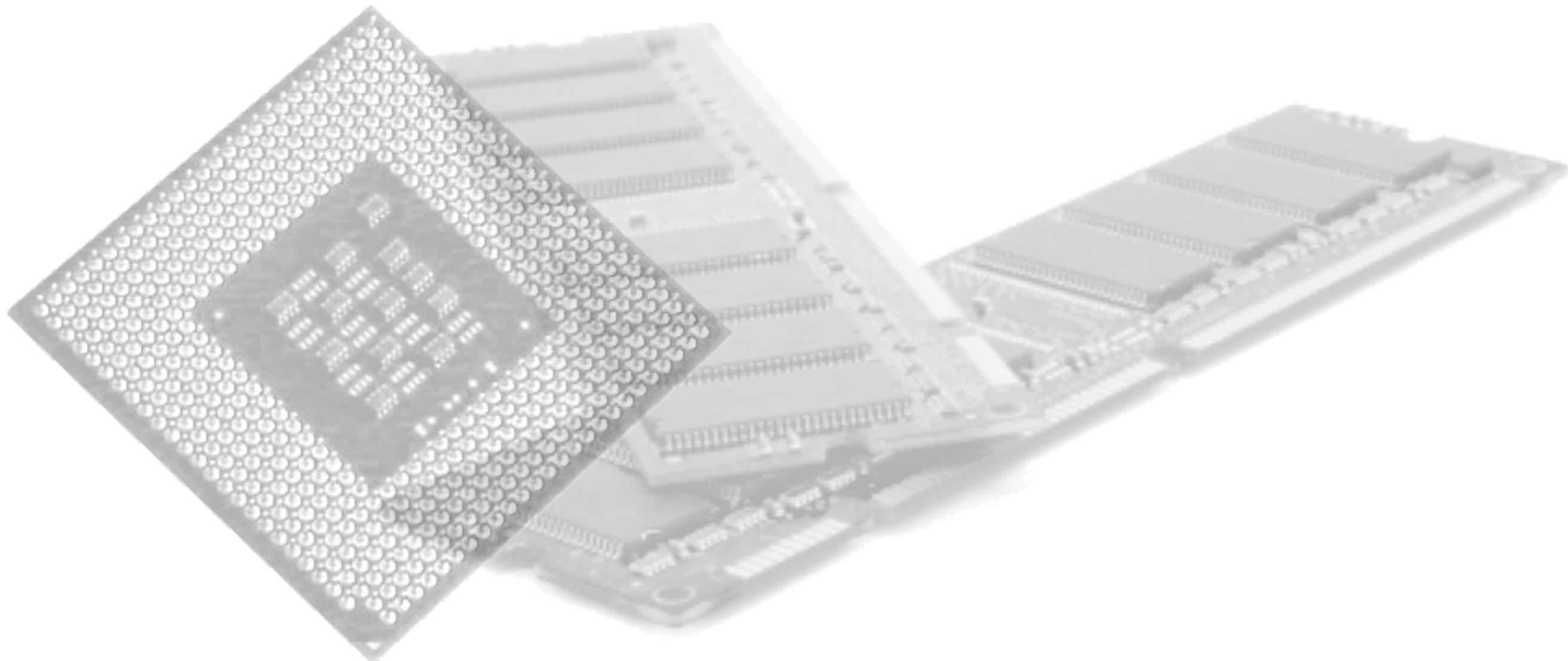
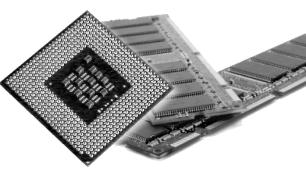


Zahlendarstellung

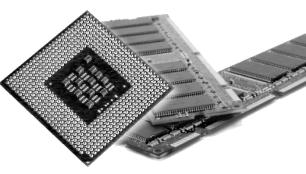




Stellenwertsystem

- Dezimal: 0,1,2,3,4,5,6,7,8,9

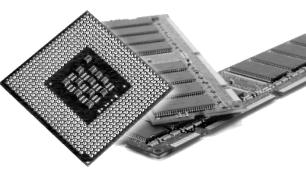
Dezimal		Dezimal
0		
1		
...		
9		
1 0	$= 1 \times 10^1 + 0 \times 10^0$	$= 10 + 0 = 10$
1 1	$= 1 \times 10^1 + 1 \times 10^0$	$= 10 + 1 = 11$
1 2	$= 1 \times 10^1 + 2 \times 10^0$	$= 10 + 2 = 12$
...		
2 8	$= 2 \times 10^1 + 8 \times 10^0$	$= 20 + 8 = 28$
...		
1 3 4	$= 1 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$	$= 100 + 30 + 4 = 134$
...		



Stellenwertsystem

- Oktal: 0,1,2,3,4,5,6,7

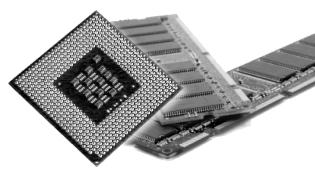
Oktal		Dezimal
0		
1		
...		
7		
1 0	$= 1 \times 8^1 + 0 \times 8^0$	$= 8 + 0 = 8$
1 1	$= 1 \times 8^1 + 1 \times 8^0$	$= 8 + 1 = 9$
1 2	$= 1 \times 8^1 + 2 \times 8^0$	$= 8 + 2 = 10$
...		
2 0	$= 2 \times 8^1 + 0 \times 8^0$	$= 16 + 0 = 16$
...		
1 3 4	$= 1 \times 8^2 + 3 \times 8^1 + 4 \times 8^0$	$= 64 + 24 + 4 = 92$
...		



Stellenwertsystem

- Hexadezimal: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

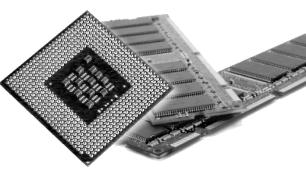
Hexadezimal	Dezimal
0	
...	
9	
A	= 10
B	= 11
...	
F	= 15
1 0	$= 1 \times 16^1 + 0 \times 16^0 = 16 + 0 = 16$
1 1	$= 1 \times 16^1 + 1 \times 16^0 = 16 + 1 = 17$
...	
1 C 4	$= 1 \times 16^2 + 12 \times 16^1 + 4 \times 16^0 = 256 + 192 + 4 = 452$
...	



Stellenwertsystem

- Binär: 0,1

Binär		Dezimal
0		
1		
1 0	$= 1 \times 2^1 + 0 \times 2^0$	$= 2 + 0 = 2$
1 1	$= 1 \times 2^1 + 1 \times 2^0$	$= 2 + 1 = 3$
1 0 0	$= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	$= 4 + 0 + 0 = 4$
1 0 1	$= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	$= 4 + 0 + 1 = 5$
1 1 0	$= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	$= 4 + 2 + 0 = 6$
1 1 1	$= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	$= 4 + 2 + 1 = 7$
1 0 0 0	$= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	$= 8 + 0 + 0 + 0 = 8$
1 0 0 1	$= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	$= 8 + 0 + 0 + 1 = 9$
...		
1 1 0 1 1 0	$= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$	$= 32 + 16 + 0 + 4 + 2 + 0 = 54$
...		



