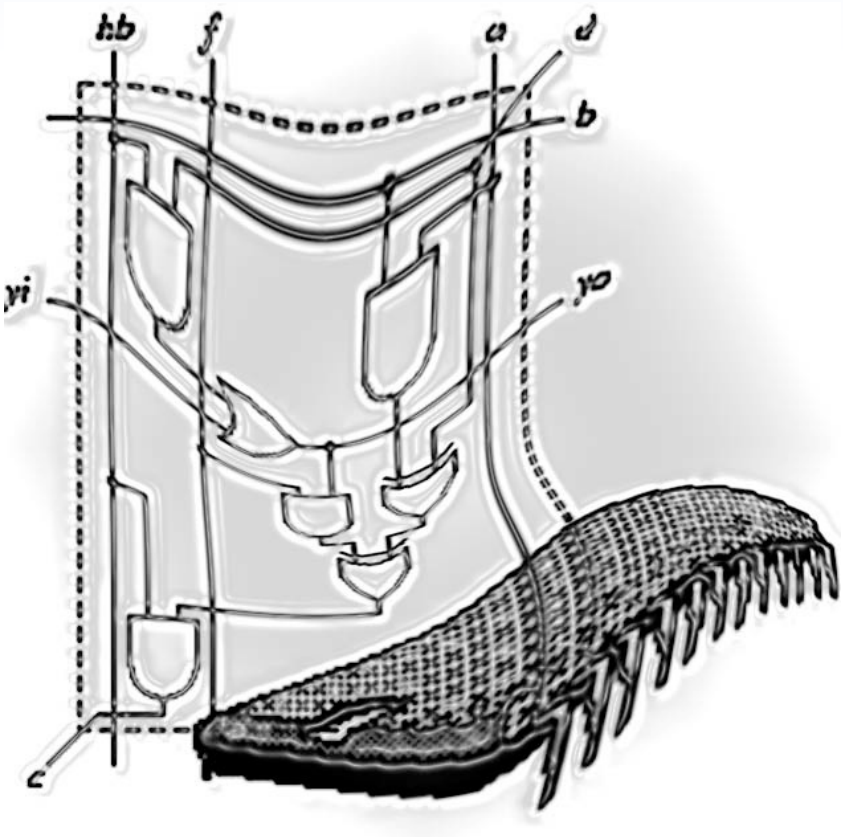
Technische Informatik I

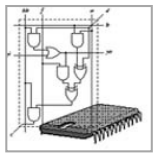


Kapitel 1

Zahlendarstellung

Prof. Dr. Dirk W. Hoffmann

Hochschule Karlsruhe ◆ University of Applied Sciences ◆ Fakultät für Informatik



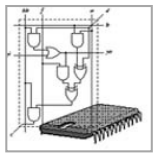
Zahlensystemkonvertierung

- Motivation
  - Jede nichtnegative Zahl z lässt sich in der Form

$$z = \sum_{i=0}^{n} a_i b^i$$

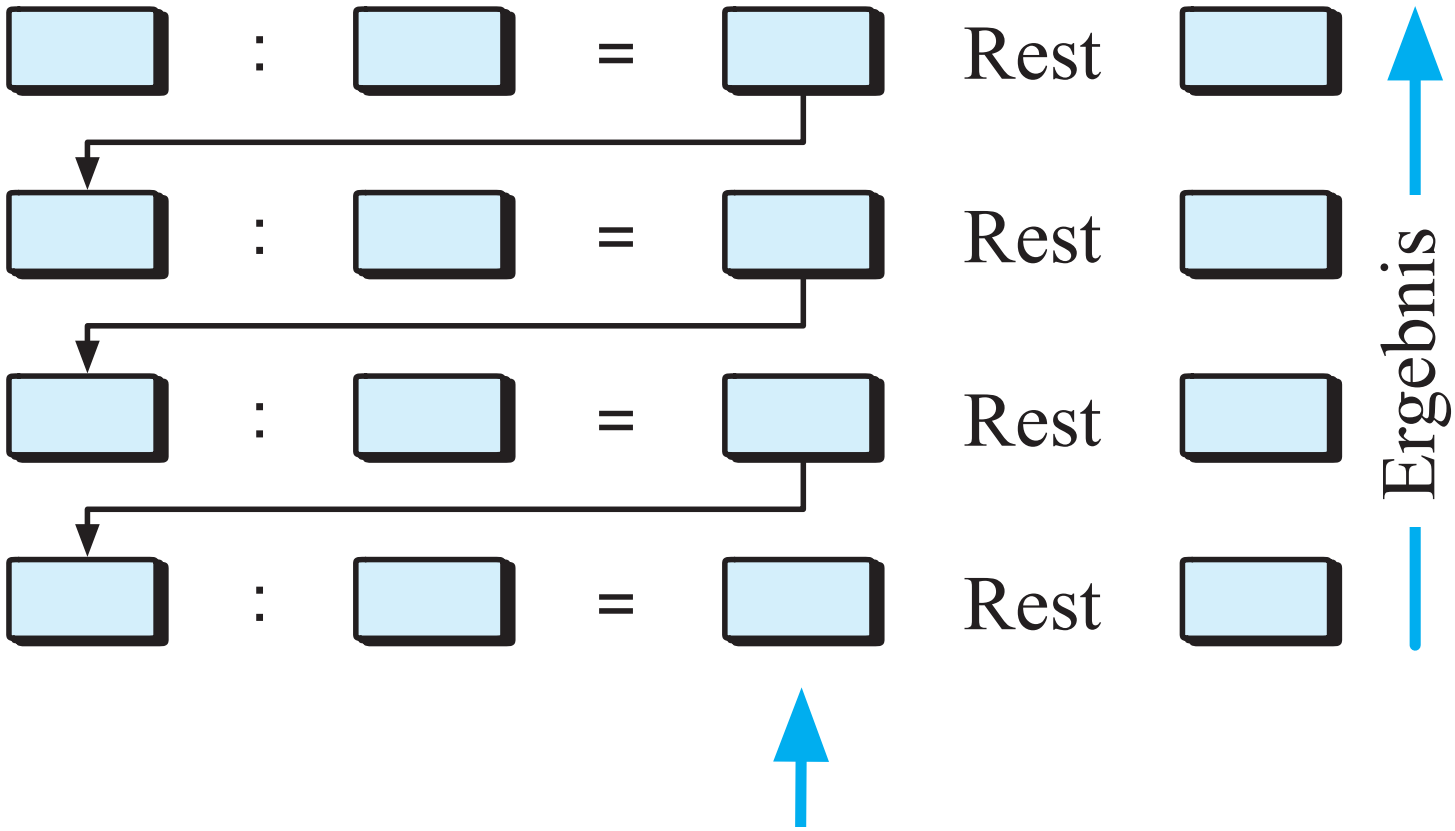
darstellen mit

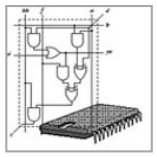
- $0 \leq a_i < b$
- *b* heißt die *Basis* der Zahlendarstellung



