

# Rechnerarchitektur

Speicher

Univ.-Prof. Dr.-Ing. Rainer Böhme

Wintersemester 2020/21 · 13. Jänner 2021

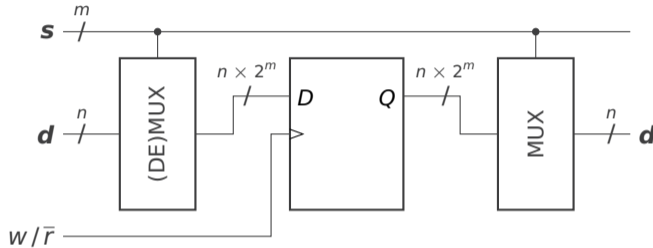
# Erinnerung

## Nicht vergessen !

Melden Sie sich online bis **spätestens 13.01.2021** zum ersten Klausurtermin an.  
Nachmeldungen per E-Mail können wir **nicht** berücksichtigen.

# Ausgangspunkt

Wiederholung der Skizze unseres bisherigen Speichermodells:



## Problemfelder

- Kosten-Nutzen-Verhältnis
- Persistenz

# Technologien zum Speichern von Information

- **Modifikation von Strukturen**

Lochkarte, Schallplatte

- **Magnetismus**

Magnetkernspeicher,  
Magnetband, Diskette,  
Festplatte

- **Elektrische Ladung**

Kondensator, isoliertes Gatter

- **Rückkopplung**

Flipflop, Schwingkreis

- **Optik**

Barkode, CD-ROM, DVD

## Vergleichskriterien

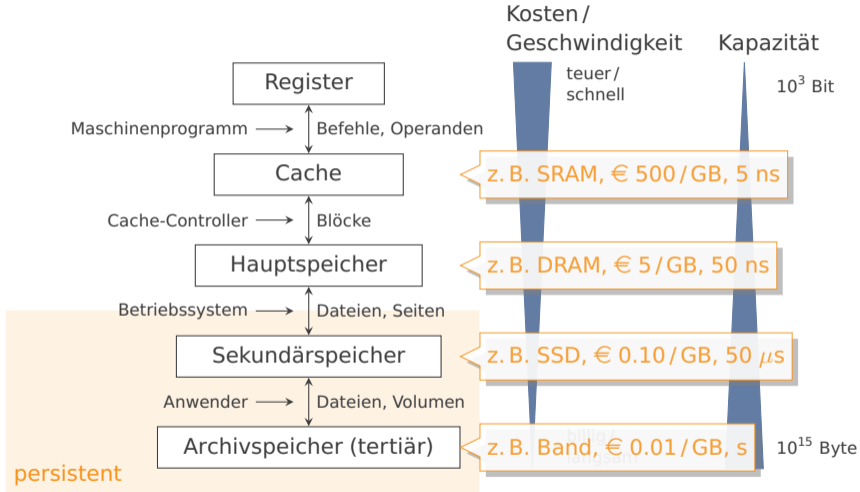
- Persistenz
- Geschwindigkeit  
(Zugriff, Übertragung)
- Kapazität
- Dichte
- Energiebedarf
- Robustheit
- Kosten

Es gibt auch **Kombinationen**, z. B. magneto-optisch.

# Gliederung heute

- 1. Speicherhierarchie**
2. Aufbau und Ansteuerung von dynamischem RAM
3. Aufbau und Ansteuerung von Flash-Speicher
4. Ausblick

# Speicherhierarchie



Quelle für Preise: J. McCallum, <http://www.jcmit.com>, eigene Rundung und Aktualisierung 2020

