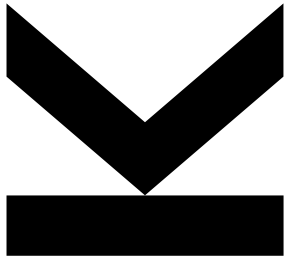
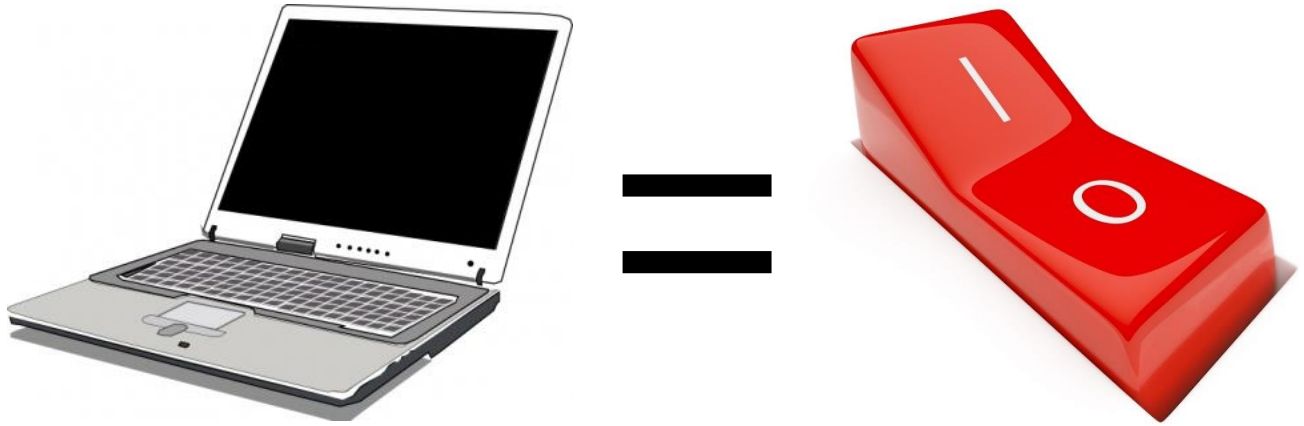
DIGITALE SCHALTUNGEN



Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

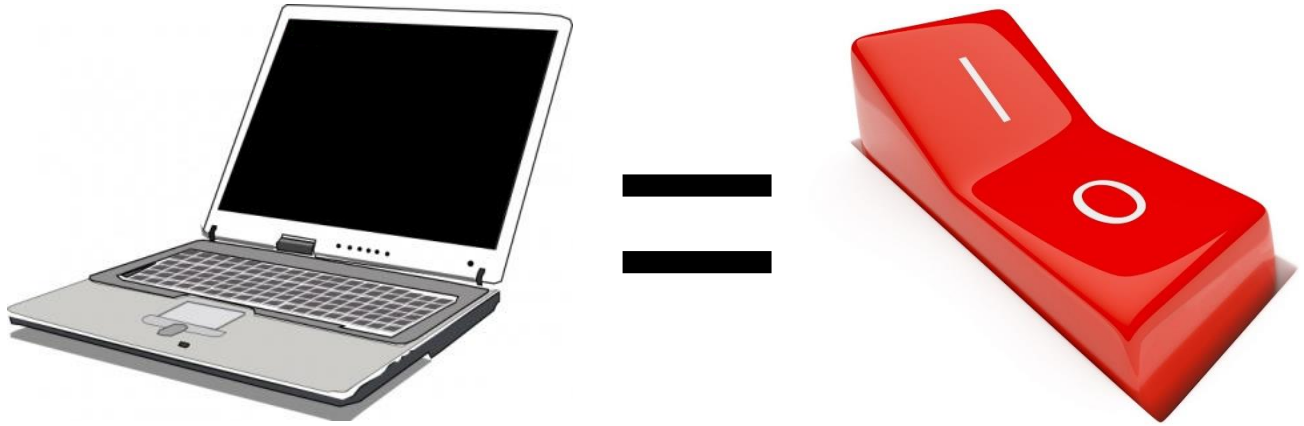
Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

INHALT DER VORLESUNG



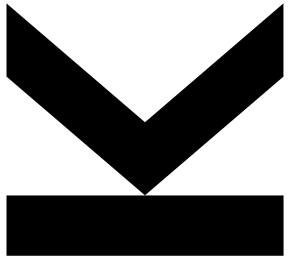
- **Grundlagen**
  - Beschreibungen über „0“ und „1“ (Boolesche Algebra)
  - Beschreibungen von Schaltungen
- **Rechnen**
  - Darstellung von Zahlen
  - Digitale Schaltungen für Addition, Subtraktion, Multiplikation
- **Speichern**
  - Sequentielle Schaltungen
  - Speicherelemente
- **Steuern**
  - Endliche Automaten
  - Synthese von Steuerwerken
- **Entwerfen**
  - Synthese von allgemeinen Schaltungen
  - Logikminimierung

INHALT DER VORLESUNG



- **Grundlagen**
  - Beschreibungen über „0“ und „1“ (Boolesche Algebra)
  - Beschreibungen von Schaltungen
- **Rechnen**
  - Darstellung von Zahlen
  - Digitale Schaltungen für Addition, Subtraktion, Multiplikation
- **Speichern**
  - **Sequentielle Schaltungen**
  - **Speicherelemente**
- **Steuern**
  - Endliche Automaten
  - Synthese von Steuerwerken
- **Entwerfen**
  - Synthese von allgemeinen Schaltungen
  - Logikminimierung

