

# Digitaltechnik

Wintersemester 2025/2026

Vorlesung 1



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

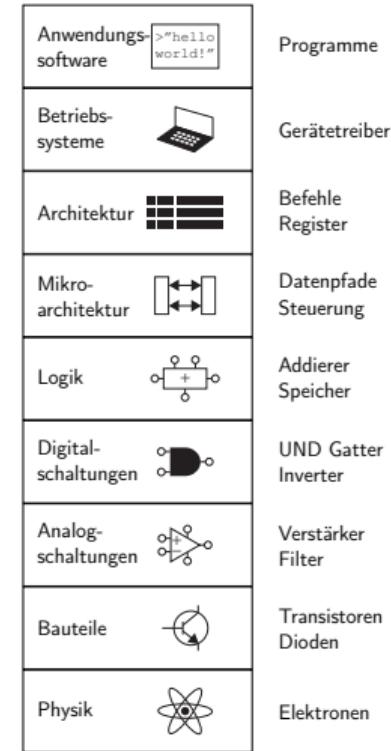


ENCRYPTO  
CRYPTOGRAPHY AND  
PRIVACY ENGINEERING

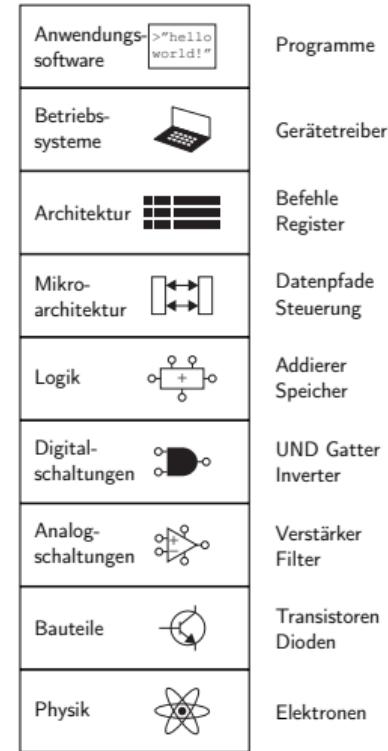


Herzlich Willkommen  
in Ihrem Studium!

- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Prof. Dr.-Ing. Thomas Schneider er/ihm
- Leiter des Fachgebiets Cryptography and Privacy Engineering (ENCRYPTO)
- 2003-2008: Dipl.-Inf. an der FAU Erlangen-Nürnberg Nebenfach Elektrotechnik
  - In meinem Studium hieß Digitaltechnik „Technische Informatik 1“ und „Technische Informatik 3“ war ganzes Semester E-Technik für Informatiker:innen - nun nur noch 1 Vorlesung in DT ☺
- 2008-2011: Promotion an der RU Bochum in IT-Sicherheit
- 2011-2018: Nachwuchsforschungsgruppenleiter an der TUDA
- seit 2018: Professor am Fachbereich Informatik



- M.Sc. Nora Khayata        sie/ihr
- Wissenschaftliche Mitarbeiterin  
Cryptography and Privacy Engineering Group (ENCRYPTO)
- Kontakt: [digitaltechnik@crypto.cs.tu-darmstadt.de](mailto:digitaltechnik@crypto.cs.tu-darmstadt.de)
- Sprechstunde: Mittwoch 15:00-16:00 oder nach Vereinbarung
- Raum: S2|20 Raum 213



- M.Sc. Jan Filipp er/ihm
- Wissenschaftlicher Mitarbeiter
- Cryptography and Privacy Engineering Group (ENCRYPTO)
- Kontakt: [digitaltechnik@crypto.cs.tu-darmstadt.de](mailto:digitaltechnik@crypto.cs.tu-darmstadt.de)
- Sprechstunde: Nach Vereinbarung
- Raum: S2|20 Raum 214



- M.Sc. Andreas Brüggemann        dey/denen/deren/dey
- Wissenschaftliche:r Mitarbeiter:in  
Cryptography and Privacy Engineering Group (ENCRYPTO)
- Kontakt: [digitaltechnik@crypto.cs.tu-darmstadt.de](mailto:digitaltechnik@crypto.cs.tu-darmstadt.de)
- Sprechstunde: Mittwoch 15:00-16:00 oder nach Vereinbarung
- Raum: S2|20 Raum 214



- <https://crypto.de>
- Mission: Wir entwerfen, optimieren und generieren kryptographische Protokolle zum Schutz von Daten in Anwendungen.
- Hierbei verwenden wir u.a. Konzepte aus der Digitaltechnik:
  - Kombinatorische Schaltkreise
  - Schaltkreisminimierung
  - Hardwaresynthesewerkzeuge



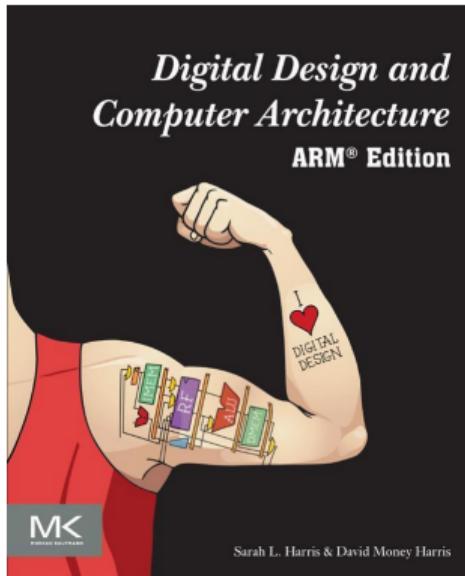
- Vorlesungs- und Übungsbetrieb entwickeln sich fortlaufend weiter.
- Wenn Sie Verbesserungsvorschläge haben, lassen Sie es uns wissen!
- Dafür können Sie das anonyme Feedback im Moodle verwenden, uns eine E-Mail schreiben ([digitaltechnik@crypto.cs.tu-darmstadt.de](mailto:digitaltechnik@crypto.cs.tu-darmstadt.de)) oder uns einfach nach der Vorlesung ansprechen.

