WBS TRAINING

Programmierung(2)

UNSER ♥ SCHLÄGT FÜR BILDUNG

Agenda

- **Interne Zahlendarstellung**
  - Positive Ganzzahlen
  - Negative Ganzzahlen
  - Gleitkommazahlen (nach IEEE 754)

- **Fachpraktische Anwendungen**

Interne Zahlendarstellung – Ganzzahlen

- In ANSI C (und vielen anderen Programmiersprachen) besitzt ein Integer (mindestens) **4 Byte** Speicherplatz.
- Da 4 Byte = 4x8 Bit = **32 Bit** können also 2^{32} unterschiedliche Zahlen dargestellt werden.
- Würde man sich nur für **positive Zahlen** interessieren, so könnten dies die Zahlen von 1 bis 2^{32} sein.
- Um aber auch negative Zahlen und 0 darstellen zu können, wird stattdessen der Bereich zwischen $\mathbf{-2^{31}}$ und $\mathbf{2^{31}-1}$ gewählt (also aufgeteilt in die beiden gleich großen Bereiche -2^{31} bis -1 sowie 0 bis 2^{31} -1).
- Da bei der internen (und also binären) Darstellung von Zahlen nur die Bitzustände 0 oder 1 zur Verfügung stehen, fehlt aber zunächst eine Darstellungsmöglichkeit des **Vorzeichens**.
- Wir werden dieses Problem (unter anderem) mittels eines sogenannten „**Vorzeichenbits**“ lösen.
- Dies wird dann auch deutlich machen, worin der (relativ große) **Unterschied** bei der internen Darstellung von positiven und negativen Ganzzahlen besteht.

Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

Interne Zahlendarstellung – **Positive** Ganzzahlen

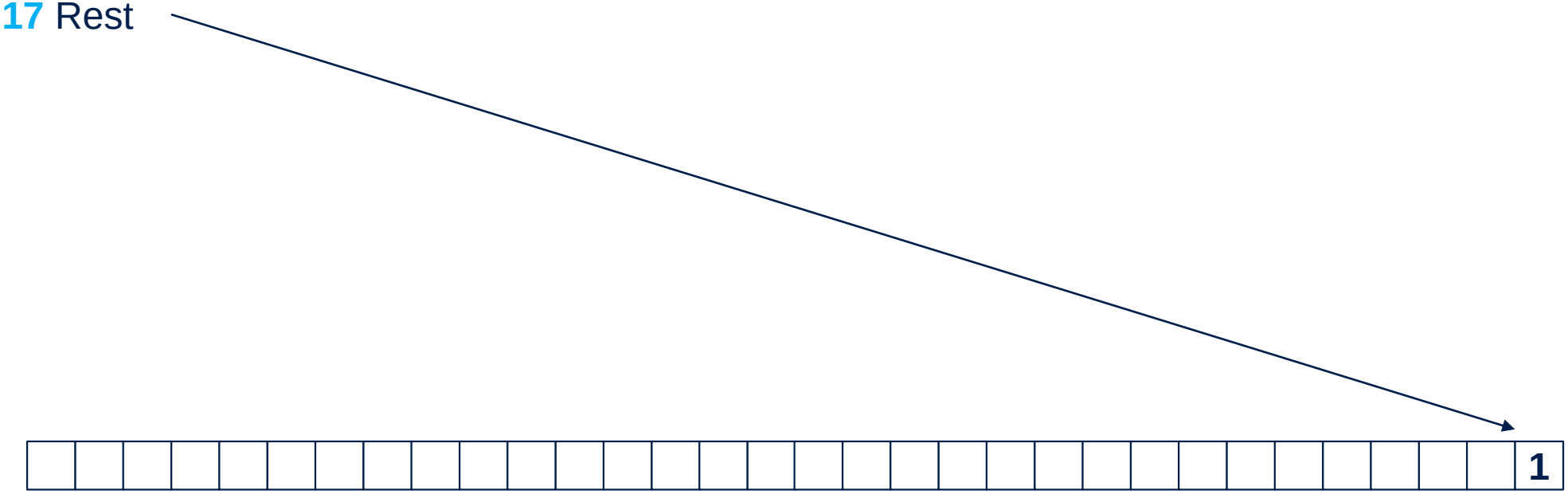
- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

