

Data rep(resentation and) floating point (numeric)

Data rep - principle decimal fix-point und floating point

- Dezimalzahl mit festen Komma (fix-point)

- $d_m d_{m-1} \cdots d_1 d_0 . d_{-1} d_{-2} \cdots d_{-n}$
- Berechnung des Werts: $d = \sum_{i=-n}^m 10^i * d_i$
- Beispiel: 123,45

- Dezimalzahl mit Gleitkomma (floating point)

- $d_m d_{m-1} \cdots d_1 d_0 . d_{-1} d_{-2} \cdots d_{-n}$



lässt sich immer konvertieren in

Normalisierung

- $d_m . d_{m-1} \cdots d_1 d_0 d_{-1} d_{-2} \cdots d_{-n} \times 10^m$

- Anwendung: Darstellung von sehr großen oder sehr kleinen Zahlen

Data rep - principle binary fix-point

- Binärzahl mit festem Komma (fix point)

$$b_m b_{m-1} \cdots b_1 b_0 . b_{-1} b_{-2} \cdots b_{-n}$$

- Bestimmung des Werts bzw. Umrechnung bin->dec:

$$d = \sum_{i=-n}^m 2^i * b_i$$

- Beispiel:

- 0b0.1 -> 0.5

- 0b11.01 -> 3.25

Data rep - principle binary fix-point - dec -> bin 1/3

- Umrechnung dec -> bin
 - Vorab wichtige Punkte:
 - Für jede **ganze Zahl** sind Dezimaldarstellung und Binärdarstellung (Hexadezimal, ...) ineinander überführbar
 - Für **gebrochene Zahlen** gilt dies **prinzipiell nicht**
 - Bsp:
 - $x = 1/10 = 0.1$
 - $x = 1/16 + 1/32 + 1/256 + 1/512 + 1/4096 + \dots$
 $= 0.00011001100110011001100110011 \dots$
 - benötigt in Binärdarstellung ∞ viele Stellen !
 - (Darstellung eines Binärbruchs ist allerdings immer mit endlich vielen Dezimalstellen möglich!)
(Wenn der Nenner aus 2er-Potenzen besteht)


Data rep - principle binary fix-point - dec -> bin 2/3

- Umrechnung dec -> bin
 - 1. Vorkomma mit Quotientenmethode (siehe Umwandlung von Ganzzahlen (int) im letzten Kapitel)
 - 2. Nachkomma mit Produktmethode (siehe nächste Folie)

- Beispiel: 17,625 -> bin?

- 1.: Vorkomma 17 -> bin

d	d div 2	d mod 2
17	8	1
8	4	0
4	2	0
2	1	0
1	0	1



-> Dec2bin(17) = 0b10001


- 2.: Nachkomma 0,625
 - > Dec2bin(0,625) = 0b101 (siehe nächste Folie)
 - > 0b10001.101

Data rep - principle binary fix-point - dec -> bin 3/3

- Algorithmus dec2bin(d):
 - Produktmethode (für Nachkommanteil von Kommazahlen)
 - Multipliziere d und Produktnachkomma fortlaufend mal 2 und notiere Vorkomma bis Produkt 1,0 ergibt (oder maximal gewünschte Stellen erreicht)
 - Vorkomma von oben nach unten gelesen ergibt das Ergebnis

- Beispiel: dec2bin(0,625)

d*2	prod	Nach komma	Vorkomma
0,625 *2	1,25	0,25	1
0,25 *2	0,5	0,5	0
0,5 *2	1,0	0	1



->Dec2bin(0,625) =0b101

Data rep - principle binary fix-point - dec -> bin - Übung

- Umrechnung dec -> bin
 - 1. Vorkomma mit Quotientenmethode
 - 2. Nachkomma mit Produktmethode
 - Multipliziere d und Produktnachkomma fortlaufend mal 2 und notiere Vorkomma bis Produkt 1,0 ergibt (oder maximal gewünschte Stellen erreicht)
 - Vorkomma von oben nach unten gelesen ergibt das Ergebnis

▪ Übung: 18,75 -> bin?

