Digitaltechnik

Prof. Dr. Eckhard Kruse

DHBW Mannheim

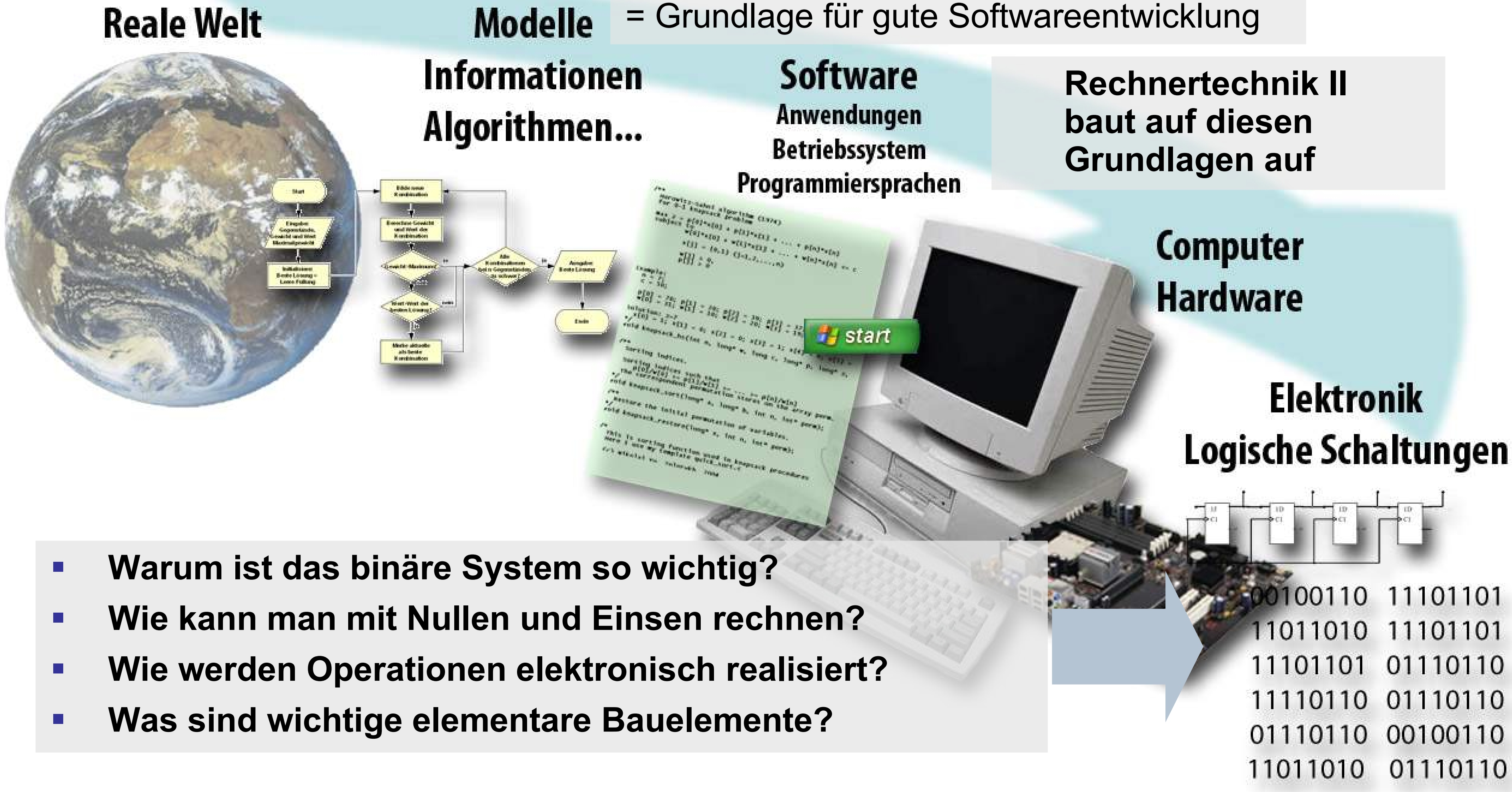
Organisatorisches – TINF21AI

- Termine:
  - s. Kurskalender
  - Pausen in der VL nach Bedarf
- Vorlesung + Übungen
  - Übungen mit Digitalsimulator (→ eigene Laptops)
  - 3 Labortermine
- Fragen: Am besten direkt in der VL
  - E-Mail: eckhard.kruse@dhbw-mannheim.de
  - Raum 344 B, Tel. (0621) 4105 1262
- Vorlesungsfolien
  - Nach jeder Vorlesung an E-Mail-Verteiler
- Leistungsnachweis
  - 90-minütige Klausur am Ende des Semesters

