Disclaimer

Die Inhalte dieser Vorlesung (Schriftlich als auch mündlich) stehen in keinerlei Verbindung zur Exxeta AG (Albert-Nestler-Straße 19, 76131 Karlsruhe). Der Dozent Valentin Müller trägt alleinig die Verantwortung für die Ausgestaltung und die Inhalte der Vorlesung.

Grundlagen der IT

WWI22EG/EH

Valentin Müller

Valentin Müller

Associate Consultant – Digital Sales Solutions

- Business Analyst & Scrum Master
  (Enterprise Architecture Management, Automobil-Bank)

Dualer Student WI E-Health – DHBW Mannheim

- Project Management Office (Curalie)
- Web-Entwickler (Fresenius Netcare)
- Project Management Office (Fresenius Kabi)

valentin.mueller@exxeta.com +49 172 9883410 https://www.linkedin.com/in/valentin-lauritz-müller-79a41b189/

Vorstellungsrunde

- Name
- Unternehmen
- E-Government oder E-Health?
- Warum habt ihr den Studiengang gewählt?

Organisation

Modul: Grundlegende Konzepte der IT

1. Semester

Grundlagen der IT

Grundlagen	**Rechnerarchitektur**
• Geschichtliche Entwicklung • Zahlendarstellungen (binär, hexadezimal) • Komplementdarstellungen • Fließkommadarstellung • arithmetische Operationen • Zeichensätze (ASCII, Unicode)	• Komponenten eines Rechnersystems • Von-Neumann Architektur • Interrupts, Asynchronität • Ereignissteuerung • Memory Management (HW) • Parallele Architekturen

Präsenzzeit: 24 Stunden je 45 Minuten
Selbststudium: 36 Stunden

2. Semester

Kommunikations- und Betriebssysteme

Kommunikationssysteme	Betriebssysteme

Modulhandbuch EH: https://www.dhbw.de/fileadmin/user/public/SP/MA/Wirtschaftsinformatik/E-Health.pdf

Modulhandbuch EG: https://www.dhbw.de/fileadmin/user/public/SP/MA/Wirtschaftsinformatik/E-Government.pdf

Vorlesungstermine

Datum	Uhrzeit	Format
Di, 04.10.22	13:30 - 16:30 Uhr (4 Stunden)	Vor-Ort
Di, 25.10.22	16:15 – 20:00 Uhr (9 Stunden)	Online
Fr, 28.10.22	16:00 – 17:30 Uhr (11 Stunden)	Online
Fr, 04.11.22	16:00 – 17:30 Uhr (13 Stunden)	Online
Do, 10.11.22	18:00 – 20:00 Uhr (15 2/3 Stunden)	Online
Mi, 16.11.22	18:00 – 20:00 Uhr (18 1/3 Stunden)	Online
Di, 29.11.22	16:30 – 18:30 Uhr (21 Stunden)	Online / Vor-Ort
Mi, 30.11.22	13:00 – 17:00 Uhr (26 1/3 Stunden)	Online / Vor-Ort
Di, 06.12.22	16:00 – 20:00 Uhr	Online
Mi, 07.12.22	16:00 – 20:00 Uhr	Online

Klausur: Am Ende des 2. Semester
Dateiablage & Online-Raum: https://moodle.dhbw-mannheim.de/course/view.php?id=9669 (Grundlagen der IT (Hr. V. Müller))

Literatur

- **Fit fürs Studium – Informatik | Boockmeyer, Fischbeck & Neubert | Rheinwerk Computing, 2018 | 978-3-8362-4406-0**
- **Grundkurs Informatik | Hartmut Ernst, Jochen Schmidt, Gerd Beneken | Springer Vieweg, 2020 | 978-3-6583-0331-0**
- Einführung in die Informatik | Heinz Peter Gumm, Manfred Sommer | De Gruyter, 2012 | 978-3-4867-1995-6
- Informationstechnik: Hardware – Software – Netzwerke | Peter Bühler, Patrick Schlaich, Dominik Sinner |Springer Viewweg, 2018 | 978-3-6625-4732-8

https://www.mannheim.dhbw.de/service/bibliothek/ausleihe-online-zugriff

Geschichtliche Entwicklung

Definition: IT (Informationstechnik)

„Gesamtheit der (auf den Erkenntnissen und Methoden von **Informatik und Elektrotechnik** aufbauenden) **Verfahren zur automatischen** Sammlung, Speicherung, **Verarbeitung** und Übertragung **von Informationen**." ~ *Digitales Wörterbuch der Deutschen Sprache*

„Informationstechnik (kurz IT) steht für die Technik zur **Elektronischen Datenverarbeitung (EDV)** und der hierzu verwendeten **Hard- und Software-Infrastruktur**." ~ *Wikipedia*

