

KV-Diagramme

Schaltungssynthese

Latches, FlipFlops

# Gliederung

- KV-Minimierung
  - KPI
  - API
  - REPI
- Schaltungssynthese
- Latches
- FlipFlops

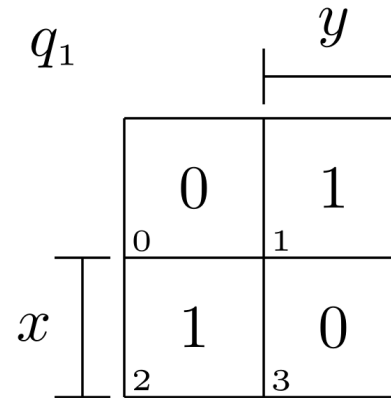
# Wichtige Symbole



- Achtung! Häufige Fehlerquelle!
- Sollte auf euren Formelzettel

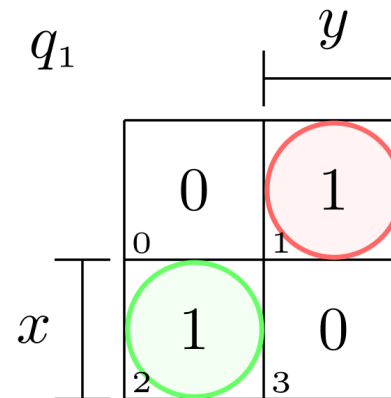
## KV-Diagramme

x	y	$q_1$
0	0	0
0	1	1
1	0	1
1	1	0

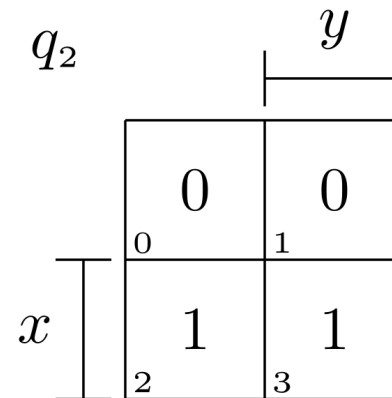


# KV-Diagramme

x	y	$q_1$
0	0	0
0	1	1
1	0	1
1	1	0

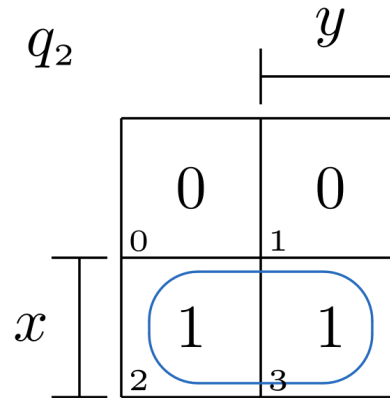


x	y	$q_2$
0	0	0
0	1	0
1	0	1
1	1	1



# KV-Diagramme

x	y	$q_2$
0	0	0
0	1	0
1	0	1
1	1	1



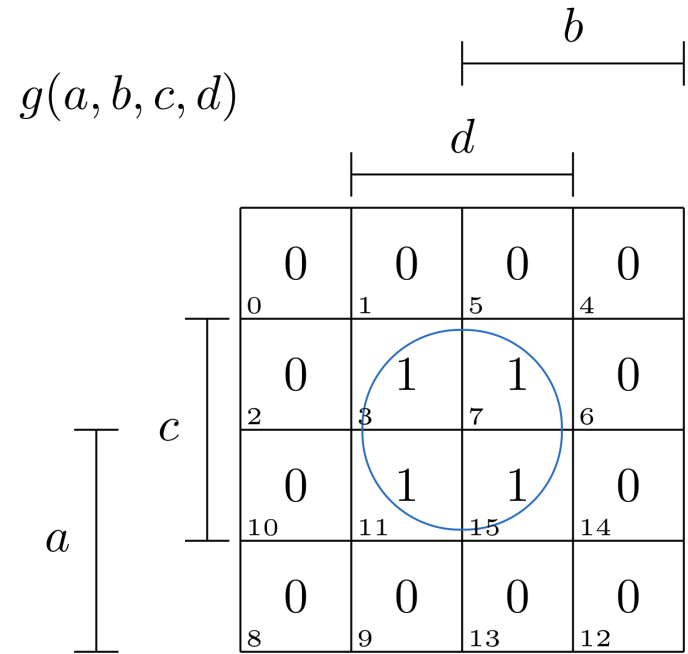


# KV-Diagramme

- **Kernprimimplikant KPI** (essentieller Primterm): Primimplikant, der zur Realisierung einer Funktion unbedingt erforderlich ist. Die Minterme aus denen er entstand, können nicht anders überdeckt werden. Diese werden zur Minimierung zwingend benötigt!
