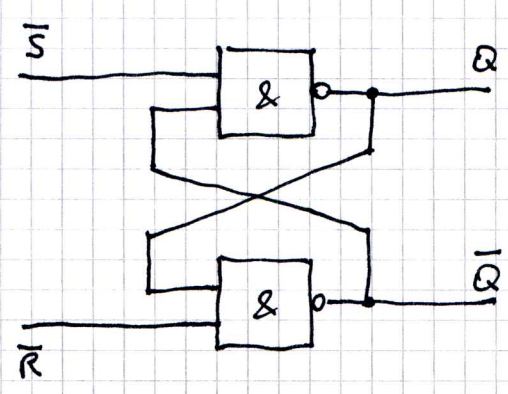


C1

FlipFlop - eine bistabile Schaltung

Alle Speicherglieder enthalten im Kern eine FlipFlop-Schaltung, deren wesentliches Merkmal eine Rückkopplung der Ausgänge auf die Eingänge ist:



\bar{S} bedeutet : Der Setz-Eingang ist aktiv, falls $S = \text{low}$
 \bar{R} " : Der Reset-Eingang ist aktiv, falls $R = \text{low}$
 \bar{S} und \bar{R} dürfen nicht beide gleichzeitig low gesetzt werden!

Analyse:

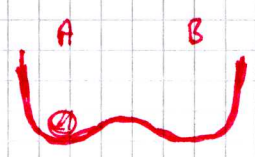
\bar{S}	\bar{R}	Q	\bar{Q}
1	0	0	1
0	1	1	0
1	1	1	0
0	0	1	1

Reset $\rightarrow Q = 0$
 Set $\rightarrow Q = 1$
 stabil, Q & \bar{Q} unverändert (speichern!)

↖ unsinnig, nicht erlaubt, verboten

Das ist also ein primitives RS-FlipFlop, welches immerhin erlaubt, einen 1-bit-Zustand zu speichern oder zu ändern.
 "Die Mutter aller Speicherglieder" ... gerichtet!

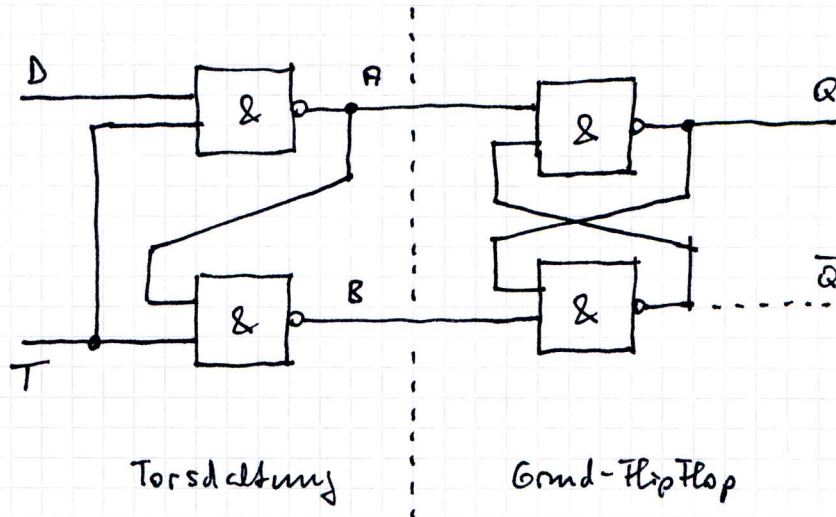
mechanische Analyse:



(C2)

Ein Fangspeicher für 1 bit

Ein schon recht komfortable Speicher für 1 bit erhält man, wenn man das Grund-Flipflop noch mit einer "Torschaltung" versieht:

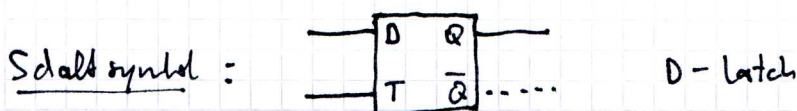


Analyse

- a) falls $T = \text{low} \Rightarrow A = \text{high} \ \& \ B = \text{high}$
 $\Rightarrow Q$ und \bar{Q} behalten ihre Wert nach Ct
 Das nennt man speichern, das Tor ist geschlossen
- b) falls $T = \text{high} \Rightarrow$ Der Dateneingang D ist wirksam:
- $D = \text{high} \Rightarrow A = \text{low} \Rightarrow B = \text{high}$
 $\Rightarrow Q = \text{high} \ \& \ \bar{Q} = \text{low}$ (nach Ct)
 also $Q = D$
 - $D = \text{low} \Rightarrow A = \text{high} \ \text{und} \ B = \text{low} \Rightarrow$
 $\Rightarrow Q = \text{low} \ \& \ \bar{Q} = \text{high}$ (nach Ct)
 also $Q = D$, ebenfalls!

Zusammenfassung:

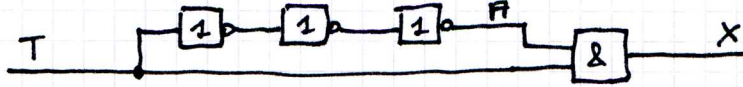
$T = \text{low}$	\rightarrow speichern	, das Tor ist zu
$T = \text{high}$	$\rightarrow Q = D$, Q folgt D, das Tor ist offen



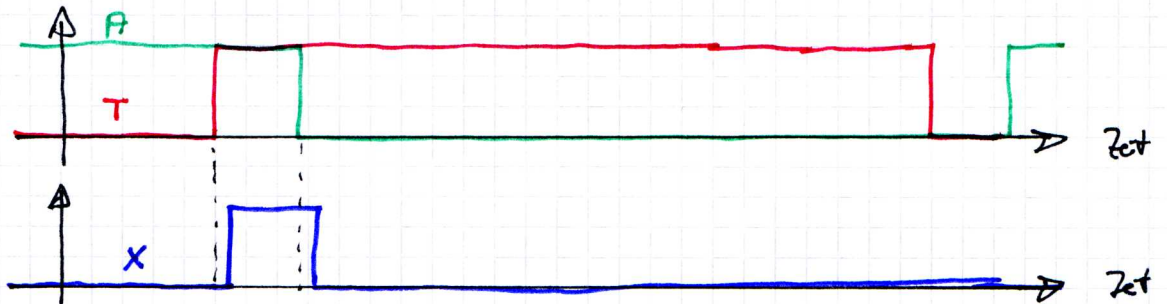
(C3)

Besseres Timing durch Flankensteuerung

Betrachten wir die folgende Teilschaltung:



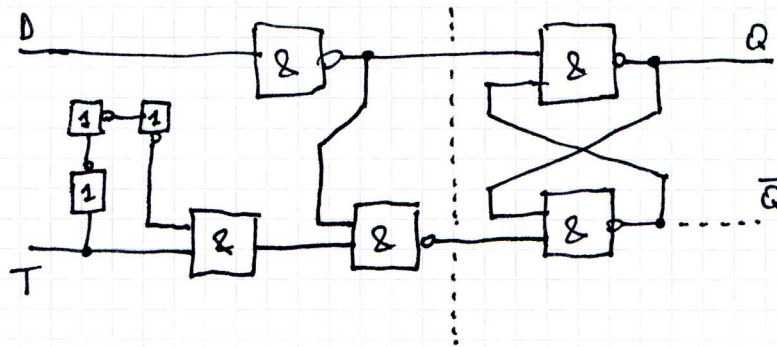
Berücksichtigen wir, dass jeder der 3 Inverter eine Schaltzeit von $t_{inv} = 1 \text{ ns}$ benötigt, so ergibt sich der folgende zeitliche Verlauf für die drei Signale T, A und X:



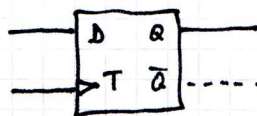
Wenn kein Anstieg der Takts um low auf high (\uparrow) entsteht ein sehr kurzer Puls. *)

Benutzt man diese Puls, um das Tor des Speichers zu öffnen, so ist der Zeitpunkt der Datenübernahme sehr stark festgelegt!

