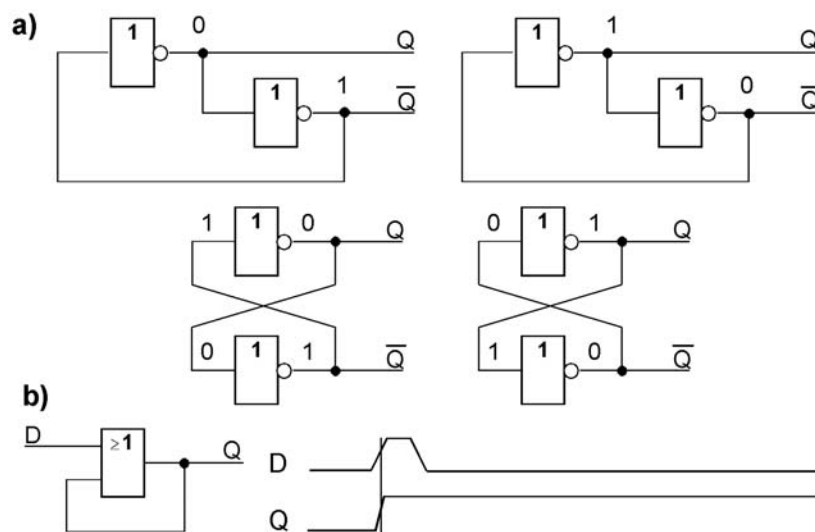


## 4. Speicherelemente

### 4.1 Latches und Flipflops

#### 4.1.1 Speicherung durch gesteuerte Selbsthaltung

Die Speicherelemente in modernen Digitalschaltungen beruhen auf dem Prinzip der Selbsthaltung. Hierbei werden Ausgangssignale auf Eingänge zurückgeführt. Abb. 4.1 veranschaulicht, dass sich durch Selbsthaltung stabile Speicherzustände ergeben.



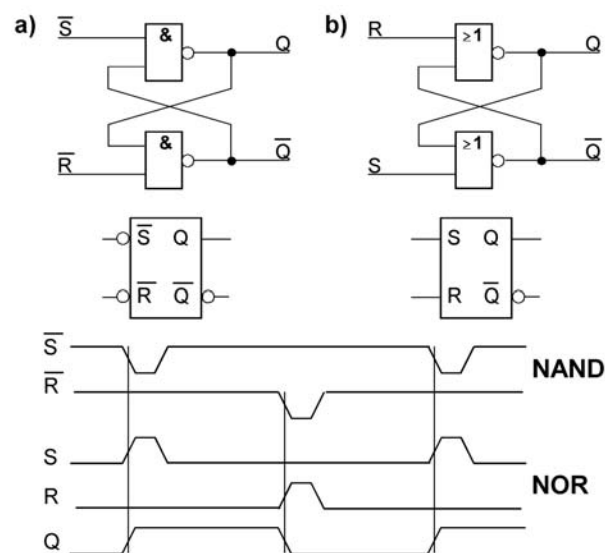
**Abb. 4.1** Speichern nach dem Prinzip der Selbsthaltung. a) zwei in Reihe geschaltete Inverter mit Rückführung. Es sind die beiden möglichen Signalbelegungen angegeben. Die untere Reihe zeigt die gleiche Schaltung in der allgemein üblichen Darstellung. Die überkreuzten Verbindungen sollen die Tatsache der Selbsthaltung besonders eindringlich hervorheben. Sie erwecken den Anschein, es handle sich um zwei Rückführungen. Tatsächlich gibt es aber – wie aus der oberen Reihe ersichtlich – nur eine. b) Selbsthaltung durch ODER-Verknüpfung. Diese Schaltung hält eine kurzzeitig an D angelegte Eins "ewig".

- Werden zwei Inverter in Reihe geschaltet und wird der Ausgang des zweiten auf den Eingang des ersten zurückgeführt, so sieht der erste Inverter die Ausgangsbelegung des zweiten. Da diese Belegung nach zweimaliger Negation wieder am Ausgang erscheint, ändert sich nichts; die Schaltung wird in einem von zwei möglichen Zuständen verharren.
- Wird der Ausgang eines ODER-Gatters auf einen seiner Eingänge zurückgeführt, so genügt es, am anderen Eingang kurzzeitig einen Einspegel anzulegen. Daraufhin wird das Gatter die Ausgangsbelegung über die Rückführung halten.

Beide Schaltungen halten zwar ihren jeweiligen Zustand, sind aber nicht steuerbar – und daher kaum zu gebrauchen. Die Steuerbarkeit kann durch Erzwingen bestimmter Ausgangsbelegungen oder durch zeitweises Trennen der Rückführung erreicht werden. Die erste Auslegung ergibt ein RS-Latch, die zweite ein D-Latch. "RS" steht für Rücksetzen und Setzen, "D" steht für Datenspeicherung.

#### 4.1.2 Latches zum Setzen und Rücksetzen (RS-Latches)

Um die Ausgangsbelegungen zu beeinflussen, werden anstelle der rückgekoppelten Inverter Gatter mit invertierenden Ausgängen eingesetzt. Um eine Eins zu speichern, ist der S-Eingang zu aktivieren (Setzen), um eine Null zu speichern, der R-Eingang (Rücksetzen). Die Eingänge R und S sind bei Einsatz von NAND-Gattern aktiv Low und bei Einsatz von NOR-Gattern aktiv High.



R	S	Q (NAND)	Q (NOR)
0	0	Verboten*	Q**
0	1	0	1
1	0	1	0
1	1	Q**	Verboten*

\*: S. Text. \*\*: Die bisherige Belegung bleibt erhalten.

**Abb. 4.2** RS-Latches. a) mit NAND-, b) mit NOR-Gattern. Von oben nach unten: Schaltungen – Schaltsymbole – Ablaufbeispiele – Funktionstabelle.

#### Der verbotene Zustand

Was geschieht, wenn R und S beide aktiv sind? Diese Belegung wird üblicherweise als "verboten" bezeichnet. So steht es auch in der Funktionstabelle. Tatsächlich kann man diese Belegung aber anlegen. Sie führt dazu, dass beide Ausgänge gleiche Pegel annehmen

(NANDs: High; NORs: Low). Das wird in manchen Schaltungen auch ausgenutzt. Es sind aber zwei Problemstellen zu beachten:

- In diesem Zustand führen beide Ausgänge die gleiche Belegung. Die eine Ausgangsbelegung ist dann nicht mehr die Negation der anderen. Werden beide Ausgangsbelegungen in nachgeordneten Schaltungsteilen ausgewertet, können Funktionsfehler auftreten.
- Was geschieht, wenn beide Eingangsbelegungen (R und S) gleichzeitig inaktiv werden? Das ist das eigentlich Verbotene. Bei inaktiven Eingängen liegt ein stabiler Speicherzustand nur dann, vor, wenn beide Ausgänge entgegengesetzt belegt sind. Aus einer Ausgangsbelegung 0, 0 oder 1, 1 wird die Schaltung auf irgend eine Weise in einen stabilen Zustand zurückfallen. Das kostet aber Zeit und kann mit unregelmäßigen Signalverläufen verbunden sein.

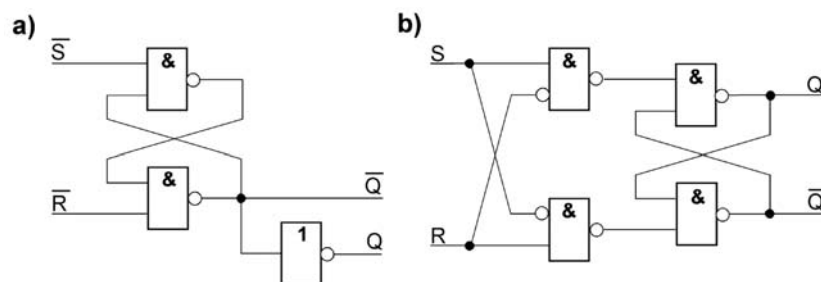
Wenn es vorkommen kann, dass die verbotene Signalbelegung anliegt, stehen zwei Schaltungsauslegungen zur Wahl:

### 1. Vorherrschende (dominierende) Eingänge

Da nicht mehr gilt, dass die eine Ausgangsbelegung die Negation der anderen ist, muss festgelegt werden, welches Eingangssignal mit Vorrang wirken soll. Das Ausgangssignal wird dann von jenem Gatter geliefert, das von diesem dominierendem Eingangssignal direkt beeinflusst wird. Eine ggf. erforderliche Invertierung ist anderweitig zu erledigen (z. B. mit einem zusätzlichen Negator).

### 2. Den bisherigen Zustand beibehalten (XOR-Latch)

Das Latch wird so angesteuert, dass der bisherige Zustand beibehalten wird.

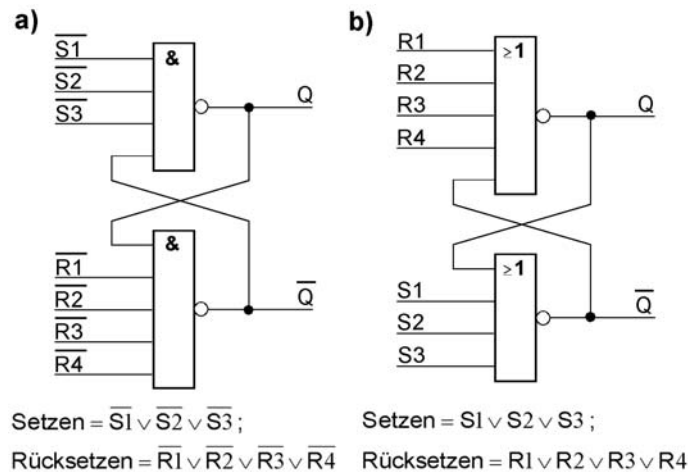


**Abb. 4.3** Latches, die mit der verbotenen Eingangsbelegung zurechtkommen. a) dominierende Wirkung. Im Beispiel soll das Rücksetzsignal (R) Vorrang haben. Deshalb ist das Ausgangssignal an jenem Gatter abzugreifen, das vom Rücksetzsignal direkt beeinflusst wird (Pfeil). b) Beibehalten des bisherigen Zustandes. Ein Zustandswechsel ist nur bei den Eingangsbelegungen  $S = 1, R = 0$  oder  $S = 0, R = 1$  erlaubt (XOR-Latch).

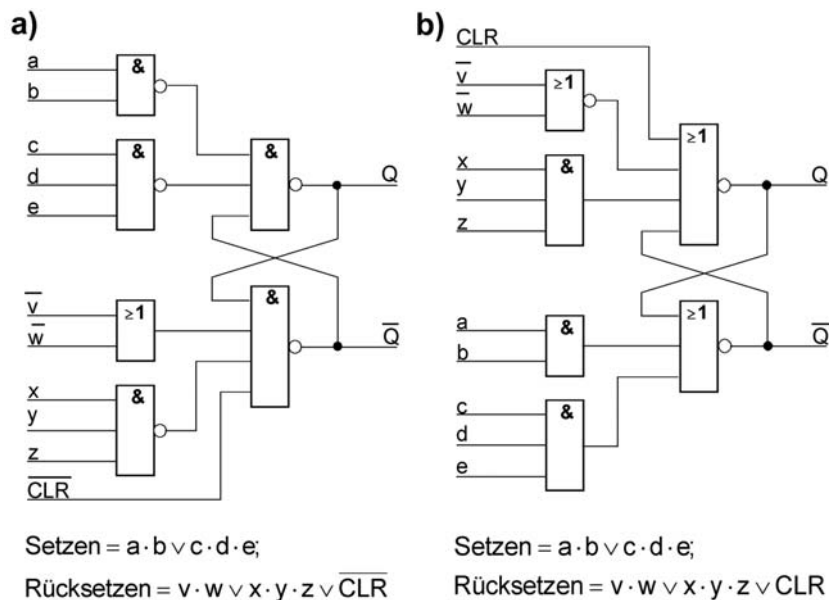
### Mehrere Setz- und Rücksetzeingänge

Die Gatter, die das RS-Latch bilden, können auf mehr als zwei Eingänge erweitert werden.

Diese zusätzlichen S- und R-Eingänge wirken disjunktiv (ODER-Verknüpfung). In manchen Anwendungen (vor allem in Steuerschaltungen) ergeben sich mehr oder weniger komplizierte Setz- und Rücksetzfunktionen. Werden diese in einer disjunktiven Normalform (SOP) dargestellt, kann die jeweilige disjunktive Verknüpfung in die Gatter des Latches einbezogen werden.



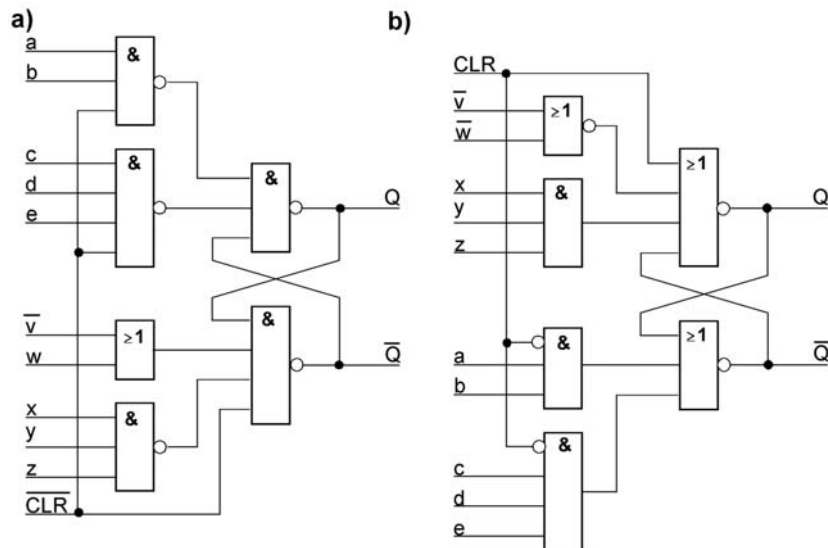
**Abb. 4.4** RS-Latches mit mehreren Setz- und Rücksetzeingängen. a) NAND mit invertiert, b) NOR mit direkt wirkenden Eingangssignalen.



**Abb. 4.5** Latches mit Setz- und Rücksetzfunktionen, die in disjunktiver Normalform dargestellt sind. a) NAND-Latch. UND-Verknüpfungen mit NAND-Gattern, invertierte Signale auch mit ODER-Gattern. Signale, die das Latch direkt beeinflussen (hier: CLR) sind aktiv Low. b) NOR-Latch. UND-Verknüpfungen mit UND-Gattern, invertierte Signale auch mit NOR-Gattern. Signale, die das Latch direkt beeinflussen (hier: CLR) sind aktiv High.

### Gesamt- oder Systemrücksetzen

Oft ist es notwendig, Latches oder Flipflops in einen Grundzustand zu versetzen. Dazu ist ein Rücksetzsignal vorzusehen. Soll dieses Signal dominierend wirken (d. h. den Grundzustand immer herbeiführen, gleichgültig welche anderen Signalbelegungen anliegen), kann man die Ausgangssignale so anschließen, wie in Abb. 4.3a dargestellt. Eine Alternative besteht darin, das Signal in alle konjunktiven Verknüpfungen der jeweils anderen Seite invertiert einzubeziehen.