35:2 = **17** Rest
1

Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

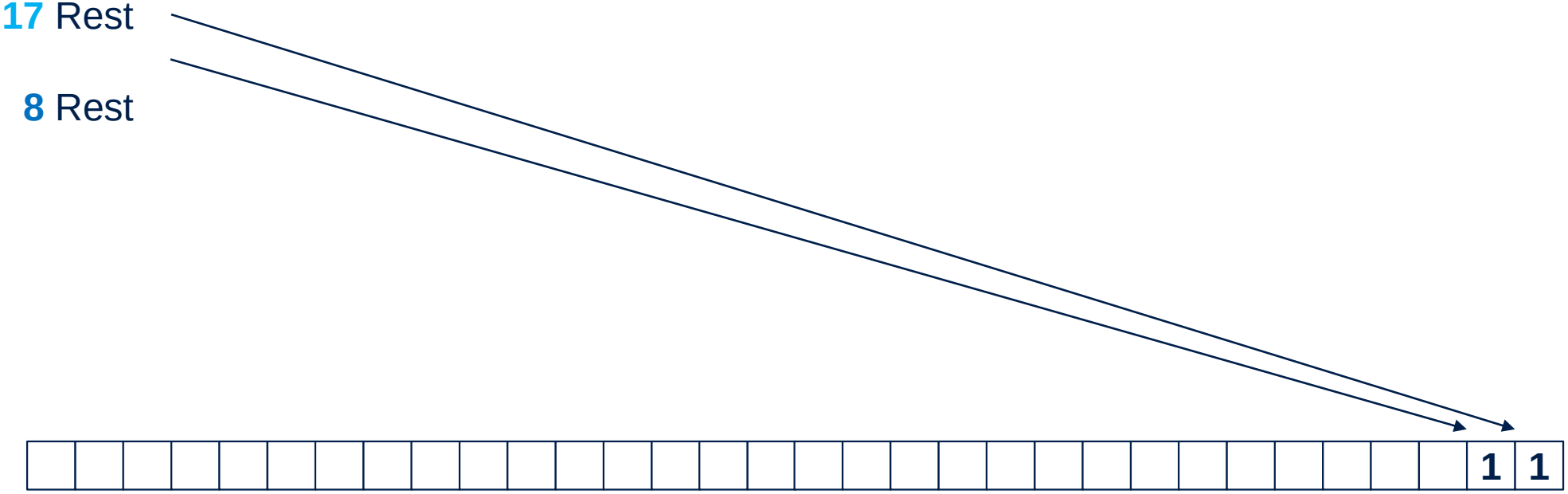
35:2 = **17** Rest **1**



Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

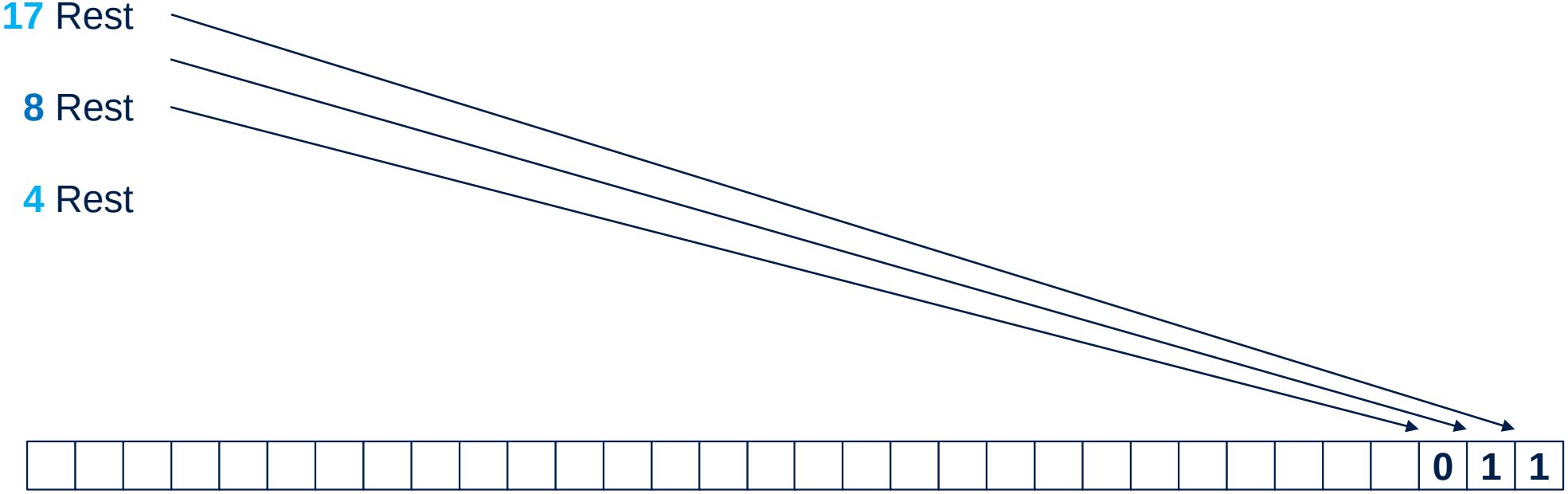
35:2 = **17** Rest **1**
17:2 = **8** Rest **1**



Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

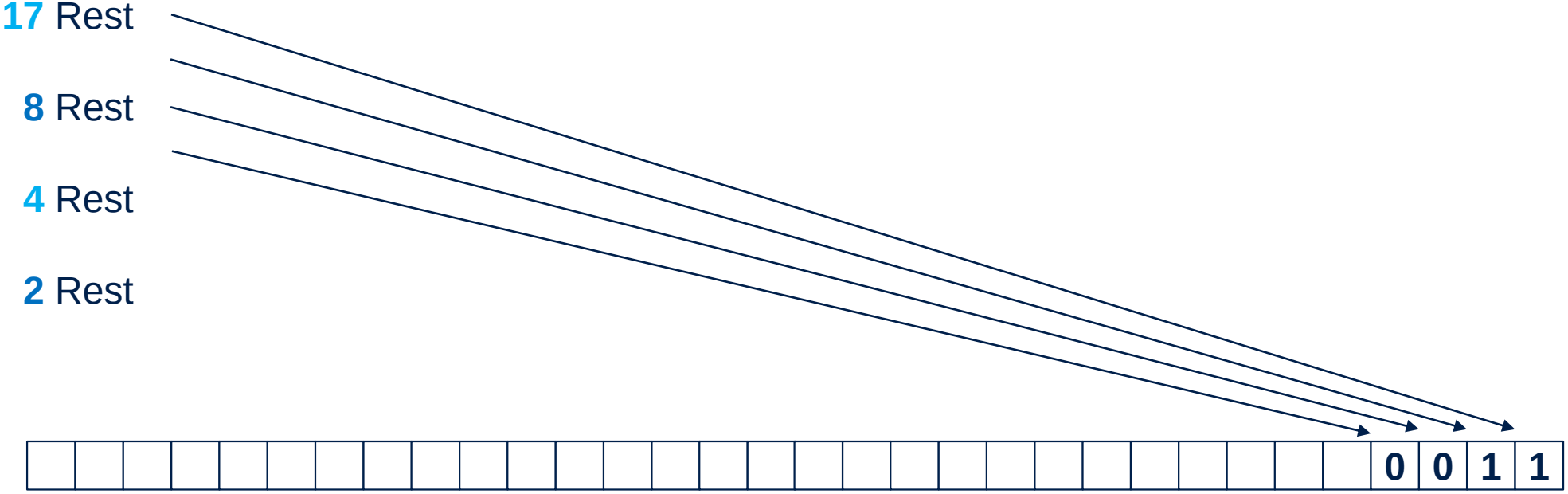
35:2 = **17** Rest **1**
17:2 = **8** Rest **1**
8:2 = **4** Rest **0**



Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

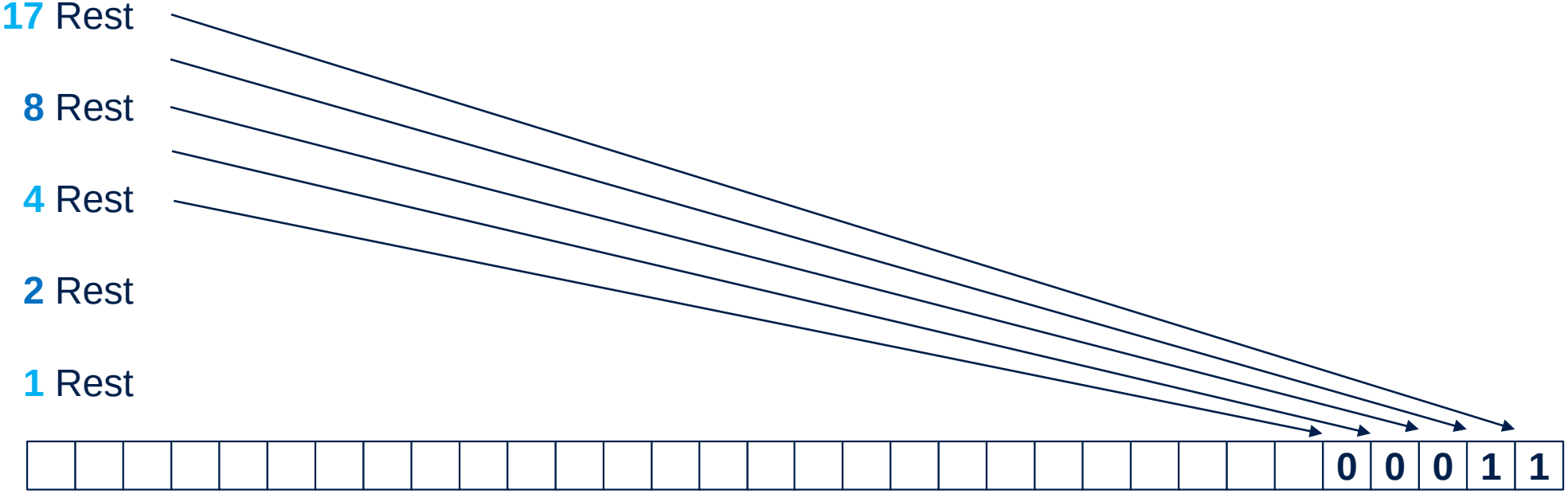
35:2 = **17** Rest **1**
17:2 = **8** Rest **1**
8:2 = **4** Rest **0**
4:2 = **2** Rest **0**



