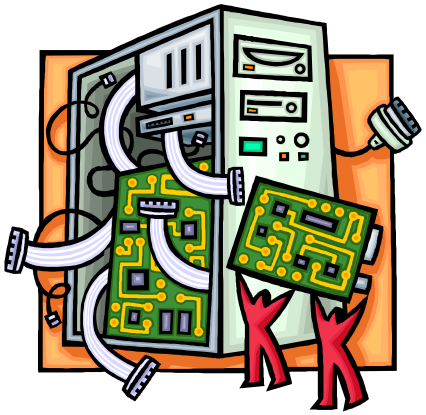
Rechtlicher Hinweis und Copyright

Umgang mit dem Kursmaterial

- Das Kursmaterial ist durch das Copyright des Dozenten und/oder anderer Autoren geschützt.
- Das Material darf nur von Studenten der angegebenen Fachhochschule und nur zu Ausbildungszwecken im Rahmen des angegebenen Kurses verwendet werden.
- Die Veröffentlichung oder Verbreitung des Materials ist ausdrücklich untersagt. Dazu zählen das Veröffentlichen auf Webseiten, Onlinespeichern wie Dropbox, Verbreiten per Email, die Verwendung in Vorträgen oder Publikationen etc.

Vorlesung_09

Standard-Schaltnetze und -Schaltwerke

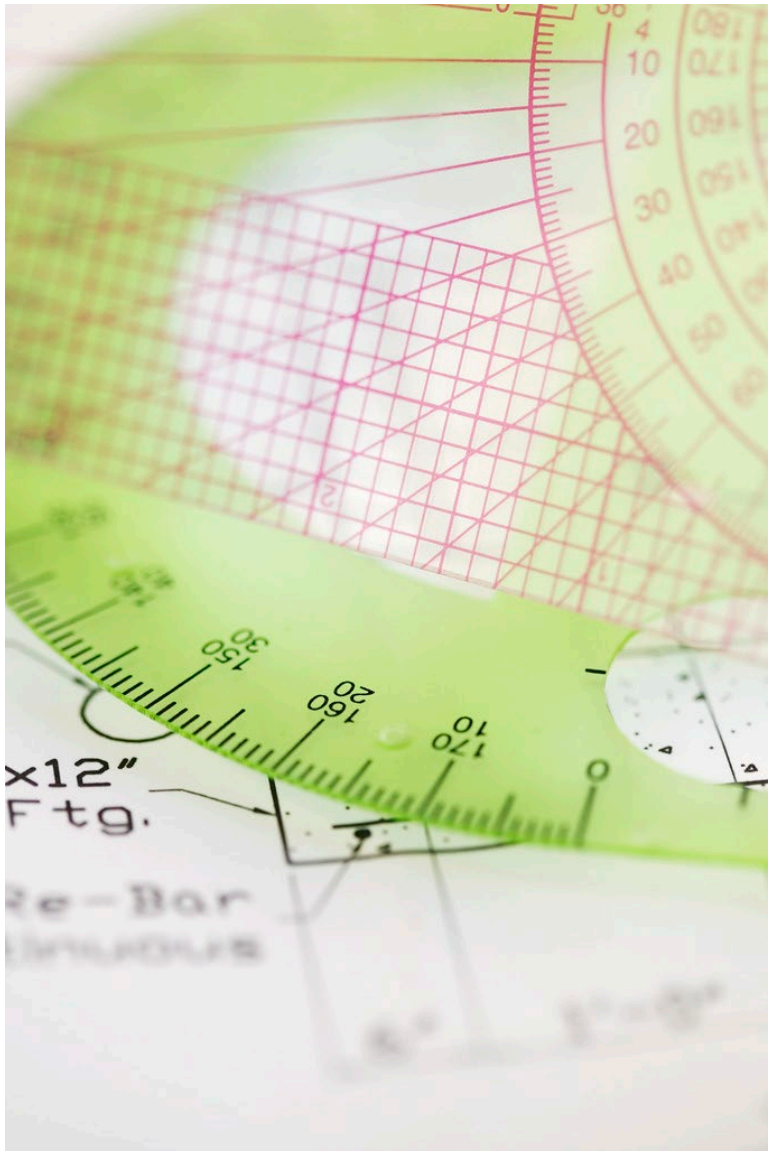


Technische Grundlagen der Informatik

Prof. Dr. Claus Führner
Fakultät Informatik
Ostfalia Hochschule für angewandte Wissenschaften
Braunschweig/Wolfenbüttel

Agenda

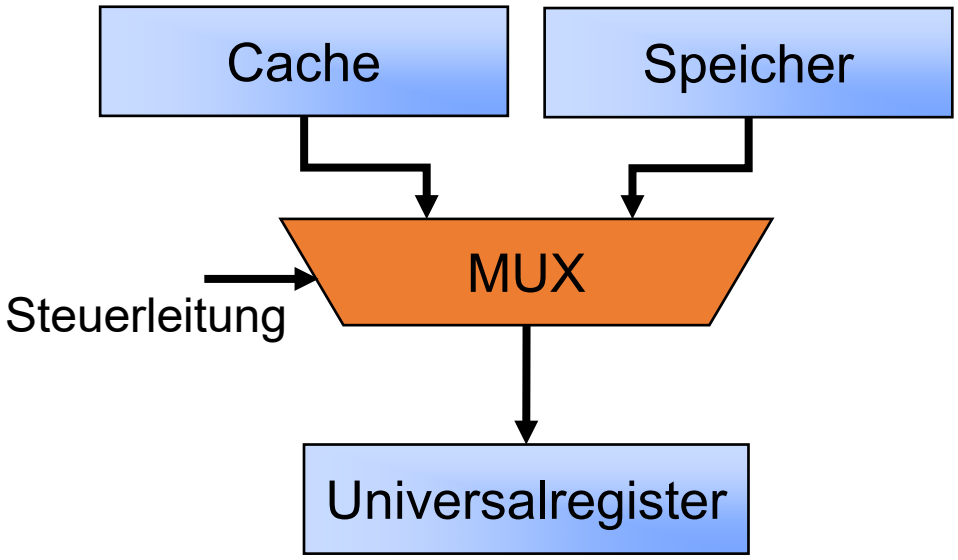
- Standard-Schaltnetze
  - Multiplexer
  - Demultiplexer
  - Programmierbare Logik
- Standard-Schaltwerke
  - (Auffang-)Register
  - Schieberegister
  - Zähler



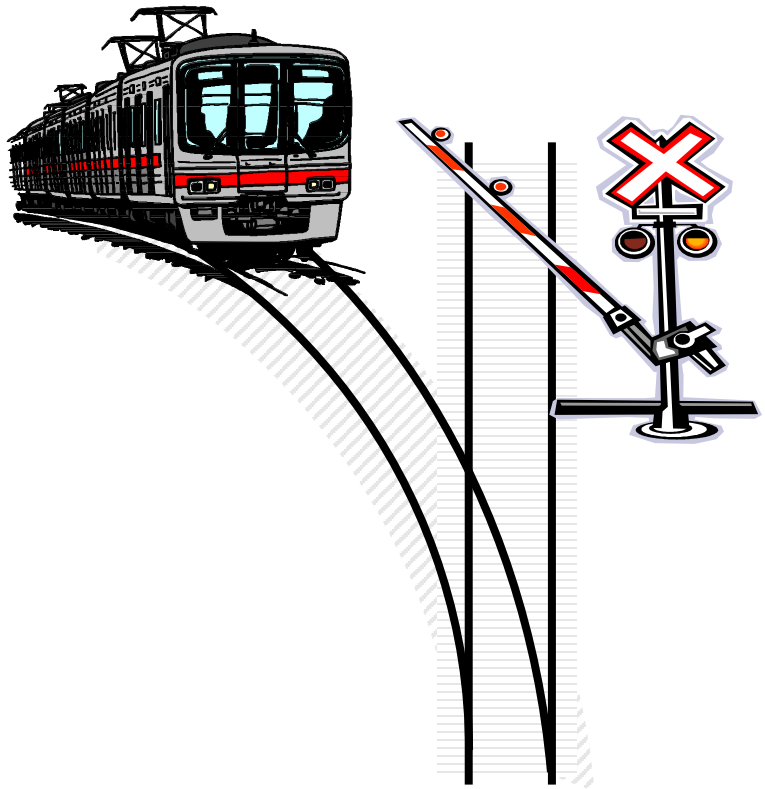
Multiplexer

Analogie

Ein Multiplexer führt mehrere Datenquellen in einem Ausgang zusammen, wobei die gewünschte Datenquelle über eine Steuerleitung ausgewählt wird. Beispiel (aus CPU):



Analogon Schienenverkehr: der Multiplexer entspricht einer (stumpf befahrenen) Weiche