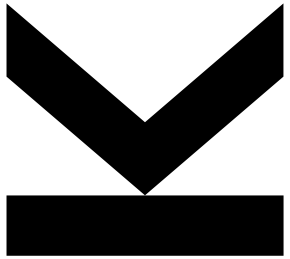
SPEICHERN



Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

SEQUENTIELLE SCHALTUNGEN

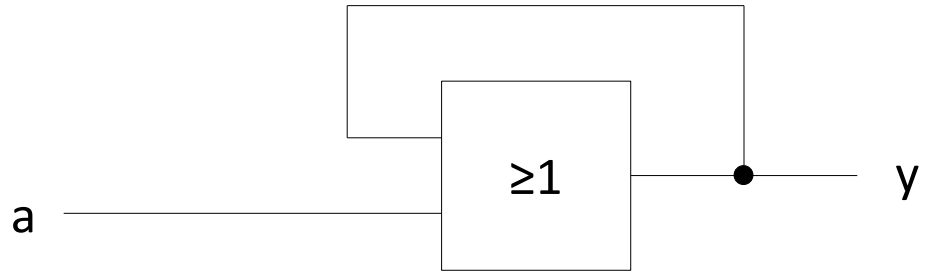


Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

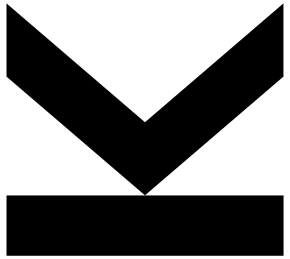
RÜCKKOPPLUNGEN

- Was passiert, wenn Ausgang einer Schaltung auf Eingang zurückwirkt?



- Annahme: *a* und *y* am Anfang 0
  - Solange *a* auf 0 bleibt, keine Änderung an y
  - *a* wird 1, dann wird *y* auch 1 und bleibt so
  - Schaltung hat „Gedächtnis“

SPEICHERBAUSTEINE

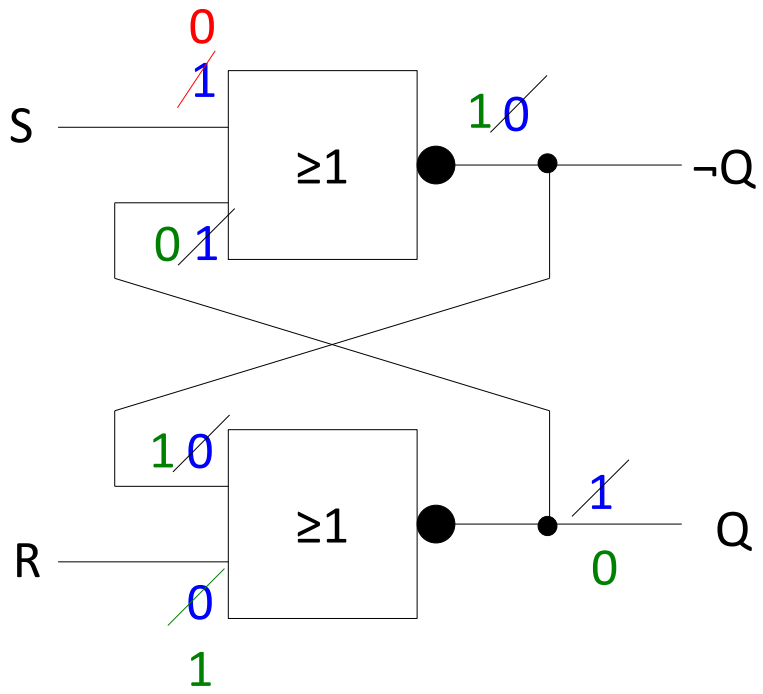


Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

RS-FLIPFLOP

- Schaltung zum Speichern von Information
- Eingänge zum Setzen (Set) und Löschen (Reset)



Annahme: *S=1, R=0*

Annahme: *S=0, R=0*

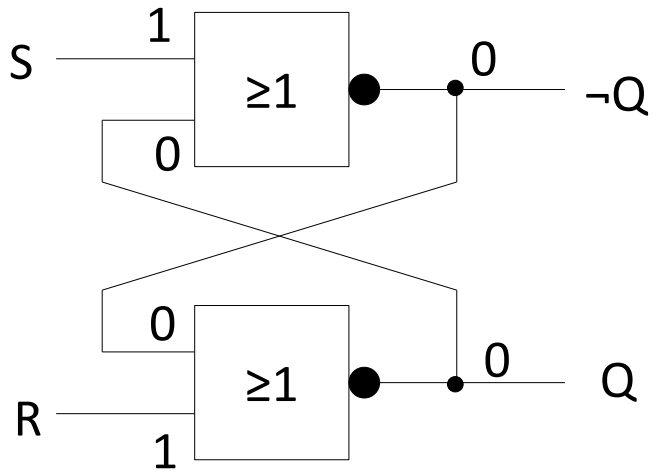
Annahme: *S=0, R=1*

RS-FLIPFLOP: WAHRHEITSTABELLE

R	S	Q(t+1)	¬Q(t+1)	Verhalten
0	**0**	**Q(t)**	**¬Q(t)**	**Speichern**
1	**0**	**0**	**1**	**Rücksetzen**
0	**1**	**1**	**0**	**Setzen**
1	1	?	?	?