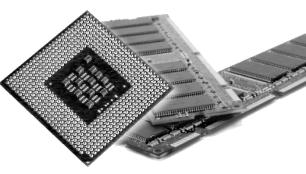
Stellenwertsystem

Definition

Ein Stellenwertsystem zur Basis b ist wie folgt definiert:

$$(a_n \dots a_3 a_2 a_1 a_0, a_{-1} a_{-2} a_{-3}) = \sum_{i=-\infty}^n a_i b^i$$

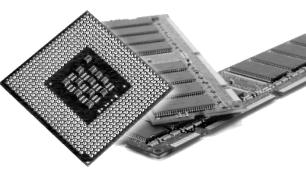
$$b \in \mathbb{N}, b > 1, a_i \in \mathbb{Z}, a_i \in [0, b-1]$$



Stellenwertsysteme

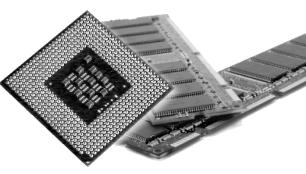
Basis b	Bezeichnung	Ziffernbereich
2	Binär, Dual	0,1
8	Oktal	0,1,...,7
10	Dezimal	0,1,...,9
16	Hexadezimal	0,1,...,9,A,B,C,D,E,F

- Der Wert einer Zahl (=Ziffernfolge) ist abhängig von der Basis. Ohne Angabe der Basis ist der Wert nicht eindeutig definiert.



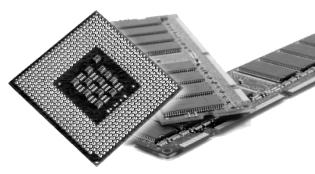
Schreibweise Zahlensysteme

- **5₍₁₆₎**
- 5_{16}
- $[5]_{16}$ Bsp.: Hexadezimal
- 5_{Hex}
- $5_{\text{Hexadezimal}}$
- **5h**
- **0x5**

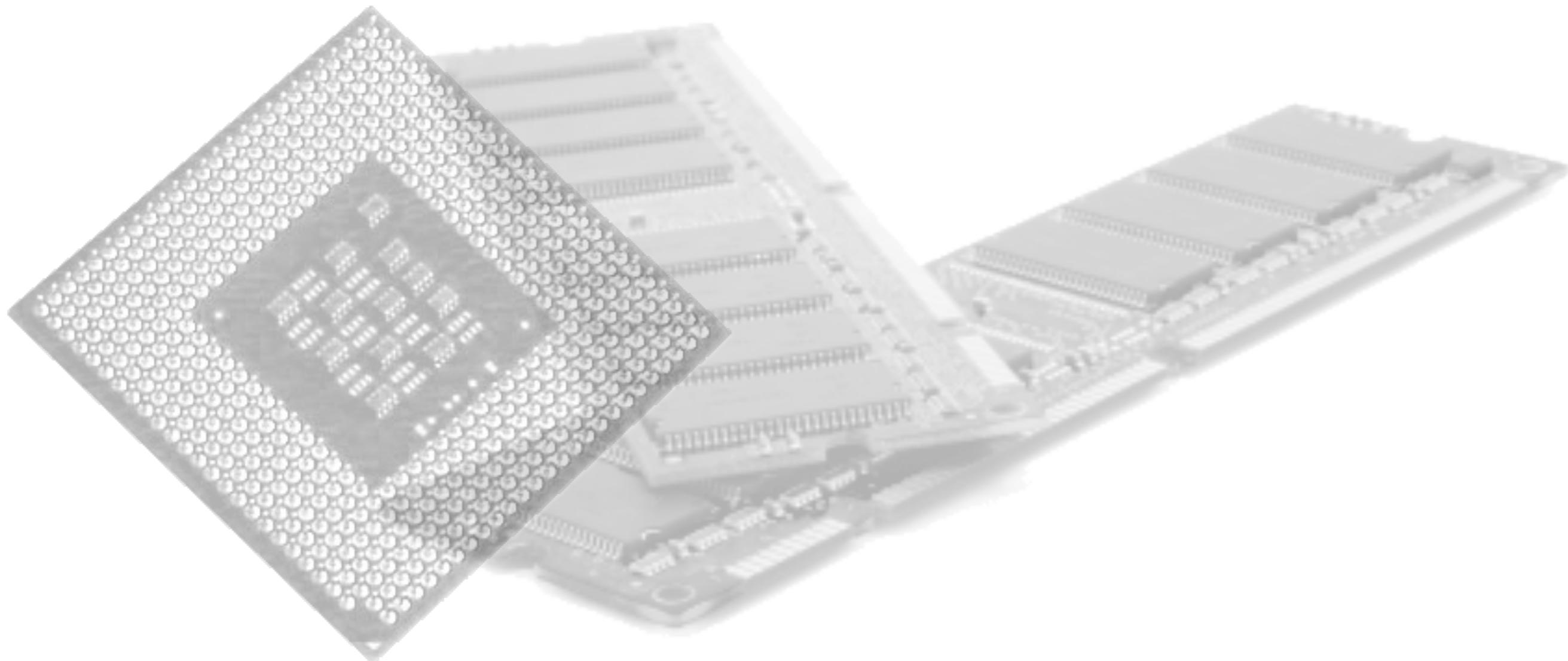


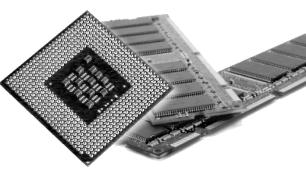
Schreibweise Zahlensysteme

- Für Assembler Wichtig:
 - Hex: **0xABC8**
 - Hex: **4ABC**
 - Hex: **0ABC8h** (ABC8h ist FALSCH)
 - Oct: **0o712**
 - Bin: **0b101101011**



Arithmetik





Addition und Subtraktion

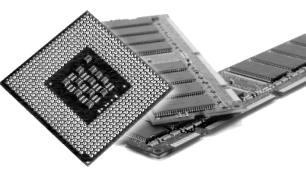
Für die beiden b -adischen Zahlen x und y gemäß

$$x = (x_N \dots x_2 x_1 x_0, x_{-1} x_{-2} \dots x_{-M}) = \sum_{i=-M}^N x_i b^i$$

$$y = (y_N \dots y_2 y_1 y_0, y_{-1} y_{-2} \dots y_{-M}) = \sum_{i=-M}^N y_i b^i$$

wird die Summe $x + y$ bzw. die Differenz $x - y$ nach folgender Regel gebildet:

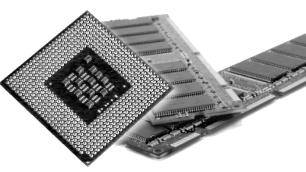
$$x \pm y = \sum_{i=-M}^N x_i b^i \pm \sum_{i=-M}^N y_i b^i = \sum_{i=-M}^N (x_i \pm y_i) b^i$$



Addition im 10-er System

- $123_{(10)} + 258_{(10)} =$

$$\begin{array}{r} & 1 & 2 & 3 \\ + & 2 & 5 & 8 \\ \hline & 1 & & \\ & & & \text{Übertrag bei 10} \\ \hline & 3 & 8 & 1 \end{array}$$



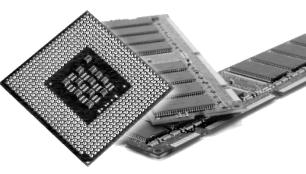
Addition und Subtraktion

Hinweis:

Für den eventuell eintretenden Ziffern- Überlauf ($x_i + y_i > b - 1$) wird ein **Übertrag** eingeführt, der auf die nächst links stehende Ziffer addiert wird.

$$\begin{array}{rcl} x & = & 1260315,2 \\ y & = & 1271423,3 \\ \hline x + y & = & 2531738,5 \end{array}$$

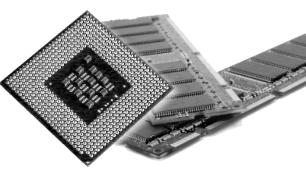
The diagram shows a vertical addition of two floating-point numbers. The numbers are aligned by their decimal points. The digit '6' in the first number and '7' in the second number are highlighted with a yellow box, indicating they are being added together. A small orange '1' is positioned above the yellow box, representing the carry-over from this addition step to the next column.



