

Vorlesung 5

Themen dieser Vorlesung sind

- Sequentielle Logik
- Latch, Flipflop Funktionsweise
- Statische Speicherzellen
- Statisches Latch und Flipflop, Flipflop mit Reset
- RS-Latch
- Toggle-Flipflop
- Latch als Differenzschaltung

Sequenzielle Schaltungen

Bei Sequenziellen Schaltungen hängt der Ausgang nicht nur von den Eingangsvariablen, sondern auch vom Zustand der Schaltung. Sequenzielle Schaltungen brauchen Speicherzellen.

Flipflops (Kippschaltung) waren die ersten elektronischen Speicherzellen.

Abbildung 1 zeigt eine Klassifizierung.

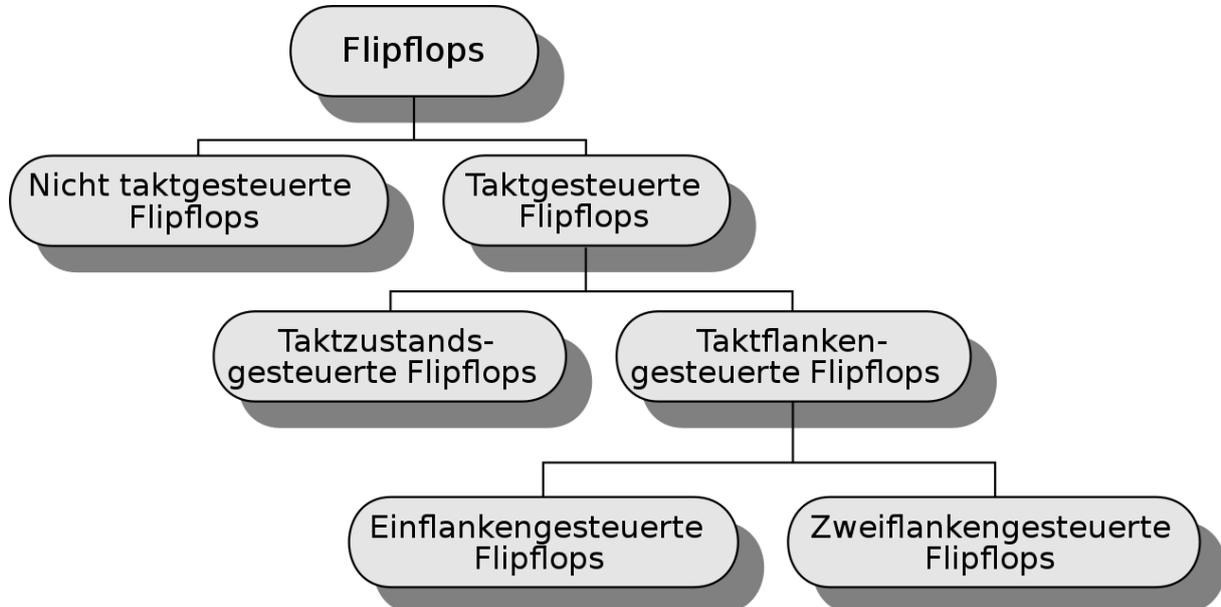


Abbildung 1: Klassifizierung von Speicherzellen. <https://de.wikipedia.org/wiki/Flipflop>

Wir benutzen folgende Termine:

Latch: Nichttaktgesteuerte und Taktzustandsgesteuerte Speicherzellen (Abbildung 1)

Flipflop: Taktflankengesteuerte Speicherzellen

Wir werden hier hauptsächlich die D-Latches und die das so genannten master-slave (MS) Flipflop betrachten. Ein MS Flipflop besteht aus zwei Latch-es in Reihe. Es gibt auch andere Realisierungen.

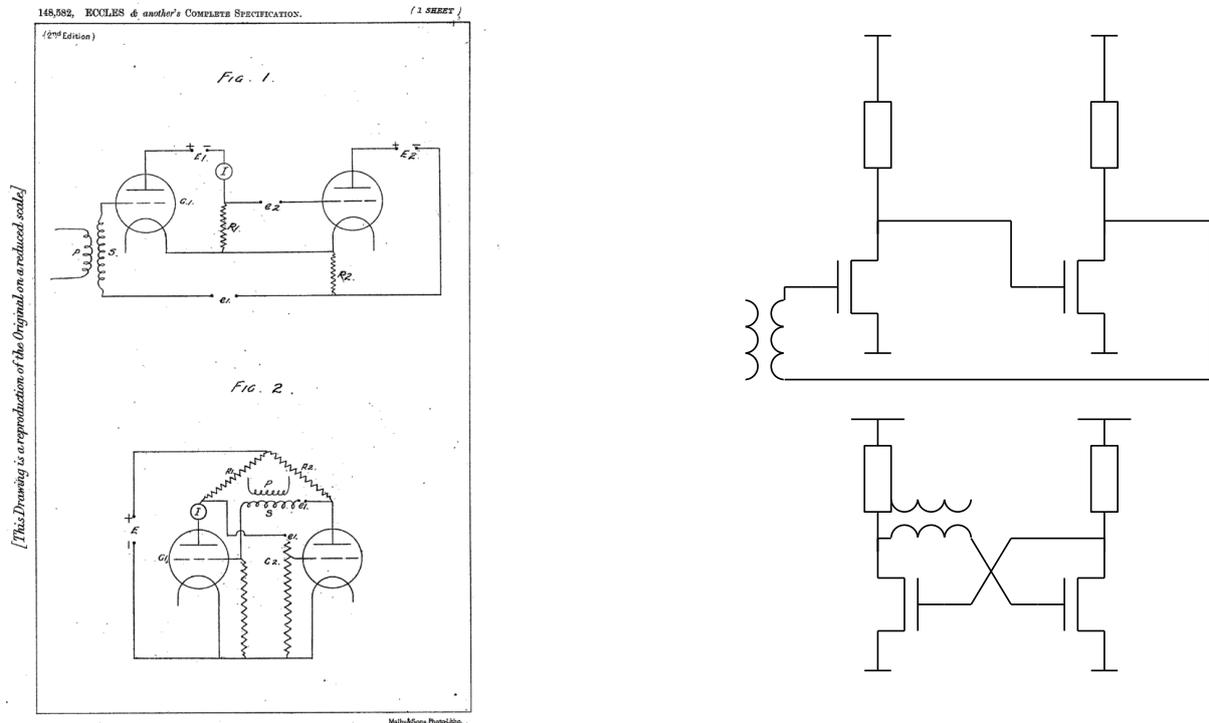


Abbildung 2: Links Zeichnung aus dem Patent, rechts Äquivalente Schaltung mit MOSFETs

<https://de.wikipedia.org/wiki/Flipflop> Die Flipflopschaltung wurde von den Engländern William Henry Eccles und Frank W. Jordan an rückgekoppelten Radoröhrenverstärkern auf der Suche nach Zählschaltungen entdeckt. Am 21. Juni 1918 wurde das Patent GB 148,582 angemeldet. Ursprünglich erhielt sie die Bezeichnung Eccles-Jordan-Schaltung.

In der ersten Vorlesung wurden das Flipflop und das Latch eingeführt. Als Speicherelement haben wir einen Kondensator gehabt. Das Latch (Abbildung 3, links) speichert ein Eingangsniveau auf einem Kondensator, wenn Load Signal 1 ist. Wenn Load 0 ist, bleibt der Zustand erhalten.

Ein Flipflop ist in Abbildung 3, rechts gezeigt. Es besteht aus zwei Latches (Latch1 und Latch2). Das Load-Signal des ersten Latch ist an negiertes Taktsignal CkB angeschlossen und das Load-Signal des zweiten Latch an das nichtnegierte Signal Ck.

(B in CkB, bedeutet „bar“, Englisch für Negationsstrich.)

Flipflop-Ausgang ist der Ausgang des zweiten Latch (Q2). Im Flipflop wird der Eingangswert D auf die steigende Taktflanke gespeichert. Spätere Änderungen am D-Eingang haben keine Wirkung auf den Ausgang bis zur nächsten Taktflanke.

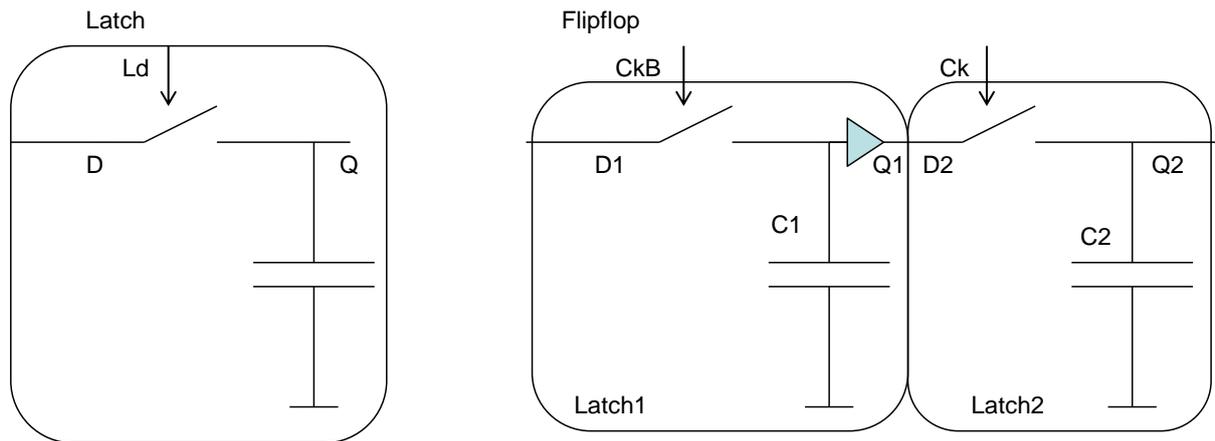


Abbildung 3: Latch und Master-Slave Flipflop

Die Funktionsweise vom Latch erinnert an eine unvollständige Schleuse mit einem Tor (Abbildung 4).