Wir erhalten so einen flankengetriebenen Datenspeicher:



Schalt symbol:



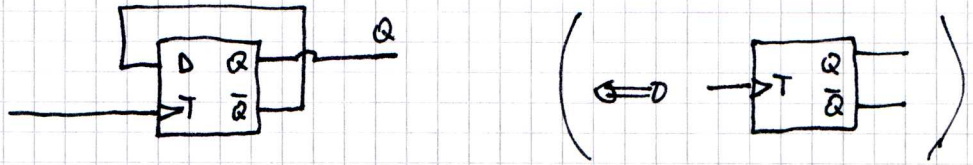
▷ für 'flankengetrieben' !!
D-Flipflop

*) Sie kann nun einen kurzen Impuls bei fallender Taktschranke (\downarrow) erzeugen ??

(C4)

Eine Anwendung: Bau eines simplen Binärzählers

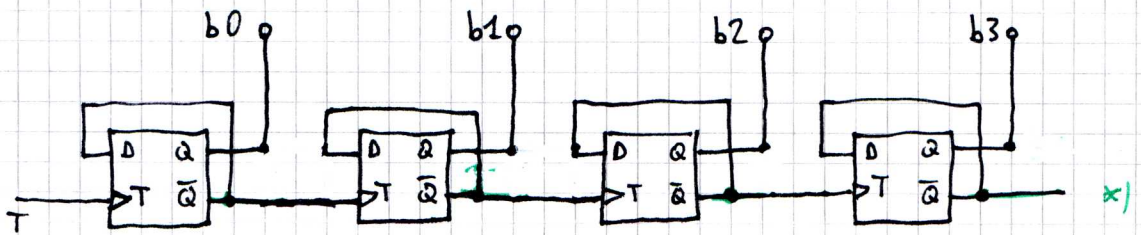
Betrachten wir die folgende Beschreibung eines D-Flipflops:



Was geschieht bei jeder Takt T?

Q wechselt immer bei \uparrow des Takts!

- Also:
- 2 Takte an T \rightarrow 1 Takt bei Q
 - Frequenzen können so halbiert (gerichtet, ...) werden
 - Wir können so einen simplen Binärzähler bauen:



b0 wechselt bei jeder Takt T den Zustand,
b1 " " " zweiten Takt T den Zustand,
b2 " " " dritten " " " "
b3 " " " vierten " " " "

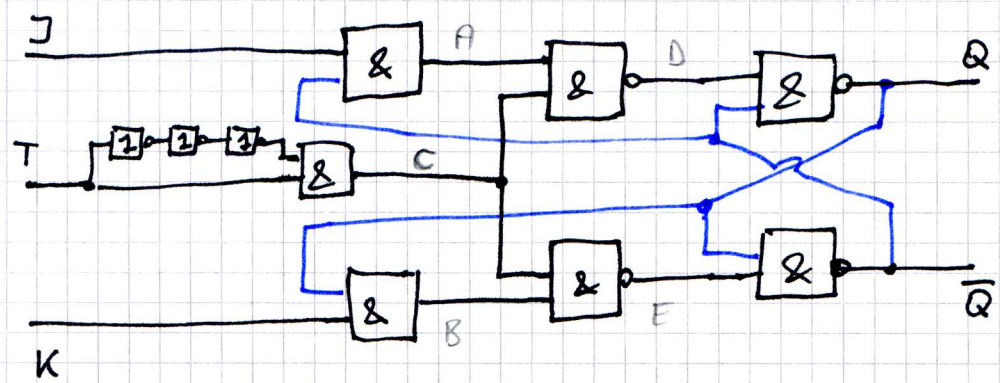
Allerdings können wir diese Zähler nicht auf 0000 zurückstellen,
z.B.

für Schaltkreise bei steigender Taktflanke !!

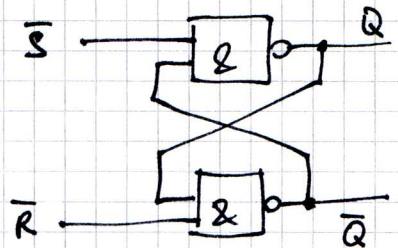
C5

Das JK-FlipFlop

Interessante Eigenschaften hat das folgende Speicherglied, welches nur eindeutig funktioniert bei Flankensteuerung:



Zunächst erwecke wir ein einfaches RS-FlipFlop:



Sind D und E beide high, so kenne die Werte in Q und Q-bar gespeichert. Dies ist immer der Fall für T=0!