Addition und Subtraktion

- Binär:
 - $11,625_{10} + 13,25_{10}$

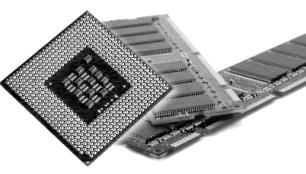
$$\begin{array}{rcl} x & = & 1011,101 \\ y & = & 1101,010 \\ \hline & & 1111 \\ x+y & = & 11000,111 \end{array}$$



Addition im 2-er System

- $101101_{(2)} + 101110_{(2)} =$

$$\begin{array}{r} & 1 & 0 & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 & & & \\ \text{Übertrag bei 2} & & & & & & \end{array}$$

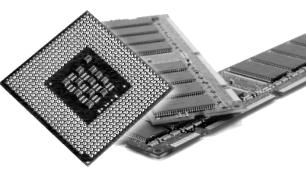


Addition im 5-er System

- $423,2_{(5)} + 14,4_{(5)} =$

$$\begin{array}{r} 4 \quad 2 \quad 3, \quad 2 \\ + \quad \quad 1 \quad 4, \quad 4 \\ \hline 1 \quad \quad 1 \end{array} \qquad \text{Übertrag bei 5}$$

The diagram shows the addition of two binary numbers in base 5. The first number is 423,2₍₅₎ and the second is 14,4₍₅₎. The addition is performed column by column from right to left. The result is 443,1₍₅₎. Blue numbers above the digits indicate carries: a 1 above the third column and another 1 above the fourth column, both labeled "Übertrag bei 5" (carry over 5).

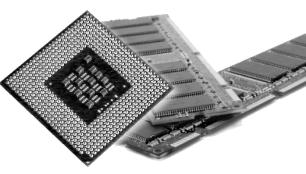


Subtraktion in Dezimal

- $542_{(10)} - 145_{(10)} =$

10 von der nächsten Stelle "holen"

$$\begin{array}{r} & 5 & 4 & 2 \\ - & 1 & 4 & 5 \\ \hline & 1 & 1 & \\ & 3 & 9 & 7 \end{array}$$

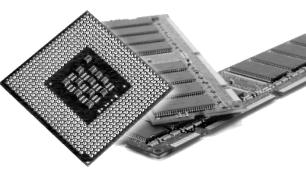


Subtraktion in Binär

- $110_{(2)} - 11_{(2)} =$

2 von der nächsten Stelle "holen"

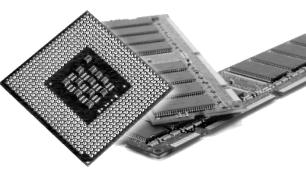
$$\begin{array}{r} & 1 & 1 & 0 \\ - & & 1 & 1 \\ \hline & 1 & 1 & \end{array}$$



Multiplikation

Das Produkt der beiden b -adischen Zahlen x und y berechnet sich wie folgt:

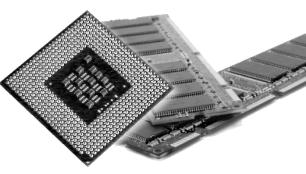
$$\begin{aligned}x * y &= \left(\sum_{i=-M}^N x_i b^i \right) * \left(\sum_{j=-M}^N y_j b^j \right) \\&= \sum_{i,j=-M}^N (x_i b^i * y_j b^j) = \sum_{i,j=-M}^N (x_i * y_j b^{i+j})\end{aligned}$$



Multiplikation

$$\begin{array}{r} & \mathbf{1} & \mathbf{2,} & \mathbf{1} & * & \mathbf{1} & \mathbf{5} \\ \times & 1 & 5 & & & & \\ & 3 & 0 & & & & \\ & & 1 & 5 & & & \\ \hline & 1 & 8 & 1, & 5 & & \end{array}$$

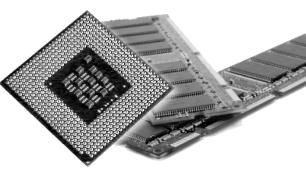
$$12,1_{(10)} * 15_{(10)} = 181,5_{(10)}$$



Multiplikation

$$\begin{array}{r} \mathbf{1} \quad \mathbf{4,} \quad \mathbf{2} \quad * \quad \mathbf{3} \quad \mathbf{7,} \quad \mathbf{4} \\ \begin{array}{r} 3 \quad 7 \quad 4 \\ 1 \quad 4 \quad 9 \quad 6 \\ \hline \end{array} \\ \begin{array}{r} 7 \quad 4 \quad 8 \\ \hline 1 \quad 2 \quad 1 \\ \hline 5 \quad 3 \quad 1, \quad 0 \quad 8 \end{array} \end{array}$$

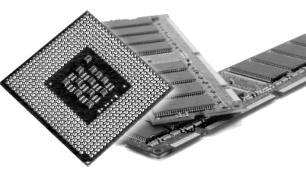
$$14,2_{(10)} * 37,4_{(10)} = 531,08_{(10)}$$



Multiplikation

- Gleiches Prinzip für Binärzahlen
 - Beispiel: **1 0 1**

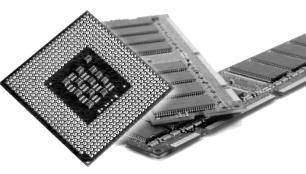
$$\begin{array}{ccccccccc} & 1 & 0 & 1 & 1 & 1 & * & 1 & 0 & 1 \\ 1 & 0 & 1 & & & & 0 & & & \\ & 1 & 0 & 1 & & & & & & \\ & & 1 & 0 & 1 & & & & & \\ & & & 1 & 0 & 1 & & & & \\ \hline & 1 & & & & & & & & \\ 1 & 1 & 0 & 1 & 1 & 1 & & & & \end{array}$$



Multiplikation

$$\begin{array}{r} & 1 & 1 & 1 & * & 1 & 1 \\ 1 & & 1 & & & & \\ & 1 & & 1 & & & \\ & & 1 & 1 & & & \\ & & & & 1 & 1 & \\ \hline 1 & 1 & 1 & & & & \\ 1 & 0 & 1 & 0 & 1 & & \end{array}$$

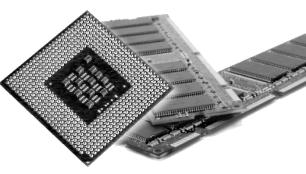
$$111_{(2)} * 11_{(2)} = 10101_{(2)}$$



Multiplizieren in Hex

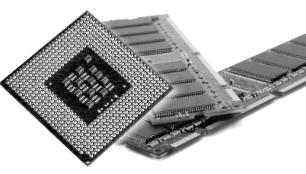
- $19_{(16)} * A1_{(16)} =$

$$\begin{array}{r} 1 \quad 9 \\ * \quad \quad \quad A \quad 1 \\ F \quad A \\ \hline 1 \quad 9 \\ \hline F \quad B \quad 9 \end{array}$$



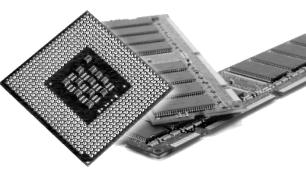
Division

- Dezimalsystem
 - Betrachtung erste Ziffer des Dividenden
 - wenn Divisor n-mal Bestandteil ($n = \{0, \dots, 9\}$)
 - n als nächste Ziffer anschreiben
 - n-faches des Divisors von Dividenden subtrahieren
 - für weitere Dividenden-Stelle bis 0 wiederholen