„Alle **technischen Mittel, die der maschinellen** oder maschinell unterstützten Erzeugung, Speicherung, **Verarbeitung** oder Übertragung **von Informationen dienen**; alle dazu erforderlichen **Komponenten** einschließlich der **Programme** und der technischen Voraussetzungen für die Kommunikation." ~ *Wirtschaftslexion24*

https://www.dwds.de/wb/Informationstechnik (Aufgerufen am 24.09.22)
https://de.wikipedia.org/wiki/Informationstechnik (Aufgerufen am 24.09.22)
http://www.wirtschaftslexikon24.com/d/informationstechnik/informationstechnik.htm (Aufgerufen am 24.09.22)

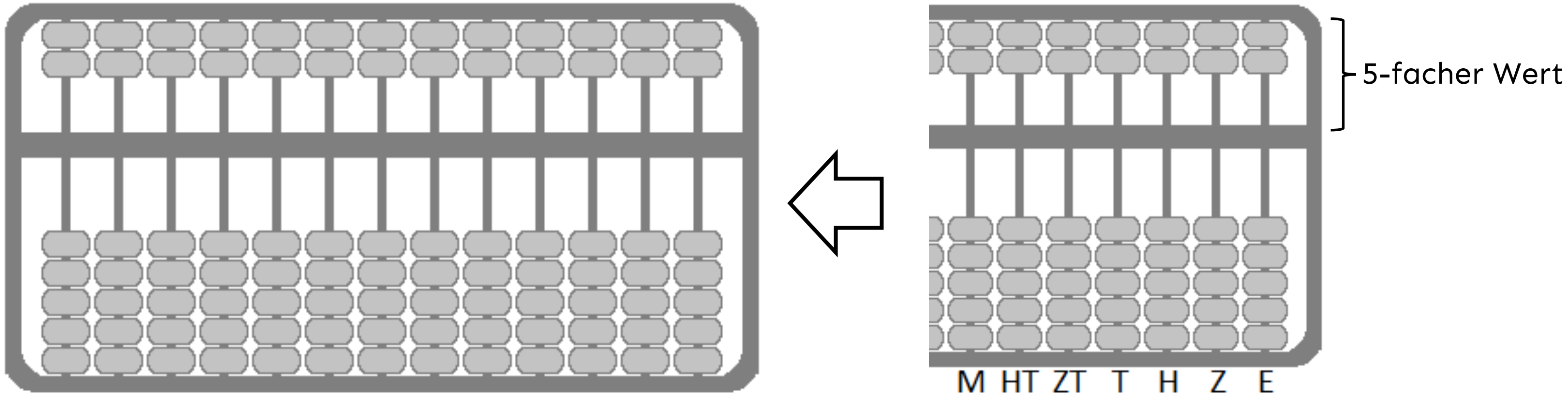
Geschichte der Informatik

Zähl- und Rechensysteme (2700-2300 v. Chr.)

Wurzeln der Informatik:

- Bestreben des Menschen durch den Einsatz von Werkzeugen und Maschinen die Arbeit zu erleichtern
- Nicht nur körperliche Arbeit, sondern auch geistige Tätigkeiten

Entstehung von Rechenhilfen – „Abakus":



http://www.mathematische-basteleien.de/abakus.htm (Aufgerufen am 24.09.2022)
https://de.wikipedia.org/wiki/Abakus_(Rechenhilfsmittel) (Aufgerufen am 24.09.2022)

Geschichte der Informatik

Mechanische Rechenmaschinen (Ab 1600)

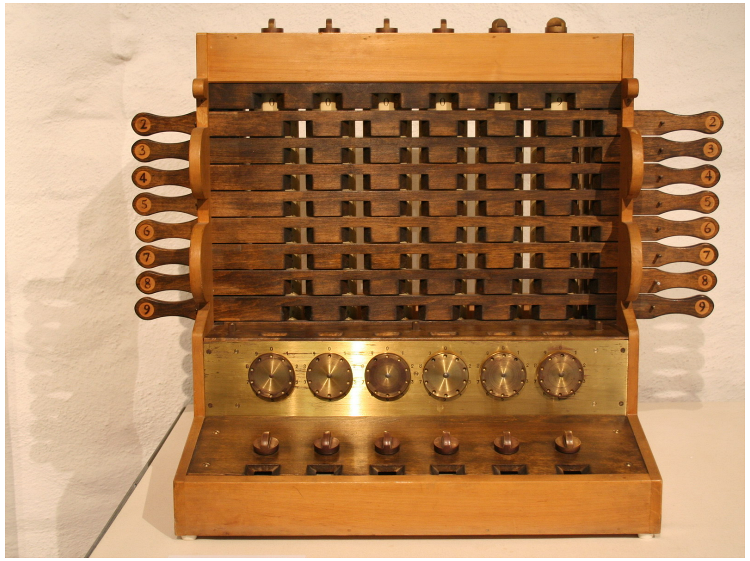
Einführung des Dezimalsystems:

- Nutzung des römischen Ziffernsystems bis ins Mittelalter (Mitte des 16. Jhr.)
- Adam Ries(e) brachte das indisch-arabische Dezimalsystem nach Europa

Mechanische Rechenmaschinen:

- Ersetzung von Rechensteinen durch Zahnräder

- Älteste dokumentierte Addiermaschine stammt von Wilhelm Schickard (1624)
- Weiterentwicklung der mechanische Rechenmaschinen durch Blaise Pascal und Gottfried Wilhelm Leibniz



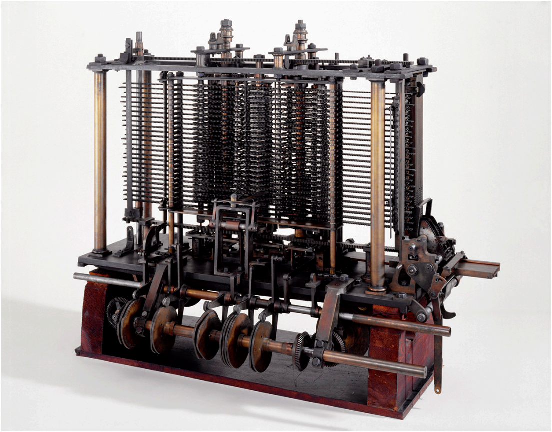
https://de.wikipedia.org/wiki/Wilhelm_Schickard#/media/Datei:Schickardmaschine.jpg (Aufgerufen am 24.09.2022)

Geschichte der Informatik

Computer-Konzept und Lochkarten (Ab 1830)

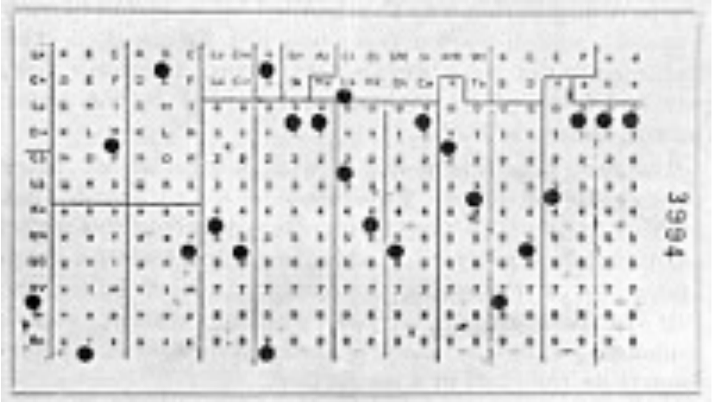
Computer-Konzept:

- Charles Babbage entwirft die „Analytical Engine“
- Bestandteile: Rechenwerk, Steuerwerk, Speicher, Eingabe und Ausgabe



Lochkarten:

- Mechanisches Speichermedium
- Metallstift fährt über die Lochkarte und berührt einen elektrischen Kontakt, wenn ein Loch eingefügt ist
- Nutzung bei der Volkszählung 1890 in den USA



https://de.wikipedia.org/wiki/Analytical_Engine#/media/Datei:Babbages_Analytical_Engine,_1834-1871._(9660574685).jpg (Aufgerufen am 24.09.2022)

Geschichte der Computer

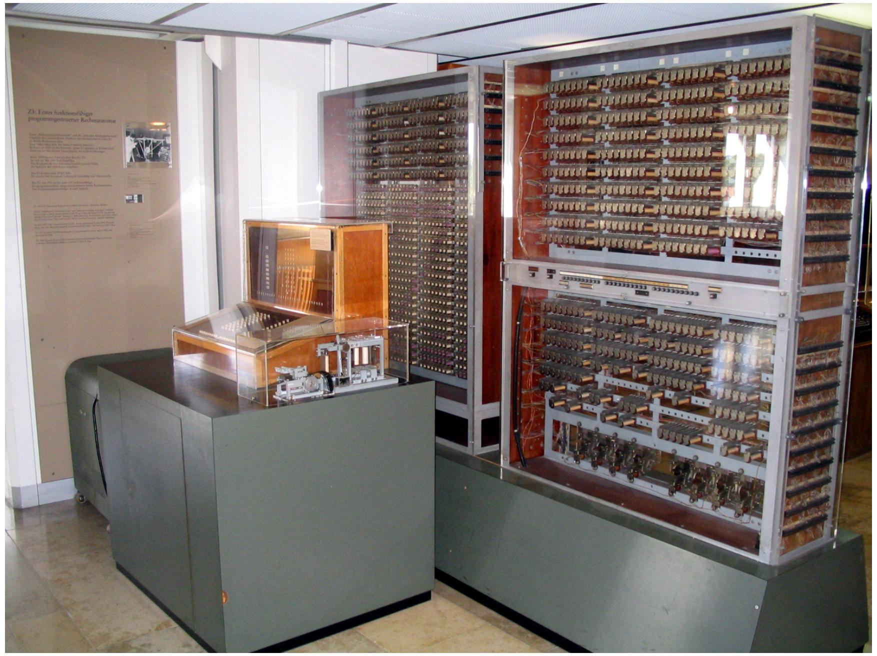
0. Generation – Elektromechanische Rechenmaschinen (Ab 1910)