0 und 1 steuern die Welt

Digitaltechnik
= verstehen, wie Computer intern arbeiten
= Grundlage für gute Softwareentwicklung

Rechnertechnik II baut auf diesen Grundlagen auf



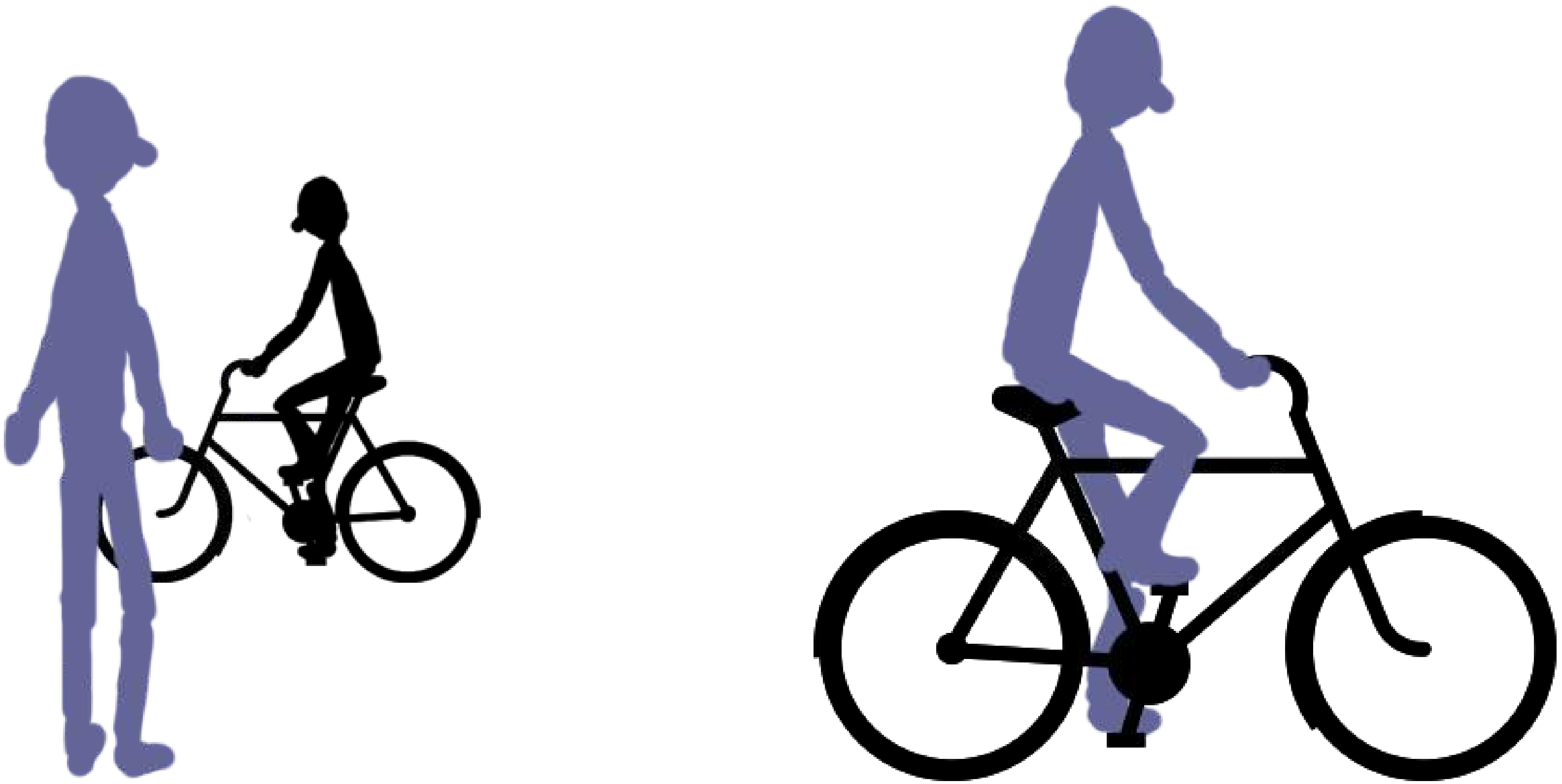
- **Warum ist das binäre System so wichtig?**
- **Wie kann man mit Nullen und Einsen rechnen?**
- **Wie werden Operationen elektronisch realisiert?**
- **Was sind wichtige elementare Bauelemente?**