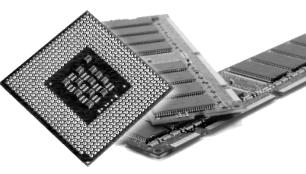
Division

$$\begin{array}{r} 660 : 12 = 55 \\ - 60 \\ \hline 60 \\ - 60 \\ \hline 0 \end{array}$$



Division

$$\begin{array}{r} 9 & 2 & 8 & 8 : & 3 & 6 = & \boxed{2 & 5 & 8} \\ \hline - & 7 & 2 \\ \hline 2 & 0 & 8 \\ - & 1 & 8 & 0 \\ \hline 2 & 8 & 8 \\ - & 2 & 8 & 8 \\ \hline 0 \end{array}$$

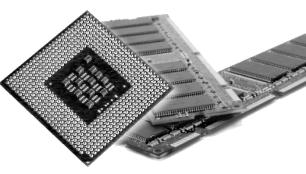


Division

- Gleiches Prinzip für Binärzahlen
- Divisor immer 0 oder 1 mal Bestandteil des Dividenden

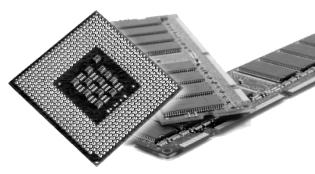
- Beispiel:

$$\begin{array}{r} 1 \quad 0 \quad 1 \quad 0 \quad 1 \\ \hline & 1 \quad 0 \quad 1 \\ - & \hline & 1 \quad 0 \quad 0 \\ - & \hline & 1 \quad 1 \\ - & \hline & 1 \quad 1 \\ - & \hline & 0 \end{array} = \boxed{1 \quad 1 \quad 1}$$

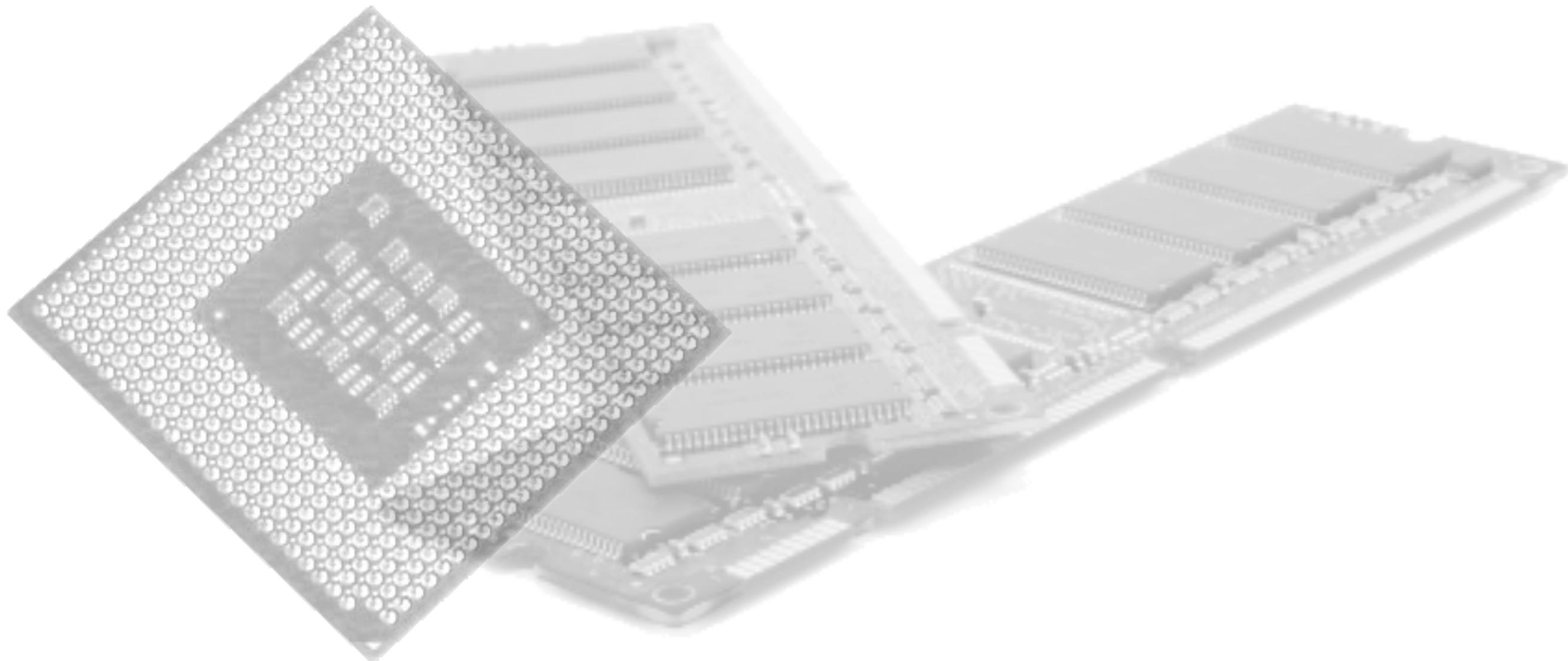


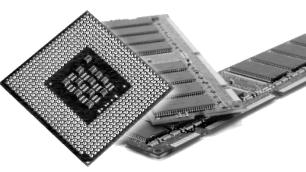
Division

$$\begin{array}{r} 100001 : 11 = 1011 \\ - \quad 11 \\ \hline 100 \\ - \quad 11 \\ \hline 11 \\ - \quad 11 \\ \hline 0 \end{array}$$



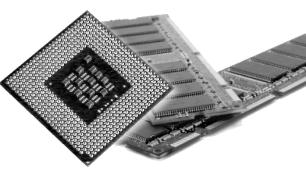
Konvertierung





Konvertierung

- Umformung von einem Stellenwertsystem in ein anderes
- z.B. $1111_{(2)} = 15_{(10)}$ (binär nach dezimal)
- Horner-Schema
- Verfahren der fortgesetzten Division mit Rest
- Spezialfälle bei der Konvertierung
- Festkommazahlen
- Gleitkommazahlen

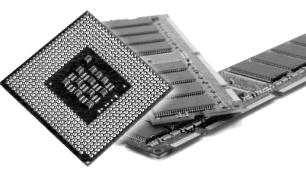


Umrechnung nach Dezimal

- $101,11_{(2)} = 1 * \mathbf{2^2} + 0 * 2^1 + 1 * \mathbf{2^0} + 1 * \mathbf{2^{-1}} + 1 * \mathbf{2^{-2}}$
 $= \mathbf{4} + 0 + \mathbf{1} + \mathbf{0,5} + \mathbf{0,25} = 5,75$

$$2^{-2} = \frac{1}{2^2}$$

- $5,5_{(16)} = 5 * 16^0 + 5 * \mathbf{16^{-1}} =$
 $5 + 5 * \mathbf{0,0625} = 5,3125$

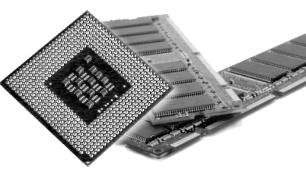


Horner-Schema

Algorithmus

Eine c -adische Zahl u wird in eine b -adische Zahl v konvertiert, indem die folgende Formel angewandt wird:

$$(u)_c = \sum_{i=0}^n u_i b^i = ((\dots((u_n b + u_{n-1}) b + u_1) b + u_0) = (v)_b$$



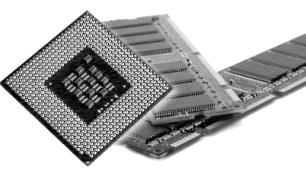
Horner-Schema

- $2173_{(8)}$ in Dezimal:

The diagram illustrates the expression tree for the formula $((2 * 8 + 1) * 8) + 7 * 8 + 3 =$. The root node is an equals sign (=). Four arrows point from the numbers 2, 1, 7, and 3 to their respective positions in the formula. The number 3 is followed by a red parenthesis containing the digit 8, indicating that 3 is multiplied by 8.

```
graph TD; 2 --> "((2 * 8 + 1) * 8)"; 1 --> "((2 * 8 + 1) * 8)"; 7 --> "+7"; 3["3 (8)"] --> "*8";
```

1 1 4 7 (10)



Horner-Schema

Kovertierung von binär nach dezimal:

$$(11001)_2 =$$

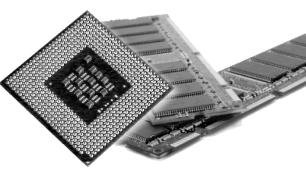
$$1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 =$$

$$\left(\left((1 * 2 + 1) * 2 + 0 \right) * 2 + 0 \right) * 2 + 1 =$$

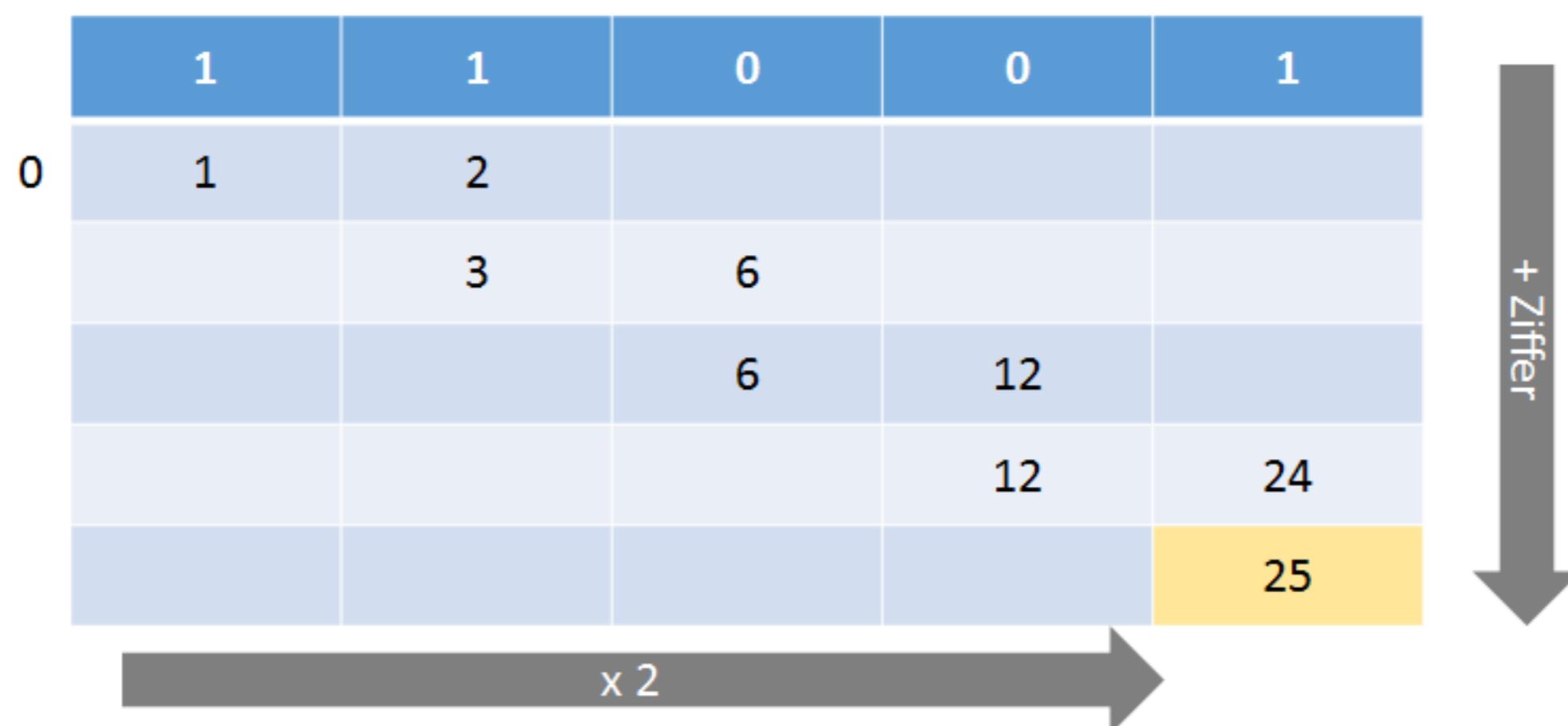
$$\left(\left((2 + 1) * 2 + 0 \right) * 2 + 0 \right) * 2 + 1 =$$

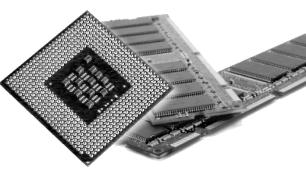
$$(3 * 2 * 1 + 0) * 2 + 1 =$$

$$12 * 2 + 1 = 25 = (25)_{10}$$



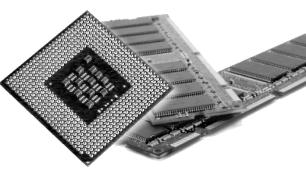
Horner-Schema





Horner-Schema

- Ausnutzung des Stellenwertsystems wiederholte Verschiebung von Ziffern (div/mod)
- Umwandlung dual → dezimal:
 - Dualziffern von links nach rechts durchlaufen
 - Ziffer (1 oder 0) addieren, Resultat mit 2 multiplizieren (außer nach letzter Stelle)
- Umwandlung dezimal → dual:
 - Dezimalzahl wiederholt durch 2 teilen (ganzzahlig)
 - jeweils Rest (0 oder 1) notieren
 - notierte Ziffern ergeben Dualzahl (umgekehrt)



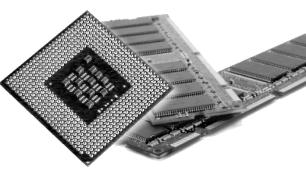
Verfahren der fortgesetzten Division mit Rest

- Das Verfahren der fortgesetzten Division mit Rest stellt eine Umkehrung des Horner-Schemas dar.

$$z = q * d + r$$

$$z = (z \text{ div } d) * d + (z \text{ mod } d)$$

$$\begin{aligned} z &= (a_n a_{n-1} \dots a_1 a_0)_2 = a_n * 2^n + a_{n-1} * 2^{n-1} + \dots + a_1 * 2^1 + a_0 * 2^0 \\ &= (a_n * 2^{n-1} + a_{n-1} * 2^{n-2} + \dots + a_1) * 2^1 + a_0 * 2^0 \\ &= (a_n a_{n-1} \dots a_1)_2 * 2^1 + a_0 \end{aligned}$$



Beispiel

$z = 29$

$z \text{ div } 2$

$z \text{ mod } 2$

29

14

1

14

7

0

7

3

1

3

1

1

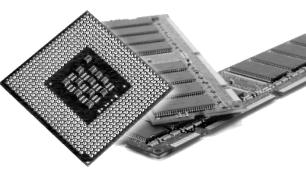
1

0

1

Rest von unten nach oben gelesen
ergibt das Ergebnis

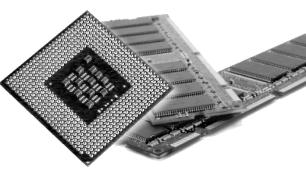
$\Rightarrow (1\ 1\ 1\ 0\ 1)_2$



von Dezimal zu Binär

- $1,8125_{(10)}$ zu Binär:

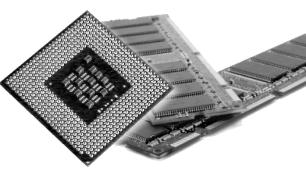
$$\begin{array}{rcl} 0,8125 & *2 & = \\ 0,625 & *2 & = \\ 0,25 & *2 & = \\ 0,5 & *2 & = \\ & & \downarrow \\ & & = 1,1101_{(2)} \end{array}$$



Beispiel 2

$z = 105$	$z / 2$	$z \text{ mod } 2 \text{ (Rest)}$
105	52	1
52	26	0
26	13	0
13	6	1
6	3	0
3	1	1
1	0	1

=> $1101001_{(2)}$



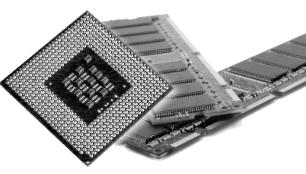
Beispiel 2

- $0,2525_{(10)}$ zu Binär:

0,2525	$\cdot 2$	=	0,505
0,505	$\cdot 2$	=	1,01
0,01	$\cdot 2$	=	0,02
0,02	$\cdot 2$	=	0,04
0,04	$\cdot 2$	=	0,08
0,08	$\cdot 2$	=	0,16
0,16	$\cdot 2$	=	0,32
0,32	$\cdot 2$	=	0,64
0,64	$\cdot 2$	=	1,28

...

$$= 0, \mathbf{01000001\dots}_{(2)}$$

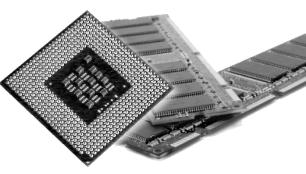


Dezimal zu 5er- System

$z = 167$	$z / 5$	$z \text{ mod } 5 \text{ (Rest)}$
167	33	2
33	6	3
6	1	1
1	0	1

↑

=> $1132_{(5)}$

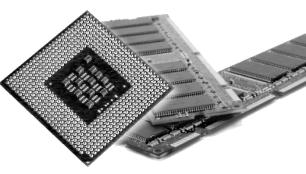


Dezimal zu 5er- System

- $0,568_{(10)}$ zu 5er:

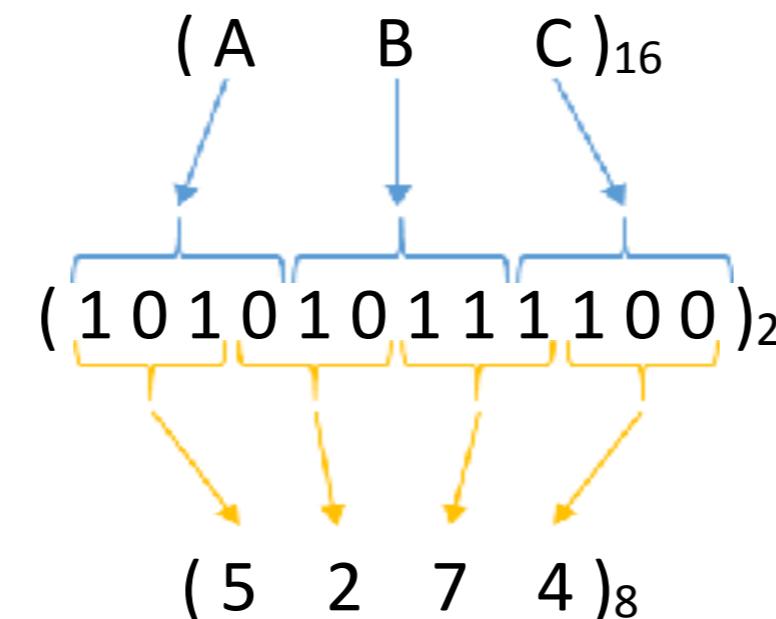
$$\begin{array}{r} \mathbf{0,568} \\ \times 5 \\ \hline \mathbf{2,84} \end{array}$$
$$\begin{array}{r} \mathbf{0,84} \\ \times 5 \\ \hline \mathbf{4,2} \end{array}$$
$$\begin{array}{r} \mathbf{0,2} \\ \times 5 \\ \hline \mathbf{1,0} \end{array}$$

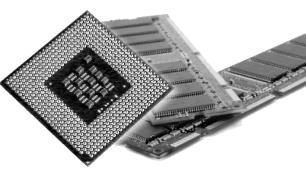
$= 0, \mathbf{241}_{(5)}$



Spezialfälle

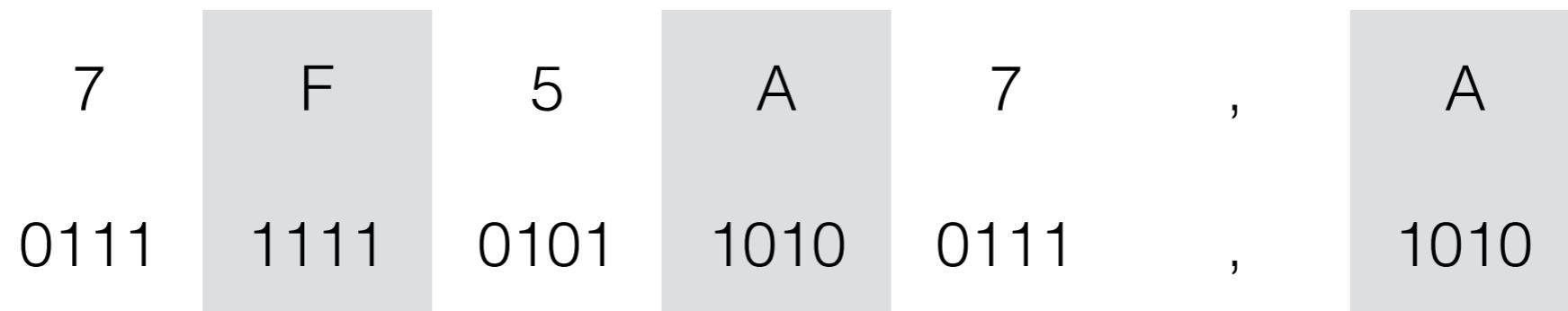
- Konvertierungen zwischen **hexadezimal – binär** und **oktal** sind sehr einfach durchzuführen:



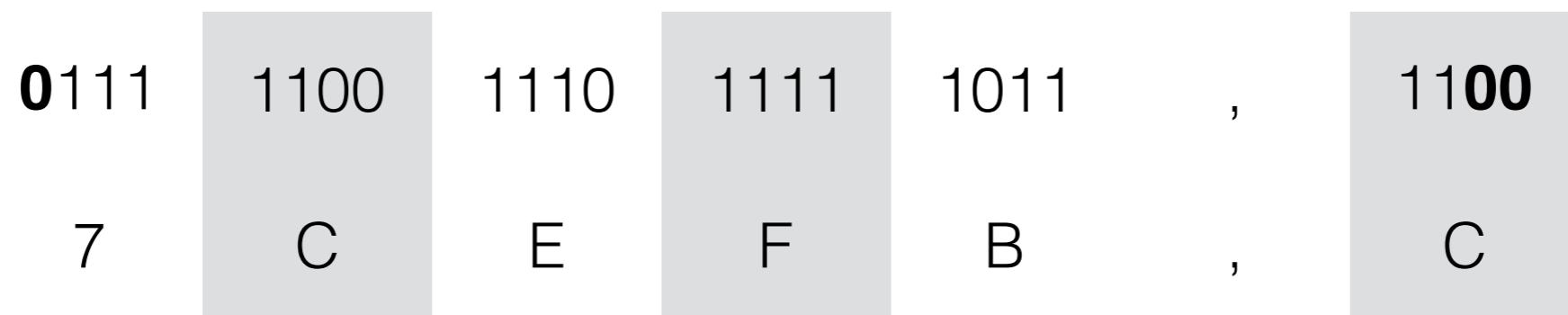


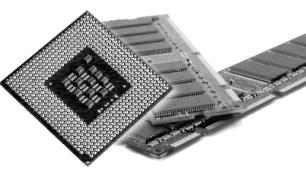
Spezialfälle Beispiele

7F5A7,A Hex zu Bin: 1111111010110100111,101



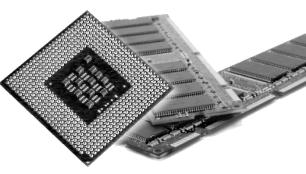
1111100111011111011,11 Bin zu Hex: 7CEFB,C