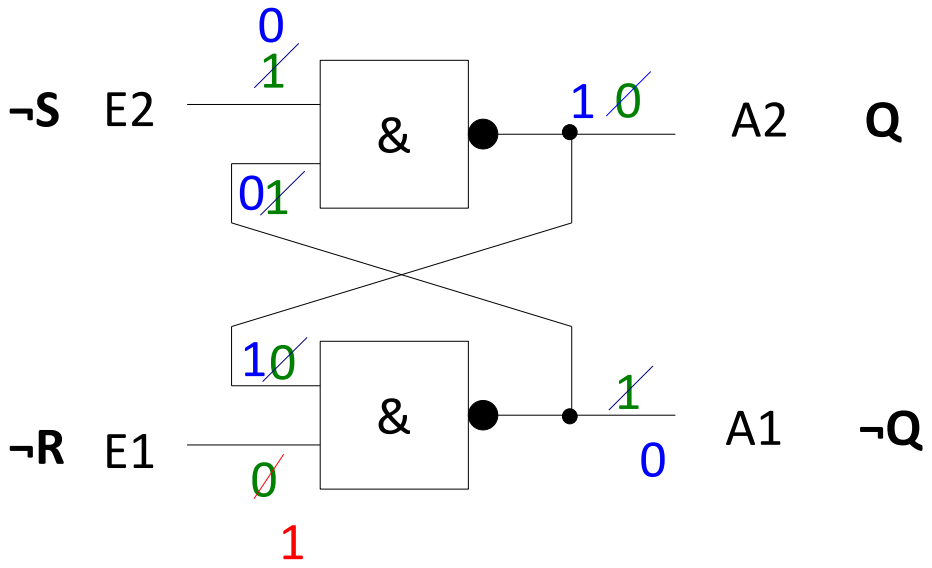
- *Q(t)* : aktueller Zustand
- *Q(t+1)* : Folgezustand
- Beide Eingänge 0 ⇒ Zustand wird gespeichert
- Was passiert bei *R=1* und *S=1*?

RS-FLIPFLOP: VERBOTENER ZUSTAND



- Beide Ausgänge werden 0!
  Prinzipiell nicht erlaubt, da $\neg Q$ immer die Negation von Q sein soll
- Zustand hält solange, bis R oder S wieder 0 werden

RS-FLIPFLOP MIT NAND



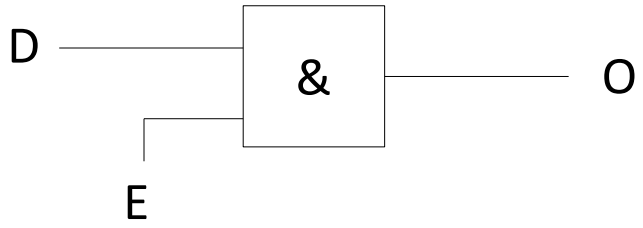
E1	E2	A1(t+1)	A2(t+1)
0	1	1	0
1	1	A1(t)	A2(t)
1	0	0	1
0	0	1	1

- *S=0* und *R=0* verbotener Zustand
- *S* und *R* sind negiert, Q und ¬Q vertauscht

NACHTEIL DES RS-FLIPFLOP

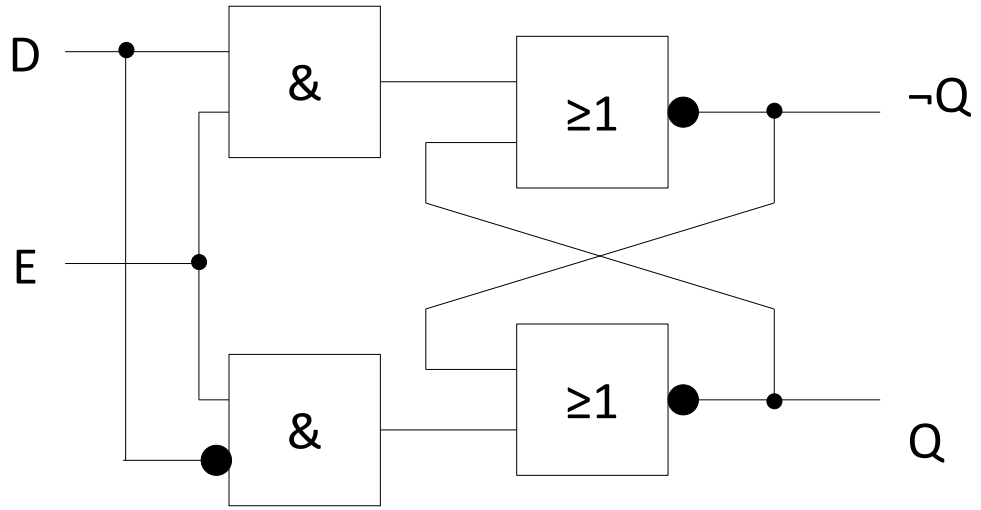
- Beim Speichern eines Wertes 0 oder 1 muss man den Wert kennen!
- Ziel: Speichern *unbekannter* Werte