Eigentlich selbstverständlich...

Bewährte Regeln für effizientes gemeinsames Arbeiten, Besprechungen usw. → gilt auch für diese Vorlesung:

- Pünktlichkeit (Vorlesungsbeginn, Pausenende)
- Anwesenheit: von Körper + Geist
- Anzahl gleichzeitig redender Personen ≤ 1 (Ausnahme Teamarbeit)
- Konzentration auf das Geschehen
    - Laptops nur ggf. zum Mitschreiben / für Übungsaufgaben
- Handys ausschalten, keine Telefonate
- ggf. Feedback zum Arbeitsprozess
    - Stoff zu schnell / zu langsam? Pausenbedarf?

Wie funktioniert Lernen?

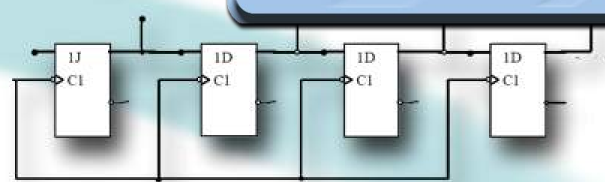


Vorlesungsthemen (s. Studienplan)

Elektronische Realisierung
- Dotierung, np-Übergänge ...
- Elementare Gatter
- Technologien (TTL, CMOS, ...)
- ...

Standardbaugruppen
- Flip-Flops
- Zähler
- Schieberegister
- ...

Schaltalgebra
- Logische Verknüpfungen
- Gatter + Schaltnetze
- Schaltungstransformation



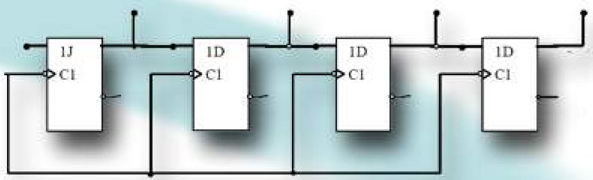
Übungen

Zahlentheorie
- Binärcodierung
- Hexadezimal usw.
- Binäres Rechnen

00100110 11101101
11011010 11101101
11101101 01110110
11110110 01110110
01110110 00100110
11011010 01110110

Vorlesungsziele (s. Studienplan)

- Grundlagen der Digitaltechnik verstehen
- Bool'sche Algebra anwenden, Umsetzung bool'scher Funktionen und Gleichungen verstehen
- Sinn und Funktionsweise digitaler Standardbaugruppen erfassen



- Interesse wecken für das Innere des Computers
- Solide Basis schaffen für die darauf aufbauenden Themen (insbesondere Rechnertechnik II)
- Praktische Erfahrungen (Labor)

00100110 11101101
11011010 11101101
11101101 01110110
11110110 01110110
01110110 00100110
11011010 01110110

Digitaltechnik

1. Zahlentheorie

Prof. Dr. Eckhard Kruse

DHBW Mannheim

Zahlen bitte!

1974_{10}

$7B6_{16}$

11110110110_{2}

3666_{8}

$\text{MCMLXXIV}_{\text{römisch}}$

Alle Darstellungen repräsentieren den gleichen Wert: 1974

Wie funktionieren Zahlen?