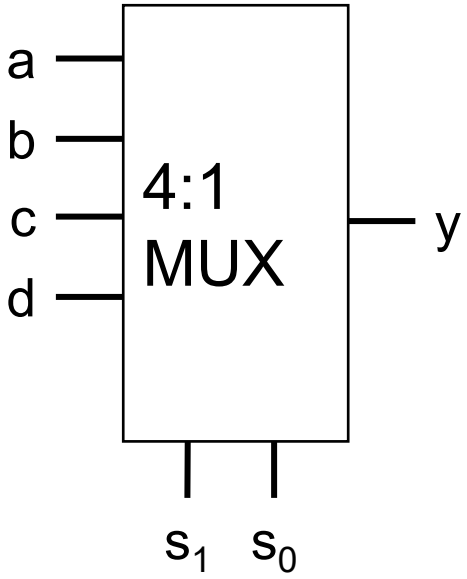
Multiplexer

Beschreibung

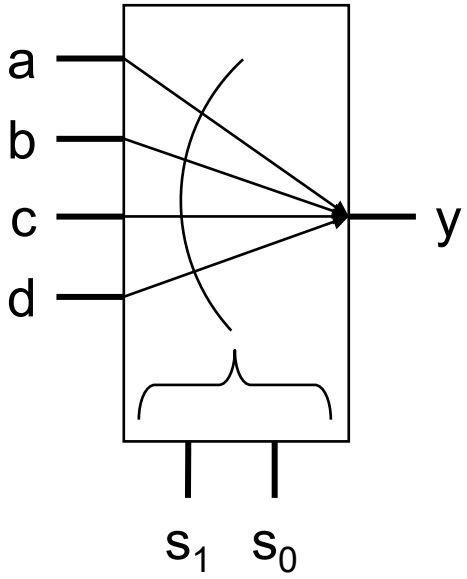
Eigenschaften:

- Baustein mit mehreren Eingängen und einem Ausgang
- Über *n* Steuerleitungen wird einer der 2^n Eingänge durchgeschaltet

Schaltbild und logisches Verhalten eines 1 aus 4 Multiplexers:

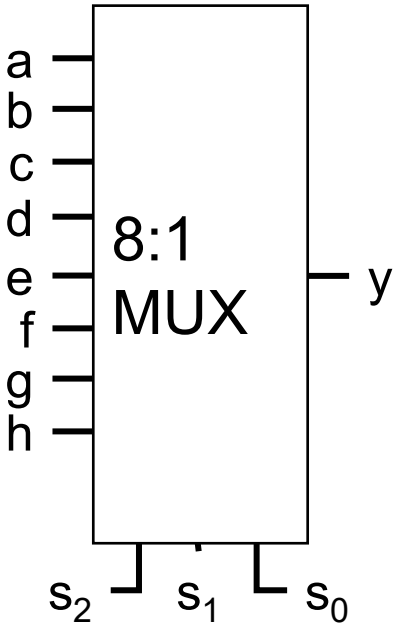


s_1	s_0	y
0	0	a
0	1	b
1	0	c
1	1	d



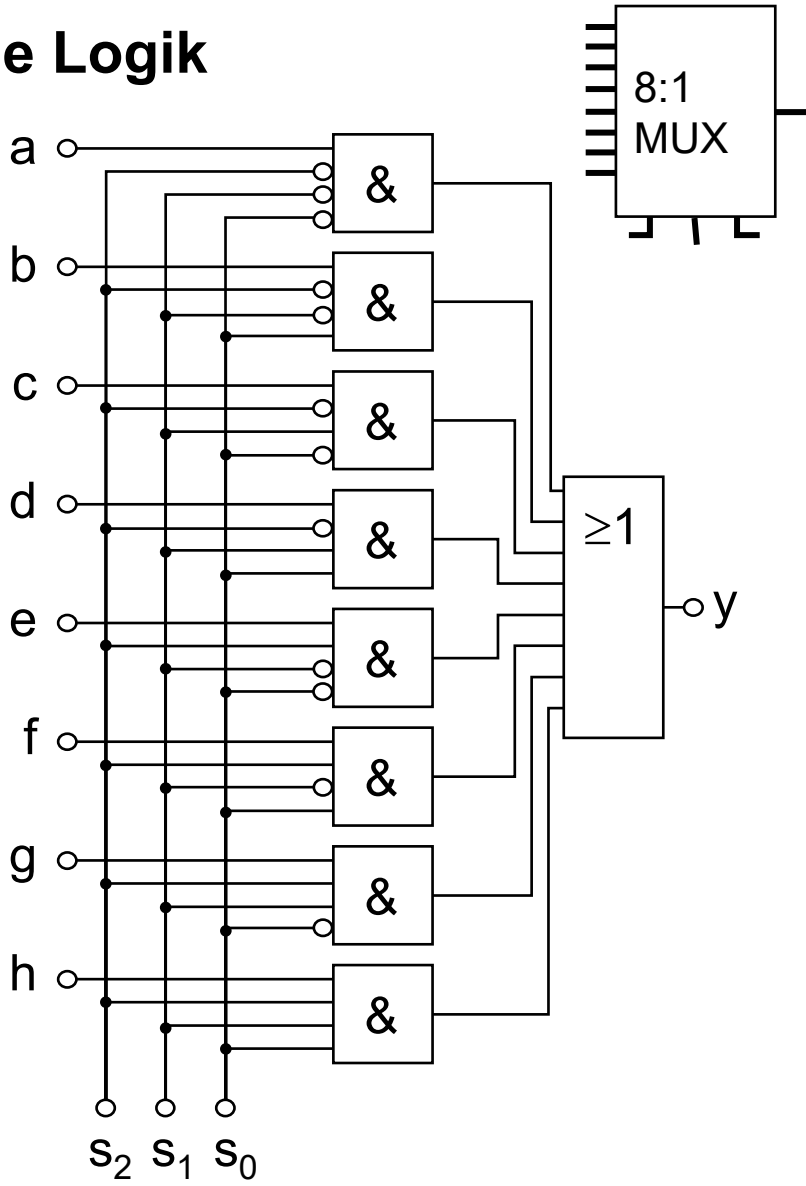
Multiplexer

Realisierung 8:1 MUX als zweistufige Logik



s_2	s_1	s_0	y
0	0	0	a
0	0	1	b
0	1	0	c
0	1	1	d

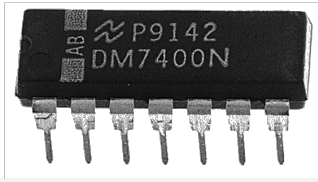
s_2	s_1	s_0	y
1	0	0	e
1	0	1	f
1	1	0	g
1	1	1	h

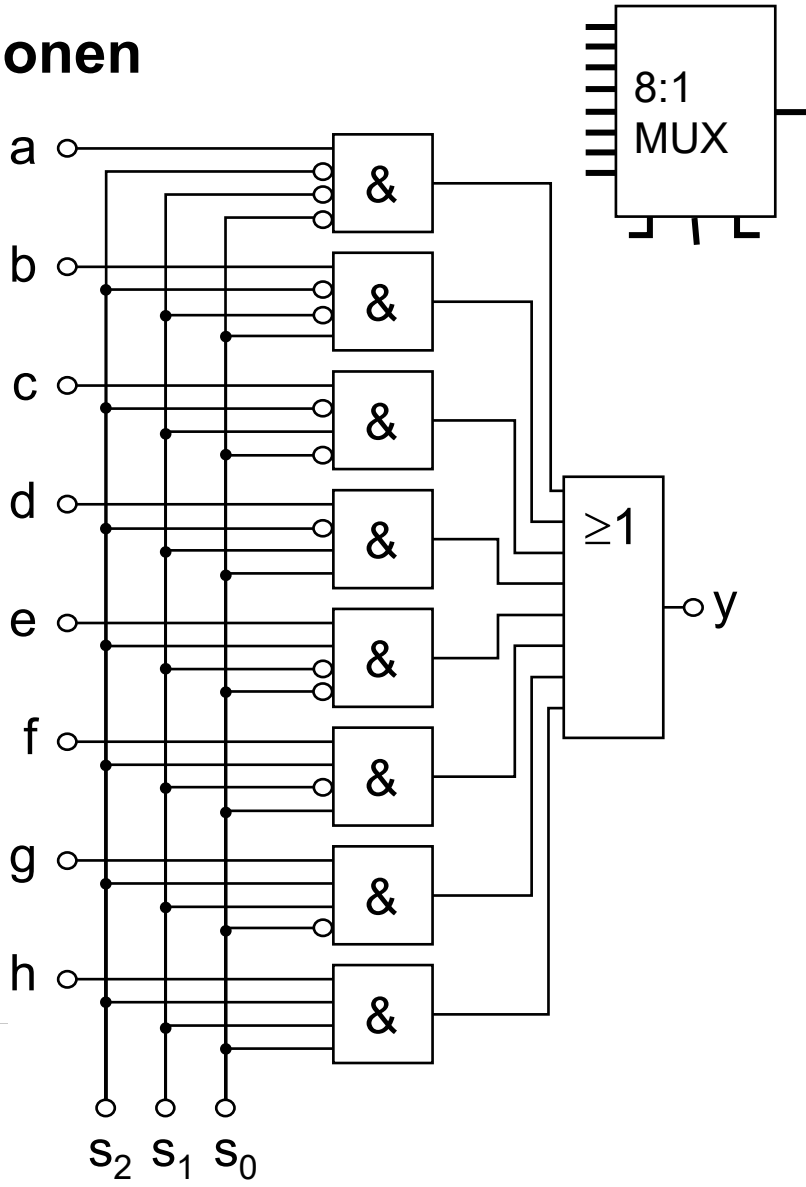


Multiplexer

Realisierung beliebiger Schaltfunktionen

- Struktur des internen Schaltnetzes eines Multiplexers aus AND und OR entspricht genau der DNF
- $2^n : 1$ Multiplexer ermöglichen die Realisierung einer DNF mit n Eingangsvariablen
- Eingangsvariablen mit Steuer-eingängen s_0, s_1, ... verbinden
- Eingang auf 1 setzen, wenn Minterm vorhanden (y=1 für Eingangsvariablen). Sonst 0.
- Multiplexer sind als fertige Bausteine erhältlich, z. B. 74HC151 8:1 MUX