Nur bei ansteigender Flanke an T (J) entsteht bei C ein kurzer high-Puls.

J=1 & K=0 : $S=0, R=1$
 bei $Q=1 \text{ \& } \bar{Q}=0$ folgt $A=0, D=1 \rightarrow \text{hold!}$
 bei $Q=0 \text{ \& } \bar{Q}=1$ folgt $A=1$; D goes low at 0
 $\rightarrow \text{Set} \rightarrow \underline{Q=1, \bar{Q}=0}$

J=0 & K=1 wie vorher, aber Reset ist $\bar{Q}=0, Q=1$

J=0 & K=0 $\Rightarrow A=0 \text{ \& } B=0 \Rightarrow D=1 \text{ \& } E=1 \Rightarrow \underline{\text{hold}}$

J=1 & K=1

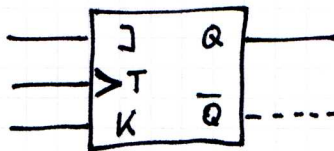
- aus $Q=1$ & $\bar{Q}=0$ folgt $A=0$ und $B=1$, somit $D=1$ und ein hohes Pulz und $E=0 \Rightarrow$ Reset, $Q=0$ & $\bar{Q}=1$
 - aus $Q=0$ & $\bar{Q}=1$ folgt (symmetrisch) Set, $Q=1$ & $\bar{Q}=0$
- d.h. in jedem Fall wechselt der Zustand, sog. 'Toggle'

Zusammenfassung:

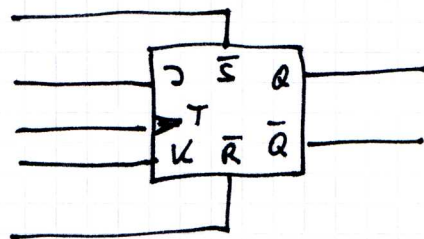
Bei JK-FlipFlop wechselt der Zustand in Q und \bar{Q} ^{hin} bei einem entsprechenden Flanke in T , und dies nach folgender Tabelle

J	K	┌
1	1	Toggle ($Q_{\text{neu}} = \text{NOT}(Q_{\text{alt}})$)
1	0	Set ($Q_{\text{neu}} = 1$)
0	1	Reset ($Q_{\text{neu}} = 0$)
0	0	Hold ($Q_{\text{neu}} = Q_{\text{alt}}$)

Schaltplan:



Es gibt auch JK-Speicher mit zusätzlichen Set- und Reset-Eingängen, die meist durch ein low-Signal aktiviert werden:

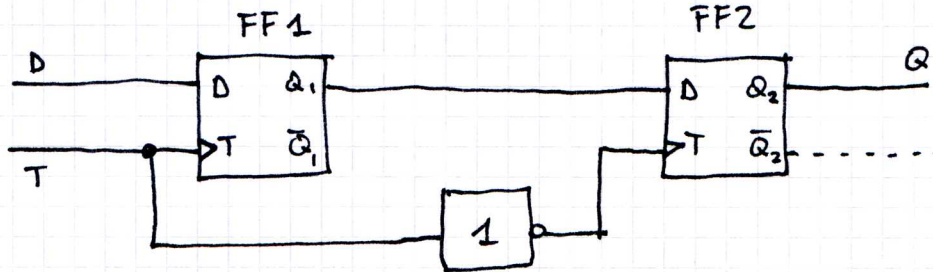


Siehe auch aber Bild (in C7) ein ~~schwierig~~ und lassen Bausteine studieren!

(C6)

Master - Slave - Speicher

Was könnte der Sinn der folgenden Anordnung von zwei D-FlipFlops sein?



FF1 (der 'Master') übernimmt bei steigender Flanke von T die den Wert von D nach Q_1 . FF2 macht dann noch gar nichts, der 'alte' Wert von Q_2 kann am Ausgang gelesen werden. Erst bei fallender Flanke von T übernimmt FF2 (der 'Slave') den neuen Wert von Q_1 nach Q_2 .

