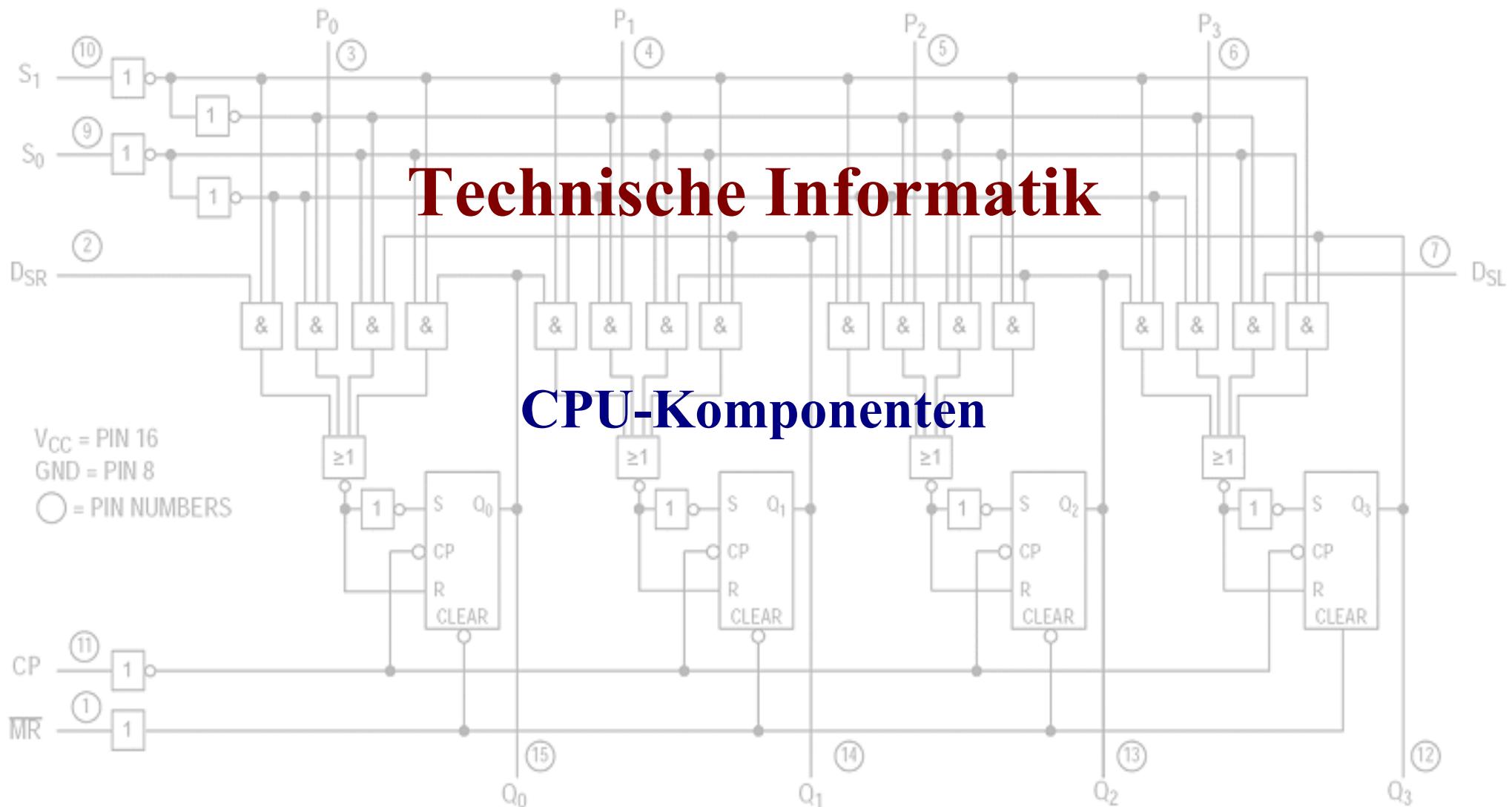


# Technische Informatik

## CPU-Komponenten

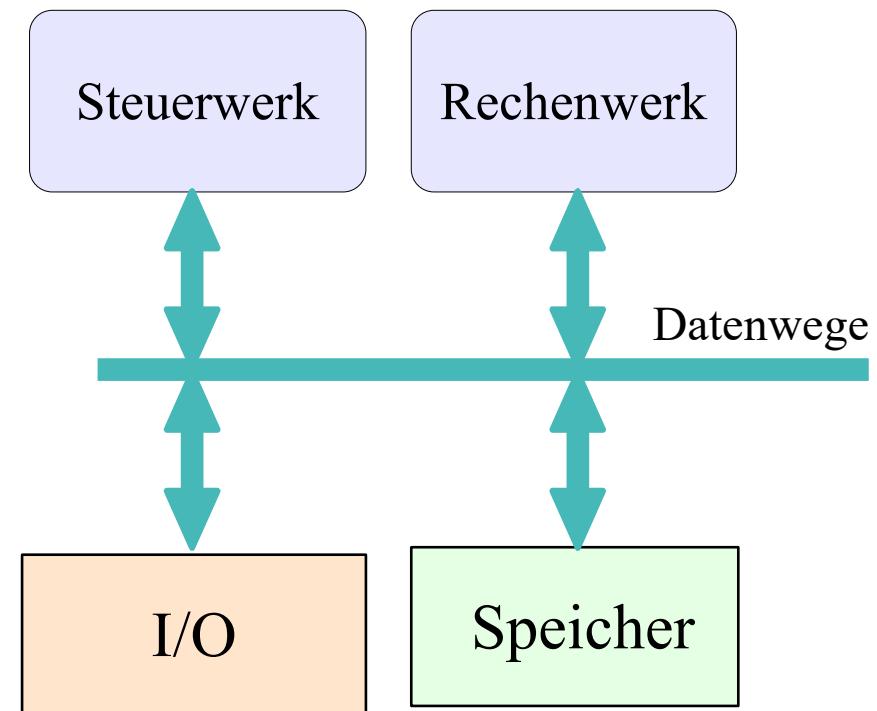


# Grundlegende Strukturen

- Ausgangspunkt:
  - Ein Rechner ist ein programmierbarer, endlicher Automat (State Machine)
  - Programmierbarkeit: Speicher enthält Daten und Anweisungen

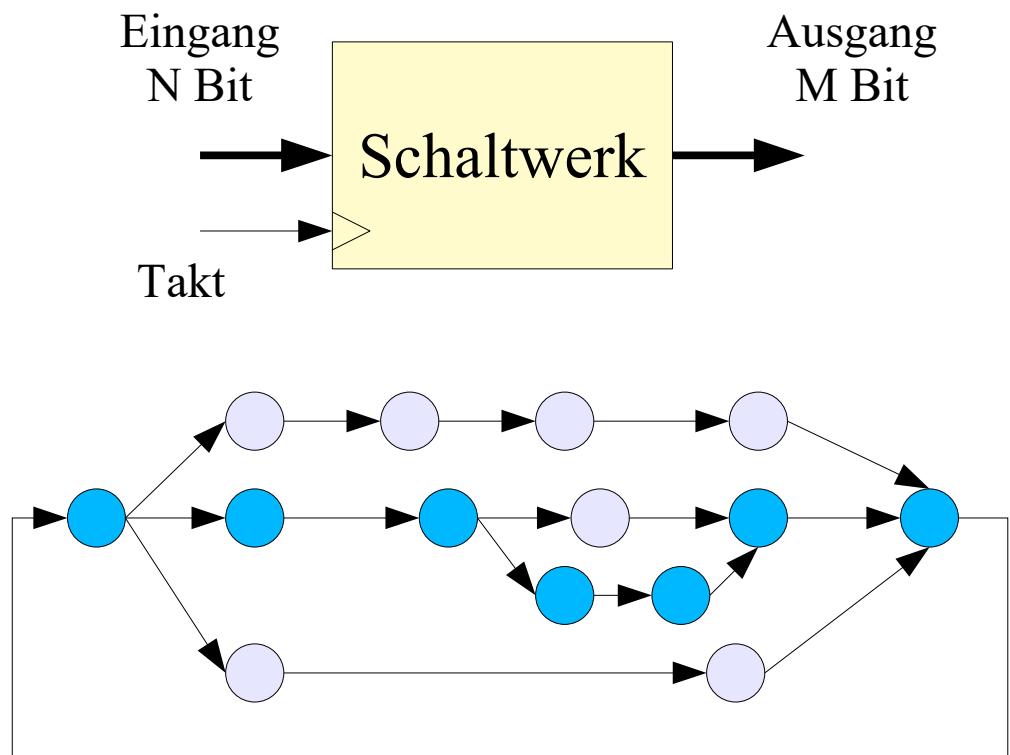
- Grundkomponenten:

- Zentraleinheit (CPU)
  - Steuerwerk
  - Rechenwerk
- Speicher
- I/O
- Datenwege



# Schaltwerk – Endlicher Automat

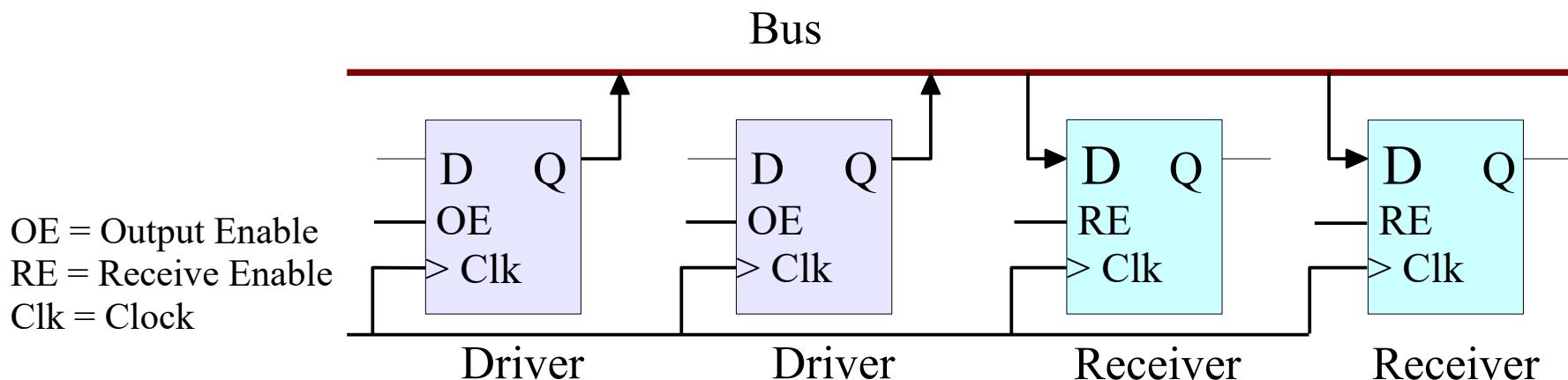
- Schaltwerke aus
  - kombinatorischer Logik (AND, OR, ...)
  - und Speicherelementen (Flip-Flop)
- Abhängig von den N Eingangsbits werden verschiedene Sequenzen durchlaufen und jeweils M Ausgangsbits gesetzt
- Implementierung
  - direkt in Hardware („fest verdrahtet“)
  - durch EEPROM („programmierbar“)



# Bus-System

- Datenübertragung durch Bus-Systeme:

- Die Aus- und Eingänge mehrerer Komponenten bzw. Baugruppen werden an eine gemeinsame Datenleitung angeschlossen
- Eingang, Datensenke:
  - Werden ggf. aktiviert (*Receive Enable*)
- Ausgang, Datenquelle:
  - Zu einem Zeitpunkt darf nur eine Quelle aktiv sein (*Output Enable*), d.h. Teilnehmer müssen sich vom Bus trennen lassen
  - Tri-State = Ausgang hochohmig („abgeschaltet“), wenn OE = 0