Konvertierung zwischen Zahlensystemen

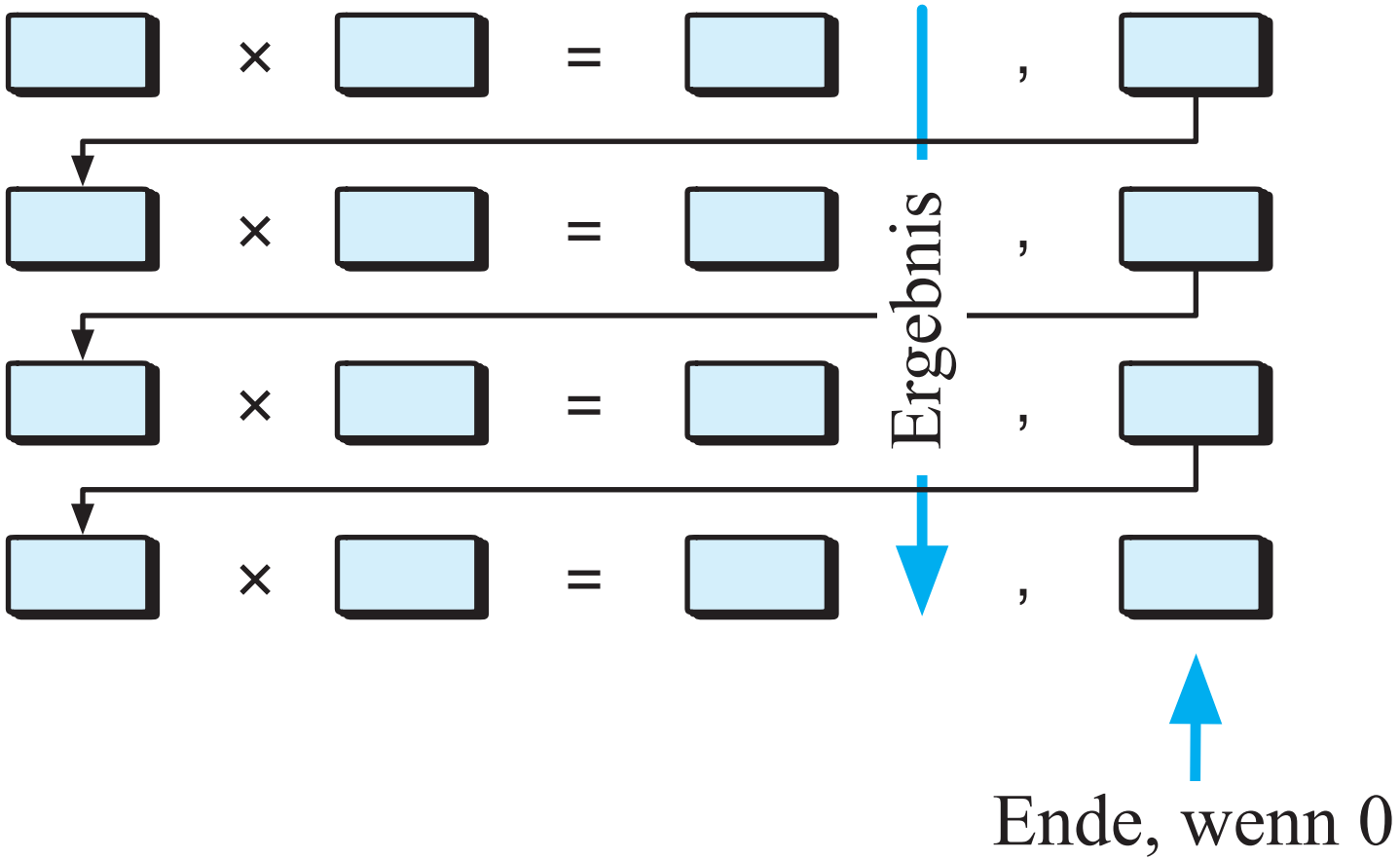
- Verfahren für ganze Zahlen





Konvertierung zwischen Zahlensystemen

- Erweiterung: Konvertierung des Nachkommaanteils



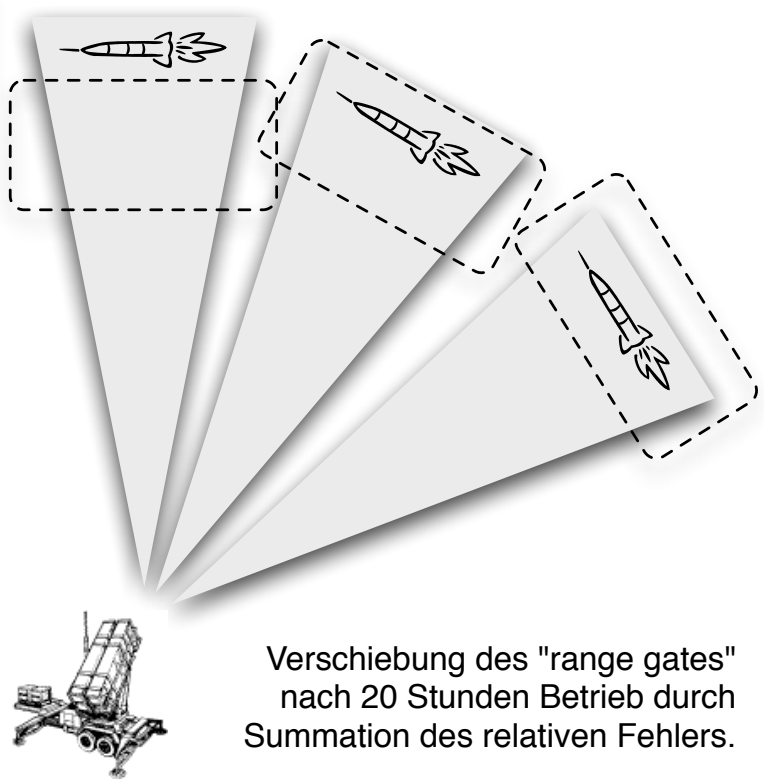
Das Patriot-Abwehrsystem

- Einsatzgebiete
  - Abwehr von Flugzeugen und Cruise Missiles
  - Seit Anfang der 90er auch gegen Short Range Ballistic Missiles
- Golfkrieg I
  - Einsatz am 25.2.1991
  - Eine Patriot-Rakete verfehlt eine irakische Scud-Rakete aufgrund eines Software-Fehlers
    - Einschlag in eine amerikanische Kaserne in Saudi-Arabien
    - 28 Tote, 100 Verletzte

