



Einführung in die Rechnerarchitektur

Wintersemester 2017/2018

Tutorübung 2

30.10–03.11.2017

1. (a) Wie kann man überprüfen, ob eine Zahl gerade oder ungerade ist?
(b) Wie kann man eine bestimmte Bitposition überprüfen oder auf 0 bzw. 1 setzen („zwingen“)? Warum geht das Erzwingen auf 1 nicht universell mit einer Addition?
2. (a) Wodurch kann der 80386-Befehl `NEG EAX` ersetzt werden?
(b) Wie addiert man eine vorzeichenlose Zahl im Register AL korrekt auf eine Zahl im Register EBX?
(c) Man übersetze folgende Berechnung in ein 80386-Programm:
$$24 \cdot a + b + c + 1234$$
Dabei sind a,b,c vorzeichenlose 32-Bit-Werte und liegen bereits in den Registern EAX, EBX bzw. ECX vor. Das Ergebnis soll in EDX stehen.
(d) Die Division des 80386 legt den Divisionsrest (Remainder) im Register EDX ab, führt also eine Modulo-Berechnung durch. Für welche Teiler könnte man die Modulo-Berechnung wesentlich schneller durchführen? Welche logische Funktion würde sich anbieten?
3. Anhand der Intel-Dokumentation sollen die zu den Opcodes für die Befehle `ADD EAX, 0x12345678`, `ADD EAX, EBX` und `MOV AX, 0x10` gehörenden Bytefolgen (als Hexwerte) zusammengebaut werden. Dies entspricht einem „Assemblieren“ von Hand.
4. Festkommarechnung
 - (a) Was passiert, wenn man die in der Zentralübung erwähnte Formel zur Berechnung des Werts einer Binärzahl ($a = \sum_{i=0}^n a_i \cdot 2^i$) auf negative Indices erweitert?
 - (b) Welchen Wertebereich kann man mit 8 binären Vorkommastellen (ohne Vorzeichen) und 8 binären Nachkommastellen (8.8) erreichen?
 - (c) Wieviele Bit bräuchte man mindestens, um Zahlen von 0 bis 100 mit einer absoluten Genauigkeit kleiner 0.005 darzustellen?
 - (d) Wie sieht die Addition bzw. Subtraktion in Festkommarechnung aus? Was muss man beachten?
 - (e) Wie sieht die Multiplikation in Festkommarechnung aus? Was muss man beachten?
 - (f) Wie würde im 80386 die Multiplikation von AX mit π im 8.8-Format aussehen?