Multiplexer

Realisierung beliebiger Schaltfunktionen mit 2:1 MUX

c	b	a	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b	a	y_0
0	0	0
0	1	0
1	0	1
1	1	0

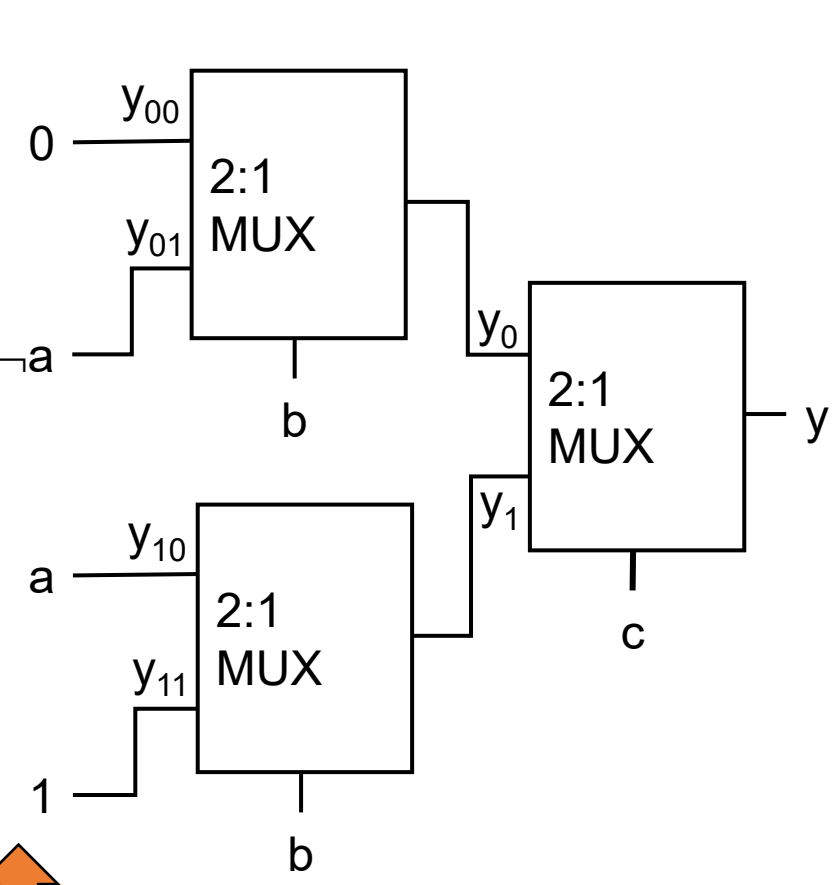
b	a	y_1
0	0	0
0	1	1
1	0	1
1	1	1

a	y_{00}
0	0
1	0

a	y_{01}
0	1
1	0

a	y_{10}
0	0
1	1

a	y_{11}
0	1
1	1



Vereinfachung spart 3. Stufe aus 4x 2:1 MUX

Demultiplexer

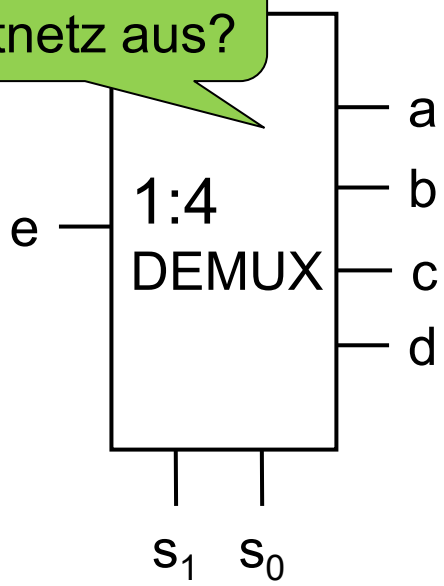
Beschreibung

Eigenschaften:

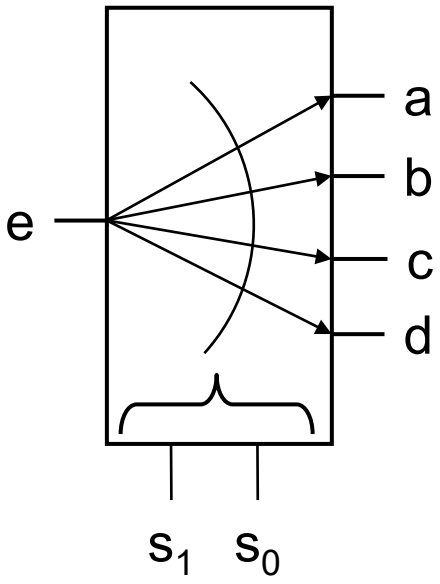
- Umkehrung der Multiplexer-Funktionalität
- Ein Eingang wird abhängig von *n* Steuerleitungen auf einen von 2^n Ausgängen geschaltet

Schaltbild und logisches Verhalten eines 1 auf 4 Demultiplexers:

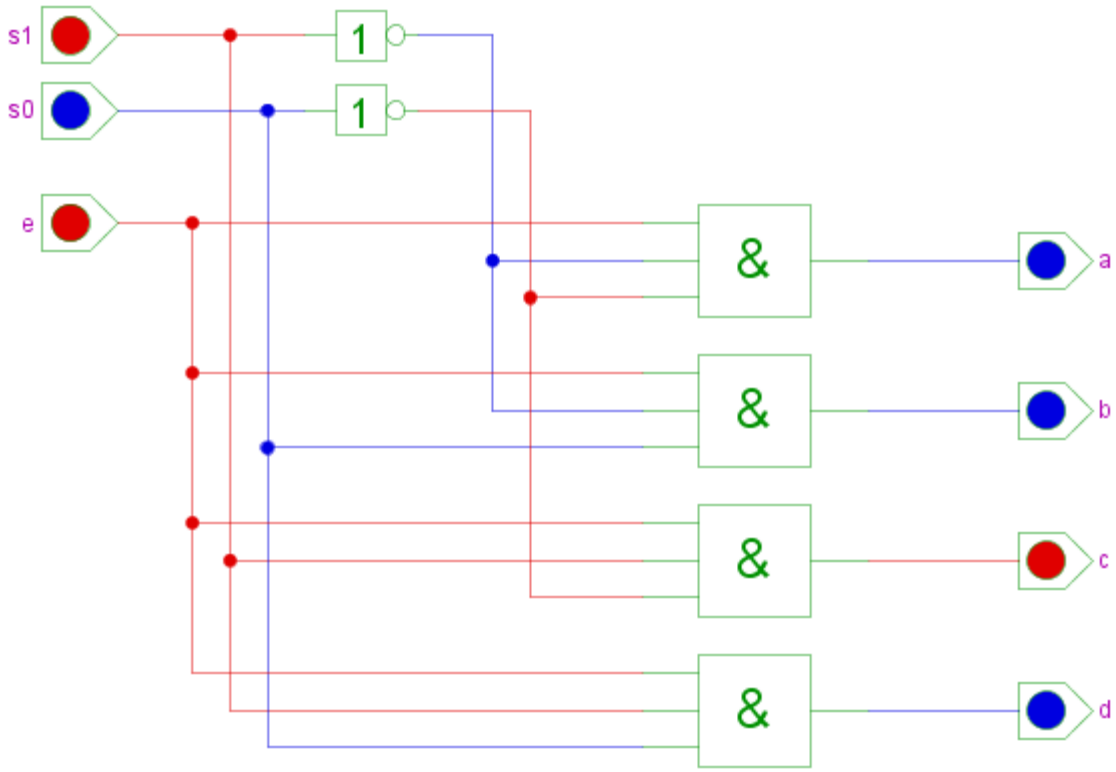
Wie sieht das Schaltnetz aus?