Darstellung von Zahlen

- Computer kennt nur 0 und 1
- Wo fängt die Zahl an?
- Wo ist sie zu Ende?
- Feste Länge pro Zahl definieren:
- z.B. Integer: 32 Bit, Long: 64 Bit (Beispielhaft, je nach Architektur und Sprache verschieden)



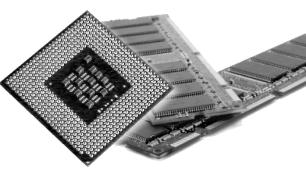
Darstellung von Zahlen

468.749.823₍₁₀₎

0001 1011 1111 0000 1000 1101 1111 1111

32
Bit

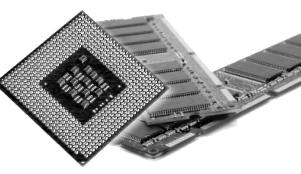
Kürzere Schreibweise in Hex: 0x1BF08DFF



Ganze Zahlen

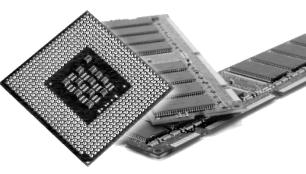
- Die Nutzung des 2er-Komplement ermöglicht die Darstellung von negativen und positiven ganzen Zahlen
- Vorzeichenbit repräsentiert Zahlenbereich
- Subtraktion durch Umwandlung in Addition mit negativen Zahlen

$$z = -a_n * 2^n + a_{n-1} * 2^{n-1} + \dots + a_1 * 2^1 + a_0 * 2^0$$



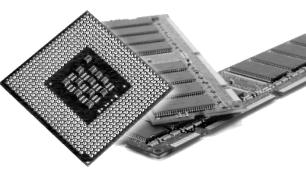
2er Komplement

- Aus der positiven Zahl:
- $16.608_{(10)} \Rightarrow 0100000011100000_{(2)}$
- die negative Zahl:
- $-16.608_{(10)} \Rightarrow ?_{(2)}$



2er Komplement

- Von rechts bis zur ersten 1 abschreiben (incl. erste 1) und dann die restlichen Stellen umdrehen (aus 1 wird 0 und aus 0 wird 1):
- 0100000011 100000 <= von rechts beginnen
- Umdrehen, abschreiben
- 1011111100 100000

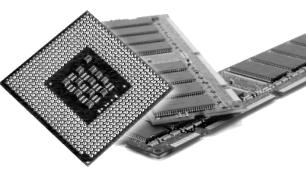


2er Komplement

- Interpretation:
 - das höchstwertige Bit hat eine negative Wertigkeit

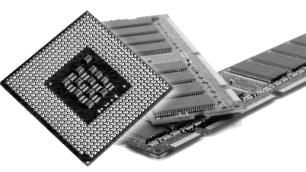
Wertigkeit	-128	64	32	16	8	4	2	1	Dezimal
Bitfolge	0	0	0	1	1	0	1	0	= 26
Bitfolge	1	1	1	0	0	1	1	0	= -26

- $00011010_{(2)} = 16 + 8 + 2 = 26$
- $11100110_{(2)} = -128 + 64 + 32 + 4 + 2 = -26$



2er Komplement - Subtraktion

- $7_{10} - 9_{10}$: 0000 0111₂ - 0000 1001₂
 - 2er Komplement von 0000 1001: **1111 0111**
 - Addition: 0000 0111 **+ 1111 0111 = 1111 1110**
 - negatives Ergebnis: 2er Komplement von 1111 1110: **0000 0010** (-2_{10})
- $8_{10} - 5_{10}$: 0000 1000₂ - 0000 0101₂
 - 2er Komplement von 0000 0101: **1111 1011**
 - Addition: 0000 1000 **+ 1111 1011 = 1 0000 0011**
 - positives Ergebnis: Stellen beachten: **0000 0011** (3_{10})



Festkommazahlen

- Darstellung durch Kommmazahlen mit fester Anzahl n Stellen vor dem Komma und m Stellen nach dem Komma (Festkommadarstellung, fixed point)

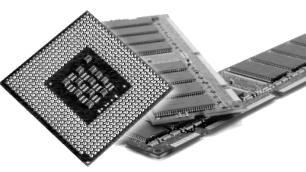
$$z = (a_n a_{n-1} \dots a_1 a_0 , b_1 b_2 \dots b_{m-1} b_m)_2$$

$$= (a_n * 2^n + a_{n-1} * 2^{n-1} + \dots + a_0 , b_1 * 2^{-1} + \dots + b_m * 2^{-m})_2$$

- Behandelte Rechenverfahren können direkt übernommen werden, evtl. Anpassungen
- Stellenkorrektur

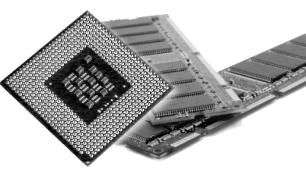
$$0000\ 1010 * 0000\ 1100 = 0111\ 1000$$

$$0000\ 1,010 * 0000\ 1,100 = 0000\ 1,111$$



Festkommazahlen - Hinweise

- Genauigkeitsverlust bei kleinen Beträgen
 - $00123,456 : 100 = 00001,234$
 - d.h. zwei signifikante Ziffern gehen verloren
- Überlauf bei hohen Beträgen
 - $00123,456 \cdot 1000 = (1)23456,000$
 - zu viele Stellen für vorgesehene Darstellung
- Daher **Gleitkommadarstellung (floating point)**
 - Idee: signifikanten Ziffern und ihrer Position getrennt dargestellt (Exponentialschreibweise)



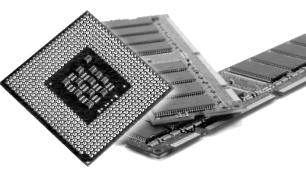
Gleitkommazahlen

- Nutzung von Exponent und Mantisse
 - Exponent zeigt Nachkommastelle bzw. Position der Mantisse (in Bezug zu einer Basis)
 - Mantisse zeigt darstellbare Stellen

$$123,456 = 0,123456 \times 10^3$$

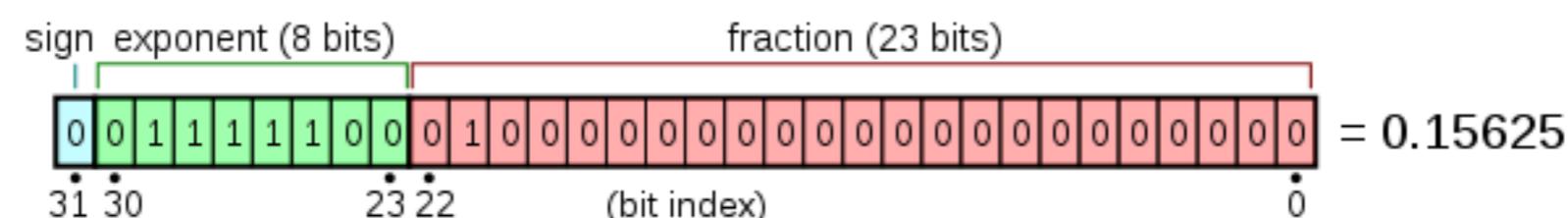
Mantisse (m) Exponent (e)
Basis (b)