Beispiel

Dezimal:

1974 = 1×1000 + 9×100 + 7×10 + 4×1

→ „Zehnersystem“:

- Zehn verschiedene Ziffern: 0 1 2 3 4 5 6 7 8 9
- Wertigkeiten der Stellen verzehnfachen sich

Hexadezimal:

7B6 = 7×256 + 11×16 + 6×1 = 1974

→ „Sechzehnersystem“:

- Sechzehn verschiedene Ziffern: 0 1 2 3 4 5 6 7 8 9 A B C D E F
- Wertigkeiten der Stellen versechzehnfachen sich

Die zuvor gezeigten Darstellungen (abgesehen von den römischen Zahlen) sind **Stellenwertsysteme.**

Stellenwertsysteme

Definition

Ein Stellenwertsystem zur Basis *b*

- Es gibt b verschiedene Ziffersymbole für die Werte 0 bis $b\text{-}1$

  $z_i \in \{ symbol_0, symbol_1, \ldots, symbol_{b-1} \}$

- Eine Zahl wird durch eine Gruppe von n Ziffern z_i dargestellt:

  $z_{n-1} \ldots z_2 \, z_1 \, z_0$

- die Stellen haben die Wertigkeiten: $b^{n-1}, \ldots b^3, b^2, b, 1$

Der dargestellte Wert w berechnet sich somit als:

$$w = z_{n-1} b^{n-1} + \ldots + z_2 b^2 + z_1 b^1 + z_0 b^0 = \sum_{i=0}^{n-1} z_i b^i$$

Mein eigenes Zahlensystem

Übung

Übung

1.1 Mein eigenes Zahlensystem

Erfinden Sie Ihr eigenes Zahlensystem mit Hilfe der von Spielkarten bekannten Symbole:

♦ ♥ ♠ ♣

a) Definieren Sie die Wertigkeit der Ziffern
b) Schreiben Sie die Zahlen 3, 10, 20, 64, 255, 1024
c) Welches ist die größte mit 6 Stellen darstellbare Zahl ?
d) Wie würden Zahlensysteme mit weniger Symbolen aussehen?
e) Wieviele Symbole benötigen Sie mindestens?

Minus und Nachkommastellen

Definition

Wie funktionieren z.B. -7 0,0002 -0,345 ?

Negatives Vorzeichen

Das negative Vorzeichen – (minus) multipliziert den angegebenen Wert mit -1.

Nachkommastellen im Stellenwertsystem zur Basis b

- Die rechts neben dem Komma stehenden Stellen haben die Wertigkeiten: $b^{-1}, b^{-2}, b^{-3}, \ldots b^{-n}$

- Der dargestellte Wert w berechnet sich somit als: $w = \sum_{i \in Z} z_i b^i$

(wobei i<0: stellen rechts des Kommas, i≥0 Stellen links des Kommas)

Das funktioniert für beliebige Zahlensysteme, ist aber unüblich für andere als das dezimale System.

Warum gerade dezimal?

Warum ist das Dezimalsystem so beliebt?

Von Menschen und Computern

Menschen haben/mögen Zeh(n)en

0123456789

Wir sind an unser Dezimalsystem gewöhnt, aber genau zehn Ziffern zu unterscheiden, ist eigentlich (fast*) willkürlich.
(* s. links)

Computer mögen es einfach

01

Ein einfacheres Zahlensystem, d.h. mit weniger Ziffern/Zuständen, gibt es nicht.

Wie funktionieren Binärzahlen?

Beispiel

Dezimal:

1974 = 1×1000 + 9×100 + 7×10 + 4×1

→ „Zehnersystem“:

- Zehn verschiedene Ziffern: 0 1 2 3 4 5 6 7 8 9
- Wertigkeiten der Stellen verzehnfachen sich

Binär:

11110110110 = 1×1024 +1×512 + 1×256 + 1×128+ 1×32 + 1×16 + 1×4 + 1×2 = 1974

→ „Zweiersystem“:

- Zwei verschiedene Ziffern: 0 1
- Wertigkeiten der Stellen verdoppeln sich

Beispiele ...

Immer wieder benötigt: Die Zweierpotenzen!

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536

Binärsystem

Definition

Binärsystem (auch: Dualsystem)

`11011011010011011000`

- 1679 von Leibnitz begründet
- ideal für (digitale*) Computer:
  - Sehr einfache Abbildung von Zahlen in die elektronische Welt: Strom fließt/fließt nicht
  - robust/störunanfällig: Kleinere Strom-/Spannungsschwankungen spielen keine Rolle

(* es gibt auch Analog-Computer)

Wo im Alltagsleben gibt es binäre Darstellungen / Zustände?