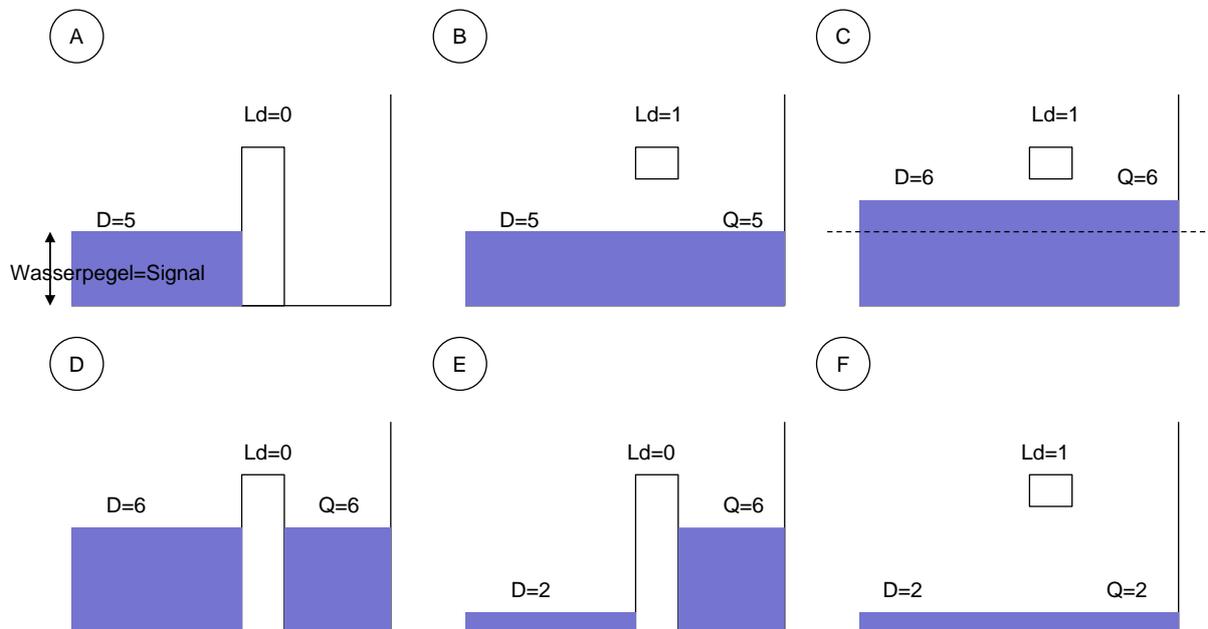


Abbildung 4: Latch Funktionsweise

Der Wasserpegel illustriert ein digitales Eingangssignal. Das Signal ist am Anfang 5 (A). Im Moment (B) wird das Schleusentor geöffnet (Signal Load wird 1) und das Wasser kommt in die rechte Schleusenkammer, die den Latch-Ausgang modelliert. Wenn sich der Wasserpegel, während das Tor offen ist (C) ändert, wird die Änderung am Ausgang sofort sichtbar. Das Latch ist transparent. Im Moment D wird das Tor geschlossen (Load = 0). Der Eingangswert wird in der rechten Kammer gespeichert. Der Eingang ändert sich im Moment E, aber der Ausgang bleibt gespeichert. Der Ausgang ändert sich erst wenn Load wieder logisch Eins wird (F).

Flipflop

Die Funktionsweise vom Flipflop erinnert an eine Schleuse mit zwei Toren (Abbildung 5).

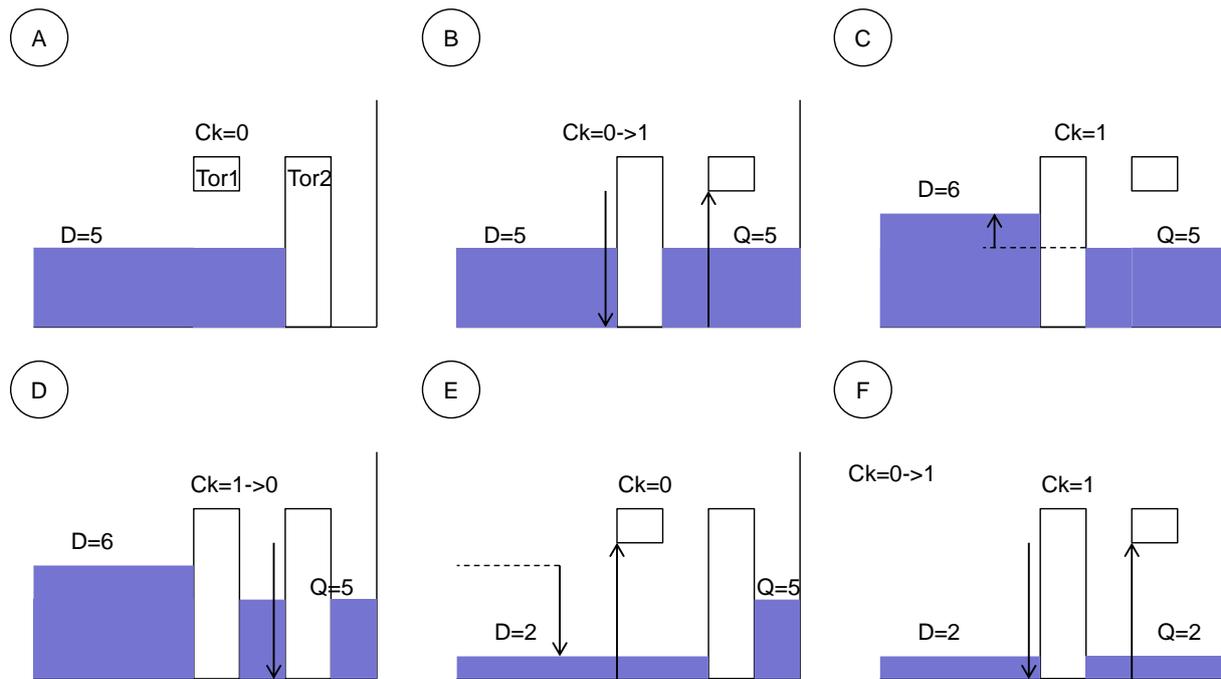


Abbildung 5: Flipflop Funktionsweise

Tor 1 wird auf $Ck = 0$ geöffnet und auf $Ck = 1$ geschlossen. Tor 2 wird aus $Ck = 1$ geöffnet und auf $Ck = 0$ geschlossen. Die Tore sind also in Gegenphase, wie bei einer echten Schleuse. Ein Tor ist immer geschlossen. Es gibt eine linke Kammer (Eingangssignal), eine mittlere Kammer und die rechte Kammer (Ausgangssignal). Wir fangen an mit dem Zustand $Ck = 0$ (A) an, wo Tor 1 offen und Tor 2 zu ist. Der Eingangssignal $D = 5$ wird im Moment B, wo sich Ck von 0 auf 1 ändert, am Ausgang sichtbar, da sich in diesem Moment das Tor 2 öffnet. Gleichzeitig schließt Tor 1 und der Wert wird gespeichert. Im Moment C ändert sich der Eingang von 5 auf 6. Diese Änderung kommt an den Ausgang nicht an, das Tor 1 zu ist.

Im Moment D ändert sich Ck von 1 auf 0. Wir nehmen an, dass sich das Tor 2 zuerst schließt (D) und das Tor 1 kurze Zeit danach öffnet (E). Der Wasserpegel in der rechten Kammer (am Ausgang) bleibt 5. Das Eingangssignal ändert sich danach auf 2. Diese Änderung wird erst bei der nächsten steigenden Taktflanke (F) an Ausgang übertragen.

Es ist wichtig, dass bei der Ck Änderung von 1 auf 0, zuerst beide Tore geschlossen werden und dann das erste geöffnet wird. Dann kommt es bei der fallenden Taktflanke zu keiner Änderung am Ausgang. Das wird im Beispiel in Abbildung 6 nicht gewährleistet. Das Flipflop ist für kurze Zeit transparent und funktioniert nicht richtig.

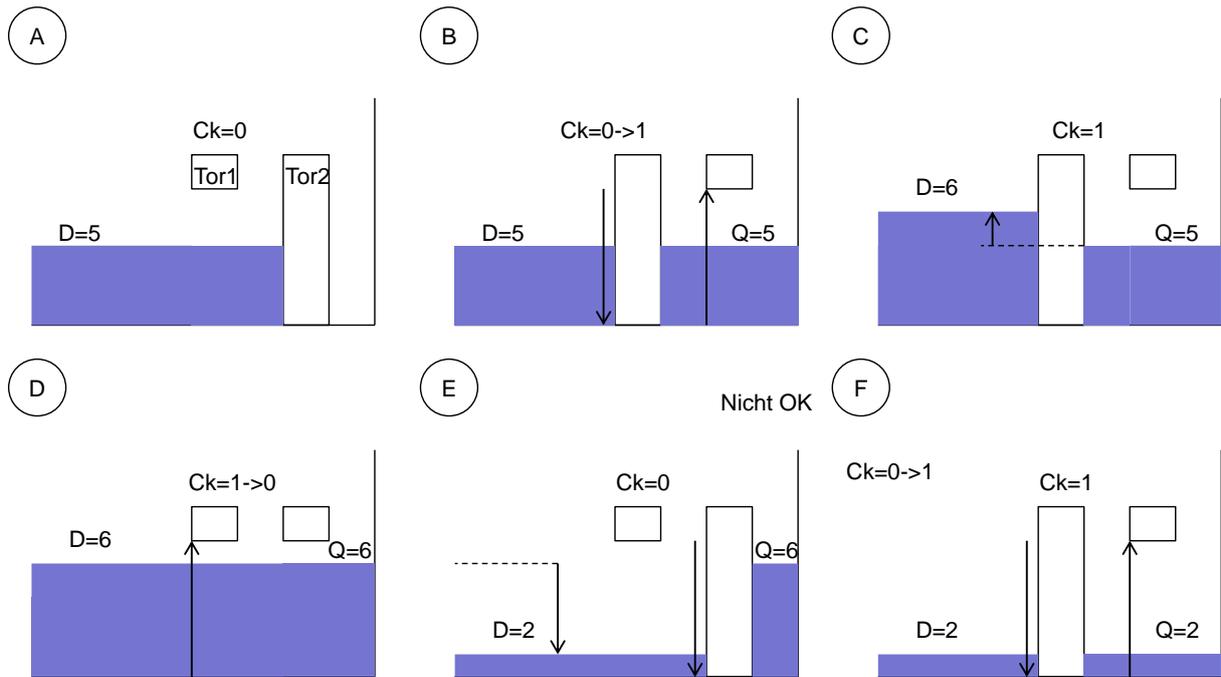


Abbildung 6: Flipflop mit „Bug“

Der Nachteil einer Latch Schaltung mit dem Kondensator ist es, dass sie den Zustand nicht beliebig lange halten kann. Der Kondensator wird langsam entladen. Solche Schaltungen nennt man dynamische Speicherzellen.

