

**Prof. Jürgen Becker**

becker@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

**Digitaltechnik**

**Funktionseinheiten  
der Digitaltechnik**

**Zentrale Aufgabe** von **digitalen Systemen**  
ist die **Speicherung** von **Daten**

- Dafür werden **unterschiedliche physikalische Effekte** ausgenutzt:
  - **Rückkopplungsstrukturen** (FlipFlops)
  - elektrische **Ladungen** auf **Kondensatoren**
  - **magnetische Effekte**
- **Es lässt sich folgende Reihenfolge angeben:**
  - **Statische Speicher:** **kleine, schnelle Speicher** zur Speicherung von Zwischenergebnissen meist auf **Basis** von **FlipFlops**
  - **Dynamische Speicher:** **größere, mäßig schnelle Speicher** für größere Datenmengen auf **Basis** von **Ladungsspeichereffekten**
  - **Massenspeicher:** **große, langsame Speicher** für große Datenmengen und zur Archivierung auf der **Basis magnetischer Effekte**
- Im folgenden sollen nur die beiden ersten Speichertypen betrachtet werden

Anhand einiger wichtiger hauptsächlich **funktionaler Merkmale** lassen sich die **verschiedenen Speichertypen klassifizieren**:

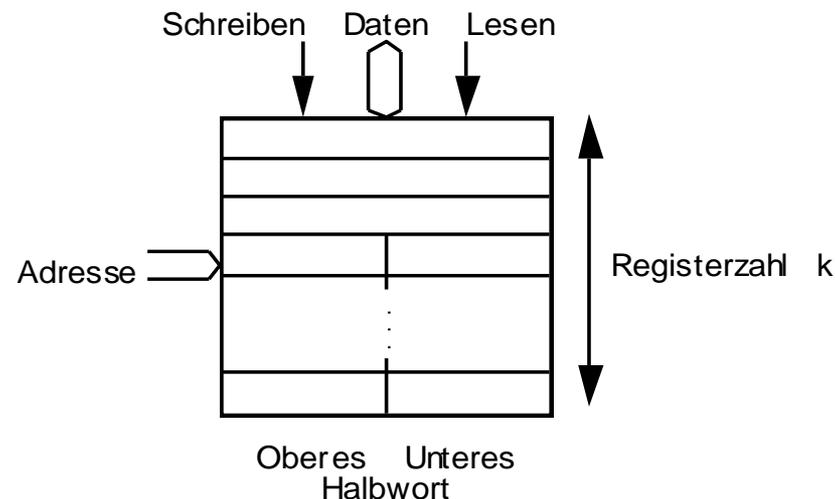
- **Organisation der abgelegten Daten**: im allgemeinen gruppiert man
  - mehrere **Bits** zu einem **Wort** und
  - mehrere **Worte** wiederum zu **Blöcken**
  
- **Art der Zugriffsmethode**:
  - **Random Access Memory (RAM)**
  - **Sequential Access Memory (SAM)**
  
- **Art der Zugriffstechnologie**:
  - **Read/Write Memory (SRAM, DRAM)**
  - **Read Only Memory (ROM)**
  - **Write Once, Read Many (WORM)**
  - **Programmable Read Only Memory (PROM)**
  - **Electrical Writable PROM (EPROM)**
  - **Electrical Writable and Erasable PROM (EEPROM)**

- **Art der Datenspeicherung:** (Grad der Flüchtigkeit)
  - Ladungsspeicher (benötigen Refreshs)
  - Rückkopplungsspeicher (benötigen stetige Energiezufuhr)
  - Speicher mit fixierten Ladungsträgern (Speicherung über längere Zeit ohne stetige Energiezufuhr möglich)
- **Verwendete Schaltungstechnik:**
  - Unterteilung nach schaltungstechnischer Realisierung
- **Verwendete Technologie:** bestimmt maßgeblich
  - Größe,
  - Arbeitsgeschwindigkeit,
  - Kosten,
  - Zuverlässigkeit,
  - usw.

des **Speichers**

# Registerspeicher, Registersätze

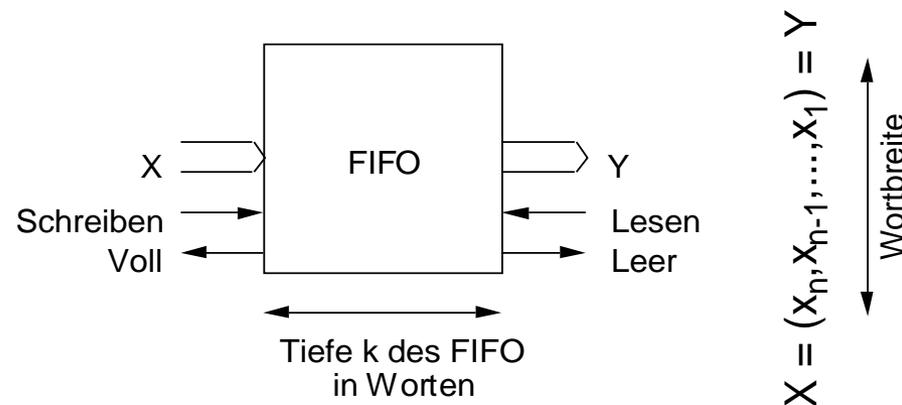
- **Register** werden eingesetzt zur **Speicherung** von Operanden, Konstanten usw.
- **Organisation** meist in Form **gleichlanger Speicherworte**, mit der Möglichkeit zur Verwendung von Halbworten
- einzelne **Register** sind **datentechnisch nicht** miteinander **gekoppelt**
- die Auswahl eines **speziellen Registers** erfolgt über **Selektionsleitungen** (*Adressleitungen*) wie sie von Multiplexern her bekannt sind