- 1.: Vorkomma 18 -> bin

d	d div 2	d mod 2
18	9	0
9	4	1
4	2	0
2	1	0
1	0	1

-> Dec2bin(18) = 0b10010

- 2.: Nachkomma 0,75 -> bin

d*2	prod	Nachkomma	Vorkomma

->Dec2bin(0,75) =

Data rep - principle binary floating point

- Binärzahl mit festem Komma (fix point)

- $$b_m b_{m-1} \cdots b_1 b_0 . b_{-1} b_{-2} \cdots b_{-n}$$

- Binärzahl mit Gleitkomma (floating point)

- $$b_m . b_{m-1} \cdots b_1 b_0 . b_{-1} b_{-2} \cdots b_{-n} \times 2^m$$

Normalisierung

Data rep - floating point - IEEE 754

- Beschreibt allgemeine Zahl: $V = (-1)^s * M * 2^E$
- Sign bit (1 Bit):
 - $s=1 \rightarrow$ negativ, $s=0 \rightarrow$ positiv
- Exp field (k Bits):
 - Codiert Exponent E
- Fraction field (n Bits):
 - Codiert Signifikant M

Data rep - floating point - IEEE 754 - float 1/4

4 Fälle

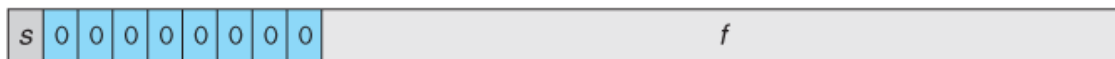
Single precision



1. Normalized



2. Denormalized



3a. Infinity



3b. NaN

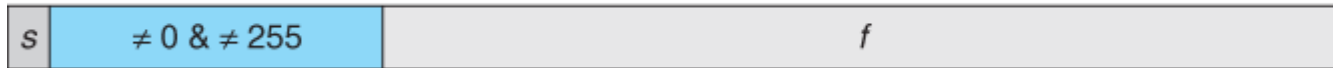


Data rep - floating point - IEEE 754 - float 2/4

▪ Fall 1 wenn $\text{exp} \neq 0$ und $\neq 255$ (normalized)

- Allgemeine Zahl: $V = (-1)^s * M * 2^E$
- Float single precision 32bit (4 Byte)

1. Normalized



- Sign bit (one Bit s): Bit 31
 - $s=1 \rightarrow$ negativ, $s=0 \rightarrow$ positiv
- Exp field (k Bits): 8 Bits (Bits 30 – 23)
 - Codiert exponent entspr: $E = \text{exp} - \text{Bias}$
 - Bias=127 $\rightarrow E = \dots -126$ to $+127$
- Frac(tion) field (n Bits): 23 Bits (Bits 22 – 0)
 - Codiert significand entspr: $M = 1.f_{22} \dots f_1 f_0 \rightarrow M = 1 \dots 2$
- **Anwendung: normale Zahlendarstellung**

Data rep - floating point - IEEE 754 - float 2/4 - Beispiel

▪ Fall 1 wenn $\text{exp} \neq 0$ und $\neq 255$ (normalized)

- Allgemeine Zahl: $V = (-1)^s * M * 2^E$
- $M = 1.f_1f_2 \dots f_{23}f_{24}$
- $E = \text{exp} - 127$

▪ Beispiel

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- $s = 1 \rightarrow -$
- $\text{exp} = 64 + 32 + 16 + 8 + 4 + 1 = 125 \rightarrow E = 125 - 127 = -2$
- $\text{frac} = 0.5 \rightarrow M = 1.5$
- $V = -1 * 1.5 * 2^{(-2)} = -1 * 3/2 * 1/4 = -3/8 = -0.375$

Data rep - floating point - IEEE 754 - float 2/4 - Übung

▪ Fall 1 wenn $\text{exp} \neq 0$ und $\neq 255$ (normalized)

- Allgemeine Zahl: $V = (-1)^s * M * 2^E$
- $M = 1.f_{22} \dots f_{1f0}$
- $E = \text{exp} - 127$

▪ Übung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

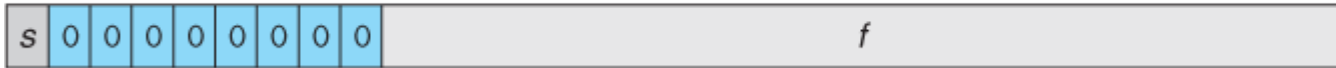
▪ TODO: Wert in Dezimal berechnen

Data rep - floating point - IEEE 754 - float 3/4

▪ Fall 2 wenn $\text{exp}=0$ (denormalized)

- Allgemeine Zahl: $V = (-1)^s * M * 2^E$
- Float single precision 32bit (4 Byte)

