# Grundlagen der Informatik

Prof. Dr. J. Schmidt
Fakultät für Informatik

GDI – WS 2017/18

Zahlendarstellung – Binäres Rechnen 1

Hochschule Rosenheim
University of Applied Sciences

# Leitfragen 2.3



- Wie wird in Computern gerechnet?
- Welche logischen Grundoperatoren können unterschieden werden?
- Welche charakteristischen Merkmale weist die binäre Addition und Subtraktion auf?

# Logische Operationen (1)



- In Computersystemen werden logische Operationen grundsätzlich bitweise durchgeführt

- Drei logische Grundfunktionen
  - Logisches UND (**AND**)
  - Logisches ODER (**OR**)
  - Logisches NICHT (Inversion oder Negation **NOT**)

- Alle anderen logischen Operationen können durch Verknüpfung der Grundfunktionen abgeleitet werden

# Logische Operationen (2)

- Definition über Wahrheitstafeln (Grundfunktionen)

| | | | | |
|---|---|---|---|---|
| **OR** | $1 \vee 1 = 1$ | $0 \vee 1 = 1$ | $1 \vee 0 = 1$ | $0 \vee 0 = 0$ |
| **AND** | $1 \wedge 1 = 1$ | $0 \wedge 1 = 0$ | $1 \wedge 0 = 0$ | $0 \wedge 0 = 0$ |
| **NOT** | $\neg 1 = 0$ | $\neg 0 = 1$ | | |

# Aufgaben



- $$\begin{array}{rl} & \mathtt{10011} \\ \vee & \mathtt{10101} \end{array} = 10111$$

- $$\begin{array}{rl} & \mathtt{10011} \\ \wedge & \mathtt{10101} \end{array} = 10001$$

- $\neg\ \mathtt{10101} = 01010$

# Logische Operationen (4)



- Weitere wichtige logische Funktion
  Exklusiv Oder (exclusive OR, XOR)

  - a XOR b = $(a \wedge \neg b) \vee (\neg a \wedge b)$
  - Wahrheitstabelle

| a | b | a XOR b |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |

# Binäre Addition (1)



- Rechenregeln für die Addition zweier binärer Zahlen
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 Übertrag 1
  - 1 + 1 + 1(vom Übertrag) = 1 Übertrag 1
- Identisch mit den Regeln des logischen XOR plus Übertrag

# Binäre Addition (2)



- Beispiel: Addieren Sie in binärer Arithmetik die Zahlen 11 und 14

- Rechnung:

| | | | | | |
|---|---|---|---|---|---|
| | | 1 | 0 | 1 | 1 |
| | + | 1 | 1 | 1 | 0 |
| **Übertrag** | 1 | 1 | 1 | | |
| **Ergebnis** | 1 | 1 | 0 | 0 | 1 |

- Berechnen Sie die Summe folgender Zahlen in binärer Arithmetik
  - 45 und 54

$45 = 101101$

$54 = 110110$

$1111\;1$

$99 = 1100011$

$1+2+32+64 = 99$

  - 151.875 und 27.625

$(151.875)_{10} = (10010111,111)_2$

$(27.625)_{10} = (11011,101)_2$

$11111\;1\;1\;11$

$(179,5)_{10} = (10110011,100)_2$

$128+32+16+2+1+0,5 = (179,5)_{10}$

# Binäre Subtraktion (1)



- Rechenregeln für die Subtraktion zweier binärer Zahlen
  - 0 – 0 = 0
  - 1 – 1 = 0
  - 1 – 0 = 1
  - 0 – 1 = 1   Übertrag  –1

1 - 1 - 1 = 1 Übertrag - 1

# Binäre Subtraktion (2)

- Beispiel: Die Aufgabe 13 – 11 soll in binärer Arithmetik gelöst werden

- Rechnung:

| | | | | | |
|---|---|---|---|---|---|
| | | 1 | 1 | 0 | 1 |
| | – | 1 | 0 | 1 | 1 |
| **Übertrag** | | | –1 | | |
| **Ergebnis** | | 0 | 0 | 1 | 0 |

# Binäre Subtraktion (3)



- Wie werden negative Zahlen dargestellt?
  - Üblicherweise durch ihren Betrag mit vorangestelltem Minuszeichen
- Ist diese Darstellung auch rechnerintern denkbar?
  - Ja, aber
    - Gesonderte Vorzeichenrechnung müsste durchgeführt werden
    - Benötigt ein Rechenwerk, das sowohl addieren als auch subtrahieren kann
- Gibt es eine Möglichkeit, um nur mit einem Addierwerk auszukommen?
  - Subtraktion auf Addition zurückführen