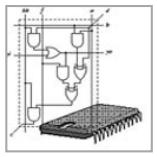
Das Patriot-Abwehrsystem

- Wie werden Zielobjekte erfasst und verfolgt?
  - Zur Zielobjektverfolgung wird permanent der Zielkorridor berechnet
  - Verwendete Parameter zur Berechnung
    - Auswertung der Radardetektion
    - Zeit der letzten Radardetektion seit Inbetriebnahme in Sekunden
  - Zur Berechnung der Zeit in Sekunden wird die Systemzeit, die in Zehntelsekunden gespeichert ist, mit 0,1 multipliziert

Wie wird 0,1 im Binärsystem dargestellt?

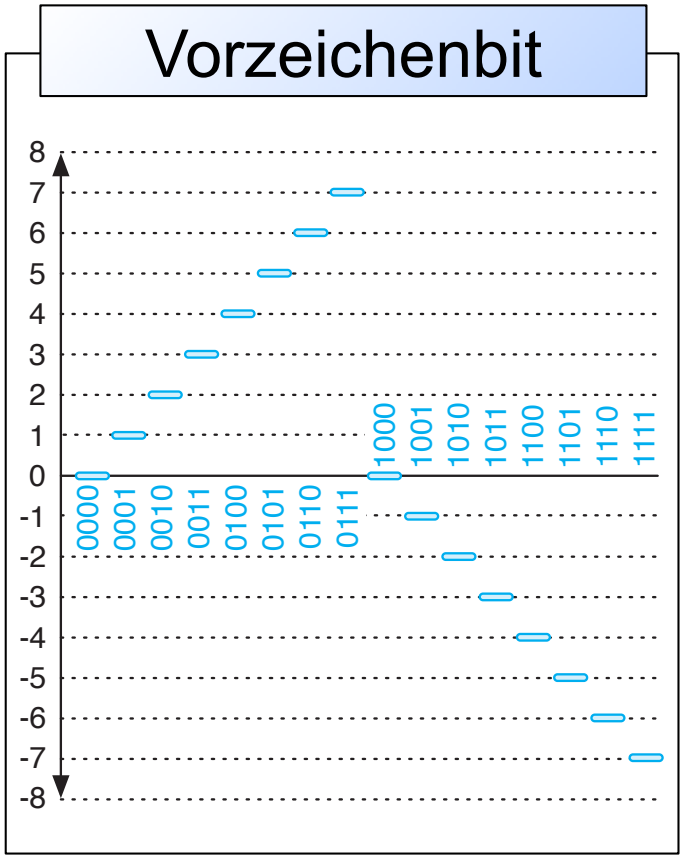


Verschiebung des "range gates" nach 20 Stunden Betrieb durch Summation des relativen Fehlers.