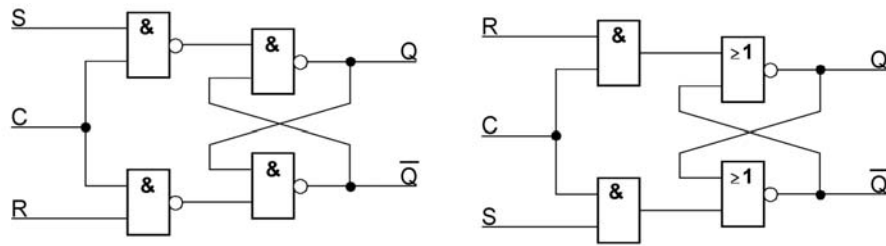
**Abb. 4.6** Latches mit dominierend wirkendem Initialisierungssignal. Im Beispiel soll das Signal CLR ein Rücksetzen herbeiführen. a) NAND-Latch. CLR wirkt aktiv Low. b) NOR-Latch. CLR wirkt aktiv High. In beiden Schaltungen ist CLR so in die UND-Verknüpfungen der Setzseite einbezogen, dass das Latch bei aktivem CLR nicht gesetzt werden kann, wie die anderen Signale belegt sind.

### Taktsteuerung von RS-Latches

Soll das Schalten eines RS-Latches nur bei Anliegen eines aktiven Taktsignals erlaubt sein, so ist das Taktsignal mit den Setz- und Rücksetzeingängen konjunktiv zu verknüpfen.

#### Vorrangiges Setzen oder Rücksetzen

Bei gemeinsamer Taktsteuerung über Setz- und Rücksetzeingänge ist der Zustand  $R = 1, S = 1$  wirklich verboten, da R und S gleichzeitig inaktiv werden, wenn das Taktsignal inaktiv wird. Sofern in der Anwendungsschaltung tatsächlich beide Signale zusammen mit dem Takt aktiv sein können, kann man durch Erweitern der konjunktiven Verknüpfungen eine Rangfolge festlegen, so dass das Latch entweder gesetzt oder rückgesetzt wird (Setz- oder Rücksetzdominanz).

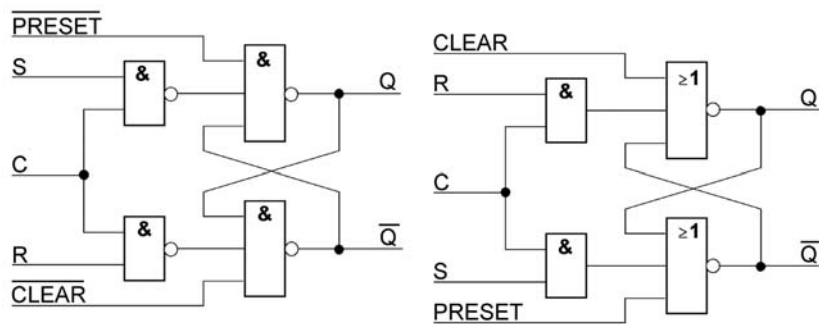


R	S	Q
0	0	Q**
0	1	1
1	0	0
1	1	Verboten*

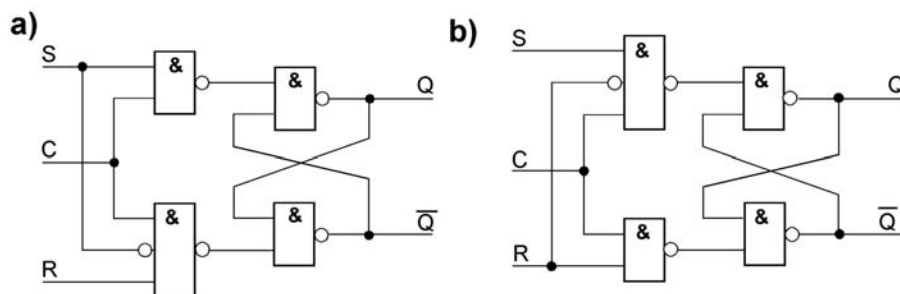
\*: S. Text.

\*\* : Die bisherige Belegung bleibt erhalten.

**Abb. 4.7** Taktgesteuerte RS-Latches.



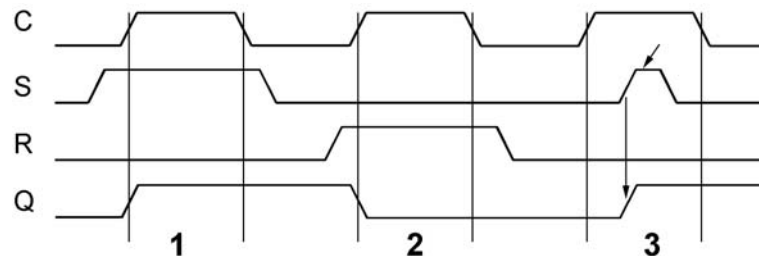
**Abb. 4.8** Taktgesteuerte RS-Latches mit zusätzlichen Initialisierungs- und Stelleingängen.



**Abb. 4.9** Vermeiden des verbotenen Zustandes. a) Setzdominanz. Bei aktivem S-Eingang wird das Rücksetzen verhindert. b) Rücksetzdominanz. Bei aktivem R-Eingang wird das Setzen verhindert.

*RS-Latches fangen Impulse*

Impulse am Setzeingang führt zum Setzen, Impulse am Rücksetzeingang zum Rücksetzen. RS-Latches reagieren aber auch auf Störimpulse.



**Abb. 4.10** Fehlerhaftes Schalten eines RS-Latches (Ablaufbeispiel). 1 - korrektes Setzen; 2 - korrektes Rücksetzen; 3 - ist das Taktsignal aktiv, so führt ein Störimpuls am Setzeingang (Pfeil) dazu, dass das Latch gesetzt wird. Infolgedessen verharrt der Ausgang Q auf Eins, obwohl eigentlich eine Nullbelegung erwartet wird.

### 4.1.3 Latches als Datenspeicher (D-Latches)

Um ein Datenbit zu speichern, braucht man einen Dateneingang (D) sowie eine Möglichkeit, das Übernehmen des anliegenden Datenwertes zu erlauben oder zu sperren. Hierzu wird typischerweise ein Erlaubnissignal vorgesehen, das zumeist als Takt- oder Torsignal bezeichnet wird (Enable E, Clock C, Gate G). D-Latches können durch Beschalten von RS-Latches oder nach dem Prinzip der trennbaren Rückführung aufgebaut werden.

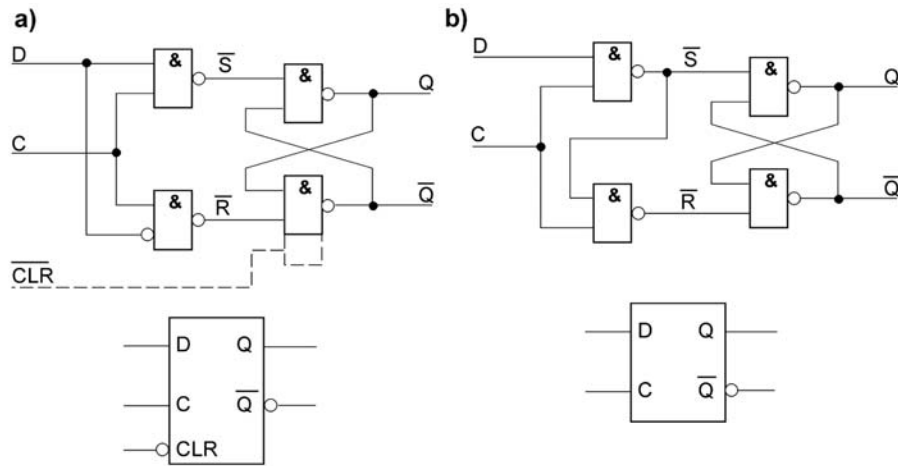
#### *D-Latches aus RS-Latches*

Die Eingangsbeschaltung ergibt sich unmittelbar aus den Anforderungen an die Funktionsweise. Das Latch ist bei aktivem Erlaubnissignal (Taktsignal) C zu setzen, wenn eine Eins, und zu löschen, wenn eine Null zu speichern ist.

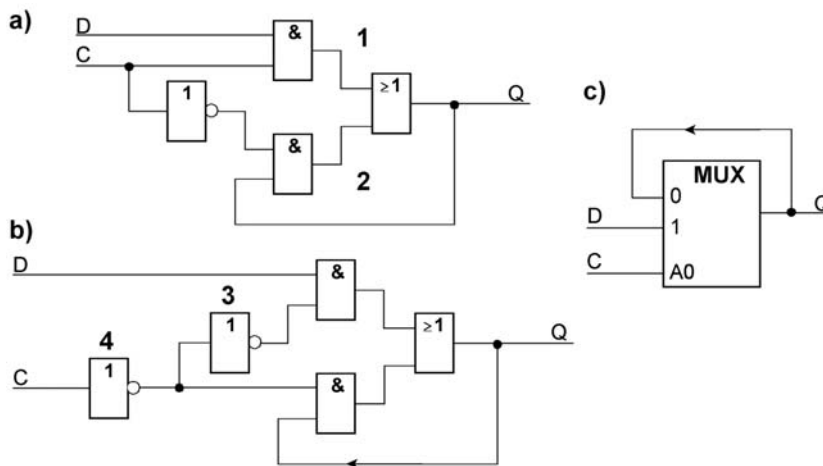
#### *D-Latches nach dem Prinzip der trennbaren Rückführung*

Das D-Latch ist im Grunde eine Auswahlerschaltung. Zur Datenübernahme wird der Dateneingang ausgewählt, zum Halten des gespeicherten Bits die Rückführung. Hierbei muss eine bestimmte Schaltfolge eingehalten werden: zuerst die Rückführung auswählen, dann die Weitergabe vom Dateneingang zum Datenausgang sperren.

Naive Auslegungen ähnlich Abb. 4.12a werden nicht immer funktionieren. Wenn das Taktsignal C nach Low schaltet, wird erst die Weitergabe vom Dateneingang zum Ausgang unterbrochen und dann die Rückführung geschlossen (Schaltverzögerung des Negators). Infolgedessen kann die Datenbelegung verlorengehen. Abb. 4.12b zeigt eine funktionsfähige Schaltung. Die Verzögerungszeit des Negators 3 wird ausgenutzt, um während des Umschaltens den Dateneingang weiterhin durchgesteuert zu halten. Soll das Taktsignal aktiv High wirken, ist ein zusätzlicher Negator 4 vorzuschalten.



**Abb. 4.11** Aus RS-Latches abgeleitete D-Latches. Oben Schaltungen, darunter typische Schaltsymbole. a) RS-Latch mit Eingangsbeschaltung. Zudem ist die Einordnung eines unabhängigen Löschesignals (CLR) angedeutet. b) Schaltungsvereinfachung (Einsparung der Invertierung des Datensignals).



**Abb. 4.12** D-Latches nach dem Prinzip der trennbaren Rückführung. a) nicht immer funktionsfähig; b) funktionsfähig; c) mit Multiplexer. 1 - Signalweg vom Dateneingang zum Ausgang (Takt signal = High); 2 - Datenspeicherung durch Rückführung (Takt signal = Low); 3 - Verzögerung; 4 - Invertierung des Taktsignals (s. Text).

### Multiplexer als Latches

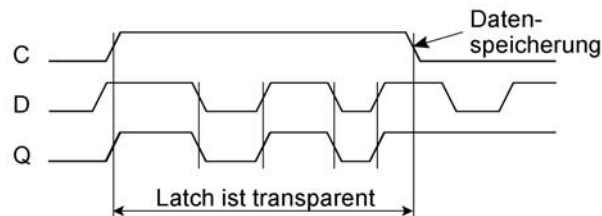
Die Anordnung von Abb. 4.12b ist an sich nichts anderes als ein 2-zu-1-Multiplexer. Das Auswahl signal dient als Takt signal:

- Takt = 1: der Dateneingang D wird zum Ausgang durchgeschaltet; die Ausgangsbelegung folgt der Eingangsbelegung (das Latch verhält sich als Durchreiche).
- Takt = 0: der Ausgang ist auf den ausgewählten Eingang zurückgeführt; die jeweils letzte Belegung wird gespeichert (Selbsthaltung). Auch hier muss gewährleistet sein, dass erst die Rückführung geschlossen und dann die Datenweitergabe getrennt wird.

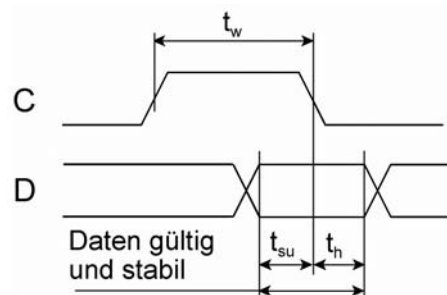


*D-Latches sind transparent*

Bei aktivem Taktsignal erscheint der Signalverlauf am Dateneingang unverändert am Ausgang. Das D-Latch ist dann durchlässig (transparent). Die Zeitkennwerte von Takt und Daten werden auf High-Low-Flanke des Taktsignals bezogen.



**Abb. 4.13** Takt und Daten beim D-Latch. Solange das Taktsignal aktiv ist (High-Pegel), erscheint der Signalverlauf am Dateneingang unverändert am Datenausgang. Wenn das Taktsignal von High nach Low schaltet, wird die Datenbelegung gespeichert, die am Eingang anliegt,.



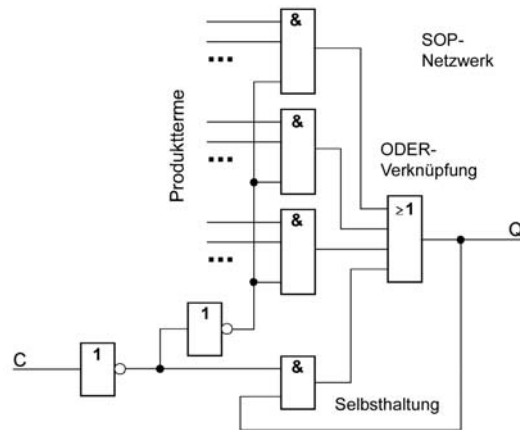
**Abb. 4.14** Zeitkennwerte von Takt und Daten.  $t_w$  - minimale Taktimpulsdauer;  $t_{su}$  - minimale Vorhaltezeit;  $t_h$  - minimale Haltezeit.

*D-Latches und kombinatorische Netzwerke im Verbund*

In Schaltungen, die aus kombinatorischen Netzwerken und nachgeordneten D-Latches bestehen, kann man die Selbsthaltefunktion in die ODER-Verknüpfungen der SOP-Netzwerke einbeziehen. Dann hat das Latch keine zusätzliche Verzögerungszeit; die Durchlaufverzögerung bis zum Ausgang des Latches entspricht der Verzögerungszeit des kombinatorischen Netzwerks..

*D-Latches sind unempfindlich gegen Störimpulse*

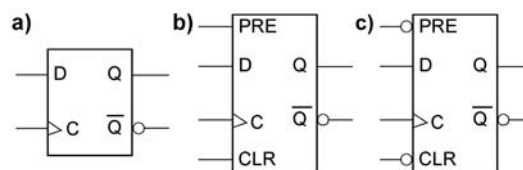
Das D-Latch fixiert die Belegung des Dateneingangs dann, wenn das Taktsignal inaktiv wird. Störimpulse am Dateneingang erscheinen bei aktivem Taktsignal zwar am Ausgang, werden aber – im Gegensatz zum RS-Latch – nicht gefangen. Ein zum D-Latch erweitertes RS-Latch wird zwar durch einen Störimpuls zunächst so gesetzt wie in Abb. 4.10 gezeigt. Es wird aber sofort wieder zurückgesetzt, wenn der Störimpuls verschwindet.



**Abb. 4.15** Ein Verbund aus D-Latch und SOP-Netzwerk. Bei aktivem Taktsignal C erscheint die Datenbelegung nach Durchlauf des SOP-Netzwerks ohne weitere Verzögerung am Ausgang Q.