s_1	s_0	a	b	c	d
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e



Realisierung 1:4 Demultiplexer

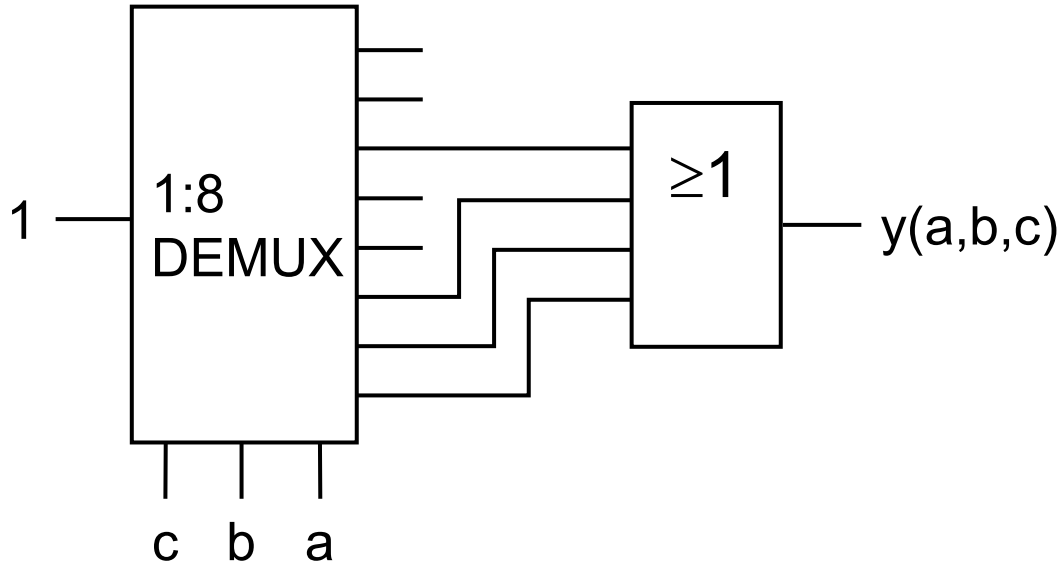


Demultiplexer

Realisierung beliebiger Schaltfunktionen

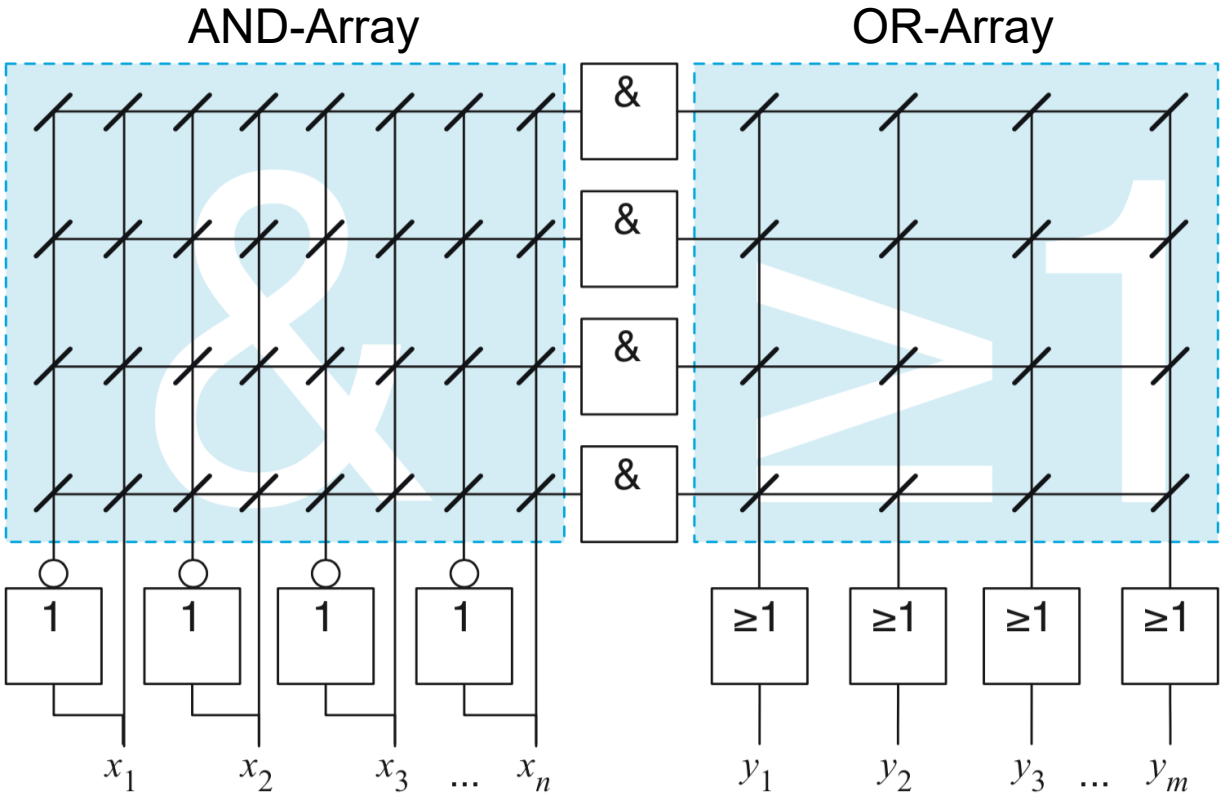
- Setze den Eingang des Demultiplexers auf e=1
- Der Demultiplexer erzeugt alle 2^n Minterme seiner n Steuerleitungen
- Ein ODER-Gatter erzeugt die Einsmenge der Funktion

c	b	a	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Programmierbare Logik

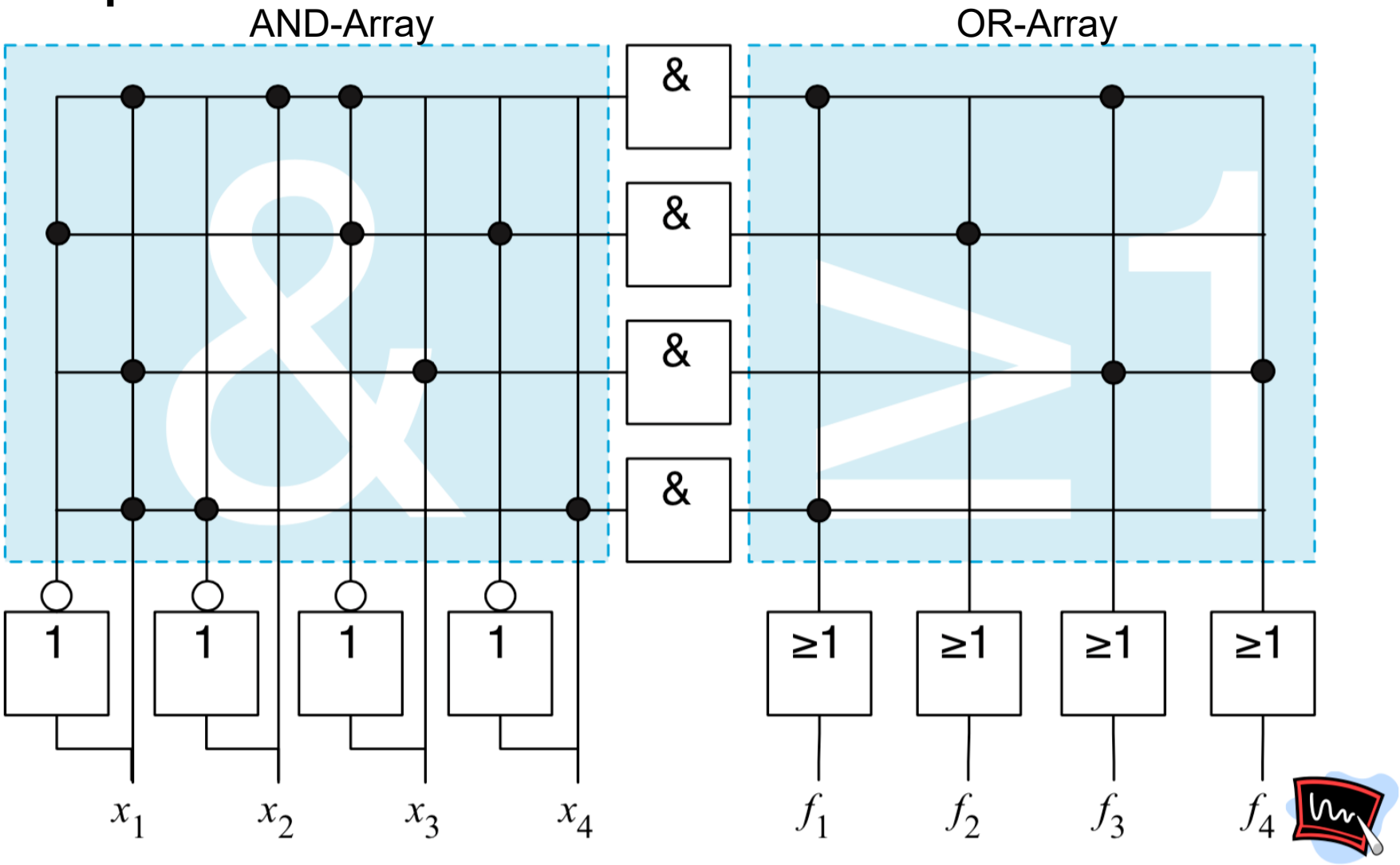
Prinzip



- Oberbegriff: PLD Programmable Logic Device
- PLA (Programmable Logic Array): AND-Array veränderbar, OR-Array veränderbar
- PAL (Programmable Array Logic): AND-Array veränderbar, OR-Array fest
- PROM (Programmable Read Only Memory): AND-Array fest, OR-Array veränderbar