- im folgenden werden zwei weitere Speicherstrukturen behandelt, die auf Eigenschaften von Registern aufbauen

# Pufferspeicher (*First-In-First-Out, FIFO*)

- meist **wortorganisierte Speicher** mit konstanter Wortbreite
- dienen häufig zur **Kopplung** von **unterschiedlich schneller Baugruppen** (bzgl. Takt, Verarbeitungsgeschwindigkeit)



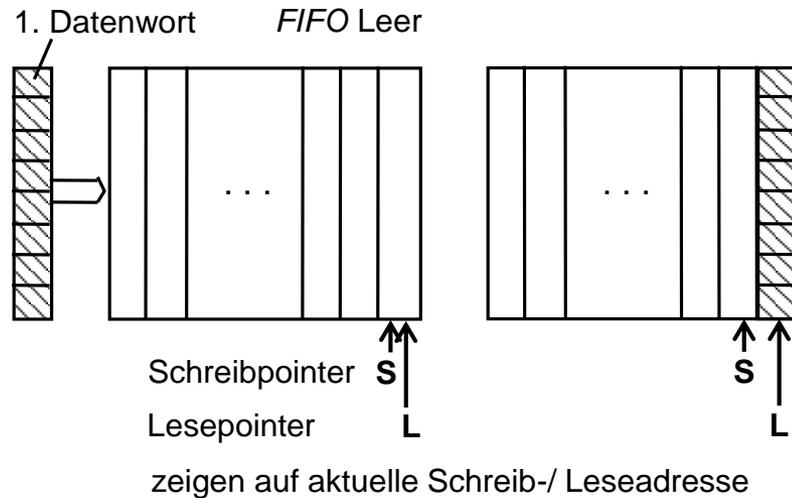
## Funktionsprinzip:

- auf Eingangsseite eingeschriebenes Wort wird solange zum Ausgang vorgeschoben, bis dieser erreicht wird oder ein belegter Platz erreicht wird
- das **Signal "voll"** zeigt an, ob **alle Plätze** im **FIFO belegt** sind und weiteres *Schreiben* möglich ist
- das **Signal "leer"** zeigt an, ob das **FIFO leer** ist
- wird Wort ausgelesen  $\rightarrow$  alle dahinterliegenden Worte rücken um eine Position vor

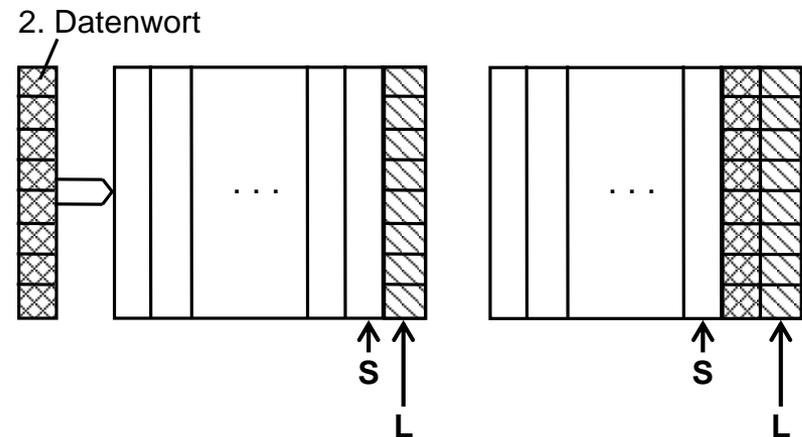
# Pufferspeicher (*First-In-First-Out, FIFO*)

## Beispiel: *Schreiben* und *Lesen* in einem *FIFO*-Speicher

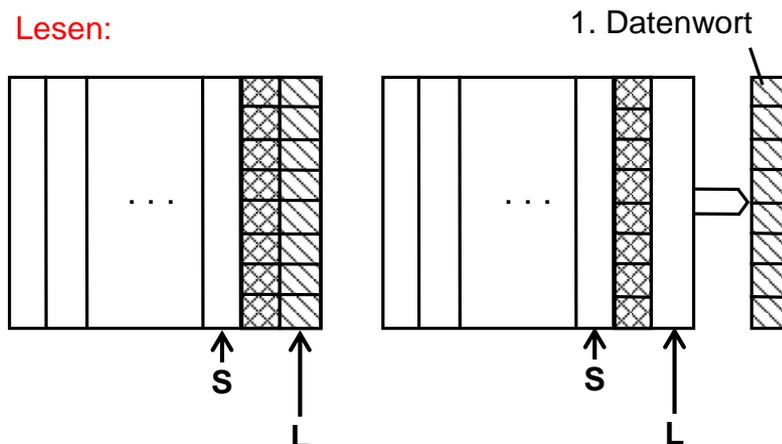
Schreiben



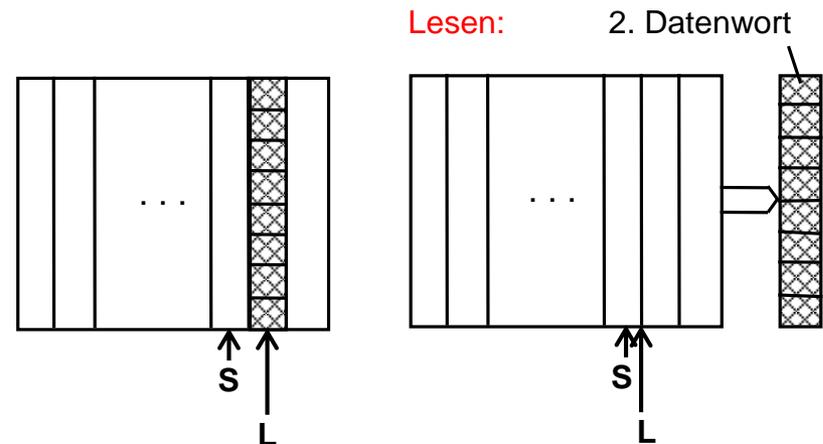
Schreiben



Lesen:

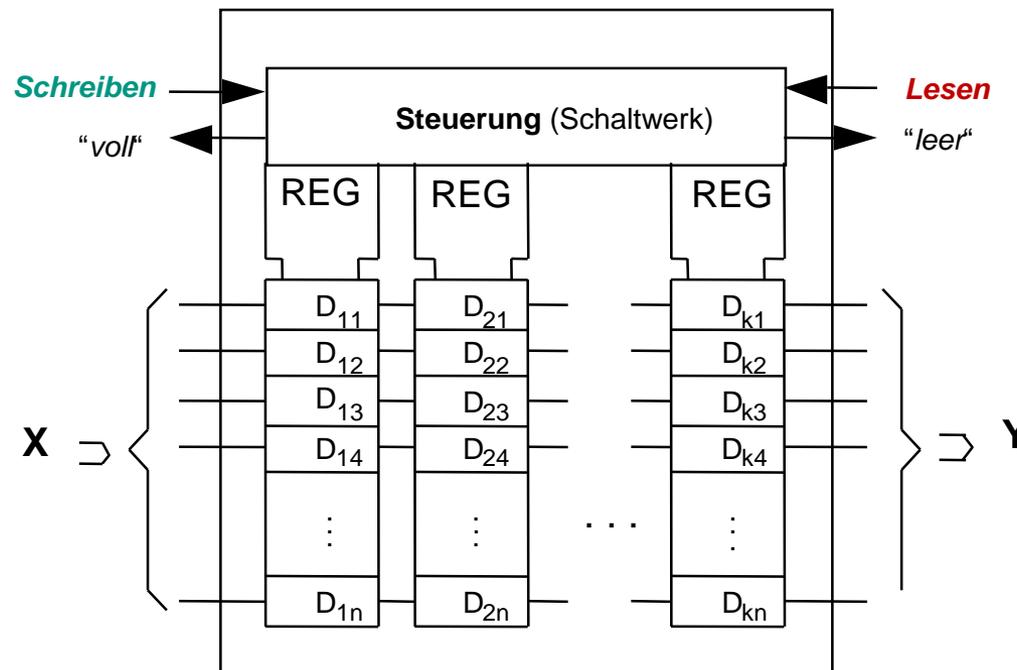


Lesen:



# Pufferspeicher (*First-In-First-Out, FIFO*)

- **FIFO-Realisierung** kann auf verschiedene Weise geschehen
- naheliegend ist die Verwendung von **Schieberegistern entsprechender Größe** und **Verschaltung**:



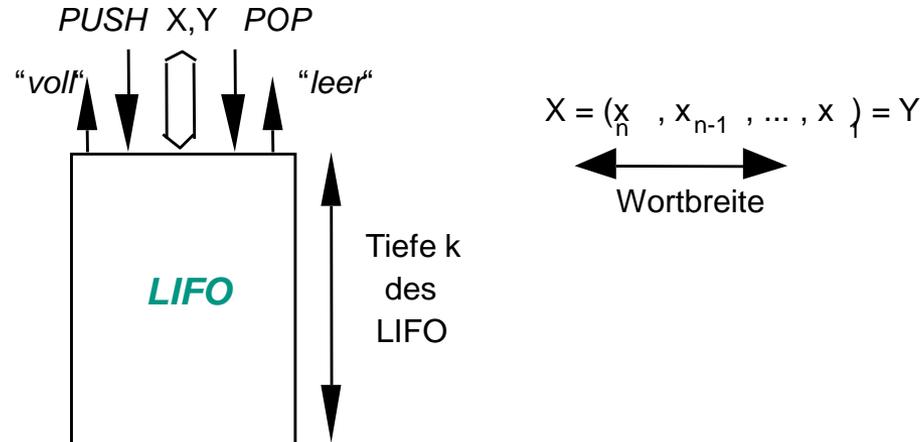
# Stapelspeicher (*Last-In-First-Out, LIFO*)

## Funktionsweise:

- das **zuletzt eingespeicherte Datenwort** (*PUSH*) wird **zuerst** auf dem *LIFO* **ausgelesen** (*POP*)

