

Organisatorisches

- Plagiate, auch aus dem Internet

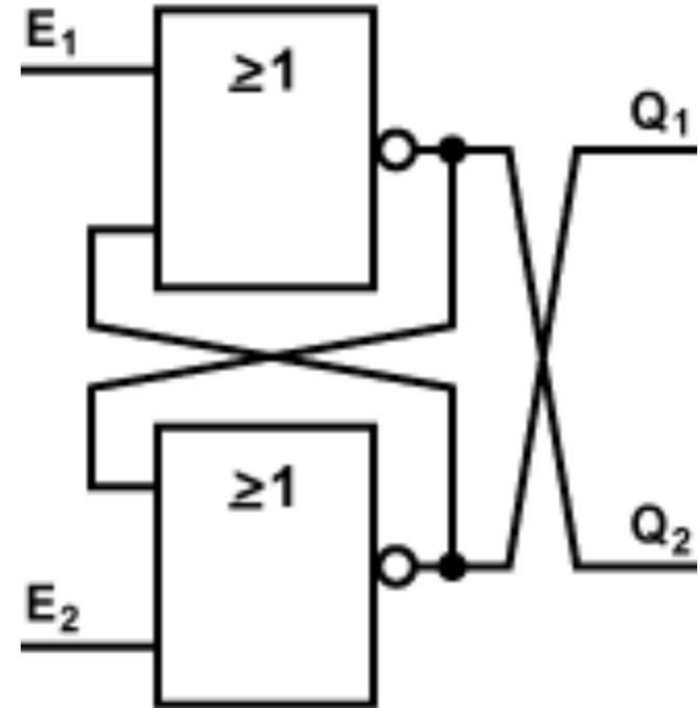
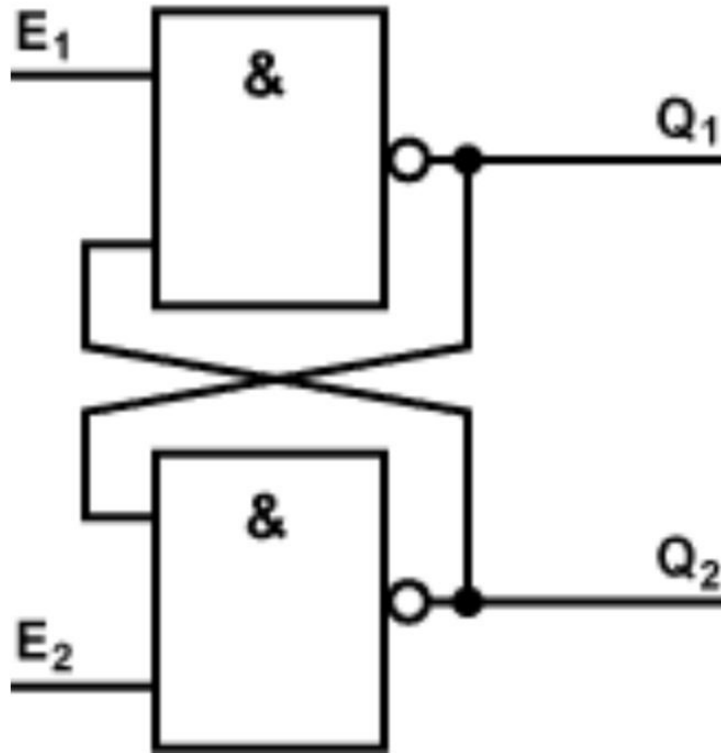
Richtige echte Flipflops

- Speichern einen Zustand (1 oder 0)
- Nutzt Rückkopplung
 - Ausgänge als Eingänge
 - Möglich durch Totzeiten
- Zeit kommt ins Spiel

RS-Flipflops

- Eingang S = Setzen des Bits
- Eingang R = Reset(=0) des Bits
- Ausgang Q und \overline{Q}
 - Keine Negation für \overline{Q} notwendig
 - \overline{Q} wird durch den Aufbau des FF generiert
- Zwei Möglichkeiten: NAND oder NOR