Damit erreicht man, dass während desselben Takts neue Daten übernommen ~~wird~~ können und die alten noch gelesen werden können!

┌ übernehmen von Daten

└ Wartezeit der neuen Daten an den Ausgängen

(C7)

Ein Deluxe - Universal Speicherbaustein

Setzt man zwei JK-Speicher als Master und Slave hintereinander und stellt die Jase nach mit Set- und Reset-Eingängen aus, so erhält man einen JK-MS-Speicher mit Set und Reset.

Dieser Speicherbaustein stellt alle gewünschten Funktionen zur Verfügung, er ist somit eine deluxe-Universalvariante.

→ Kopie "JK-MS-Speicher"

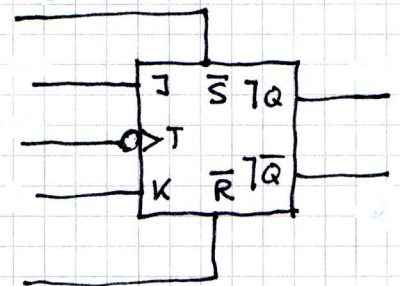
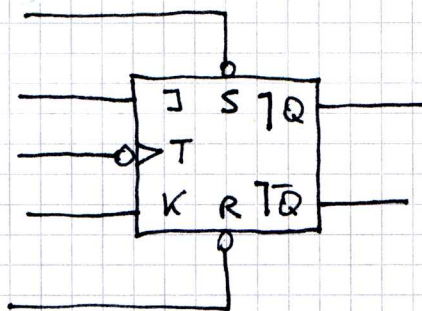
Das Funktionsschema ist dasselbe wie beim JK-Speicher, nur können jetzt rechts gleichzeitig nach die alten Daten gelesen werden, während links ^{schon} die neue hineinkommen. Erst bei \lceil ~~Antida~~ ^{erhöht} die neuen Daten ~~weiter~~ ^{de} und ~~andere~~ - Änderung.

J	K	R	S	Q _{neu} (Mod JK)
-	-	0	1	0
-	-	1	0	1
0	0	1	1	Q _{alt}
0	1	1	1	0
1	0	1	1	1
1	1	1	1	Q _{alt}

Reset }
 Set } sofort
~~Store, Hold~~ }
 (Reset) last 0 }
 (Set) last 1 } nach \lceil
 Toggle }

