



Prof. Dr. Bernd Becker
Dr.-Ing. Christoph Scholl
Dipl. Inf. Tobias Schubert
Freiburg, 11. Januar 2001

11. Übungsblatt zur Vorlesung

Technische Informatik II

Aufgabe 1
Punkte (4)

Geben Sie eine minimale Menge von RE-TI-II Befehlen und die dabei benötigten Datenpfade an, so dass jeder Datenpfad zumindest einmal benutzt wird.

Aufgabe 2
Punkte (4)

Beweisen Sie das folgende Lemma der Vorlesung:

Sei $x \in \mathbb{B}^{32}, y \in \mathbb{B}^{24}, 0 \leq \langle x \rangle + [y] < 2^{32}$ und
sei $\langle x \rangle + \langle sex^t(y) \rangle = \langle c, s \rangle$ mit $c \in \mathbb{B}, s \in \mathbb{B}^{32}$.
Dann gilt: $\langle x \rangle + [y] = \langle s \rangle$.

Aufgabe 3
Punkte (1, 2, 2, 3)

Schreiben Sie mit den vorhandenen Maschinenbefehlen des RE-TI-II Rechners für die folgenden Shift-Operationen jeweils ein Programm, das die entsprechende Operation auf dem Inhalt der Speicherzelle $M(100)$ ausführt und das Ergebnis in $M(101)$ speichert.

- a) Linksshift:
 $lsh : \mathbb{B}^{32} \rightarrow \mathbb{B}^{32}, (a_{31}, \dots, a_0) \rightarrow (a_{30}, \dots, a_0, 0)$
- b) Zyklischer Linksshift:
 $clsh : \mathbb{B}^{32} \rightarrow \mathbb{B}^{32}, (a_{31}, \dots, a_0) \rightarrow (a_{30}, \dots, a_0, a_{31})$
- c) Zyklischer Rechtsshift:
 $crsh : \mathbb{B}^{32} \rightarrow \mathbb{B}^{32}, (a_{31}, \dots, a_0) \rightarrow (a_0, a_{31}, \dots, a_1)$
- d) Rechtsshift:
 $rsh : \mathbb{B}^{32} \rightarrow \mathbb{B}^{32}, (a_{31}, \dots, a_0) \rightarrow (0, a_{31}, \dots, a_1)$

Hinweis: Erläutern Sie zunächst Ihre Idee und dokumentieren Sie anschließend die einzelnen Zeilen Ihres Programmes ausreichend, da nicht kommentierte Programme nicht bewertet werden können.

Maschinenbefehle der RE-TI-II

Load Befehle	$I[26] = 0$	$I[25 : 24] = D$
$I[31 : 28]$	Befehl	Wirkung
0100	LOAD D_i	$D := M(\langle \hat{i} \rangle)$
0101	LOADIN1 D_i	$D := M(\langle IN1 \rangle + [i])$
0110	LOADIN2 D_i	$D := M(\langle IN2 \rangle + [i])$
0111	LOADI D_i	$D := 0^8 i$
Store Befehle	MOVE: $I[27 : 24] = S, D$	
$I[31 : 28]$	Befehl	Wirkung
1000	STORE i	$M(\langle \hat{i} \rangle) := ACC$
1001	STOREIN1 i	$M(\langle IN1 \rangle + [i]) := ACC$
1010	STOREIN2 i	$M(\langle IN2 \rangle + [i]) := ACC$
1011	MOVE S, D	$D := S$
Compute Befehle	$I[25 : 24] = D$	
$I[31 : 26]$	Befehl	Wirkung
000010	SUBI D_i	$[D] := [D] - [i]$
000011	ADDI D_i	$[D] := [D] + [i]$
000100	OPLUSI D_i	$D := D \oplus 0^8 i$
000101	ORI D_i	$D := D \vee 0^8 i$
000110	ANDI D_i	$D := D \wedge 0^8 i$
001010	SUB D_i	$[D] := [D] - [M(\langle \hat{i} \rangle)]$
001011	ADD D_i	$[D] := [D] + [M(\langle \hat{i} \rangle)]$
001100	OPLUS D_i	$D := D \oplus M(\langle \hat{i} \rangle)$
001101	OR D_i	$D := D \vee M(\langle \hat{i} \rangle)$
001110	AND D_i	$D := D \wedge M(\langle \hat{i} \rangle)$
Jump Befehle		
$I[31 : 27]$	Befehl	Wirkung
11000	NOP	$\langle PC \rangle := \langle PC \rangle + 1$
11001	JUMP $> i$	$\langle PC \rangle := \begin{cases} \langle PC \rangle + [i] & \text{falls } [ACC] c 0 \\ \langle PC \rangle + 1 & \text{sonst} \end{cases}$
11010	JUMP $= i$	
11011	JUMP $\geq i$	
11100	JUMP $< i$	
11101	JUMP $\neq i$	
11110	JUMP $<_i$	$\langle PC \rangle := \langle PC \rangle + [i]$
11111	JUMP $_i$	