UND-GATTER ALS „TOR“



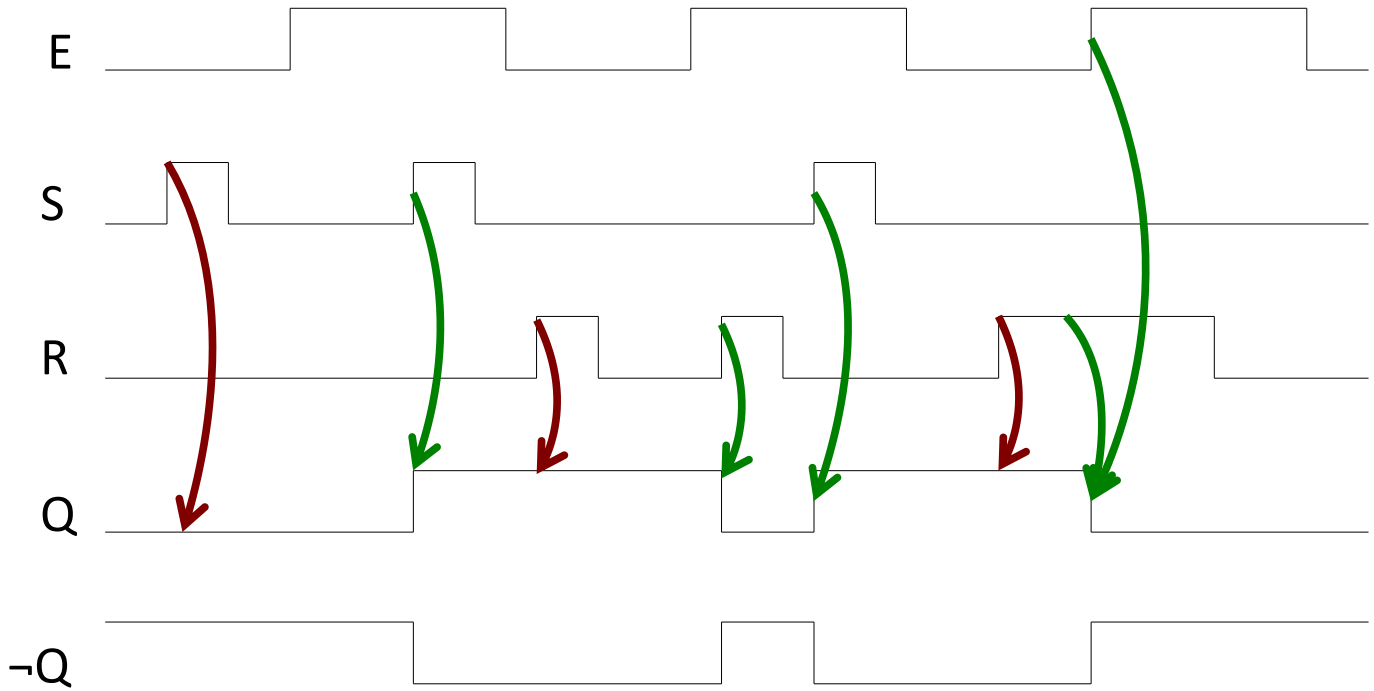
- Und-Gatter kann wie ein „Tor“ interpretiert werden
  - Ein Dateneingang
  - Ein Steuereingang
- Steuereingang=1
  - Dateneingang wird auf Ausgang durchgelassen
- Steuereingang=0
  - Datenineingang wird nicht durchgelassen

D-LATCH



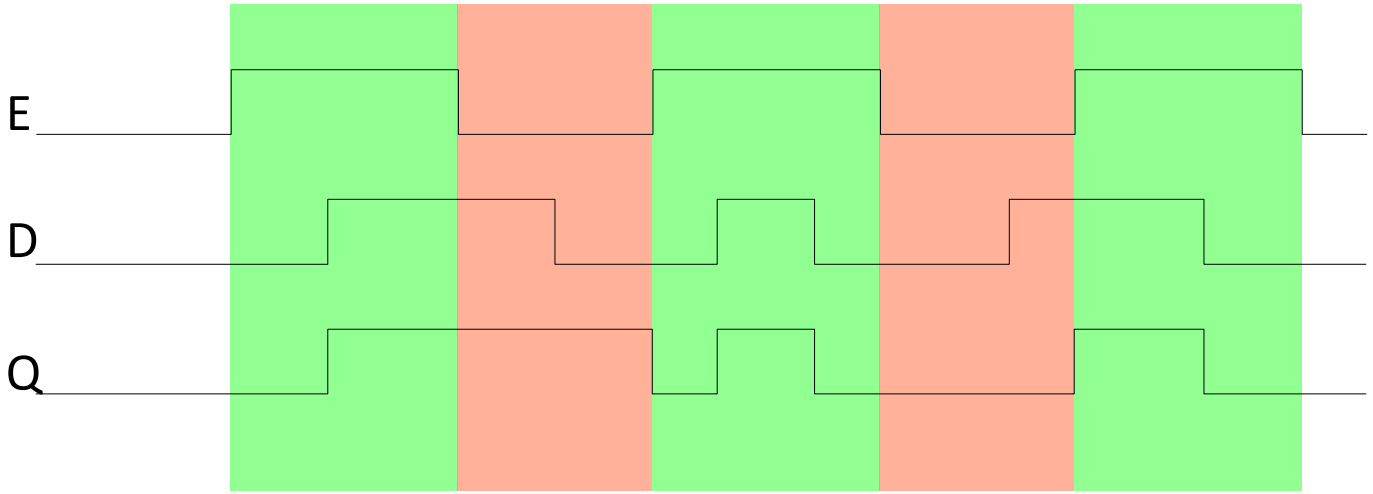
- *S* und *R* werden aus *D* berechnet
  S = D, R = ¬D
- Enable steuert die Datenübernahme