- In unserer Lehrveranstaltung und an der gesamten TU Darmstadt sollen sich alle Menschen wohl und sicher fühlen!
  - Darum erwarten wir einen respektvollen Umgang miteinander.
  - Dazu zählt auch die Nutzung der bevorzugten Vornamen, Pronomen, usw.
  - **Wir tolerieren keine Belästigung, Diskriminierung, o.Ä.**
- Wenn Sie Belästigung, Diskriminierung, o.Ä. erfahren oder beobachten, informieren Sie uns bitte unverzüglich (das DT Team in Person, per E-Mail oder anonym in Moodle).
- Weitere Anlaufstellen:
  - Antidiskriminierungsstelle der TU: <https://www.tu-darmstadt.de/universitaet/diversity/antidiskriminierungsstelle/index.de.jsp>
  - AStA Antidiskriminierungs-Beratung:  
<https://www.astा.ту-darmstadt.de/de/antidiskriminierung>
  - AStA Queerreferat: <https://www.astा.ту-darmstadt.de/de/queerreferat>
  - Gleichstellungsteam des FB Informatik:  
[gleichstellungsbeauftragte@informatik.tu-darmstadt.de](mailto:gleichstellungsbeauftragte@informatik.tu-darmstadt.de)

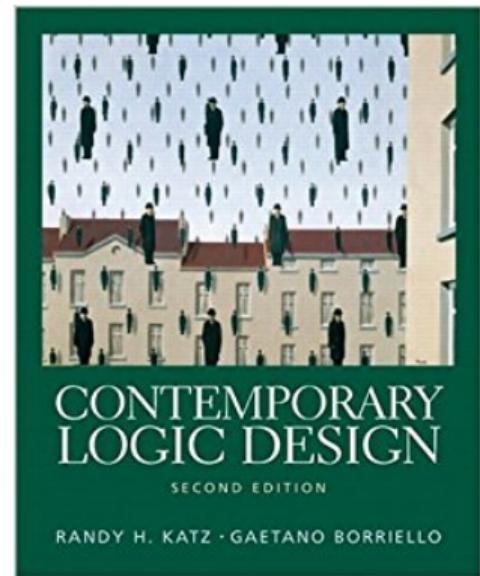
- Lernen ist ein aktiver Prozess, den **Sie** vorantreiben müssen.  
Wir wollen Sie dabei bestmöglich unterstützen.
- Lernen Sie **kontinuierlich**, nicht nur vor der Klausur.
- Stellen Sie **Fragen!**
  - In Vorlesungen
  - In Übungen
  - In Moodle Foren
- Diskutieren Sie **Ihre** Lösung in der Übung / Kleingruppe.



- **Ziel: Inhalte vermitteln & Interaktion**
- Wöchentlich (15 Vorlesungen insgesamt)
  - Donnerstag 08:00-09:40
  - S101/A1 (Audimax)
  - Einzelne Vorlesungs-Aufzeichnungen (vorwiegend alte Aufzeichnungen aus WS24/25)
- Beginn: 16.10.25
- Weihnachtspause: 22.12.2025 – 09.01.2026
  - Erste Vorlesung im neuen Jahr: 15.01.26
  - Der Übungsbetrieb wird ab dem 12.01.26 fortgesetzt
- Ende: 13.02.26 ⇒ danach Klausurvorbereitung
- Material im Moodle Kurs: <https://crypto.de/DT-Moodle>
  - Vorlesungsfolien und -aufzeichnungen, Übungen, Musterlösungen, Hausaufgaben



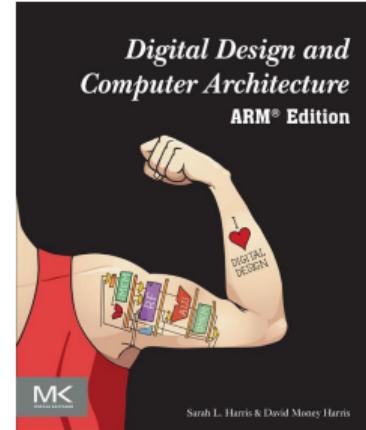
Sarah L. Harris & David Money Harris,  
*Digital Design and Computer Architecture*,  
Morgan Kaufmann, ARM Edition, 2016



Randy H. Katz,  
*Contemporary Logic Design*,  
Addison-Wesley, 2nd ed., 2005

- **Ziel: Wissen vertiefen & Details gut erklärt nachlesen**
- Die Vorlesungsfolien basieren auf Vorlagen von Sarah Harris (Weiterentwickelt durch Wolfgang Heenes und Raad Bahmani)
- Hervorragende Lehrbücher:
  - Harris & Harris: im Semesterapparat der ULB und als eBook verfügbar
  - Katz: in der Lehrbuchsammlung der ULB verfügbar
- ⇒ Wir verweisen in den jeweiligen Vorlesungen auf bestimmte Kapitel daraus (max. ≈ 20 Seiten pro Woche) und empfehlen sehr, diese *vorlesungsbegleitend* zu lesen!
- Am Ende der Kapitel v. Harris gibt es weitere Übungsaufgaben zur Klausurvorbereitung

- Online über die ACM Digital Library kostenlos verfügbar:
  - <https://dl.acm.org/doi/pdf/10.5555/2815529>
- Als eBook kostenlos über ULB beziehen:
  - <https://hds.hebis.de/ulbda/Record/HEB361909470>
- Companion Website mit Zusatzmaterialien wie Codebeispielen:
  - <https://pages.hmc.edu/harris/ddca/ddcaarm.html>



- **Ziel: Aufgaben selbstständig bearbeiten**
- 14 Übungsblätter
- Bearbeitung in Übungsgruppen (etwa 30 Personen)
- Erarbeitung von Lösungsstrategien und Besprechung verschiedener Lösungsansätze
- Lösungsvorschläge werden später bereitgestellt

- **Ziel: Eigene Lösungen in Gruppe von ca. 30 Studierenden mit Tutor:in bearbeiten**
- **Gruppenanmeldung in Moodle**
  - Bis Freitag 17.10.25 12:00: Gruppenwunsch eintragen
  - Bis Freitag 31.10.25 23:59: Gruppenwechsel möglich
    - Gruppenwechsel: Einschreibung/Wechsel in Gruppe mit freien Kapazitäten
- **Gruppenanmeldung in TUCaN nicht maßgeblich**
- **Wöchentliches Treffen von Gruppe und Tutor:in in Person**

- Klausur darf nur nach Erwerb der Studienleistung geschrieben werden.
- Studienleistung muss durch Hausaufgaben und Quiz erworben werden.
  - Schriftliche Abgaben zu 6 Hausaufgabenblättern in Moodle
    - Bearbeitung und Abgabe in Gruppen mit bis zu 3 Personen
    - 20 erreichbare Punkte pro Blatt
  - Wöchentliches Online-Quiz auf Moodle
    - Bearbeitung alleine
    - 10 erreichbare Punkte pro Quiz, insgesamt 13 Quizze
  - Studienleistung wird erworben bei
    - Mindestens **50%** der Punkte aus den Hausaufgabenblättern (60 Punkte)
    - **UND** mindestens **50%** der Punkte aus den Online-Quizzes (65 Punkte)
- Alte Studienleistungen für Digitaltechnik aus vorherigen Semestern gelten weiterhin

