Übung 7

Rechnerarchitektur
Wintersemester 18/19
Prof. Dr.-Ing. Jochen Schiller

Ausgabe 30.12.2018
Abgabe 14.12.2018, 10:15 (s.t.)

Eigenschaften des IEEE-P 754

Die Basis b ist gleich 2.

Das erste Bit der Mantisse wird implizit zu 1 angenommen, wenn die Charakteristik nicht nur Nullen enthält.

Normalisierung: das erste Bit der Mantisse (die implizite 1) steht vor dem Komma.

Ist die Charakteristik gleich 0, entspricht dies dem gleichen Exponenten wie die Charakteristik 1.

Das erste Bit der Mantisse wird aber dann explizit dargestellt

Auch die Null ist darstellbar

Eigenschaften des IEEE-P 754

Sind alle Bits der Charakteristik gleich 1, signalisiert dies eine Ausnahmesituation.

Wenn zusätzlich die Mantisse gleich Null ist, wird die Situation „overflow" (bzw. die „Zahl" $\pm\infty$) kodiert.

Dies erlaubt es dem Prozessor, eine Fehlerbehandlung einzuleiten.

Intern arbeiten Rechner nach dem IEEE-Standard mit 80 Bit, um Rundungsfehler unwahrscheinlicher zu machen.

Charakter.	Zahlenwert	Bemerkung
0	$(-1)^{Vz}$ 0,Mantisse · 2^{-1022}	
1	$(-1)^{Vz}$ 1,Mantisse · 2^{-1022}	
...	$(-1)^{Vz}$ 1,Mantisse · $2^{\text{Charakteristik} - 1023}$	
2046	$(-1)^{Vz}$ 1,Mantisse · 2^{1023}	
2047	Mantisse = 0: $(-1)^{Vz}\infty$	Overflow
2047	Mantisse ≠ 0: NaN	Not a number

Aufgabe 1: Fließkommazahlen

Fragen

Beantworten Sie folgende Fragen:

1. Nennen Sie Beispiele für Festlegungen, die der IEEE 754 Standard mitbringt. Warum ist eine solche Standardisierung sinnvoll?
2. Was ist die betragsmäßig größte bzw. kleinste darstellbare Zahl im IEEE-754 32bit Standard? Geben Sie auch die Bits an.

Addition

Rechnen Sie die folgenden Zahlen in IEEE-754 32bit Darstellung um und addieren Sie sie in dieser Darstellung miteinander. Runden Sie korrekt! Stellen Sie das Ergebnis sowohl im IEEE-754 Format als auch als Dezimalzahl dar.

- -592,183940
- 0,91213

Multiplikation

Rechnen Sie die folgenden Zahlen in IEEE-754 32bit Darstellung um und multiplizieren Sie sie in dieser Darstellung miteinander. Runden Sie korrekt! Stellen Sie das Ergebnis sowohl im IEEE-754 Format als auch als Dezimalzahl dar.

- 3981.1729
- -2.91762

Aufgabe 2: Floating Point Rechner

Implementieren Sie einen Floating Point Rechner in Software. Zwei gegebene 32-bit IEEE-754 Zahlen (*operand1* und *operand2*) sollen addiert werden können. Das Ergebnis soll ebenfalls im 32-bit IEEE-754 Floating Point Format (*) an die Adresse *result* geschrieben werden. Verwenden Sie keine Floating Point Register/Befehle. Denken Sie auch an Normalisierung, NaN und "Unendlichkeit". Sie können von folgender Signatur ausgehen (C-Datei wird gestellt):

```
void calc_add(float operand1, float operand2, float* result);
```

*) Sie müssen sich nicht zwingend um korrekte Rundung kümmern. Auch die korrekte Behandlung von Charakteristik=0 als Sonderfall ist ein Bonus.

VZ	char	Mantisse
1 bit	8 bit	23 bit

32 Bit

$15{,}25_{10} = +1111{,}01 \cdot 2^0 = 1{,}11101 \cdot 2^3$

wird von Pc generiert (nicht hinschreiben)

$127 + 3$
$130 = \ldots$

Festkomma

Normalisiert ein $\hat{=} 1{,}xxx \cdot 2^z$

char Exzess Darstellung

offset 127 [01111111] addieren mit basis für char

VZ	char	Mantisse
0		11101000... bis 23 Bit

Addition I 0 | 10000010 | 11101000...