- **Absolut eliminierbarer Primimplikant API**: Primimplikant, dessen Minterme (Maxterme) alle von Kernprimimplikanten überdeckt werden. Diese können zur Minimierung weggelassen werden.
- Alle weiteren Primimplikanten sind **relativ eliminierbare Primimplikanten (REPI)**. Hier muss zur Minimierung eine Auswahl erfolgen!

# KV-Diagramme

- Kernprimimplikant KPI:
  - Primimplikant, der zur Realisierung einer Funktion unbedingt erforderlich ist
  - Die Minterme aus denen er entstand, können nicht anders überdeckt werden
  - KPI werden immer maximal groß gehalten

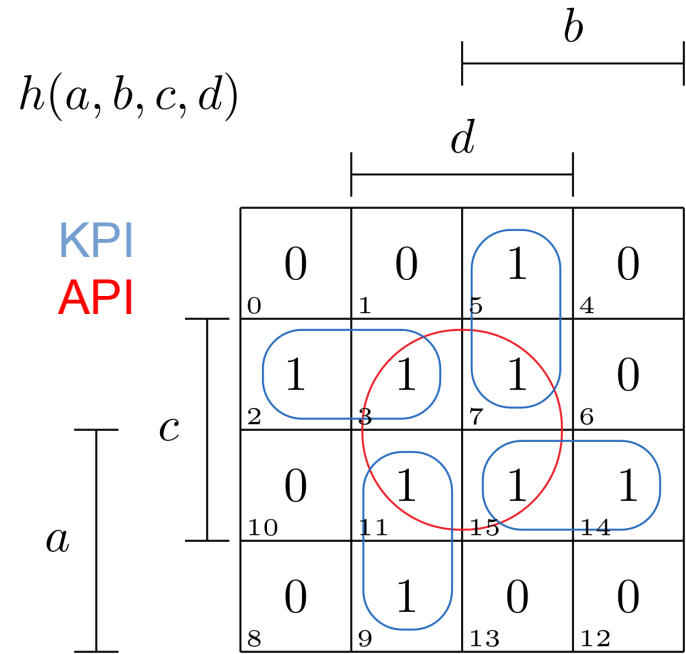


# KV-Diagramme

- Absolut eliminierbarer Primimplikant

API:

- Primimplikant, dessen Minterme (Maxterme) alle von Kernprimimplikanten überdeckt werden
- Können zur Minimierung weggelassen werden



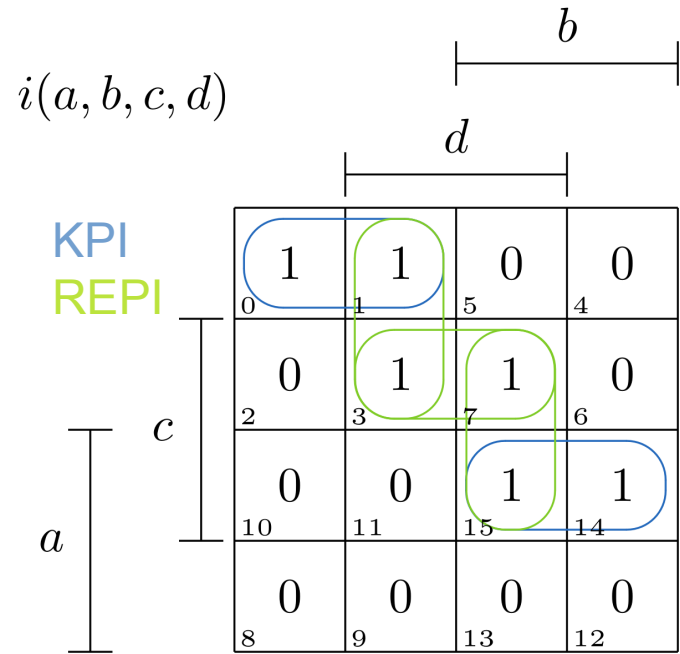
# KV-Diagramme

- Relativ eliminierbarer Primimplikant  
REPI:

- Kann auf mehrere Weisen realisiert werden
- Wenn sich für eine Variante entschieden wird, kann die andere ignoriert werden



- **Ein REPI steht niemals allein**



# Vorlesungsaufgabe

$j(a, b, c, d)$

1	1	0	1
1	1	1	0
1	0	1	1
1	0	0	1

0 1 5 4

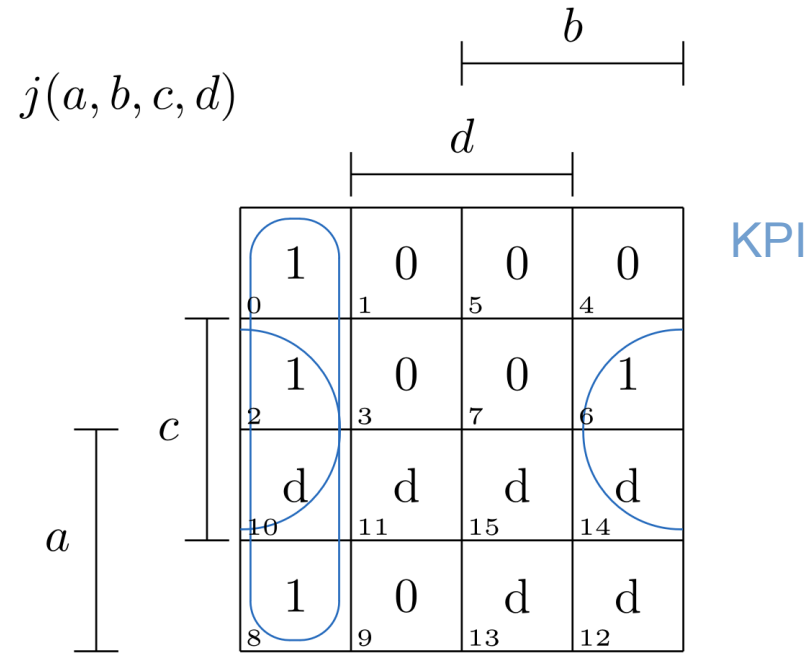
2 3 7 6

10 11 15 14

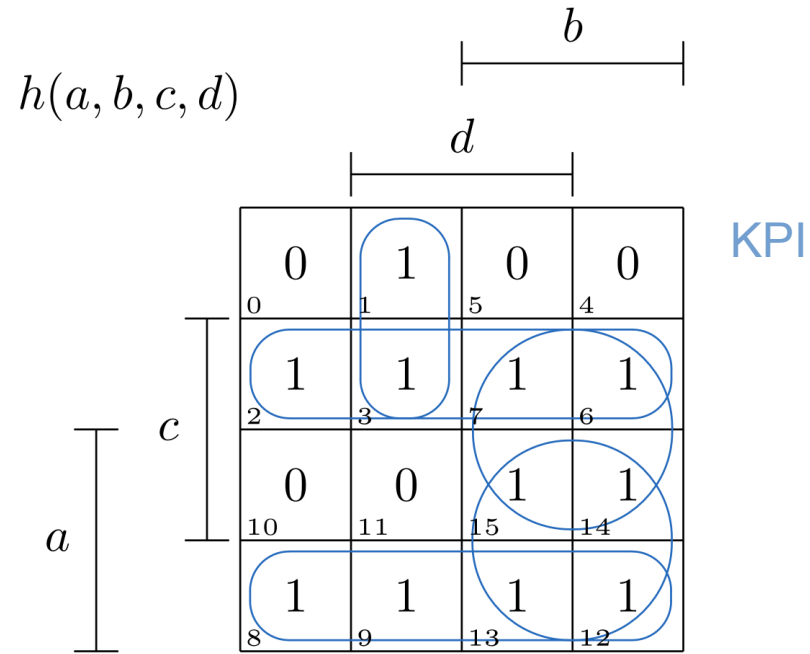
8 9 13 12

$a$   $c$   $d$   $b$