RS-FF Schaltungen



Zustandsübergangsdiagramm

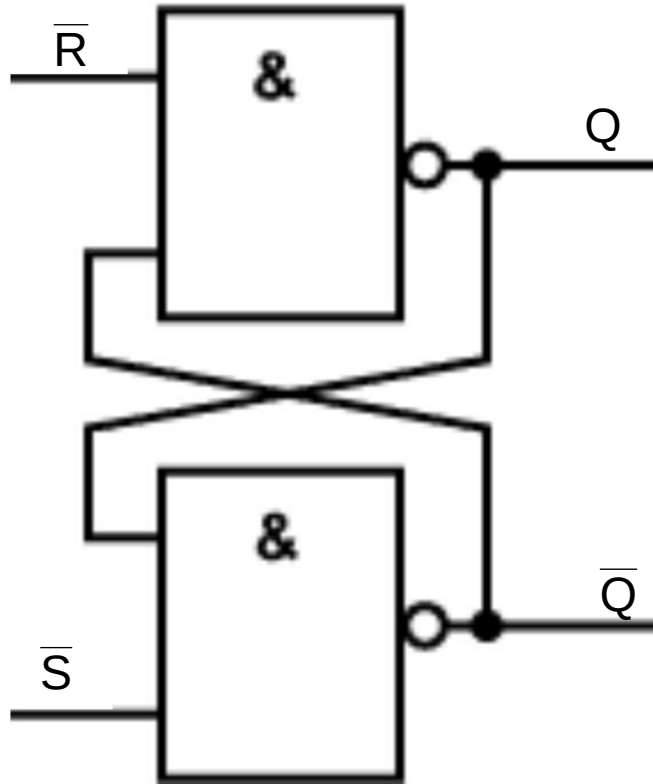
- Benutze Q_n wie eine Eingangsvariable
 - Rückkopplungen erlauben
- Berechnung Q_{n+1} anhand des Schaltnetzes
- Beispiel: RS-Flipflop (Annahme $X_n = \overline{Y}_n$)

R	S	X_n	Y_n	X_{n+1}	Y_{n+1}
0	0	0	0
0	0	0	1

bzw.

R	S	Q_n	\overline{Q}_n	Q_{n+1}	\overline{Q}_{n+1}
0	0	0	0
0	0	1	0

RS-FF Wahrheitstabelle



\overline{S}	\overline{R}	Q_n	\overline{Q}_n	Q_{n+1}	\overline{Q}_{n+1}
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	0	1	0

RS-FF Wahrheitstabelle

\bar{S}	\bar{R}	Q_n	\bar{Q}_n	Q_{n+1}	\bar{Q}_{n+1}	Funktion
0	0	0	1	1	1	
0	0	1	0	1	1	
0	1	0	1	0	1	
0	1	1	0	0	1	
1	0	0	1	1	0	
1	0	1	0	1	0	
1	1	0	1	0	1	
1	1	1	0	1	0	

NAND vs. NOR

NAND

\overline{S}	\overline{R}	Q_1	Q_2	Funktion
0	0	1	1	Verboten
0	1	0	1	Zurücksetzen
1	0	1	0	Setzen
1	1	*	*	Speichern

NOR

S	R	Q_1	Q_2	Funktion
0	0	*	*	Speichern
0	1	1	0	Setzen
1	0	0	1	Zurücksetzen
1	1	1	1	Verboten

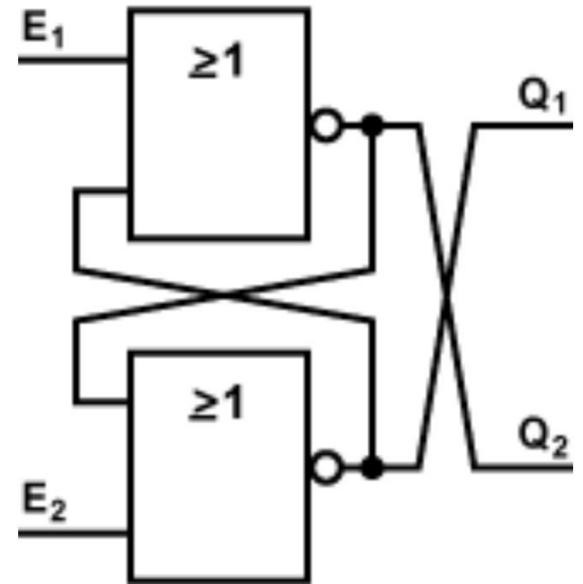
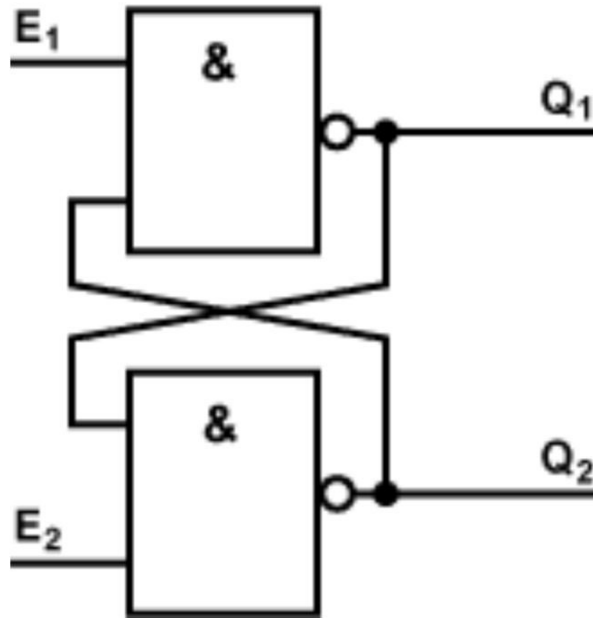
FF – undefinierte Zustände