RS-LATCH: ZEITDIAGRAMM



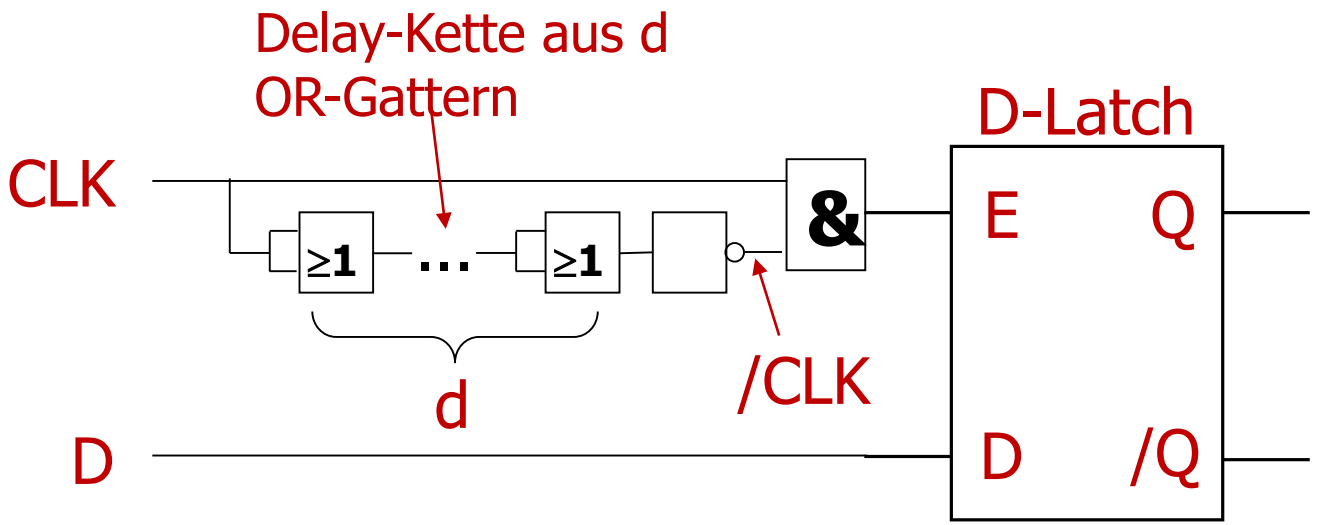
EIGENSCHAFTEN EINES D-LATCHES

- Das D-Latch heißt transparent, wenn das Schreibsignal aktiv ist.
- E muss lange genug aktiv sein, damit sich der neue Zustand im RS-FF einstellen kann.
- Das D-Latch ist pulsgesteuert (Schreibpuls E)
- Einfache 1-Bit Speicherzelle

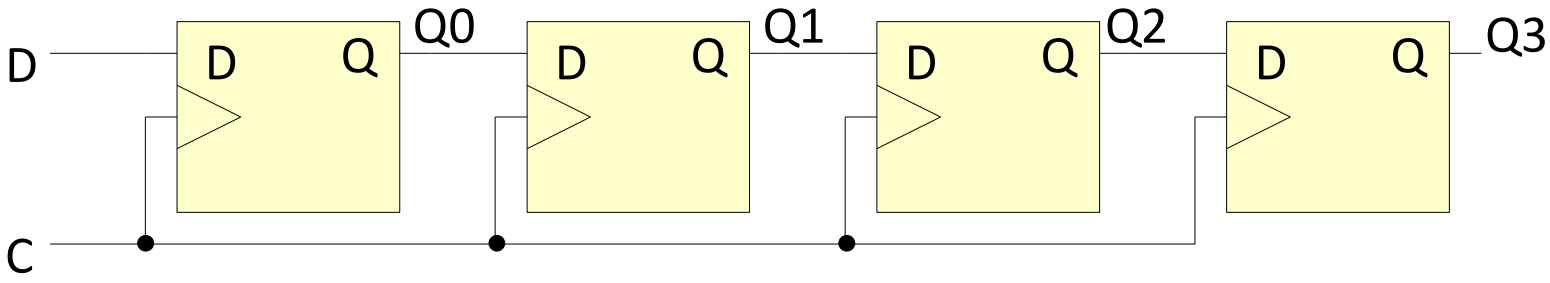


TAKTFLANKENGESTEUERTE D-FLIPFLOPS

- Taktflankengesteuerte Flip-Flops wie das D-Flip-Flop übernehmen Daten zu einem bestimmten Zeitpunkt (kein transparenter Modus!), nämlich bei der steigenden Flanke des sog. Clocksignals
- Vorteil:
  Daten müssen lediglich bei der steigenden Taktflanke stabil sein (zzgl. Setup- und Holdzeit)

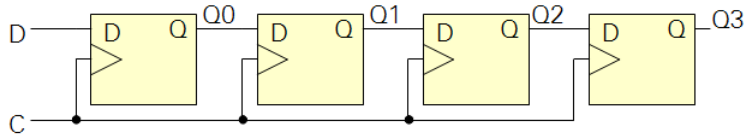


SCHIEBEREGISTER #1



- Alle D-FF übernehmen bei steigender Flanke
- Änderungen geschehen gleichzeitig
  Q0=D, Q1=Q0, Q2=Q1, Q3=Q2
- Information wird um eine Bitposition nach rechts geschoben

SCHIEBEREGISTER #2

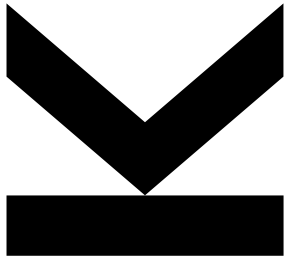


Wert am Eingang vor dem Takt

Takt	D	Q0	Q1	Q2	Q3
		0	1	0	1
1	0	0	0	1	0
2	1	1	0	0	1
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Belegung vor dem Start