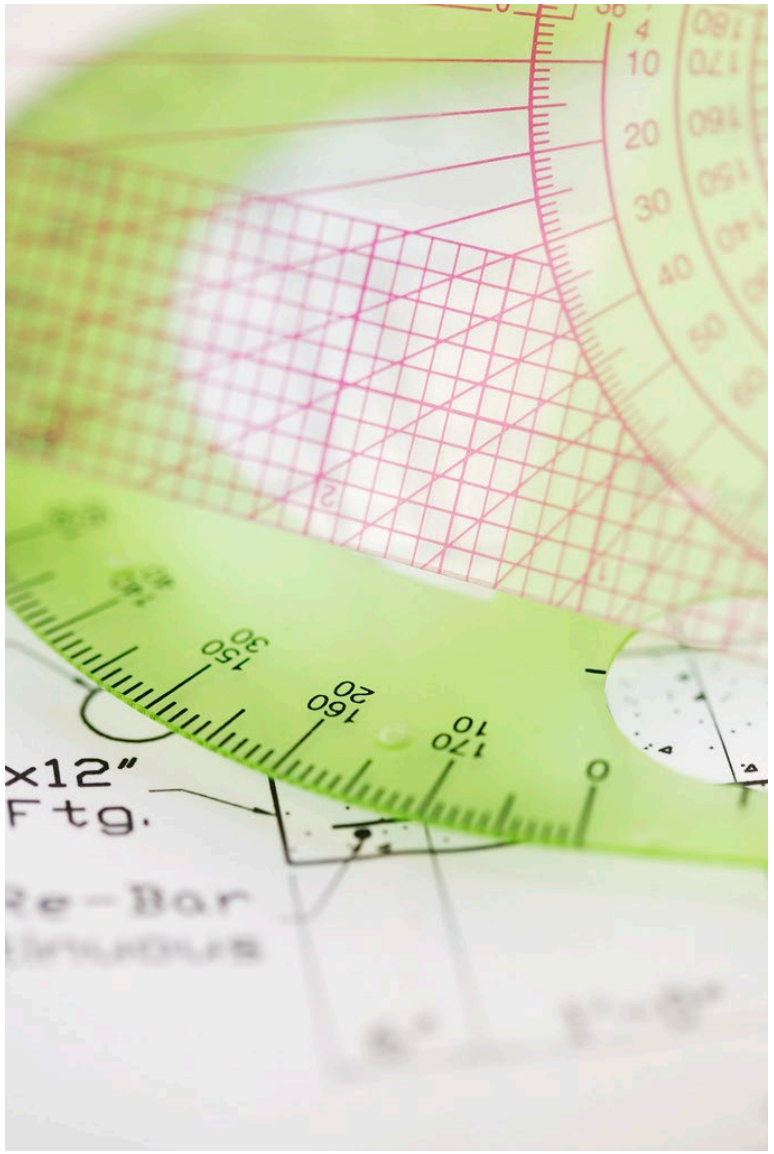
Programmierbare Logik

Beispiel



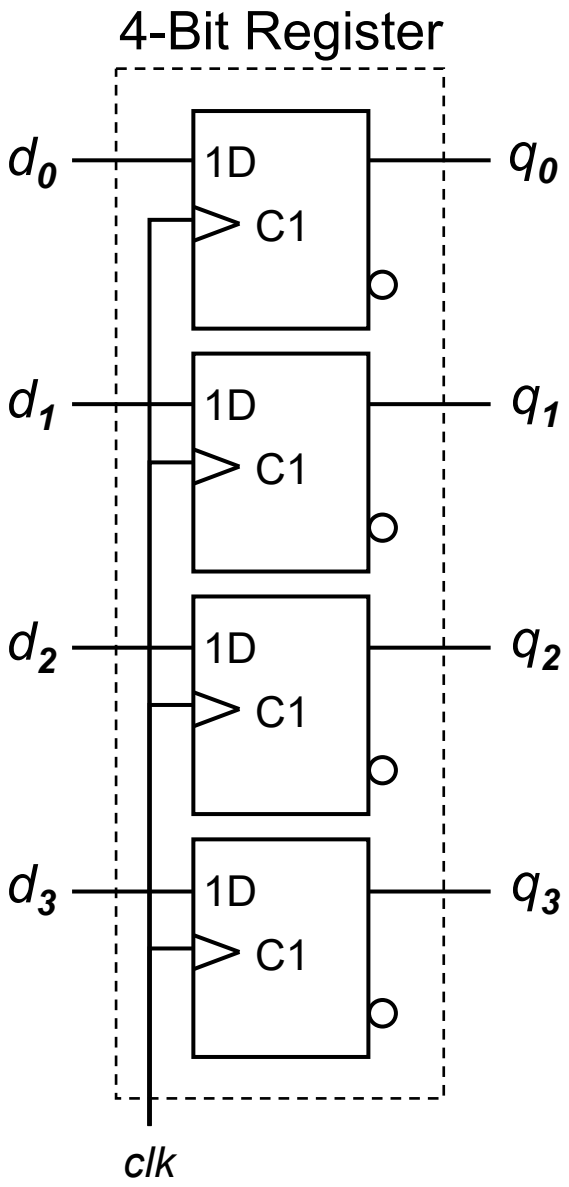
Agenda

- Standard-Schaltnetze
  - Multiplexer
  - Demultiplexer
  - Programmierbare Logik
- Standard-Schaltwerke
  - (Auffang-)Register
  - Schieberegister
  - Zähler



Register

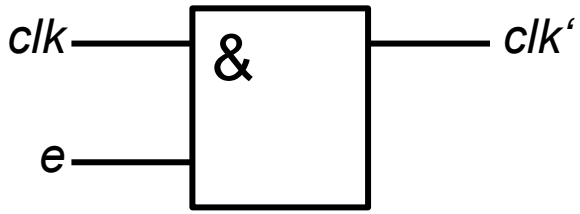
- Aufbau
  - Aneinanderreihung von einzelnen Flipflops
    - "Breite" des Registers = Anzahl der Flipflops
    - Typische Bit-Breiten: 8, 16, 32, 64, 128
  - Alle Flipflops teilen sich dieselbe Taktleitung
- Anwendung: Schneller Speicher in Prozessoren
  - für den Programmierer und direkt ansprechbar
  - für Zwischenergebnisse, Anzahl variiert
- Eingänge und Ausgänge
  - n-fach entsprechend den Einzelflipflops:
    z.B. D, Set oder Reset
  - 1-fach (ein Eingang wirkt auf alle Flipflops des Registers): Takteingang *clk*, Aktivierung (Enable) *e*
- Variationen (später!)
  - Schieberegister
  - Universalregister, Akkumulator



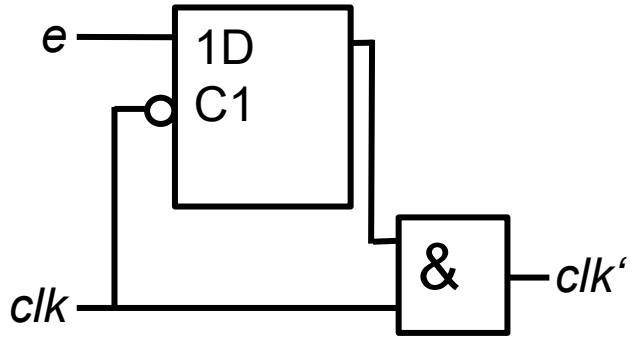
Standard-Schaltwerke

Enable-Logik

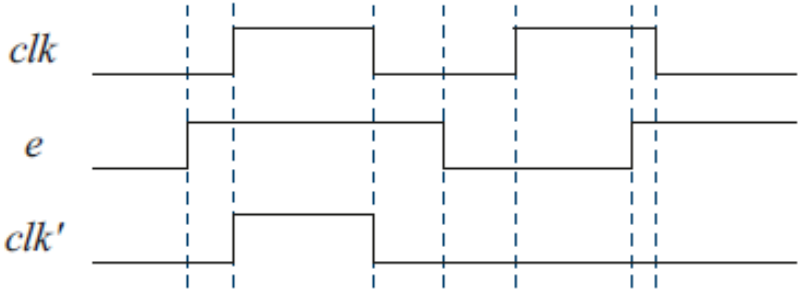
- Für Nutzung in größeren Schaltungen: Enable-Eingang *e* zur Aktivierung
- Schaltung reagiert nur auf Eingangssignale, wenn *e=1* und Takt
- Für Register:
  - Datenwort am Eingang nur in speichern bei positiver Taktflank und *e=1*
  - Zustand halten bei *e=0*
- Realisierung für positiv taktflankengesteuerte Elemente z. B. durch Vorverarbeitung Taktsignal
  - Variante 1 manchmal problematisch
  - Variante 2 mit D-Latch



