Aufgabenblatt 10

Ausgabe 10/01/2011, Abgabe bis 17/01/2011 12:00

Name(n):
Matrikelnummer(n):
Übungsgruppe:

Aufgabe 10.1 Befehlskodierung (15 Punkte)

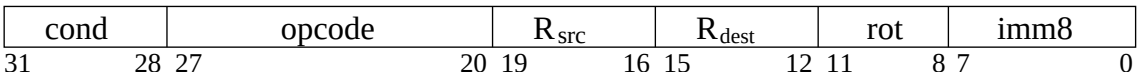
Entwerfen Sie eine möglichst einfache und einheitliche Befehlskodierung, um alle der folgenden Befehle in 32-bit Befehlsworten unterzubringen:

- 7 Befehle mit einer 5-bit Registernummer und einer 24-bit Adresse.
- 500 Befehle mit zwei 5-bit Registernummern und einem Adressoffset.
  Wie viele Bits stehen maximal für diesen Adressoffset zur Verfügung?
- 50 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 10.2 Darstellung von Immediate-Operanden (4+4+4+4+4 Punkte)

Die für eingebettete Systeme und Mobilgeräte sehr beliebte 32-bit ARM-Architektur verwendet das folgende Befehlsformat mit Immediate-Operanden für die arithmetischen Befehle:



Der bei diesen Befehlen verwendete Immediate-Wert ergibt sich aus folgendem Ausdruck:

Immediate-Wert $= \mathrm{imm8} \cdot 2^{(2*\mathrm{rot})}$

mit $0 \leq \mathrm{imm8} \leq 255$ und $0 \leq \mathrm{rot} \leq 12$.

Überlegen Sie sich die jeweilige 12-bit Kodierung der folgenden Immediate- Werte, oder begründen Sie, warum ein Wert nicht dargestellt werden kann:

a) 167
b) 267
c) 1233125376 $(= 2^{23} * 147)$
d) 2684354560 $(= 2^{31} + 2^{29})$
e) 1212416 $(= 2^{15} * 37)$

Aufgabe 10.3 Befehlsformate (40+10 Punkte)

Vergleichen Sie 0-, 1-, 2- und 3-Adress-Maschinen, indem Sie für jede Maschine ein Programm zur Berechnung des folgenden Ausdrucks schreiben:

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

Die verfügbaren Befehle der entsprechenden Maschinen sind:

0-Adress-Maschine (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)

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 (nur Speicheroperanden)

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

3-Adress-Register-Maschine (load-store RISC-Architektur)

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

Dabei bedeuten M und K jeweils eine 16-bit Speicheradresse, und MEM[M] den Inhalt des Speichers an der Adresse M, während X,Y,Z für eine 4-bit Registernummer kodieren.

Die 0-Adress-Maschine verfügt über einen unbegrenzten Stack, die 1-Adress-Maschine über einen Akkumulator, und die 3-Adress-Maschine über 16 Universalregister. Die 2-Adress-Maschine benutzt nur Speicheroperanden.

a) Schreiben Sie (möglichst kurze) Programme für die Berechnung von $W = (A+B*C)/(D-E*F)$ auf den vier Maschinen. Dabei bedeuten $A..F$ und W die Speicheradressen für die Operanden bzw. den Resultatwert. Verwenden Sie falls nötig die Speicheradressen von $H..V$ für Zwischenergebnisse.

b) Wenn die Befehlskodierung jeweils 8-bit für den Opcode verwendet (und natürlich 16-bit für eine Speicheradresse bzw. 4-bit für eine Registernummer), wieviele Bits werden dann für jedes der obigen vier Programme benötigt?

Welche Maschine hat also die kompakteste Kodierung (gemessen an der Programmgröße in Bits) für dieses Programm?

Aufgabe 10.4 Hardware für Sprungbefehle (15 Punkte)

Wir betrachten einen 32-bit Rechner (mit 32-bit Wortbreite für Speicher und Register, 32-bit Befehlen, und 32-bit Adressen). Der Rechner soll für die Ablaufsteuerung und Sprungbefehle die folgenden Operationen mit dem Programmzähler PC ausführen können:

- PC = PC + 4 (nächster Befehl)
- PC = PC + 24-bit Offset (PC-relativer Branch)
- PC = Register[i] + 20-bit Offset

Dabei sind die Offsets als Immediate-Werte im Befehlswort kodiert; und zwar in Zweierkomplementdarstellung, um auch Sprünge zu kleineren Adressen durchführen zu können.

Skizzieren Sie das Blockschaltbild der Hardware (Register, Rechenwerke, Multiplexer), um diese Befehle ausführen zu können. Versuchen Sie, mit möglichst wenig Komponenten auszukommen.