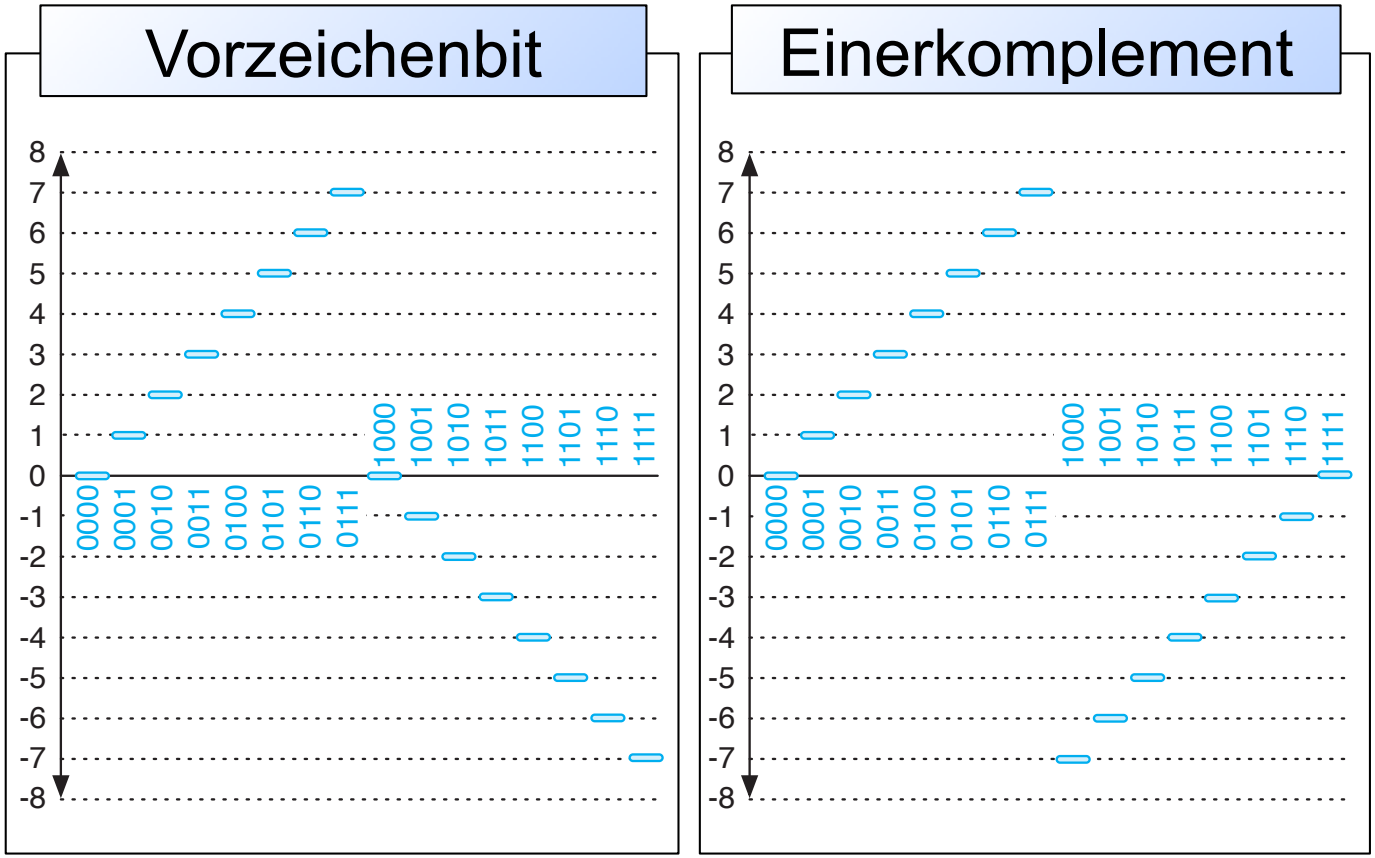


Darstellung ganzer Zahlen

- Unterscheidung positiver und negativer Zahlen
  - Vorzeichenbit
    - Vorzeichen wird durch ein einzelnes Bit definiert
  - Einerkomplement
    - Negative Zahlen werden durch das Invertieren aller Bits gebildet
  - Zweierkomplement
    - Negative Zahlen: Invertieren und Addieren von 1