# Vorsätze für Maßeinheiten

Präfix	Aussprache	Menge dezimal	Menge binär
E	Exa	$(10^3)^6 = 10^{18}$	$(2^{10})^6 = 2^{60}$
P	Peta	$(10^3)^5 = 10^{15}$	$(2^{10})^5 = 2^{50}$
T	Tera	$(10^3)^4 = 10^{12}$	$(2^{10})^4 = 2^{40}$
G	Giga	$(10^3)^3 = 10^9$	$(2^{10})^3 = 2^{30}$
M	Mega	$(10^3)^2 = 10^6$	$(2^{10})^2 = 2^{20}$
k	Kilo	$(10^3)^1 = 10^3 = 1\,000$	$(2^{10})^1 = 2^{10} = 1\,024$
		$(10^3)^0 = 1$	
m	Milli	$(10^3)^{-1} = 10^{-3}$	
$\mu$	Mikro	$(10^3)^{-2} = 10^{-6}$	
n	Nano	$(10^3)^{-3} = 10^{-9}$	
p	Piko	$(10^3)^{-4} = 10^{-12}$	

**Aufgepasst!** Kapazitäten von Speicherbausteinen werden immer noch **binär** angegeben. Bei Speichermedien und in der Datenübertragung ist die **dezimale** Interpretation verbreitet.

# Akronyme für flüchtigen Speicher

- **RAM** (*random access memory*): Speicher mit wahlfreiem Zugriff auf beliebige Adressen (früher im Gegensatz zu Bandspeicher)
- **SRAM** (*static . . .*): statisches RAM, am besten vergleichbar mit D-Flipflops, 4–8 Transistoren pro Bit
- **DRAM** (*dynamic . . .*): dynamisches RAM, das Inhalte nach Auslesen und durch Entladung eines Kondensators im Zeitverlauf vergisst, 1 Transistor pro Bit (→ viel dichter und damit billiger als SRAM)
- **SDRAM** (*synchronous DRAM*): Steuerleitungen wirken bei steigender Flanke des **Speichertakts** (i. d. R. niedriger als CPU-Takt)
- **DDR-SDRAM** (*double data rate . . .*): Leistungssteigerung durch Auslesen mehrerer benachbarter Bits pro Zugriff

Alle **flüchtigen Speicher** verlieren ihren Inhalt ohne Stromversorgung.



# Akronyme für persistenten Speicher

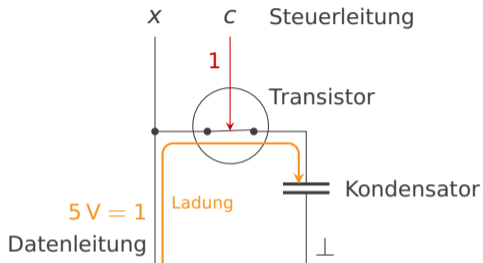
(persistent = nicht flüchtig: Inhalt bleibt ohne Stromversorgung erhalten)

- **ROM** (*read only memory*): nur Lesezugriff
- **PROM** (*programmable ROM*): einmaliger Schreibzugriff (z. B. durch Durchbrennen von „Sicherungen“ (*fuses*) bei der Herstellung)
- **EPROM** (*erasable PROM*): elektrisch programmierbares („brennen“) und durch UV-Licht lösches PROM
- **EEPROM** (*electrically erasable . . .*): Löschen und Wiederbeschreiben geschieht elektrisch  
(→ heute bekannt als Flash-Speicher in SSDs)

# Gliederung heute

1. Speicherhierarchie
2. **Aufbau und Ansteuerung von dynamischem RAM**
3. Aufbau und Ansteuerung von Flash-Speicher
4. Ausblick

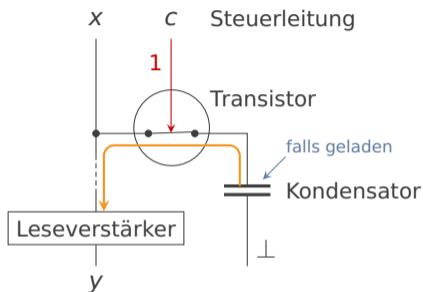
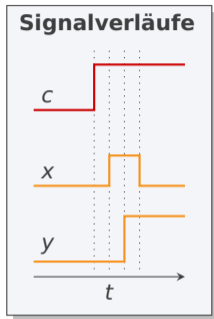
# DRAM-Speicherzelle



## Schreiben einer DRAM-Zelle

1. Setze die Datenleitung  $x$  auf das abzuspeichernde Potenzial.
2. Setze die Steuerleitung  $c = 1$ , sodass der Transistor leitet.
3. Nach Abschluss der (Ent-)Ladung kann  $c = 0$  gesetzt werden.