## Anschaulicher Vergleich: ein Tellerstapel

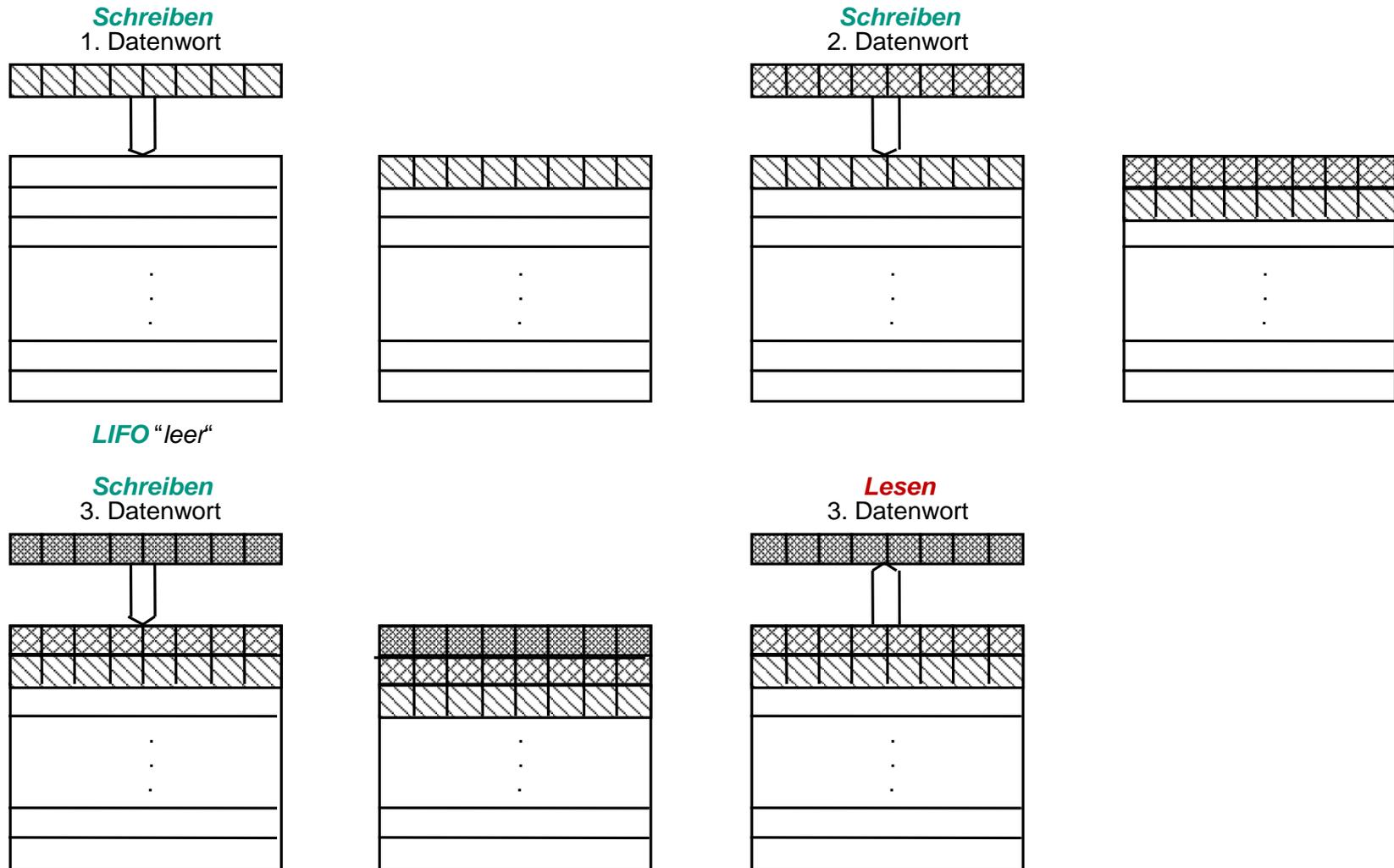
- eignet sich zu **Berechnungen** mit **häufigem zwischenspeichern** von **Ergebnissen** und gleichzeitiger Einhaltung einer **Ordnung** der **Ergebnisse**



- auch beim *LIFO* geben **spezielle Signale** den **Status** des **Speichers** an

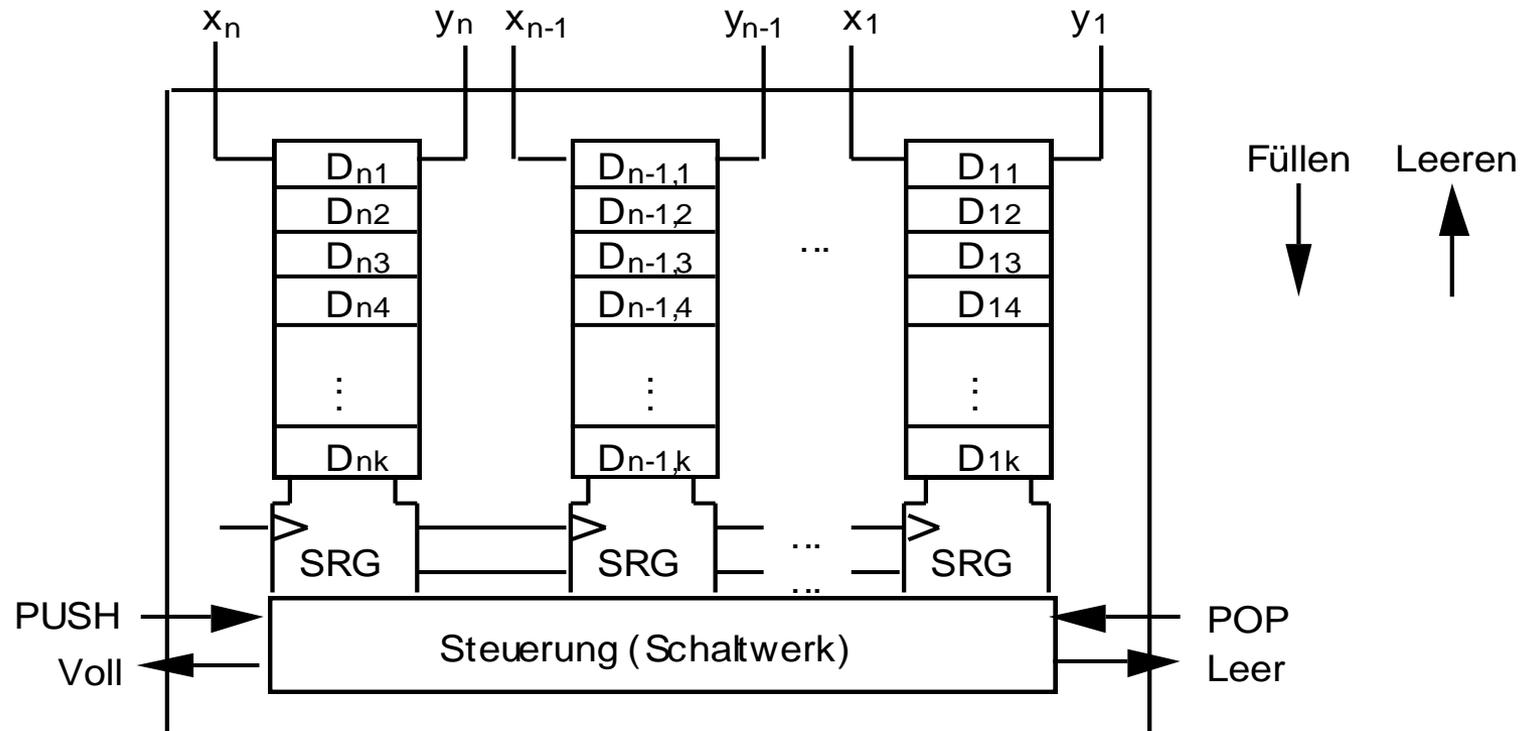
# Stapelspeicher (*Last-In-First-Out, LIFO*)