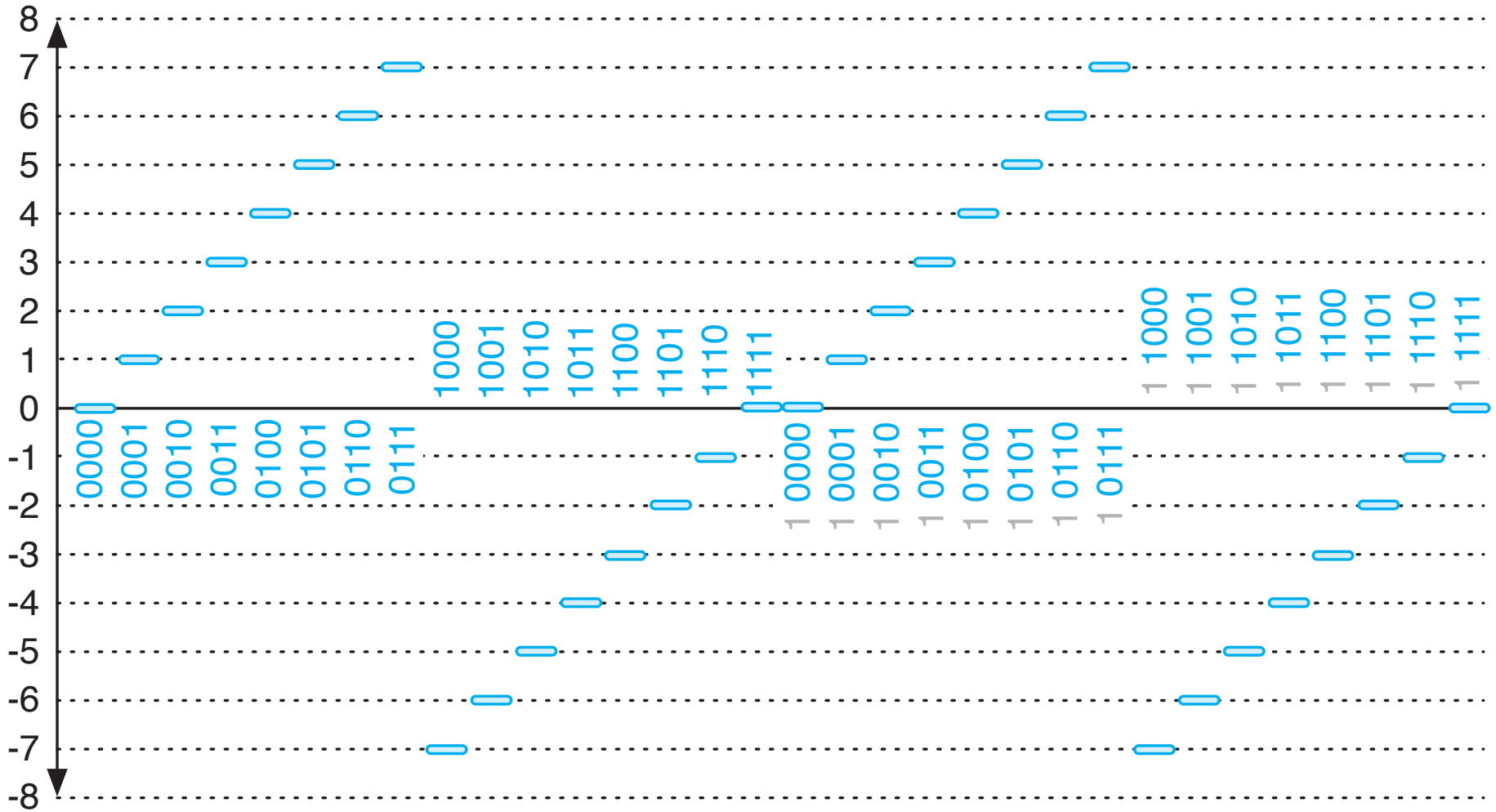
- Vorzeichenbitdarstellung
  - Speicherung des Zahlenwerts als
    - Vorzeichen (das am weitesten links stehende Bit)
    - und Betrag (die restlichen Bits)

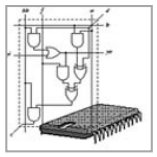


- Einerkomplement
  - Speicherung negativer Zahlen durch das Invertieren aller Bits