Interne Zahlendarstellung – **Positive** Ganzzahlen

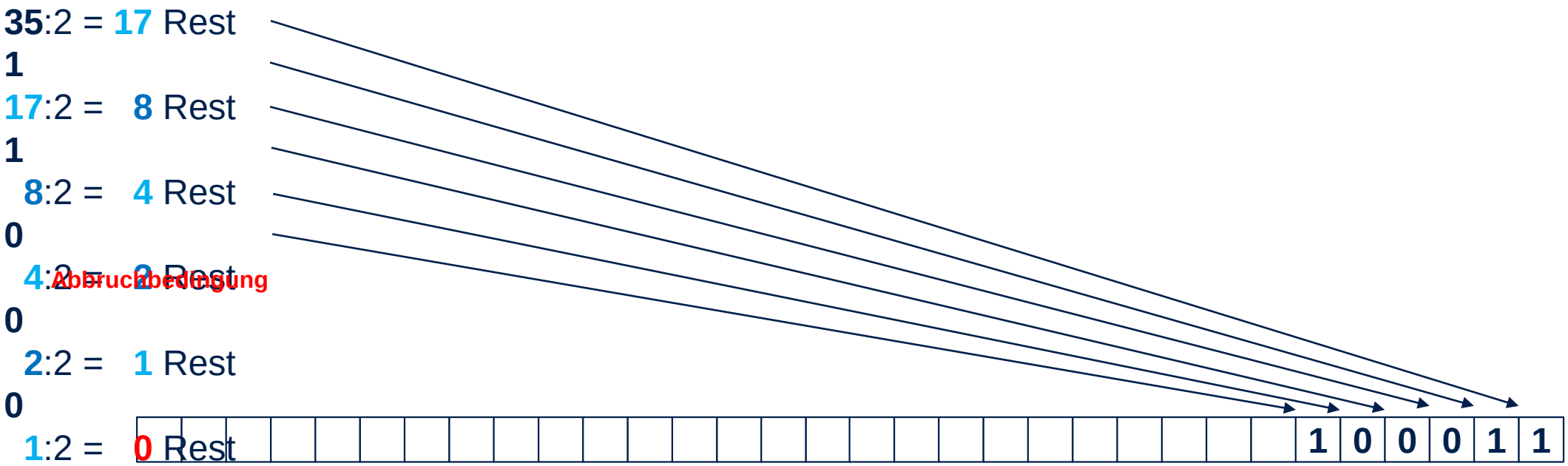
- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

35:2 = **17** Rest
1
17:2 = **8** Rest
1
8:2 = **4** Rest
0
4:2 = **2** Rest
0
2:2 = **1** Rest
0



Interne Zahlendarstellung – **Positive** Ganzzahlen

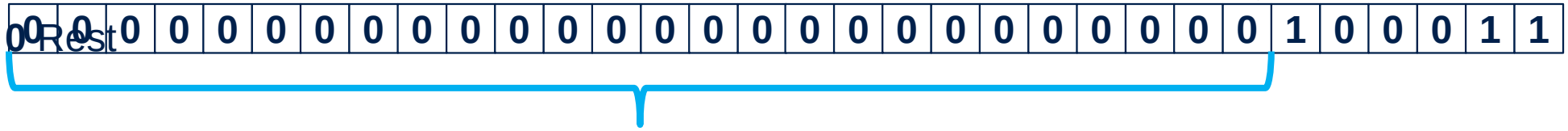
- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:



Interne Zahlendarstellung – **Positive** Ganzzahlen

- Sie haben bereits in vorangegangenen Bausteinen kennengelernt, wie man eine dezimal dargestellte Zahl in ihre binäre Darstellung umrechnet.
- Als Auffrischung betrachten wir als erstes Beispiel die **dezimal** dargestellte Zahl **35**:

35:2 = **17** Rest **1**
17:2 = **8** Rest **1**
8:2 = **4** Rest **0**
4:2 = **2** Rest **0**
2:2 = **1** Rest **0**
1:2 = **0** Rest **1**



Alle noch nicht gefüllten Felder erhalten den Wert **0**

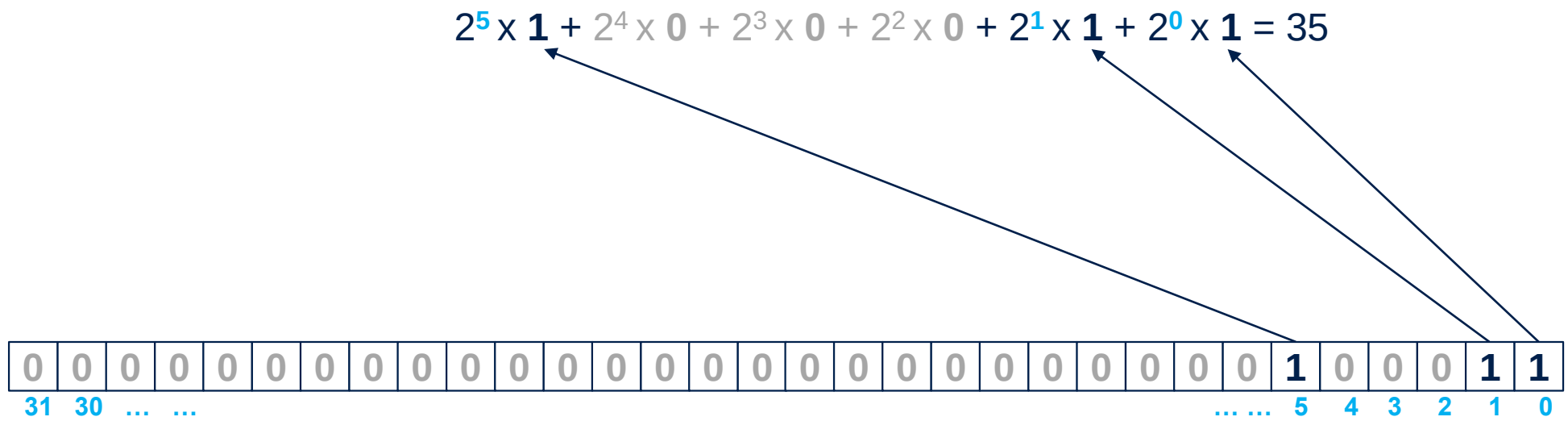
Interne Zahlendarstellung – **Positive** Ganzzahlen