**Beispiel:** *LIFO* mit **Worten** zu je **8 Bit**



# Stapelspeicher (*Last-In-First-Out, LIFO*)

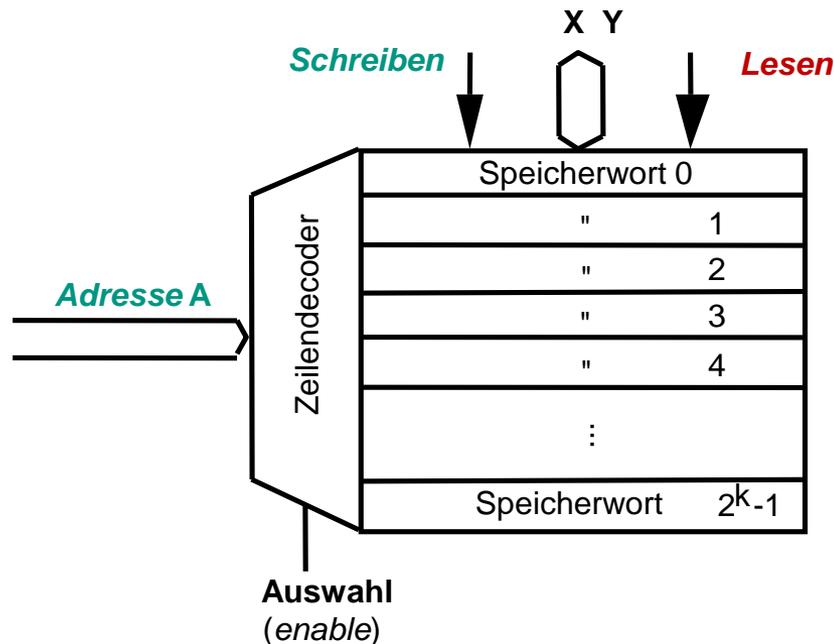
auch **LIFOs** lassen sich auf der **Basis** von **Schieberegistern realisieren**:



# Allgemeine Speicher, Arbeitsspeicher

- **matrixförmig organisierte Anordnungen** von **Speicherzellen** des *dynamischen* oder *statischen* Typs
- sind im allgemeinen *wortorganisiert*
- der **Schreib-/Lese-Vorgang** geschieht durch Angabe von **Adressen**, die über einen **Decoder** zu einer **Leitungsselektion** führt

## Schema eines Speichers:



$$X = (x_n, x_{n-1}, \dots, x_1) = Y$$

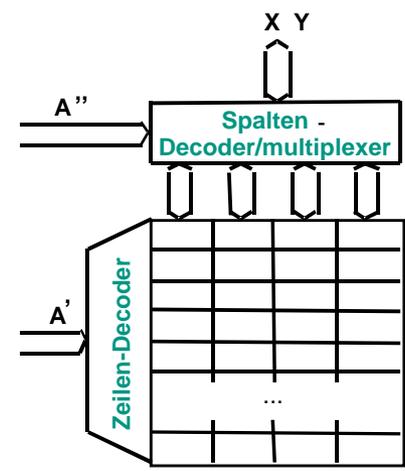
$$A = (a_{k-1}, a_{k-2}, \dots, a_0)$$

# Allgemeine Speicher, Arbeitsspeicher

- unter einer **Kapazität  $K$**  oder der **Speichergröße** eines **Speichers** wird das Produkt  **$K = n \cdot 2^k$**  verstanden
- Aus **schaltungstechnischen** oder **technologischen Gründen** wird meist eine Form der **Speichermatrix angestrebt**, welche dem **Quadrat** möglichst nahe kommt
  - dadurch meist **mehrere Datenwörter** in einem **Speicherwort** enthalten
  - es bedarf daher eines **zweiten Decoders** (Spalten-Decoder), um das entsprechende Datenwort im Speicherwort zu identifizieren
  - sei  **$N$**  die **Länge** des **Speicherworts** und  **$n$**  die **Länge** des **Datenworts** erhält man:  **$r = \lceil \lg R \rceil$** , mit  **$R = N / n$** , die **Anzahl** der **Adressbits** des **Spalten-Decoders**
    - die Adressbits des Zeilendekoders können somit um  **$r$**  Bits gekürzt werden