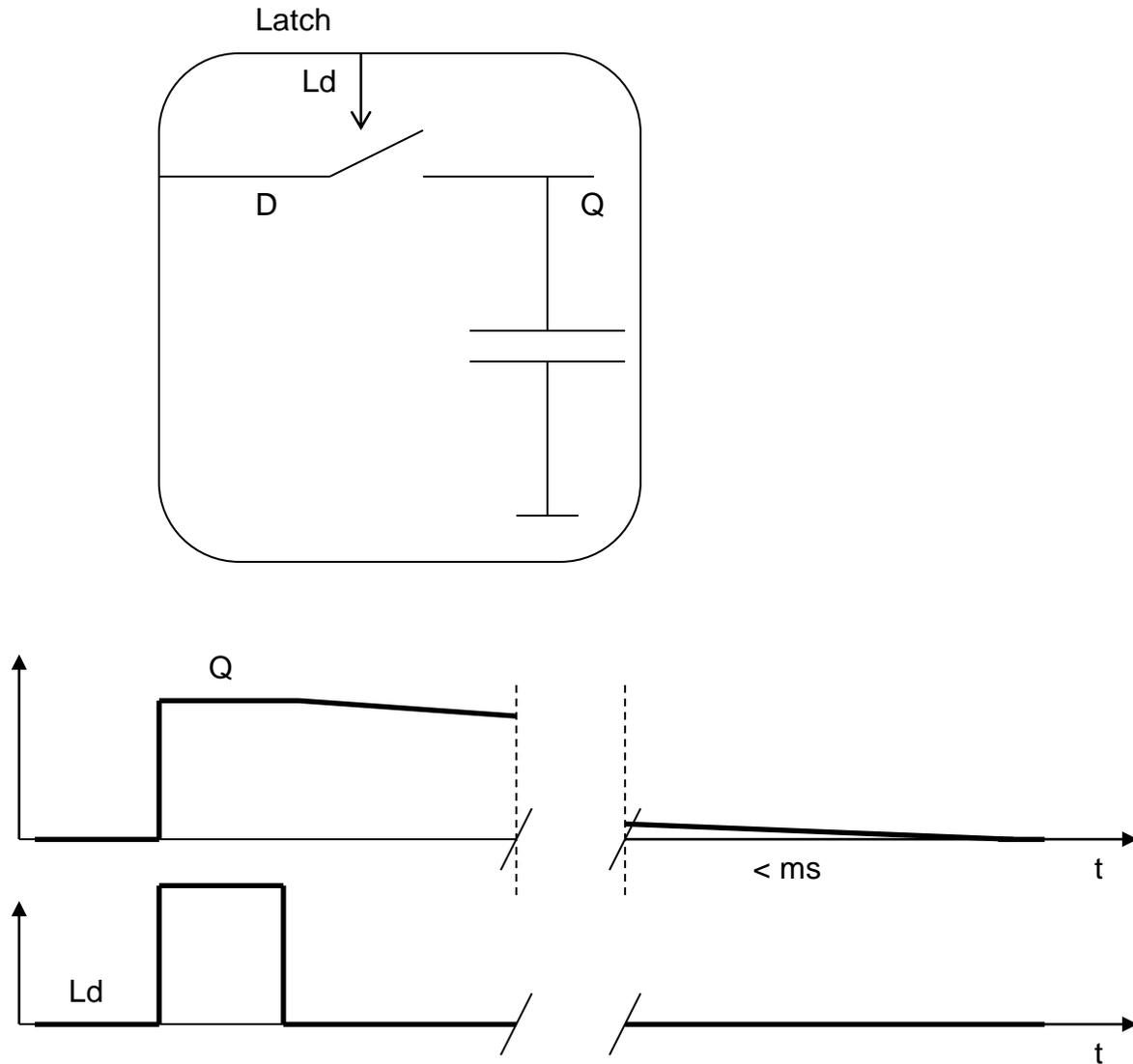


Abbildung 7: Dynamische Speicherzelle – Kondensator wird entladen und die Zelle wechselt den Zustand

Wir werden jetzt zeigen, wie man ein Latch baut, das den Zustand so lange hält, bis die Spannungsversorgung ausgeschaltet wird. Solche Speicherzellen nennt man statische Speicherzellen. Statische Speicherzellen basieren auf einer Mitkopplung (einer positiven Rückkopplung). Betrachten wir zwei in Reihe geschalteten Invertieren. Ihre Kennlinie sieht wie in Abbildung 8 aus.

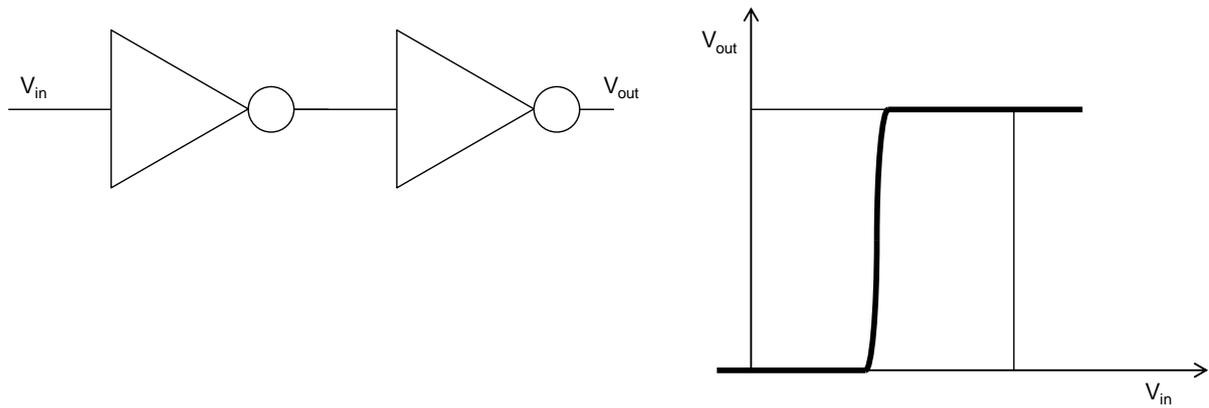


Abbildung 8: Zwei Inverter - Kennlinie

Wenn wir nun den Ausgang des zweiten Inverters mit dem Eingang des ersten verbinden, gilt $V_{in} = V_{out}$. Der Zustand der Schaltung liegt also im Schnittpunkt der Kennlinie $V_{out} = f(V_{in})$ und der Gerade $V_{out} = V_{in}$ (Abbildung 9).

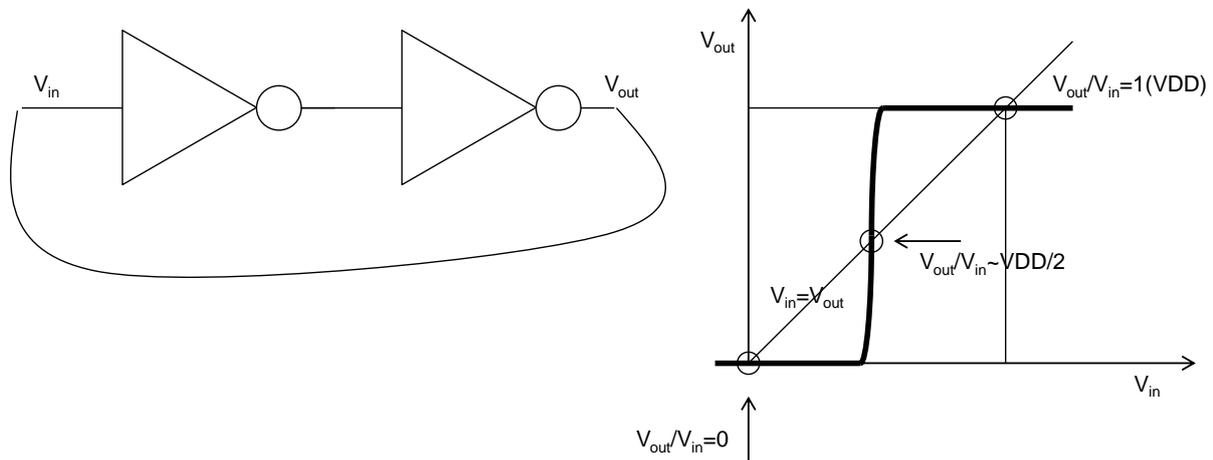


Abbildung 9: Statische Speicherzelle

Es gibt drei Schnittpunkte: $V_{out}/V_{in} = 0$ (logisch 0), $V_{out}/V_{in} = V_{DD}$ (logisch eins) und $V_{out}=V_{in} \sim V_{DD}/2$ (undefiniert). Die ersten zwei Arbeitspunkte sind stabil.

Beweis:

Kleine Störung Δ führt zu: $V_{in} = V_{out} - \Delta$ (Abbildung 10). Diese V_{in} -Änderung beeinflusst V_{out} in den ersten zwei Arbeitspunkten nicht.