- Donnerstag 08:00-09:40: Vorlesung
  - Folien bereits vorher im Moodle
- Online-Quiz jede Woche
  - Veröffentlichung nach jeder Vorlesung
  - Abgabefrist im Moodle: Donnerstag 12:00 zwei Wochen später
- Montag - Freitag: Bearbeitung von Übungsblatt zur Vorlesung der vorherigen Woche in Ihrer Übungsgruppe
  - Genauer Termin entsprechend Ihrer Gruppeneinteilung
  - Musterlösung wird Freitags um 18:00 hochgeladen
- Hausaufgabenblatt zu Inhalten der vorherigen zwei Übungen und entsprechenden Vorlesungen alle zwei Wochen
  - Veröffentlichung zu Beginn jeder ungeraden KW
  - Abgabefrist im Moodle: Freitag 23:59 zwei Wochen später

Eine Tabelle mit den Deadlines finden Sie auch im Moodle!

Kalenderwoche	Vorlesung	Online-Quiz	Übung	Hausaufgabe
KW42	1(Do)		-	
KW43	2(Do)	1	1	
KW44	3(Do)	2	2	
KW45	4(Do)	3	3	
KW46	5(Do)	4	4	
KW47	6(Do)	5	Donnerstag 12:00	
KW48	7(Do)	6	Freitag 23:59	C (5+6)
KW49	8(Do)	7		D
KW50	9(Do)			

## Für jede Abgabe:

- 1 Eine Person erstellt Abgabegruppe mit **sicherem** Kennwort
  - Sie sind dafür verantwortlich, dass niemand anderes Zugriff auf Ihre Abgabe hat
- 2 Anderen Gruppenmitgliedern Gruppenname und Kennwort mitteilen
- 3 Andere Mitglieder treten Gruppe bei
- 4 Namen und Matrikelnummern aller Gruppenmitglieder in Abgabe schreiben
- 5 Ein Gruppenmitglied lädt die Abgabe hoch

### Neue Gruppe anlegen

Gruppenname

Kennwort    
Drücken Sie Enter, um Änderungen zu speichern

**Neue Gruppe anlegen** **Abbrechen**

### Gruppe beitreten TestGruppe

Wollen Sie sich wirklich in die Gruppe *TestGruppe* einschreiben?

Kennwort nötig

**Gruppe beitreten TestGruppe** **Abbrechen**

- Wir empfehlen ausdrücklich, in Gruppen abzugeben!
- Abgabegruppen können und müssen individuell pro Abgabe gewählt werden und können bis zur Abgabefrist geändert werden
- Stellen Sie sicher, dass alle Mitglieder der Gruppe beitreten
- **Unser Tipp:** Sobald Sie Gruppenmitglieder gefunden haben, richten Sie bereits Gruppen für alle 6 Abgaben ein, später kann man immer noch wechseln
- Die Abgabefrist (Freitag 23:59) ist strikt, spätere Abgaben werden nicht bewertet
- **Zusammenarbeit außerhalb der Abgabegruppen ist nicht gestattet!**
  - Plagiate verletzen Wissenschaftsethik und Urheberrecht!
  - Wir behalten uns vor, alle involvierten Abgaben mit 0 Punkten zu bewerten
- **Unser Tipp:** Hausaufgaben als fortlaufende Klausurvorbereitung
  - Beste Möglichkeit, eigenständig Aufgaben zu bearbeiten und Korrektur zu erhalten
  - Bleiben Sie bis zur letzten Hausaufgabe dabei!
- **Parallel: Moodle-Quiz nicht vergessen!**

- Forum für Ankündigungen der Veranstalter:innen (automatisch abonniert)
- Spezifische Foren für Fragen zum fachlichen Inhalt und der Organisation
  - Vorlesungen
  - Übungen
  - ...
- Gruppeninternes Forum
- Abgabegruppensuche
- Diskussion und Austausch unter den Student:innen ist erwünscht!
- Foren werden von Tutor:innen + Veranstalter:innen aktiv betreut (Antwortzeit  $\approx 1$  Arbeitstag)

- Interaktiver Moodle Kurs mit Lehreinheiten und Übungsaufgaben (in Englisch):  
<https://moodle.informatik.tu-darmstadt.de/course/view.php?id=757>
  - Entwickelt von Prof. Dr.-Ing. Abdulhadi Shoufan (Khalifa University, UAE)
  - Verweise in Vorlesungen und Übungen auf korrespondierende Einheiten
    - Beispiele: LQ2-1 | RQ4-4 | EX2-2-9: C2F | Quiz-1
- Freiwilliges Zusatzangebot, keine Gewährleistung auf Korrektheit des Inhalts  
⇒ Nur die Vorlesungsfolien und Übungsblätter sind für die Klausur maßgeblich

- Zeitpunkt: 03.03.26 11:00-13:00
  - Raumzuteilung wird rechtzeitig bekannt gegeben
- **Anmeldung in TUCaN bis 31.01.2026 notwendig**
  - Für Fachprüfung (ab 01.11.2025) **und ggf. Studienleistung** (ab 01.09.2025)
- **Kein Notenbonus** nach §25(2) APB
- **Keine Hilfsmittel** (Taschenrechner, etc.) zugelassen
  - Ausnahme: Hilfsblatt (siehe Moodle, wird gestellt) und gedrucktes Wörterbuch
- **Nachteilsausgleich:** Frühstmöglich [hier](#) melden und uns **spätestens bis 17.02.2026** Bescheid geben (je früher desto besser)
- Wiederholungsklausur: Sommersemester 2026
- Wir stellen eine Altklausur zur Verfügung