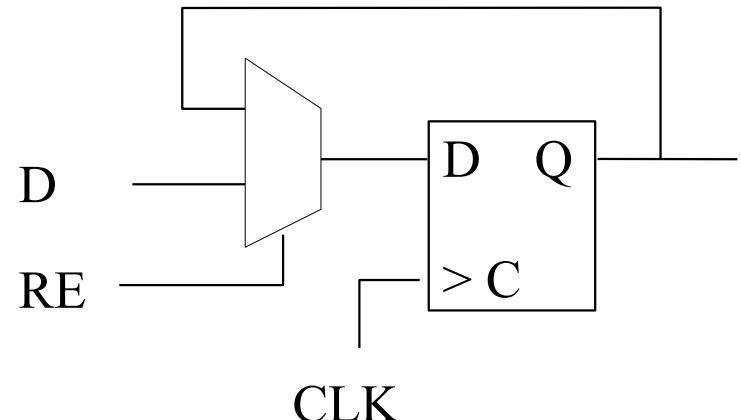


# Datenwege

- Eingang (Read from Bus)

- **Multiplexer**

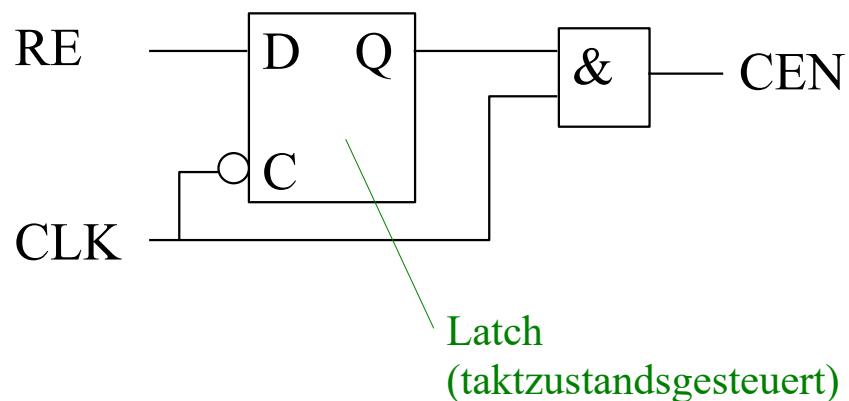
Schaltung:



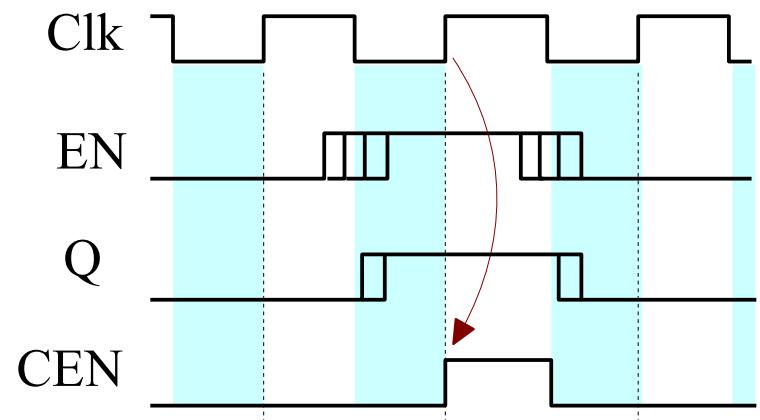
- **Clock Gate**

- Das Steuersignal RE (receive enable) unterbricht das Taktsignal CLK
    - Anwendung: Datenübernahme eines Flip-Flops steuern

Schaltung:



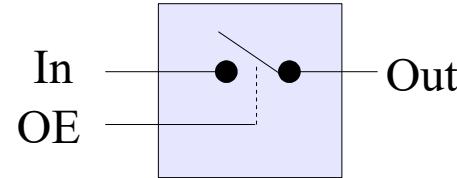
Zustandsdiagramm:



# Bus-System

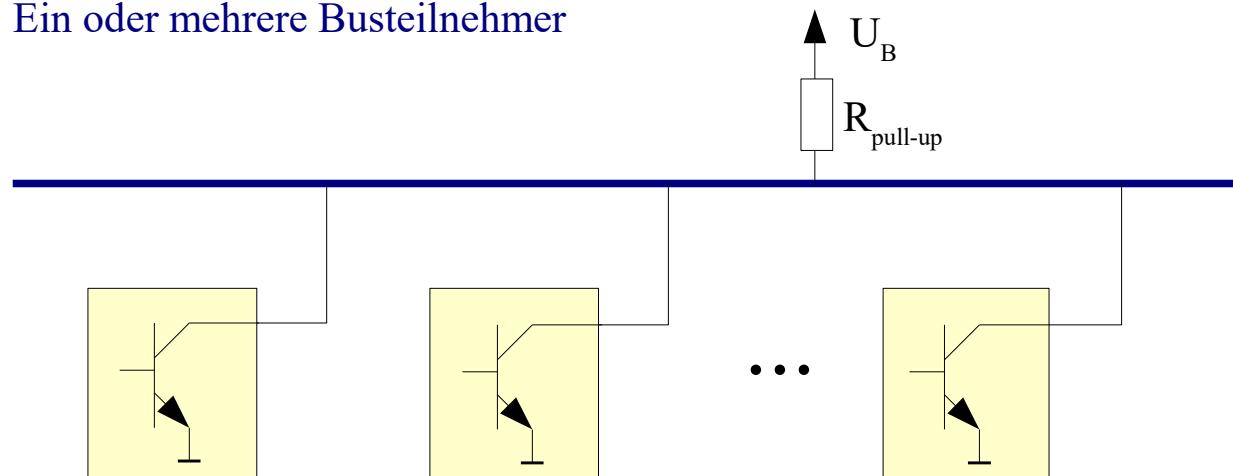
- Write to Bus:

- Tri-State = Ausgang hochohmig („abgeschaltet“), wenn OE = 0
- **Transmission-Gate** oder komplementäre Transistoren



- **Open-Drain**

- Schreibende Busteilnehmer sind ausgangsseitig über einen Transistor mit der Busleitung verbunden. Der Ausgang ist hochohmig, wenn der Transistor sperrt (= geöffneter Schalter)
- Logische „1“ (rezessiv): Alle Busteilnehmer schreiben eine „1“
- Logische „0“ (dominant): Ein oder mehrere Busteilnehmer schreiben eine „0“



# Register

- Register

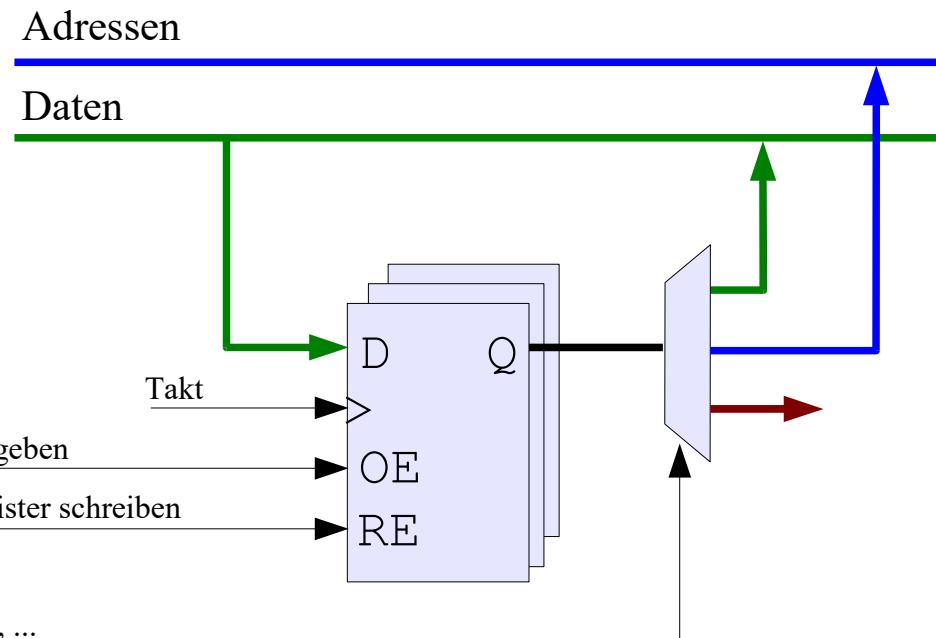
- Datenspeicher, der direkt vom Steuerwerk adressiert wird
  - General Purpose Register
    - Für beliebige Zwecke flexibel einsetzbar  
(z.B. Zwischenspeichern von Daten, Adressberechnungen etc.)
  - Adressregister
    - Stellen Adressen für Speicherzugriffe zur Verfügung
  - Spezialregister
    - Verwendungszweck und/oder zusätzliche Funktionen  
(Program Counter, Stack Pointer, Status Register,...)