# DRAM-Speicherzelle



## Auslesen einer DRAM-Zelle

1. Schalte die Datenleitung an den Eingang eines Leseverstärkers.
2. Setze die Steuerleitung  $c = 1$ , sodass der Transistor leitet.
3. Schalte ein stabiles Flipflop, falls ein Impuls gemessen wird.

# DRAM-Auffrischung

**Der Kondensator jeder DRAM-Zelle entlädt sich beim Auslesen und durch Leckströme von selbst.**

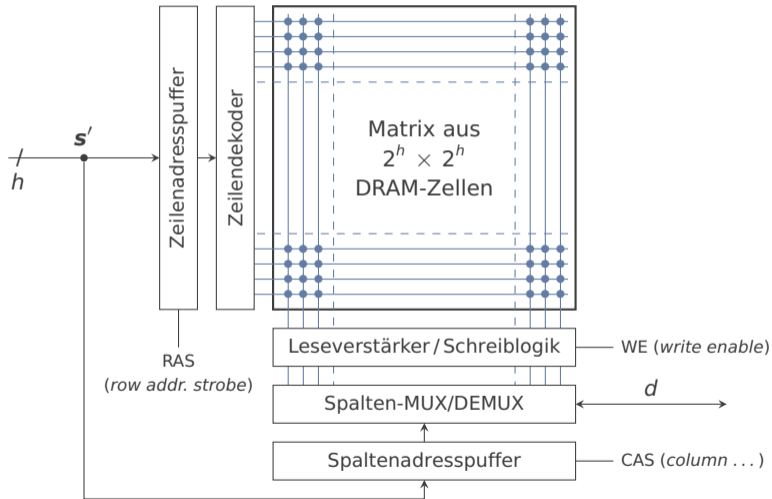
**Abhilfe:** Nach dem Auslesen und zyklisch ca. alle 30–60 Millisekunden wird der aktuelle Wert erneut geschrieben. Diese Aufgabe übernimmt i. d. R. die Ansteuerungslogik oder ein Speicher-Controller.

DRAM verbraucht deshalb im Betrieb mehr Energie (und erzeugt mehr Wärme) als SRAM.

**Trend:** Leckströme und Energieverbrauch mit Spannung senken:

- 2.5 V bei DDR-SDRAM (2000)
- 1.8 V bei DDR2-SDRAM (2004)
- 1.5 V bei DDR3-SDRAM (2007)
- 1.2 V bei DDR4-SDRAM (2012)
- 1.1 V bei DDR5-SDRAM (2020)

# Aufbau von DRAM-Bausteinen



# Ansteuerung beim Lesezugriff

1. WE deaktivieren.
2. Anlegen der **Zeilenadresse** und Übernahmen in den Zeilenadresspuffer durch Aktivierung des Steuersignals RAS.  
→ Bitvektor der aktivierten Zeile liegt am Leseverstärker an.
3. Anlegen der **Spaltenadresse** und Übernahmen in den Spaltenadresspuffer durch Aktivierung des Steuersignals CAS.  
→ Der Spaltenmultiplexer gibt das gewählte Bit aus.