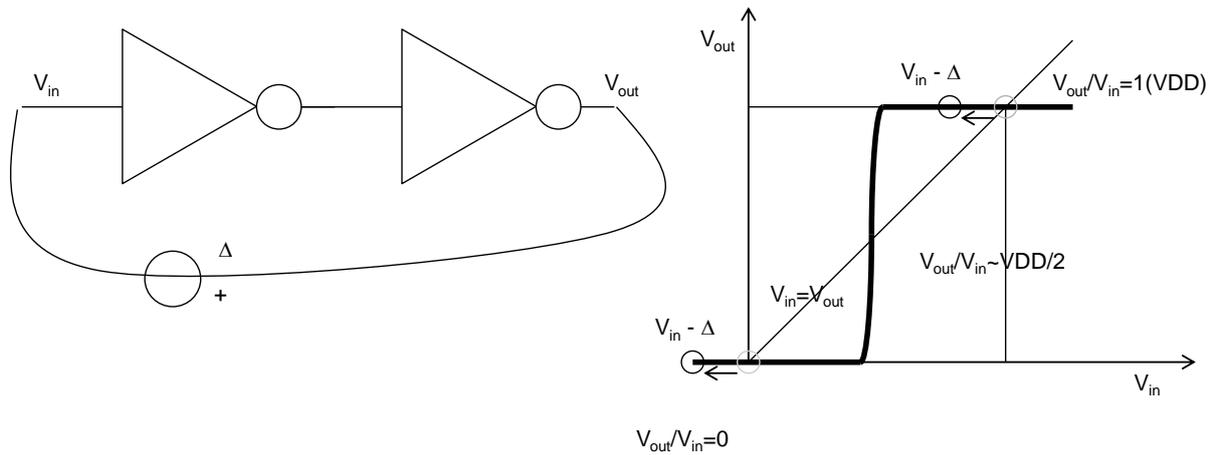


Abbildung 10: Stabile Arbeitspunkte

Wenn wir den dritten Arbeitspunkt betrachten, sehen wir, dass kleine Änderung von V_{in} (verursacht z.B. durch das Rauschen), wegen der steilen Kennlinie zur noch größeren Änderung von V_{out} führt. Die Schaltung kommt aus dem instabilen Arbeitspunkt heraus, und „kippt um“ entweder in den Arbeitspunkt $V_{out}/V_{in} = 0$ oder in den Arbeitspunkt $V_{out}/V_{in} = VDD$. Eine Analogie wäre eine Münze die nie auf ihrem Rand stehen bleibt, oder eine Kugel die auf einer Spitze nicht lange bleiben kann (Abbildung 12).

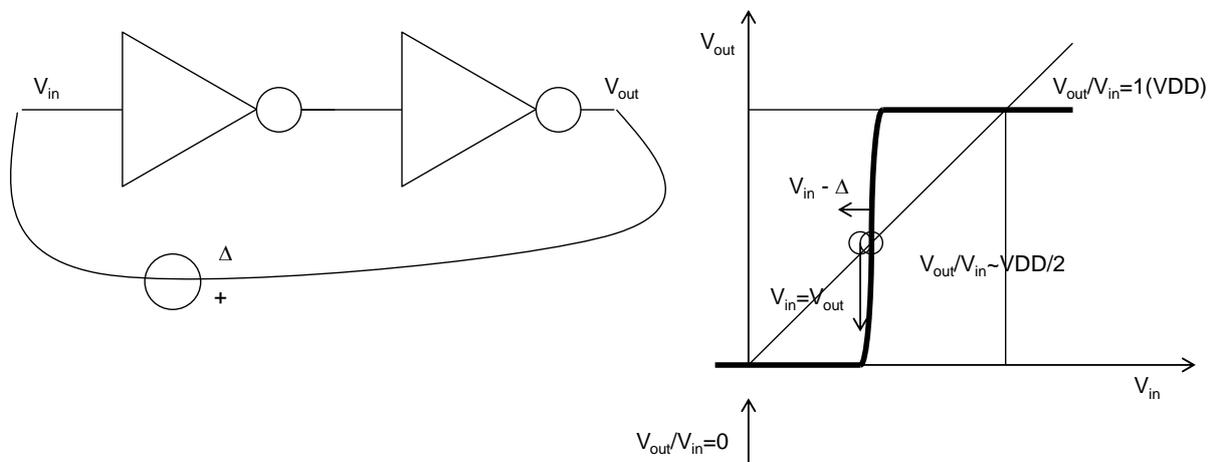


Abbildung 11: Instabiler Arbeitspunkt

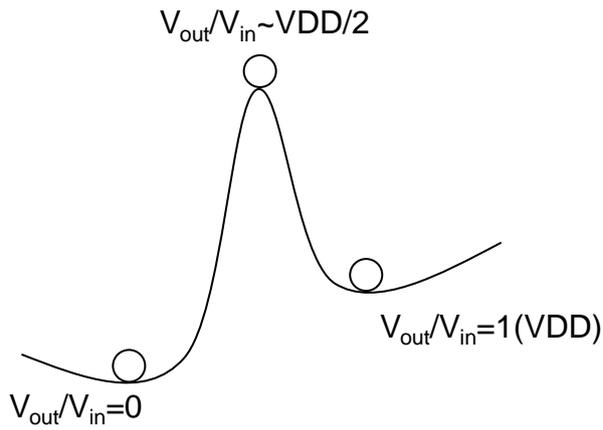


Abbildung 12: Analogie

Diese Schaltung ist die Basis einer SRAM Zelle. Die zwei Schaltern werden benutzt, und die RAM Zelle in gewünschten Zustand zu bringen oder sie auszulesen (Abbildung 13).

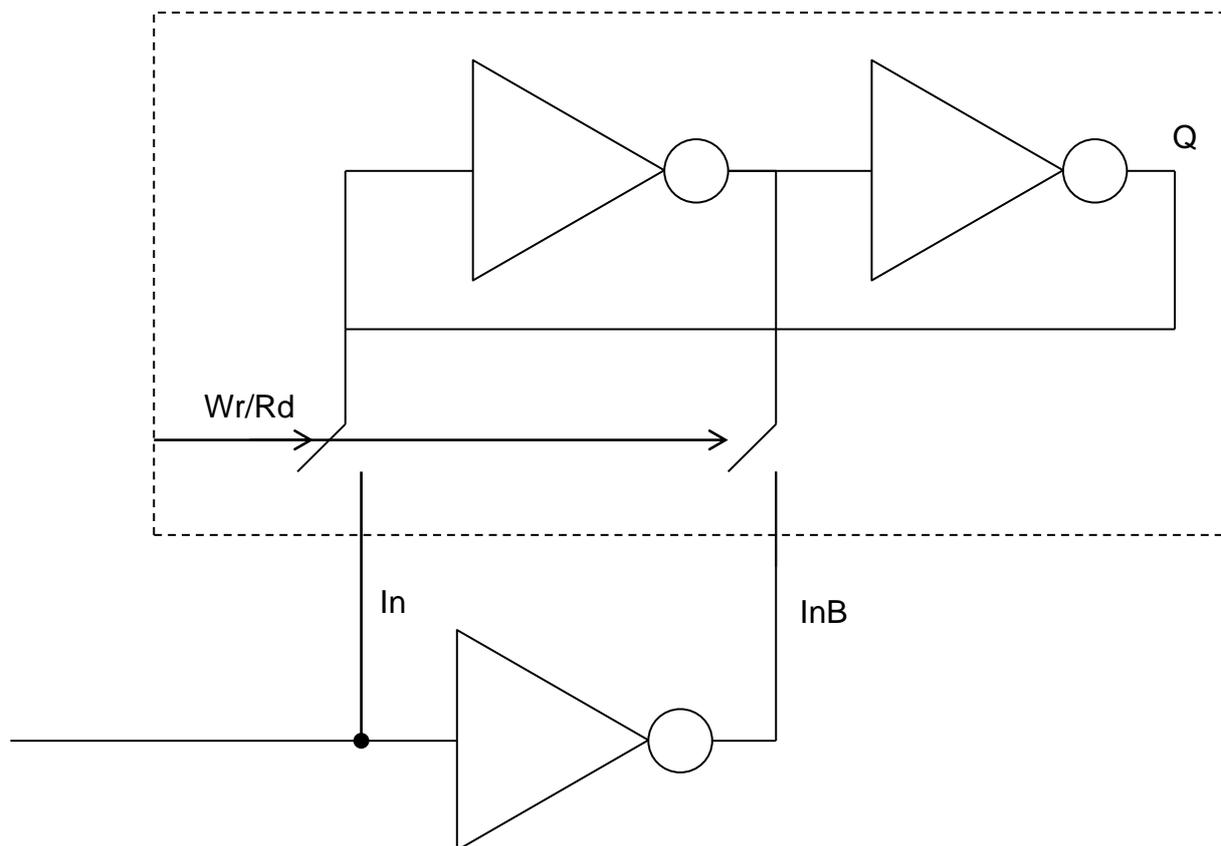


Abbildung 13: Statische RAM Zelle

Ein Latch basiert auf einer modifizierten Version der Speicherzelle von Abbildung 13. Es gibt mehrere Implementierungen. Man kann z.B. zwei NAND oder NOR Gatter benutzen, worauf wir später eingehen werden. Auch ein Multiplexer kann benutzt werden (Abbildung 14). Der

Select-Eingang ist dann an das Load Signal angeschlossen. Eingang X1 ist der Latch Eingang D. X0 wird an den Ausgang des Multiplexers angeschlossen und so eine Rückkopplung gemacht.