- Auch die umgekehrte Richtung, also die Umrechnung ausgehend von einer binär dargestellten Zahl, hin zu deren dezimalen Darstellung sei noch einmal in Erinnerung gerufen:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

Interne Zahlendarstellung – **Positive** Ganzzahlen

- Auch die umgekehrte Richtung, also die Umrechnung ausgehend von einer binär dargestellten Zahl, hin zu deren dezimalen Darstellung sei noch einmal in Erinnerung gerufen:



Interne Zahlendarstellung – **Negative** Ganzzahlen

- Bei der internen Darstellung von negativen Ganzzahlen nutzen wir das erste Bit als „**Vorzeichenbit**“.
- Falls das Vorzeichenbit **1** ist, so wird eine **negative Zahl** dargestellt.
- Falls es hingegen **0** ist, so wird eine **positive Zahl** dargestellt.
- Entsprechend ist die **größte positive 32-Bit-Integer-Zahl**:

2^{31}-1 =

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- Man könnte nun annehmen, dass sich die interne Darstellung von **–x** und **+x** stets nur durch das Vorzeichenbit unterscheiden müsse. Dies hätte allerdings zur Folge, dass es **zwei** unterschiedliche Darstellungen für die Zahl 0 gäbe, nämlich ein „negatives 0“ und ein „positives 0“:

-0 =

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+0 =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Die im Folgenden vorgeführte Darstellung negativer Ganzzahlen wird dieses Problem vermeiden:

Interne Zahlendarstellung – **Negative** Ganzzahlen

- **Die Berechnung der interne Darstellung einer negativen Zahl zeigen wir am Beispiel -35 :**

1.) Wir berechnen zunächst die binäre Darstellung von **+**35, die uns ja schon bekannt ist:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

2er-Komplement

2.) Wir bilden das „1er-Komplement“ (alle Nullen und Einsen werden einfach ausgetauscht)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0

3.) Wir addieren 1 (falls es an der Stelle ganz links zum Überlauf käme, so würde dieser ignoriert)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1

Das Ergebnis des dritten Schrittes ist dann bereits die gesuchte Lösung!

Der Vollständigkeit halber ermitteln wir nun auch noch die Darstellung von „- 0“ :

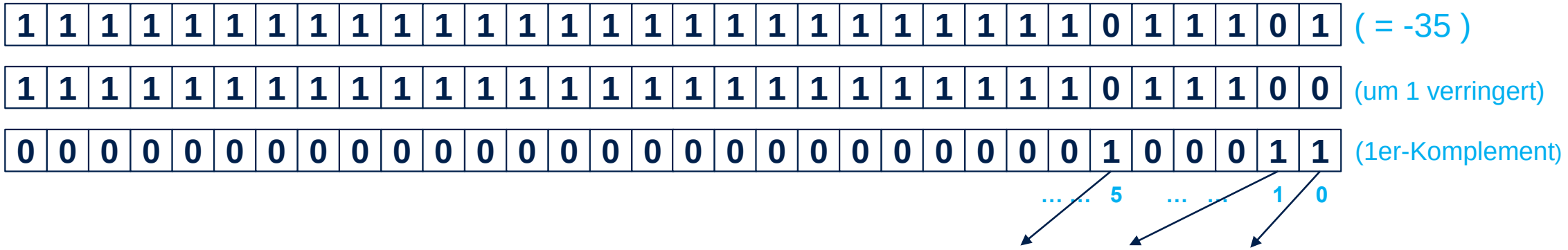
+0 =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1er-Komplement:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-0 =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

+1

Die Darstellung von „-0“ und „+0“ ist also identisch!

Interne Zahlendarstellung – **Negative** Ganzzahlen

- Falls hingegen die Aufgabe darin besteht, eine binär dargestellte negative Ganzzahl wieder in eine **dezimale Darstellung umzuwandeln**, so müssen zunächst die selben 3 Schritte in umgekehrter Reihenfolge durchgeführt werden:



Wir erhalten ein (positives) **dezimales (Zwischen)-Ergebnis**: $2^5 \times 1 + 2^1 \times 1 + 2^0 \times 1 = \mathbf{35}$

Für das **Endergebnis** muss dieses Zwischenergebnis lediglich um das Vorzeichen „-“ ergänzt werden:

Interne Zahlendarstellung – **Negative** Ganzzahlen

- Falls hingegen die Aufgabe darin besteht, eine binär dargestellte negative Ganzzahl wieder in eine **dezimale Darstellung umzuwandeln**, so müssen zunächst die selben 3 Schritte in umgekehrter Reihenfolge durchgeführt werden:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	(= -35)
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	(um 1 verringert)
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	(1er-Komplement)

Wir erhalten ein (positives) **dezimales (Zwischen)-Ergebnis**: $2^5 \times 1 + 2^1 \times 1 + 2^0 \times 1 = \mathbf{35}$

Für das **Endergebnis** muss dieses Zwischenergebnis lediglich um das Vorzeichen „-“ ergänzt werden: **-35**

Interne Zahlendarstellung – **Negative** Ganzzahlen

- Wem die Bildung eines **1er-Komplimentes** und das **Addieren von 1** reichlich aufwendig erscheint (und die Vermeidung einer doppelten Darstellbarkeit der 0 vergleichsweise unbedeutend) dem sei im Folgenden noch eine weitere Motivation für diese Vorgehensweise aufgezeigt.
- Tatsächlich kann man nun beliebige Zahlen addieren, **ohne** auf deren Vorzeichen achten zu müssen.
- Dies soll abschließend am Beispiel (**+**35) + (**-**35) gezeigt werden:

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	(= +35)
+	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	(= - 35)
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(= 0)

(Dieser Überlauf wird ignoriert)

Interne Zahlendarstellung – Gleitkommazahlen

- Wir können eine dezimal dargestellte Kommazahl auch in binärer Form darstellen.
- Allerdings wird es dabei in der Regel zu Rundungsfehlern kommen.
- Daher wird intern mit einer größeren Anzahl von Stellen gearbeitet, als ausgegeben werden.
- Dies hat dann zur Folge, dass die Rundungsfehler in der Regel erst an Stellen sichtbar werden, die gar nicht mehr zu den ausgegebenen zählen.
- Dennoch ist es empfehlenswert, in allen Fällen, in denen an Stelle von Gleitkommazahlen auch Ganze Zahlen verwendet werden könnten, diese vorzuziehen. (Beispiel: €-Preise nicht als Kommazahl, sondern als ganzzahlige Cent-Zahl.)
- Natürlich wird es aber auch Fälle geben, in denen Kommazahlen unverzichtbar sind. Wie diese intern dargestellt werden, wollen wir uns nun im Folgenden genauer anschauen:

Interne Zahlendarstellung – Gleitkommazahlen

- Bei den folgenden Erläuterungen werden wir die Darstellung von FLOAT-Zahlen betrachten.
- Dies bietet sich an, da diese weniger Stellen besitzen als dies bei DOUBLE-Zahlen der Fall ist.
- Das Prinzip der Darstellung wird allerdings in beiden Fällen identisch sein. Die geringfügigen Unterschiede bei Double-Zahlen werden wir daher erst im Anschluss ansprechen.