Variante 1



Variante 2

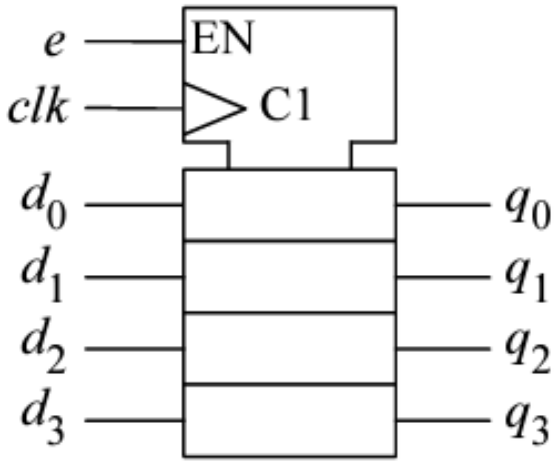


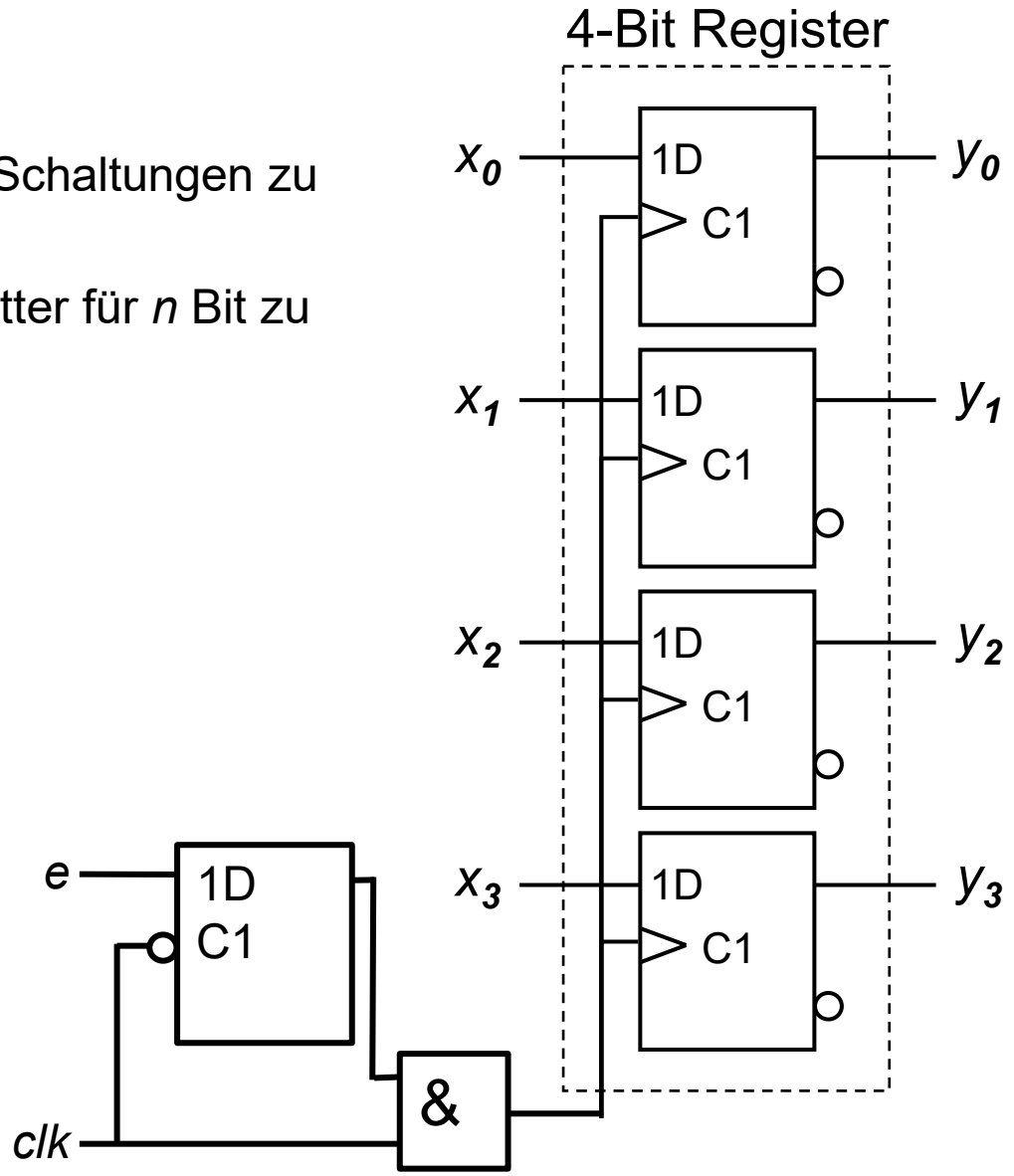
Standard-Schaltwerke

RTL-Darstellung

- Gatter-Darstellung für größere Schaltungen zu kleinteilig
- Zusammenfassen einzelner Gatter für *n* Bit zu einem Schaltsymbol

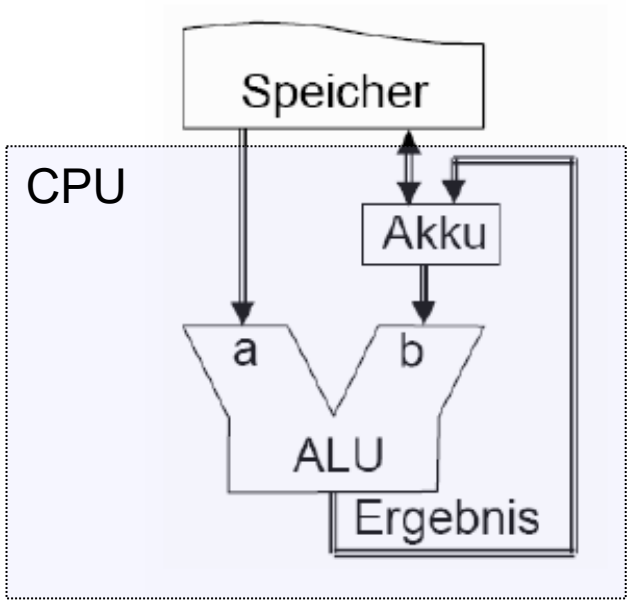
Schaltsymbol





Register in einer einfachen CPU

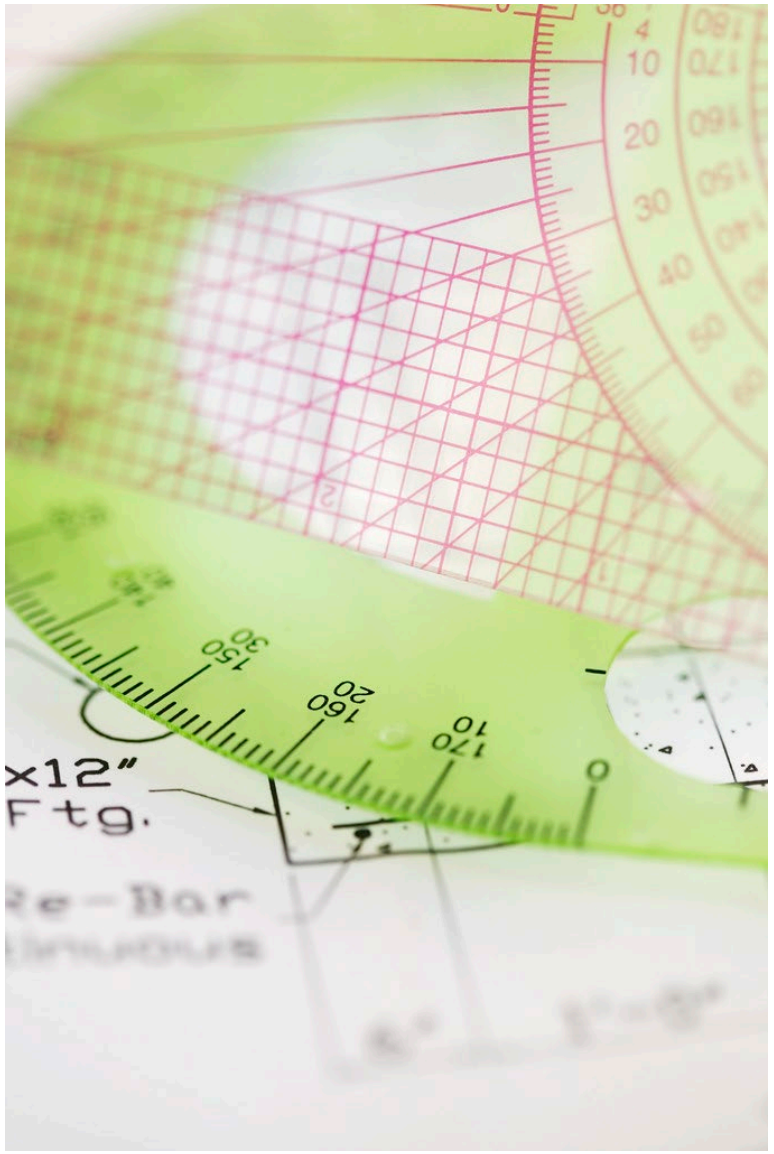
- Mit einen Register und einer Recheneinheit (ALU) lässt sich schon ein wesentlicher Teil einer einfachen CPU aufbauen:



- Der Akkumulator
  - ist ein Register und ist Quelle und Ziel von arithmetischen Operationen.
  - kann aus (externem) Speicher und ALU gespeist werden.

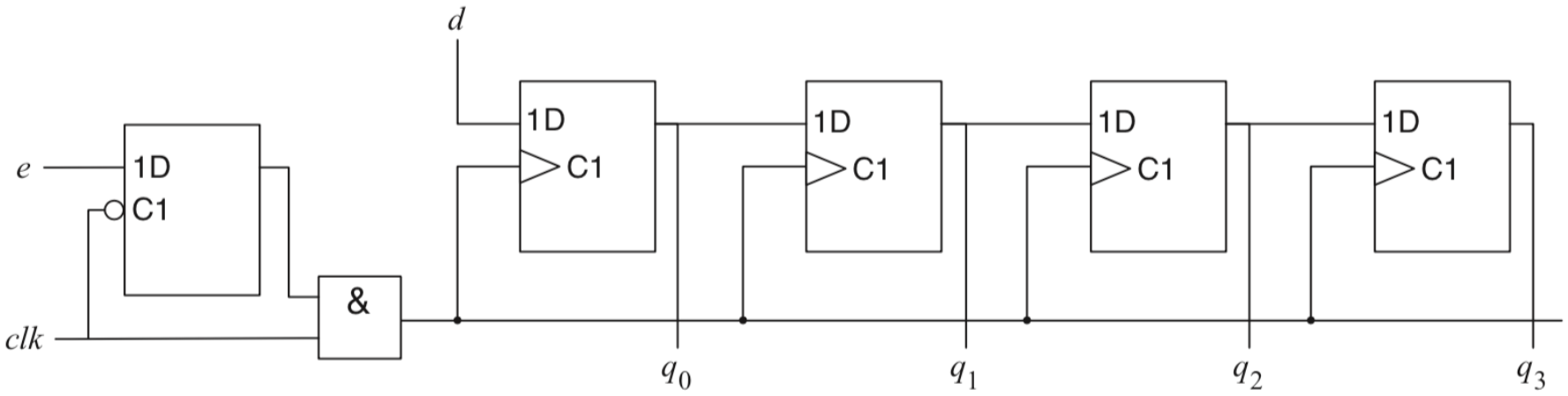
Agenda

- Standard-Schaltnetze
  - Multiplexer
  - Demultiplexer
  - Programmierbare Logik
- Standard-Schaltwerke
  - (Auffang-)Register
  - Schieberegister
  - Zähler