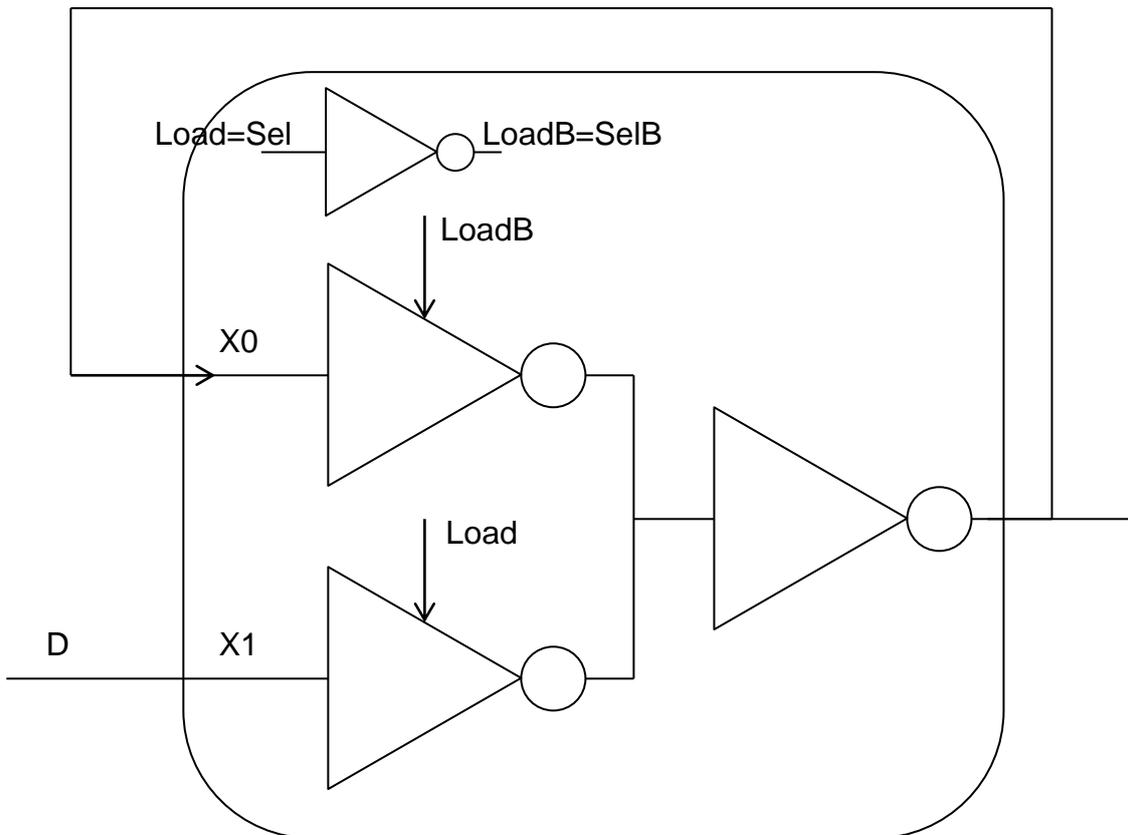


Abbildung 14: Latch basierend auf Multiplexer

Wenn Load Eins ist, ist das Latch transparent. Es gibt einen Signalpfad zwischen dem Eingang und dem Ausgang (Abbildung 15).

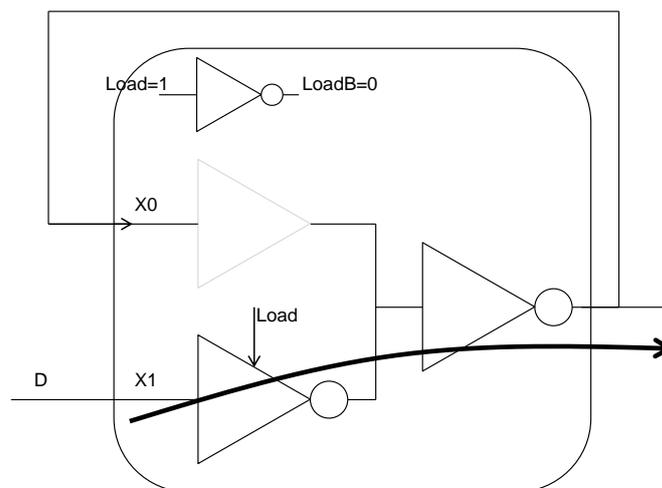


Abbildung 15: Latch ist transparent

Wenn Load Null ist, wird die Rückkopplung aktiviert, ähnlich wie bei der RAM Zelle (Abbildung 16). Der Multiplexer speichert den Zustand.

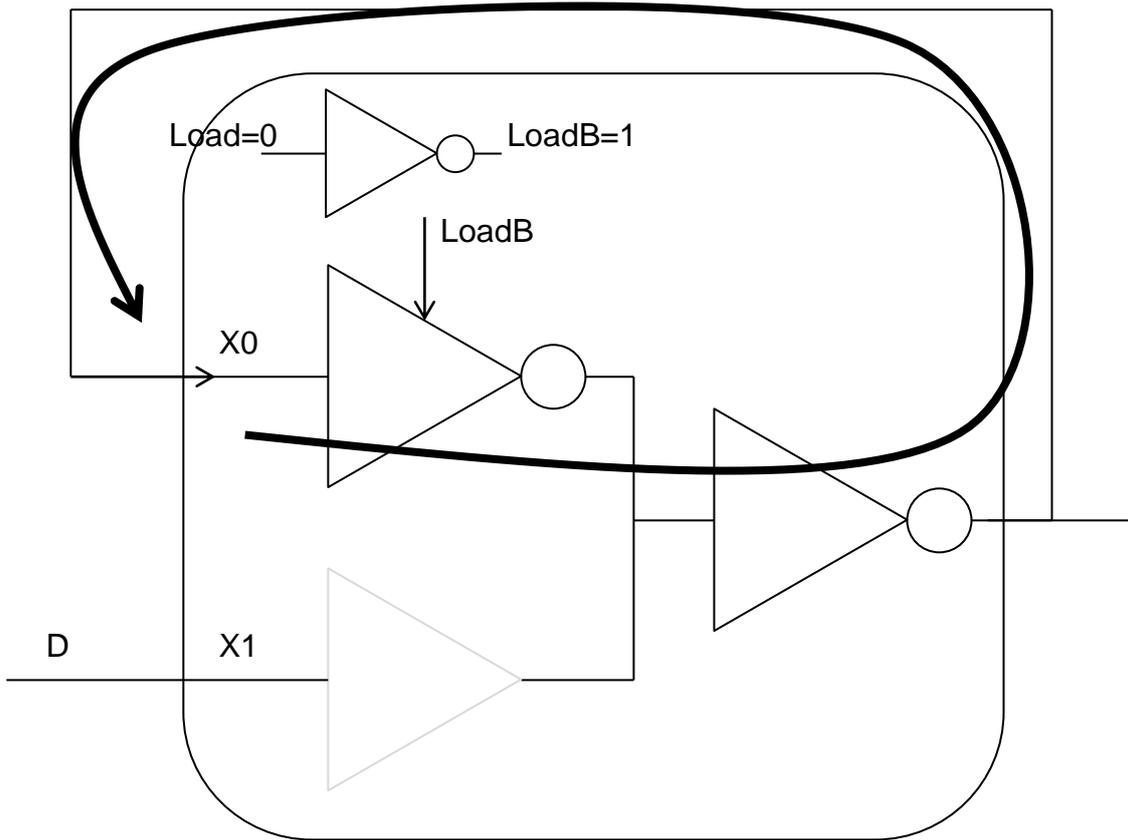


Abbildung 16: Speicherzustand

Ein master-slave Flipflop bauen wir mithilfe von zwei Latches auf (Abbildung 17). Load Eingang des zweiten Latch (master Latch) wird an Ck angeschlossen (Sel Eingang dessen Multiplexers = Ck), Load Eingang des ersten Latch (slave) an das negierte Signal CkB.

Es soll vermieden werden, dass beide Latches gleichzeitig transparent werden, insbesondere wenn sich Ck von 1 auf 0 ändert (inaktive Flanke).

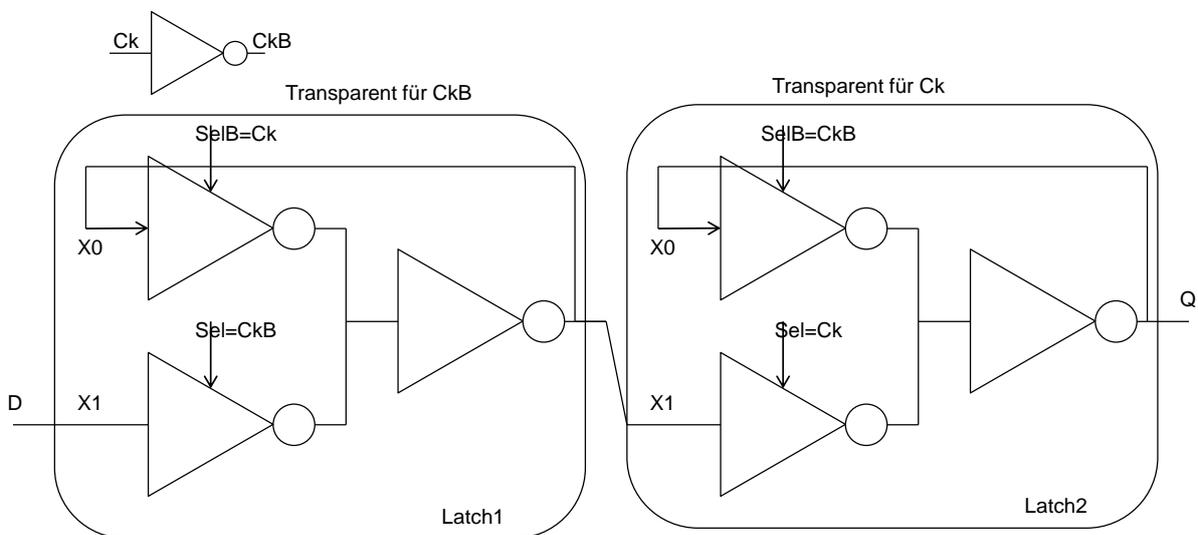


Abbildung 17: Flipflop

Das könnte passieren, wenn die Taktinvertern zu langsam sind. Wir haben gesehen, dass eine kapazitive Last die CMOS Schaltungen verlangsamt. Stellen wir uns folgende Schaltung vor: Wir bauen aus 1024 Flipflops ein 1kBit Register auf. Um Platz zu sparen, verwenden wir zwei kleine Inverter für alle Flipflops (Abbildung 18). Hier kann folgendes Problem auftreten:

Beide Inverter haben große kapazitive Last. Die Taktflanken werden langsam und es kann passieren, dass beide Latches im Flipflop kurze Zeit Load = 1 sehen. Das ist mit grauen Zeitbereichen in Abbildung 18 illustriert.

