
Lösungsvorschlag zu Übungsblatt 2

Abgabe spätestens bis: 16.11.2020 10:00 Uhr

- Dieses Übungsblatt soll in den in der Übungsgruppe festgelegten Teams abgegeben werden (Einzelabgaben sind erlaubt, falls noch keine Teamzuteilung erfolgt ist).
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

Aufgabe 5+6 (*Maschinenprogramme*)

Für alle Maschinenprogramme in dieser Aufgabe gelten folgende Vorgaben:

- Es sind **ausschließlich** die Maschinenbefehle aus der Vorlesung zulässig.
- Für Adressen sind die Notationen aus Kapitel 2.7 einzuhalten, insbesondere: Der erste Befehl des Programms liegt an Adresse P1, der Datenteil beginnt mit Adresse D1 und der Stack Frame mit Adresse S1.
- Es wird angenommen, dass an Adressen im Datenteil und im Stack, deren Inhalt nicht im Programm mit 0 initialisiert wird, bei Programmstart nur **nicht-negative ganze Zahlen** gespeichert sind (da die in der Vorlesung eingeführten Maschinenbefehle nicht geeignet sind für Berechnungen mit negativen Zahlen).

Beispiel: Folgendes Programm erhöht den an S1 gespeicherten ganzzahligen Wert um 1 und gibt diesen zurück:

P1: INKREMENT S1
P2: RÜCKGABE S1

Dieses Maschinenprogramm realisiert folgende C-Funktion (unter der Annahme, dass S1 die Adresse zum Eingabeparameter a ist):

```
int inkrement(int a)
{
    ++a;
    return a;
}
```

a) (Maschinenprogramm erstellen)

- (i) (*, 2 Minuten) Erstellen Sie ein Maschinenprogramm, das den Wert an S1 verdoppelt und dann zurückgibt.

P1: ADD S1, S1
P2: RÜCKGABE S1

- (ii) (**, 4 Minuten) Erstellen Sie ein Maschinenprogramm, das den Wert an S1 verdreifacht und dann zurückgibt.

P1: INIT S2
P2: ADD S2, S1
P3: ADD S1, S2
P4: ADD S1, S2
P5: RÜCKGABE S1

- (iii) (**, 6 Minuten) Erstellen Sie ein Maschinenprogramm, das 0 zurückgibt, wenn der Wert an S1 gleich Null ist, und sonst 1 zurückgibt.

P1: SPRUNGO P3, S1
P2: RÜCKGABE D2
P3: RÜCKGABE D1
D1: 0
D2: 1

- (iv) (***, 8 Minuten) Erstellen Sie ein Maschinenprogramm, das die an S1 und S2 gespeicherten Werte multipliziert und das Ergebnis zurückgibt.

P1: INIT S3
P2: SPRUNGO P6, S2
P3: ADD S3, S1
P4: DEKREMENT S2
P5: SPRUNG P2
P6: RÜCKGABE S3

b) (Maschinenprogramm zu C-Funktion)

In dieser Teilaufgabe ist jeweils eine C-Funktion vorgegeben. Sie sollen ein Maschinenprogramm erstellen, das diese C-Funktion realisiert. Geben Sie dazu jeweils an, welche Adressen Sie für Eingabeparameter und lokale Variablen benutzen.

- (i) (*, 3 Minuten)

```
int get_odd(int a)
{
    a = a + a;
    --a;
    return a;
}
```

Eingabeparameter a = Adresse S1

P1: ADD S1, S1
P2: DEKREMENT S1
P3: RÜCKGABE S1

(ii) (**, 5 Minuten)

```
int is_double_of_a(int a, int b)
{
    int c = a + a;
    int d = c - b;
    if (d == 0) {
        return 1;
    }
    return 0;
}
```

Eingabeparameter a = Adresse S1

Eingabeparameter b = Adresse S2

Variable c = Adresse S3

Variable d = Adresse S4

P1: INIT S3

P2: ADD S3, S1

P3: ADD S3, S1

P4: INIT S4

P5: ADD S4, S3

P6: SUB S4, S2

P7: SPRUNG0 P9, S4

P8: RÜCKGABE D1

P9: RÜCKGABE D2

D1: 0

D2: 1

(iii) (***, 8 Minuten)

```
int is_odd(int a)
{
    int r = 0;
    while (a != 0) {
        if (r == 0) {
            ++r;
        } else {
            --r;
        }
        --a;
    }
    return r;
}
```

