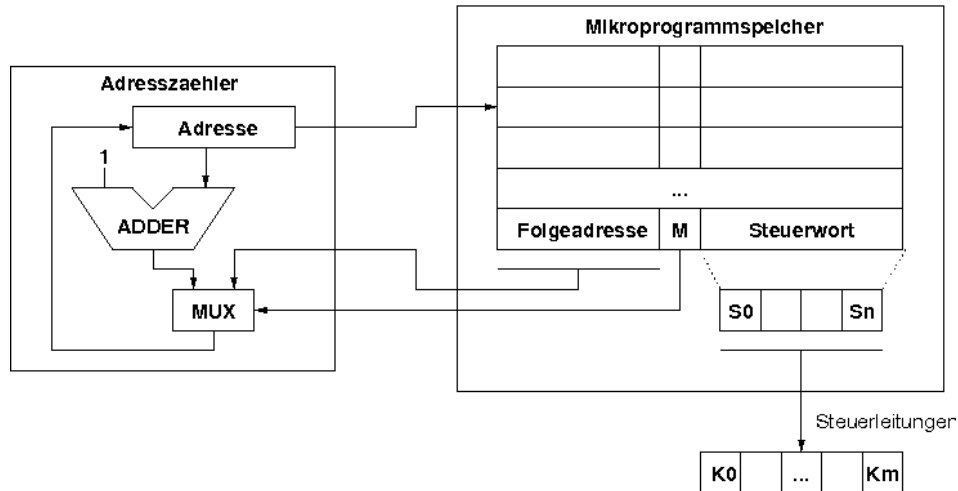
Formulieren Sie ein Maschinenprogramm für den SIP, der das Minimum von drei Speicherstellen in eine vierte schreibt:

```
M[R] = min( M[a], M[b], M[c] )
```

Es soll angenommen werden, dass eine Speicherstelle $M[NULL]$ bei Lesezugriffen immer 1 liefert.

Aufgabe 3 [10 Pkt.]

Die folgende Abbildung stelle ein einfaches Mikroprogramm-Steuerwerk dar. Das Steuerwort ($S_0 - S_n$) besteht aus n Bits und die auf m Steuerleitungen ($K_0 - K_m$) abgebildet werden.



- [3 Pkt.] Geben sie für den Fall $n = m$ an, ob es sich um *horizontale* oder *vertikale* Mikroprogrammierung handelt (*Begründen sie!*)
- [3 Pkt.] Was müsste angepasst werden, um das Steuerwerk auf horizontale/vertikale Mikroprogrammierung zu ändern? Welche Vor-/Nachteile ergeben sich durch eine solche Änderung?
- [4 Pkt.] Wie müsste das Steuerwerk verändert werden, damit auch externe Signale in das verwendete Mikroprogramm einbezogen werden können?

Aufgabe 4 [6 Pkt.]

- [2 Pkt.] Was verstehen wir unter einer *indiziert indirekten Adressierung* und wofür wird sie eingesetzt?
- [4 Pkt.] Welche Auswirkung hat der Assemblerbefehl `LEAX ADDR, PCR` und warum wird er verwendet?

Aufgabe 5 [3 Pkt.]

Worin besteht der Unterschied zwischen `JMP` und `BRA`? Nenne Vor-/Nachteile dieser Unterscheidung!

Aufgabe 6 [8 Pkt.]

- a) [2 Pkt.] Wofür wird ein Stack hauptsächlich verwendet?
- b) [3 Pkt.] Stelle die Befehle `push`, `pop` und `peek` in der RTL dar (*jeweils mit Register A als Parameter*)!
- c) [3 Pkt.] Was beschreibt ein Stackframe, wozu wird er benötigt und wie ist er aufgebaut?

Aufgabe 7 [3 Pkt.]

Aus welchem Grund haben sich *Interrupts* im Vergleich zum *Polling* durchgesetzt (*Nenne Unterschiede sowie Vor- und Nachteile*)?

Aufgabe 8 []

- a) Welche (Unter-)Arten von Abhängigkeiten zwischen Maschinenbefehlen wurden in der Vorlesung betrachtet? Welche Lösungsstrategien sind möglich?
- b) Klassifizieren Sie die Datenabhängigkeiten in folgendem Codefragment:

```
0x1000 sub $2, $7, $3
0x1020 or $5, $5, $4
0x1040 beq $2, $5, 20
0x1060 sw $5, 200($1)
0x1080 or $2, $15, $19 // Einsprungstelle
0x10A0 sw $2, 100($1)
0x10C0 or $9, $15, $7
```

- c) Fügen Sie an geeigneter Stelle NOPs ein und/oder variieren Sie die Reihenfolge der Abarbeitung so, dass ein reibungsloser Ablauf garantiert werden kann. Gehen Sie davon aus, das Hardwareseitig keine Hazard-Behandlung erfolgt.

Aufgabe 9 [8 Pkt.]

Gib für das folgende Assemblerprogramm die verwendeten Kontrollstrukturen an (*jeweils Zeilennummer und Kontrollstruktur*) und stelle das Programm in C-Syntax dar!

```
1      ORG $1000
2  ARRAY  RMB    10      ; 10 Byte Array
3  MAIN   LDA    #10
4         PSHS   A
5         LEAX  ARRAY
6         PSHS   X
7         BSR   SORT
8         LEAS  2, S
9  END    BRA    END
10  SORT  PSHS   U
11         LEAU  , S
12         LEAS -1, S
13         PSHS   X
14  L1    LDA    #1
15         STA  -1, U
16         LDX  2, U
17  L2    PSHS   A
18         LDA  , X+
19         CMPA , X
20         BHS  L3
21         PSHS   X
22         LEAX -1, X
23         PSHS   X
24         BSR  SWITCH
25         LEAS 1, S
26         PULS  X
27         LDA  #0
28         STA  -1, U
29  L3    PULS   A
30         INCA
31         CMPA 3, U
32         BNE  L2
33         LDA  -1, U
34         CMPA #1
35         BNE  L1
36         LEAS -2, U
37         PULS  X
38         LEAS , U
39         PULS  U
40         RTS
41  SWITCH PSHS   U
42         LEAU  , S
43         PSHS   X
44         PSHS   Y
45         LDX  2, U
46         LDY  3, U
47         LDA  , X
48         PSHS   A
49         LDA  , Y
50         STA  , X
51         PULS  A
52         STA  , Y
53         PULS  Y
54         PULS  X
55         PULS  U
56         RTS
57         END
```