Addition im Einerkomplement

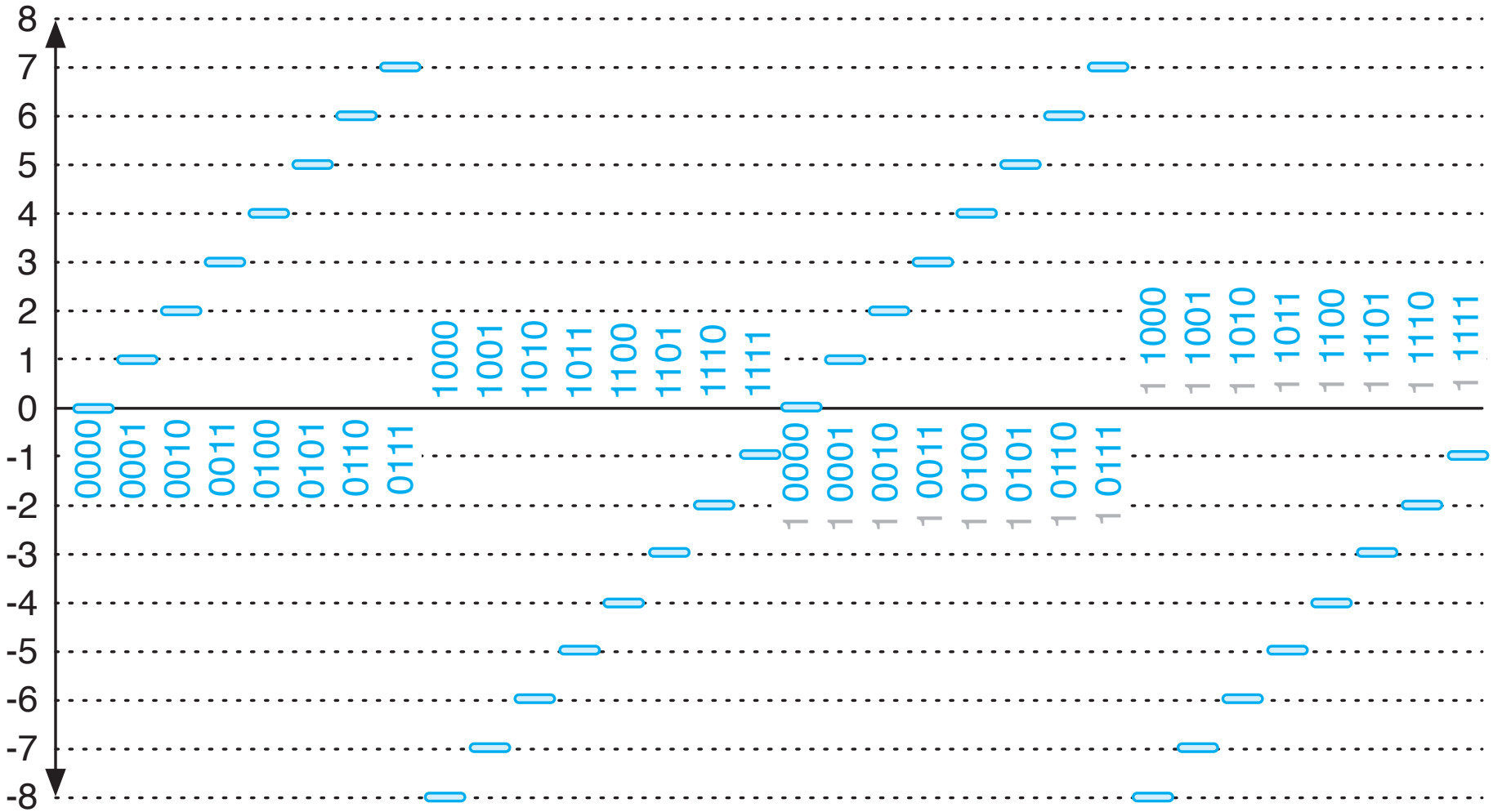
- Probleme bei der Addition
  - Der negative Zahlenbereich geht fast nahtlos in den positiven über
  - Durch die Doppeldarstellung der Null ist das Ergebnis um 1 zu klein