# Finde den Fehler



# Finde den Fehler

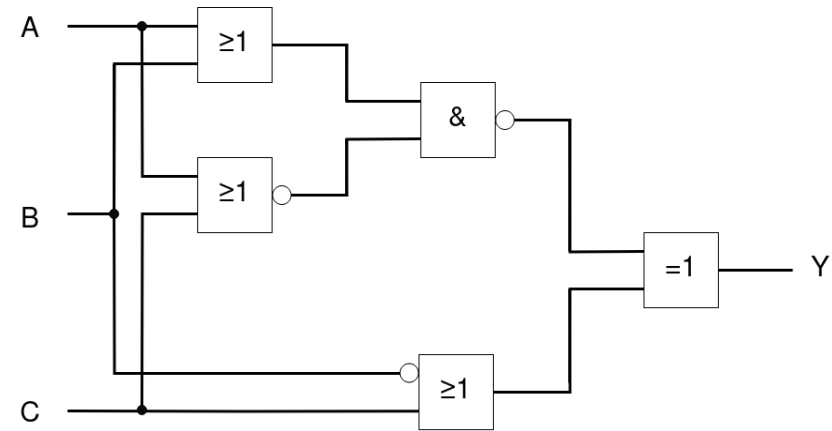


## Schaltungssynthese

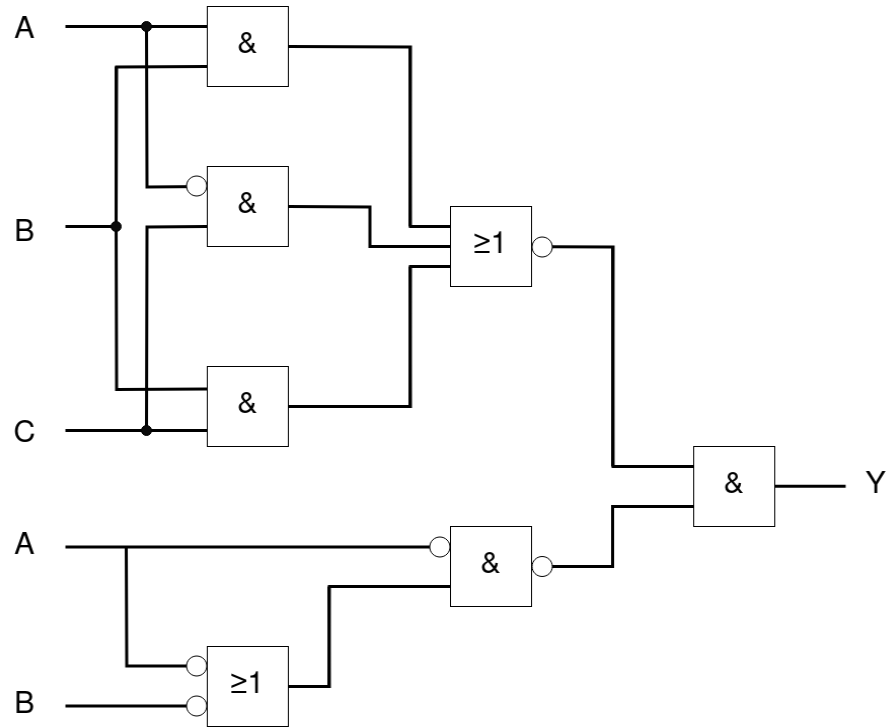


# Schaltungssynthese

- Richtung ist dabei egal, aus Erfahrung ist aber von hinten nach vorne leichter
- Achtet auf Negationen, meist auf der Klausur sehr klein gedruckt
- Teilt die Schaltung in kleinere Teile auf