Binärsystem

Definition

Binärsystem (auch: Dualsystem)

`11011011010011011000`

- 1679 von Leibnitz begründet
- ideal für (digitale*) Computer:
  - Sehr einfache Abbildung von Zahlen in die elektronische Welt: Strom fließt/fließt nicht
  - robust/störunanfällig: Kleinere Strom-/Spannungsschwankungen spielen keine Rolle

(* es gibt auch Analog-Computer)

Binär - wo noch?

- Fußgängerampel (rot oder grün)
- Elektrische Signale oder Schaltzustände, z.B. Lichtschalter
- Magnetische Speicherung auf Festplatten
- Aussagenlogik (wahr oder falsch)
- Morsealphabet (langes und kurzes Signal, na ja nicht ganz... die Pausen sind auch wichtig)
- Entscheidungen: Ja/nein
- Meldungen im Kurs

Binärdarstellung in Computern

Definition

Bit (binary digit)

Kleinste Informationseinheit: eine binäre Stelle (0 oder 1)

Byte

Zusammenfassung von 8 Bit: 2^8 = 256 mögliche Werte (0-255)
z.B. 10101100

Word, Long Word

Zusammenfassungen von 2, 4 oder 8 Byte
2-Byte-Werte: 0 – 65535 4-Byte-Werte: 0 – 4294967295
Word-Breite hängt vom Rechner ab (vgl. 16-/32-/64-Bit-CPU)

Die Darstellung von negativen Zahlen, reellen Zahlen, Buchstaben, Texten, Maschinensprachbefehlen usw. beruht letztlich auch immer auf einer Umsetzung in die binäre Darstellung.

z.B. ASCII-Code für Buchstaben: 'A' = 65 = 01000001

Binär, dezimal ...

Binärdarstellung:
- benötigt viel Platz
- für Menschen ungewohnt, nicht intuitiv
+ nah am Rechner

Dezimaldarstellung:
+ vertrautes, intuitives System
+ platzsparend
- fern der internen Rechnerdarstellung

Gibt es einen guten Kompromiss?

Hexadezimal- und Oktalsystem

Binärdarstellung:
- benötigt viel Platz
- für Menschen ungewohnt, nicht intuitiv
+ nah am Rechner

Dezimaldarstellung:
+ vertrautes, intuitives System
+ platzsparend
- fern der internen Rechnerdarstellung

Das beste aus beiden Welten ...

Hexadezimal   Oktal

Außerdem: Passt gut zur Bytegröße!

Hexadezimalsystem

Definition

Hexadezimalsystem

- von griech. "hexa" (6) und lat. "decem" (10) → 16
- Stellenwertsystem zur Basis 16
- Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  (gelegentlich auch mit Kleinbuchstaben)
- Ideal als Kurzschreibweise für Binärzahlen und für die Darstellung von Computerspeicherinhalten:
  - zwei Ziffern = ein Byte (0-255) = 8 Bit
  - Verschiedene Schreibweisen, z.B. 0xF1, $F1, $F1_{16}$, $F1_{hex}$