Hinweis:

Es wird sich zeigen, dass in diesem Fall zwei unterschiedliche Darstellungen für die Zahl „0“ existieren. Da das hier vorgestellte (und in den meisten Programmiersprachen gebräuchliche) Verfahren (IEEE 754) dadurch aber auf ein 2-er-Komlement verzichten kann, erscheint dies letztlich durchaus akzeptabel.

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

Vorzeichenbit

Vorzeichenbit=**0**: **positive** Zahl
Vorzeichenbit=**1**: **negative** Zahl

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

Exponent (Charakteristik)

(bei FLOAT-Zahlen üblicherweise **8-stellig** … zulässige Werte liegen zwischen **-126** und **127**)

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

Mantisse

(bei FLOAT-Zahlen üblicherweise 23-stellig)

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv

=> Vorzeichenbit = **0**

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1
=> 10 (dezimal) = **1010** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1
=> 10 (dezimal) = 1010 (binär) Zahlen, die nicht sehr nahe bei 0 liegen, können **normiert** dargestellt werden => führende 1 kann eingespart werden

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1
=> 10 (dezimal) = **1010** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 (=„Bias“)
=130 (dezimal)

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1
=> 10 (dezimal) = **1010** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 (=„Bias“)
=130 (dezimal) = **10000010** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   10:2 = 5 Rest 0
   5:2 = 2 Rest 1
   2:2 = 1 Rest 0
   1:2 = 0 Rest 1
   => 10 (dezimal) = **1010** (binär)

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
   (4-1) + 127 (=„Bias“)
   =130 (dezimal) = **10000010** (binär)

d) Nachkommawert -> binäre Darstellung
   0,375*2 = **0**,75
   0,75 *2 = **1**,5
   0,5 *2 = **1**,0

Interne Zahlendarstellung – Gleitkommazahlen

- Auch **FLOAT-Zahlen** werden mit **32 Bit** dargestellt. Diese werden in **3 Abschnitte** zerlegt:

1. Beispiel - Umrechnung der dezimal dargestellten Zahl **10,375** in Gleitkommadarstellung:

a) 10, 375 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   10:2 = 5 Rest 0
   5:2 = 2 Rest 1
   2:2 = 1 Rest 0
   1:2 = 0 Rest 1
   => 10 (dezimal) = **1010** (binär)

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
   (4-1) + 127 (=„Bias“)
   =130 (dezimal) = **10000010** (binär)

d) Nachkommawert -> binäre Darstellung
   0,375*2 = **0**,75
   0,75 *2 = **1**,5
   0,5 *2 = **1**,0
   0,0 *2 = **Abbruchbedingung** => **gefundene Ziffern** werden in die Mantisse eingetragen UND **restlichen Stellen** werden mit **0** aufgefüllt

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ

=> Vorzeichenbit = **1**

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 (=„Bias“)
=130 (dezimal)

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 (=„Bias“)
=130 (dezimal) = **10000010** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 („Bias“)
=130 (dezimal) = **10000010** (binär)

d) Nachkommawert -> binäre Darstellung
0,4*2 = **0**,8
0,8*2 = **1**,6
0,6*2 = **1**,2
0,2*2 = **0**,4
0,4*2 = **0**,8
0,8*2 = **1**,6
0,6*2 = **1**,2
0,2*2 = **0**,4
...
...

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
13:2 = 6 Rest 1
6:2 = 3 Rest 0
3:2 = 1 Rest 1
1:2 = 0 Rest 1
=> 13 (dezimal) = **1101** (binär)

c) Exponent
Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{\text{(Anzahl der Exponentenstellen-1)}}$) – 1)
(4-1) + 127 (=„Bias“)
=130 (dezimal) = **10000010** (binär)

d) Nachkommawert -> binäre Darstellung
0,4*2 = **0**,8
0,8*2 = **1**,6
0,6*2 = **1**,2
0,2*2 = **0**,4
0,4*2 = **0**,8
0,8*2 = **1**,6
0,6*2 = **1**,2
0,2*2 = **0**,4
...
... => Periode: **0110 0110 0110 0110 0110 …**

WBS TRAINING

Interne Zahlendarstellung – Gleitkommazahlen

2. Beispiel - Umrechnung der dezimal dargestellten Zahl **-13,4** in Gleitkommadarstellung:

a) -13, 4 ist negativ
   => Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
   13:2 = 6 Rest 1
   6:2 = 3 Rest 0
   3:2 = 1 Rest 1
   1:2 = 0 Rest 1
   => 13 (dezimal) = **1101** (binär)

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{(\text{Anzahl der Exponentenstellen-1})}$) – 1)
   (4-1) + 127 (=„Bias“)
   =130 (dezimal) = **10000010** (binär)

d) Nachkommawert -> binäre Darstellung
   0,4*2 = **0**,8
   0,8*2 = **1**,6
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   0,4*2 = **0**,8
   0,8*2 = **1**,6
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   ...
   ...

Die erste nicht mehr darstellbare Ziffer ist eine 0 => kein Rundungsbedarf

WBS TRAINING

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv

=> Vorzeichenbit = **0**

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
=> Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
=> 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 $^{(\text{Anzahl der Exponentenstellen-1})}$) – 1)
   (21-1) + 127 (=„Bias“)
   =147 (dezimal)

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
   (21-1) + 127 (=„Bias“)
   =147 (dezimal) = **10010011** (binär)

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
   (21-1) + 127 (=„Bias“)
   =147 (dezimal) = **10010011** (binär)

d) Nachkommawert -> binäre Darstellung
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   0,4*2 = **0**,8

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + $(2^{(\text{Anzahl der Exponentenstellen-1})} - 1)$
   (21-1) + 127 (=„Bias“)
   =147 (dezimal) = **10010011** (binär)

d) Nachkommawert -> binäre Darstellung
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   0,4*2 = **0**,8
   } **Abbruchbedingung (kein weiterer Speicherplatz in der Mantisse)**

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
   (21-1) + 127 (=„Bias“)
   =147 (dezimal) = **10010011** (binär)

d) Nachkommawert -> binäre Darstellung
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   0,4*2 = **0**,8
   0,8*2 = **1**,6 **Die erste nicht mehr darstellbare Ziffer ist eine 1 => Rundungsbedarf**