#### 4.1.4 Flipflops

Ein Flipflop ist ein taktgesteuerter 1-Bit-Speicher, der seine Ausgangsbelegung nur in Bezug auf eine jeweils spezifizierte Taktflanke ändert, d. h. auf die ansteigende oder abfallende Flanke des Taktimpulses. Manche Flipflops haben zusätzliche, unabhängig vom Takt wirkende Setz- und Rücksetzeingänge (Preset, Clear). Das einfachste Flipflop ist das D-Flipflop. Es ist ebenso ein Datenspeicher wie das D-Latch. Die Datenbelegung, die am D-Eingang anliegt, wird aber nur dann gespeichert, wenn das Taktsignal beispielsweise von Low nach High schaltet. Außerhalb dieses Zeitpunktes haben beliebige Änderungen am Dateneingang keinen Einfluss auf den Ausgang. Die Zeitkennwerte von Takt und Daten werden auf die betreffende Flanke des Taktsignals bezogen.



**Abb. 4.16** Schaltsymbole von D-Flipflops. a) die einfachste Ausführung; b), c) mit zusätzlichen asynchronen Setz- und Rücksetzeingängen (PRE = Preset, CLR = Clear). b) Setz- und Rücksetzeingänge aktiv High (typische Funktionselemente in programmierbaren Schaltkreisen); c) Setz- und Rücksetzeingänge aktiv Low (typische Flipflopschaltkreise (SSI)).

#### Flipflops als synchrone Automaten

Flipflops sind die einfachsten taktgesteuerten (synchrone) Automaten. Sie können u. a. durch Automaten Gleichungen, Automatentabellen und Zustandsdiagramme (Zustandsgraphen) beschrieben werden. Die Zustandsmenge des Flipflops umfasst nur zwei Zustände, die man dem Ausgangssignal (Q) direkt zuordnen kann:  $Q = 0$  und  $Q = 1$ . Das Verhalten eines Flipflops hängt von der Belegung der Eingänge und von seinem aktuellen Zustand ab. Die

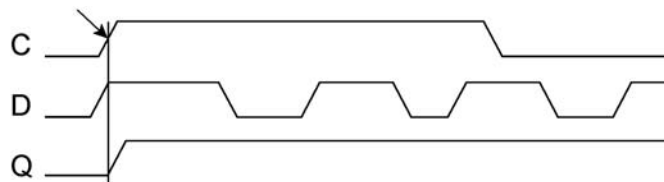
Verhaltensbeschreibung gibt an, wie der im jeweils nächsten Taktzeitpunkt einzunehmende Zustand  $Q(t+1)$  in Abhängigkeit von der Belegung der Eingänge  $E(t)$  und vom Zustand im aktuellen Taktzeitpunkt  $Q(t)$  gebildet wird (Automatengleichung, Überföhrungsfunktion):

$$Q(t+1) = f(E(t), Q(t))$$

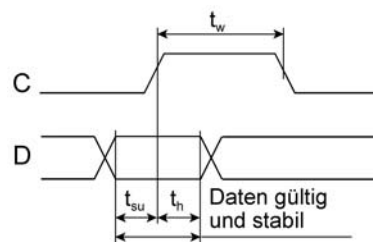
In vereinfachter Schreibweise:

$$Q^1 = f(E, Q) \quad (4.1)$$

Die schaltungstechnische Verwirklichung dieser Funktion läuft darauf hinaus, einem taktflankengesteuerten Speicherglied eine kombinatorische Eingangsbeschaltung vorzuordnen, die die eigentlichen Eingänge mit dem Flipflopausgang verknüpf.



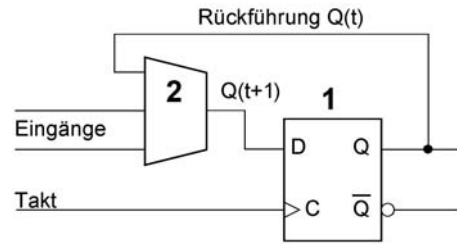
**Abb. 4.17** Taktflankensteuerung eines D-Flipflops (Ablaufbeispiel; vgl. mit Abb. 4.13). Die Datenbelegung, die am D-Eingang anliegt, wird nur dann übernommen, wenn das Taktsignal von Low nach High schaltet. Außerhalb dieses Zeitpunktes haben beliebige Änderungen am Dateneingang keinen Einfluss auf den Ausgang.



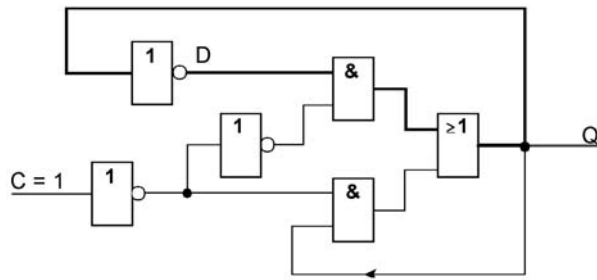
**Abb. 4.18** Zeitkennwerte von Takt und Daten.  $t_w$  - minimale Taktimpulsdauer;  $t_{su}$  - minimale Vorhaltezeit;  $t_h$  - minimale Haltezeit. Viele Flipflops haben eine Haltezeit von Null. Dann darf sich die Belegung des D-Eingangs sofort nach der Datenübernahme ändern.

*Weshalb fällt das Latch nicht unter diese Definition?*

Weil man die Ausgänge eines Latches nicht über kombinatorische Verknüpfungen auf dessen Eingänge zurückföhren darf. Sofern die Überföhrungsfunktion eine Änderung des aktuellen Zustandes ergibt ( $Q(t+1) = \overline{Q}$ ), würde die Anordnung bei aktivem Takt ins Schwingen geraten. Man kann jedoch Flipflops aus zwei hintereinandergeschalteten Latches aufbauen, die mit zeitversetzten Taktimpulsen angesteuert werden.



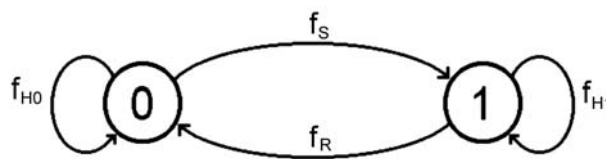
**Abb. 4.19** Das Flipflop als synchroner Automat. 1 - D-Flipflop (Datenspeicher); 2 - Schaltnetz der Überföhrungsfunktion.



**Abb. 4.20** Die Ausgange eines Latches durfen nicht auf dessen Eingange zuruckgeföhrt werden. Aus dem Beispiel ist ersichtlich, dass die Anordnung ins Schwingen kommt, solange das Taktsignal C aktiv ist. Mit  $Q = 0$  ergibt sich  $D = 1$  und somit  $Q = 1$ , dies föhrt wieder zu  $D = 0$  und somit  $Q = 0$  usw. Die Ruckföhruug eines invertierten Signals entspricht einer Phasenverschiebung um  $180^\circ$  (Selbsterregungsbedingung).

**Das Grundmodell der Zustandsübergange**

Da es nur zwei Zustande gibt, ist ohne weiteres zu uberblicken, welche Zustandsübergange uberhaupt moglich sind.



Aktueller Zustand Q	Folgezustand Q'	Übergang	Schaltfunktion
0	0	Zustand 0 wird gehalten.	$f_{H0}$ (Haltefunktion Zustand 0)
0	1	Zustand 0 wird verlassen (Übergang zum Zustand 1).	$f_S$ (Setzfunktion)
1	1	Zustand 1 wird gehalten.	$f_{H1}$ (Haltefunktion Zustand 1)
1	0	Zustand 1 wird verlassen (Übergang zum Zustand 0).	$f_R$ (Rucksetzfunktion)

**Abb. 4.21** Die Zustandsübergange der Flipflops.

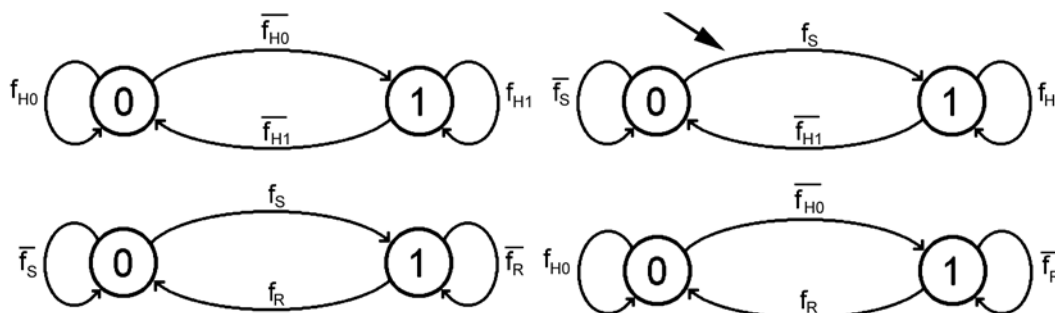
Die in Abb. 4.21 angegebenen Schaltfunktionen sind Funktionen der Eingangsvariablen. Da es in jedem Zustand nur zwei Möglichkeiten gibt – Halten oder Verlassen –, kommt man mit zwei Funktionen aus; die jeweilige Haltefunktion ist die Negation der Setz- oder Rücksetzfunktion und umgekehrt. In der Überföhrungsfunktion (4.1) ergibt ein erföller Funktionsausdruck den Folgezustand 1, ein nicht erföller den Folgezustand 0. Deshalb liegt es nahe, die Zustandsübergänge ausschlieÖlich mit der Setzfunktion  $f_S$  und der Haltefunktion  $f_{HI}$  zu beschreiben:

$$Q^1 = f_S \cdot \bar{Q} \vee f_{HI} \cdot Q \quad (4.2)$$

Ist der Folgezustand 1 unabhangig vom aktuellen Zustand herbeizuföhren (Setzen des Flipflops), so müssent gemaß (4.2) die betreffenden Bedingungen sowohl in die Setz- als auch in die Haltefunktion einflieÖen. Manchmal ergeben sich besser überschaubare Ausdröcke, wenn man die Überföhrungsfunktion in drei Teilfunktionen zerlegt<sup>1)</sup>:

$$Q^1 = f_{S0} \cdot \bar{Q} \vee f_{HI} \cdot Q \vee f_{S1} \quad (4.3)$$

Diese Gleichung beschreibt die Struktur eines universellen Flipflops.

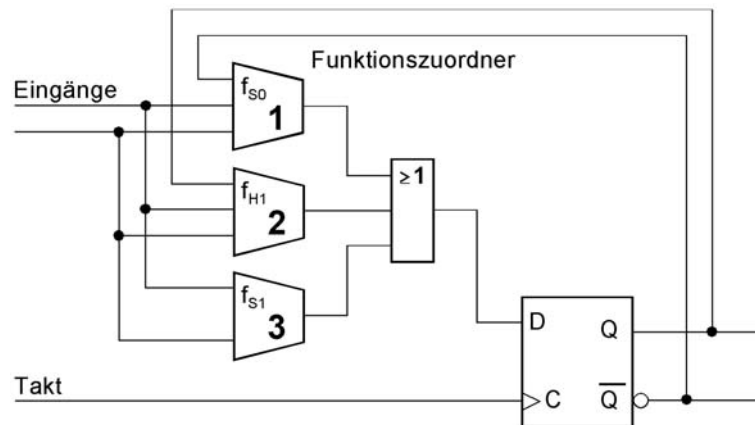


**Abb. 4.22** Es genügen zwei Schaltfunktionen, um die Zustandsübergänge zu beschreiben. Der Pfeil zeigt auf den Zustandsgraphen, der der Überföhrungsfunktion (4.2) entspricht.

### Flipflop-Arten

En Flipflop hat nur zwei Zustände (0 oder 1). Begnügt man sich – wie beim RS-Latch – mit zwei Eingängen, so hängt der Folgezustand von drei Binarvariablen ab. Mit drei Variablen kann man  $2^3 = 256$  verschiedene Schaltfunktionen bilden. Es kann also nicht mehr als 256 verschiedene Flipflops mit zwei Eingangen geben (darunter sind allerdings auch die Festwerte 0 und 1 sowie alle kombinatorischen Schaltfunktionen, die von einer und von zwei Variablen abhangen, und andere hier nicht besonders brauchbare Funktionselemente). In der Praxis beschrankt man sich auf nur wenige Typen.

1: In (4.3) kann es Teilfunktionen geben, die konstant = 0 sind.



**Abb. 4.23** Die grundsätzliche Struktur eines universellen Flipflops . gemäß (4.3).1 - Setzfunktion zum Verlassen des Zustands 0; 2 - Haltefunktion in Zustand 1 (= Negation des Rücksetzens); 3 - allgemeine Setzfunktion (Erzwingen des Zustands 1).

#### *Automatentabellen und Wahrheitstabellen*

Die Automatentabelle (Funktionstabelle) gibt für jede Eingangsbelegung den zugehörigen Folgezustand  $Q^1$  an. Dabei kann jeweils einer von vier möglichen Funktionswerten zugewiesen werden: 0, 1, der aktuelle Zustand  $Q$  und der invertierte aktuelle Zustand  $\bar{Q}$ . Hieraus ist sofort erkennbar, ob der Zustand zugewiesen, geändert oder gehalten wird. Die Wahrheitstabelle weist dem Folgezustand  $Q^1$  jeweils nur einen der Werte 0 oder 1 zu, und zwar in Abhängigkeit von den Eingangsbelegungen und vom aktuellen Zustand  $Q$ . Diese Darstellung ist nicht so übersichtlich wie die Automatentabelle, erlaubt es aber, die Schaltfunktionen unmittelbar abzulesen.

#### *D-Flipflop*

Das D-Flipflop hat das einfachste Verhalten aller Flipflops. Es übernimmt lediglich die am Dateneingang anliegende Belegung (Datenspeicher- oder Verzögerungswirkung; D = Data oder Delay). Die Überföhrungsfunktion:

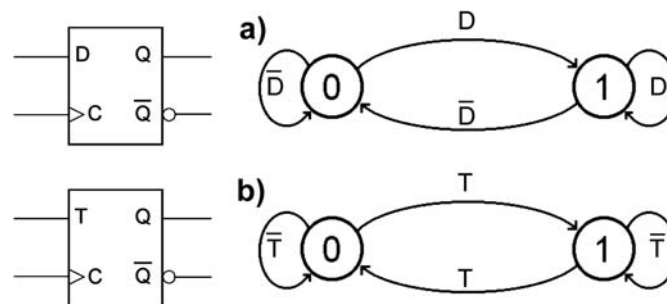
$$Q^1 = D \quad (4.4)$$

Für D kann jede beliebige Schaltfunktion eingesetzt werden. Somit eignet sich das D-Flipflop als Funktionselement zum Aufbau beliebiger Zustandsautomaten, darunter aller anderen Flipflops. Hierzu ist lediglich die gewünschte Überföhrungsfunktion als kombinatorische Schaltung darzustellen, die dem D-Eingang vorgeschaltet wird. Deshalb sind die Flipflops in programmierbaren Schaltkreisen typischerweise als D-Flipflops ausgelegt.

#### *T-Flipflop*

Das T-Flipflop ) wechselt bei aktivem Eingangssignal T mit jedem Takt seinen Zustand (T = Toggle). Bei inaktivem T hält es den bisherigen Zustand. Die Überföhrungsfunktion:

$$Q^1 = T \oplus Q \quad (4.5)$$



**Abb. 4.24** Flipflops mit einem Eingang. a) D-Flipflop; b) T-Flipflop.

D, T	a) $Q^1 D$	b) $Q^1 T$
0	0	Q
1	1	$\bar{Q}$

**Tabelle 4.1** Die Automatentabellen der Flipflops mit einem Eingang. a) D-Flipflop; b) T-Flipflop.

D, T	Q	a) $Q^1 D$	b) $Q^1 T$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

**Tabelle 4.2** Die Wahrheitstabellen der Flipflops mit einem Eingang. Grau hinterlegte Felder kennzeichnen das Halten des bisherigen Zustands ( $Q^1 = Q$ ). a) D-Flipflop; b) T-Flipflop.