## Beschleunigung bei Mehrfachzugriff

- Neue Spaltenadresse anlegen (Fast Page Mode, FPM), ca.  $2\times$
- Spaltenadresse automatisch hochzählen (Burst-Modus), ca.  $16\times$

4. WE aktivieren, um Bitvektor zurückzuschreiben.

# Ansteuerung beim Schreibzugriff

1. Anlegen der **Zeilenadresse** und Übernahmen in den Zeilenadresspuffer durch Aktivierung des Steuersignals RAS.  
→ Bitvektor der aktivierten Zeile liegt am Leseverstärker an.
2. Anlegen der **Spaltenadresse** und Übernahmen in den Spaltenadresspuffer durch Aktivierung des Steuersignals CAS und WE aktivieren.  
→ Der Spaltendemultiplexer fügt Bit  $d$  in die Zeile ein.  
→ Die Zeile wird zurückgeschrieben.



# Erweiterungen zur Erhöhung der Zuverlässigkeit

## Behandlung von Fertigungs- und Umgebungseinflüssen

- **Kodierung** zur Fehlererkenntnis und Fehlerkorrektur
- z. B. zusätzliche Paritätsbits (binäre Quersumme), Kreuzsicherung, Schreiben unter Berücksichtigung von defekten Zellen, ...

Vertiefung in **Rechnernetze und Internettechnik**, Pflichtmodul, 3. Semester

## Behandlung von Fehlern der Ansteuerung (insb. Programmierung)

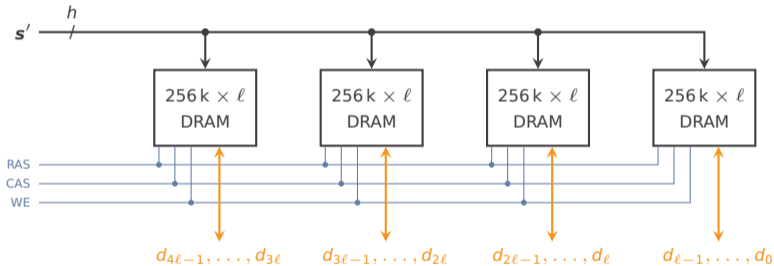
- **Speicherschutz**: CPU prüft Adressbus vor Zugriff.
- Ausnahmebehandlung bei Verletzung
- Berechtigungen werden im privilegierten Modus (d. h. vom Betriebssystem) vergeben.

Vertiefung in **Betriebssysteme**, Pflichtmodul, 2. Semester

# Hauptspeicher aus mehreren DRAM-Bausteinen

Beispiel:  $h = 9 \Rightarrow 2^{18} = 256 \text{ k} \times \ell$  Bit pro Baustein

## Variante 1: Erweiterung der Wortbreite

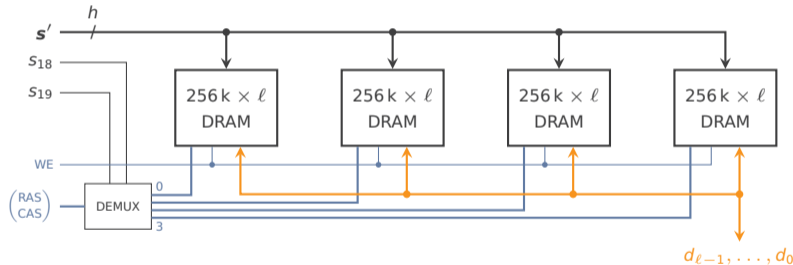


Gemeinsame Adress- und Steuerleitungen; getrennte Datenleitungen

# Hauptspeicher aus mehreren DRAM-Bausteinen

Beispiel:  $h = 9 \Rightarrow 2^{18} = 256 \text{ k} \times \ell$  Bit pro Baustein

## Variante 2: Erweiterung des Adressraums



Gemeinsame Datenleitungen; Auswahl des Bausteins durch Schaltung der Steuersignale über die höchstwertigen Adressleitungen

# Hörsaalfragen



24 82 94 16

Es sei  $\ell = 8$  Bit.