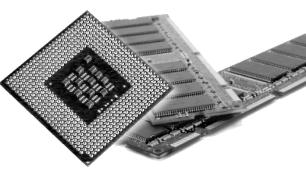
- Darstellung nicht eindeutig
 - Deshalb „Normalisierung“ erforderlich



IEEE floating point (IEEE 754)

Name	Common name	Base	Digits	Decimal digits	Exponent bits	Decimal E max	Exponent bias ^[6]	E min	E max
binary16	Half precision	2	11	3.31	5	4.51	$2^4 - 1 = 15$	-14	+15
binary32	Single precision	2	24	7.22	8	38.23	$2^7 - 1 = 127$	-126	+127
binary64	Double precision	2	53	15.95	11	307.95	$2^{10} - 1 = 1023$	-1022	+1023
binary128	Quadruple precision	2	113	34.02	15	4931.77	$2^{14} - 1 = 16383$	-16382	+16383
binary256	Octuple precision	2	237	71.34	19	78913.2	$2^{18} - 1 = 262143$	-262142	+262143
decimal32		10	7	7	7.58	96	101	-95	+96
decimal64		10	16	16	9.58	384	398	-383	+384
decimal128		10	34	34	13.58	6144	6176	-6143	+6144



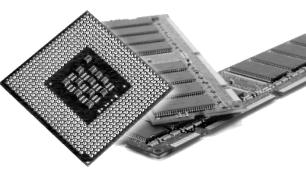


Gleitkommazahlen

- Hauptstandard: IEEE-Format als Normalisierung
- Single/Double Precision

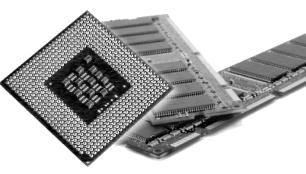
Größe	32 Bit	64 Bit
Vorzeichen	1	1
Exponent	8	11
Mantisse	23	52

- Vorzeichen: 0 positiv, 1 negativ
- Exponent: Wertebereich -126 bis +127
- Mantisse: normalisiert auf 1,...
 - 0 nicht normalisierbar -> minimale Mantisse/Exponent
 - Sonderfälle: NaN und „Unendlich“



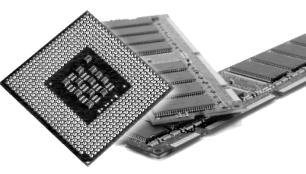
Gleitkommazahlen

- $12,25_{(10)}$ in Binäre Gleitkommazahl umwandeln:
 - Vorzeichen: 1 Bit (0: positiv, 1: negativ)
 - Länge des Exponenten: 5 Bit
 - Länge der Mantisse: 6 Bit
 - Normalisierung auf 1,...



Gleitkommazahlen

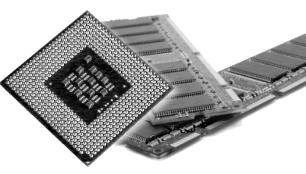
- 12,25₍₁₀₎ in Binäre Gleitkommazahl umwandeln:
 - Vorzeichen:
 - positiv -> 0
 - Umwandeln:
 - $12_{(10)} = 1100_{(2)}$
 - $0,25_{(10)} = 0,01_{(2)}$
 - Normalisieren:
 - $1100,01 = 1,10001 * 2^3$



Gleitkommazahlen

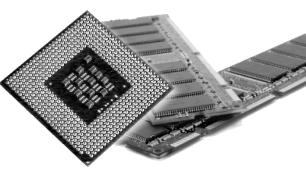
- 12,25₍₁₀₎ in Binäre Gleitkommazahl umwandeln:
 - Normalisieren:
 - $1100,01 = 1,10001 * 2^3$
 - Mantisse:
 - 10001
 - Exponent:
 - $3_{(10)} = 11_{(2)}$

	VZ	Exponent						Mantisse					
12,25	0	0	0	0	1	1	1	0	0	0	1	0	



Gleitkommazahlen IEEE 754

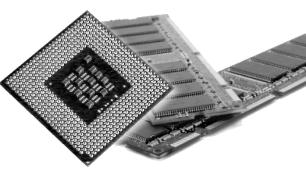
- $x = s * m * b^e$
- Exponent:
 - fester Biaswert B wird addiert: $E = e + B$
 - $B = 2^{r-1} - 1$
 - r = Anzahl der Stellen des Exponenten



Gleitkommazahlen

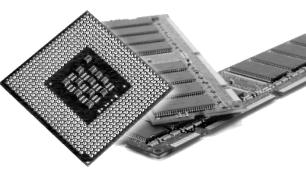
- 12,25₍₁₀₎ in Binäre Gleitkommazahl umwandeln:
 - Normalisieren:
 - $1100,01 = 1,10001 \cdot 2^3$
 - Mantisse:
 - 10001
 - Exponent:
 - $3 + 2^{r-1} - 1 = 3 + 2^{5-1} - 1 = 3 + 16 - 1 = 18_{(10)} = 10010_{(2)}$

	VZ	Exponent					Mantisse					
12,25	0	1	0	0	0	1	0	1	0	0	1	0



Gleitkommazahlen - Konvertierung

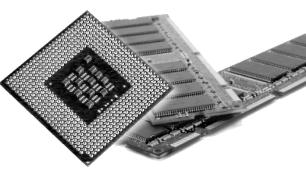
- Umrechnung zwischen dezimal – binär durch Horner-Schema (erweiterte Form)
 - Vorkommazahl umrechnen
 - Nachkommazahl umrechnen
 - Normalisierung
 - Exponent berechnen
 - Vorzeichen bestimmen
 - Ergebnis darstellen



Gleitkommazahlen - Arithmetik

- Addition/Subtraktion
 - Normalisierung beider Zahlen: gleicher Exponent (höhere)
 - Mantissenwerte addieren / subtrahieren
 - Ggf. Normalisierung des Ergebnisses

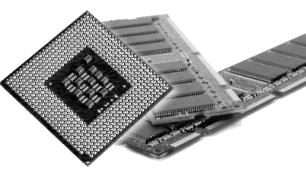
$$\begin{aligned} & 0,1234 \times 10^4 - 0,9876 \times 10^3 \\ = & 0,1234 \times 10^4 - 0,09876 \times 10^4 \\ = & 0,02464 \times 10^4 \\ = & 0,2464 \times 10^3 \end{aligned}$$



Gleitkommazahlen - Arithmetik

- Multiplikation/Division
 - Mantissenwerte multiplizieren / dividieren
 - Exponentenwerte addieren / subtrahieren
 - Ggf. Normalisierung

$$\begin{aligned} & 0,1234 \times 10^4 \quad \times 0,9876 \times 10^3 \\ = & 0,1234 \times 0,9876 \times 10^4 \times 10^3 \\ = & 0,12187 \times 10^7 \end{aligned}$$

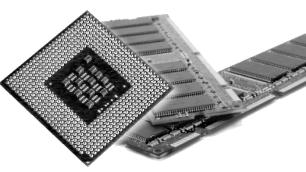


Zusammenfassung

- Horner-Schema
 - $(((a_3 * \text{Basis} + a_2) * \text{Basis}) + a_1) * \text{Basis} + a_0$
- Spezialfälle
 - Hex <-> Bin <-> Oct
- Gleitkommazahlen

$$123,456 = 0,123456 \times 10^3$$

Mantisse (m) Exponent (e)
 Basis (b)



Konvertierungen

- Hexadezimal- / Binärtabelle

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BIN	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111