# Vorlesungsaufgabe



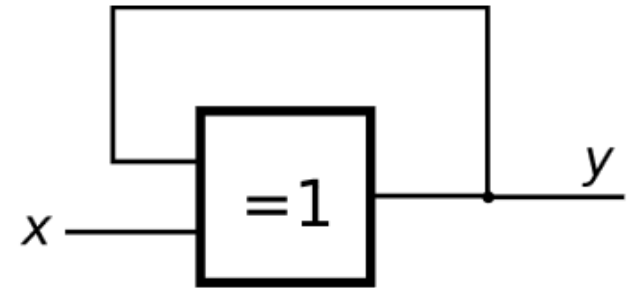
## Vorlesungsaufgabe

$$\overline{AB + \overline{C}(\overline{A} + \overline{B}) + CD}$$

## Latches

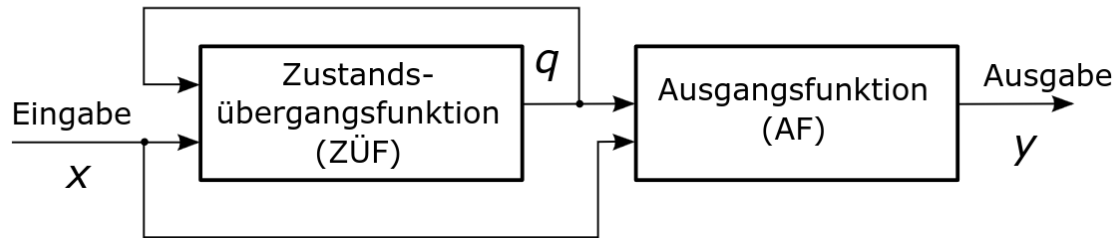
# Latches

- Zustandsübergänge:
  - Rückkopplung in der Schaltung sorgt dafür, dass die Ausgabe vom aktuellen Zustand  $q^t$  abhängt
  - Ausgabe kann nur vom Zustand abhängen (Moore) oder zusätzlich von der Eingabe (Mealy)
  - Aufteilung in Zustandsübergangsfunktion (ZÜF) und Ausgangsfunktionen (AF)

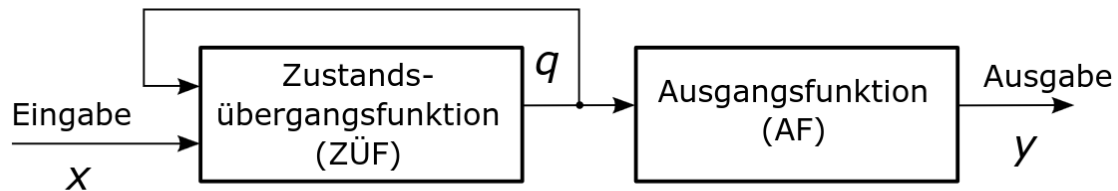


# Latches

## MEALY Automat:



## MOORE Automat:



# Latches

- Zustandsübergänge:
  - Hier nutzen wir  $y^{t+\tau}$  um Zeitpunkt zu kennzeichnen, der eine kleine Zeiteinheit nach dem Umschalten stattfindet
  - Zustände sind stabil, wenn es keinen Zustandswechsel gibt

$y^t$	$x$	$y^{t+\tau}$	
0	0	0	← Ausgang bleibt stabil: $y^{t+\tau} = y^t = 0$
0	1	1	← Ausgang wechselt von 0 auf 1
1	0	1	← Ausgang bleibt stabil: $y^{t+\tau} = y^t = 1$
1	1	0	← Ausgang wechselt von 1 auf 0

# Latches

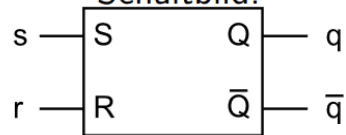
- Wichtige Latches:



## RS-Latch

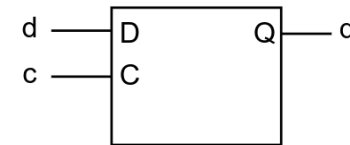
$r$	$s$	$q^{t+\tau}$	
0	0	$q^t$	speichern
0	1	1	setzen
1	0	0	rücksetzen
1	1	–	nicht zulässig

Schaltbild:



## D-Latch

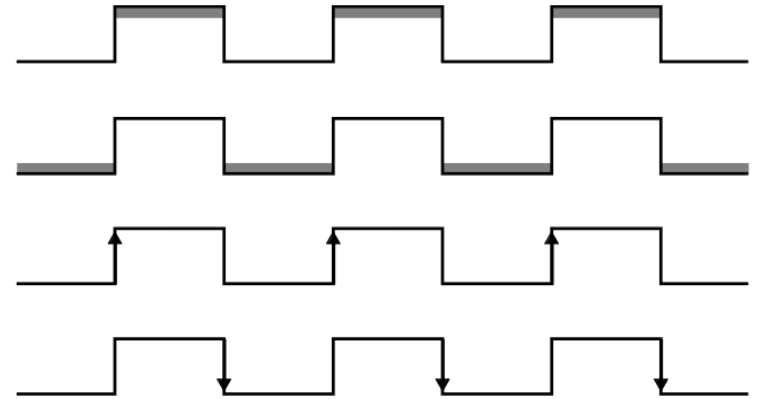
$d$	$q^{t+1}$
0	0
1	1





# Latches

- Taktpegelsteuerung:
  - Latches
  - Wechseln, solange der Takt 1 ist
- Taktflankensteuerung:
  - FlipFlops
  - Wechseln nur an der Vorder- bzw. Rückflanke des Takts