SPEICHER

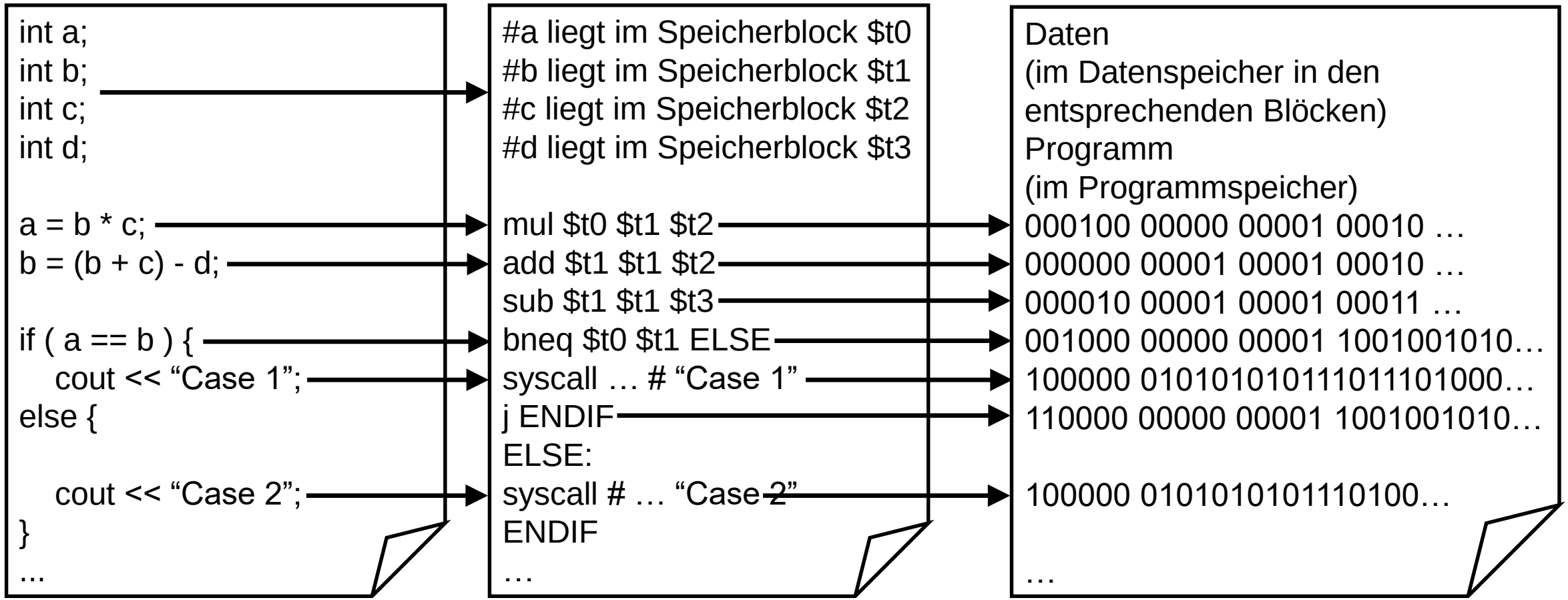


Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

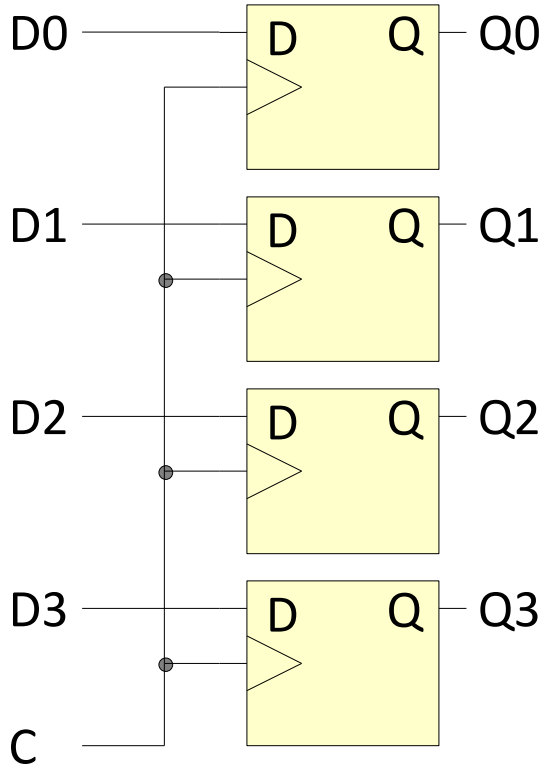
RECHNERSICHTEN

- Höhere Programmiersprache
- Assembly
- Maschinensprache



REGISTER

- Gruppierung von D-FF
- Gemeinsames Taktsignal
- Separater Ein- und Ausgang für jedes FF
- Alle FF übernehmen gleichzeitig die Eingangssignale
- Z.B. zur Speicherung von Variablen, Zwischenergebnissen, etc.

