



Übungen zur „Technischen Informatik I“, WS 2004/05

Nr. 9, Abgabe: Dienstag, 4. Januar vor der Vorlesung



**Frohe Weihnachten und
 einen guten Rutsch ins Jahr 2005!**

Am 21./22.12. fallen die Vorlesung und die Übungen aus!

A. Hausaufgaben

46. Rundung von Gleitkommazahlen

3 Punkte

Der Standard IEEE 754 für Gleitkommazahlen sieht vier verschiedene Formen der Rundung vor:

- Rundung zur nächsten Zahl (*round to nearest*) mit der Vorzugsrundung auf eine gerade Zahl in der Stelle geringster Wertigkeit (voreingestellt)
 - Abschneiden der nicht darstellbaren Ziffern (*truncation/round to zero*)
 - Rundung zur nächstkleineren Zahl (*round to $-\infty$*)
 - Rundung zur nächstgrößeren Zahl (*round to $+\infty$*)
- (a) Tragen Sie in eine Tabelle der folgenden Form die Rundungen der Dezimalzahlen $+/- 1,145$, $+/- 1,146$ und $+/- 1,144$ auf jeweils zwei signifikante Nachkommastellen für jeden der vier IEEE 754-Rundungsmodi ein:

Dezimalzahl	Rundungsmodus			
	<i>nächste</i>	0	$-\infty$	$+\infty$
...				

- (b) Wie verändern sich die Rundungen, wenn alle Beispielzahlen um eine vierte Nachkommastelle erweitert werden, die jeweils mit der Ziffer 4 belegt wird?

47. Addition von Gleitkommazahlen

4 Punkte

Welche Summen haben die folgenden Dezimalbrüche, wenn man sie in normierte binäre Gleitkommazahlen umwandelt und nur jeweils 4 Bits für die Speicherung der Mantisse und des Exponenten (im Zweierkomplement) zur Verfügung hat? Verwenden Sie zur Rundung das *round-to-zero*-Prinzip. Geben Sie die Ergebnisse in binärer Darstellung an und vergleichen Sie sie mit der Summe der Dezimalbrüche (bei unbegrenzter Genauigkeit).

(a) $(0.5)_{10} + (-0.4375)_{10}$

(b) $(0.015625)_{10} + (0.484375)_{10}$

48. Mikroprogrammierung

5 Punkte

In dieser Aufgabe soll der **Mikrocodesimulator** verwendet werden, der als selbst entpackendes Archiv auf der Internetseite zur Vorlesung sowie entpackt auf dem NT-Server Bangkok unter `M:\2004WS\TechnischeInformatikI\Tools\MicroSim2000` liegt. Ihre Lösungsdatei schicken Sie bitte **zusätzlich** per email an Ihren Tutor.

Schreiben Sie ein lauffähiges, ausführlich kommentiertes Mikroprogramm, das zwei vorzeichenlose 1-Byte-Zahlen x und y von den Adressen 00h und 01h aus dem RAM in die Register R_1 und R_2 liest, im Register R_0 den Wert x^y berechnet und das Ergebnis als 4-Byte-Wert an die Adresse 04h in den RAM zurückschreibt. Beachten Sie, dass $x^0 := 1$ für alle x .

B. Mündliche Aufgaben

49. 3-Bit-Booth

Erweitern Sie den Multiplikationsalgorithmus von Booth, so dass in jeder Iteration 3 benachbarte Bits $b_{i+1}b_i b_{i-1}$ betrachtet werden. Es wird jeweils in Zweiersprüngen weitergeschoben, d.h. in der nächsten Iteration werden die Bits $b_{i+3}b_{i+2}b_{i+1}$ analysiert.

- (a) Welche Operationen sind jeweils auszuführen, wenn Multiplikand und $2 \cdot$ Multiplikand in Registern zur Verfügung stehen? Vervollständigen Sie hierzu die folgende Tabelle.

b_{i+1}	b_i	b_{i-1}	Situationsbeschreibung	auszuführende Aktion
0	0	0		
\vdots	\vdots	\vdots		

- (b) Finden Sie ein Beispiel für 6-Bit-Zahlen, bei der dieser 3-Bit-Booth Algorithmus effizienter ist als die Version mit 2 Bits.

50. Iterative Division

- (a) Optimieren Sie das iterative Divisionsverfahren der Vorlesung, indem Sie analog zur Optimierung des Multiplikationsverfahrens
- eine Konfiguration mit einem n-Bit-Addierer/Subtrahierer erstellen und
 - das Akkumulatorregister sowohl für die Ermittlung des Restes als auch des Quotienten verwenden.

Geben Sie die modifizierte Konfiguration und das zugehörige Ablaufdiagramm an. Erläutern Sie die vorgenommenen Modifikationen.

- (b) Zeigen Sie, wie mit Ihrem Verfahren $(01110101)_2$ durch $(1011)_2$ dividiert wird.

51. Mikroprogrammierte Multiplikation

Schreiben Sie ein lauffähiges, ausführlich kommentiertes Mikroprogramm, das die Multiplikation durch fortgesetzte Addition iterativ realisiert. Multiplikator und Multiplikand seien je 16 Bits lang und vorzeichenlos. Verwenden Sie den in der Vorlesung besprochenen Multiplikationsalgorithmus. Das letzte Bit eines Registers kann mit einem konstanten Bitmuster als „Maske“ abgefragt werden.