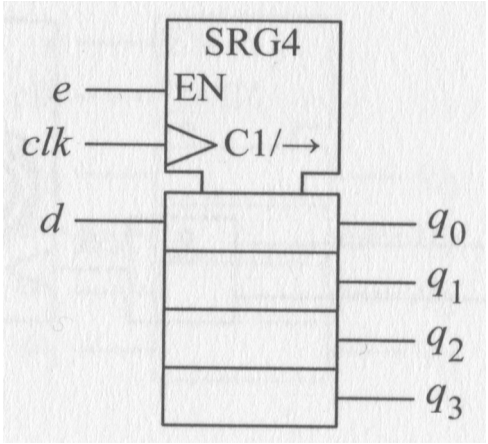


Schieberegister

Aufbau und Funktion

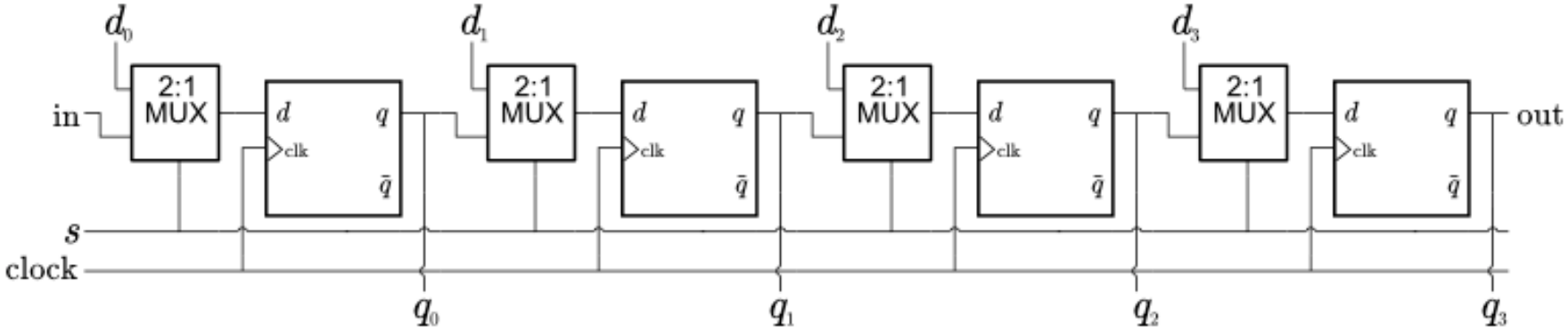


- Ein *n*-Bit Schieberegister hat einen Eingang *d* und *n* parallele Ausgänge $q_0, q_1, \ldots, q_{n-1}$
- Mit jedem Takt wird das Eingangssignal d nach rechts weitergeschoben:
  $d \Rightarrow q_0,\ q_0 \Rightarrow q_1,\ \ldots,\ q_{n-2} \Rightarrow q_{n-1}$
- Realisierung z.B. in Serie geschaltete D-Flipflops
- Synchrone Taktung



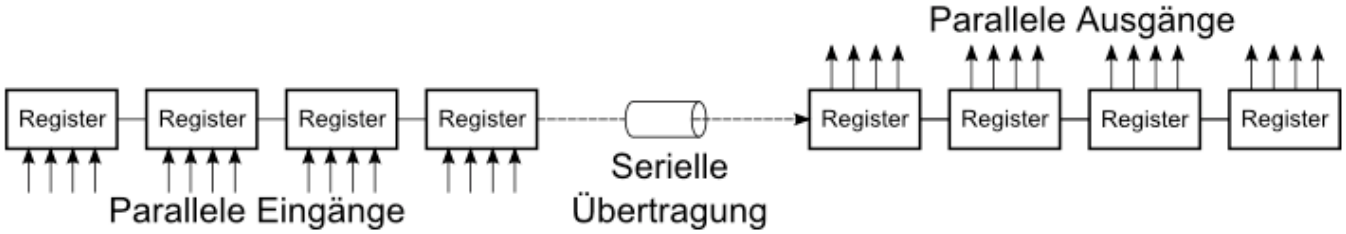
Schieberegister

Erweiterung



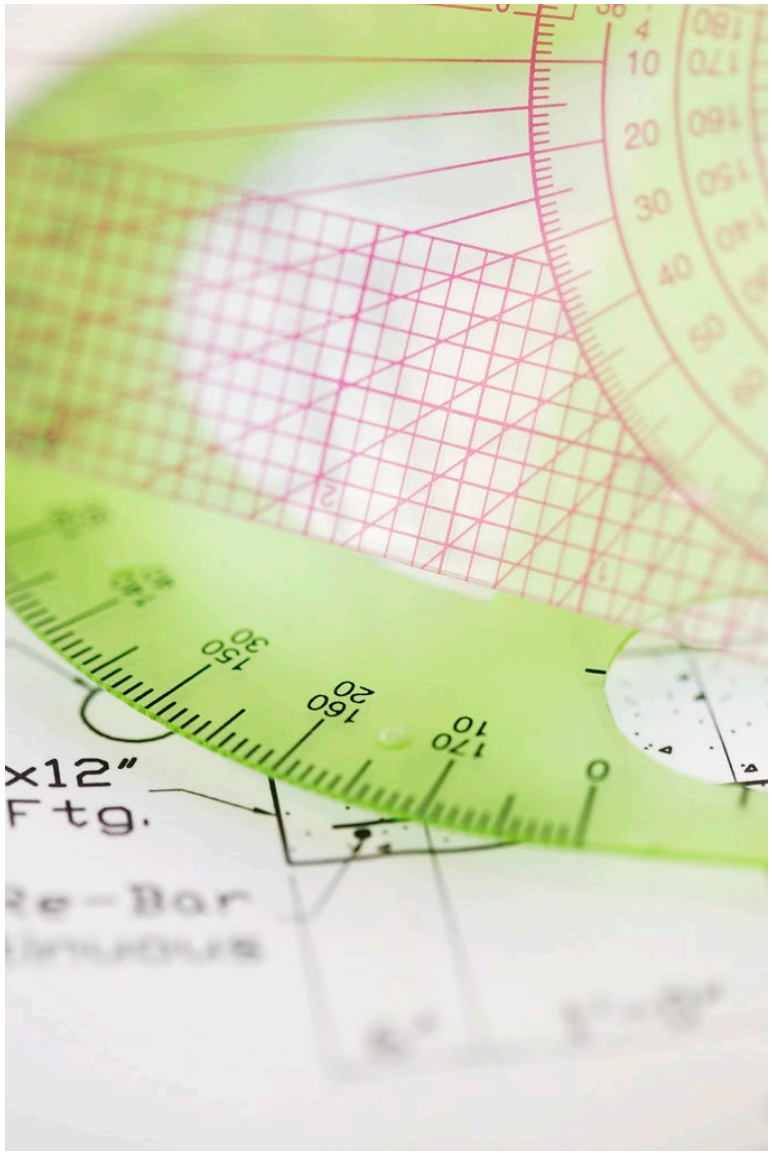
- Multiplexer erlauben Umschalten zwischen zwei Betriebsarten:
  1. Schiebeoperation mit Lesen von *in* (wie Standard-Schieberegister)
  2. Paralleles Laden von Daten (zum späteren Herausschieben nach *out*)

- Anwendung: Serielle Datenübertragung



Agenda

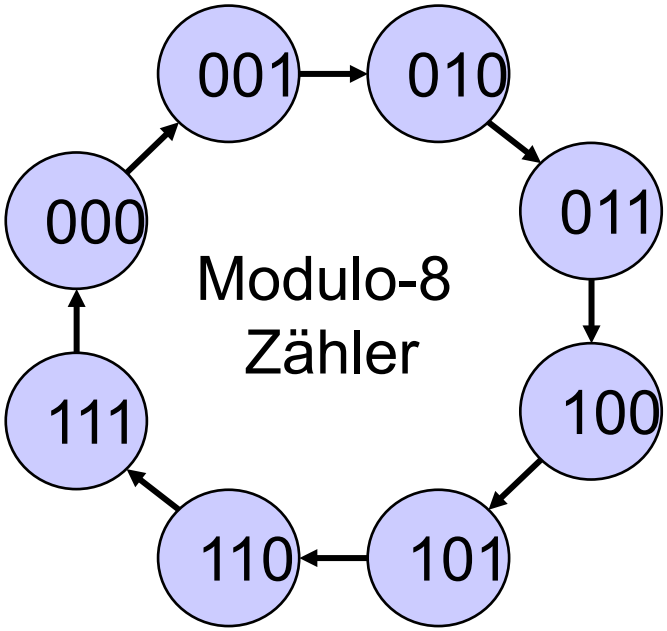
- Standard-Schaltnetze
  - Multiplexer
  - Demultiplexer
  - Programmierbare Logik
- Standard-Schaltwerke
  - (Auffang-)Register
  - Schieberegister
  - Zähler