Interne Zahlendarstellung – Gleitkommazahlen

3. Beispiel - Umrechnung der dezimal dargestellten Zahl 2^{20} + 0,6 = **1.048.576,6** in Gleitkommadarstellung:

a) 1.048.576,6 ist positiv
   => Vorzeichenbit = **0**

b) Vorkommawert -> binäre Darstellung
   2^{20} ist in binärer Darstellung eine 1 mit 20 Nullen
   => 2^{20} (dezimal) = **1 0000 0000 0000 0000 0000** (binär)
   => in der Mantisse ist nur noch Platz für **3 weitere Ziffern!**

c) Exponent
   Formel: (Anzahl der binären Stellen vor dem Komma-1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
   (21-1) + 127 (=„Bias“)
   =147 (dezimal) = **10010011** (binär)

d) Nachkommawert -> binäre Darstellung
   0,6*2 = **1**,2
   0,2*2 = **0**,4
   0,4*2 = **0**,8
   0,8*2 = **1**,6

Ergebnis der Rundung

Hinweis:

Falls alle Stellen der Mantisse mit 1 gefüllt sind, so führt eine Rundung zu einem Überlauf der Mantisse.
=> Der Exponent wird um 1 erhöht

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ

=> Vorzeichenbit = **1**

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
...
... => Periode: **0110 0110 0110 0110 0110 …**

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
} => Vorperiode: **001**

0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
...
...
} => Periode: **0110 0110 0110 0110 0110 …**

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

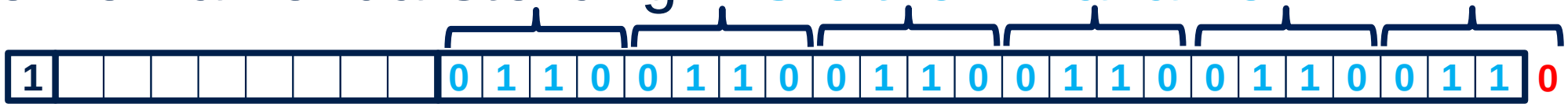
b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4 **Sonderfall: Die Notation innerhalb der Mantisse geschieht nach der ersten 1, die ermittelt wurde.**
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
...
...

Interne Zahlendarstellung – Gleitkommazahlen



kein Rundungsbedarf

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **0**,4
...
...

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
…
…

e) Exponent **(Sonderfall: Andere Formel!)**

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung

0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
…
…

e) Exponent **(Sonderfall: Andere Formel!)**

Formel: - (Anzahl der **Nullen** vor der ersten **Eins** +1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
- (2+1) + 127 (=„Bias“)
=124 (dezimal)

Interne Zahlendarstellung – Gleitkommazahlen

4. Beispiel - Umrechnung der dezimal dargestellten Zahl **-0,175** in Gleitkommadarstellung **(Sonderfall: Zahl ist kleiner als 1)**

a) -0,175 ist negativ
=> Vorzeichenbit = **1**

b) Vorkommawert -> binäre Darstellung
Sonderfall: Da der Vorkommawert 0 ist, entfällt dieser Punkt!

c) Exponent
Sonderfall: Da der Vorkommawert 0 ist, kann der Exponent erst nach der Umwandelung der Nachkommaziffern ermittelt werden!

d) Nachkommawert -> binäre Darstellung
0,175*2 = **0**,35
0,35 *2 = **0**,7
0,7 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
0,2 *2 = **1**,4
0,4 *2 = **0**,8
0,8 *2 = **1**,6
0,6 *2 = **1**,2
…
…

e) Exponent **(Sonderfall: Andere Formel!)**

Formel: - (Anzahl der **Nullen** vor der ersten **Eins** +1) + (2 (Anzahl der Exponentenstellen-1)) – 1)
- (2+1) + 127 (=„Bias“)
=124 (dezimal) = **1111100** (binär)

Interne Zahlendarstellung – Gleitkommazahlen – Normalisierte Zahlen

- Wir hatten zuvor bereits festgestellt, dass die darstellbaren Werte des (8-stelligen) Exponenten zwischen den Werten **-126** und **+127** liegen.
- Durch die **Addition des Bias** (bei 8-stelligen Exponenten beträgt dieser 127) liegen die Werte der Charakteristik also zwischen **1** und **254**.
- Bei allen Zahlen mit einer solchen Charakteristik kann auf die Notation der „führenden 1“ verzichtet werden, was als **normalisierte Darstellung** bezeichnet wird.
- Entsprechend werden solche Zahlen auch als „**Normalisierte Zahlen**“ bezeichnet.
- Die (betragsmäßig) **kleinste** darstellbare normalisierte Zahl lautet demnach: (32 Bit, 8-stellige Charakteristik)

2^{-126} =

- Die **größte** darstellbare normalisierte Zahl ist:

$(2-2^{-23})*2^{127}$ =

Interne Zahlendarstellung – **Denormalisierte Zahlen** – Darstellung der 0

- Zahlen, deren Exponenten-Bits alle 0 sind, werden als „**Denormalisierte Zahlen**“ bezeichnet.
- Zu diesen zählt auch die Null, die auf zweierlei Weise (als **+0** und als **-0**) dargestellt werden kann.
- Diese Unterscheidung kann sinnvoll sein, wenn ein Rechenergebnis zwar positiv (oder negativ) ist, aber so nahe bei 0 liegt, dass eine interne Darstellung ungleich 0 nicht mehr möglich ist.
- Die beiden Darstellungen sehen wie folgt aus: (32 Bit, 8-stellige Charakteristik)

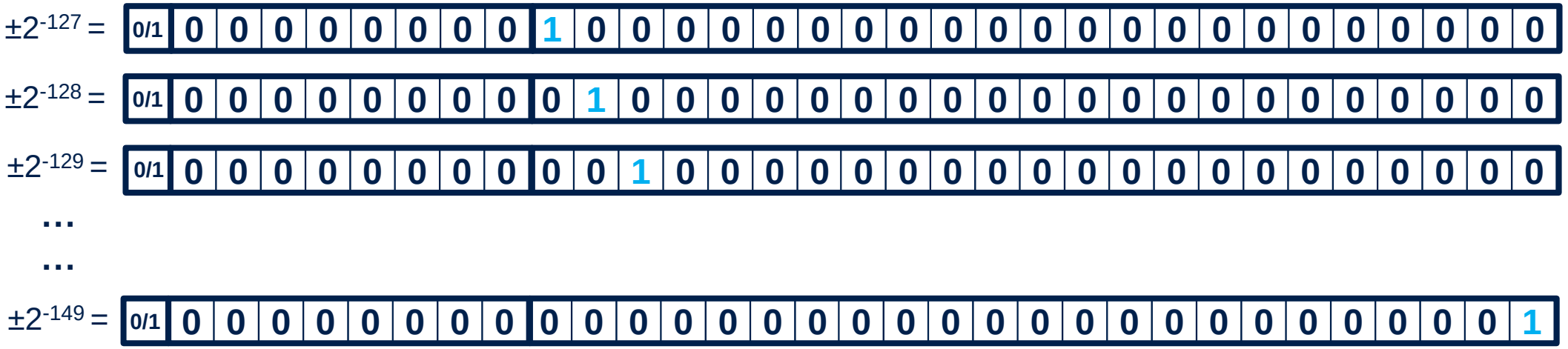
-0 = 

+0 = 

- Bei einem Größen-Vergleich dieser beiden Zahlen werden diese allerdings als identisch betrachtet.