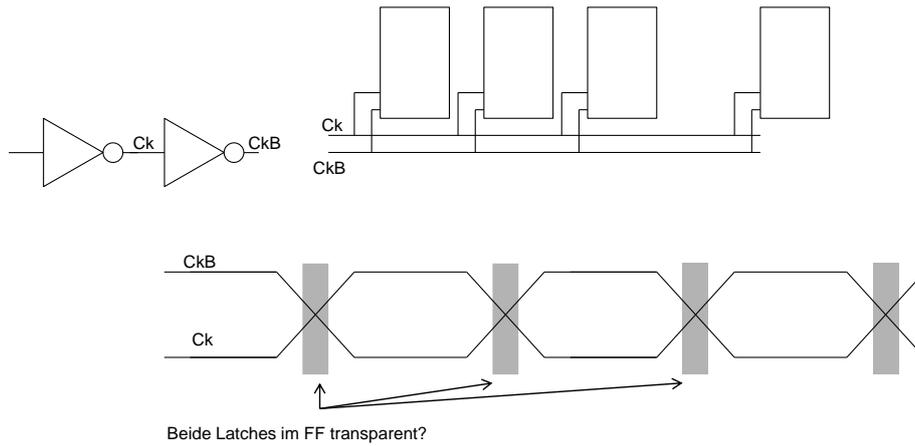


Abbildung 18: Probleme mit langsamen Taktflanken

Um solche Probleme zu vermeiden, werden die Takt-Inverter in den Flipflops eingebaut (Abbildung 19 und Abbildung 20).

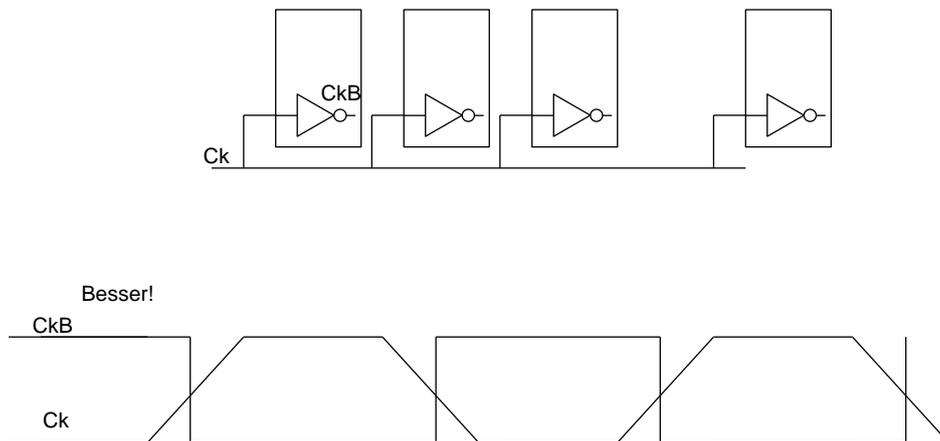


Abbildung 19: Lösung 1

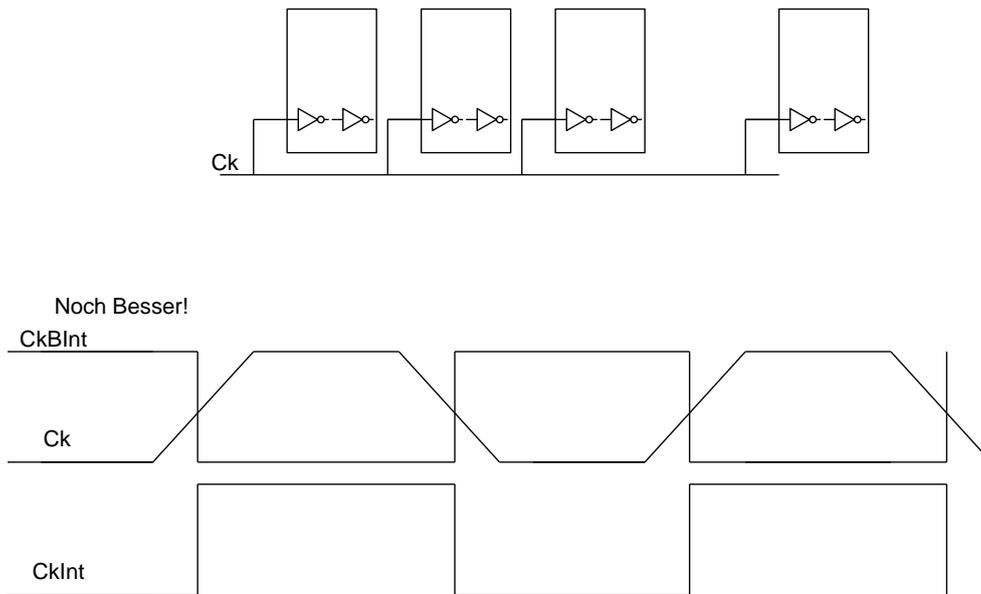


Abbildung 20: Lösung 2

Reset

Nach dem Einschalten der Spannungsversorgung befindet sich ein Flipflop in einem unbekanntem logischen Zustand. Wir könnten uns vorstellen, dass alle Flipflops zuerst in den astabilen Zustand kommen und dann in logisch Eins oder Null Zustand kippen.

Um einen unbekanntem Anfangszustand zu vermeiden, werden die Flipflops oft erweitert, die in asynchrones Reset Signal haben. Die Schaltung eines Flipflops mit dem asynchronen Reset ist in Abbildung 21 gezeigt.

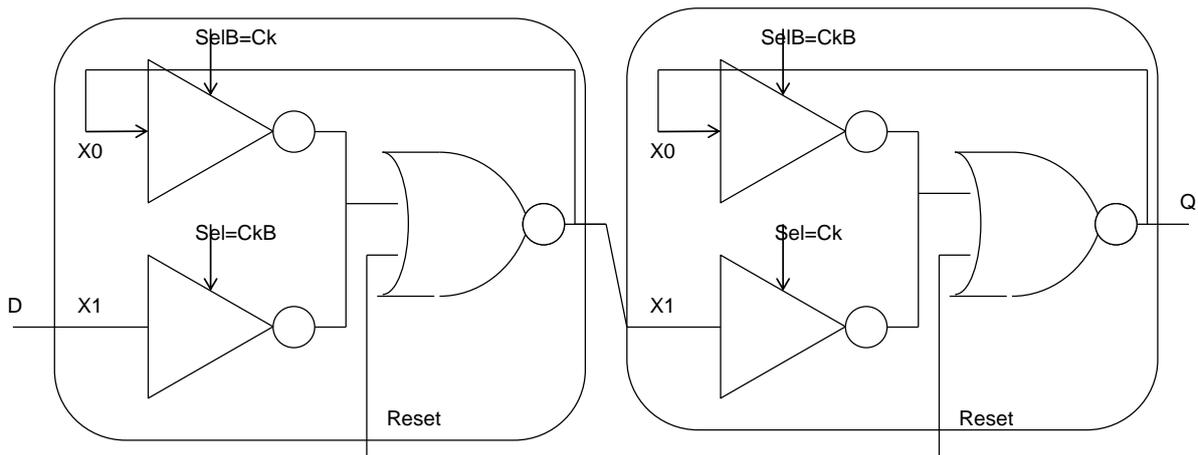


Abbildung 21: Flipflop mit Reset

Betrachten wir dieses Flipflop im $Ck = 0$ Zustand (Abbildung 22). $Reset = 1$ erzeugt logisch Null am Ausgang Q. Diese Null erzeugt Eins am NOR Eingang. Somit bleibt Ausgang null, auch wenn Reset ausgeschaltet wird. Ein logisch Eins am NOR Eingang ist ausreichend, um

Null zu erzeugen. Auf diese Weise bleibt Null gespeichert, auch wenn Reset wieder inaktiv wird.

