technische universität
dortmund

fakultät für informatik

Rechnerstrukturen, Teil 1

Vorlesung 4 SWS WS 18/19

Prof. Dr. Jian-Jia Chen
Fakultät für Informatik – Technische Universität Dortmund
jian-jia.chen@cs.uni-dortmund.de
http://ls12-www.cs.tu-dortmund.de

5. Rechnerarithmetik

1. Einleitung ✓
2. Addition natürlicher Zahlen ✓
3. Multiplikation natürlicher Zahlen ✓
4. Addition ganzer Zahlen ✓
5. **Addition von Fließkommazahlen**
6. Multiplikation von Fließkommazahlen

5.5 Addition von Fließkommazahlen

Fließkommazahlen

Darstellung gemäß IEEE 754-1985

$$x = (-1)^{s_x} \cdot m_x \cdot 2^{e_x}$$

$$y = (-1)^{s_y} \cdot m_y \cdot 2^{e_y}$$

$$z = x + y = (-1)^{s_z} \cdot m_z \cdot 2^{e_z}$$

- s Vorzeichenbit
- m Mantisse (Binärdarstellung, inklusive impliziter 1)
- e Exponent (Exzessdarstellung, $b = 2^{l-1} - 1$)

Beobachtung

- Addition ist einfach wenn $e_x = e_y = e$ und $s_x = s_y = s$ gilt:
- $z = (-1)^s \cdot (m_x + m_y) \cdot 2^e$

5.5 Addition von Fließkommazahlen

Algorithmus zur Addition

Ziel: Berechne $z = x + y$

Ohne Beschränkung der Allgemeinheit: Sei $e_x \geq e_y$
$\Rightarrow$ sonst werden x und y vertauscht

Wir unterscheiden zwei Fälle:

1. $s_x = s_y$: Beide Zahlen haben das gleiche Vorzeichen
2. $s_x \neq s_y$: Die Zahlen haben unterschiedliche Vorzeichen

Generelles Problem: Bei der Addition können signifikante Bits verloren gehen!

5.5 Addition von Fließkommazahlen

Algorithmus zur Addition

Fall 1: $s_x = s_y$ (gleiches Vorzeichen)

$$z = x + y = (-1)^{s_x} [m_x 2^{e_x} + m_y 2^{e_y}]$$
$$= (-1)^{s_x} \cdot 2^{e_x} [m_x + m_y 2^{e_y - e_x}]$$

Für z gilt:

1. $s_z := s_x$
2. $e_z := e_x$
3. $m_z := m_x + m_y 2^{e_y - e_x}$

Falls $m_z \geq 2$: Ergebnis Normalisieren

$e_z := e_z + 1$

$m_z := m_z / 2$

5.5 Addition von Fließkommazahlen

Algorithmus zur Addition

Fall 2: $s_x \neq s_y$ (unterschiedliche Vorzeichen)

$$z = x + y = (-1)^{s_x} [m_x 2^{e_x} - m_y 2^{e_y}]$$
$$= (-1)^{s_x} \cdot 2^{e_x} [m_x - m_y 2^{e_y - e_x}]$$

Für z gilt:

1. $s_z := s_x$
2. $e_z := e_x$
3. $m_z := m_x - m_y 2^{e_y - e_x}$

Falls $m_z = 0$: $e_z := e_{min} - 1$
Sonst: $1 \leq m_z \cdot 2^q < 2$ für eine ganze Zahl q

$e_z := e_z - q$
$m_z := m_z \cdot 2^q$

5.5 Addition von Fließkommazahlen

Algorithmus zur Addition

Fall 2: $s_x \neq s_y \Rightarrow x - y \Rightarrow$ Umwandlung von y ins Zweierkomplement

Beispiel: $m_y = 1.0101\ 0000\ 0000\ 0000\ 0000\ 000$
Vorzeichenerweiterung: $01.0101\ 0000\ 0000\ 0000\ 0000\ 000$
Einerkomplement + 0.0...1: $10.1011\ 0000\ 0000\ 0000\ 0000\ 000$