Z1

- Erster Computer, der nach dem Prinzip von Babbage funktionierte
- Anstelle von mechanischen Zahnrädern wurden elektromechanische Schaltelemente (Relais) genutzt

Z3

- Erster voll funktionsfähiger programmierbarer Computer der Welt
- Programme wurden über Lochstreifen eingegeben



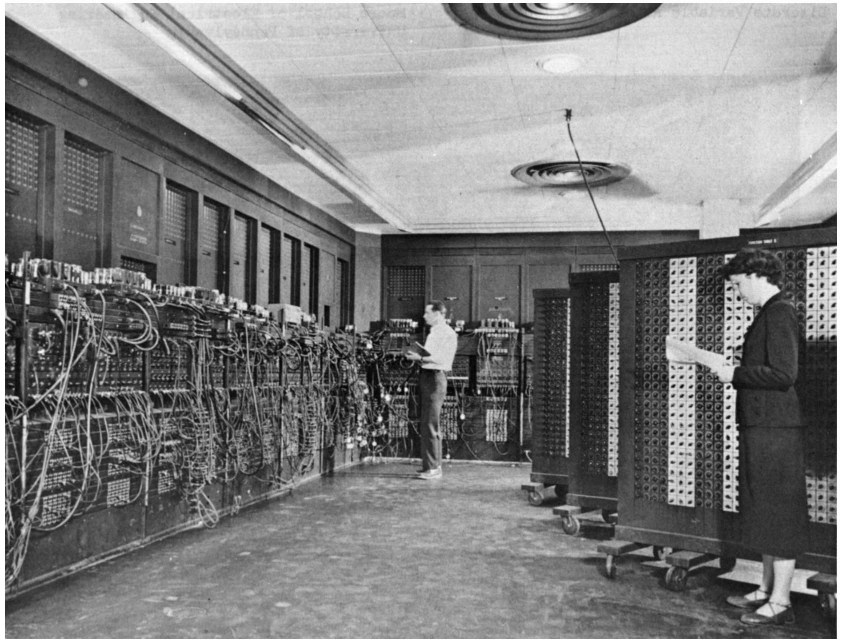
https://de.wikipedia.org/wiki/Zuse_Z3#/media/Datei:Z3_Deutsches_Museum.JPG (Aufgerufen am: 22.09.2022)

Geschichte der Computer

1. Generation – Röhrenrechner (Ab 1946)

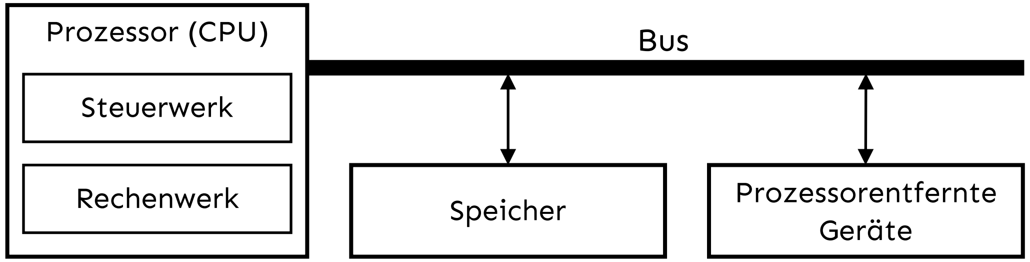
ENIAC (Electronic Numeric Integrator and Computer)

- Erster mit Elektronenröhren arbeitende Computer
- Bestand aus über 18.000 Röhren und nahm 140m^2 in Anspruch
- Nutzung zur Berechnung von Bahnen für Flugkörper