2. Denormalized

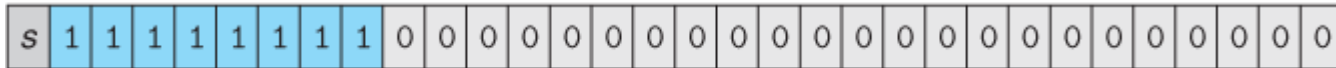


- Sign bit (one Bit s): Bit 31
 - $s=1 \rightarrow$ negativ, $s=0 \rightarrow$ positiv
- Exp field (k Bits): 8 Bits (Bits 30 – 23)
 - Codiert exponent $E = 1 - \text{Bias}$
 - Bias=127 $\rightarrow E=-126$
- Fraction field (n Bits): 23 Bits (Bits 22 – 0)
 - Codiert significand $M = 0.f_{22}...f_1f_0 \rightarrow M = 0...1$
- **Anwendung: 0 oder kleine Zahlen repräsentieren**

Data rep - floating point - IEEE 754 - float 4/4

- Float single precision 32bit (4 Byte)
 - **3a Fall wenn $\text{exp}==255$ und $\text{frac}==0$ (special value, infinity)**

3a. Infinity

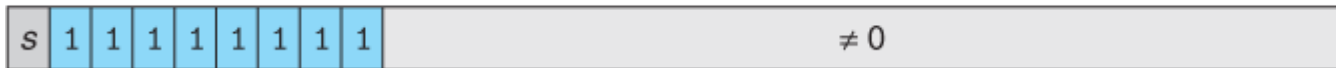


- Sign bit (one Bit s): Bit 31

- $s=1 \rightarrow -\infty$, $s=0 \rightarrow +\infty$

- **3b Fall wenn $\text{exp}==255$ und $\text{frac}!=0$ (special value, NaN)**

3b. NaN



- NaN (not a number)

Data rep - floating point - IEEE 754 – double, extended double

- Gleitkommazahlen mit doppelter Genauigkeit (double)
 - 1 Bit sign
 - 11 Bits exponent (bias ist 1023)
 - 52 Bit signifcand
- Gleitkommazahlen mit erweiterter Genauigkeit (extended double)
 - 1 Bit sign
 - $e \geq 15$ Bits exponent (bias entsprechend $2^{e-1} - 1$)
 - ≥ 63 Bit signifcand
- ansonsten gleiche Logik

Data rep - floating point - rounding

- nicht alle gebrochenen Zahlen innerhalb der Wertegrenzen sind mit floating point darstellbar
- Beispiel
 - darstellbar: 0.5
 - darstellbar: 12345
 - nicht darstellbar: 0.1
- Rounding = Bestimme eine geeignete darstellbare Zahl x^* für eine möglicherweise nicht darstellbare Zahl x .
 - x^* kann eine der beiden darstellbaren Zahlen x^+, x^- sein, für die gilt:

$$x^- \leq x \leq x^+$$

- **verschiedene** Rundungsregeln sind möglich

Data rep - floating point - conv dec to fp

- Formeln für Fall 1. normalisiert
 - Allgemeine Zahl:

$$V = (-1)^s * M * 2^E$$
 - $M = 1.f_{22}...f_1 f_0 = 1.m$
 - $E = \text{exp} - 127 \rightarrow \text{exp} = E + 127$

Algorithmus

- 1. Vorzeichen bestimmen (neg - $s=1$, pos $\rightarrow s=0$)
- 2. Umwandeln dec \rightarrow bin
 - 2a Vorkomma Teil
 - 2b Nachkomma Teil
- 3. Normalisieren
- 4. m bestimmen
- 5. exp bestimmen

- Beispiel 18,75 \rightarrow float
- 1.: pos $\rightarrow s=0$
 - 2a.: 18 $\rightarrow 0b10010$
 - 2b: 0.75 $\rightarrow 0b0.11$
- 3.: 10010.11 \rightarrow
 $1.001011 * 2^4$
- 4.: $M = 1.001011$
 $\rightarrow m = 001011$
- 5.: $4 = \text{exp} - 127 \rightarrow \text{exp} = 131$
 $\rightarrow \text{exp} = 10000011$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

\rightarrow in hex: 0x41960000

Data rep - floating point - conv dec to fp - Übung

- Formeln für Fall 1. normalisiert
 - Allgemeine Zahl:

$$V = (-1)^s * M * 2^E$$
 - $M = 1.f_2f_3...f_{23}$ $f_0 = 1$
 - $E = \text{exp} - 127 \rightarrow \text{exp} = E + 127$
- Übung 17,625 -> float
 - 1.: pos -> s=?
 - 2a.: 17 -> 0b10001
 - 2b.: 0.625 -> 0b101

Algorithmus

- 1. Vorzeichen bestimmen (neg -> s=1, pos->s=0)
- 2. Umwandeln dec -> bin
 - 2a Vorkomma Teil
 - 2b Nachkomma Teil
- 3. Normalisieren
- 4. m bestimmen
- 5. exp bestimmen

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

-> in hex: 0x