Das Schaltsymbol

oder

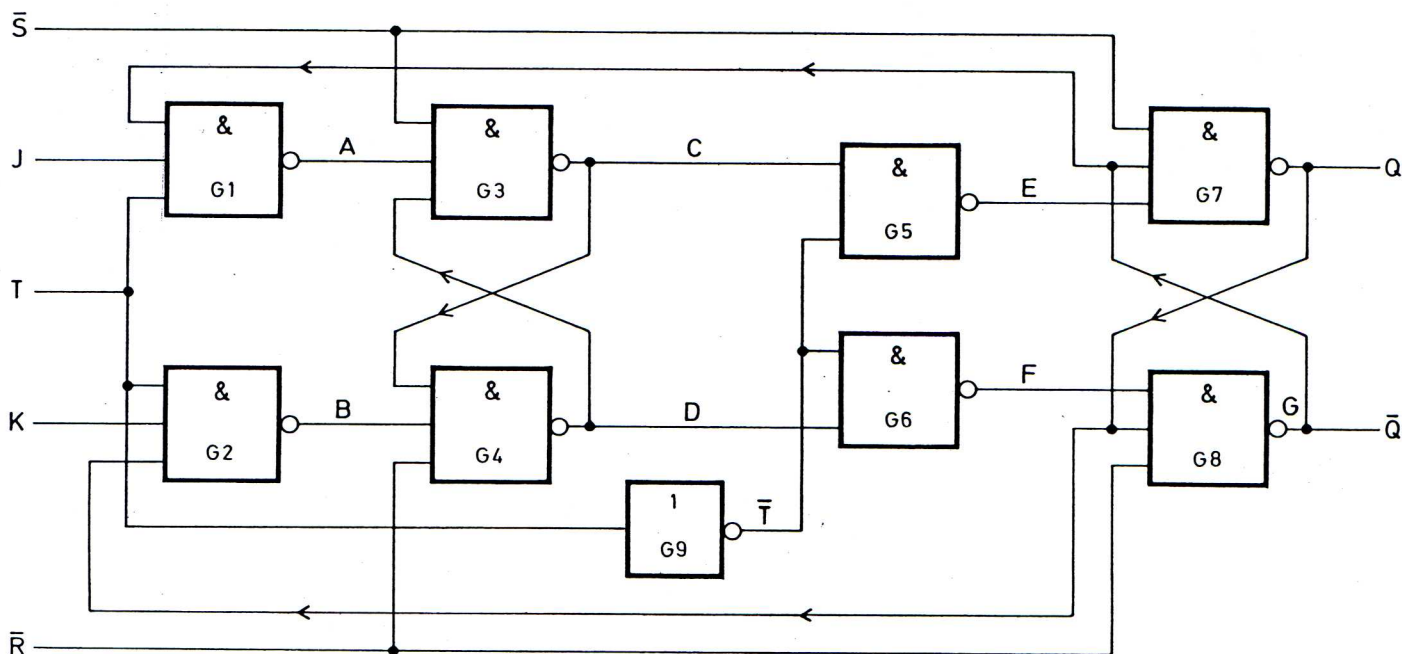


5 Eingänge und 2 Ausgänge ...

Der JK-MS-Speicher

Speicherglieder haben viele verschiedenen Anwendungen. Es wäre aber unwirtschaftlich, für jede Anwendung einen speziellen Typ zu bauen. Daher hat man sich auf einige wenige Standardtypen geeinigt. Darunter ist der JK-MS-Speicher die **Maximalvariante**. Es ist ein Doppelspeicher (MS von Master-Slave) mit zwei Dateneingängen (J und K, die Namen haben keine besondere Bedeutung) sowie einem Set- und einem Reset-Eingang.

Er kann wie folgt aufgebaut sein:



Die Set-Leitung und die Reset-Leitung werden aktiviert, wenn sie auf low gezogen werden. Daher sind sie auch mit den Querstrichen versehen.

Geht \bar{S} auf low, so wird der Ausgang C des Meisters als auch der Ausgang Q des Sklaven auf 1 gesetzt. Wie bei jedem anständigen FlipFlop sind dann die Werte von D und \bar{Q} entsprechend 0. !

Der Reset-Eingang R macht genau das umgekehrte, was ja auch aus der Symmetrie der Anordnung hervorgeht.

Es ist unsinnig, sowohl S als auch R auf low zu ziehen.

Wir können also davon ausgehen, dass S und R high sind, wenn wir untersuchen, wie die Schaltung in Abhängigkeit der Eingänge J und K arbeitet, wenn das Taktsignal T zuerst high und dann wieder low geht. Dabei müssen wir aber den momentanen Zustand von Q und \bar{Q} mitberücksichtigen, da ja diese Werte auf die Eingangsgatter G2 und G1 rückgekoppelt werden!

Wir analysieren die Schaltung in einer Tabelle (wie immer ...).

Analyse JK-MS-Speicher, p.1

$$\underline{T=0, S=1, R=0} \quad (\text{Reset})$$

$$\Rightarrow T=0 \Rightarrow A=1 \text{ \& } B=1$$

$$R=0 \Rightarrow D=1 \text{ \& } \bar{Q}=1$$

$$D \& D, A, S \text{ alle } 1 \Rightarrow C=0 \Rightarrow E=1$$

$$S, E, \bar{Q}=1 \Rightarrow Q=0 \Rightarrow \bar{Q}=1$$

$$\text{Schaltung stabil, } Q=0, \bar{Q}=1$$

$$\underline{T=1, S=1, R=0} \quad (\text{Reset})$$

$$R=0 \Rightarrow D=1 \text{ und } \bar{Q}=1$$

$$\bar{T}=1 \Rightarrow \bar{T}=0 \Rightarrow F=1 \text{ und } E=1$$

$$S, E \text{ und } \bar{Q}=1 \Rightarrow Q=0 \Rightarrow \bar{Q}=1$$

$$\text{Schaltung stabil, } Q=0, \bar{Q}=1$$

Das Set-Ereignis, ausgelöst durch $S=0$, arbeitet ganz anders. Set und Reset sind also jederzeit möglich (asynchron, unabhangig von Takt).