Eingabeparameter a = Adresse S1

Variable r = Adresse S2

P1: INIT S2

P2: SPRUNG0 P9, S1

P3: SPRUNG0 P6, S2

P4: DEKREMENT S2

P5: SPRUNG P7

P6: INKREMENT S2

P7: DEKREMENT S1

P8: SPRUNG P2

P9: RÜCKGABE S2

c) (C-Funktion zu Maschinenprogramm)

In dieser Teilaufgabe ist jeweils ein Maschinenprogramm vorgegeben. Sie sollen eine C-Funktion erstellen, die durch dieses Maschinenprogramm realisiert wird. Geben Sie dazu jeweils an, welche Konstanten, Eingabeparameter und lokale Variablen Sie den verwendeten Adressen zuordnen, und geben Sie der C-Funktion einen Namen, der die von ihr (bzw. die vom Maschinenprogramm) ausgeführte Aufgabe beschreibt.

(i) (*, 4 Minuten)

```
P1: INIT S2
P2: ADD S2,S1
P3: SUB S1,S2
P4: SUB S1,S2
P5: RÜCKGABE S1
```

Eingabeparameter a = Adresse S1

```
int get_neg(int a)
{
    return -a;
}
```

(ii) (**, 6 Minuten)

```
P1: INIT S3
P2: ADD S3,S2
P3: SUB S3,S1
P4: SUB S3,S1
P5: SPRUNG0 P7,S3
P6: RÜCKGABE D1
P7: RÜCKGABE D2
D1: 0
D2: 1
```

Eingabeparameter a = Adresse S1

Eingabeparameter b = Adresse S2

```
int is_double(int a, int b)
{
    return (b == 2 * a);
}
```

(iii) (***, 8 Minuten)

```
P1: INIT S2
P2: SPRUNG0 P6,S1
P3: ADD S2,S1
P4: DEKREMENT S1
P5: SPRUNG P2
P6: RÜCKGABE S2
```

Eingabeparameter a = Adresse S1
Variable b = Adresse S2

```
int gauss_sum(int a)
{
    int b = 0;
    while (a != 0) {
        b = b + a;
        --a;
    }
    return b;
}
```

Aufgabe 7 (*Abarbeitung von Maschinenbefehlen durch die CPU*)

In dieser Aufgabe sollen Sie die Abarbeitung eines Maschinenbefehls durch die CPU mittels

Fetch/Decode/Execute/Write-Back

beschreiben, wobei Sie dabei auf eine detaillierte Darstellung der Kommunikation zwischen Speicherwerk und Steuerwerk **verzichten** (siehe Folien 44 und 45 aus Kapitel 2).

Geben Sie dabei insbesondere alle relevanten Register und deren jeweilige Belegung an.

Betrachten Sie dazu das folgende Maschinenprogramm:

```
P1: INIT S2
P2: SPRUNG0 P6,S1
P3: ADD S2,S1
P4: DEKREMENT S1
P5: SPRUNG P2
P6: RÜCKGABE S2
```

a) (*, 6 Minuten) Beschreiben Sie die Abarbeitung des Maschinenbefehls an Adresse P1

1. **Fetch:**
Hole den Inhalt von SZ P1 über RM nach IR
Setze Befehlszähler PC auf den nächsten Adresswert P2
2. **Decode:**
Erkenne, dass es sich um die INIT-Operation für die SZ S2 handelt
(E) Entfällt
3. **Execute:**
(V) Führe INIT-Operation aus (Ergebnis 0 liegt in AR)
4. **Write-Back:**
(A) Schreibe Ergebnis (0) von AR über WM nach SZ S2

b) (*, 6 Minuten) Beschreiben Sie die Abarbeitung des Maschinenbefehls an Adresse P2