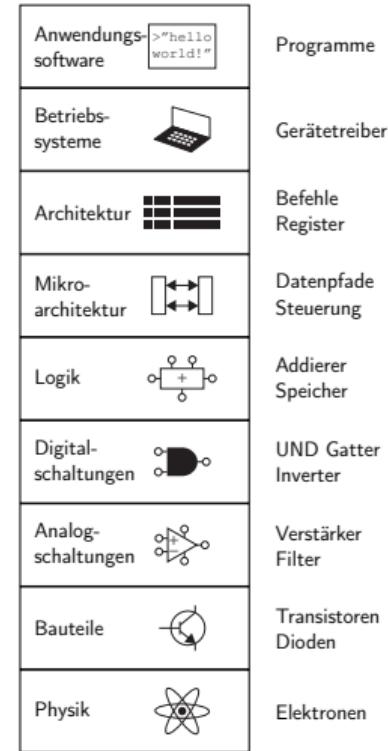
- Minimalprogramm:
  - Studienleistung (= Hausaufgaben & Quizze) & Klausur bestehen
- Dringend empfohlen:
  - Vorlesungen
  - Übungen
  - Hausaufgaben & Quizze auch über Studienleistung heraus
- Lohnenswert:
  - Literatur, insbesondere „Digital Design and Computer Architecture“
- Zusatzangebote (optional):
  - Digital Logic Design Moodle Kurs

# Fragen zur Organisation?

# Agenda



- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Computer Systeme haben die Welt verändert
  - Handys, Internet, Medizintechnik, Unterhaltung, ...
- Die Anforderungen steigen ständig
- Die Erfüllung dieser Anforderungen erfordert ein tiefgreifendes Verständnis von Computern



- "In der Informatik geht es genauso wenig um Computer wie in der Astronomie um Teleskope." (Edsger Dijkstra)
- Wissenschaft der methodischen und *automatischen* Bearbeitung von Informationen
- <https://de.wikipedia.org/wiki/Informatik>

Angewandte Informatik und Realisierungen

Technische Informatik

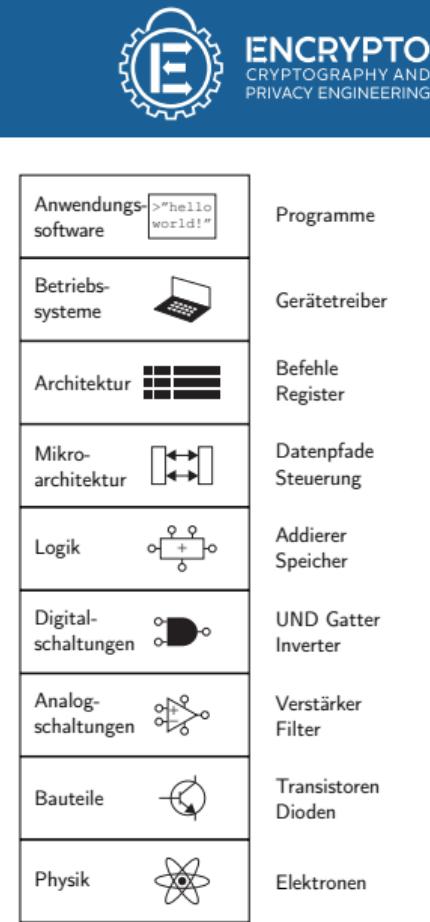
Praktische Informatik

Theoretische Informatik

- Teilgebiet der Informatik, das sich mit Entwurf und Realisierung von Computersystemen befasst
- Teilgebiete:
  - Sensortechnik
  - **Digitaltechnik**
  - Rechnerarchitektur
  - Betriebssysteme
  - Compiler
  - ...
- Digitaltechnik
  - Befasst sich mit dem Entwurf digitaler Schaltungen

# Agenda

- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Beherrschen von Komplexität
- Digitale Abstraktion
- Zahlensysteme
- ...

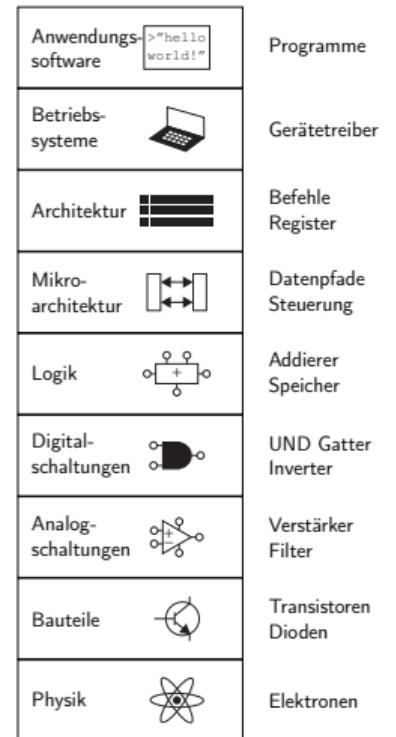


Harris 2016  
Kap. 1.1 - 1.4.4

- Abstraktion
- Disziplin
- Wesentliche Techniken
  - Hierarchie (**Hierarchy**)
  - Modularität (**Modularity**)
  - Regularität (**Regularity**)

- Wichtiges und zentrales Konzept der Informatik
- Verstecken (für eine spezielle Aufgabe) „unnötiger“ Details
- Verstehen der Abstraktionsebenen ist aber für alle Aufgaben hilfreich

# Ein abstrakter Computer? Schichtenmodell eines Computers!



Wesentliche Eigenschaften eines Schichtenmodells sind:

- Untere Schicht erbringt Dienstleistungen für die nächst höhere Schicht
- Obere Schicht nutzt nur die Dienste der nächst niedrigeren Schicht
- Eindeutige Schnittstellen zwischen den Schichten
- Vorteile einer sauberen Schichtenstruktur:
  - Austauschbarkeit einzelner Schichten, ohne benachbarte Schichten oder das gesamte System ändern zu müssen
  - Benutzer:in braucht nur die von ihr:ihm zu bearbeitende Schicht zu kennen
  - Darunterliegende Schichten bilden fest definierte Funktionalität
  - Für manche Aufgaben ist genauere Kenntnis der unteren Schichten dennoch notwendig (z.B. Programmierung von Gerätetreibern)
- Nachteil ist eine ggf. geringere Leistungsfähigkeit des Systems

