



Aufgabenblatt 11 Ausgabe: 10.01., Abgabe: 17.01. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 11.1 (Punkte 10+15)

Befehlscodierung: Entwerfen Sie eine möglichst einfache und einheitliche Befehlscodierung für eine 32-bit RISC Architektur. Dabei sind folgende Befehle in den 32-bit Befehlsworten unterzubringen:

- 15 Befehle mit einer 4-bit Registeradresse und einem 24-bit Immediate
 - 120 Befehle mit zwei 4-bit Registeradressen und einem Immediate-Operanden
Wie viele Bits stehen maximal für den Operanden zur Verfügung?
 - 30 Befehle ohne Adressen oder Registerangaben
- (a) Skizzieren Sie für die drei Befehlsformate beispielhaft eine Aufteilung der 32-bit Worte in Bit-/Funktions-Gruppen und begründen Sie ihre Lösung.
- (b) Nennen Sie Beispiele für jedes Befehlsformat. Dabei sollen die folgenden Befehle vorkommen, vergleiche Folie 834ff der RSB-Vorlesung:
- ALU-arithmetisch: `add`, `sub`, `inv`
 - ALU-bitweise logisch: `and`, `or`, `xor`, `not`
 - ALU-schiebend: `shl`, `sra`, `srl`, `rol`, `ror`
 - Datentransfer: `load`, `store`
 - unbedingte Sprünge: `jump`
 - bedingte Sprünge: `jumpeq`, `jumpgt`
 - Unterprogramme: `call`, `return`
 - sonstiges: `noop`, `halt`

Wenn passend, sollten auch Varianten der Befehle skizziert werden: mit ein oder zwei Registeradressen, mit oder ohne Immediate-Operanden etc.

Aufgabe 11.2 (Punkte 4-10)

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:

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

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	$X = \text{MEM}[M]$
STORE M, X	$\text{MEM}[M] = X$
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	$X = Y / Z$

Schreiben Sie für alle vier Maschinen (möglichst kurze) Programme für die Berechnung von $R = (A * B - C) / (D + E * F)$. Dabei stehen $A \dots F$ und R für Speicheradressen der Operanden bzw. des Resultats. Zwischenergebnisse können (bei Bedarf) auf ungenutzten Speicheradressen ($G \dots Q$) abgelegt werden.

Aufgabe 11.3 (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	50
30	40
40	70
50	80
60	20
70	60
80	30

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

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

Aufgabe 11.4 (Punkte 10+10)

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

- (a) Was sind Eigenschaften von SRAM und DRAM Speicher?
- (b) Was bedeuten Little-Endian und Big-Endian bei der Speicheradressierung? Was würde im Speicher stehen, wenn ab der Adresse $0x30000$ die 32-bit Dezimalzahl -2024 (natürlich als 2-Komplement Zahl) gespeichert ist? Geben Sie den Speicherinhalt für Little- und Big-Endian Systeme an.