Von-Neumann-Architektur

- Sequentielle Abarbeitung von Programmen
- Einheitlich genutzter Speicher für Programme und Daten



https://de.wikipedia.org/wiki/ENIAC#/media/Datei:Eniac.jpg (Aufgerufen am 24.09.2022)

Geschichte der Computer

2. Generation – Transistoren-Rechner (Ab 1955)

- Elektronenröhren waren fehleranfällig und hatten einen hohen Energie- und Platzverbrauch
- Lösung waren kleinere, sparsamere und schnellere Transistoren
- Firmen, wie IBM und Siemens, bauen erste serienmäßige Computer
- Höhere Programmiersprachen, wie FORTRAN und COBOL entstanden

Elektronenröhren

Transistoren

https://de.wikipedia.org/wiki/Elektronenröhre#/media/Datei:Radio_vacuum_tubes.jpg (Aufgerufen am 24.09.2022)
https://de.wikipedia.org/wiki/Transistor#/media/Datei:Transistors-white.jpg (Aufgerufen am 24.09.2022)

Geschichte der Computer

3. Generation – Integrierte Schaltkreise (Ab 1960)

Integrierter Schaltkreis

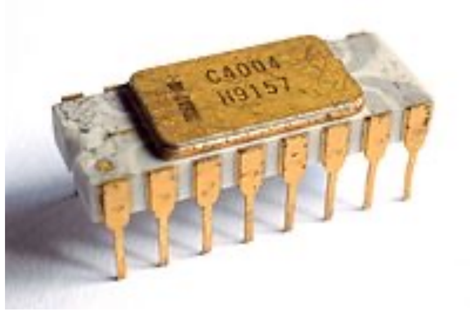
- Technische Verbesserungen ermöglichten es immer mehr Transistoren und andere elektronische Bauteile zu verbauen
- Elektronische Schaltung, die auf einem wenige Millimeter großen Halbleiterplättchen Platz findet

Geschichte der Computer

4. Generation – Mikroprozessoren (Ab 1970)

Mikroprozessoren

- Technische Verbesserungen verkleinerten die Integrierten Schaltkreise
- Anfang der 1970er Jahre konnten so viele integrierte Schaltkreise auf einem Chip integriert werden, um eine vollständige CPU abzubilden



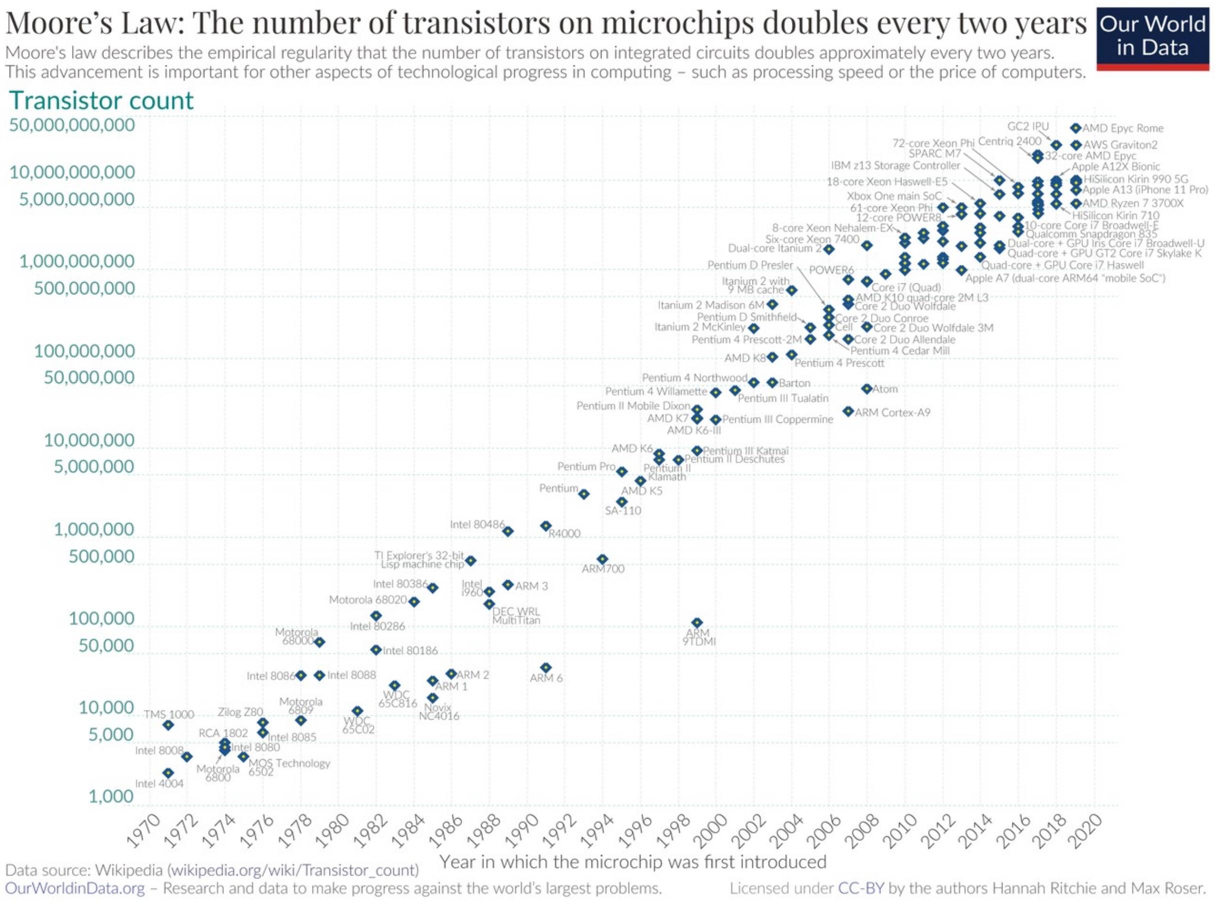
Intel 4004
(4-Bit-Mikroprozessor)