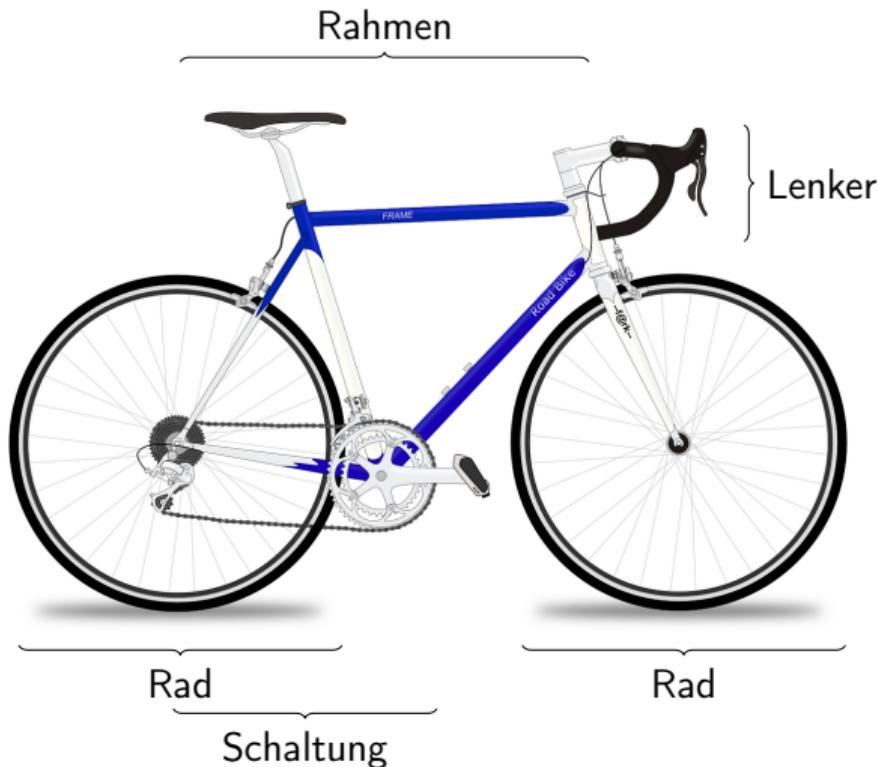
- Disziplin ist die wissenschaftliche Beschränkung der Realisierungsmöglichkeiten
  - Erlaubt produktivere Arbeit auf höheren Entwurfsebenen
- Beispiel: Digitale Entwurfsdisziplin
  - Arbeite mit diskreten statt mit stetigen Spannungspegeln
  - Digitalschaltungen sind einfacher zu entwerfen als analoge  
Folge: Erlaubt den Entwurf komplexerer Schaltungen
  - Digitale Systeme ersetzen zunehmend analoge  
Digitalkamera, digitales Fernsehen, moderne Handys, CD, DVD, ...

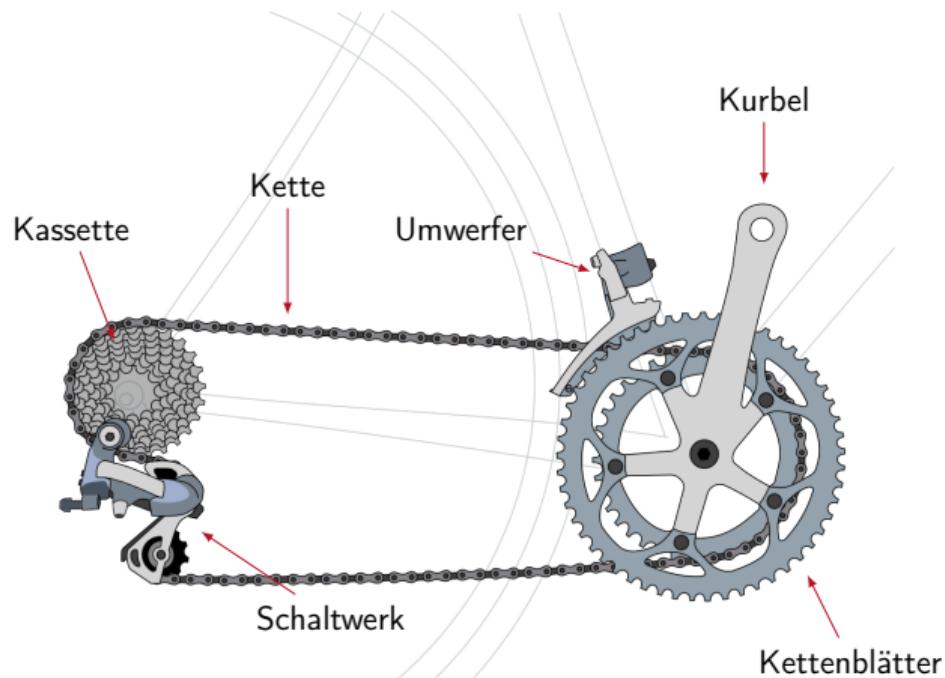
- Die meisten physikalischen Größen haben stetige Werte
  - Elektrische Spannung
  - Frequenz einer Schwingung
  - Position einer Masse
- Berücksichtigen unendlich viele Werte der Größe nicht praktikabel
- Digitale Abstraktion: Berücksichtigt nur endlich viele Werte
  - Untermenge aus einem stetigen Wertebereich

- Hierarchie (Hierarchy)
  - Aufteilen eines Systems in Module und Untermodule
- Modularität (Modularity)
  - wohldefinierte Schnittstellen und Funktionen
- Regularität (Regularity)
  - bevorzuge einheitliche Lösungen für einfachere Wiederverwendbarkeit

- Alltagsbeispiel für Anwendung der drei Y's
- Komplexer Gebrauchsgegenstand
- Entwicklung begann im 19. Jahrhundert
- Verschiedenste historische Modelle und Bauweisen
- Heutige Fahrräder weitestgehend standardisiert durch Normen und Industriestandards







CC-BY-SA 3.0 Keithonearth

- Funktion der Räder
  - Übertragung des Gewichts von Fahrer:in an Boden und von Drehung zu Vorwärtsbewegung
- Funktion der Schaltung
  - Weitergabe von Drehung der Kurbel an Reifen als Getriebe
- Im Idealfall sind Funktionen unabhängig und beeinflussen sich nicht
- Schnittstelle zwischen Hinterrad und Schaltung
  - Gemeinsame Achse