Mantisse verschieben: Vorne mit Einsen auffüllen

Beispiel: $-m_y \cdot 2^{-5} = 11{,}1111\ 0101\ 1000\ 0000\ 0000\ 000$

Begründung:

$$m_y \cdot 2^{-5} = 00,0000\ 1010\ 1000\ 0000\ 0000\ 000$$
$$-(m_y \cdot 2^{-5}) = 11,1111\ 0101\ 1000\ 0000\ 0000\ 000$$

5.5 Addition von Fließkommazahlen

Beispiel Addition von Gleitkommazahlen

x	1	1001	0101	111	0010	0000	0000	0000	0000
y	1	1001	0100	110	0001	1000	0000	0000	0000

$e_x > e_y$, Vorzeichen gleich, also zunächst nur $s_z := 1$

Mantisse m_y um $e_x - e_y = 1$ Stelle nach rechts verschieben

➡ 0,111000011

Mantissen addieren

	1,	1	1	1	0	0	1		
+	0,	1	1	1	0	0	0	0	1	1
1	0,	1	1	0	0	0	1	0	1	1

Normalisieren

- Komma um 1 Stelle nach links verschieben ➡ 1,0110001011
- Exponent um 1 vergrößern ➡ 1001 0110

Z	1	1001	0110	011	0001	0110	0000	0000	0000

5.5 Addition von Fließkommazahlen

Noch ein Beispiel zur Addition von Gleitkommazahlen

x	1	1000	0101	010	0000	0000	0000	0000	0000
y	0	1000	0100	101	1010	0000	0000	0000	0000

Es gilt: $e_x > e_y$

Da $s_x \neq s_y$ muss s_y invertiert werden

Vorzeichenwechsel bei m_y in Zweierkomplementdarstellung

X	1	1000	0101	1,	010	0000	0000	0000	0000	0000
aus	0	1000	0100	01,	101	1010	0000	0000	0000	0000
wird y	1	1000	0100	10,	010	0110	0000	0000	0000	0000

5.5 Addition von Fließkommazahlen

Noch ein Beispiel zur Addition von Gleitkommazahlen (2)

x	1	1000	0101	1	,	010	0000	0000	0000	0000	0000
y	1	1000	0100	10	,	010	0110	0000	0000	0000	0000

jetzt e_y an e_x anpassen, m_y verschieben

x	1	1000	0101	1	,	010	0000	0000	0000	0000	0000
y	1	1000	0101	11	,	001	0011	0000	0000	0000	0000
z	1	1000	0101	100	,	011	0011	0000	0000	0000	0000

Erinnerung „überfließende“ 1 einfach ignorieren

z	1	1000	0101	0	,	011	0011	0000	0000	0000	0000

Normalisieren Komma um zwei Stellen nach rechts verschieben

Exponent zum Ausgleich um zwei verkleinern

z	1	1000	0011	100	1100	0000	0000	0000	0000

5. Rechnerarithmetik

1. Einleitung ✓
2. Addition natürlicher Zahlen ✓
3. Multiplikation natürlicher Zahlen ✓
4. Addition ganzer Zahlen ✓
5. Addition von Fließkommazahlen ✓
6. **Multiplikation von Fließkommazahlen**

5.6 Multiplikation von Fließkommazahlen

Gleitkommazahlen-Arithmetik

Darstellung gemäß IEEE 754-1985

$$x = (-1)^{s_x} \cdot m_x \cdot 2^{e_x}$$

$$y = (-1)^{s_y} \cdot m_y \cdot 2^{e_y}$$

- s Vorzeichenbit
- m Mantisse (Binärdarstellung, inklusive impliziter 1)
- e Exponent (Exzessdarstellung, $b = 2^{l-1} - 1$)

Ergebnis $z = (-1)^{s_z} \cdot m_z \cdot 2^{e_z}$

Vereinfachung Wir ignorieren das Runden.