TMS1000
(4-Bit-Mikroprozessor)

https://de.wikipedia.org/wiki/Intel_4004 (Aufgerufen am 24.09.2022)
https://de.wikipedia.org/wiki/Texas_Instruments_TMS1000#/media/Datei:TI_TMS1000NP_1.jpg (Aufgerufen am 24.09.2022)

Moores Law (Mooresches Gesetz)

- Gordon Moore (Mitgründer von Intel) stellte 1965 die Beobachtung auf, dass sich die Zahl der Transistoren in integrierten Schaltungen regelmäßig verdoppelt
- Je nach Quelle beträgt der von Moore genannte Zeitraum 12, 18 oder 24 Monate
- Das Gesetz hat sich bis heute als korrekt erwiesen → Ca. alle 20 Monate findet eine Verdopplung statt



https://de.wikipedia.org/wiki/Mooresches_Gesetz#/media/Datei:Moore's_Law_Transistor_Count_1970-2020.png (Aufgerufen am 01.10.2022)

Zahlen-darstellungen

Dezimalsystem (Zehnersystem)

- **Standardsystem zur Beschreibung von Zahlen**
- **Ziffern pro Stelle:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

5284 =

Tausender	Hunderter	Zehner	Einer
5	2	8	4
5×10^3	2×10^2	8×10^1	4×10^0
5000	200	80	4

$= 5 \times 10^3 + 2 \times 10^2 + 8 \times 10^1 + 4 \times 10^0 = 5284$

$Matisse \times Basis^{Exponent}$ ⇨ Basis = 10 → Zehner- bzw. Dezimalsystem

Binärsystem (Dualsystem)

- **Standardsystem in der Informatik**
  - Zwei Zustände können in verschiedenen Formen dargestellt werden
  - Bspw. Spannung = 1; Keine Spannung = 0
- **Ziffern pro Stelle:** 0, 1

Bits (*)

$10101_2 =$

16	8	4	2	1
1	0	1	0	1
1×2^4	0×2^3	1×2^2	0×2^1	1×2^0
16_{10}	0_{10}	4_{10}	0_{10}	1_{10}

$$= 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 = 21_{10}$$

$Matisse \times Basis^{Exponent}$ ⇨ Basis = 2 → Binärsystem bzw. Dualsystem

* 8 Bits = 1 Byte

Hexadezimalsystem

- **Häufige genutzte Alternative zum Binärsystem in der Informatik**
  - Binärzahlen werden sehr lange, deshalb Darstellung im Hexadezimalsystem
  - Einfache Umwandlung von Binär- in Hexadezimalzahlen und umgekehrt
- **Ziffern pro Stelle:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(=10), B(=11), C(=12), D(=13), E(=14), F(=15)

$0xA2D =$

256	16	1
A	2	D
10×16^2	2×16^1	13×16^0
2560_{10}	32_{10}	13_{10}

$= 10\times16^2 + 2\times16^1 + 13\times16^0 = 2605_{10}$

$Matisse \times Basis^{Exponent}$ ➡ Basis = 16 → Hexadezimalsystem

Umwandlung: Binärzahl ←→ Dezimalzahl

Binärzahl → Dezimalzahl

- *Siehe Folie „Binärsystem (Dualsystem)"*

Dezimalzahl → Binärzahl

$$\left.\begin{array}{l} \;\;21 \div 16 = \mathbf{1} \\ \underline{-16\;\;} \\ \;\;\;\;5 \div 8 = \mathbf{0} \\ \underline{\;\;\;-0\;\;} \\ \;\;\;\;5 \div 4 = \mathbf{1} \\ \underline{\;\;\;-4\;\;} \\ \;\;\;\;1 \div 2 = \mathbf{0} \\ \underline{\;\;\;-0\;\;} \\ \;\;\;\;1 \div 1 = \mathbf{1} \\ \underline{\;\;\;-1\;\;} \\ \;\;\;\;0 \end{array}\right\} = 10101_2$$