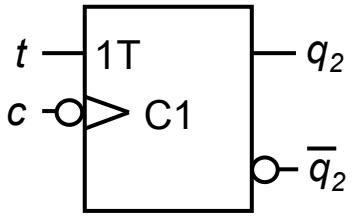
Zähler

Funktion

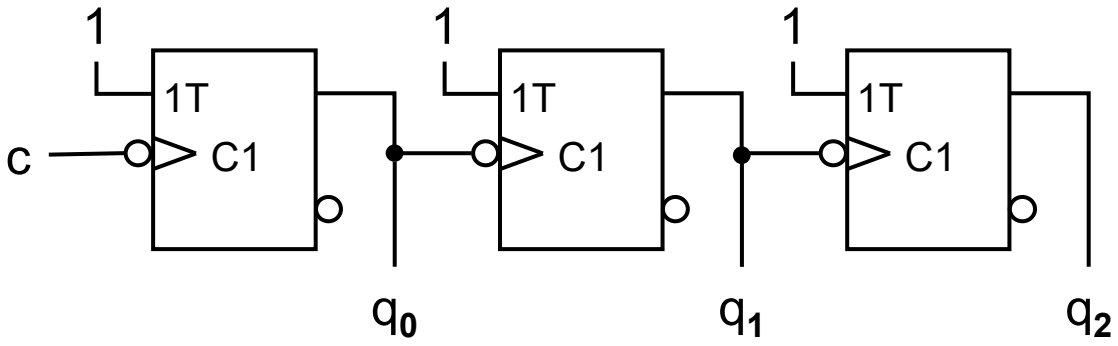
- Ausgänge $q_n,\ldots q_0$ (interpretiert als Binärzahl)
- In jedem Takt wird die Ausgabe inkrementiert
- Zusätzliche Eingänge: Set, Reset, Vorwärts- und Rückwärtsmodus
- Anwendung: Abzählen von Impulsen, kontinuierliche Adressierung

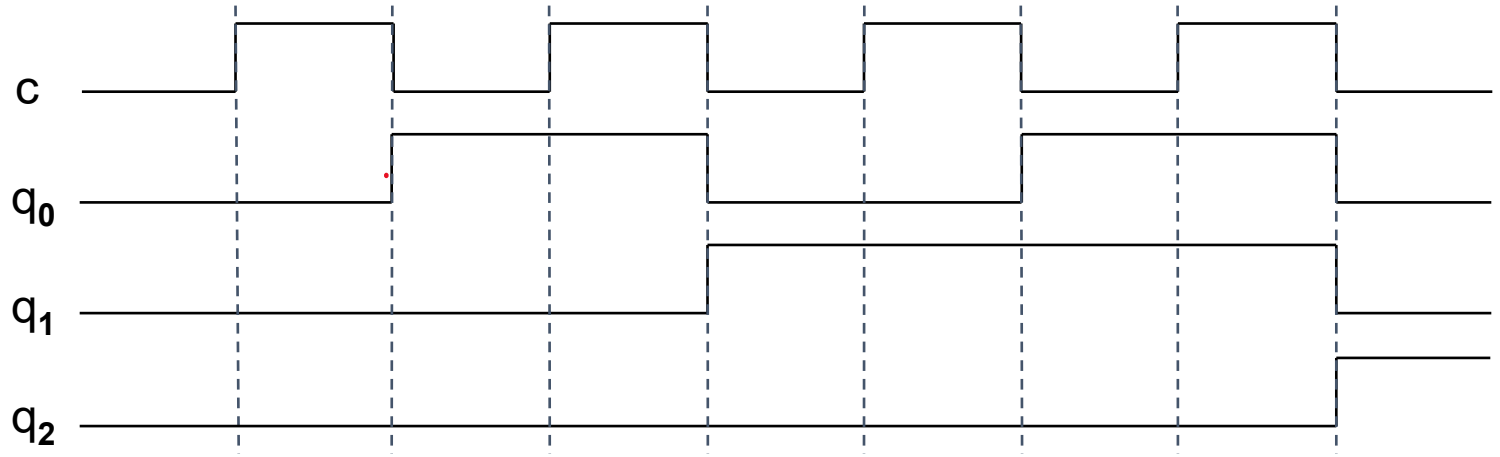


Asynchroner Zähler mit T(oggle)-Flipflops



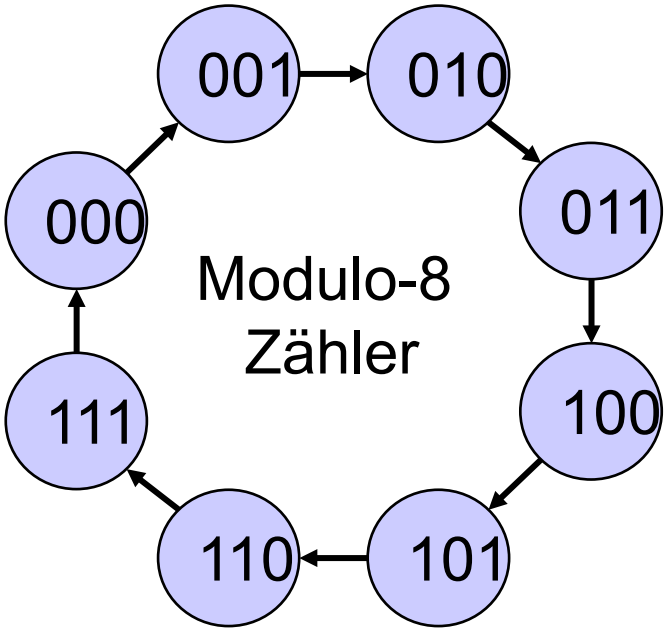
c	t	q_{n+1}
0/1/↑	-	q_n
↓	0	q_n
↓	1	$\neg q_n$





Synchroner Zähler

- Ausgänge $q_n,\ldots q_0$ (interpretiert als Binärzahl)
- In jedem Takt wird die Ausgabe inkrementiert
- Zusätzliche Eingänge (optional): Set, Reset, Vorwärts- und Rückwärtsmodus
- Anwendung: Abzählen von Impulsen, kontinuierliche Adressierung



q_2	q_1	q_0	q_2'	q_1'	q_0'
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Synchroner Zähler

- Übergangsfunktion:

q_2	q_1	q_0	q_2'	q_1'	q_0'
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Zähler

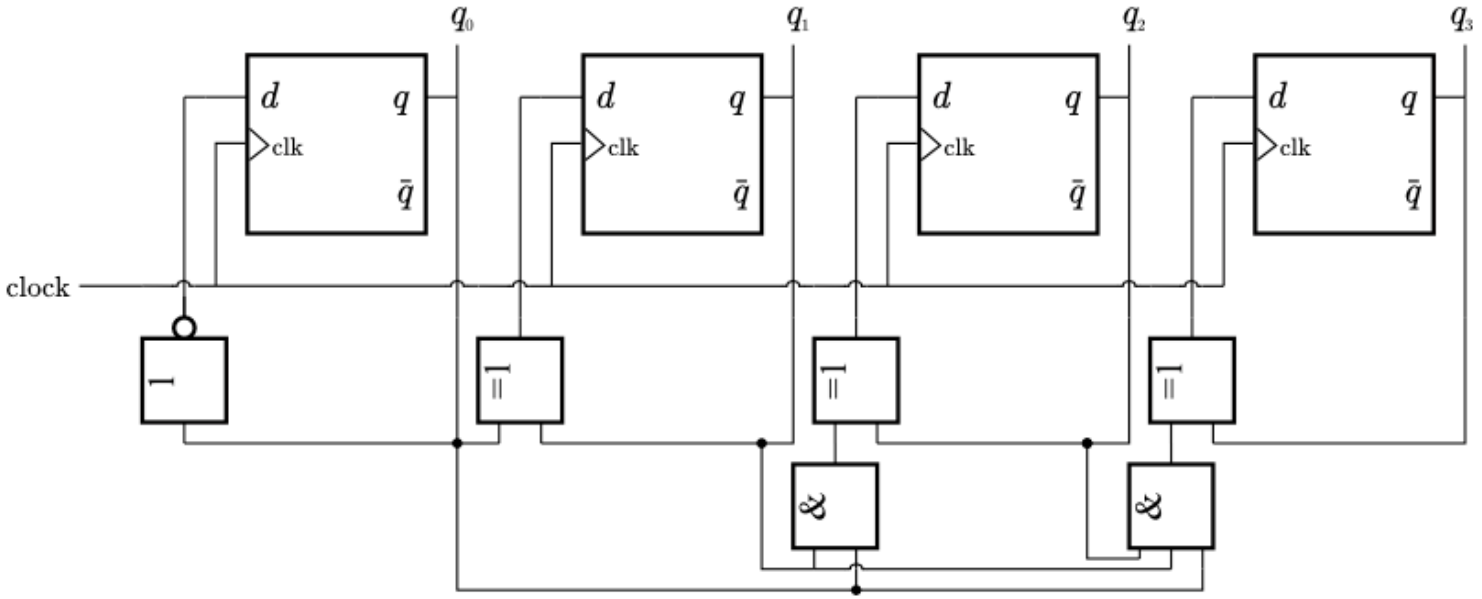
Synchroner Zähler

- Übergangsfunktion:

$$q_0^{t+1} = \neg q_0^t$$
$$q_1^{t+1} = q_1^t \nleftrightarrow q_0^t$$
$$q_2^{t+1} = q_2^t \nleftrightarrow (q_1^t \wedge q_0^t)$$

… und weiter $q_3^{t+1} = q_3^t \nleftrightarrow (q_2^t \wedge q_1^t \wedge q_0^t)$

q_2	q_1	q_0	q_2'	q_1'	q_0'
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0



Agenda

- Standard-Schaltnetze
  - Multiplexer
  - Demultiplexer
  - Programmierbare Logik
- Standard-Schaltwerke
  - (Auffang-)Register
  - Schieberegister
  - Zähler