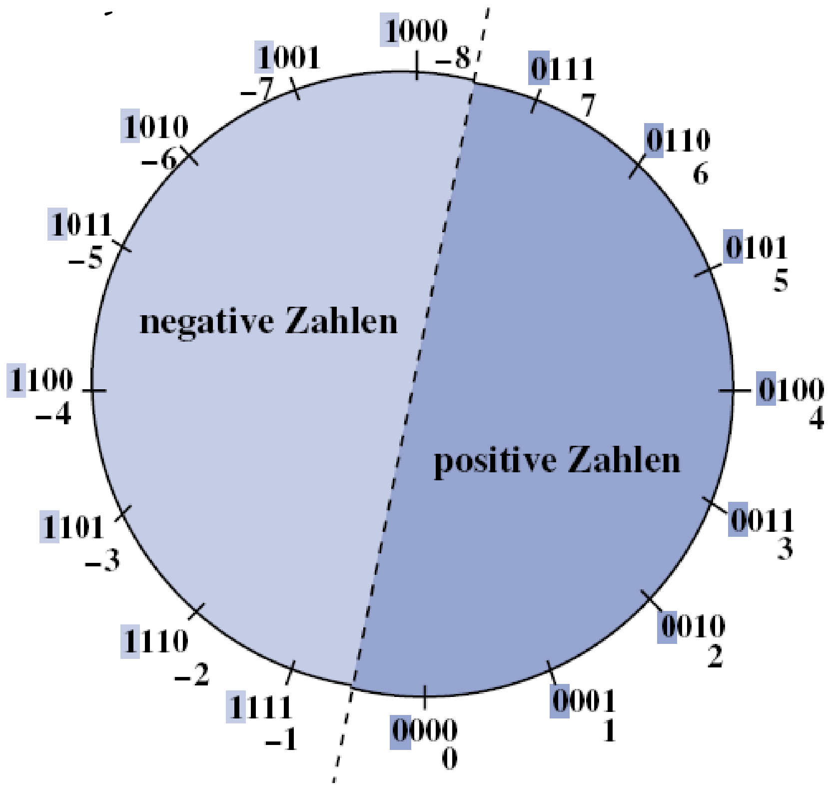
# Verfahren der Komplementbildung



- Zwei Arten der Komplementbildung, wobei B für das Zahlensystem steht
  - B-Komplement und
  - (B-1)-Komplement
- B-Komplement ist technisch leichter realisierbar
  - es wird vorwiegend mit dem B-Komplement (Zweier-Komplement) gearbeitet
- Komplementdarstellung wird nur für ganze Zahlen verwendet
  - Gleitkommadarstellung → später

- Bitkombinationen zu positiven und negativen Zahlen
  - Alle Kombinationen, bei denen das 1. Bit gesetzt ist (**„Vorzeichenbit“**) repräsentieren negative Zahlen
  - Wenn 4 Bit verfügbar

```
0000 = 0        1000 = -8
0001 = 1        1001 = -7
0010 = 2        1010 = -6
0011 = 3        1011 = -5
0100 = 4        1100 = -4
0101 = 5        1101 = -3
0110 = 6        1110 = -2
0111 = 7        1111 = -1
```



# Zweier-Komplement (2)



- Nach welchem Prinzip sind die einzelnen negativen Zahlen den Bitmustern zugeordnet worden?

1. Ist das 1. Bit 1, so handelt es sich um eine negative Zahl.

2. Der Wert einer negativen Zahl wird dann im Zweier-Komplement dargestellt.

Zweier-Komplement zu einem Wert bedeutet, dass zunächst

- jedes einzelne Bit invertiert (umgedreht) wird,
- und dann auf die so entstandene Bitkombination die Zahl 1 aufaddiert wird

# Zweier-Komplement (3)

- Beispiele

**Zweier-Komplement zu 5**

| | |
|---|---|
| Dualdarstellung von 5: | `0101` |
| Negieren von 5: | `1010` |
| +1: | `0001` |
| = –5: | `1011` |

**Zweier-Komplement zu –5**

| | |
|---|---|
| Dualdarstellung von –5: | `1011` |
| Negieren von –5: | `0100` |
| +1: | `0001` |
| = 5: | `0101` |

- Eine Maschine muss nicht subtrahieren können,
  - sondern kann jede Subtraktion a – b durch eine Addition a + (–b) realisieren
  - Beispiel

**2 – 4 = 2 + (–4)**

```
  0010 =  2
+ 1100 = -4
-----------
  1110 = -2
```

**6 – 2 = 6 + (–2)**

```
  0110 =  6
+ 1110 = -2
-----------
 1|0100 =  4
```

Überlauf findet statt.