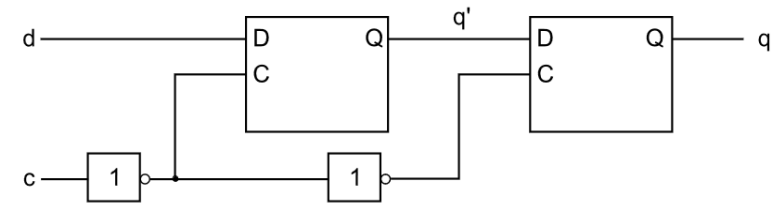
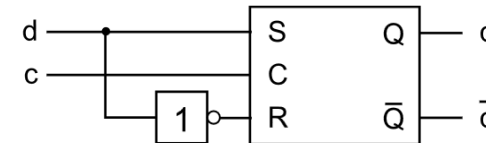
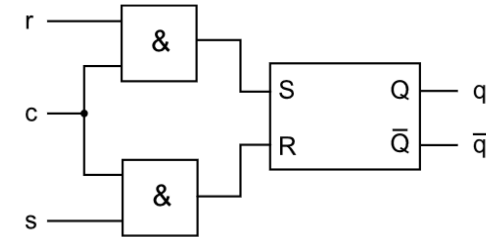


## FlipFlops

# FlipFlops

- Aufbau:

- Aus einem RS-Latch machen wir einen RS-Latch mit Taktpegelsteuerung
- Aus einem RS-Latch mit Takt machen wir einen D-Latch
- Aus zwei D-Latches machen wir einen D-FlipFlop



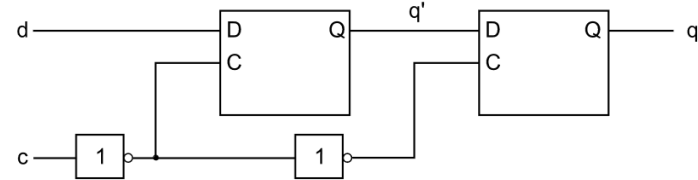
# FlipFlops

Livedemo

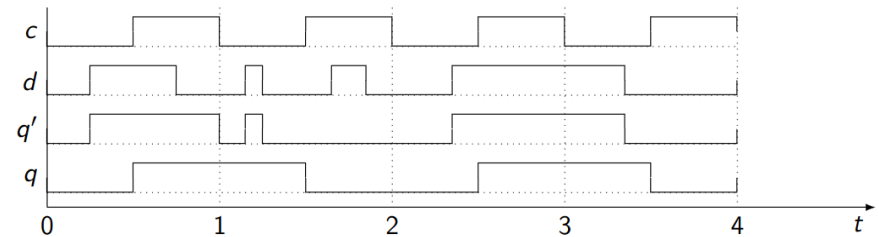
# FlipFlops

- Wichtige Eigenschaften:

- Ausgang wechselt nur an der Taktflanke
- Speichert den aktuell anliegenden Zustand bei jedem Takt
  - einfache Art von Speicher
- Im Gegensatz zum Latch stabil



Beispiel-Timing:



# FlipFlops

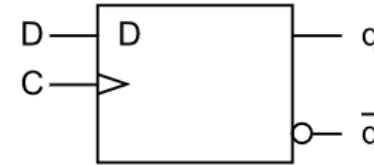
- Wichtige FlipFlops:



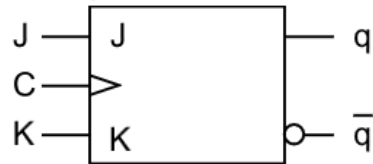
RS-Flipflop (Taktflankensteuerung)



D-Flipflop (Taktflankensteuerung)



JK-Flipflop (Taktflankensteuerung)



T-Flipflop (Taktflankensteuerung)