Fur den Rest der Analyse sei $R=1=S$ vorausgesetzt!

Fallt $T \downarrow$, so steigt $\bar{T} \uparrow$ und es wird $E=\bar{C}$ und $F=\bar{D}$, ist z.B. $C=0$ ~~so folgt~~ $D=1$ so folgt

$$E=1 \text{ und } F=0; \text{ also } \bar{Q}=1=D \text{ und } Q=\bar{E}=C$$

Fur $C=1$ und $D=0$ ist es ganz entsprechend.

Vorne kann dadurch nichts andern, da $T=0$!

Der Slave ubernimmt also das Ergebnis $C=0$ im Master, und es gilt $C=Q$ und $D=\bar{Q}$.

Analyse JK-MS-Speicher, p.2

Was geschieht aber wenn, wenn T high geht?]
 Jeder falls geht \bar{T} auf low und die Werte von Q und \bar{Q} bleiben.

$Q_{alt} = C_{alt}$	J	K	A	B	C_{neu}	D_{neu}	T=1!
0	0	0	1	1	0	1	①
1	0	0	1	1	1	0	
0	0	1	1	1	0	1	②
1	0	1	1	0	0	1	
0	1	0	0	1	1	0	③
1	1	0	1	1	1	0	
0	1	1	0	1	1	0	④
1	1	1	1	0	0	1	

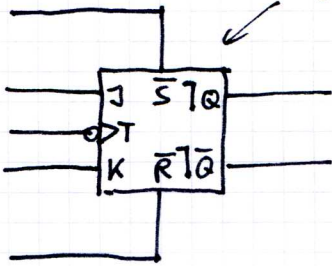
zuerst ~~folgt~~ folgt daraus

- ① Hold !
- ② Load 0, Reset
- ③ Load 1, Set
- ④ Toggle

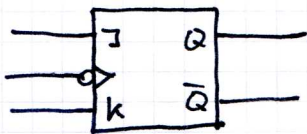
und immer gilt $C = \bar{D}$, wie es sein soll.

Speicherglieder und Flip Flops, ^{- eine} Überfahrt

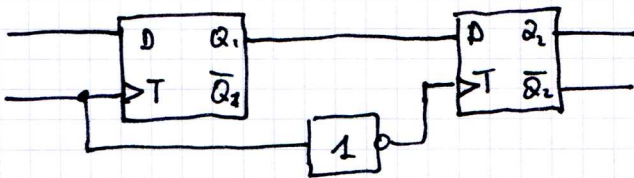
↑ für Output ist nun nach folgender Flanke ...



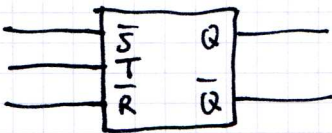
JK-MS-Speicher mit Set und Reset



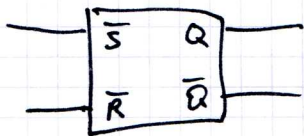
JK-Speicher ohne Set und Reset, ~~mit Set~~ ohne MS!



Doppel-D-Speicher oder MS-Speicher mit Flankesteuerung (ohne JK...)



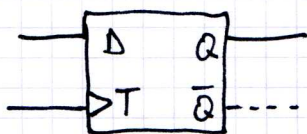
RS-Speicher ~~ohne~~ Flanke mit Totenergie



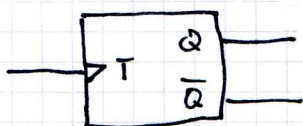
RS-Speicher ohne Totenergie



Fangspeicher mit Totenergie



Flankesteuerter ~~Totenergie~~ Fangspeicher



Taktgesteuerter Flip Flop, bei jeder gleich Takt flanke kippen Q und Q-bar (hier ohne Reset...)