Kein Problem, da Ergebnis im darstellbaren Bereich liegt

# Komplementdarstellung



- Vorsicht
  - Liefert die Rechenoperation ein Ergebnis, das nicht im darstellbaren Zahlenbereich liegt, dann erhält man beim Überlauf ein falsches Ergebnis
  - Beispiel
    Bei 5 verfügbaren Stellen soll die Subtraktion $(-9)_{10} - (13)_{10}$ im Dualsystem durchgeführt werden

Darstellbarer Zahlenbereich: $-2^4 \dots 2^4 - 1 = -16 \dots +15$

$$
\begin{array}{rrcr}
 & (-9)_{10} & : & (10111)_2 \\
+ & (-13)_{10} & : & (10011)_2 \\
\hline
 & (+10)_{10} & : & 1|(01010)_2
\end{array}
$$

Das vorne überlaufende Bit geht verloren
➔ Falsches Ergebnis!

- Bilden Sie zu folgenden Zahlen das entsprechende B-Komplement
  - $10101_{(2)}$
  - $785_{(10)}$
  - $453_{(16)}$

$$\begin{array}{r} 01010 \\ +\quad 1 \\ \hline 01011 \end{array}$$

$$\begin{array}{r} 214 \\ +\quad 1 \\ \hline 215 \end{array}$$

$$\begin{array}{r} BAC \\ +\quad 1 \\ \hline BAD \end{array}$$

- Subtrahieren Sie folgende Zahlen im B-Komplement mit 8 verfügbaren Stellen und B = 2
  - $(57)_{10} - (122)_{10}$

$(00111001)_2$
$+(11000110)_2$
$(10111111)_2$

01111010 → +122
10000101
+ 1 1
10000110 → −122

10111111 −65
01000000
+ 1
01000001

64 + 1 = 65

# (B-1)-Komplement (1)

- Im Dualsystem: Einer-Komplement
  - Zahlendarstellung ist symmetrisch
  - Es gibt eine **positive** und eine **negative** 0

| | | |
|---|---|---|
| 0000 | +0 | (positive Null) |
| 0001 | 1 | |
| 0010 | 2 | |
| 0011 | 3 | |
| 0100 | 4 | |
| 0101 | 5 | |
| 0110 | 6 | |
| 0111 | 7 | |
| 1000 | −7 | |
| 1001 | −6 | |
| 1010 | −5 | |
| 1011 | −4 | Alle Kombinationen, bei denen das 1. Bit (Vorzeichenbit) |
| 1100 | −3 | gesetzt ist, repräsentieren dabei negative Zahlen. |
| 1101 | −2 | |
| 1110 | −1 | |
| 1111 | −0 | (negative Null) |

- Regeln für die Bildung eines Einer-Komplements
  - Ist das 1. Bit mit 1 besetzt, so handelt es sich um eine negative Zahl (eventuell die negative Null 111…111)
  - Der Wert einer negativen Zahl wird dann im Einer-Komplement dargestellt.

    Einer-Komplement zu einem Wert bedeutet, dass zunächst jedes einzelne Bit invertiert (umgedreht) wird
  - Führt die Addition des Komplements zum Überlauf einer 1, muss zum Ergebnis diese 1 hinzuaddiert werden („Einer-Rücklauf“).

- Beispiel: $(14)_{10} - (7)_{10}$ bei 5 verfügbaren Stellen und B = 2

**Einer-Komplement zu** $(7)_{10} = (00111)_2$:

Invertieren von 7 (−7): 11000

$$
\begin{array}{rl}
\text{Dualdarstellung von } 14: & 01110 \\
+\,-7: & 11000 \\
\hline
: & 1|00110 \\
\text{Aufaddieren von } 1: & 00001 \\
\hline
=7: & 00111
\end{array}
$$

- Beispiel: $(9)_{10} - (13)_{10}$ bei 5 verfügbaren Stellen und B = 2

**Einer-Komplement zu** $(13)_{10} = (01101)_2$:

Invertieren von 13 (−13): 10010

Dualdarstellung von 9: 01001

$+ -13$: 10010

$= -4$: 11011

- Subtrahieren Sie folgende Zahlen im (B–1)-Komplement mit 8 verfügbaren Stellen und B = 2
  - $(43)_{10} - (11)_{10}$
  - $(17)_{10} - (109)_{10}$

00001011 → +11
11110100 → −11

00101011
\+ 11110100
1|00011111
\+ 1
00100000 → 32

01101101 → 109
10010010 → −109

00010001
\+ 10010010
10100011 −92
01011100 → 92

# Historisches Beispiel: Comptometer

Rechnet im B(10)-Komplement
Zahleneingabe im 9-Komplement

Große Zahlen
Beschriftung für Addition

Kleine Zahlen
Beschriftung für Subtraktion
(als 9-Komplement)