### RS-Flipflop

Das RS kann durch Aktivieren des S-Eingangs gesetzt und durch Aktivieren des R-Eingangs zurückgesetzt werden. Sind beide Eingangssignale inaktiv, hält es den bisherigen Zustand (Speicherfunktion). Die Belegung  $R = S = 1$  wird in der Theorie ausgeschlossen ( $R \cdot S = 0$  als Nebenbedingung der Überföhrungsfunktion). Ob  $R = S = 1$  in der Schaltungspraxis ein tatsächlich verbotener Zustand ist, hängt vom Aufbau des Flipflops ab. Wird das RS-Verhalten durch kombinatorische Beschaltung gemäß eine der folgenden Überföhrungsfunktionen (4.6) oder (4.7) realisiert, ergeben sich Vorrangwirkungen (Setz- oder Rücksetzdominanz), die man im Schaltungsentwurf ausnutzen kann.

*RS-Flipflop mit Setzdominanz*

Die übliche Überföhrungsfunktion des RS-Flipflops:

$$Q^1 = S \vee Q \cdot \bar{R} \quad (4.6)$$

Gemäß dieser Gleichung ergibt die Eingangsbelegung 1, 1 den Folgezustand 1. Das Setzen hat also Vorrang vor dem Rücksetzen.

*RS-Flipflop mit Rücksetzdominanz*

Soll das Rücksetzen Vorrang vor dem Setzen haben, muss die Eingangsbelegung 1,1 den Folgezustand 0 ergeben. Das kann mit folgender Überföhrungsfunktion erreicht werden:

$$Q^1 = S \cdot \bar{R} \vee Q \cdot \bar{R} \quad (4.7)$$

*JK-Flipflop*

Das JK-Flipflop<sup>2)</sup> ist ein abgewandeltes RS-Flipflop. J entspricht S, K entspricht R. Der Zustand J = K = 1 ist erlaubt. Liegt diese Belegung an, verhält sich das JK-Flipflop wie ein T-Flipflop (Zustandswechsel). Die Überföhrungsfunktion:

$$Q^1 = \bar{Q} \cdot J \vee Q \cdot \bar{K} \quad (4.8)$$

*DE-Flipflop*

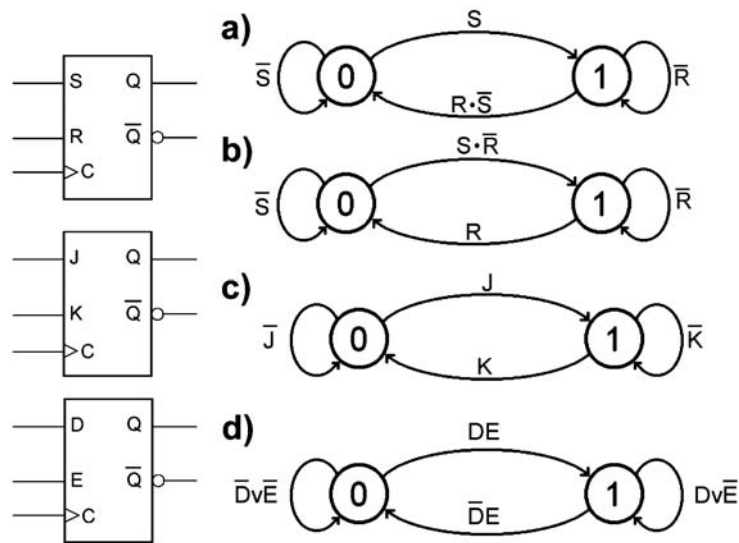
Das DE-Flipflop ist ein D-Flipflop mit einem zusätzlichen Erlaubniseingang E (in Schaltsymbolen zumeist als CE = Clock Enable bezeichnet). Dessen Wirkung: nur bei aktivem E wird die Eingangsbelegung übernommen, ansonsten wird der bisherige Zustand gehalten. Die DE-Funktion betrifft das Umschalten zwischen dem Ausführen der jeweiligen "eigentlichen" Funktion (z. B. Datenübernahme (Laden), Schieben oder Zählen) und dem Nichtstun ohne Datenverlust (= Halten des aktuellen Zustands). Deshalb ist das DE-Flipflop die Grundform des universellen Flipflops in vollsynchron arbeitenden Schaltungen. Die Überföhrungsfunktion:

$$Q^1 = \bar{E} \cdot Q \vee E \cdot D \quad (4.9)$$

---

2: Historische Anmerkung: An sich kann man dem verbotenen Zustand (R = S = 1) des RS-Flipflops beliebige Folgezustände zuordnen. JK-Flipflops kamen mit den ersten Digitalschaltkreisen auf. Es ging seinerzeit darum, Universalbauelemente mit vielseitigen Einsatzmöglichkeiten anzubieten. In den 60er Jahren wurden die Flipflops vor allem in Zähl- und in Steuerschaltungen eingesetzt (u. a. beruhten die numerischen Werkzeugmaschinensteuerungen jener Zeit auf kunstvoll verschalteten Zähleranordnungen – echte Computer waren viel zu teuer). JK kann man als eine Art Marketing-Idee ansehen, um beiden Hauptanwendungen jener Zeit gerecht zu werden (Toggle-Funktion zum Zählen, RS-Funktion zu Steuerungszwecken).





**Abb. 4.25** Flipflops mit zwei Eingängen. a) RS-Flipflop (Setzdominanz); b) RS-Flipflop (Rücksetzdominanz); c) JK-Flipflop; d) DE-Flipflop.

S, J, D	R, K, E	a) $Q^1$ RS	b) $Q^1$ RS	c) $Q^1$ JK	d) $Q^1$ DE
0	0	Q	Q	Q	Q
0	1	0	0	0	0
1	0	1	1	1	Q
1	1	1	0	$\bar{Q}$	1

**Tabelle 4.3** Die Automatentabellen der Flipflops mit zwei Eingängen. a) RS-Flipflop (Setzdominanz); b) RS-Flipflop (Rücksetzdominanz); c) JK-Flipflop; d) DE-Flipflop.

S, J, D	R, K, E	Q	$Q^1$ RS*	a) $Q^1$ RS	b) $Q^1$ RS	c) $Q^1$ JK	d) $Q^1$ DE
0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	0
1	0	1	1	1	1	1	1
1	1	0	–	1	0	1	1
1	1	1	–	1	0	0	1

a\*: Nebenbedingung  $R \cdot S = 0$  (verbotene Eingangsbelegungen).

**Tabelle 4.4** Die Wahrheitstabellen der Flipflops mit zwei Eingängen. Grau hinterlegte Felder kennzeichnen das Halten des bisherigen Zustands ( $Q^1 = Q$ ). a) RS-Flipflop (Setzdominanz); b) RS-Flipflop (Rücksetzdominanz); c) JK-Flipflop; d) DE-Flipflop.

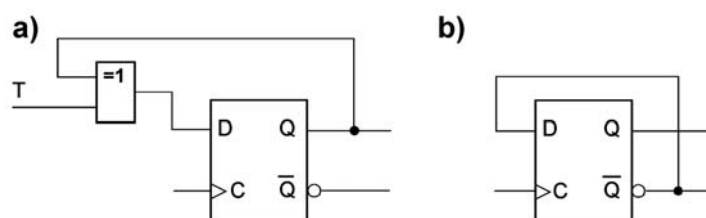
### Wechselseitigen Wandlungen

Alle rückkopplungsfähigen Flipflops lassen sich durch Zusatzbeschaltung in jede beliebige andere Art umwandeln. Die Schaltgleichungen ergeben sich, indem man die Überföhrungsfunktionen gleichsetzt ( $Q^1$  (Ziel-Flipflop) =  $Q^1$  (Realisierungsbasis)) und die so gewonnene Boolesche Gleichung nach den Ansteuervariablen des Flipflops auflöst, das als Realisierungsbasis dienen soll. Alternativ dazu kann man sie aus der Wahrheitstabelle des Ziel-Flipflops ablesen, wenn man deren Einträge gemäß der Funktionsweise der Realisierungsbasis auswertet. Als Realisierungsbasis kommen vor allem in Betracht:

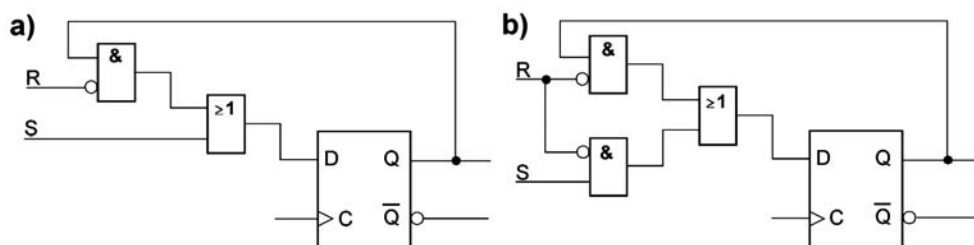
- Als Bauelemente (SSI): D-Flipflops und JK-Flipflops.
- In programmierbaren Schaltkreisen: D-Flipflops und T-Flipflops.

#### D-Flipflops

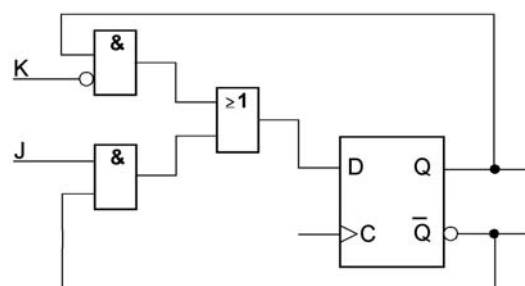
Die jeweilige Funktion ergibt sich, indem man dem D-Eingang ein Schaltnetz vorordnet, das der Überföhrungsfunktion des gewünschten Flipfloptyps entspricht.



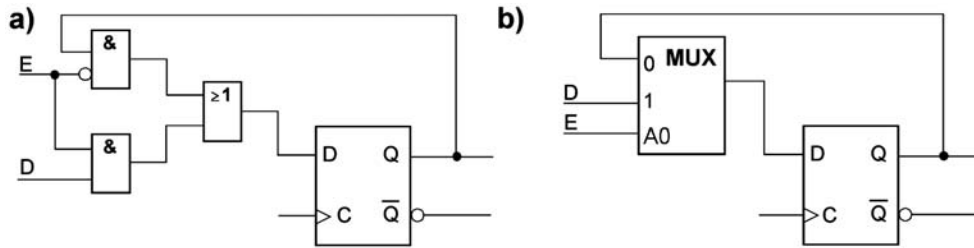
**Abb. 4.26** T-Flipflop mit D-Flipflop. a) gesteuert (vgl. (4.5)); b) ungesteuert (2:1-Teiler).



**Abb. 4.27** RS-Flipflop mit D-Flipflop. a) Setzdominanz (vgl. (4.6)); b) Rücksetzdominanz (vgl. (4.7)).



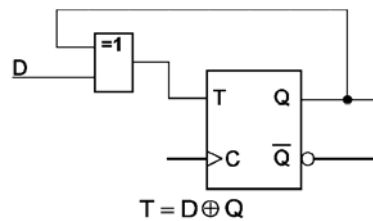
**Abb. 4.28** JK-Flipflop aus D-Flipflop (vgl. (4.8)).



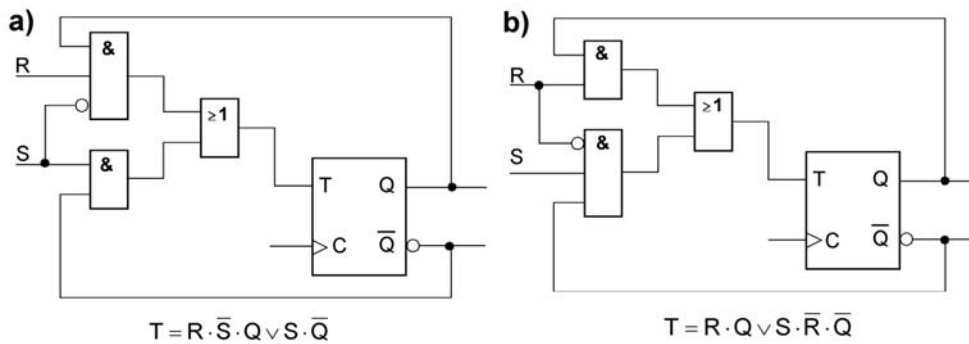
**Abb. 4.29** DE-Flipflop aus D-Flipflop (vgl. (4.9)). a) mit zweistufigem Schaltnetz; b) mit 2-zu-1-Multiplexer.

*T-Flipflops*

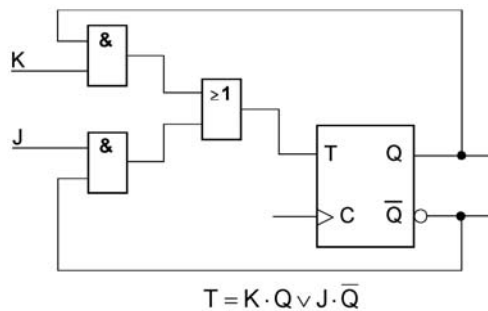
Das Ansteuersignal T muss dann aktiv werden, wenn sich der Folgezustand vom aktuellen Zustand unterscheidet. Um die Schaltgleichungen aufzustellen, genügt es, in der Wahrheitstabelle des Ziel-Flipflops die Zeilen aufzusuchen, in denen  $Q \neq Q^1$  ist.



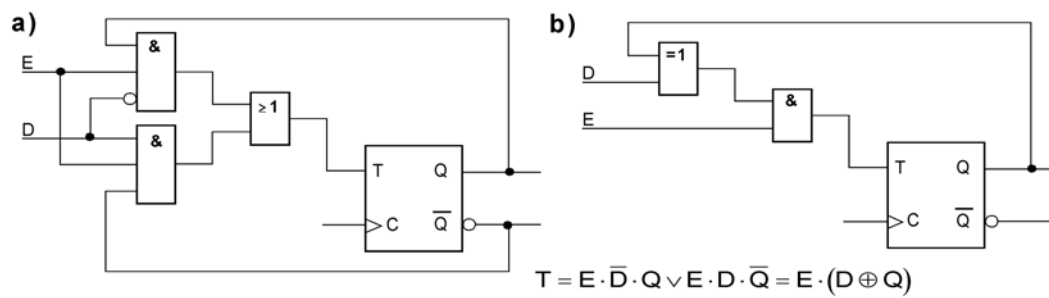
**Abb. 4.30** D-Flipflop aus T-Flipflop.



**Abb. 4.31** RS-Flipflop aus T-Flipflop. a) Setzdominanz; b) Rücksetzdominanz.



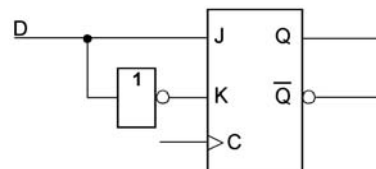
**Abb. 4.32** JK-Flipflop aus T-Flipflop.



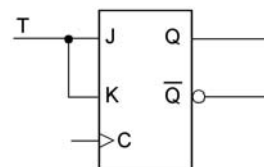
**Abb. 4.33** DE-Flipflop aus T-Flipflop. a) mit zweistufigem Schaltnetz; b) mit XOR-Verknüpfung.