II 1 | 10001010 | 11111000...

Bsp $1{,}34 \cdot 10^3$
$+2{,}60 \cdot 10^5$

! exakt gleicher Exponent

$\approx 0{,}0134 \cdot 10^5$
$2{,}60 \cdot 10^5$

Zahl mit kleinerem Exponenten wird angepasst

10000010
− 10001010
= ..... = 8

Differenz

Also I Mantisse um 8 verschieben

$\Rightarrow$ 0,00000001 11101000...

wird von Pc generiert

immer wenn Zahl größer als > 1

II 1,11111000
I −0,00000001 11101
= −1,11101100001100

VZ unterschiedlich $\Rightarrow |X_1| \geq |X_2|$

X_1
$-X_2$
= Vorzeichen von X_1 (betragsgrößere − betragskleine)

VZ	char	Mantisse
1	10001010	11101100001100

Multiplizieren $1,34 \cdot 10^{3}$ } Exponenten egal
$2,6 \cdot 10^{4}$

$1,$

```
1010 * 0101
       1010
      0000
     1010
+   0000
    01100010
```

Bei 0 nur 0 schreiben
Bei 1 linke Zahl abschreiben

→ char addieren
Mantisse mul

1. Operanden holen
2. Optionen
   rfd = VZ OP1
   rgd = char OP1
   r10d = Mantisse OP1
   esi = VZ OP2
3. Exponent Vergleich
   3 Fälle: -gleich
   Op1 > Op2
   Op1 < Op2
4. explizite 1 hinzufügen
   Mantisse + 2^{23}
5. exponenten angleichen
6. Rechnen $\begin{matrix}\nabla\\ \circ\end{matrix}$ 2 Fälle
   - Vorzeichen =
   - Vorzeichen ≠
7. Normalisieren 3 Fälle
   - 1x,xx ← 1 Rechts
   - 1,xxx ← nichts
   - 0,xxx einschiften

char nach $\overset{23}{\leftarrow}$ und den $\overset{1}{\rightarrow}$

8. explizite 1
9. Ergebnis Bauen
10. zurückschalten

Addition

Rechnen Sie die folgenden Zahlen in IEEE-754 32bit Darstellung um und addieren Sie sie in dieser Darstellung miteinander. Runden Sie korrekt! Stellen Sie das Ergebnis sowohl im IEEE-754 Format als auch als Dezimalzahl dar.

- -592,183940
- 0,91213

1. IEEE umwandeln
2. addieren

591,27181

$592 = 1001001000 \cdot 2^0$

$0,183940 = 0,0010111100101100010 \cdot 2^0$

$0,91213 = 0,11101001100000010111101 \cdot 2^0$

0,183940 · 2
0,36788 · 2
0,73576
1,47152
0,94304
1,88608

$-592,183940 = 1001001000,0010111100101100010 \cdot 2^0$

$\Rightarrow 1,0010010000010111100101100010 \cdot 2^9$

$127 + 9 = 136 = 10001000$

$-592,183940 =$ | Vz | char (136) | Mantisse I

1 | 10001000 | 00100000010111100101100010

gerundet von 011 zu 10

$0,91213 - 592,183940 = -591,28181$

592,183940 — $1,0010010000,0010111100101100010 \cdot 2^9$
0,91213 — $0,0000000001110100110000000 \cdot 2^9$

$1001001111010010110011010$

Multiplikation

Rechnen Sie die folgenden Zahlen in IEEE-754 32bit Darstellung um und multiplizieren Sie sie in dieser Darstellung miteinander. Runden Sie korrekt! Stellen Sie das Ergebnis sowohl im IEEE-754 Format als auch als Dezimalzahl dar.