Output Enable: Registerinhalt ausgeben

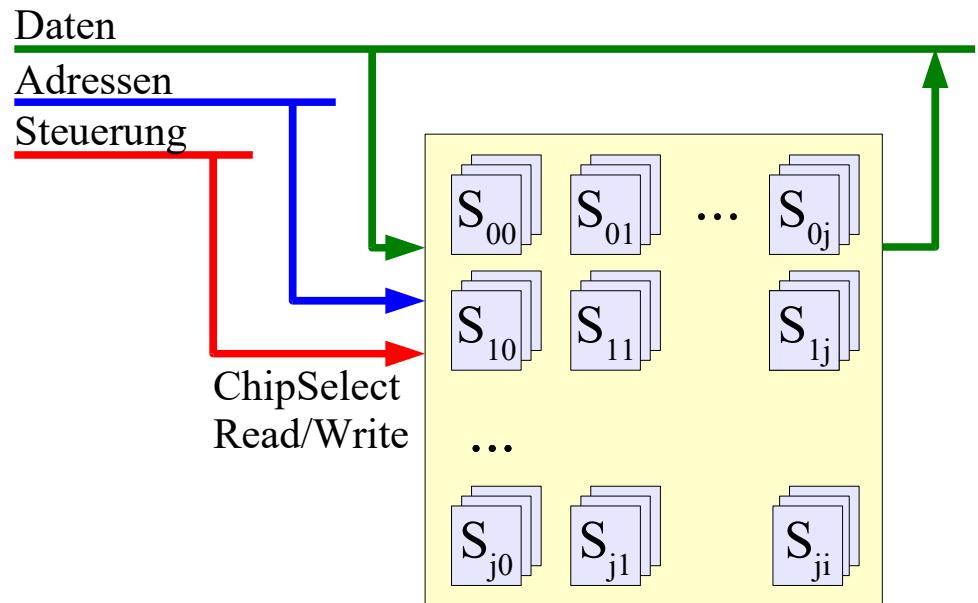
Receive Enable: Daten in das Register schreiben

Registerinhalt als Datum, Adresse, ...



# Speicher

- Datenspeicher
- Adresse (N Bit):
  - Wählt ein von bis zu  $2^N$  Speicherelementen
- Steuersignale:
  - ChipSelect schaltet Speicher aktiv
  - Read/Write entscheidet, ob Daten gelesen oder geschrieben werden



# Rechnerarithmetik

- Rechnerarithmetik
- ALU (Arithmetic Logic Unit)
  - Schaltnetze oder Schaltwerke, die arithmetische und logische Operationen durchführen können
  - Die auszuführende Operation wird durch Steuersignale festgelegt
  - Status-Flags
    - Bits, die Zusatzinformationen zu arithmetischen und logischen Operationen anzeigen
    - Carry flag (CF)
      - Carry-Bit der arithmetischen Operation
    - Zero flag (ZF)
      - Ergebnis einer Operation ist gleich 0
    - Sign flag (SF)
      - 0 wenn Wert positiv, 1 bei negativem Wert
    - Overflow flag (OF)
      - Überlauf, Ergebnis größer als darstellbar