	E	5	2	F	= 58671
Wertigkeiten	4096	256	16	1	

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

Hexdump

Beispiel

```
00002890  00 00 00 e8 28 7f 00 00  85 c0 74 05 83 c3 08 eb  |....(.....t.....|
000028a0  cf a1 ac b9 05 08 83 c4  3c 5b 5e 5f 5d c3 80 3b  |........<[^_]..;|
000028b0  2b 0f 84 e1 00 00 00 c7  44 24 10 04 00 00 00 8b  |+.......D$......|
000028c0  15 30 b7 05 08 c7 44 24  0c 34 71 05 08 c7 44 24  |.0....D$.4q...D$|
000028d0  08 20 71 05 08 89 54 24  14 89 5c 24 04 c7 04 24  |. q...T$..\$...$|
000028e0  d4 6f 05 08 e8 77 78 00  00 8b 04 85 34 71 05 08  |.o...wx.....4q..|
000028f0  83 f8 01 0f 84 85 00 00  00 83 f8 01 72 65 83 f8  |............re..|
00002900  02 74 45 83 f8 03 75 99  c7 04 24 02 00 00 00 e8  |.tE...u...$.....|
00002910  ac 7e 00 00 85 c0 74 89  31 db be 1c b7 05 08 c7  |.~....t.1.......|
00002920  44 24 08 02 00 00 00 8b  14 9e c7 04 24 00 00 00  |D$..........$...|
00002930  00 89 54 24 04 e8 06 e9  ff ff 89 04 9e 43 83 fb  |..T$.........C..|
00002940  01 76 dc e9 59 ff ff ff  bd df 6f 05 08 bb 04 70  |.v..Y.....o....p|
00002950  05 08 89 2d 1c b7 05 08  89 1d 20 b7 05 08 e9 3e  |...-...... ....>|
00002960  ff ff ff be e9 6f 05 08  b9 e9 6f 05 08 89 35 20  |.....o....o...5 |
00002970  b7 05 08 89 0d 1c b7 05  08 e9 23 ff ff ff b8 01  |..........#.....|
00002980  70 05 08 bf 01 70 05 08  a3 20 b7 05 08 89 3d 1c  |p....p... ....=.|
00002990  b7 05 08 e9 09 ff ff ff  c7 44 24 04 0a 00 00 00  |.........D$.....|
000029a0  8d 7b 01 89 3c 24 e8 f5  e7 ff ff 85 c0 89 c6 74  |.{..<$.........t|
000029b0  6f c7 44 24 04 0a 00 00  00 8d 68 01 89 2c 24 e8  |o.D$......h..,$.|
000029c0  dc e7 ff ff 85 c0 74 42  89 3c 24 e8 40 95 00 00  |......tB.<$.@...|
000029d0  c7 44 24 08 05 00 00 00  89 c3 c7 44 24 04 10 70  |.D$........D$..p|
000029e0  05 08 c7 04 24 00 00 00  00 e8 52 e8 ff ff 89 44  |....$.....R....D|
000029f0  24 08 89 5c 24 0c c7 44  24 04 00 00 00 00 c7 04  |$..\$..D$.......|
00002a00  24 01 00 00 00 e8 16 ec  ff ff c6 06 00 89 ee 89  |$...............|
```

Adresse Speicherinhalte ASCII-Darstellung

Hexadezimal und binär

Übung

Übung

1.2 Hexadezimal und binär

Welchen (dezimalen) Wert haben die folgenden Zahlen:

```
Hex:    F1, 37, 40, A0, B7, 231A, A000, 10000
Binär: 11010011, 00110011, 10101010, 11111111,
       00001010 11010001,  11111111 11111111
```

(die untere Zeile sind 16-Bit-Zahlen, die Leerzeichen sind zur Übersicht eingefügt.)

Zahlensysteme: Umrechnung

Beispiel

Binär → Dezimal: Addiere die Wertigkeitspositionen

2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1

$$64 + 0 + 16 + 0 + 4 + 0 + 1 = 85_{10}$$

Hexadezimal → Dezimal: Addiere die Wertigkeitspositionen * Ziffernwert

16^3	16^2	16^1	16^0
A	3	5	B

$$10*4096 + 3*256 + 5*16 + 11*1 = 41819_{10}$$

Zahlensysteme Umrechung

Übung

Und wie funktioniert es von dezimal nach hexadezimal / binär?

Dezimal in binär umrechnen

Umrechnung Dezimalzahl in Binärzahl:

1. Dividiere durch 2, notiere den Rest
2. Solange Divisionsergebnis ungleich 0: Wiederhole 1.
3. Die Restbeträge von rechts nach links notiert ergeben die Binärzahl.

137_{10} →

$137 : 2 = 68$	Rest	1
$68 : 2 = 34$	Rest	0
$34 : 2 = 17$	Rest	0
$17 : 2 = 8$	Rest	1
$8 : 2 = 4$	Rest	0
$4 : 2 = 2$	Rest	0
$2 : 2 = 1$	Rest	0
$1 : 2 = 0$	Rest	1

Ende der Berechnung

→ 10001001_2

Dezimal in hexadezimal umrechnen

Umrechnung Dezimalzahl in Hexadezimalzahl:

1. Dividiere durch 16, notiere den Rest
2. Solange Divisionsergebnis ungleich 0: Wiederhole 1.
3. Der Restbeträge von rechts nach links als Hexadezimalziffern notiert ergeben die Hexadezimalzahl.