$$A'' = (a_{r-1}, \dots, a_0)$$

$$A' = (a_{k-1}, \dots, a_r)$$



Speicherkapazität

$$K = n \cdot 2^r \cdot 2^{(k-r)}$$

$$K = n \cdot 2^k$$

# Allgemeine Speicher, Arbeitsspeicher

## Beispiel:

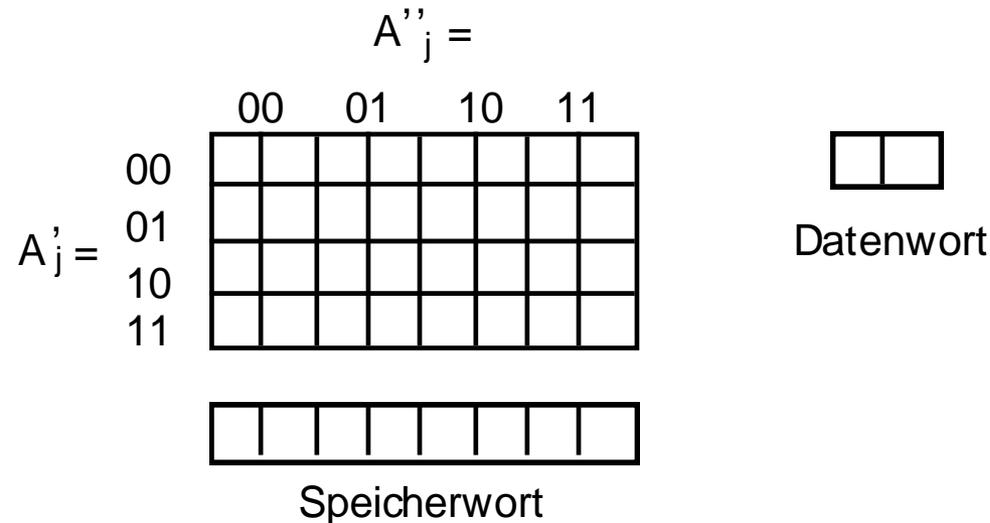
$$n = 2 \quad N = 8 \quad k = 4 \quad K = 2 \cdot 2^4 = 32 \text{ Bit}$$

$$R = 8 / 2 \quad r = \lceil \text{ld } R \rceil = \lceil \text{ld } 4 \rceil = 2$$