1. In welchem DRAM-Baustein wird das vierte Bit der Adresse  $0x020000$  bei **Variante 1** (Erweiterung der Wortbreite) gespeichert?
2. In welchem DRAM-Baustein wird das vierte Bit der Adresse  $0x020000$  bei **Variante 2** (Erweiterung des Adressraums) gespeichert?
  - a. erster Baustein (von links)
  - b. zweiter Baustein
  - c. dritter Baustein
  - d. vierter Baustein (ganz rechts)

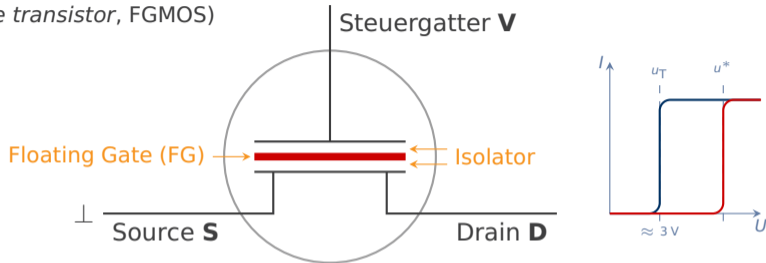
Zugang: <https://arsnova.uibk.ac.at> mit Zugangsschlüssel **24 82 94 16**. Oder scannen Sie den QR-Kode.

# Gliederung heute

1. Speicherhierarchie
2. Aufbau und Ansteuerung von dynamischem RAM
3. **Aufbau und Ansteuerung von Flash-Speicher**
4. Ausblick

# Feldeffekttransistor mit isoliertem Gatter

(engl. *floating gate transistor*, FGMOS)



- Im Floating Gate können Ladungsträger durch quantenmechanische Tunneleffekte „eingesperrt“ werden. Dazu sind Spannungen  $\approx 10\text{ V}$  nötig.
- Wenn das FG nicht geladen ist, reagiert der FGMOS wie ein Transistor: Er leitet, wenn die Spannung am Steuergatter größer als  $u_T$  ist.
- Ein geladenes FG schirmt das Steuergatter ab und verschiebt den Arbeitspunkt des Transistors auf höhere Spannungen  $u^*$ .
- Im Zustand des FG lässt sich (mindestens) ein Bit speichern.

# Ansteuerung eines FGMOS

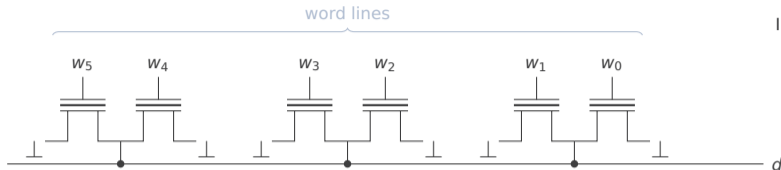
## Zustands(übergangs)tabelle

Steuergatter <b>V</b>	Drain <b>D</b>	FG-Ladung	Transistor	Zustand
<b>Lesezugriff</b>				
$u_T < 3.3\text{ V}$	$> \perp$	ungeladen	leitet	logisch 1
$u_T < 3.3\text{ V} < u^*$	$> \perp$	negativ geladen	sperrt	logisch 0
$> u^*$	$> \perp$	negativ geladen	leitet	logisch 0
<b>Schreib- und Löschezugriff</b>				
$> 10\text{ V}$	$\perp$	steigt		$\rightarrow 0$
$\perp$	$> 10\text{ V}$	fällt		$\rightarrow 1$

# Aufbau von Flash-Bausteinen

## NOR-Flash

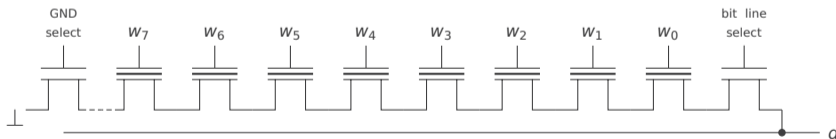
Intel seit 1988



Die Datenleitung  $d$  geht auf 0, wenn an mindestens einer  $word\ line\ w_i$  eine Spannung über dem Arbeitspunkt des Feldeffekttransistoren anliegt.