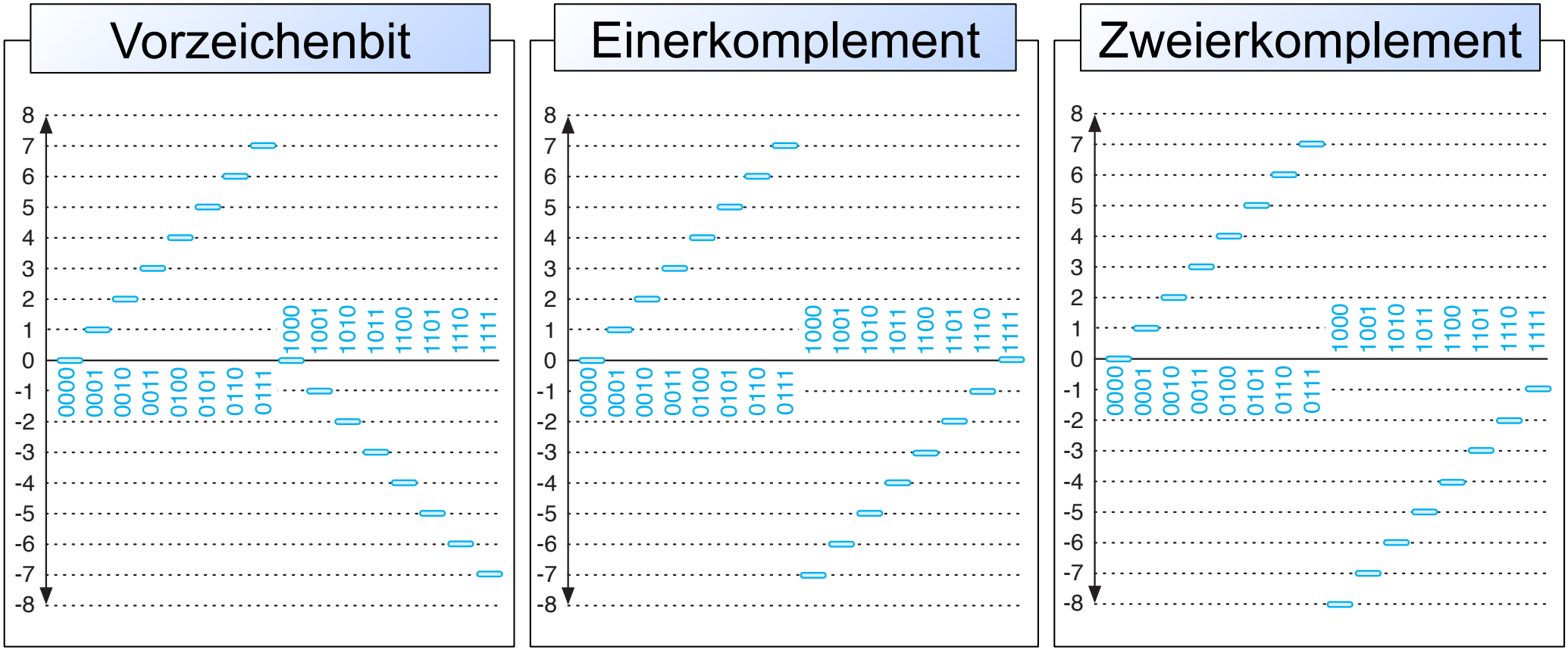


Übergang zum Zweierkomplement

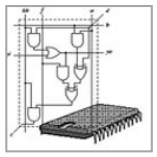
- Wie kann die Addition vereinfacht werden?
  - Idee: Der Zahlenstrahl wird begradigt
  - Alle negative Zahlen werden um 1 abgesenkt (☞ Zweierkomplement)



Darstellung ganzer Zahlen



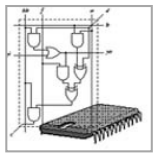
- Zweierkomplement
  - Speicherung negativer Zahlen durch
    - das Bilden des Einerkomplements
    - und anschließender Addition von 1



Binary Coded Decimals (BCD)

- BCD = Binärcodierte Dezimalziffern
  - Dezimalzahlen werden Ziffer für Ziffer codiert
  - Jede Ziffer wird durch 4 Bits dargestellt
  - Von den 16 Bitkombinationen werden nur 10 benötigt
    - Die Bitfolgen 1010 .. 1111 bleiben unbenutzt

Die BCD-Tabelle			
0000 = 0	0100 = 4	1000 = 8	(1100 = C)
0001 = 1	0101 = 5	1001 = 9	(1101 = D)
0010 = 2	0110 = 6	(1010 = A)	(1110 = E)
0011 = 3	0111 = 7	(1011 = B)	(1111 = F)



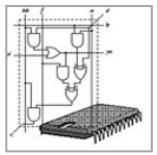
Festkommazahlen

- Format

b_{31}	b_{30}	b_{29}	b_{28}	b_{27}		b_1	b_0
$\pm$	2^{-1}	2^{-2}	2^{-3}	2^{-4}		2^{-30}	2^{-31}

$$x = (-1)^{b_{N-1}} \sum_{k=1}^{N-1} b_{N-1-k} \cdot 2^{-k}$$

- Darstellbarer Zahlenbereich:]-1;1[
- Konstanter Abstand zwischen benachbarten Zahlen (2^{-N+1})
- Zahlenbereich ist bezüglich der Multiplikation abgeschlossen
  - Produkt zweier Zahlen aus dem Intervall]-1;1[liegt wieder in]-1;1[



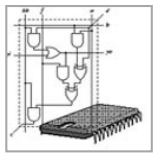
Gleitkommazahlen

- Motivation
  - Jede rationale Zahl lässt sich in der folgenden Form darstellen

$$(-1)^{Vz} \cdot 0{,}M \cdot 2^{E}$$

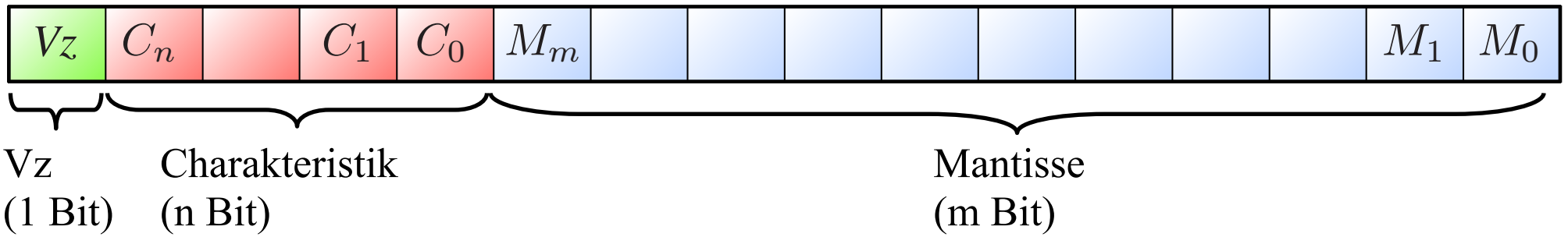
    - *Vz* ist das Vorzeichenbit (0 = „+“, 1 = „-“)
    - *M* heißt Mantisse
    - *E* heißt Exponent

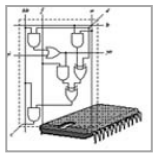
- Beispiele
  - $1010000000000_2 = (-1)^0 \cdot 0{,}101_2 \cdot 2^{(13)}$
  - $-1010000000000_2 = (-1)^1 \cdot 0{,}101_2 \cdot 2^{(13)}$
  - $0{,}00001010001_2 = (-1)^0 \cdot 0{,}1010001_2 \cdot 2^{(-4)}$
  - $-0{,}00001010001_2 = (-1)^1 \cdot 0{,}1010001_2 \cdot 2^{(-4)}$