### JK-Flipflops

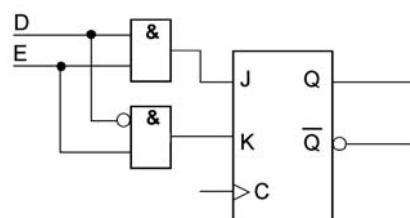
Das JK-Flipflop kann auf einfache Weise in ein D-, DE- oder T-Flipflop gewandelt werden. Es ist direkt als RS-Flipflop nutzbar ( $S = J$ ,  $R = K$ ), wenn die Belegung  $R = S = 1$  nie auftritt. Ansonsten muss diese Belegung durch Zusatzbeschaltung ausgeschlossen werden.



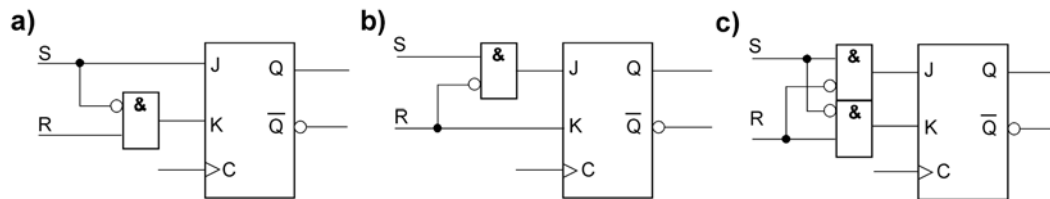
**Abb. 4.34** D-Flipflop aus JK-Flipflop.



**Abb. 4.35** T-Flipflop aus JK-Flipflop.



**Abb. 4.36** DE-Flipflop aus JK-Flipflop.



**Abb. 4.37** RS-Flipflop aus JK-Flipflop. a) Setzdominanz; b) Rücksetzdominanz.

### Den Anfangszustand erzwingen: Setzen und Rücksetzen

Manchmal ist es erforderlich, Flipflops in einen bestimmten Grundzustand zu versetzen, und zwar unabhängig von der Belegung der Steuersignaleingänge. Der häufigste Einsatzfall: das Einschaltrücksetzen (Power-on Reset). Der Grundzustand kann asynchron oder synchron erzwungen werden.

#### *Asynchrones Setzen und Rücksetzen*

Typische Bezeichnungen: Setzen = Preset, Rücksetzen = Clear. Diese Signale wirken unabhängig vom Takt; sie haben Vorrang vor den synchronen (vom Takt ausgelöst) Zustandsänderungen. Solche Setz- und Rücksetzfunktionen müssen in der Innenschaltung des Flipflops vorgesehen sein; man kann sie nicht durch Außenbeschaltung hinzufügen. Hat das Flipflop sowohl Setz- als auch Rücksetzeingänge, sind folgende Auslegungen üblich:

- Zu einer Zeit darf nur eine dieser Funktionen aktiv sein.
- Eines dieser Signale wirkt dominierend. Das ist typischerweise das Rücksetzen (Clear).

Die Unabhängigkeit vom Takt ist manchmal ein Vorteil. So kann das Setzen oder Rücksetzen beim Einschalten schon dann wirken, wenn noch gar kein Taktsignal anliegt.

#### *Synchrones Setzen und Rücksetzen*

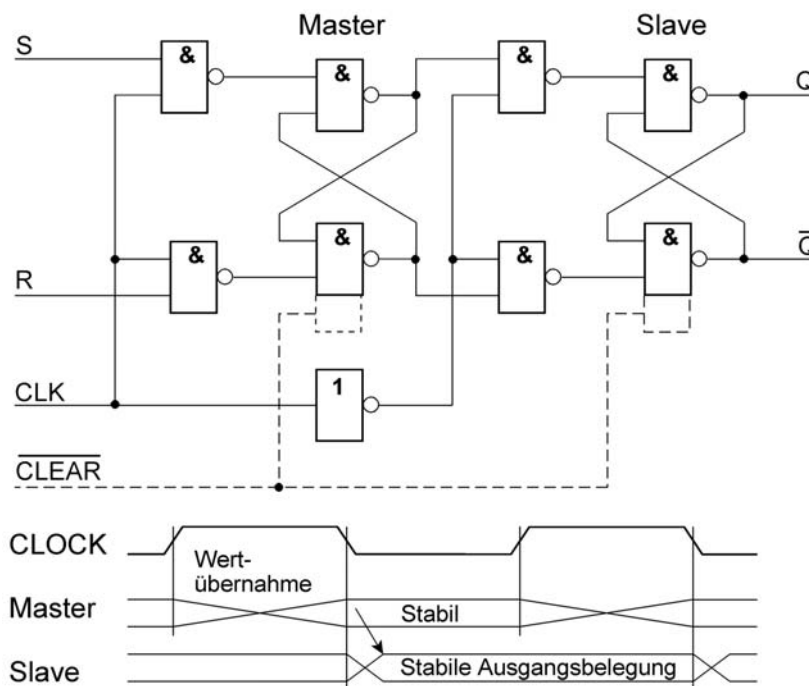
Typische Bezeichnungen: Setzen = Set, Rücksetzen = Reset. Diese Signale werden mit der gleichen Taktflanke wirksam wie die anderen Steuersignale, wirken aber mit Vorrang. Solche Funktionen können durch zusätzliche Beschaltung der Daten- oder Steuersignaleingänge verwirklicht werden. Sind beide Wirkungen vorgesehen, ist der Vorrang festzulegen (Setz- oder Rücksetzdominanz).

### 4.1.5 Der innere Aufbau der Flipflops

Flipflops sind Funktionselemente, deren Innenschaltung als gegeben hinzunehmen ist. Die meisten Flipflops sind durch ihre Verhaltensbeschreibung (z. B. Funktionstabelle) hinreichend genau dokumentiert. Manchmal ist es aber von Vorteil, die Prinzipien des inneren Aufbaus zu kennen. Ein Problem beim Entwerfen von Flipflopschaltungen besteht darin, die Taktflankensteuerung mit logischen Schaltungsmitteln nachzubilden, da naheliegende Lösungen mit Kondensatoren oder Verzögerungsgliedern für die Schaltungsintegration ungeeignet sind.

### Master-Slave-Flipflops

Die Flankensteuerung ergibt sich durch Hintereinanderschalten zweier taktgesteuerter Latches, wobei das zweite mit dem invertierten Takt beschaltet ist. Im Gegensatz zum Latch dürfen sich Änderungen an den Eingängen nicht unmittelbar an den Ausgängen bemerkbar machen; die Ausgangsbelegung soll sich allein mit der übernehmenden Taktflanke ändern. Herkömmliche Master-Slave-Flipflops schalten ihre Ausgänge mit der High-Low-Flanke des Taktes. Ist das Taktsignal High, wird die Belegung der Eingänge in das Master-Latch übernommen. Das Slave-Latch hält weiterhin die zuvor gespeicherte Belegung. Wird nun das Taktsignal Low, so wird die weitere Übernahme in den Master verhindert. Zugleich wird das Slave-Latch aktiviert und somit transparent. Infolgedessen erscheint die Belegung des Masters an den Ausgängen des Flipflops. Wird das Taktsignal wieder High, wird das Slave-Latch gesperrt, und das Master-Flipflop kann die nun aktuelle Eingangsbelegung übernehmen.



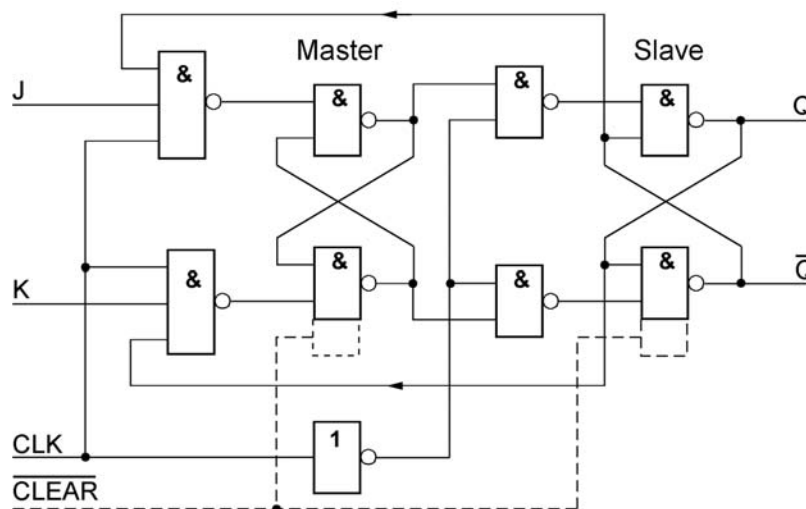
**Abb. 4.38** RS-Master-Slave-Flipflop.

Latch	Takt = High	Takt = Low
Master	Aktiv; Ausgänge folgen den Eingängen.	Gesperrt; die zur Taktflanke (High-Low) herrschende Belegung wird gehalten.
Slave	Gesperrt; die vorherige Belegung des Masters wird gehalten.	Aktiv. Damit wird die Belegung des Masters zu den Ausgängen des Flipflops durchgereicht. Die Ausgangsbelegung ändert sich aber nicht, weil der Master gesperrt ist.

**Tabelle 4.5** Zur Funktionsweise des Master-Slave-Flipflops.

### Das JK-Master-Slave-Flipflop

Das "klassische" JK-Flipflop ist ein aus dem RS-Typ abgewandeltes Master-Slave-Flipflop, wobei die über Kreuz auf die Eingänge zurückgeführten Ausgänge das gewünschte Verhalten bei  $J = K = 1$  gewährleisten. Setzen des Masters: mit  $J = 1$  und  $Q = 0$ . Rücksetzen des Masters: mit  $K = 1$  und  $Q = 1$ . Bei  $J = 1$ ,  $K = 0$  und  $Q = 1$  wird der Master nicht gesetzt, bei  $J = 0$ ,  $K = 1$  und  $Q = 0$  nicht rückgesetzt (das Setzen oder Rücksetzen ist nicht erforderlich, da bei diesen Eingangsbelegungen der Folgezustand dem aktuellen Zustand entspricht). Sind hingegen  $J$  und  $K$  beide  $= 1$ , wird die invertierte Ausgangsbelegung in den Master eingetragen (Setzen, wenn  $Q = 0$ ; Rücksetzen, wenn  $Q = 1$ ).



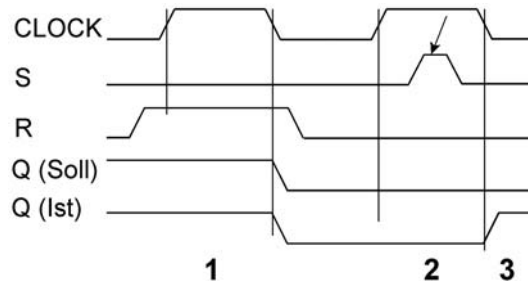
**Abb. 4.39** JK-Master-Slave-Flipflop.

### Master-Slave-Flipflops mit RS-Latches sind störempfindlich

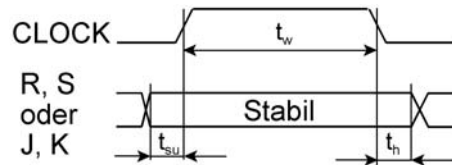
Liegt das Taktsignal auf High-Pegel, so wird in Flipflops ähnlich Abb. 4.38 und 4.39 jeder High-Impuls auf S, R oder J, K im Master gefangen. Hierdurch kann der Speicherzustand des Flipflops verändert werden. In den Datenblättern ist deshalb die Vorhaltezeit in Bezug auf die Takt-Vorderflanke (Low-High) spezifiziert, während die Ausgänge mit der Rückflanke (High-Low) schalten. Mit anderen Worten: solange das Taktsignal High ist, darf sich an den Steuereingängen des Flipflops nichts ändern. Die gesamte Impulsdauer  $t_w$  des Taktsignals geht somit für die Schaltzeit in der Kombinatorik verloren.

### Zweiflankensteuerung

Das Störproblem lässt sich vermeiden, wenn man die Eingangsbelegung mit der Vorderflanke (Low-High) des Taktes in den Master übernimmt, so dass die weiteren Signalverläufe an den Eingängen den Zustand des Flipflops nicht mehr beeinflussen können. Die Datenbelegung wird mit der Low-High-Flanke des Taktsignals übernommen und mit der High-Low-Flanke ausgegeben (Zweiflankensteuerung). Die Vorhalte- und Haltezeitangaben beziehen sich auf die Vorderflanke des Taktes. Die zugehörige Ausgangsbelegung erscheint jedoch nach wie vor mit der Rückflanke. Dieses Schema erlaubt sehr große Takttoleranzen, die gesamte Impulsdauer  $t_w$  des Taktsignals geht aber für die Schaltzeit in der Kombinatorik verloren.



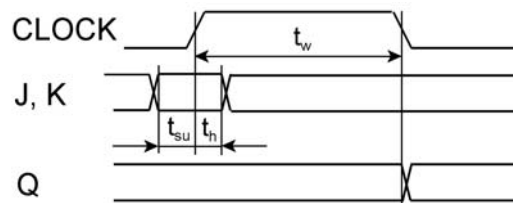
**Abb. 4.40** Fehlschaltung durch Übernahme von Störimpulsen in den Master (Ablaufbeispiel). 1 - mit  $R = 1$  wird das Flipflop zurückgesetzt. 2 - in diesem Taktzyklus sind  $S = R = 0$ , so dass sich der Zustand des Flipflops eigentlich nicht ändern sollte. 3 - der Störimpuls auf S (Pfeil) wird aber im Master gefangen. Infolgedessen ändert sich der Zustand des Flipflops (3).



**Abb. 4.41** Definition der Vorhaltezeit  $t_{su}$  und der Haltezeit  $t_h$  beim herkömmlichen Master-Slave-Flipflop.

#### Datenausblendung (Data Lockout)

Das ist eine technische Lösung der Zweiflankensteuerung. Der Master wird nur bis kurz nach der Low-High-Flanke des Taktsignals offengehalten. Hierzu wird das Master-Latch mit kurzen Taktimpulsen angesteuert, die über eine Verzögerungsstufe gebildet werden.

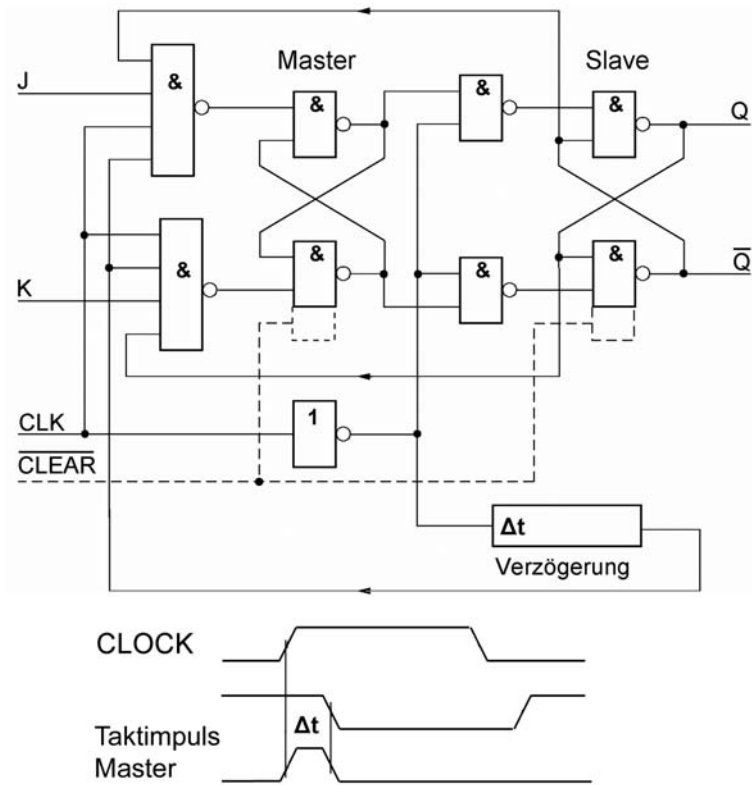


**Abb. 4.42** Zweiflankensteuerung. Definition der Vorhaltezeit  $t_{su}$  und der Haltezeit  $t_h$  beim Master-Slave-Flipflop mit Datenausblendung (Data Lockout).