## NAND-Flash

Toshiba seit 1989



Die  $bit\ line$  geht auf 0, wenn an allen  $word\ lines\ w_i$  eine Spannung über den jeweiligen Arbeitspunkten der Feldeffekttransistoren anliegt.



# Vergleich von Flash-Technologien

## NOR-Flash

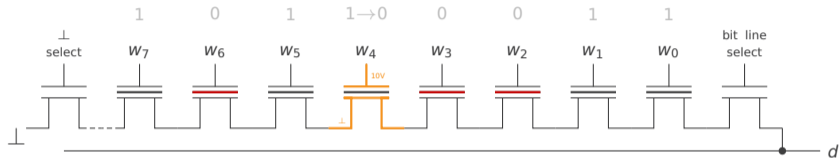
- + adressierbar wie RAM
- + wahlfreies Lesen und Schreiben
- + i. d. R. fehlerfrei
- Schreiben nur langsam
- hoher Energieverbrauch
- geringe Speicherdichte
- teuer (pro Bit)
  
- geeignet für Programmcode

## NAND-Flash

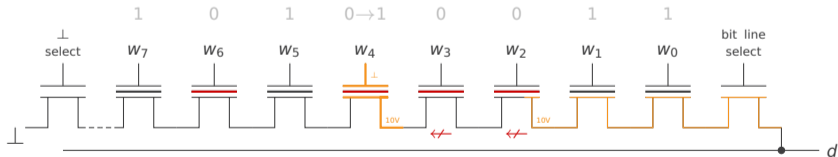
- + billig (pro Bit)
- + hohe Speicherdichte
- + schnell
- + wahlfreies Lesen
- Löschen nur blockweise
- komplizierte Ansteuerung
- Produktionsfehler nicht vernachlässigbar
- Abnutzung
  
- geeignet für Dateisysteme

# Schreiben und Löschen bei NAND-Flash

## NAND-Flash schreiben



## NAND-Flash löschen



# Flash Translation Layer (FTL)

**Verbergen** der **Nachteile von NAND-Flash** in einer Steuerschicht:

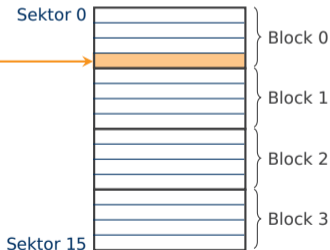
1. Übersetzt logische (L\*N) in physische (P\*N) Adressen
2. Gleichmäßige Verteilung der Löschoptionen (*wear leveling*)
3. Ausschluss fehlerhafter Blöcke (ggf. Fehlerkorrektur)

## Variante 1: Sektor-Mapping

„Schreibe nach LSN 9.“



LSN	PSN
⋮	⋮
6	13
7	12
8	4
9	3
10	2
⋮	⋮



**Neue Anforderung:** Konsistenz der Verwaltungsdaten bei Stromausfall

# Flash Translation Layer (FTL)

**Verbergen** der **Nachteile von NAND-Flash** in einer Steuerschicht:

1. Übersetzt logische (L\*N) in physische (P\*N) Adressen
2. Gleichmäßige Verteilung der Löschoptionen (*wear leveling*)
3. Ausschluss fehlerhafter Blöcke (ggf. Fehlerkorrektur)