- 3981.1729
- -2.91762

• = -11,615,5497

$2^{11}\ 2^{10}\ 2^{9}\ 2^{8}\ 2^{7}\ 2^{6}\ 2^{5}\ 2^{4}\ 2^{3}\ 2^{2}\ 2^{1}\ 2^{0}$

3981 = 1 1 1 1 1 0 0 0 1 1 0 1

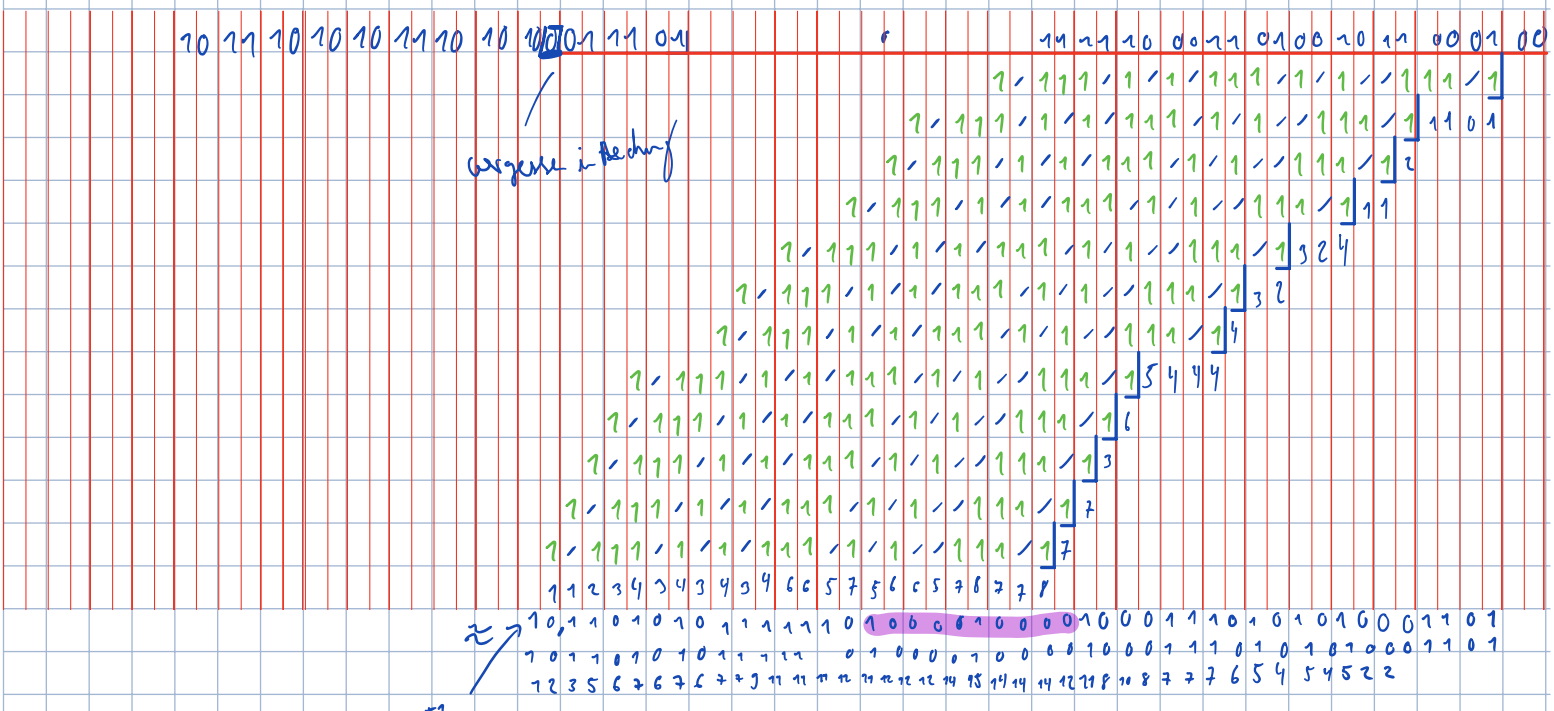
$2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ 2^{-5}\ 2^{-6}\ 2^{-7}\ 2^{-8}\ 2^{-9}\ 2^{-10}\ 2^{-11}\ 2^{-12}\ 2^{-13}\ 2^{-14}\ 2^{-15}\ 2^{-16}$

0,1729 = 0,0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 1

-2,91764 = 1 0, 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 1

3981,1729 = 1 1 1 1 1 0 0 0 1 1 0 1, 0 0 1 0 1 1 0 0 0 1 0 0

-2,91764 = 1 0, 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1 0 1



addiere der exponenten für char 1+1+11 = 13

char 130