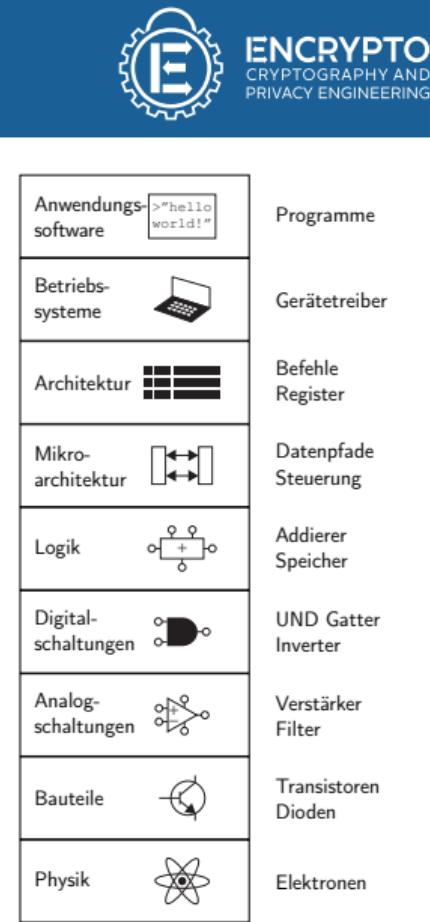


- Verschiedene Schaltungen für gleiche Räder
  - Voraussetzung: Kompatibilität mit Kassette an Achse
- Unterschiedliche Lenker an gleichen Rahmen
- Gleicher Rahmen für unterschiedliche Räder
  - Voraussetzung: Raddurchmesser und -breite passend
- Voraussetzung für Massenproduktion, einfache Reparatur und Modifikation

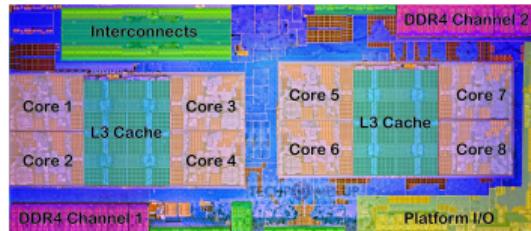
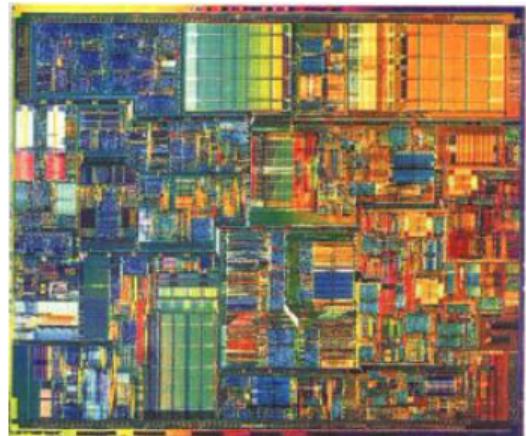


# Agenda

- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Komplexität technischer Systeme steigt ständig (vgl. Moore'sches Gesetz)
  - 2000: Intel Pentium 4:  
 $42 \cdot 10^6$  Transistoren auf  $217 \text{ mm}^2$
  - 2017: AMD Ryzen:  
 $4,8 \cdot 10^9$  Transistoren auf  $192 \text{ mm}^2$
- ⇒ ohne rechnergestützte Hilfsmittel nicht zu beherrschen
- ⇒ Hardwarebeschreibungssprachen zum Beherrschen von Komplexität
  - Hierarchie
  - Modularität
  - Regularität

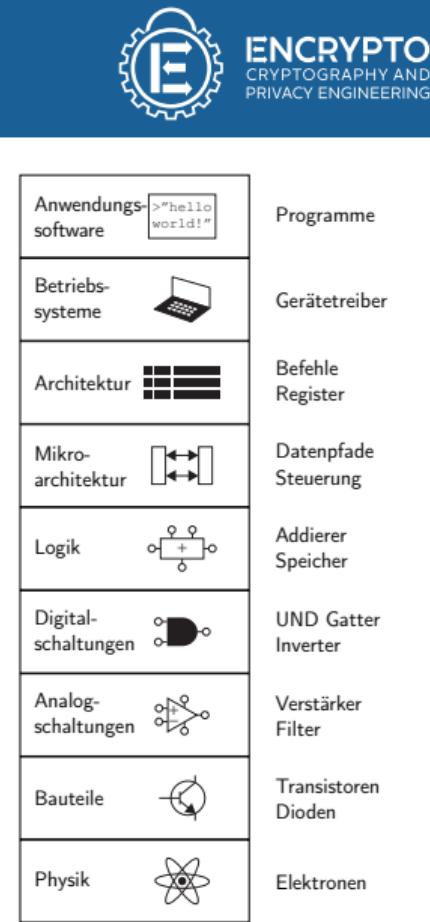


- seit Beginn der Rechnerentwicklung:
  - Suche nach verständlichen und einheitlichen Beschreibungssprachen für
    - Designspezifikation
    - Simulation
    - Verifikation
    - Dokumentation
  - ⇒ nutzt auch der Kommunikation zwischen Entwickler:innen
- zunächst Hochsprachen (bspw. Pascal, LISP, Petri-Netze) zur Hardware-Beschreibung eingesetzt
- 1960/70: Register-Transfersprachen
  - Datentransfer zwischen *Registern* durch kombinatorische Operatoren
  - ⇒ synchrone sequentielle Schaltungen als Abstraktionslevel

- Im Rahmen dieser Veranstaltung:
- SystemVerilog
  - auf Verilog HDL basierende Hardware Description and Verification Language (HDVL)
  - 2002 erschienen
  - Verilog und SystemVerilog sind neben VHDL die am meisten verwendeten HDLs der Branche.
- **Konzepte und Details werden in den weiteren Vorlesungen und Übungen Schritt für Schritt eingeführt.**

# Agenda

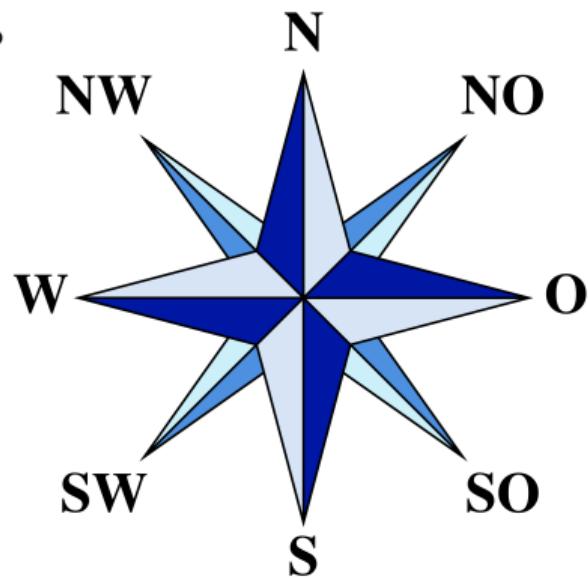
- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Beschränkung auf nur zwei unterschiedliche Werte
- Können unterschiedlich heißen
  - 1, WAHR, TRUE, HIGH, ...
  - 0, FALSCH, FALSE, LOW, ...
- Können unterschiedlich repräsentiert werden
  - elektronisch (Spannungspegel)
  - mechanisch (Zahnradstellungen)
  - hydraulisch (Flüssigkeitsstände)
  - aber auch Quantenzustände, ...