- Funktioniert auch in andere Zahlensysteme
- Jeweils größten Potenz der entsprechenden Basis

Umwandlung: Dezimalzahl → Binärzahl

Restwertmethode

Ablauf

- Fortlaufende Division der umzuwandelnden Zahl mit Rest
- **Dividend:** Umzuwandelnde Zahl (1. Schritt) / Ergebnis der vorherigen Division (2. – n. Schritt)
- **Divisor:** Basis des Zahlensystems, in das die Zahl umgewandelt wird
- **Rest:** Stellen der umgewandelten Zahl

- Funktioniert auch in andere Zahlensysteme
- Divisor wird jeweils angepasst

Beispiel

$21 \div 2 = 10\ Rest\ 1$

$10 \div 2 = 5\ Rest\ 0$

$5 \div 2 = 2\ Rest\ 1$

$2 \div 2 = 1\ Rest\ 0$

$1 \div 2 = 0\ Rest\ 1$

Beginnend bei Bit 1 aufsteigend (Rechts nach links)

$21_{10} = 10101_{10}$

Umwandlung: Hexadezimalzahl ←→ Binärzahl

Hexadezimalzahl → Binärzahl

$0xA2D =$

256	16	1
A	2	D
1010_2	0010_2	1101_2

$= 101000101101_2$

Binärzahl → Hexadezimalzahl

$0x22D =$

256	16	1
2	2	D
0010_2	0010_2	1101_2

$= 1000101101_2$

Von rechts nach links lesen

Übung: Umwandeln in verschiedene Zahlensysteme

1) $0x10\text{AE} =?_{10}$
2) $0x545 = ?_2$ (12-bit Darstellung)
3) $50_{10} = ?_2$ (Byte Darstellung)
4) $54_{10} \; = \; 0x?$
5) $0011\ 1011\ 1001_2 = 0x?$

Alle Rechnungen mit Rechenweg durchführen.

Arithmetische Operationen

Rechnen mit Binärzahlen

Grundregeln der binären Addition:

1. 0 + 0 = 0
2. 0 + 1 = 1
3. 1 + 0 = 1
4. 1 + 1 = 0 Übertrag 1

Beispiel:

$$\begin{array}{rr} & 0\,1\,0\,1\,1\,1\,0\,0_2 \\ & +\,0\,1\,0\,0\,0\,1\,1\,1_2 \\ \text{Übertrag:} & 1\,0\,1\,1\,1\,0\,0\,0 \\ \hline & 1\,0\,1\,0\,0\,0\,1\,1_2 \end{array}$$

Grundregeln der binären Subtraktion:

1. 0 – 0 = 0
2. 1 – 1 = 0
3. 1 – 0 = 1
4. 0 – 1 = 1 Übertrag -1

Beispiel:

$$\begin{array}{rr} & 0\,1\,0\,1\,1\,1\,0\,0_2 \\ & -\,0\,1\,0\,0\,0\,1\,1\,1_2 \\ \text{Negativer Übertrag:} & -\,0\,0\,0\,0\,1\,1\,1\,0 \\ \hline & 0\,0\,0\,0\,0\,1\,0\,1_2 \end{array}$$

Hinweis:
Zweifache Subtraktion nacheinander rechnen.

0 - 1 – 1 = (0 – 1) - 1

0 - 1 = 1 Übertrag -1
1 – 1 = 0

- Ausführung einer binären Subtraktion ist für Computer schwierig
- Rückführung der Subtraktion auf die Addition: $a - b = a + (-b)$

Negative Zahlen im Binärsystem

Natürlichen Zahlen
$\mathbb{N} \epsilon \{0, 1, 2, \dots\}$

$0_2, 1_2, 10_2, \dots$

Ganze Zahlen
$\mathbb{Z} \epsilon \{\dots, -2, -1, 0, 1, 2, \dots\}$

1) ~~**Vorzeichen Minus und Plus**~~

$\dots, -10_2$
$-1_2, 0_2, 1_2, 10_2, \dots$

- Digitale Schaltungen können kein Plus oder Minus darstellen

2) ~~**Höchstwertiges Bit* als Vorzeichen**~~
(1 = Minus | 0 = Plus)