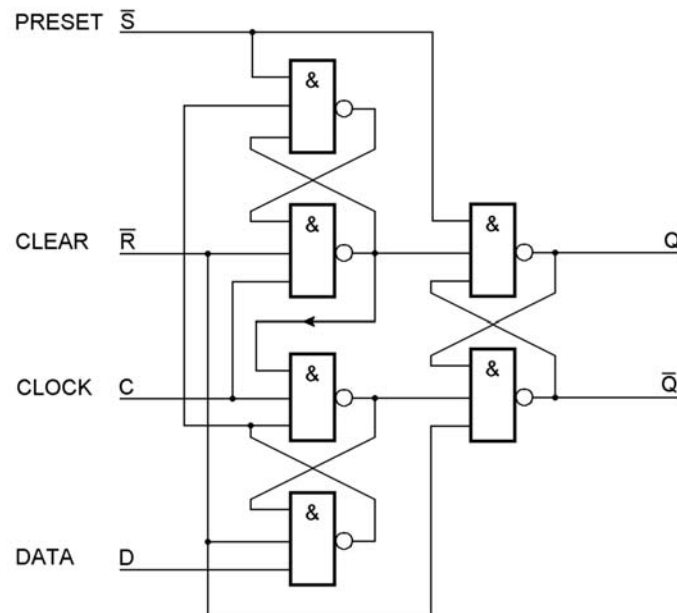
#### Flankengesteuerte Flipflops

Moderne Flipflops sind nahezu ausnahmslos flankengesteuert (Edge Triggered). Es gibt also wirklich nur eine schaltende Signalfanke (Einflankensteuerung); bei D-Typen Low-High (Positive Edge), bei manchen JK-Typen High-Low (Negative Edge). Die Flankensteuerung läßt sich mit Verknüpfungen von Latches verwirklichen (Abb. 4.43).





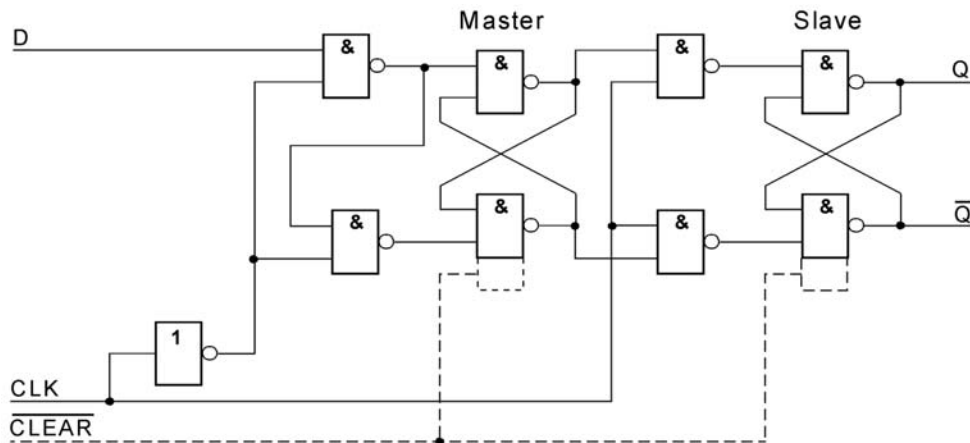
**Abb. 4.43** Master-Slave-Flipflop mit Zweiflankensteuerung durch Datenausblendung (Data Lockout).



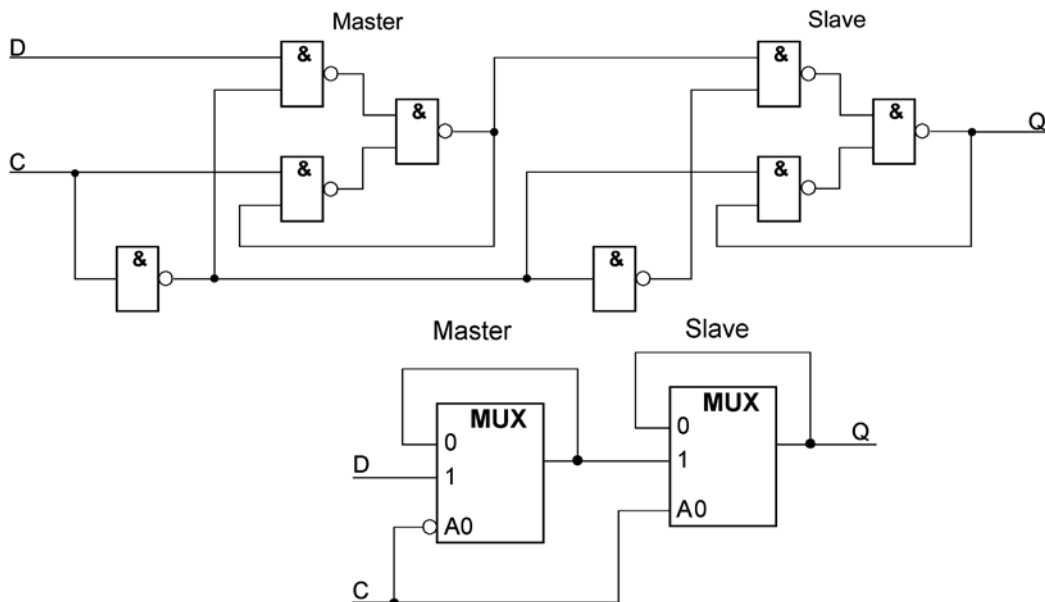
**Abb. 4.44** Flankengesteuertes D-Flipflop (Typ 7474; nach [2.12]). Funktionserklärungen sind beispielsweise in [3] und [11] zu finden.

### Flankengesteuerte Flipflops in Master-Slave-Schaltung

Die Master-Slave-Anordnung wird als D-Flipflop aufgebaut; alle anderen Übergangsfunktionen werden durch Zusatzbeschaltung realisiert. Der Grundgedanke: liegt der Takt am Master auf High, so folgt das Master-Latch zwangsläufig dem D-Eingang. Somit können Störimpulse auch nicht im Master gefangen werden (ein solcher Impuls würde das Master-Latch kurzzeitig setzen und dann wieder zurücksetzen). Damit die Übernahme – wie allgemein üblich – mit der Low-High-Flanke stattfindet, muss das Master-Latch mit einem invertierten Taktsignal angesteuert werden (Abb. 4.45, 4.46).



**Abb. 4.45** Flankengesteuertes D-Flipflop. Ein D-Latch (vgl. Abb. 4.11b) und ein RS-Latch in Master-Slave-Schaltung.



**Abb. 4.46** Flankengesteuertes D-Flipflop als Master-Slave-Anordnung von zwei D-Latches.

### Latches mit Flipflops nachbauen

Ein Flipflop mit asynchronen Setz- und Rücksetzeingängen kann als RS-Latch betrieben werden (Abb. 4.47).

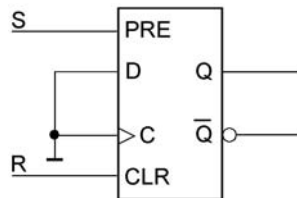


Abb. 4.47 Flipflop als RS-Latch.

## 4.1.6 Metastabilität

### Das Synchronisationsproblem

In einer synchron arbeitenden Schaltung liegen die Datensignale während des Setup-Hold-Intervalls stabil an den Eingängen der Flipflops an. Andernfalls handelt es sich um einen Fehler, der abzustellen ist. An den Schnittstellen zur Außenwelt ist aber stets damit zu rechnen, dass Signale während des Setup-Hold-Intervalls umschalten – die von außen kommenden Signale können schließlich nichts vom Taktraster der Schaltung wissen. Werden solche asynchron schaltende Signale in eine synchrone Schaltung eingeleitet, ist eine Anpassung an das Taktraster erforderlich (Synchronisation, Eintaktierung). Die Einfachlösung besteht darin, das asynchron schaltende Signal an den Dateneingang eines Flipflops anzulegen (Abb. 4.48). Einfache Latches sind für solche Einsatzfälle grundsätzlich ungeeignet.

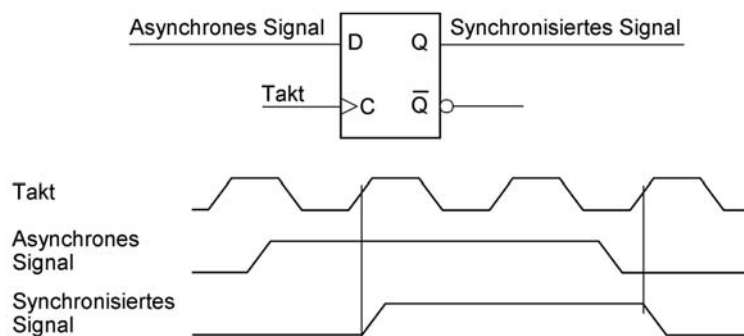
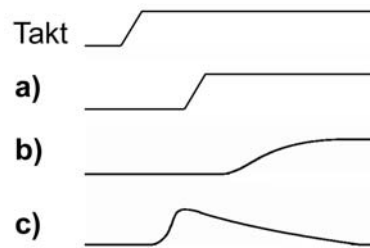


Abb. 4.48 Eine einfache Synchronisationsschaltung.

### Metastabile Zustände

Die Synchronisation ist im Grunde ein Abtastvorgang, wobei das Signal beispielsweise mit der Low-High-Flanke des Synchronisationstaktes abgetastet wird. Im Folgenden geht es nicht um die Signalerfassung und -bewertung sondern lediglich darum, was geschieht, wenn sich die Eingangsbelegung eines Flipflops im Setup-Hold-Intervall einer schaltenden Taktflanke

ändert. Das Verhalten des Flipflops hängt dann ganz vom Zufall ab. Manchmal wird nichts Besonderes geschehen; das Flipflop wird entweder seinen Zustand ändern oder den bisherigen beibehalten. Es gibt aber innerhalb des Setup-Hold-Intervalls ein kritisches Zeitfenster. Dessen Breite hängt von der Schaltungstechnologie und dem Aufbau des Flipflops ab<sup>3)</sup>. Fällt die Änderung der Eingangsbelegung in dieses Zeitfenster, so kann das Flipflop in einen Zwischenzustand gelangen, der als metastabiler Zustand bezeichnet wird. In einem solchen Zustand gibt das Flipflop Ausgangssignale ab, die nicht eindeutig einem der beiden Logikpegel entsprechen (Abb. 4.49).



**Abb. 4.49** Ausgangssignale eines Synchronisationsflipflops (nach [4.2]). a) korrektes Ausgangssignal; b), c) typische Signalverläufe in metastabilen Zuständen.

### Fehlerwirkungen

Ein metastabiler Zustand klingt mit der Zeit ab; das Flipflop wird sich allmählich auf einen der beiden Logikpegel einstellen. In der Zwischenzeit können aber Fehlfunktionen in der Schaltung auftreten. Typische Fehlermechanismen:

- Nachgeordnete binäre Eingänge können sich undefiniert verhalten, wenn der Signalpegel im verbotenen Bereich liegt.
- Sind dem Flipflop mehrere Eingänge nachgeordnet, so können diese unterschiedlich reagieren (ein Eingang sieht ein Low, ein anderer ein High, ein weiterer einen Wert im verbotenen Bereich usw.).
- Werden beide Ausgänge des Flipflops genutzt, so ist die Belegung des einen nicht immer die Negation der Belegung des anderen.

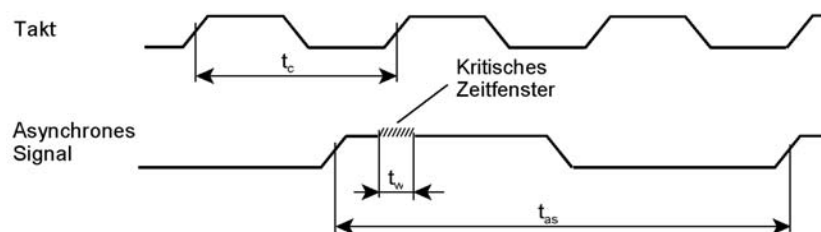
Eine einfache Schwellwertschaltung – z. B. ein Schmitt-Trigger – hinter dem Flipflop Ausgang nützt nichts, da der Signalverlauf unbestimmt ist. Dann ist aber auch der Zeitpunkt unbestimmt, zu dem die Schwellwertschaltung umschaltet. Diese Schaltflanke kann daher wiederum in das Setup-Hold-Intervall nachgeschalteter Flipflops fallen. Entwurfsfehler in Hinsicht auf die Metastabilität äußern sich typischerweise in zeitweiligen Funktionsfehlern, die oft mit einer gewissen Regelmäßigkeit auftreten (verändert man – falls möglich – die Signalfrequenzen und ändert sich dabei auch der Fehlerabstand, so ist dies ein recht sichere Bestätigung des Fehlerverdachts).

---

3: Typische Werte: 1 ... ca. 150 ps.

Wie häufig können metastabile Zustände vorkommen?

Die Wahrscheinlichkeit, dass ein metastabiler Zustand eintritt, hängt von der Dauer des kritischen Zeitfensters, der Schalthäufigkeit des asynchronen Signals und der Taktfrequenz ab. Mit Ereignissen, die zu metastabilen Zuständen führen können, ist typischerweise in Abständen von Sekunden...Minuten zu rechnen. Ein metastabiler Zustand kann dann eintreten, wenn eine Flanke des asynchronen Signals in ein kritisches Zeitfenster der Breite  $t_w$  fällt (Abb. 4.50). Die folgenden Betrachtungen beziehen sich auf ein asynchrones Signal, das mit einer konstanten Impulsfolgefrequenz schaltet. Es sind somit gegeben: (1) die Taktfrequenz  $f_c$  (bzw. die Periodendauer  $t_c$ ) und (2) die Frequenz des asynchronen Signals  $f_{as}$  (bzw. die Periodendauer  $t_{as}$ ).



**Abb. 4.50** Zur Wahrscheinlichkeit des Auftretens eines metastabilen Zustandes. Die Flanke des asynchronen Signals muss mit dem kritischen Zeitfenster zusammentreffen, das während der Low-High-Flanke des Taktes auftritt.