- Bit (Binary digit): Maßeinheit für Information  
(Unterscheidung zwischen zwei Zuständen)
  - ⇒ Antwort auf Entscheidungsfragen (bspw. „Ist die Erde eine Scheibe?“) kann mit einem Bit codiert werden.
  - ⇒ Bit ist kleinstmögliche Informationseinheit
- Warum ist eine solche Kodierung notwendig bzw. sinnvoll?
  - Technische Realisierung über Schwellwerte ist einfacher, bspw.
  - Elektrische Ladungen (0 = ungeladen, 1 geladen)
  - Elektrische Spannungen (0 = 0 Volt, 1 = 5 Volt)
  - Magnetisierung (0 = unmagnetisiert, 1 = magnetisiert)

- Mehr als zwei Zustände / Antwortmöglichkeiten müssen mit mehr Bits repräsentiert werden
- Beispiel: Aus welcher Himmelsrichtung weht der Wind?
  - 0 0 = Norden
  - 0 1 = Osten
  - 1 0 = Süden
  - 1 1 = Westen
- ⇒ 2 bit für vier Zustände
- Wie viele Bits werden benötigt zur Kodierung von {S, SW, W, ..., O, SO}?



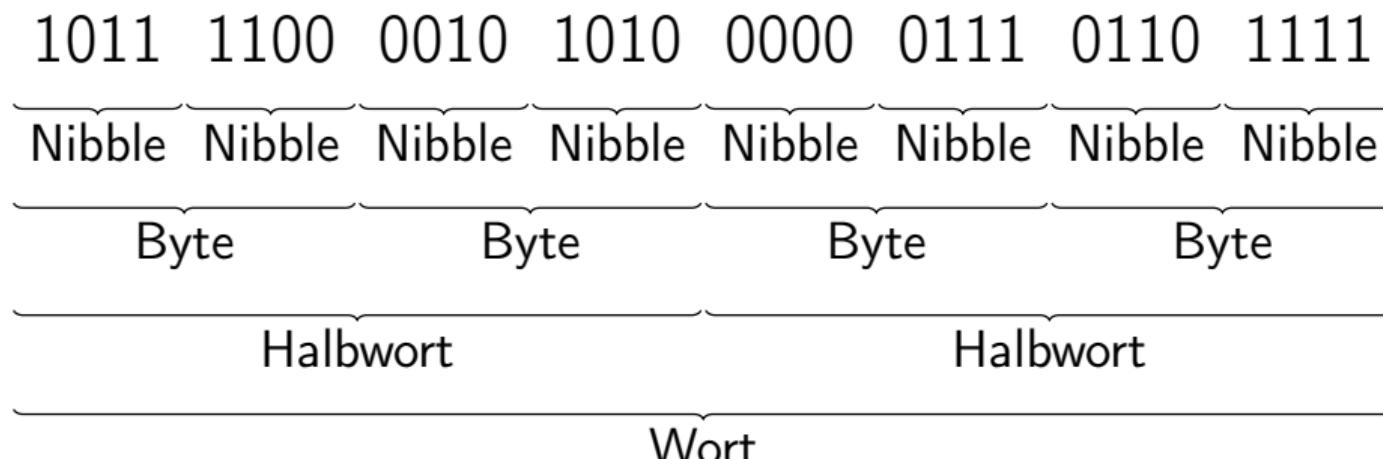
$2^0 =$	1	$2^{10} =$	1024	Kibi ( $\approx$ Tausend)
$2^1 =$	2	$2^{11} =$	2048	
$2^2 =$	4	$2^{12} =$	4096	
$2^3 =$	8	$2^{13} =$	8192	
$2^4 =$	16	$2^{14} =$	16384	
$2^5 =$	32	$2^{15} =$	32768	
$2^6 =$	64	$2^{16} =$	65536	
$2^7 =$	128	$2^{20} =$	1048576	Mebi ( $\approx$ Million)
$2^8 =$	256	$2^{30} =$	1073741824	Gibi ( $\approx$ Milliarde)
$2^9 =$	512	$2^{40} =$	1099511627776	Tebi ( $\approx$ Billion)

$$\begin{aligned}1 \text{ Ki} &= \text{Kibi} = 2^{10} = && 1024 \\1 \text{ Mi} &= \text{Mebi} = 2^{20} = && 1024 \times 1024 \\1 \text{ Gi} &= \text{Gibi} = 2^{30} = && 1024 \times 1024 \times 1024 \\1 \text{ Ti} &= \text{Tebi} = 2^{40} = && 1024 \times 1024 \times 1024 \times 1024\end{aligned}$$

---

$$\begin{aligned}1 \text{ k} &= \text{Kilo} = 10^3 = && 1000 \\1 \text{ M} &= \text{Mega} = 10^6 = && 1000 \times 1000 \\1 \text{ G} &= \text{Giga} = 10^9 = && 1000 \times 1000 \times 1000 \\1 \text{ T} &= \text{Tera} = 10^{12} = && 1000 \times 1000 \times 1000 \times 1000\end{aligned}$$

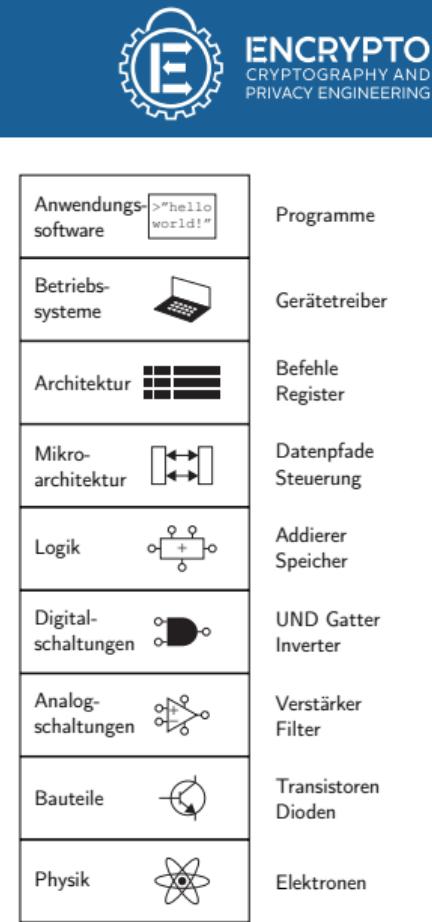
- **Achtung:** Basis oft nicht eindeutig
- bspw. bei Festplatten: Ein GB wird als 1 000 000 000 Byte  $\approx$  0,93 GiByte verkauft.