- Undefiniert, wenn $Q_n = \overline{Q}_{n+1}$ oder $Q_n = \overline{Q}_n$
 - Kann initial vorliegen
 - Bei RS-FF im „verbotenen“ Zustand
- Q wechselt (sehr) schnell hin und her
- Verhindern/Behebung: Anlegen eines „stabilen“ Zustandes

Speichern im Takt(-zustand)

- Zusätzliches Taktsignal C
- Eingänge zusätzlich vom Takt abhängig ($C = 1$)
=> Konjunktion
- Vorteil:
 - Halber Takt keine ungewollten Änderungen
- Nachteile
 - Verbotener Zustand möglich
 - Falscher Zustand für halben Takt beibehalten

Clock „einbauen“

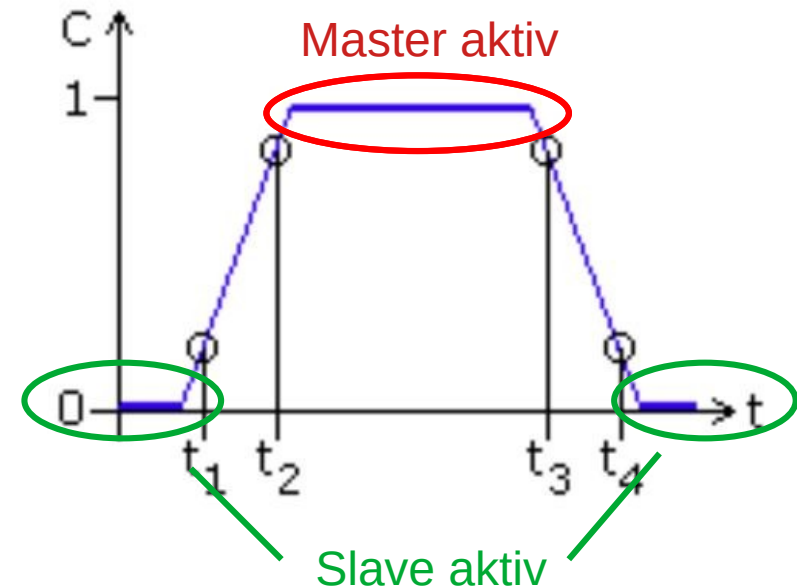


Problem: Zustandssteuerung

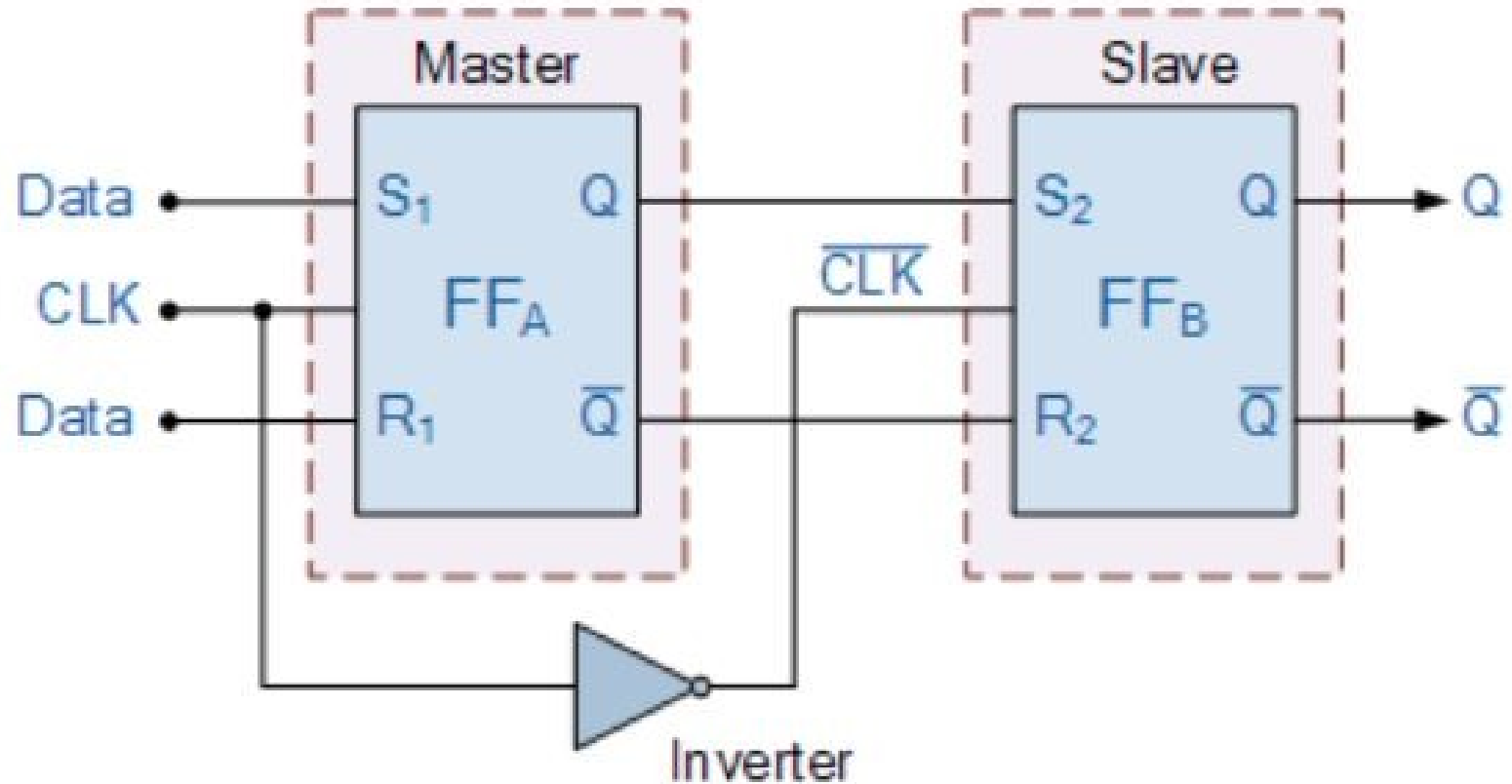
- Hasardfehler werden nicht komplett verhindert
 - Hasardfehler bei schaltendem Taktzustand
- Falscher Zustand wird einen Taktzustand beibehalten

Master-Slave-FF

- Einlesen und Ausgabe separieren
- 2 Flipflops in Reihe Master & Slave
 - Slave nutzt invertierten Takt
- Master von Funktion f abhängig
 - f kann hasardbehaftet sein
- Slave leitet das Mastersignal weiter
- FF nicht gleichzeitig aktiv