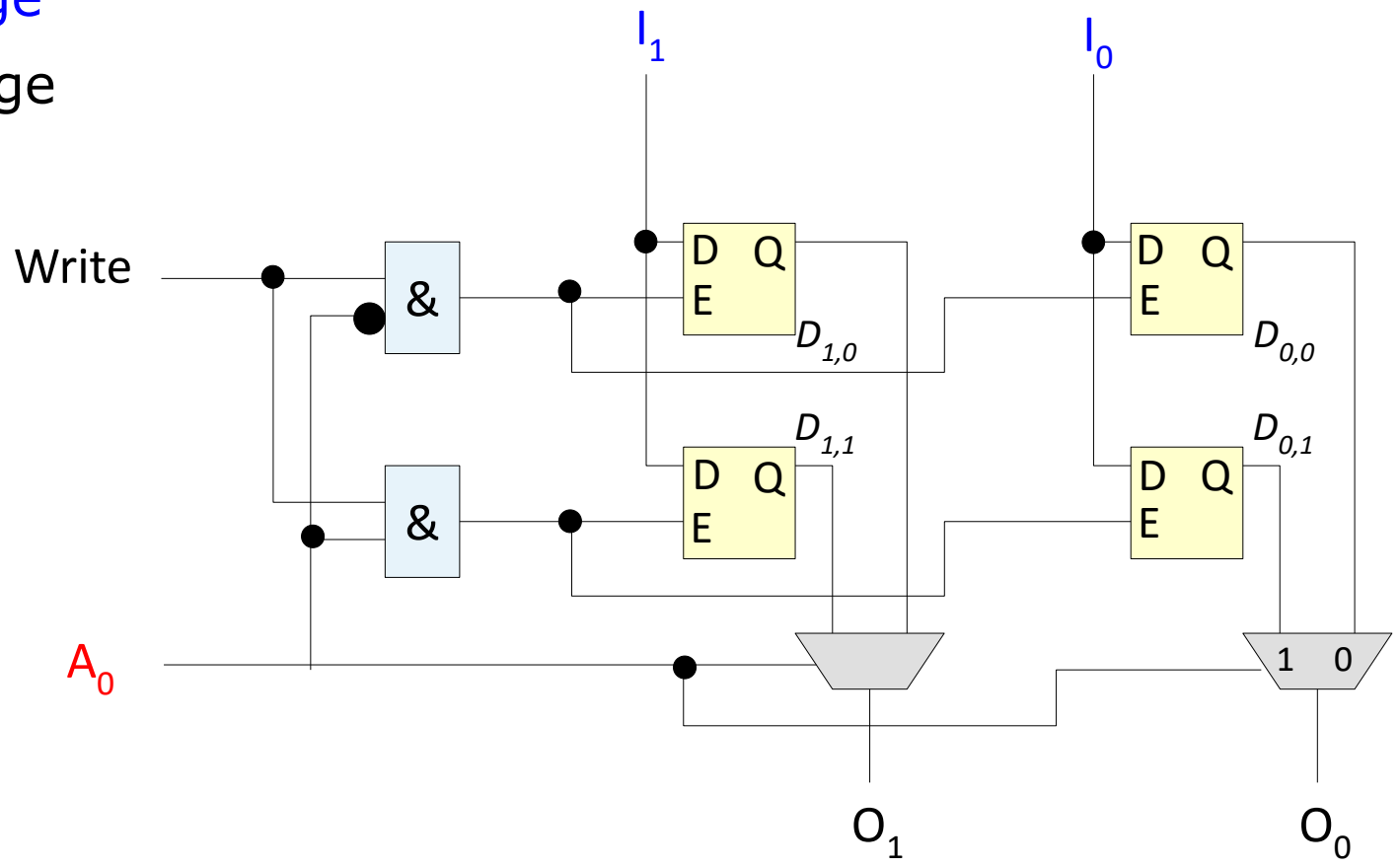


SPEICHER

- Bisher:
  - Speicherung von Bits in FlipFlops
  - Speicherung von Worten in Registern
- Gefordert:
  - Schaltung zur Speicherung großer Mengen von Informationen
  - Selektiver Zugriff auf Teile der gespeicherten Information
- Erster Lösungsansatz:
  - Speicherung in D-Latches
  - Auswahl durch Spezifikation eines Teilbereichs des Speichers(Adresse)

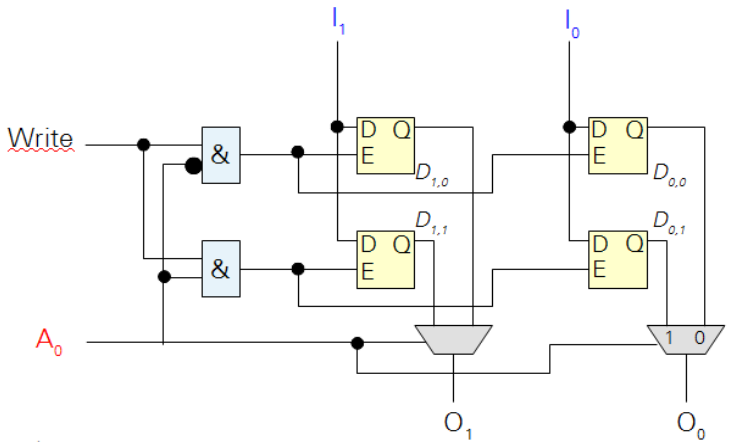
BEISPIEL: 2X2 BIT SPEICHER

- 1 Adressleitung
- 2 Dateneingänge
- 2 Datenausgänge



BEISPIEL: AUSLESEN DES SPEICHERS

- Anlegen der gewünschten Adresse
- Nach Verzögerungszeit des Multiplexers liegt Datum vor



