



Aufgabenblatt 12 Ausgabe: 15.01., Abgabe: 22.01. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 12.1 (Punkte 15)

Befehlscodierung: Entwerfen Sie eine möglichst einfache und einheitliche Befehlscodierung, um alle der folgenden Befehle in 32-bit Befehlsworten unterzubringen:

- 7 Befehle mit einer 5-bit Registernummer und 24-bit für einen Immediate-Wert, bzw. eine Speicheradresse.
- 100 Befehle mit zwei 5-bit Registernummern und einem verkürzten Immediate, bzw. einem Adressoffset. Wie viele Bits sind dafür maximal verfügbar?
- 30 Befehle ohne Adressen oder Registerangaben

Skizzieren Sie für die drei Befehlsformate die Aufteilung der 32-bit Befehlsworte in die einzelnen Gruppen und begründen Sie Ihren Entwurf.

Aufgabe 12.2 (Punkte 5-3)

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

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

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

- (a) LOAD IMMEDIATE 70
- (b) LOAD DIRECT 70
- (c) LOAD INDIRECT 70
- (d) LOAD DIRECT 50
- (e) LOAD INDIRECT 50

Aufgabe 12.3 (Punkte 5-4)

Darstellung von Immediate-Operanden: Um trotz eingeschränkter Wortlängen bei RISC-Befehlsätzen möglichst viele, häufig benötigte Werte als *Immediate* darzustellen, benutzen die Befehlsätze 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 _{src}	R _{dest}	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) 167
- (b) 267
- (c) 573 440
- (d) 1 233 125 376
- (e) 1 342 177 280

Aufgabe 12.4 (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:

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

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 = 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

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 12.5 (Punkte 5·2)

Multiple-Choice: hier sind Vorlesungsinhalte der ersten Abschnitte als Multiple-Choice Fragen formuliert. Dazu sollen Sie für die nachfolgenden Aussagen jeweils ankreuzen, ob diese stimmen oder falsch sind.

(a) Im 2-Komplement wird mit dem Vektor „10...01“ die -1 dargestellt.

stimmt stimmt nicht

(b) Der Morsecode erfüllt die Fano-Bedingung nicht.

stimmt stimmt nicht

(c) Durch Hinzufügen eines Paritätsbits wird die Fehlerkorrektur von 1-bit Fehlern möglich.

stimmt stimmt nicht

(d) Bei einem Mealy-Automaten können Ausgangswerte unabhängig vom Takt wechseln.

stimmt stimmt nicht

(e) Durch *Alignment* der Daten im Speicher wird der Zugriff schneller.

stimmt stimmt nicht