Offensichtlich kann jeder Periode  $t_{as}$  des asynchronen Signals in  $t_{as} : t_w$  aufeinander folgende kritische Zeitfenster aufgeteilt werden. Die Wahrscheinlichkeit, im Intervall  $t_{as}$  auf ein bestimmtes Zeitfenster der Dauer  $t_w$  zu treffen, ergibt sich zu:

$$1 / \frac{t_{as}}{t_w} = \frac{t_w}{t_{as}}$$

Metastabile Zustände ergeben sich, wenn eine schaltende Taktflanke mit dem kritischen Zeitfenster in der Umgebung einer Flanke des asynchronen Signals zusammentrifft. Die Gelegenheit dafür wiederholt sich zyklisch mit der Taktperiode  $t_c$ . Die Trefferrate ergibt sich somit zu:

$$\frac{t_w}{t_{as}} / t_c = \frac{t_w}{t_{as} \cdot t_c}$$

Der Kehrwert der Trefferrate ist der Mittelwert des Zeitabstands zwischen zwei metastabilen Zuständen (Fehlfunktionsabstand  $MTBF_{MS}$ ):

$$\text{MTBF}_{\text{MS}} = \frac{t_{\text{as}} \cdot t_{\text{c}}}{t_{\text{w}}} = \frac{1}{f_{\text{as}} \cdot f_{\text{c}} \cdot t_{\text{w}}} \quad (4.15)$$

Beispiel:  $f_{\text{CLK}} = 2 \text{ MHz}$ ,  $f_{\text{IN}} = 10 \text{ kHz}$ ,  $t_{\text{w}} = 50 \text{ ps}$ . Damit ergibt sich:

$$\text{MTBF}_{\text{MS}} = \frac{1}{2 \text{ MHz} \cdot 10 \text{ kHz} \cdot 50 \text{ ps}} = 1 \text{ s}$$

Im Mittel ist also etwa alle Sekunde mit dem Eintreten eines metastabilen Zustands zu rechnen.

### Fehlervermeidung

Metastabile Zustände sind grundsätzlich unvermeidbar. Was man lediglich tun kann: von der übernehmenden Taktflanke an eine gewisse Zeit zu warten (Beruhigungszeit) – in der Hoffnung, dass dann ein metastabiler Zustand, der sich womöglich eingestellt hat, wieder abgeklungen sein wird.

#### *MTBF – der Fehlfunktionsabstand*

Es verbleibt stets ein gewisses Restrisiko – das heißt eine von Null verschiedene Wahrscheinlichkeit – dafür, dass sich das Flipflop auch nach Ablauf der Beruhigungszeit noch in einem metastabilen Zustand befindet. Der mittlere Zeitabstand zwischen zwei derartigen Zuständen (Fehlfunktionsabstand) ist die MTBF (Mean Time Between Failures). Der Fehlfunktionsabstand hängt von der Beruhigungszeit (Settling Time)  $\Delta t$  und von der Auslegung des Flipflops ab. Der grundsätzliche Zusammenhang:

$$\text{MTBF} = \frac{e^{S \cdot \Delta t}}{f_{\text{as}} \cdot f_{\text{c}} \cdot T} \quad (4.16)$$

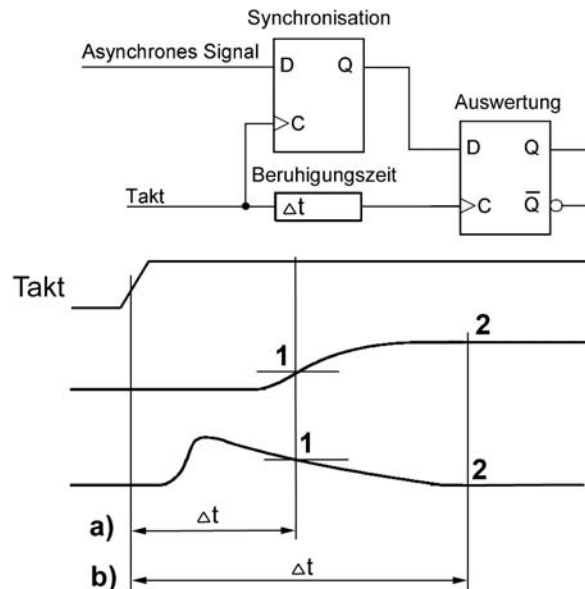
S und T sind Kennwerte, die von der Halbleitertechnologie und vom Schaltungsaufbau der Flipflops abhängen<sup>4</sup>. S wird in 1/ns oder 1/s angegeben, T in Sekunden. S ist ein Kennwert für die Geschwindigkeit, mit der das Flipflop aus dem metastabilen in einen normalen Zustand zurückfällt. Der Wert ergibt sich aus dem Verstärkungs-Bandbreiten-Produkt im Rückkopplungsweg des Master-Latches im Flipflop. T ist ein Zeitkennwert, der mit der Breite des kritischen Zeitfensters zusammenhängt.

Der Exponent in (4.16) hat den größten Einfluss auf die Dauer des Fehlfunktionsabstands. Eine geringe Verlängerung der Beruhigungszeit  $\Delta t$  oder eine kleine Erhöhung von S bewirken eine beträchtliche Verlängerung der MTBF. Die Beruhigungszeit  $\Delta t$  ist Sache des

---

4: Sie sind im Datenmaterial oder in einschlägigen Anwendungshinweisen (Application Notes) zu finden. Praxistipp: Mit den Stichworten *metastability* oder *metastable behavior* auf der Internetseite des jeweiligen Herstellers suchen.

Schaltungsentwurfs, der Kennwert  $S$  ist Sache des Schaltkreisherstellers. Moderne Flipflopschaltungen werden u. a. auch auf hohe  $S$ -Werte hin ausgelegt, um gute Metastabilitätseigenschaften zu erreichen.



**Abb. 4.51** Metastabile Zustände klingen von selbst ab, wenn man nur lange genug wartet. a) die Beruhigungszeit  $\Delta t$  ist offensichtlich zu kurz. b) die Beruhigungszeit ist offensichtlich lang genug; das auswertende Flipflop findet jeweils einen zulässigen Logikpegel an seinem Eingang vor. 1 - unzulässige, 2 - zulässige Logikpegel.

Es liegt nahe, einen Fehlfunktionsabstand anzustreben, der deutlich über den MTBF-Werten der Fehlfunktionen aus anderen Ursachen und der Ausfälle liegt (MTBF > voraussichtliche Lebensdauer)<sup>5</sup>. (4.16) kann nach der Beruhigungszeit  $\Delta t$  umgestellt werden:

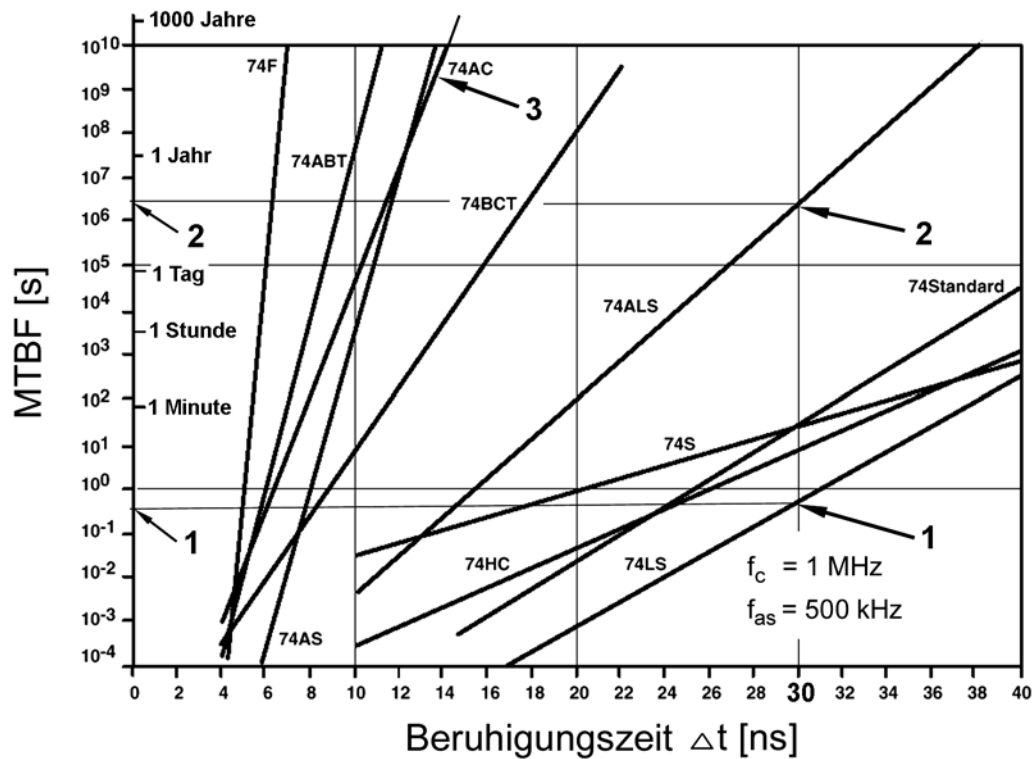
$$\Delta t = \frac{1}{S} \ln (\text{MTBF} \cdot f_{\text{as}} \cdot f_{\text{c}} \cdot T) \quad (4.17)$$

Ist die so errechnete Beruhigungszeit zu lang, wäre nach besser geeigneten Bauelementen zu suchen ( $S$  größer,  $T$  kleiner, wobei es vor allem auf  $S$  ankommt). Solche Angaben werden typischerweise in Tabellen oder in Diagrammen zusammengefasst (Tabelle 4.6, Abb. 4.52).

5: Beispiel: eine MTBF von  $10^4$  Jahren. In manchen Anwendungsbereichen (Telekommunikation, Militärwesen usw.) gehören solche Anforderungen zu den Entwurfsvorgaben. Deren Einhaltung ist rechnerisch nachzuweisen.

Baureihe	S [1/ns]	T [s]	$f_c, f_{as}$
Standard-TTL	0,74	$2,9 \cdot 10^{-4}$	$f_{as} = 0,5 f_c$
LS	0,74	$4,8 \cdot 10^{-3}$	
S	0,36	$1,3 \cdot 10^{-9}$	
ALS	1,0	$8,7 \cdot 10^{-6}$	
AS	4,0	$1,4 \cdot 10^{-3}$	
F	9,2	$1,9 \cdot 10^{-8}$	
BCT	1,51	$1,14 \cdot 10^{-6}$	
ABT	3,61	$33 \cdot 10^{-3}$	
HC	0,55	$1,46 \cdot 10^{-6}$	
AC	2,8	$1,1 \cdot 10^{-4}$	
XC4005E-3 IOB	16,1	$10^{-10}$	
XC4005E-3 CLB	19,4	$10^{-10}$	
MAX 7000 CPLD	5,023	$2,98 \cdot 10^{-17}$	
FLEX 10K	12,68	$1,01 \cdot 10^{-13}$	

**Tabelle 4.6** Metastabilitätskennwerte (Beispiele).



**Abb. 4.52** Das metastabile Verhalten verbreiteter herkömmlicher Logikfamilien. 1...3 - Ablesebeispiele. Erläuterung im Text.



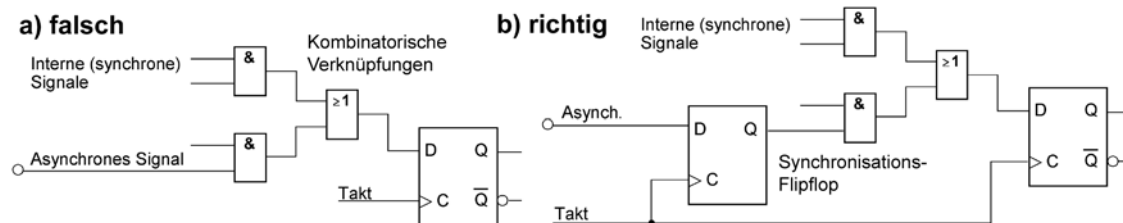
Die Ablesebeispiele beziehen sich auf eine Beruhigungszeit von 30 ns:

- 1) Schaltkreis der Baureihe 74LS. Es ergibt sich eine MTBF zwischen 0,1 und 1 s. Das ist offensichtlich viel zu wenig.
- 2) Schaltkreis der Baureihe 74ALS.  $MTBF = 10^7$  s. Ein Jahr hat  $365 \cdot 24 \cdot 3600 = 31\,536\,000$  s. Also rund 4 Monate ( $\frac{1}{3}$  Jahr). Auch damit kann man sich nicht auf dem Markt blicken lassen (an Qualitätsvorschriften und Garantiefrieten denken ...).
- 3) Ein Schaltkreis der Baureihe 74AC. Bei 30 ns zeigt Abb. 4.64 gar nichts mehr an. Rechnerisch ergeben sich ca.  $18 \cdot 10^{20}$  Jahre. Wenn eine MTBF von 1000 Jahren ausreicht, kann man die Beruhigungszeit auf etwa 15 ns verkürzen.

### Schaltungspraxis

Die Beruhigungszeit wird zumeist nicht durch eine Verzögerungsstufe, sondern durch die Taktperiode dargestellt. Wichtig ist:

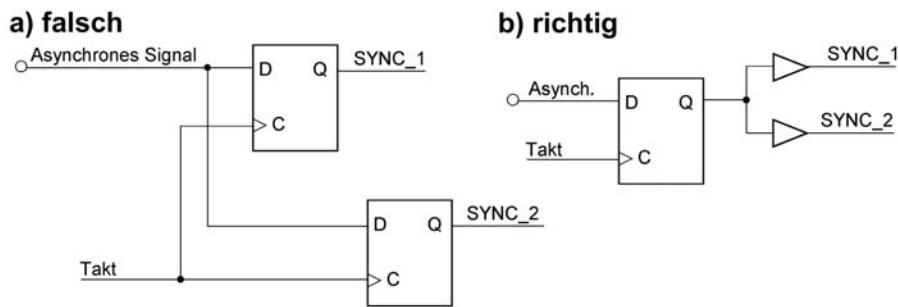
- Die asynchronen Signale nicht direkt in Verknüpfungen einbeziehen, sondern zunächst mit einem eigenen Flipflop synchronisieren. Dann liegt eine Taktperiode zwischen Abtastung und Auswertung. Diese Zeit genügt oftmals vollkommen, um die metastabilen Zustände des Synchronisationsflipflops abklingen zu lassen.
- Ein asynchrones Signal nur mit einem einzigen Synchronisationsflipflop abtasten.



**Abb. 4.53** Asynchrone Signale richtig synchronisieren, nicht direkt in die Verknüpfungen einfließen lassen (denn dann gibt es keine Beruhigungszeit). In den meisten Fällen genügt die Taktperiode zwischen der Übernahme ins Synchronisationsflipflop und der Übernahme in die Flipflops der Verarbeitungsschaltungen, um metastabile Zustände in den Flipflops der Verarbeitungsschaltungen praktisch auszuschließen.

Je “schneller” die Baureihe, desto besser das Metastabilitätsverhalten. In modernen Schaltkreisen genügen einige wenige ns Beruhigungszeit, um eine bei Weitem ausreichende MTBF zu gewährleisten<sup>6)</sup>.

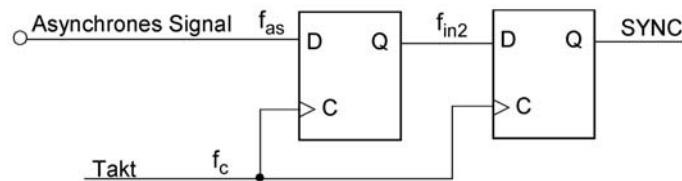
6: Ein Hersteller nennt Millionen Jahre bei etwa 5 ns. Bei Taktfrequenzen unter 100 MHz sind praktisch keine Probleme hinsichtlich der Metastabilität zu erwarten.



**Abb. 4.54** Jedes asynchrone Signal nur mit einem einzigen Flipflop abtasten. a) so könnte es sein, dass SYNC\_1 anders belegt ist als SYNC\_2. b) veranschaulicht eine Lösung, wenn das synchronisierte Signal an mehreren Stellen benötigt wird (höherer Fanout durch nachgesetzte Treiberstufen).

### Zweistufige Synchronisation

Durch Hintereinanderschalten von zwei Synchronisationsflipflops kann die MTBF deutlich verlängert werden.