- Größe eines (Halb-)Worts **hängt vom Kontext ab** (meist Registerbreite)

# Agenda

- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- natürliche Zahlen     $\mathbb{N} = \{0, 1, 2, \dots\}$
- ganze Zahlen         $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
- rationale Zahlen     $\mathbb{Q} = \left\{ \frac{a}{b} : a \in \mathbb{Z} \wedge b \in \mathbb{N} \wedge b \neq 0 \right\}$
- reelle Zahlen         $\mathbb{R}$
- komplexe Zahlen, ...

$$\begin{aligned}\text{dezimal: } 5347 &= 7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000 \\&= 7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3 \\&=: 5347_{10}\end{aligned}$$

$$\begin{aligned}\text{binär: } 1101_2 &= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 \\&= 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 \\&= 13_{10}\end{aligned}$$

	dual/binär	oktal	dezimal	hexadezimal
$b$	2	8	10	16
$Z_b$	$\{0, 1\} := \mathbb{B}$	$\{0, \dots, 7\}$	$\{0, \dots, 9\}$	$\{0, \dots, 9, A, B, C, D, E, F\}$
Literale	$1101\ 0011_2$ $0b11010011$	$323_8$ $0o323$	$211_{10}$ $0d211$	$D3_{16}$ $0xD3$

dezimal:       $5347 = 7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000$   
 $= 7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3$   
 $=: 5347_{10}$

binär:       $1101_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$   
 $= 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8$   
 $= 13_{10}$

hexadezimal:  $1F3A_{16} = 10 \cdot 16^0 + 3 \cdot 16^1 + 15 \cdot 16^2 + 1 \cdot 16^3$   
 $= 10 \cdot 1 + 3 \cdot 16 + 15 \cdot 256 + 1 \cdot 4096$   
 $= 7994_{10}$

### Definition: vorzeichenloses Stellenwertsystem

Für eine Basis  $b \in \mathbb{N} \wedge b \geq 2$  ist  $Z_b := \{0, 1, \dots, b - 1\}$  die Menge der verfügbaren Ziffern. Die Funktion  $u_{b,k}$  bildet eine Ziffernfolge der Breite  $k \in \mathbb{N}$  auf eine natürliche Zahl ab:

$$u_{b,k} : (a_{k-1} \dots a_1 a_0) \in Z_b^k \mapsto \sum_{i=0}^{k-1} a_i \cdot b^i \in \mathbb{N}$$

Trick zur effizienteren Berechnung ohne Exponentiationen  $b^i$ : Horner Schema:

$$\sum_{i=0}^{k-1} a_i \cdot b^i = ((\dots((a_{k-1} \cdot b + a_{k-2}) \cdot b + a_{k-3}) \dots) \cdot b + a_1) \cdot b + a_0$$

- polyadisches Zahlensystem
- niedrigstwertige Stelle (LSD, least significant digit):  $a_0$
- höchstwertige Stelle (MSD, most significant digit):  $a_{k-1}$
- kleinste darstellbare Zahl:  $\sum_{i=0}^{k-1} 0 \cdot b^i = 0$
- größte darstellbare Zahl:  $\sum_{i=0}^{k-1} (b-1) \cdot b^i = b^k - 1$
- Anzahl der darstellbaren Werte:  $|Z_b^k| = |Z_b|^k = b^k$
- eineindeutig (bijektiv) abbildbar auf Wertebereich  $\{0, \dots, b^k - 1\}$  für festes  $k$

## ■ Syntax: <N>'<B><wert>

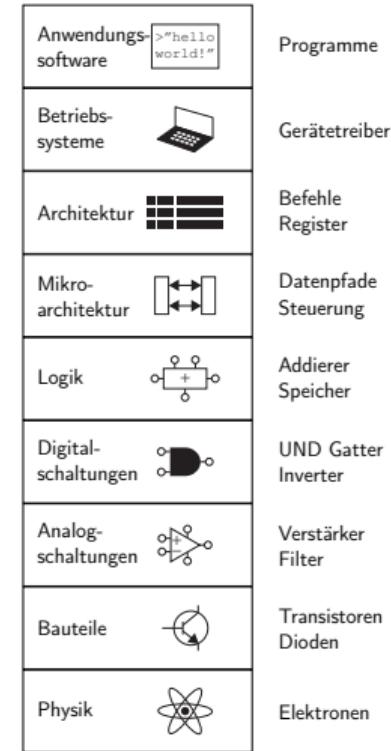
- <N> = Bitbreite (Anzahl der genutzten Bits, d.h. Stellen im Binärsystem)
- <B> = Basis (d,b,o,h)
- beide Angaben optional (default: 32'd)
- Werte werden schlussendlich automatisch umgerechnet und binär dargestellt und mit führenden 0en bis zur Bitbreite aufgefüllt
- Unterstriche als optische Trenner möglich (werden ignoriert)

Literal	Bitbreite	Basis	Dezimal	Binär
3'b101	3	binär	5	101
'b11	32	binär	3	0000...00000011
8'b11	8	binär	3	00000011
8'b1010_1011	8	binär	171	10101011
3'd6	3	dezimal	6	110
6'o42	6	oktal	34	100010
8'hAB	8	hexadezimal	171	10101011
42	32	dezimal	42	0000...0101010

# Agenda



- 1 Organisation
- 2 Einordnung der Digitaltechnik
- 3 Komplexität, Abstraktion und Schichtenmodell
- 4 Hardwarebeschreibungssprachen
- 5 Bits und Bytes
- 6 Darstellung von natürlichen Zahlen
- 7 Zusammenfassung und Ausblick



- Organisation
  - Tragen Sie bis Freitag 17.10.25 12:00 Ihren Gruppenwunsch ein
- Inhalt
  - Einordnung der Digitaltechnik
  - Beherrschung von Komplexität
  - Hardwarebeschreibungssprachen
  - Binärsystem
  - Darstellung von natürlichen Zahlen
- Nächste Vorlesung behandelt
  - Zahlensysteme
  - Logikgatter