

Grundbegriffe der Informatik

Aufgabenblatt 5

Matr.nr.:

--	--	--	--	--	--	--

Nachname:

--

Vorname:

--

Tutorium:

Nr.

--

Name des Tutors:

--

Ausgabe: 19. November 2014

Abgabe: 28. November 2014, 12:30 Uhr
im GBI-Briefkasten im Untergeschoss
von Gebäude 50.34

Lösungen werden nur korrigiert, wenn sie

- rechtzeitig,
- in Ihrer eigenen Handschrift,
- mit dieser Seite als Deckblatt und
- in der oberen **linken** Ecke zusammengeheftet
abgegeben werden.

Vom Tutor auszufüllen:

erreichte Punkte

Blatt 5:

/ 14 + 0

Blätter 1 – 5:

/ 83 + 13

Aufgabe 5.1 (1 + 2 + 1 + 3 = 7 Punkte)

Es seien a_1 , a_2 und a_3 drei paarweise verschiedene Adressen. Weiter sei c_1 eine nicht-negative ganze Zahl und es sei c_2 eine ganze Zahl derart, dass c_1 und c_2 , deren Summe, deren Differenz und deren Produkt mit 20bit in Zweierkomplementdarstellung darstellbar sind. Im Speicher stehe in Adresse a_1 die Zweierkomplementdarstellung von c_1 und in Adresse a_2 die Zweierkomplementdarstellung von c_2 .

- Schreiben Sie ein Minimalmaschinenprogramm, das die Negation von c_2 in Zweierkomplementdarstellung im Speicher bei Adresse a_2 ablegt.
- Schreiben Sie ein Minimalmaschinenprogramm, das die Summe von c_1 und c_2 in Zweierkomplementdarstellung im Speicher bei Adresse a_3 ablegt. Dabei darf der Maschinenbefehl ADD nur verwendet werden um die Zahlen 1 oder -1 mit einer anderen Zahl zu addieren.
- Schreiben Sie, unter Verwendung der vorangegangenen Programme, ein Minimalmaschinenprogramm, das die Differenz zwischen c_1 und c_2 in Zweierkomplementdarstellung im Speicher bei Adresse a_3 ablegt.
- Schreiben Sie ein Minimalmaschinenprogramm, dass das Produkt von c_1 mit c_2 in Zweierkomplementdarstellung im Speicher bei Adresse a_3 ablegt.

Lösung 5.1

a)

```
LDV a2
NOT
STV a2
LDC 1
ADD a2
STV a2
```

b)

```
while: LDC 0
      NOT
      ADD a1
      STV a1
      JMN end
      LDC 1
      ADD a2
      STV a2
      JMP while
end: HALT
```

Achtung: LDC -1 tut leider nicht das Gewünschte (ist nur 20-Bit Zweierkomplement, wird vorne mit 0000 aufgefüllt). Daher LDC 0, NOT.

- c) Man konkateniere die beiden Programmstücke aus den Teilaufgaben a) und b).
 d)

```

LDC 0
STV a3
while: LDC -1
ADD a1
STV a1
JMN end
LDV a2
ADD a3
STV a3
JMP while
end: HALT

```

Aufgabe 5.2 (3 Punkte)

Es seien a_1 und a_2 zwei verschiedene Adressen. Weiter seien c_1 und c_2 zwei ganze Zahlen, die mit 20bit in Zweierkomplementdarstellung darstellbar sind. Im Speicher stehe in Adresse a_1 die Zweierkomplementdarstellung von c_1 und in Adresse a_2 die Zweierkomplementdarstellung von c_2 . Welche ganze Zahlen in Zweierkomplementdarstellung stehen nach Ausführung des Programms

Adr.	Befehl	Adr.	Befehl	Adr.	Befehl
0000	LDV a_1	0011	LDV a_2	0110	LDV a_1
0001	XOR a_2	0100	XOR a_1	0111	XOR a_2
0010	STV a_1	0101	STV a_2	1000	STV a_1

in den Adressen a_1 und a_2 im Speicher. Gehen Sie davon aus, dass a_1 und a_2 nicht Adressen der obigen Befehlsfolge sind.

Lösung 5.2

In Adresse a_1 steht nach Ausführung des Programms der Speicherinhalt von Adresse a_2 vor Ausführung des Programms und umgekehrt. Das Programm in Pseudocode, wobei wir a_1 und a_2 als Variablen mit Anfangswerten c_1 bzw. c_2 aus $\{0, 1\}^*$ interpretieren und \oplus den bitweisen XOR-Operator bezeichnet, lautet:

$$\begin{aligned}
 a_1 &\leftarrow a_1 \oplus a_2 \\
 a_2 &\leftarrow a_2 \oplus a_1 \\
 a_1 &\leftarrow a_1 \oplus a_2
 \end{aligned}$$

Nach Ausführung des Programms hat a_2 also den Wert $c_2 \oplus (c_1 \oplus c_2)$ und a_1 den Wert $(c_1 \oplus c_2) \oplus (c_2 \oplus (c_1 \oplus c_2))$.

Für Bits x , y und z gilt $x \oplus (y \oplus x) = y$ wie man beispielsweise anhand einer Tabelle mit allen $2^2 = 4$ Fällen sieht. Da der Operator \oplus bitweise operiert, folgt $c_2 \oplus (c_1 \oplus c_2) = c_1$ und $(c_1 \oplus c_2) \oplus (c_2 \oplus (c_1 \oplus c_2)) = c_2$.

Aufgabe 5.3 (2 Punkte)

Es sei w ein Wort über Z_2 der Länge 20. Unter welchen möglichst schwachen Bedingungen haben LDC w und LDV w denselben Effekt?

Lösung 5.3

Unter der Bedingung, dass im Speicher bei Adresse w das Wort $0000 \cdot w$ steht.

Aufgabe 5.4 (2 Punkte)

Beschreiben sie die Befehlsausführungsphase des MIMA-Befehls $JMN\ adr$.

Lösung 5.4

Falls im Akkumulator, bei Interpretation als ganze Zahl in Zweierkomplementdarstellung, ein negativer Wert steht, das heißt, falls das führende Bit der Bitfolge im Akkumulator 1 ist, so wird die Adresse adr in das Befehlsadressregister IAR geladen — das ist jenes Register welches die Adresse des als nächstes auszuführenden Befehls enthält.

Andernfalls wird gar nichts getan (denn die „richtige“ nächste Adresse wurde schon in der Befehlsholphase im IAR abgelegt).