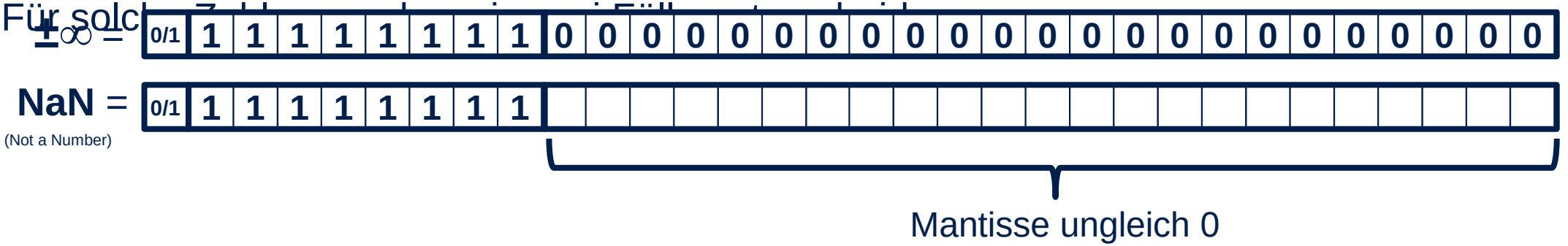
Interne Zahlendarstellung – **Denormalisierte Zahlen** – Zahlen nahe bei 0

- Auch Zahlen, deren Betrag kleiner 2^{-126} ist (aber größer oder gleich 2^{-149}) können (mit geringerer Genauigkeit) als „**Denormalisierte Zahlen**“ dargestellt werden.
- In diesem Fall müssen jedoch alle Ziffern notiert werden (eine führende 1 kann also nicht gestrichen werden).
- Wie schon bei der Darstellung der Null erwähnt, haben alle Exponenten-Bits den Wert 0.
- Die folgenden Beispiele machen das Verfahren deutlich: (32 Bit, 8-stellige Charakteristik)



Interne Zahlendarstellung – Irreguläre Werte

- Von den 256 Werten, die durch eine 8-stellige Charakteristik darstellbar sind, haben wir bisher die 254 normalisierten Fälle (1 bis 254) und den denormalisierten Fall (0) betrachtet.
- Es bleibt noch der Fall, bei dem die Charakteristik den Wert 255 hat (und also nur aus Einsen besteht).
- Für solche Zahlen unterscheiden wir zwei Fälle:



Umgekehrte Umrechnung: Interne Darstellung -> Kommazahl

- Das **Vorzeichen** der Zahl ergibt sich unmittelbar aus dem Vorzeichenbit.
- Falls alle Stellen der Charakteristik 1 sind, handelt es sich um …
  - **Unendlich** (falls alle Stellen der Mantisse 0)
  - **keine Zahl** (NaN) in allen anderen Fällen
- Falls alle Stellen der Charakteristik 0 sind, so werden die Stellen der Mantisse von links nach rechts mit den Faktoren von 2^{-127} bis 2^{-149} multipliziert und deren Ergebnisse aufsummiert.
- In allen anderen Fällen wird …
  - der Wert der Charakteristik ermittelt
  - dieser um den Bias vermindert (das Ergebnis ist der Exponenten-Wert **e**)
  - die Mantisse um eine führende 1 ergänzt
  - die Ziffernfolge der ersten **e** Stellen der ergänzten Mantisse als Vorkommawert betrachtet
  - die Ziffernfolge der restlichen Mantissen-Stellen als Nachkommawert betrachtet

Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)

Wir haben bereits (siehe Folie 40) die Zahl **-13,4** in eine interne Darstellung übersetzt:

Wir wollen diese interne Darstellung nun wieder in eine dezimale Darstellung zurück übersetzen. Dabei werden wir dann freilich feststellen, dass die interne Darstellung den Wert -13,4 nicht exakt trifft.

Hinweis: Der Rundungsfehler wird aber gering sein. Da zudem die Ausgabe von Float-Zahlen auf 6 Nachkommastellen gerundet geschieht, wird diese Abweichung für uns unsichtbar bleiben.

Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)

Das **Vorzeichenbit** hat den Wert **1**
=> Der dargestellte Wert ist **negativ**

Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)

Der dargestellte Wert ist **negativ**

Die **Charakteristik** hat den Wert

$1*2^7+0*2^6+0*2^5+0*2^4+0*2^3+0*2^2+1*2^1+0*2^0 = 130$

$\Rightarrow$ **Exponent** = 130 – 127(Bias) = **3**

$\Rightarrow$ **Normalisierte Zahl** (die Charakteristik ist ungleich 0 und ungleich 255)

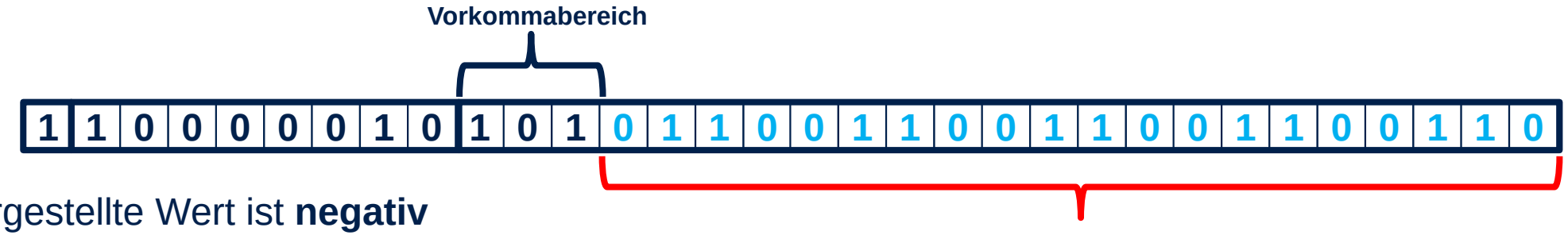
Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)

Der dargestellte Wert ist **negativ**

Exponent = 3

a) Der Exponent gibt an, **wie viele Stellen der Mantisse** (von links aus betrachtet) zum **Vorkommawert** gehören.
b) Da es sich um eine normalisierte Zahl handelt, wurde eine führende 1 „gestrichen“
=> Der Vorkommawert besteht aus den Ziffern **1101**
=> **Vorkommawert** = $\mathbf{1}*2^3+\mathbf{1}*2^2+\mathbf{0}*2^1+\mathbf{1}*2^0 = \mathbf{13}$

Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)



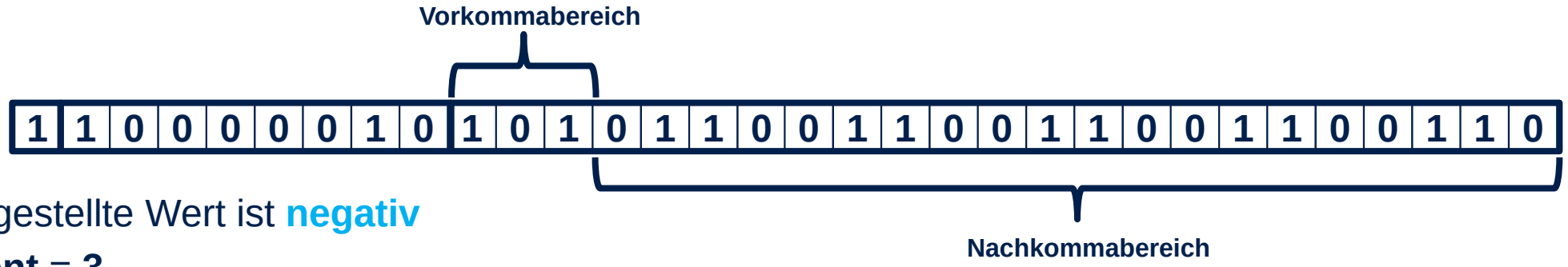
Der dargestellte Wert ist **negativ**

Exponent = **3**

Vorkommawert = **13**

Die Ziffern hinter dem Vorkommabereich bilden den **Nachkommawert**
⇒ **Nachkommawert** = **0***1/2+**1***1/4+**1***1/8+**0***1/16+**0***1/32+**1***1/64+ … = **0.3999996185**
=> **Nachkommawert** (gerundet auf 6 Nachkommastellen) = **0,4**

Umgekehrte Umrechnung: **Interne Darstellung -> Kommazahl** (Beispiel)



Der dargestellte Wert ist **negativ**

Exponent = **3**

Vorkommawert = **13**

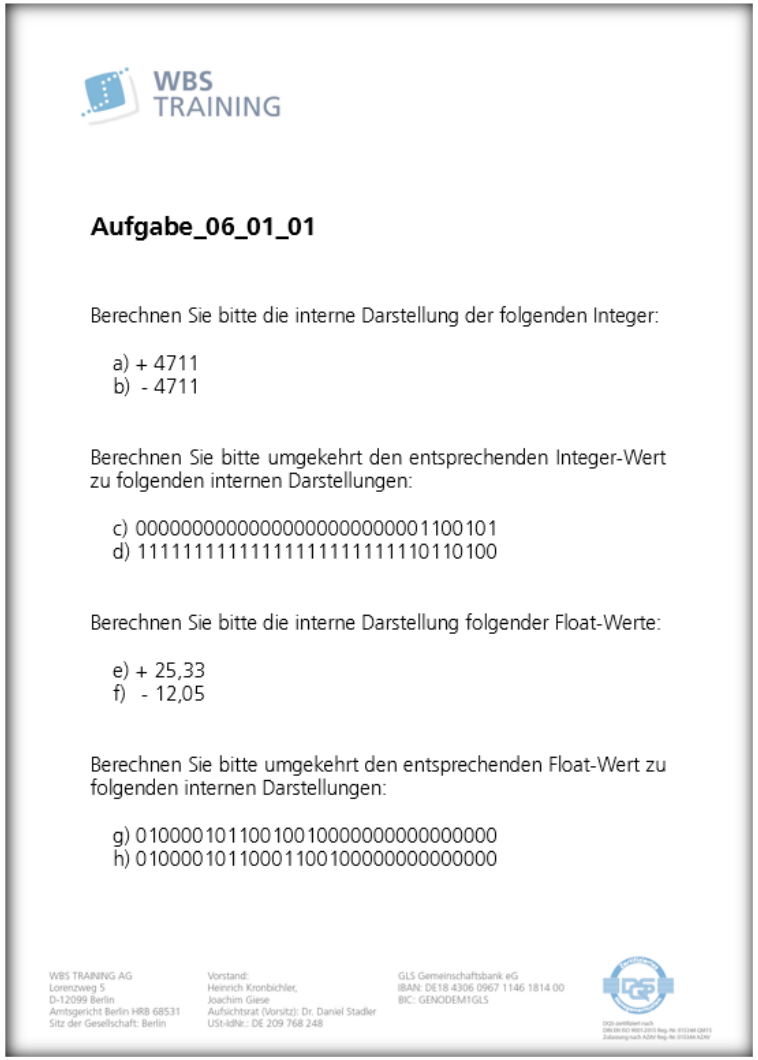
Nachkommawert (gerundet auf 6 Nachkommastellen) = **0,4**

Die auf 6 Nachkommastellen gerundete Ausgabe dieser internen Darstellung lautet also: **-13,400000**

Interne Zahlendarstellung – Double

- Wie bereits erwähnt, ähnelt sich die interne Darstellung von Double- und Float-Zahlen.
- Das zuvor für Float-Zahlen beschriebene Verfahren wird auch bei Double Zahlen angewendet. Lediglich die folgenden quantitativen Unterschiede sind zu berücksichtigen:
  - Double-Zahlen werden mit **8 Byte** (64 Bit) dargestellt
  - Der Exponent (die Charakteristik) besitzt **11 Stellen** => die Mantisse besteht aus **52 Stellen** (ebenfalls nur ein Vorzeichenbit)
  - Der Bias beträgt **1023** => es gilt also erneut: Bias = $2^{(\text{Anzahl der Exponentenstellen-1})} - 1$

Interne Zahlendarstellung – Übung A_06_01_01



WBS TRAINING

Aufgabe_06_01_01

Berechnen Sie bitte die interne Darstellung der folgenden Integer:

a) + 4711
b) - 4711

Berechnen Sie bitte umgekehrt den entsprechenden Integer-Wert zu folgenden internen Darstellungen:

c) 00000000000000000000000001100101
d) 11111111111111111111111110110100

Berechnen Sie bitte die interne Darstellung folgender Float-Werte:

e) + 25,33
f) - 12,05

Berechnen Sie bitte umgekehrt den entsprechenden Float-Wert zu folgenden internen Darstellungen:

g) 01000010110010010000000000000000
h) 01000010110001100100000000000000

WBS TRAINING AG
Lorenzweg 5
D-12099 Berlin
Amtsgericht Berlin HRB 68531
Sitz der Gesellschaft: Berlin

Vorstand:
Heinrich Kronbichler,
Joachim Giese
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler
USt-IdNr.: DE 209 768 248

GLS Gemeinschaftsbank eG
IBAN: DE18 4306 0967 1146 1814 00
BIC: GENODEM1GLS

WBS TRAINING

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!