



Prof. Dr.-Ing. Lars Hedrich  
M.Sc. Ahmad Tarraf

## HARDWAREARCHITEKTUREN UND RECHENSYSTEME AUFGABENBLATT 2

**Abgabe:** 25. April 2018 vor der Vorlesung.

Bitte vermerken Sie auf Ihrer Lösung Ihren Namen und die Übungsgruppenzugehörigkeit

### Aufgabe 1: Bits Schubsen

**(3 Punkte)**

Gegeben seien zwei Binärzahlen  $a = (a_7a_6 \dots a_0)$  und  $b = (b_7b_6 \dots b_0)$ . Zur Verfügung stehe ein Rechenwerk, das 4-Bit Binärzahlen addieren kann.

- (a) Wieviele Bits werden benötigt um das Ergebnis einer 4-Bit Addition darzustellen? Warum?

Wir möchten unser Rechenwerk verwenden um  $a + b$  zu berechnen. Hierfür können wir beliebig oft jeweils acht Bits (4 Bits der ersten und 4 Bits der zweiten Zahl) unserer Wahl in das Rechenwerk stecken und das Ergebnis bitweise weiterverwenden

- (b) Wieviele Additionen muss unser Rechenwerk für die Addition von  $a$  und  $b$  höchstens ausführen? Wie viele Bits ist das Ergebnis maximal lang?
- (c) Wieviele Additionen muss das Rechenwerk mindestens (schwieriger!) ausführen, damit das Ergebnis für alle möglichen Eingaben  $a$  und  $b$  korrekt wird?
- (d) Jetzt möchten wir  $a$  und  $b$  als Zweierkomplementzahlen auffassen. Welche Probleme können hierbei auftreten?

## Aufgabe 2: Rechnen mit Register- und Stackmaschinen (3 Punkte)

Seien  $a, b$  und  $c$  irgendwelche Zahlen in einem geeigneten Format. Für  $k \in \{0, \dots, 3\}$  soll der Ausdruck

$$\frac{(a - c) \cdot c \cdot d^2}{c \cdot b}$$

mit Hilfe einer  $k$ -Address-Maschine ausgewertet werden.

Schreiben Sie für diese Maschinen jeweils ein Programm, das hierzu verwendet werden kann. Auf allen Maschinen stehen die Register  $A, B, C, D, T$  und  $Z$  sowie ein Stack und ein Akkumulator zur Verfügung. In den ersten drei Registern sei zu Beginn  $a, b, c$  und  $d$  abgelegt, der Stack sei leer.

Die Programme sollen im Pseudocode formuliert werden. Für Register  $i$  und  $j$  kopiere `move i, j` den Inhalt von  $i$  nach  $j$ , `load i` lade den Inhalt von  $i$  in den Akkumulator, `stor i` schreibe Inhalt des Akkumulators in Register  $i$ , `push i` lege  $i$  auf den Stack und mit `pop i` werde das oberste Element vom Stack in  $i$  gespeichert.

Es stehen weiter die arithmetischen Befehle `add`, `sub`, `mul` und `div` für Addition, Subtraktion, Multiplikation und Division zur Verfügung. Diesen werden auf einer  $k$ -Adress-Maschine genau  $k$  Argumente übergeben. Die Semantik ist den Beispielen aus der Vorlesung zu entnehmen und im Zweifelsfall festzulegen.

Nach Ablauf des Programms soll das Ergebnis im Register  $Z$  gespeichert werden.

## Aufgabe 3: 68000-Assembler (4 Punkte)

a) Schreiben Sie ein 68000-Assembler-Programm, das die Auswertung von:

$$n^2 - 5n + 6$$

für ganze Zahlen  $n$  vornimmt. Die Zahl  $n$  soll zu Beginn in Register  $D0$  stehen, das Ergebnis wird in  $D1$  erwartet. Die Maschine arbeite mit Zweierkomplementdarstellung. Überlauf im Datenpfad darf ignoriert werden. Es dürfen für das Programm lediglich die in der folgenden Kurzreferenz angegebenen Operationen verwendet werden (der Multiplikationsbefehl ist nicht dabei!). Kommentieren Sie bitte alle Programmzeilen.

Kurzreferenz:

<b>68000-Assembler-Befehl</b>	<b>Beschreibung</b>
MOVE Quelle, Ziel	Schreibe Quelle nach Ziel
CMP Quelle, Ziel	Auswerten von Ziel - Quelle, setzen der Flags im Statusregister
BEQ Marke	Springe zu Marke, wenn Ziel - Quelle = 0 (Ziel = Quelle)
BNE Marke	Springe zu Marke, wenn Ziel - Quelle != 0 (Ziel != Quelle)
BGE Marke	Springe zu Marke, wenn Ziel - Quelle >= 0 (Ziel >= Quelle)
BGT Marke	Springe zu Marke, wenn Ziel - Quelle > 0 (Ziel > Quelle)
BLE Marke	Springe zu Marke, wenn Ziel - Quelle <= 0 (Ziel <= Quelle)
BLT Marke	Springe zu Marke, wenn Ziel - Quelle < 0 (Ziel < Quelle)
BRA Marke	unbedingter Sprung zu Marke
ADD Quelle, Ziel	Addition: Ziel = Ziel + Quelle
SUB Quelle, Ziel	Subtraktion: Ziel = Ziel - Quelle

b) Kann man folgenden Ausdruck mit dem obigen Programm ebenfalls berechnen?

$$n(n + 2) - 2(n + 5) + 16 - 5n$$