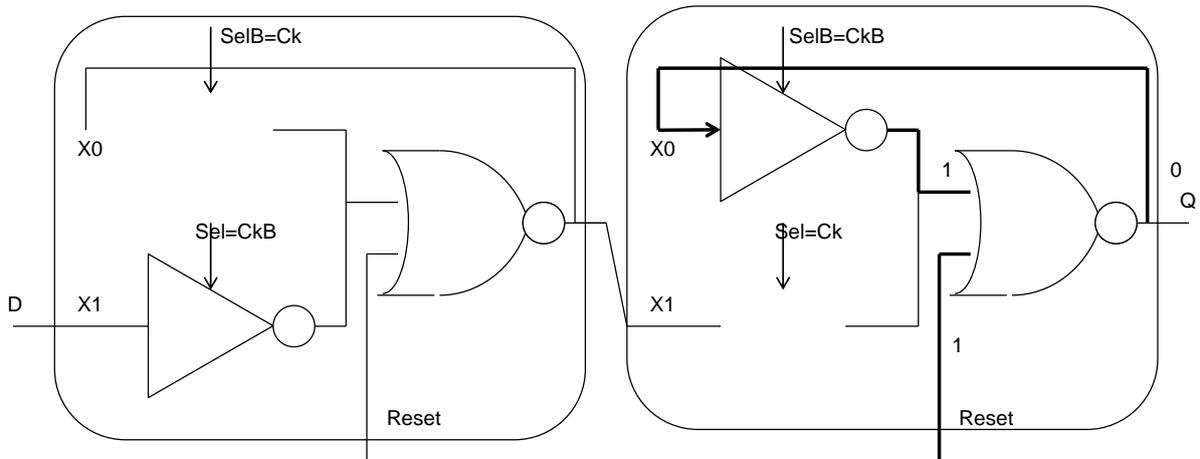


Abbildung 22: Reset für $Ck = 0$

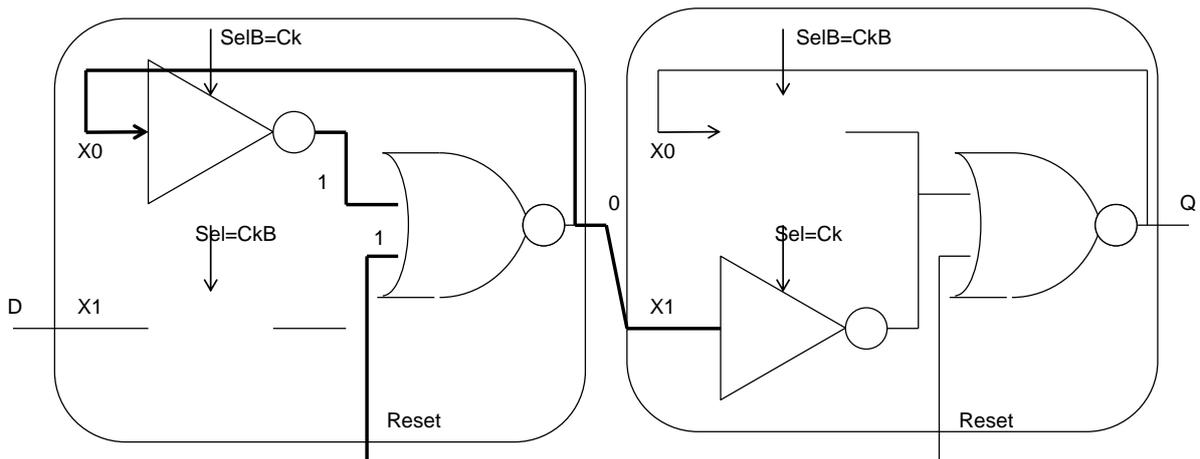


Abbildung 23: Reset für $Ck = 1$

Im Flipflop ist immer wenigstens ein Latch im Speicherzustand, so dass ein Reset immer möglich ist, wenn beide Latches die Reset Logik enthalten. Abbildung 23 zeigt das Reset während $Ck = 1$ ist.

Asynchron Reset ist stärker als der Takteingang. Wenn $Reset = 1$, ist der Flipflop Ausgang null, unabhängig von D und Ck Eingängen.

Es gibt auch weitere Implementierungen von Latches und Flipflops.

Eine Übersicht ist in [Flipflop – Wikipedia](#).

Wichtige Schaltung ist ein RS Flipflop, nach unserer Klassifizierung ist es ein RS-Latch. Anderer Name ist bistabile Kippstufe.

Diese Schaltung besteht entweder aus zwei NAND Gatter (Abbildung 24, links) oder aus zwei NOR Gatter (Abbildung 24, rechts).

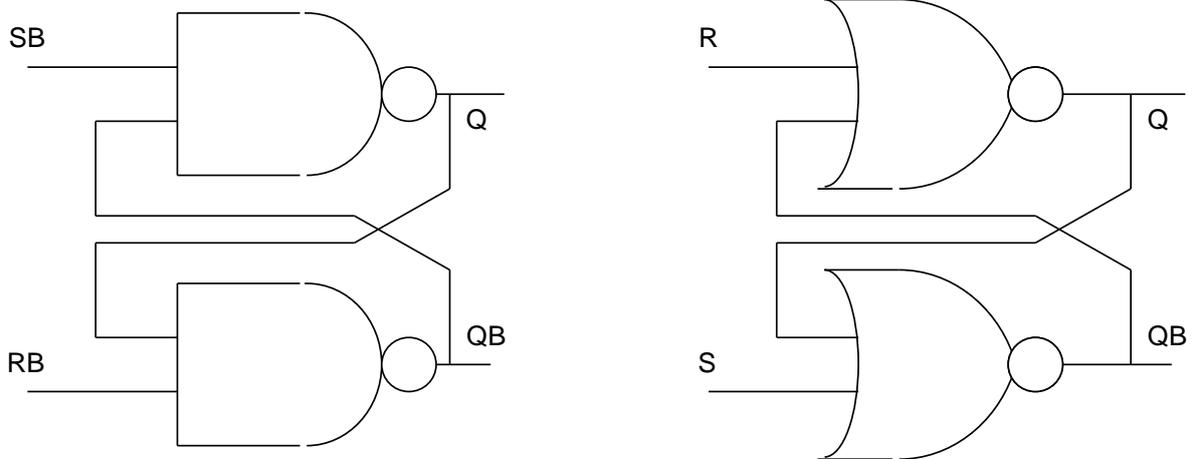


Abbildung 24: SR - Latch

Betrachten wir die NAND Variante. Die Schaltung hat zwei Eingänge, Set-Eingang SB und Reset-Eingang RB. Beide sind aktiv bei logisch Null. Nur ein der Eingänge ist in einem Moment aktiv. Wenn Set Eingang Null ist (0), ist der negierte Ausgang $QB = 0$ und der Ausgang $Q = 1$. Das Latch ist im Eins-Zustand. Es ist gesetzt worden. $QB = 0$, erzeugt $Q = 1$. Wenn SB wieder 1 wird, bleiben Q und QB Werte erhalten.

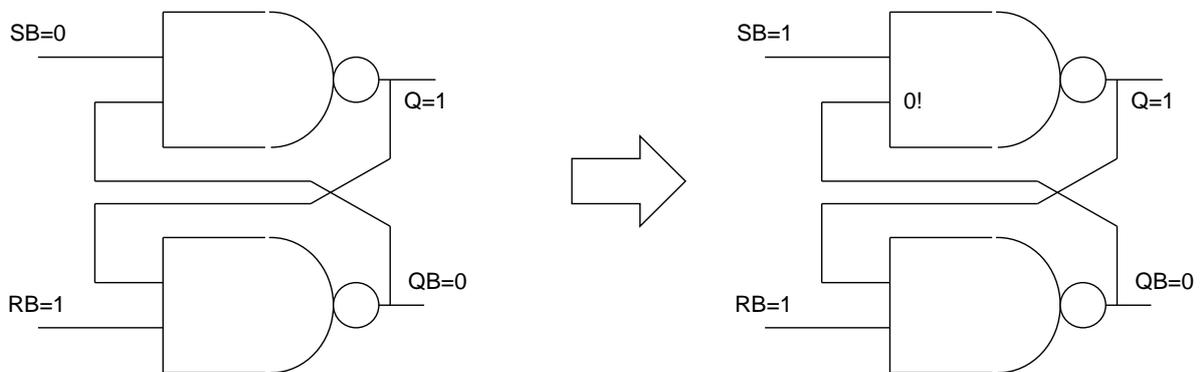


Abbildung 25: Set vom SR Latch

Bei $RB = 0$, wird das Latch in Null-Zustand versetzt ($Q = 0, QB = 1$). Es bleibt in diesem Zustand, auch wenn Reset logisch 1 wird.

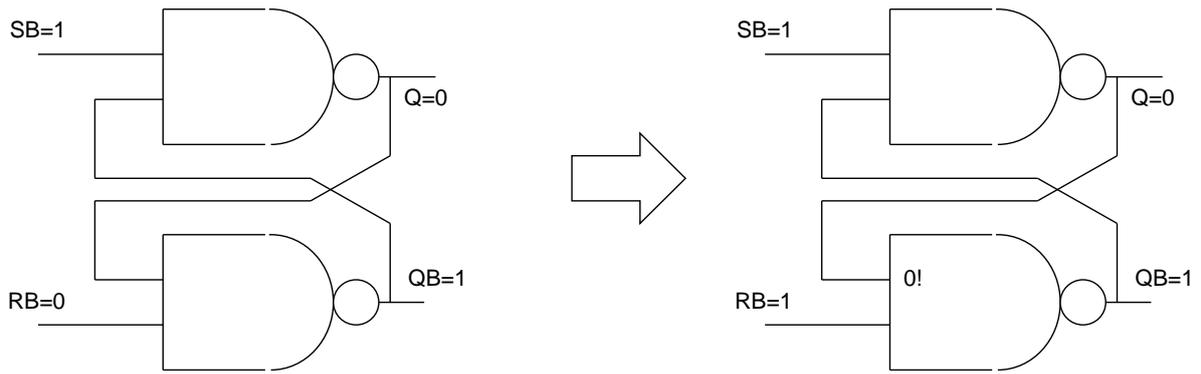
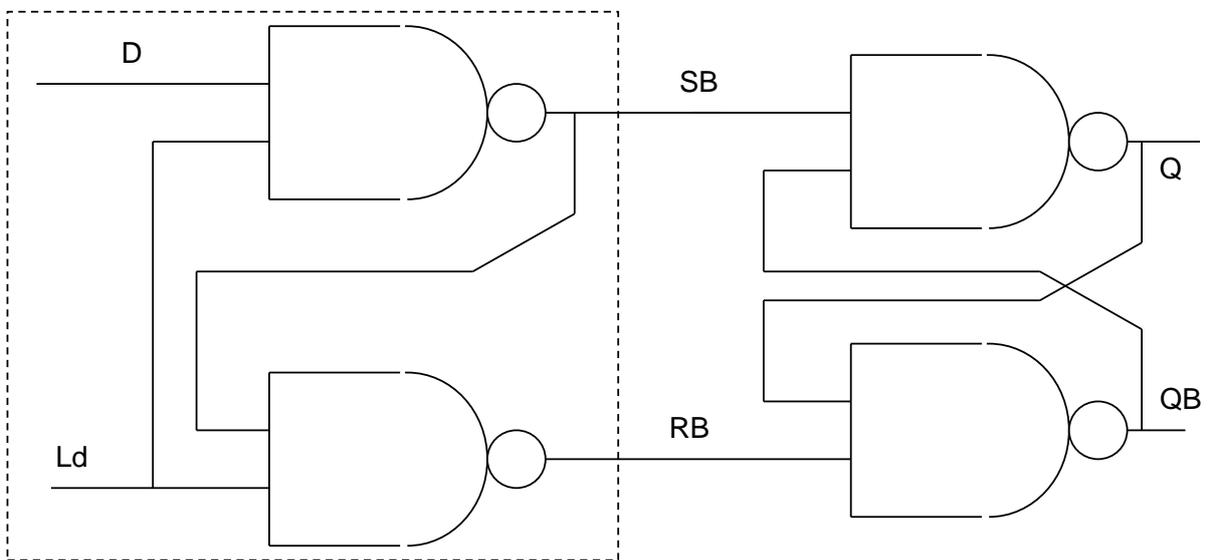


Abbildung 26: Reset

Basierend auf SR Latch kann man auch D-Latch realisieren (Abbildung 27). Die eingekreiste kombinatorische Schaltung erzeugt richtige SB und RB Werte, basierend auf D und Ld



Ld	D	SB	RB
0	0	1	1
0	1	1	1
1	0	1	0
1	1	0	1

Abbildung 27: D-Latch basierend auf SR-Latch

Man kann mit zwei dieser Latches ein master-slave Flipflop aufbauen.