Gleitkommazahlen

- Wie wird der Exponent gespeichert?
  - Der Exponent ist vorzeichenbehaftet, wird aber als vorzeichenlose Zahl repräsentiert
  - Dazu wird der um eine Konstante k erhöhte Exponent gespeichert
    - Charakteristik C = E + k
  - Beispiel: Charakteristik C = 8 Bit, k = 127
    - C = 255 ⇒ E = 128
    - C = 127 ⇒ E = 0
    - C = 0 ⇒ E = -127
- Interne Darstellung von Gleitkommazahlen





Normalisierung

- Problem: Die Gleitkommadarstellung ist nicht eindeutig!

$$0.01010111 \times 2^{14}$$
$$00.1010111 \times 2^{13}$$
$$001.010111 \times 2^{12}$$
$$0010.10111 \times 2^{11}$$

  - Abhilfe: Einheitliche Darstellung durch Normalisierung
- Normierungsregel 1
  - Die erste Nachkommastelle enthält die höchste Ziffer ungleich 0
  - Für unser Beispiel: 0.1010111×2^{13}
- Normierungsregel 2
  - Die erste Vorkommastelle enthält die höchste Ziffer ungleich 0
  - Für unser Beispiel: 1.010111×2^{12}

Normalisierung

- Beispiel: $0{,}001_2$ (4 Bit Exponent, C = E + 7)

Normalisierungsregel 1: $0{,}001_2 = 0{,}1_2 \times 2^{-2}$

Normalisierungsregel 2: $0{,}001_2 = 1{,}0_2 \times 2^{-3}$

Normalisierung

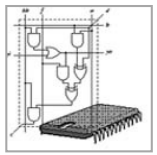
- Optimierung
  - Die höchste Ziffer ist immer 1 und muss nicht gespeichert werden
  - Die eingesparte Ziffer heißt verstecktes Bit (implizite Darstellung)
- Beispiel: $0{,}001_2$ (4 Bit Exponent, C = E + 7)

Normalisierungsregel 1: $0{,}001_2 = 0{,}1_2 \times 2^{-2}$ (explizite Darstellung)

Normalisierungsregel 1: $0{,}001_2 = 0{,}1_2 \times 2^{-2}$ (implizite Darstellung, 1 Bit versteckt)

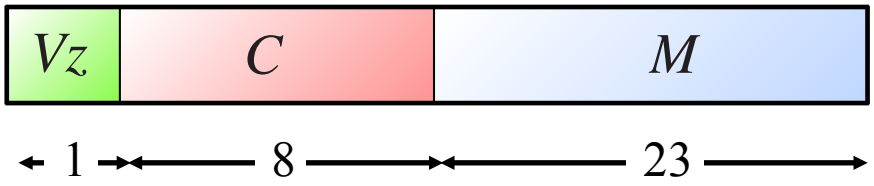
Normalisierungsregel 2: $0{,}001_2 = 1{,}0_2 \times 2^{-3}$ (explizite Darstellung)

Normalisierungsregel 2: $0{,}001_2 = 1{,}0_2 \times 2^{-3}$ (implizite Darstellung, 1 Bit versteckt)



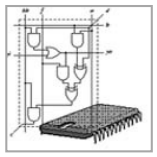
Gleitkommazahlen nach IEEE 754

- IEEE 754 Floating Point Standard
  - 1985 zur Verbesserung der Software-Kompatibilität verabschiedet
- Zwei Formate
  - Single Precision Format (32 Bit)



    - C = E + 127 Reserviert: C = 000…0_2, C = 111…1_2
  - Double Precision Format (64 Bit)

    - C = E + 1023 Reserviert: C = 000000…0_2, C = 111111…1_2



Gleitkommazahlen nach IEEE 754

Charakteristik C	Mantisse M	Single-precision-Format	Double-precision-Format
$000\ldots000$	beliebig	$(-1)^{Vz} \cdot 0,M \cdot 2^{-126}$	$(-1)^{Vz} \cdot 0,M \cdot 2^{-1022}$
$0 < C < 2^n - 1$	beliebig	$(-1)^{Vz} \cdot 1,M \cdot 2^{C-127}$	$(-1)^{Vz} \cdot 1,M \cdot 2^{C-1023}$
$111\ldots111$	$= 0$	$(-1)^{Vz} \cdot \infty$	$(-1)^{Vz} \cdot \infty$
$111\ldots111$	$\neq 0$	Not a Number (NaN)	Not a Number (NaN)

Sonderfälle: C = 000..000 Nichtnormalisierte Darstellung

C = 111..111 ± ∞ (1 / 0, -1 / 0)

NaN (0 × ∞, ∞ - ∞, ∞ / ∞)

Übungsaufgabe

Beim Debuggen eines Programms stoßen Sie im Arbeitsspeicher auf das folgende 32 Bit breite Datenwort:

C0	68	00	00

Berechnen Sie den dezimalen Wert des Datenworts,
wenn es als IEEE 754 Fließkommawert
einfacher Genauigkeit (single precision format) interpretiert wird.