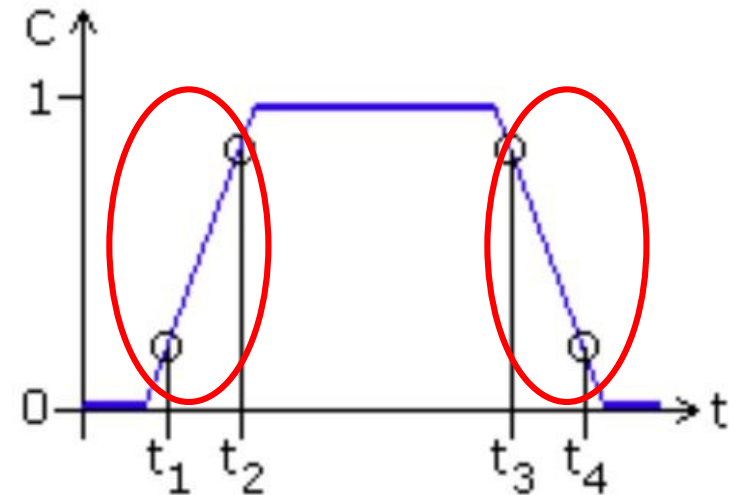


Master-Slave-FF

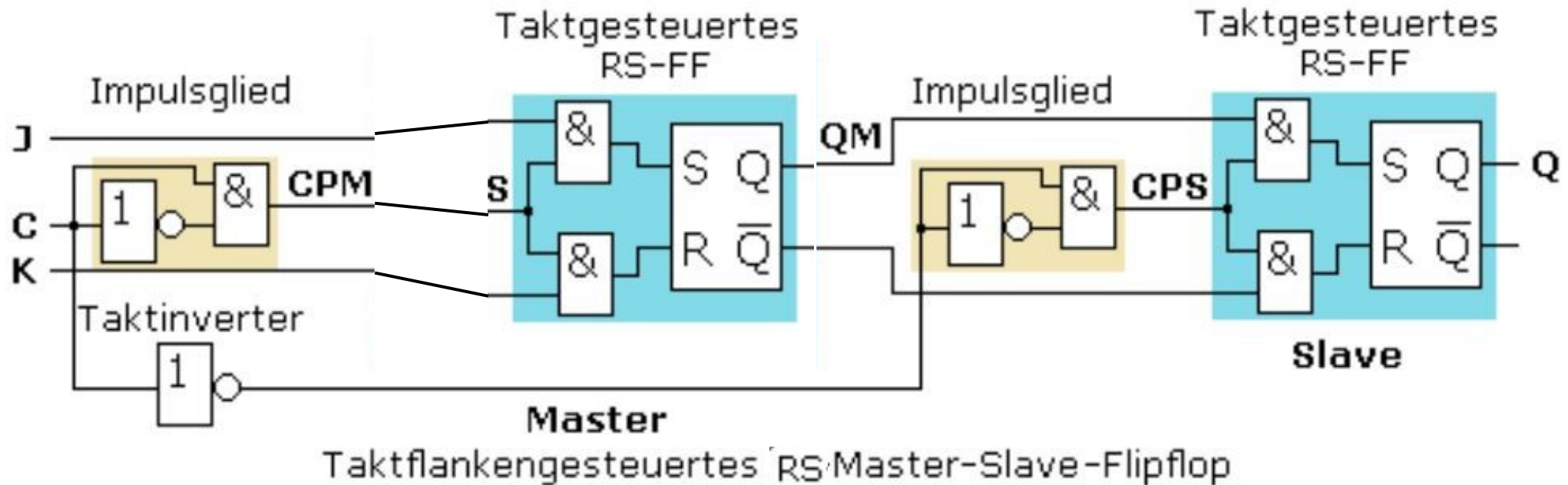


Kürzer schalten

- Schalten an einem Punkt?
- Nutze Taktflanken
 - Nicht ideal, aber gut genug
- An Taktflanken gilt $C = \bar{C}$
- Schreit nach Konjunktion



Taktflankengesteuerter RS-MS-FF



Weitere Flipflops

- Jump-Kill-Flipflop
 - Sicherer als RS (nächstes Mal mehr)
- Delay-Flipflop
 - Verzögert ein Signal
- Toggle-Flipflop
 - Wechselt im Takt zwischen 1 und 0

Aufgabe

Entwickeln Sie einen RS-Flipflop nach den folgenden Spezifikationen:

- Eingangssignale sind (a, b) , Ausgangssignale (x, y) mit $x = \bar{y}$
- Nur y wird zum Speichern in die Schaltung rückgeführt
- $(a, b) = (0, 0)$ ist ein illegaler Zustand
- $(a, b) = (0, 1)$ setzt y auf 0
- $(a, b) = (1, 0)$ setzt y auf 1
- $(a, b) = (1, 1)$ speichert den aktuellen Wert von y

a) Erstellen Sie eine Überganstabelle für den Flipflop nach folgender Vorlage:

a	b	y_n	y_{n+1}	x_{n+1}
\dots				

- b) Leiten Sie aus der Überganstabelle die DMF für y_{n+1} ab.
- c) Entwerfen Sie eine Schaltung für den Flipflop.

FF-Typen erkennen

- Auf Eingänge schauen
- Taktstrg. schält bei positiver Taktzustand/-flanke
 - Eingang kann negiert sein
- Im Skript gibt es eine interessante Folie;)

Wichtig: FF

- Zustandsfolgetabelle Reihenfolge wie in Wertetabelle
 - Nicht $Z_1 \rightarrow Z_2 \rightarrow \dots$
 - Sonst funktioniert Z-Trick nicht mehr
- Unterschied NAND- und NOR-RS-FF
- Normales NAND-RS-FF Eingänge negiert
- Verbotene Zustände
- Gute Quelle: [Elektronik Kompendium](#)