51357_{10} →

51357 : 16 = 3209 Rest	13
3209 : 16 = 200 Rest	9
200 : 16 = 12 Rest	8
12 : 16 = 0 Rest	12
Ende der Berechnung	

→ $C89D_{16}$

Hex ↔ bin ↔ dez

Übung

Übung

1.3 Hex ↔ bin ↔ dez

Transformieren Sie die folgenden Zahlen jeweils in die anderen beiden Zahlensysteme:

a) 11110000_2, 120_{10}, $F3_{16}$

b) 1011010001_2, 2500_{10}, $1E1F_{16}$

c) 11011011101_2, 10000_{10}, $AAAA_{16}$

d) Optional: Stellen Sie sich gegenseitig weitere Aufgaben.

Das Horner-Schema

Horner-Schema: Umrechnung binär → dezimal

1. Starte mit Dezimalergebnis 0
2. Multipliziere das Ergebnis mit 2 und addiere, von links beginnend, die nächste Ziffer der Binärzahl.
3. Gehe zu 2. bis die letzte Ziffer der Binärzahl abgearbeitet ist.

1 0 0 1 0 1 0 1

Anfang						0
Schritt 1	0 x 2	+	1	=	1	
Schritt 2	1 x 2	+	0	=	2	
Schritt 3	2 x 2	+	0	=	4	
Schritt 4	4 x 2	+	1	=	9	
Schritt 5	9 x 2	+	0	=	18	
Schritt 6	18 x 2	+	1	=	37	
Schritt 7	37 x 2	+	0	=	74	
Schritt 8	74 x 2	+	1	=	**149**	

Hex ↔ bin ↔ dez

Übung

Wie würde das Hornerschema für die Umrechnung von Hexadezimalzahlen aussehen?

Übung

Übung

1.4 Das Horner-Schema

a) Wenden Sie das Horner-Schema zur Umrechnung der folgenden Binärzahlen an:

110101, 1100011

b) Wenden Sie das Horner-Schema zur Umrechnung der folgenden Hexadezimalzahlen an:

3F, F13, EC90

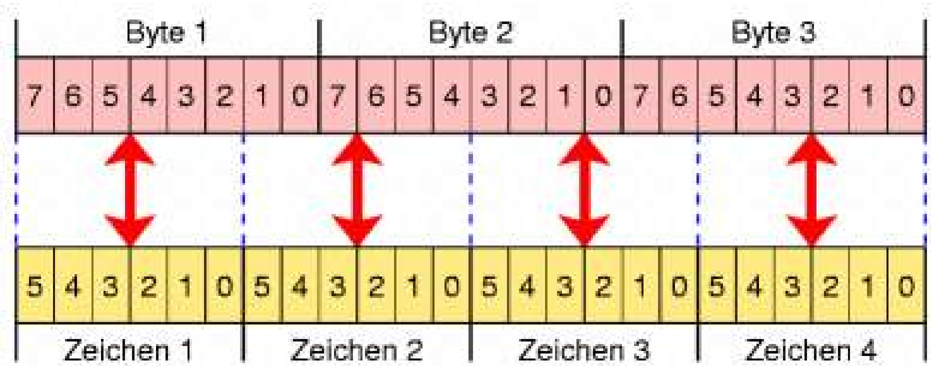
c) Vergleichen Sie das Horner-Schema mit dem zuvor vorgestellten Umrechungsverfahren (basierend auf den Stellenwerten der Ziffern). Nennen Sie Vor-/Nachteile.

Noch mehr Stellenwertsysteme...

Stellenwertsysteme zu einer Basis x spielen in der Informatik auch bei der Codierung von Binärdaten mit Hilfe darstellbarer ASCII-Zeichen eine wichtige Rolle.

Base… Encodings:

- Base32
  `A-Z, 2-7`
- Base32hex (DNSSEC)
  `0-9, A-V`
- Base58 (Base58check: Bitcoin Adr.)
- Base64 (E-Mails, MIME)
  `A-Z, a-z, 0-9, +, /    =`
- Base85 (Postscript)
- Base45 (Qr Codes)



(Bildquelle: Wikipedia)