1. **Fetch:**
Hole den Inhalt von SZ P2 über RM nach IR
Setze Befehlszähler PC auf den nächsten Adresswert P3
2. **Decode:**
Erkenne, dass es sich um die SPRUNG0-Operation für die SZ S1 und die Adresse P6 handelt
(E) Lade Inhalt von SZ S1 über RM nach AR
3. **Execute:**
(V) Führe SPRUNG0-Operation aus (Test des Inhalts von S1 auf 0)
Falls Inhalt von S1 gleich 0: Überschreibe PC mit Adresse P6

-
- 4. Write-Back:**
(A) Entfällt
- c) (*, 6 Minuten) Beschreiben Sie die Abarbeitung des Maschinenbefehls an Adresse P3
- 1. Fetch:**
Hole den Inhalt von SZ P3 über RM nach IR
Setze Befehlszähler PC auf den nächsten Adresswert P4
 - 2. Decode:**
Erkenne, dass es sich um die ADD-Operation für die SZ S1 und S2 handelt
(E) Lade Inhalt von SZ S1 über RM nach AR1
(E) Lade Inhalt von SZ S2 über RM nach AR2
 - 3. Execute:**
(V) Führe ADD-Operation aus (Ergebnis liegt in AR3)
 - 4. Write-Back:**
(A) Schreibe Ergebnis von AR3 über WM nach SZ S2

Aufgabe 8 (*Kommunikation zwischen Steuerwerk und Speicherwerk*)

In dieser Aufgabe sollen Sie detailliert die Kommunikation zwischen Steuerwerk und Speicherwerk über Register und Steuersignale bei Schreib- und Lesevorgängen beschreiben (siehe Folie 49).

Geben Sie dabei insbesondere alle relevanten Register und deren jeweilige Belegung an.

Betrachten Sie dazu das folgende Maschinenprogramm:

P1: INIT S2
P2: SPRUNG P6,S1
P3: ADD S2,S1
P4: DEKREMENT S1
P5: SPRUNG P2
P6: RÜCKGABE S2

- a) (**, 8 Minuten) Beschreiben Sie detailliert die Kommunikation zwischen Steuerwerk und Speicherwerk beim Lesen des Befehls im Fetch-Schritt der Abarbeitung des Maschinenbefehls an Adresse P1.
- **Steuerwerk 'schreibt' P1 nach AM**
 - **Steuerwerk setzt D auf Lesen (0)**
 - **Steuerwerk sendet A**
 - **Speicherwerk 'liest' Adresse von AM (P1)**
 - **Speicherwerk 'schreibt' Befehl an dieser Adresse nach RM**
 - **Speicherwerk sendet T**
 - **Steuerwerk 'schreibt' Befehl von RM nach IR**
- b) (**, 8 Minuten) Beschreiben Sie detailliert die Kommunikation zwischen Steuerwerk und Speicherwerk beim Lesen des ersten Operanden im Decode-Schritt der Abarbeitung des Maschinenbefehls an Adresse P3.
- **Steuerwerk 'schreibt' Adresse S2 nach AM**
 - **Steuerwerk setzt D auf Lesen (0)**
 - **Steuerwerk sendet Signal A**

-
- Speicherwerk 'liest' Adresse von AM (S2)
 - Speicherwerk 'schreibt' Wert an dieser Adresse nach RM
 - Speicherwerk sendet T
 - Steuerwerk 'schreibt' Wert von RM nach AR
- c) (**, 8 Minuten) Beschreiben Sie detailliert die Kommunikation zwischen Steuerwerk und Speicherwerk beim Schreiben des Ergebnisses im Write-Back-Schritt der Abarbeitung des Maschinenbefehls an Adresse P3.
- Steuerwerk 'schreibt' Adresse S2 nach AM
 - Steuerwerk 'schreibt' Wert von AR nach WM
 - Steuerwerk setzt D auf Schreiben (1)
 - Steuerwerk sendet Signal A
 - Speicherwerk 'liest' Adresse von AM (S2)
 - Speicherwerk 'liest' Wert von WM
 - Speicherwerk überschreibt Inhalt der SZ S2 mit diesem Wert
 - Speicherwerk sendet T