5.6 Multiplikation von Fließkommazahlen

Multiplikation von Gleitkommazahlen

$$x = (-1)^{s_x} \cdot m_x \cdot 2^{e_x}$$

$$y = (-1)^{s_y} \cdot m_y \cdot 2^{e_y}$$

$$z = x \cdot y = (-1)^{s_z} \cdot m_z \cdot 2^{e_z}$$

Beobachtung $z = (-1)^{s_x \oplus s_y} \cdot (m_x \cdot m_y) \cdot 2^{e_x + e_y}$

Vorgehen

1. $s_z := s_x \oplus s_y$
2. $m_z := m_x \cdot m_y$
   - Multiplikation von Betragszahlen wie gesehen,
   - implizite Einsen nicht vergessen!
3. $e_z := e_x + e_y$
   - Addition, wegen Exzessdarstellung $e_x + e_y - b$ berechnen

5.6 Multiplikation von Fließkommazahlen

Beispiel Multiplikation von Gleitkommazahlen

x	1	1000	0101	101	0000	0000	0000	0000	0000
y	1	1000	0111	110	1000	0000	0000	0000	0000

Vorzeichen $s_z = 1 \oplus 1 = 0$

Exponent Bias ist $2^{l-1} - 1 = (1000\ 0000)_2 - 1$

$$e_z := e_x + e_y - b$$

$$\begin{aligned} e_y - b &= (1000\ 0111)_2 - ((1000\ 0000)_2 - 1) \\ &= (1000\ 0111)_2 - (1000\ 0000)_2 + 1 \\ &= (111)_2 + 1 = (1000)_2 \end{aligned}$$

$$e_x + e_y - b = (1000\ 0101)_2 + (1000)_2 = (1000\ 1101)_2$$

→ $(1000\ 1101)_2$ ist vorläufiger Exponent

5.6 Multiplikation von Fließkommazahlen

Beispiel Multiplikation von Gleitkommazahlen

Mantisse

	1,	1	0	1	·	1,	1	1	0	1
	1	1	0	1						
		1	1	0	1					
			1	1	0	1				
				0	0	0	0			
+					1	1	0	1		
1	0,	1	1	1	1	0	0	1		

Normalisieren:

- Komma 1 Stelle nach links
- Exponent zum Ausgleich +1
- implizite Eins streichen

z	0	1000	1110	011	1100	1000	0000	0000	0000

5. Rechnerarithmetik

1. Einleitung ✓
2. Addition natürlicher Zahlen ✓
3. Multiplikation natürlicher Zahlen ✓
4. Addition ganzer Zahlen ✓
5. Addition von Fließkommazahlen ✓
6. Multiplikation von Fließkommazahlen ✓

5 Rechnerarithmetik: Real-world numerical catastrophes

- *Ariane 5 rocket.* Ariane 5 rocket exploded 40 seconds after being launched by European Space Agency on June 4th, 1996. (http://www.youtube.com/watch?v=gp_D8r-2hwk) Maiden voyage after a decade and 7 billion dollars of research and development. Sensor reported acceleration that was so large that it caused an overflow in the part of the program responsible for recalibrating inertial guidance. 64-bit floating point number was converted to a 16-bit signed integer, and the conversion failed. This resulted in a drastic attempt to correct the nonexistent problem, which separated the motors from their mountings, leading to the end of Ariane 5.
- *Patriot missile accident.* On February 25, 1991 an American Patriot missile failed to track and destroy an Iraqi Scud missile. Instead it hit an Army barracks, killing 26 people. The cause was later determined to be an inaccurate calculate caused by measuring time in tenth of a second. Couldn't represent 1/10 exactly since used 24 bit floating point
- *Intel FDIV Bug* Error in Pentium hardwire floating point divide circuit. Discovered by Intel in July 1994, rediscovered and publicized by math professor in September 1994. Intel recall in December 1994 cost $300 million. Another floating point bug discovered in 1997.

Source: http://introcs.cs.princeton.edu/java/91float/

Sehr wichtiger Artikel über Fließkommazahlen

David Goldberg (1991): What every computer scientist should know about floating-point arithmetic. ACM Computing Surveys 23(1):5–48.