$$A' = (a_3, a_2)$$

$$A'' = (a_1, a_0)$$

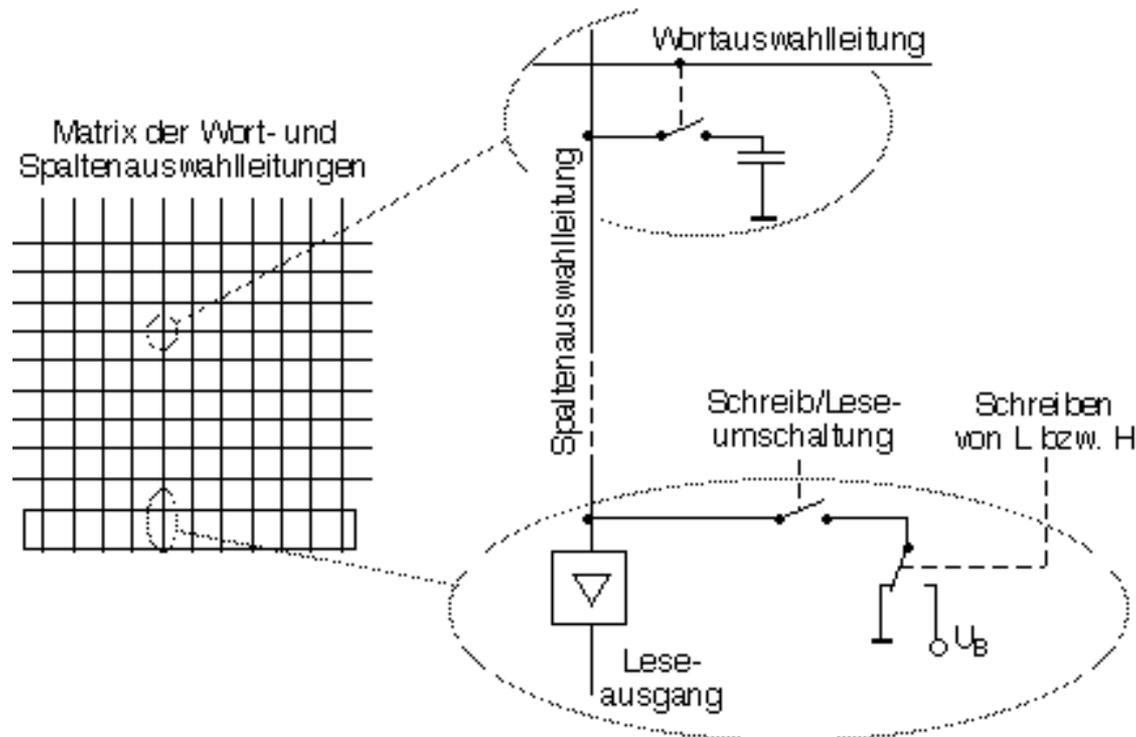
$$K = 32 \text{ Bit}$$



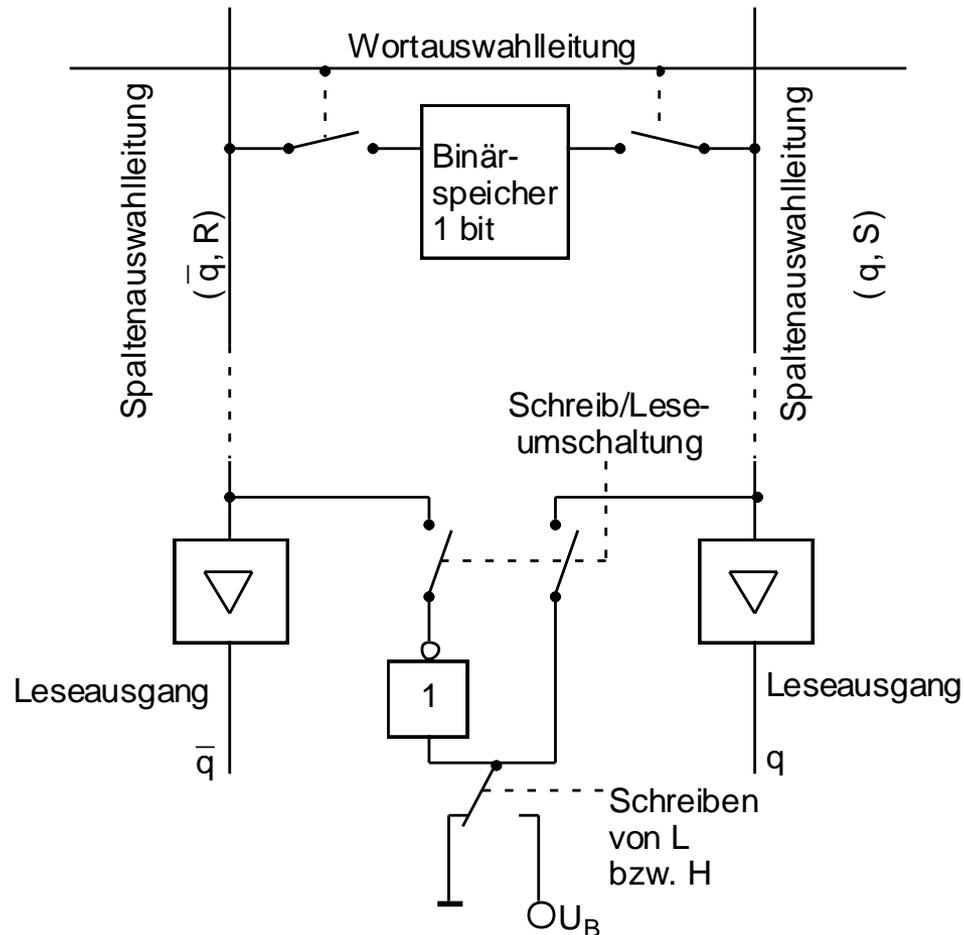
# Allgemeine Speicher, Arbeitsspeicher

- der **innere Aufbau** von **Speichern** ist meist sehr stark an **Technologien angelehnt** und soll in diesem Rahmen nicht detailliert erläutert werden
- **Grundsätzliche Konzepte** können jedoch mit Hilfe von **logischen Komponenten** vermittelt werden

## Prinzip eines Speichers mit dynamischer Zelle:



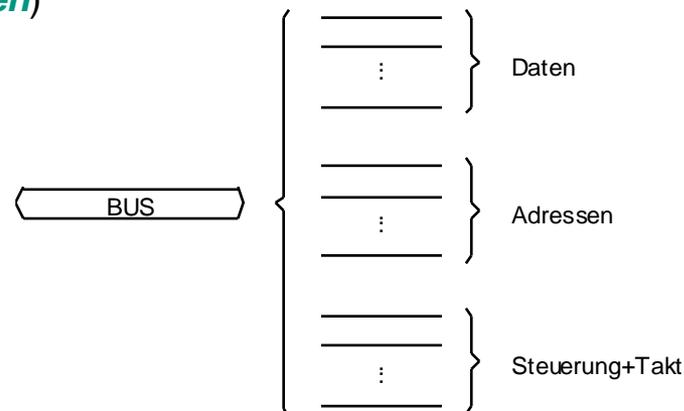
## Prinzip eines Speichers mit statischer Zelle:



- mit vorgestellten **Komponenten** und **Baugruppen** lassen sich **wichtige Funktionen** von **Digitalssystemen** realisieren
- im folgenden sollen einige wenige **Sonderfunktionen** vorgestellt werden

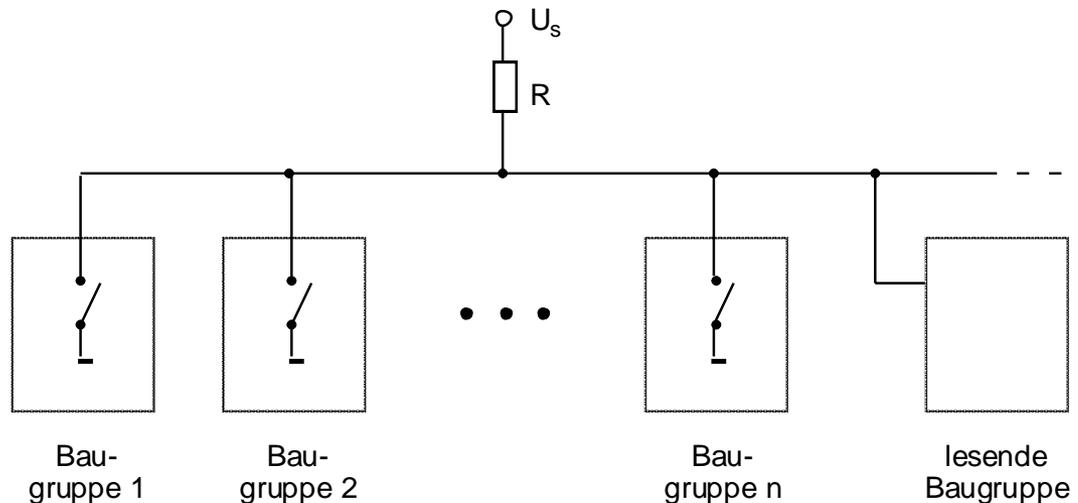
## Busanordnungen:

- in vielen Systemen werden die **Baugruppen** durch **Bündel** von **Signalen** miteinander verbunden, die meist **allen Baugruppen** in gleicher Weise **zur Verfügung stehen**
- die Signale werden in **typische Gruppen** unterteilt:
  - wie z.B. **Daten-**, **Adress-** und **Steuersignale** (einschließlich zentralem Takt)
  - oft wird **Zeitmultiplex-Betrieb** von **Leitungen** vorgesehen (alternativ für **Daten** und **Adressen**)