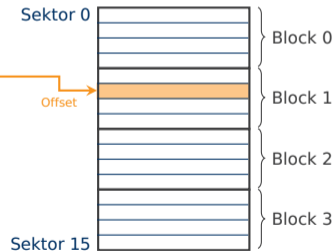
## Variante 2: Block-Mapping

„Schreibe nach LSN 9.“

$$\text{LSN} = 4 \times \text{LBN} + \text{Offset}$$

$$\Rightarrow \text{LBN} = 2, \text{Offset} = 1$$

LBN	PBN
0	2
1	3
2	1
3	0



**Neue Anforderung:** Konsistenz der Verwaltungsdaten bei Stromausfall

# Flash Translation Layer (FTL)

**Verbergen** der **Nachteile von NAND-Flash** in einer Steuerschicht:

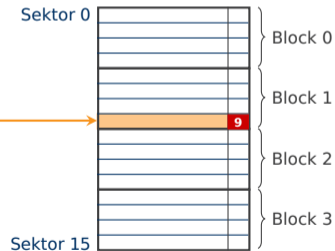
1. Übersetzt logische (L\*N) in physische (P\*N) Adressen
2. Gleichmäßige Verteilung der Löschoptionen (*wear leveling*)
3. Ausschluss fehlerhafter Blöcke (ggf. Fehlerkorrektur)

## Variante 3: Hybrides Mapping

„Schreibe nach LSN 9.“

$$\text{LBN} = \lfloor \text{LSN} : 4 \rfloor = 2$$

LBN	PBN
0	2
1	3
2	1
3	0



**Neue Anforderung:** Konsistenz der Verwaltungsdaten bei Stromausfall

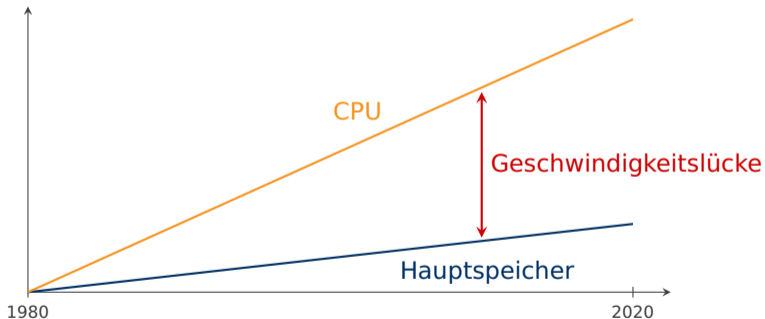
# Gliederung heute

1. Speicherhierarchie
2. Aufbau und Ansteuerung von dynamischem RAM
3. Aufbau und Ansteuerung von Flash-Speicher
4. **Ausblick**

# Betrachtung der Systemleistung

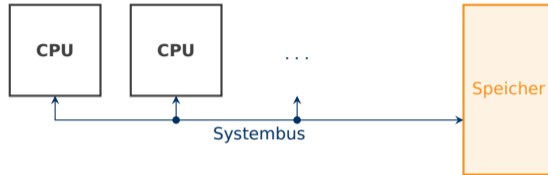
Seit 1980 wächst die Geschwindigkeit des ...

- Hauptspeichers um 7 % pro Jahr
- Prozessors um 50 % pro Jahr

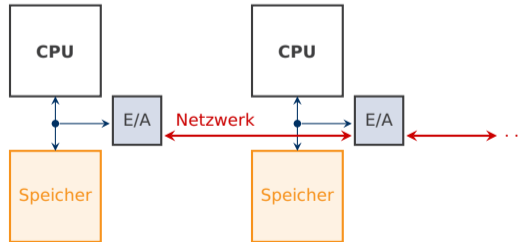


# Speicherarchitekturen für Mehrprozessorsysteme

## Shared Memory („gemeinsam genutzt“)



## Distributed Memory („verteilt“)





# Syllabus – Wintersemester 2020/21

07.10.20	1. Einführung
14.10.20	2. Kombinatorische Logik I
21.10.20	3. Kombinatorische Logik II
28.10.20	4. Sequenzielle Logik I
04.11.20	5. Sequenzielle Logik II
11.11.20	6. Arithmetik I
18.11.20	7. Arithmetik II
25.11.20	8. Befehlssatzarchitektur (ARM) I
02.12.20	9. Befehlssatzarchitektur (ARM) II
09.12.20	10. Prozessorarchitekturen
16.12.20	11. Ein-/Ausgabe
13.01.21	12. Speicher
20.01.21	13. Leistung
<b>27.01.21</b>	<b>Klausur (1. Termin)</b>