Ein synchrones Toggle Flipflop mit T Eingang, wechselt seinen Zustand auf steigende Taktflanke wenn T=1 ist. Wenn T=0, wird der Zustand nicht gewechselt. Eine Prinzip-Schaltung in in Abbildung 28 gezeigt.

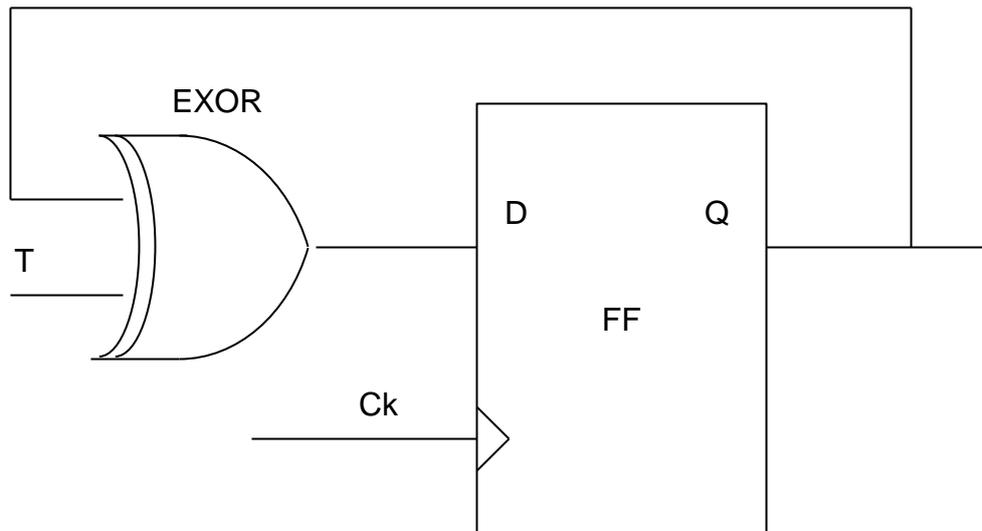


Abbildung 28: T-Flipflop

Latch als Differenzschaltung

Ein D-Latch kann auch mit Differenzverstärkern realisiert werden (Abbildung 29). Es handelt sich hier um eine Art Differenz-Stromlogik. Widerstand R kann als Polysilizium Widerstand oder mit Transistoren realisiert werden. Der PMOS Tpl dient als Widerstand und Stromquelle. Der NMOS Tnl verhindert, dass die Ausgangsspannung (Q oder QB) unterhalb $V_{DD} - V_{th}$ fällt. Tnl begrenzt somit die Potentialdifferenz zwischen logisch 0 und 1 (logic swing), was die Schaltung schneller macht.

Als Speicherelement wird ein kreuzgekoppeltes Differenzpaar (T1 und T2) verwendet (Abbildung 30). Man kann die Schaltung in gewünschten Zustand bringen wenn mit $Load = 1$ und $LoadB = 0$ der Strom im Differenzpaar (T1/T2) ausgeschaltet und das Paar T3/T4 aktiviert wird (Abbildung 31).

Die Schaltung hat einige Vorteile gegenüber dem CMOS Latch (Abbildung 14 und Abbildung 27).

1. Stromverbrauch ist konstant, es entstehen keine Störungen an Leitungsinduktivitäten
2. Schaltung ist schnell da logischer swing klein ist
3. Es werden weniger Transistoren benötigt, um Latch zu realisieren

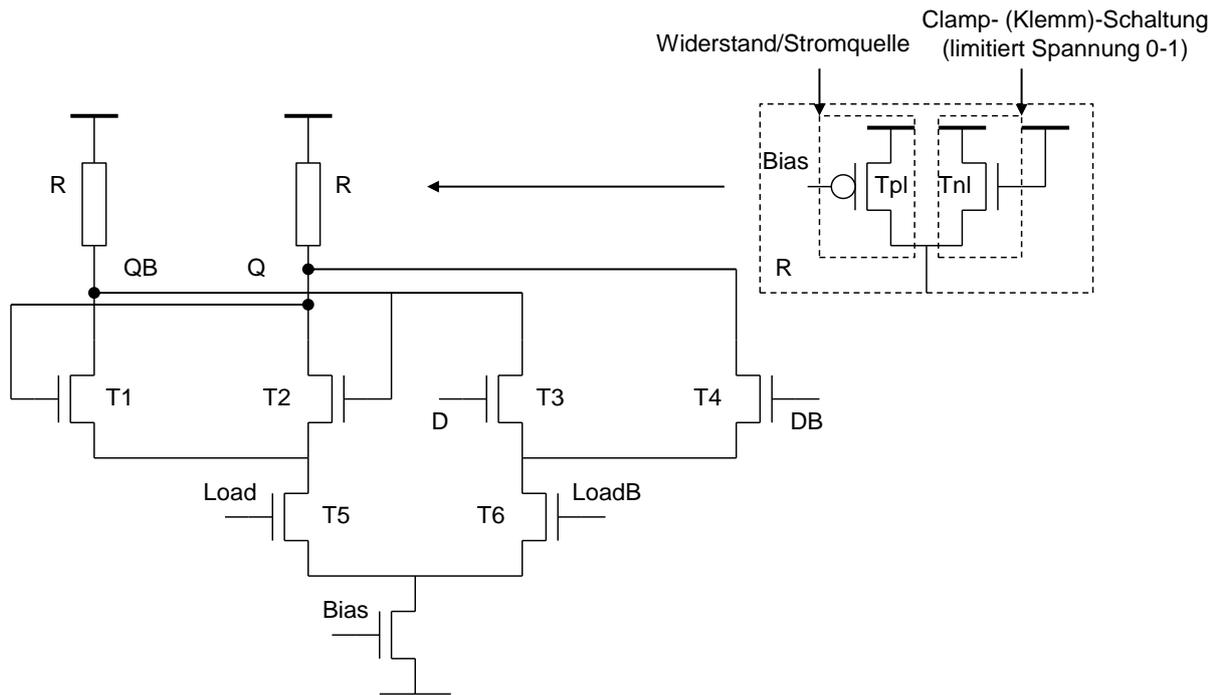


Abbildung 29: Latch mit Differenzverstärkern

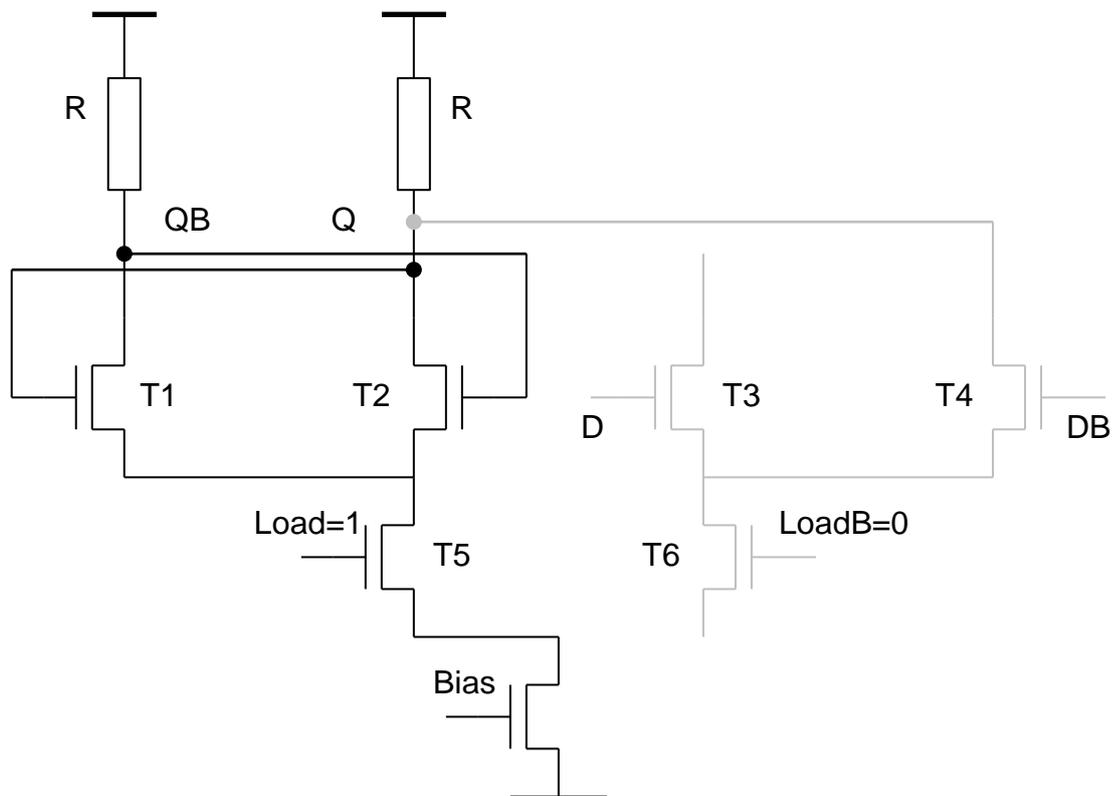


Abbildung 30: Speicherzustand