## Bustechnologie: *Open-Collector* Schaltungen

- sind an **eine Busleitung mehrere Baugruppen angeschlossen**, so müssen mehrere schreibende Ausgänge zusammengeschaltet werden

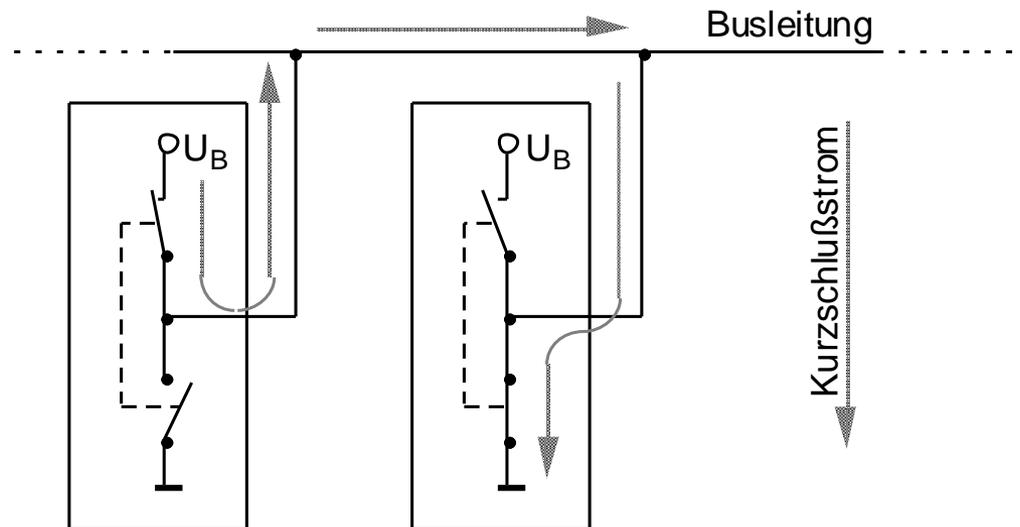


- Solche **Anordnungen** werden als **wired-Verbindungen** bezeichnet und die **realisierte logische Verknüpfung wired-OR** bzw. **wired-AND**
- wegen der unsymmetrischen Lastverhältnisse können diese Schaltungen nur begrenzt eingesetzt werden

## Bustechnologie: Tri-State Schaltungen (I)

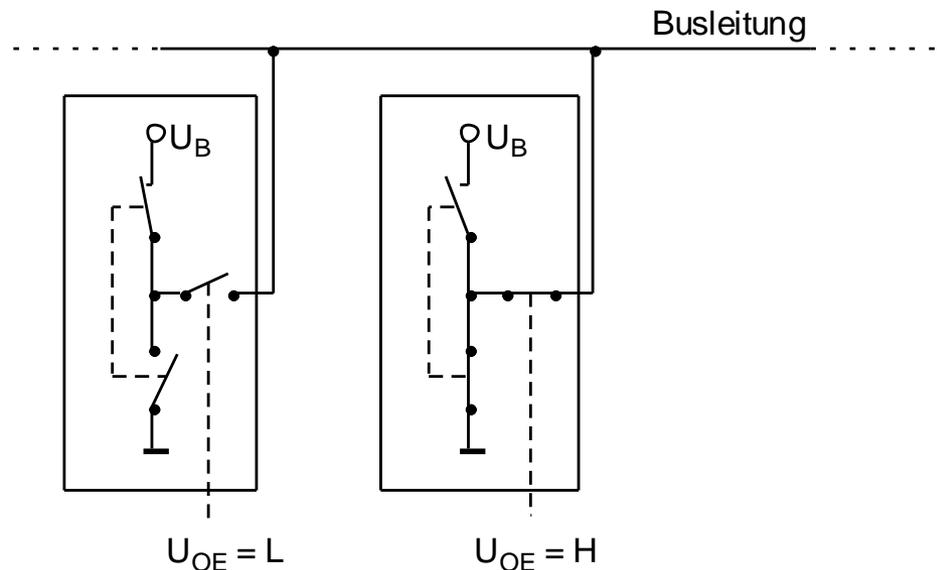
- **zwei Schalter je Variable** sind **besser** in Bezug auf das **Ausgangslastverhalten**
  - es liegt daher nahe dieses Konzept in **Busanordnungen** einzusetzen
- **Verlagerung** eines **Schalterteils** auf den **Bus** (wie beim Widerstand) ist allerdings nicht mehr möglich, was aber zu Kurzschlüssen über mehrere Bausteine hinweg führen kann

### Beispiel:



## Bustechnologie: Tri-State Schaltungen (II)

- um **Kurzschlüsse** zu **vermeiden** muss daher sichergestellt werden, dass **maximal eine Stufe** den **Bus auf  $U_B$  oder Masse** zieht
- dazu wird die **Ausgangsstufe** so **modifiziert**, dass ein **zusätzliches Signal OE** (*output enable*) bei **Nichtauswahl** der Einheit die **Ausgangsstufe vom Bus trennt**



- wird **genau eine Stufe selektiert**, werden **Kurzschlüsse vermieden**

## Bustechnologie: *Tri-State* Schaltungen (III)

- **jede Ausgangstufe** weist nun genau **drei Betriebsfälle** auf:
  - Schreiben von L,
  - Schreiben einer H, und
  - Abgetrennt
- man spricht daher von ***Tri-State* Ausgängen**
- die **Auswahl** einer **Baugruppe** für den **Buszugriff** kann über einen **Decoder** erfolgen
- werden **Busse** in Anordnungen verwendet, in denen **Benutzung** durch **mehrere Komponenten** möglich ist, kann das zu **Zugriffskonflikten** führen
  - es bedarf daher einer **Busverwaltung** (**Arbiter**)