

Übungsblatt 5 zur Vorlesung „**Technische Informatik I**“ **SS 2001**
Strey / Guenkova-Luy / Prager

Aufgabe 1

Eine IEEE 754 32-Bit Gleitkommazahl hat folgendes Format: $x = (-1)^s \cdot 1.f \cdot 2^{e-127}$ wobei die einzelnen Bits ($b_{31}..b_0$) in einem Rechner in folgender Reihenfolge abgelegt werden: ($s, e_7, ..e_0, f_{-1}, ..f_{-23}$). Details zu diesem Format finden Sie auf den Vorlesungsfolien.

- Welche Zahlen werden durch die Bitmuster `0|10000101|000000100000000000000000` und `1|10000100|100100000000000000000000` dargestellt? (Die senkrechten Striche dienen hierbei lediglich der Übersicht.)
- Bestimmen Sie die Gleitkommadarstellung der Zahlen 1, -5 und $6.25 \cdot 10^{-3}$.
- Warum wird zur Darstellung der Zahl 0 eine Ausnahmeregel benötigt?
- Warum ist es in Programmiersprachen möglich, die Zahl $1 \cdot 10^{-41}$ erfolgreich in einer 32-Bit Gleitkommazahl abzulegen, wohingegen das Zuweisen von $1 \cdot 10^{41}$ vom Übersetzer zurückgewiesen wird?
- Berechnen Sie die Summe der Zahlen aus Aufgabe a). Beachten Sie dabei, daß die Eingaben sowie das Ergebnis in normalisierter Form vorliegen.
- Mit welchen der 5 Ausnahmezustände ($\pm\infty$, ± 0 , NaN) würden Sie die Ergebnisse der Rechnungen $\sqrt{-13}$, $\log_{10}(0)$ und $\log_{10}(-1)$ codieren?

Aufgabe 2

- Entwerfen Sie das Blockschaltbild eines Addierwerkes für Gleitkommazahlen, das zwei Zahlen a und b mit $|a| > |b|$ addieren soll. Sie dürfen dabei folgende komplexen Schaltungen als fertige Komponenten (für beliebige Wortbreiten) als gegeben ansehen: Addierer und Subtrahierer für ganze Zahlen, 2er-Komplementierer, Multiplexer sowie kombinatorische Rechts- oder Linkshifter, die als zusätzliche Eingabe die Schiebeposition akzeptieren. Zusätzlich können Sie ein Schaltnetz „NLZ“ („Number of Leading Zeros“) als gegeben ansehen, welches in der Lage ist, die Anzahl der führenden Nullen eines n-Bit Wortes zu bestimmen und diese als Binärzahl codiert. Darüber hinausgehende Schaltungskomponenten müssen entworfen werden. Lassen Sie in Ihrer Schaltung eventuell auftretende Ausnahmefälle unberücksichtigt.
- Überlegen Sie sich, wie die Division von Gleitkomma-Zahlen prinzipiell realisiert werden kann. Welche Ausnahmezustände können hierbei auftreten und wie würden Sie das Ergebnis codieren?

Aufgabe 3

Die Programmiersprache Java kennt 2 (primitive) Typen von Gleitkommazahlen gemäß dem IEEE 754-Standard: `float` („Single Precision“) und `double` („Double Precision“). Ein zweifelnder Programmierer legt Ihnen folgendes Fragment eines Java-Programms vor, welches die Werte einer Funktion `some_function()` in einem Zahlenbereich aufsummieren soll:

```
float integrate() {
    float sum = 0;
    for (float x = 20000000; x < 20000010; x = x + 1) {
        sum = sum + some_function(x);
    }
    return sum;
}
```

Der Programmierer beklagt sich, daß sich das Programm grundsätzlich bei einem Aufruf der Methode `integrate` „aufhängt“, d.h. in eine Endlosschleife gerät. Können Sie ihm weiterhelfen?

- Bestimmen Sie die kleinste natürliche Zahl l_{float} , die im IEEE Single Precision Format **nicht** darstellbar ist.
- Warum funktioniert das Programmfragment nicht wie erwartet?
- Wie können Sie obiges Programm modifizieren, damit es die gewünschte Funktionalität bietet? Innerhalb welcher Grenzen für die Laufvariable x funktioniert das Programm nun korrekt?
- Wir ersetzen im obigen Programm innerhalb der Schleife $x=x+1$ durch $x=x+0.1$. Welchen Einfluß hat dies auf die Grenzen der Laufvariable x , innerhalb derer das Programm korrekt arbeitet?

Aufgabe 4

Sie haben in der Vorlesung einen hypothetischen Prozessor kennengelernt. Mit der RTL-Notation („Register Transfer Language“) kann man die Arbeitsweise von Befehlen dieses Prozessors mathematisch beschreiben. Beschreiben Sie die folgenden Befehle in RTL:

- SUB #17,D0
- BRA 15
- BMI 35

Aufgabe 5

Gegeben sei folgendes Programmfragment (C, C++ oder Java):

```
if (x!=0) && (x<=15) {
    a = a*x;
}
else {
    a = a+1;
}
```

Der Operator `!=` bedeutet dabei \neq , der Operator `&&` logisches UND. Schreiben Sie eine Sequenz von Maschinenbefehlen für den hypothetischen Prozessor der Vorlesung, welche die obigen Anweisungen realisiert. Die wichtigsten Maschinenbefehle finden Sie in der folgenden Tabelle. Das Symbol `a` steht dabei als Platzhalter einer Adresse, `x` für eine Konstante.

MOVE a,D0		BCC a	; branch if C=0
MOVE D0,a		BCS a	; branch if C=1
MOVE #x,D0		BMI a	; branch if N=1 (minus)
ADD a,D0		BPL a	; branch if N=0 (plus)
ADD #x,D0	; add x to [D0]	BRA a	
SUB a,D0		AND a,D0	
SUB #x,D0	; subtract x from [D0]	OR a,D0	
MUL a,D0	; multiply	NEG D0	
MULH a,D0	; higher bits of MUL	LSL D0	
BEQ a	; branch if Z=1	ASR D0	
BNE a	; branch if Z=0	STOP	