$\dots, \mathbf{1}010_2, \mathbf{1}001_2, \mathbf{1}000_2,$
$\mathbf{0}000_2, \mathbf{0}001_2, \mathbf{0}010_2, \dots$

(4-Bit Darstellung)

- Positive und negative Null
- Sonderbehandlung des Vorzeichenbits → Addition/Subtraktion nach Schulrechen-methode nicht möglich

3) ~~**Einerkomplement**~~

$\dots, \mathbf{1}101_2, \mathbf{1}110_2, \mathbf{1}111_2$
$\mathbf{0}000_2, \mathbf{0}001_2, \mathbf{0}010_2, \dots$

(4-Bit Darstellung)

- Positive und negative Null
- Operationen, die die Null durchschreiten liefern zunächst ein falsches Ergebnis

* Most Significant Bit (MSB)

Zweierkomplement

Positive Zahlen

- Gewohnte Binärdarstellung: $45_{10} = 00101101_2$

Negative Zahlen

1. Positive Binärdarstellung: $45_{10} = 00101101_2$
2. Inventieren aller Bits: 11010010_2
3. Addition von 1: $11010011_2 \rightarrow -45_{10} = 11010011_2$

Vorteile

- Gleichbehandlung aller Zahlen, unabhängig vom Most Significant Bit
- Subtraktion nicht erforderlich, Addition genügt

Zweierkomplement in 8-Bit Darstellung

Umwandlung: Dezimalzahl ←→ Zweierkomplement

Dezimalzahl → Zweierkomplement

- *Siehe vorherige Folie*

Zweierkomplement → Dezimalzahl

- Erstes Bit $= 0$ (Zahl positiv) 00101101_2
  1. Umwandeln in Dezimalsystem: $00101101_2 = 45_{10}$

- Erstes Bit $\neq 0$ (Zahl negativ) 11010011_2
  1. Subtraktion von 1: 11010010_2
  2. Invertieren aller Bits: 00101101_2
  3. Umwandeln in Dezimalsystem: $00101101_2 = 45_{10}$
  4. Negatives Vorzeichen ergänzen -45_{10}

  1. Invertieren aller Bits: 00101100_2
  2. Addition von 1: 00101101_2
  3. Umwandeln in Dezimalsystem: $00101101_2 = 45_{10}$
  4. Negatives Vorzeichen ergänzen -45_{10}

Short-Cut:
- Gewohnte Umwandlung von Binär in Dezimal, aber die Wertigkeit des höchsten Bits abziehen
- $00101101_2 = 0 \times (\mathbf{-128}) + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 45_{10}$
- $11010011_2 = 1 \times (\mathbf{-128}) + 1 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = -45_{10}$

Zweierkomplement in 8-Bit Darstellung

Rechnen mit dem Zweierkomplement

Subtraktion mit positiven Ergebnis $(21_{10} - 17_{10})$:

$21_{10} = 00010101_2$

$17_{10} = 00010001_2$ (Binärdarstellung des Betrags)

11101110_2 (Invertieren aller Bits)

$11101111_2 = -17_{10}$ (Addition von 1)

$$\begin{array}{lr} & 00010101_2 \\ & +\,11101111_2 \\ \text{Übertrag:} & 11111110_2 \\ \hline & \cancel{1}00000100_2 = 4_{10} \end{array}$$

Subtraktion mit negativem Ergebnis $(17_{10} - 21_{10})$:

$17_{10} = 00010001_2$

$21_{10} = 00010101_2$ (Binärdarstellung des Betrags)

11101010_2 (Invertieren aller Bits)

$11101011_2 = -21_{10}$ (Addition von 1)

$$\begin{array}{lr} & 00010001_2 \\ & +\,11101011_2 \\ \text{Übertrag:} & 00000110_2 \\ \hline & \mathbf{1}1111100_2 \end{array}$$

00000011_2 (Invertieren aller Bits)

00000100_2 (Addition von 1)

$00000100_2 = -4_{10}$

➨ Addition erfolgt nach Schulrechenmethode

Zweierkomplement in 8-Bit Darstellung

Übung: Rechnen mit Binärzahlen

1) $73_{10} - 56_{10} = ?_2 = ?_{10}$ (8-Bit Darstellung)
2) $93_{10} + 54_{10} = ?_2 = 0x?$ (12-Bit Darstellung)
3) $22_{10} + (-7_{10}) = ?_2 = 0x?$ (8-Bit Darstellung)
4) $0x2D - 0x3A = ?_2 = ?_{10}$ (8-Bit Darstellung)

Alle Rechnungen mit Rechenweg durchführen.