Übertragbar auf mehrere Adressleitungen

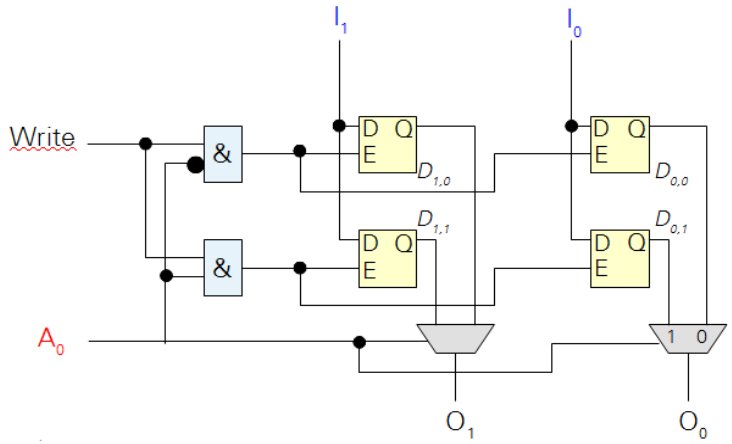
- *n* Adressleitungen ⇒ 2^n-zu-1 Multiplexer
- Manchmal auch 2^n Speicherplätze genannt

Übertragbar auf beliebige Wortbreiten

- Typische Wortbreiten: 4, 8, 16, 32, 64 Bit
- Innerhalb eines Chips meistens nur bis zu 16 Bit Wortbreite

BEISPIEL: SCHREIBEN IN SPEICHER

- Anlegen der gewünschten Adresse und des zu schreibenden Datums
- Aktivieren der Schreibleitung
- Latches übernehmen Daten
- Deaktivieren der Schreibleitung (Speicher ändert sich nicht mehr)



Übertragbar auf mehrere Adressleitungen

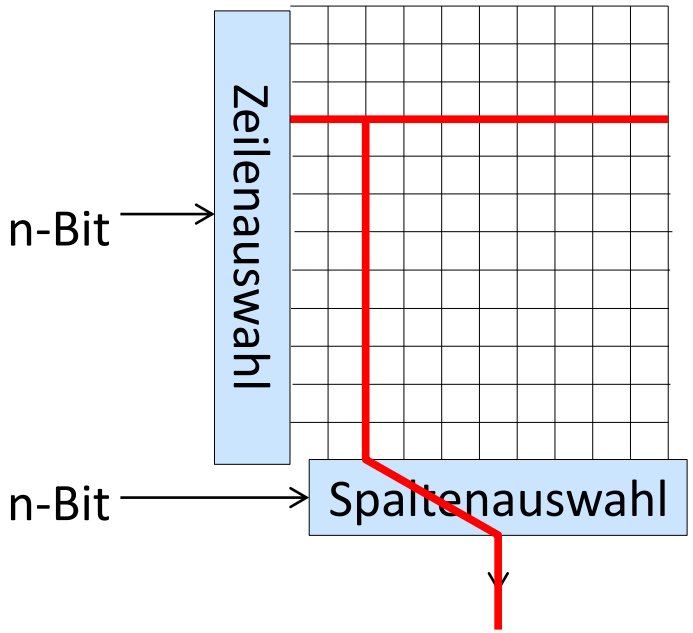
- Schreibimpuls muss jeweils für ein Wort generiert werden
- Und-Verknüpfung muss eine eindeutige Kombination der Adressbits selektieren

REALE SPEICHER #1

- Speicherung in D-Latch zu aufwändig
- In realen Speichern wird jeweils ein Bit gespeichert:
  - In einem Kondensator
    - Dynamic Random Access Memory (DRAM)
    - Verliert seine Ladung, muss daher regelmäßig aufgefrischt werden
  - In einer 6-Transistor-Zelle
    - Static Random Access Memory (SRAM)
    - Hält die Information beliebig lang
  - Spezielle Varianten für Festwertspeicher (ROM, EPROM, ...)

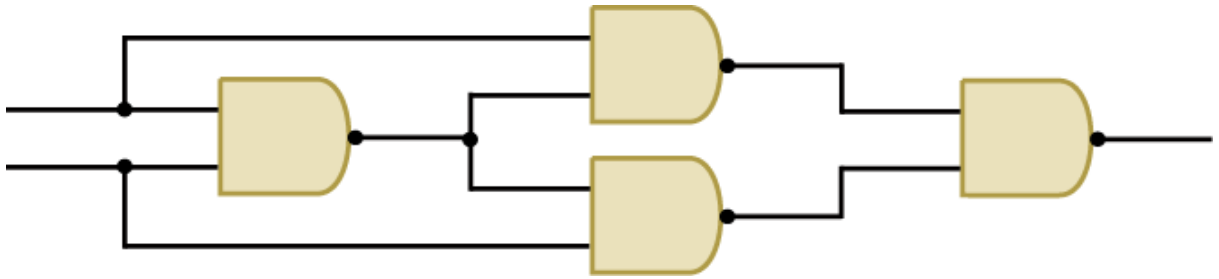
REALE SPEICHER #2

- Selektion der Ausgabewerte durch Multiplexer zu aufwändig
- Reale Selektion eines Speicherelementes
  - Anordnung der Speicherelemente in einer Matrix
  - Zweistufige Adressierung
  - Zunächst Aktivierung einer Zeile der Matrix
  - Dann Selektion eines Elementes der Zeile
  - Matrix meist Quadratisch
  - Beide Auswahlschaltungen verarbeiten die Hälfte der Adressbits



SEQUENTIELLE SCHALTUNGEN

- Kombinatorische Schaltungen



- Sequentielle Schaltungen