

# 64-041 Übung Rechnerstrukturen und Betriebssysteme



## Aufgabenblatt 12 Ausgabe: 18.01., Abgabe: 25.01. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

### Aufgabe 12.1 (Punkte 3+3+3+3+3)

*Adressierung:* Auf einer 1-Adress Maschine (Akkumulatormaschine) werden Ladebefehle mit unterschiedlichen Adressierungsmodi ausgeführt. Der Speicher enthält folgende Werte:

Adresse	Inhalt
20	60
30	40
40	20
50	40
60	20
70	80
80	50

Welcher Wert steht jeweils nach Ausführung der folgenden Befehle im Akkumulator?

- (a) LOAD IMMEDIATE 40
- (b) LOAD DIRECT 70
- (c) LOAD INDIRECT 60
- (d) LOAD DIRECT 80
- (e) LOAD INDIRECT 30

### Aufgabe 12.2 (Punkte 4+4+4+4+4)

*Darstellung von Immediate-Operanden:* Um trotz eingeschränkter Wortlängen bei RISC-Befehlsätzen möglichst viele, häufig benötigte Werte als *Immediate* darzustellen, benutzen die Befehlssätze aktueller Prozessoren einige Tricks. Ein gutes Beispiel zeigt die für eingebettete Systeme und Mobilgeräte sehr beliebte 32-bit ARM-Architektur. Dort ist unter anderem für arithmetische Befehle ein Format mit Immediate-Operanden definiert, bei dem der *Barrel-Shifter* benutzt wird, um 32-bit Immediate-Werte zu erzeugen:

$$\langle imm32 \rangle = \langle imm8 \rangle \text{ rotate-right } (\langle rot \rangle \cdot 2)$$

$\langle imm8 \rangle$  8-bit 0...255 beliebiger Ausgangswert

$\langle rot \rangle$  4-bit 0...15 Distanz der *rotate-right* Operation:  $\langle rot \rangle \cdot 2$  (Schrittweite 2)

cond	opcode	R <sub>src</sub>	R <sub>dest</sub>	rot	imm8
31 28 27	20 19	16 15	12 11	8 7	0

Überlegen Sie sich die jeweilige 12-bit Codierung ( $\langle rot \rangle \langle imm8 \rangle$ ) der folgenden Immediate-Werte oder begründen Sie, warum ein Wert nicht dargestellt werden kann.

- (a) 185
- (b) 351
- (c) 1576
- (d) 1584
- (e) 2684354560

### Aufgabe 12.3 (Punkte 4·10+5)

*Befehlsformate:* Vergleichen Sie 0-, 1-, 2- und 3-Adress Maschinen, indem Sie für jede Architektur ein Programm zur Berechnung des folgenden Ausdrucks schreiben. Dabei gilt (natürlich) Punkt- vor Strichrechnung:

$$R = (A - B^2) / (C * D + E)$$

Die verfügbaren Befehle der entsprechenden Maschinen sind unten angegeben. M und N stehen dabei für 24-bit Speicheradressen, während X, Y und Z eine 5-bit Registernummer codieren. MEM[M] sei der Inhalt des Speichers an der Adresse M.

**0-Adress** Maschine mit einen unbegrenzten Stack (TOS „top of stack“)

Mnemonic	Bedeutung
PUSH M	push; TOS = MEM[M]
POP M	MEM[M] = TOS; pop
ADD	tmp = TOS; pop; TOS = tmp + TOS
SUB	tmp = TOS; pop; TOS = tmp - TOS
MUL	tmp = TOS; pop; TOS = tmp * TOS
DIV	tmp = TOS; pop; TOS = tmp / TOS

**1-Adress** Maschine: Akkumulatormaschine mit genau einem Register

Mnemonic	Bedeutung
LOAD M	Akku = MEM[M]
STORE M	MEM[M] = Akku
ADD M	Akku = Akku + MEM[M]
SUB M	Akku = Akku - MEM[M]
MUL M	Akku = Akku * MEM[M]
DIV M	Akku = Akku / MEM[M]

**2-Adress** Maschine: benutzt nur Speicheroperanden

Mnemonic	Bedeutung
MOV M, N	MEM[M] = MEM[N]
ADD M, N	MEM[M] = MEM[M] + MEM[N]
SUB M, N	MEM[M] = MEM[M] - MEM[N]
MUL M, N	MEM[M] = MEM[M] * MEM[N]
DIV M, N	MEM[M] = MEM[M] / MEM[N]

**3-Adress** Register-Maschine: *load-store* RISC-Architektur, 32 Universalregister

Mnemonic	Bedeutung
LOAD X, M	X = MEM[M]
STORE M, X	MEM[M] = X
MOV X, Y	X = Y
ADD X, Y, Z	X = Y + Z
SUB X, Y, Z	X = Y - Z
MUL X, Y, Z	X = Y * Z
DIV X, Y, Z	X = Y / Z

- (a) Schreiben Sie für alle vier Maschinen (möglichst kurze) Programme für die Berechnung von  $R = (A - B^2) / (C * D + E)$ . Dabei stehen  $A \dots E$  und  $R$  für Speicheradressen der Operanden bzw. des Resultats. Zwischenergebnisse können (bei Bedarf) auf ungenutzten Speicheradressen ( $F \dots Q$ ) abgelegt werden.
- (b) Wenn die Befehlskodierung jeweils 8-bit für den Opcode verwendet, 24-bit für eine Speicheradresse und 5-bit für eine Registernummer, wie viele Bits werden dann für jedes der obigen vier Programme benötigt? Welche Maschine hat also die kompakteste Codierung (gemessen an der Programmgröße in Bits) für dieses Programm?

#### Aufgabe 12.4 (Punkte 10+10)

*Question & Answer:* beantworten Sie die folgenden Fragen zur Rechnerarchitektur. Kurze Antworten, stichwortartig oder wenige Sätze, genügen.

- (a) Wodurch unterscheiden sich *Little-Endian* und *Big-Endian*? Veranschaulichen Sie das am Beispiel der 32-bit Dezimalzahl:  $-42$  (natürlich als 2-Komplement Zahl)
- (b) Was ist der Unterschied zwischen CISC und RISC?