**Abb. 4.55** Zweistufige Synchronisation.

Eine Fehlfunktion liegt vor, wenn das zweite Flipflop in einen metastabilen Zustand gerät. Das kann nur dann vorkommen, wenn das erste Flipflop einen metastabilen Zustand aufweist. Der Mittelwert des Zeitabstands zwischen diesen Zuständen ist die MTBF des ersten Flipflops. Für gleiche Flipflops und gleiche Beruhigungszeiten  $\Delta t$  gibt sich

$$MTBF_2 = \frac{e^{2 \cdot \Delta t \cdot S}}{f_c \cdot f_{as} \cdot T^2}$$

Durch zweistufige Synchronisation kann man auch mit Flipflops, die vergleichsweise niedrige MTBF-Werte haben, hochzuverlässige Synchronisationsschaltungen bauen. Das kostet allerdings einen zusätzlichen Taktzyklus zwischen dem Abtasten des asynchronen und dem Auswerten des synchronisierten Signals. In manchen programmierbaren Schaltkreisen bietet sich die Möglichkeit, die Synchronisationsflipflops mit der doppelten Taktfrequenz zu betreiben. Obwohl sich infolge der kürzeren Taktzyklen die MTBF des einzelnen Flipflops verringert, hat die zweistufige Anordnung einen deutlich längeren Fehlfunktionsabstand.

## 4.2 Register

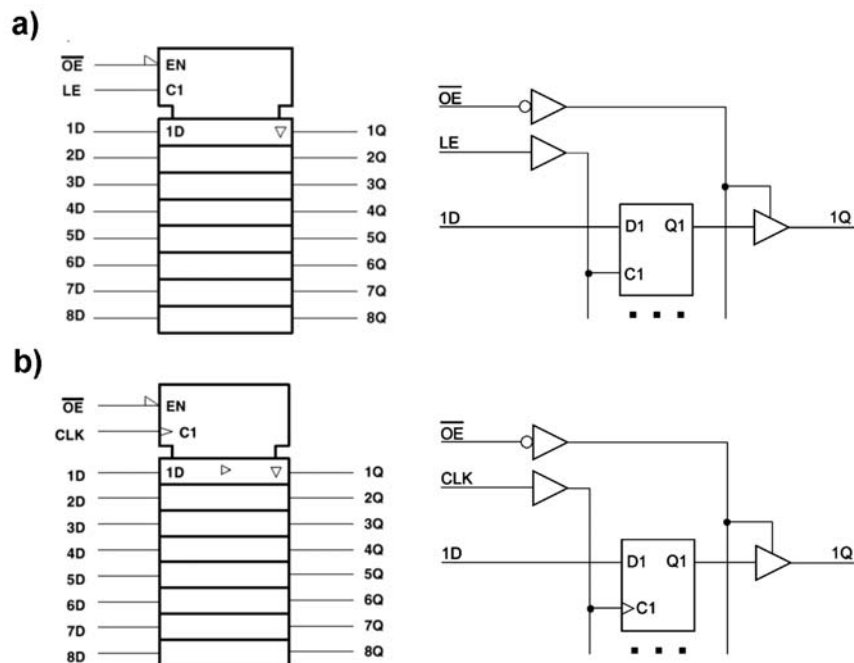
Register sind Aneinanderreihungen von Latches oder Flipflops mit gemeinsamem Takt. Durch Zusatzbeschaltung wird aus der einfachen Aneinanderreihung ein Register mit speziellen Funktionen. Einige Funktionen, die besonders häufig benötigt werden, haben den Charakter von Grundsaltungen. Das betrifft vor allem die Schieberegister und Zähler (Kapitel 6).

### 4.2.1 Standardschaltkreise

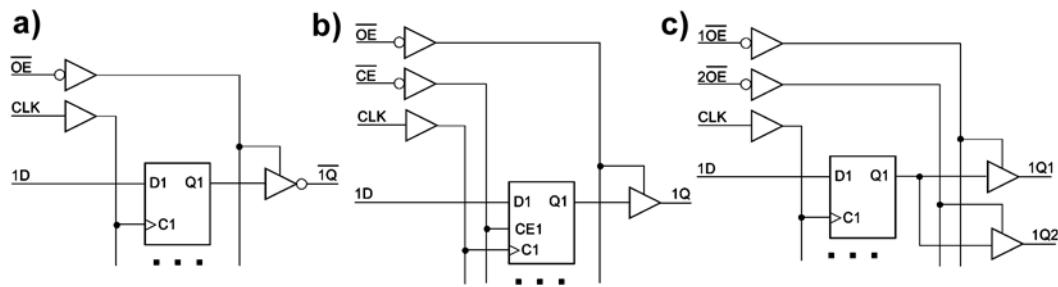
Die Schaltkreishersteller haben im Laufe der Zeit viele Registertypen entwickelt. Einige sind zu echten Industriestandards geworden (daran erkennbar, dass sie in nahezu allen Baureihen angeboten werden). Die meisten Registerschaltkreise haben Tri-State-Ausgänge. Braucht man zweiwertige Ausgänge, ist der Erlaubniseingang (OE) mit Masse zu verbinden. In vielen Baureihen werden Latch- und D-Flipfloptypen in ansonsten gleicher Auslegung (Gehäuse, Anschlussbelegung, Technologie usw.) angeboten. Gängige Breiten (Anzahl der Bitpositionen): 8, 10, 16, 18, 20, 22, 32, 36. Solche Registerschaltkreise sind vor allem als Puffer, als Haltereister, als Buskoppelstufen und zur Pegelwandlung vorgesehen.

#### Besonderheiten

Der typische Registerschaltkreis ist heutzutage das einfache Latch- oder Flipflopregister mit nicht invertierten Tri-State-Ausgängen. Viele der in den Datenbüchern angegebenen Typen mit Sonderfunktionen werden für Neuentwicklungen nicht mehr empfohlen oder gar nicht mehr gefertigt.



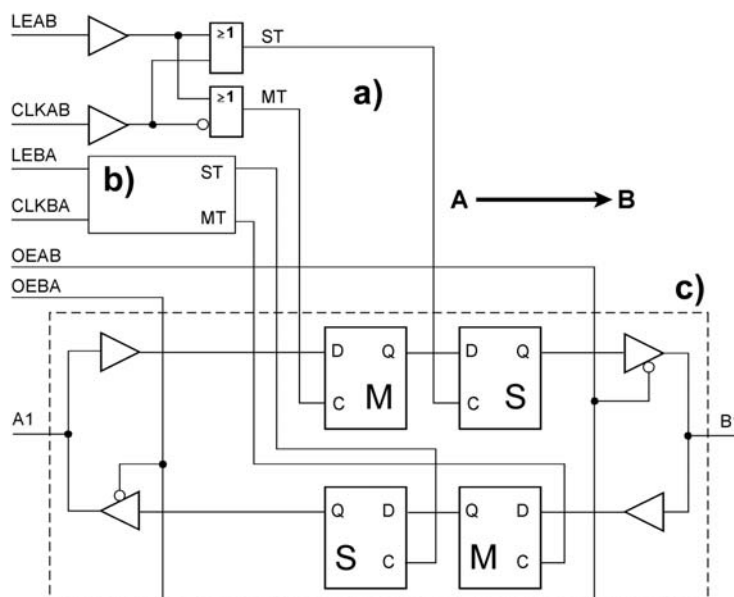
**Abb. 4.56** Typische Registerschaltkreise – eine kleine Auswahl. a) Latchregister; b) Flipflopregister. Links Schaltsymbole nach DIN 40 900, rechts Innenschaltung (nach [2.10]).



**Abb. 4.57** Registerschaltkreise mit Besonderheiten (nach [2.51]). a) invertierte Ausgänge; b) Taktsteuereingang; c) zwei Tri-State-Ausgänge.

### Universelle Buskoppelschaltkreise

Der Grundgedanke: ein Schaltkreis, vielfältige Einsatzmöglichkeiten (Treiber ohne Speicherfunktion, Latchregister, Flipflopregister).



**Abb. 4.58** Ein universeller Buskoppelschaltkreis (Universal Bus Transceiver; nach [2.10]). M - Master; S - Slave; MT - Master-Takt; ST - Slave-Takt. a) zentrale Steuerung für Richtung A-B; b) zentrale Steuerung für Richtung B-A (wie a) aufgebaut); c) eine Datenbitposition (von beispielsweise 18).

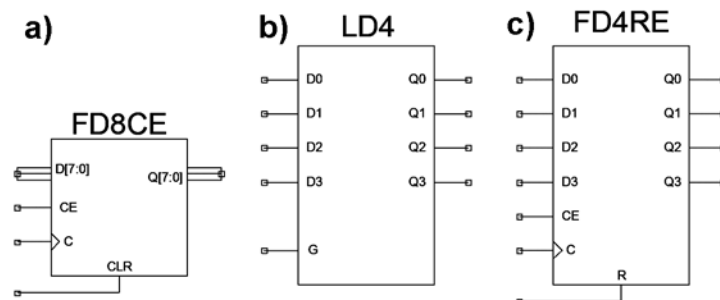
Die Universalität kommt durch trickreiches Ausnutzen der Eigenschaften von D-Latches zustande. Das soll anhand der Übertragungsrichtung von A nach B erläutert werden. Das Steuersignal LEAB entscheidet über die Funktionsweise:

- LEAB = 0: ST und MT hängen allein vom Takteingang CLKAB ab. Das Master-Latch wird aktiviert bei CLKAB = 0, das Slave-Latch bei CLKAB = 1. Die Anordnung verhält sich also wie das aus D-Latches gebildete D-Flipflop von Abb. 4.46.

- LEAB = 1: Über die ODER-Gatter werden sowohl ST als auch MT aktiv. Beide Latches wirken somit als Durchreiche (Ausgang folgt Eingang).
- Betrieb als Latchregister: durch Schalten von LEAB bei Festbelegung von CLKAB:
  - Wenn CLKAB = 0 ist, wirkt S als Latch, und M ist Durchreiche. ST schaltet mit LEAB, MT wird von CLKAB fest auf 1 gehalten.
  - Wenn CLKAB = 1 ist, wirkt M als Latch, und S ist Durchreiche. MT schaltet mit LEAB, ST wird von CLKAB fest auf 1 gehalten.

## 4.2.2 Registerentwurf

Register in programmierbaren und anwendungsspezifischen Schaltkreisen sind aus dem jeweiligen Funktionselementesortiment auszuwählen oder von Grund auf zu entwerfen. Zumeist genügt es, passende Latches oder Flipflops mit den gemeinsamen Signalen (Takt, Rücksetzen usw.) zu verbinden.



**Abb. 4.59** Register als Funktionselemente. Eine kleine Auswahl (nach [S04]). a) Flipflopregister (acht Bits) mit Erlaubniseingang und asynchronem Löschen; b) Latchregister (vier Bits); c) Flipflopregister mit Erlaubniseingang und synchronem Rücksetzen (vier Bits).

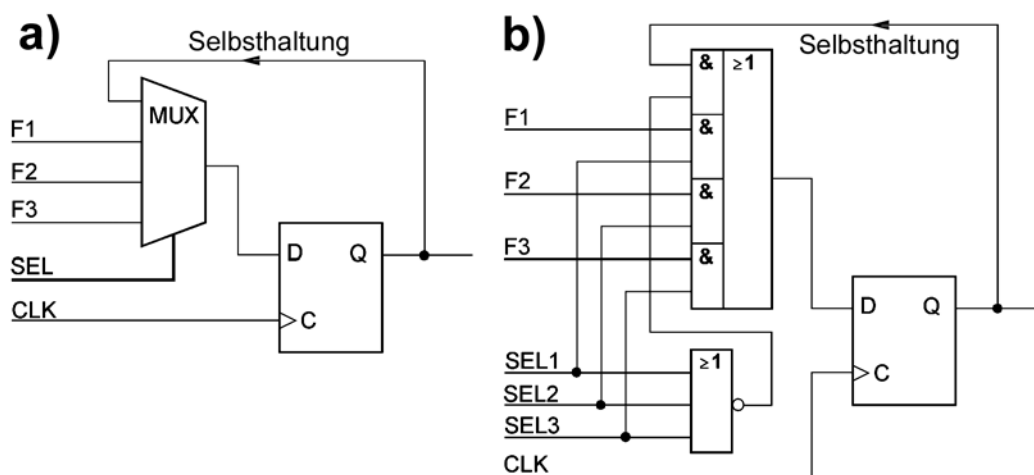
Die Vorzugslösung ist der vollsynchroner Betrieb mit durchlaufendem Takt und Erlaubnissteuerung der Datenübernahme. Das Speicherelement des steuerbaren Datenregisters ist das DE-Flipflop. Zur Taktversorgung sollten die Taktressourcen des Schaltkreises ausgenutzt werden (hierdurch erledigt sich auch das Problem der Taktverteilung). Anwendungsspezifische Taktsignale sollten soweit möglich vermieden werden. Dass vollsynchroner Lösungen gelegentlich mehr Aufwand in der Kombinatorik erfordern, spielt praktisch keine Rolle, da die benötigten Funktionszuordner in den Zellen ohnehin vorhanden sind.

## 4.2.3 Vollsynchroner Mehrfunktionsregister

Vollsynchroner Register können mehr als nur Daten speichern. Abb. 4.60 zeigt den grundsätzlichen Aufbau von Registern, die verschiedene Funktionen ausführen (Daten parallel übernehmen, in verschiedene Richtungen schieben, zählen usw.) Im allgemeinen Fall sind mehrere Funktionen F1, F2, F3 usw. vorgesehen, z. B. F1 = paralleles Laden, F2 =

Linksschieben, F3 = Vorwärtszählen usw. Für  $n$  Funktionen ist jedem Flipflop ein Multiplexer oder Datenselektor mit  $n + 1$  Eingängen vorgeschaltet. Der zusätzliche Eingang dient der Selbsthaltung. An die verbleibenden  $n$  Eingänge sind die jeweiligen funktionellen Verbindungen und Netzwerke angeschlossen (Dateneingänge, Ausgänge benachbarter Flipflops (zum Schieben), Zählnetzwerke usw.). Ist keine Funktion auszuführen, so ist die Selbsthaltung zu aktivieren. Vorrangregeln sind mit entsprechenden Zusatzbeschaltungen<sup>7)</sup> zu implementieren. Manchmal ist es von Vorteil, die kombinatorischen Verknüpfungen mit Schaltnetzen zu erledigen, die aus 2-zu-1-Multiplexern aufgebaut sind. Dann ergibt sich die Vorrangregelung aus der Reihenfolge der Multiplexer. Je näher der Multiplexer dem Flipflopeingang ist, desto höher ist die Priorität des Auswahlsignals.

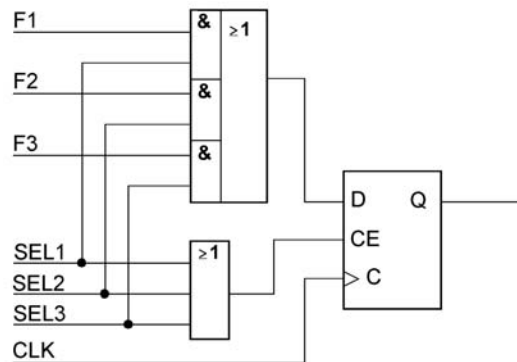
Werden DE-Flipflops eingesetzt, so entfällt die Rückführung (da sie schon im Innern der Flipflops vorgesehen ist). Stattdessen ist der Erlaubniseingang CE dann zu aktivieren, wenn eine Funktion auszuführen ist (z. B. über eine disjunktive Verknüpfung der Funktionsauswahlsignale).



**Abb. 4.60** Vollsynchrone Mehrfunktionsregister. a) mit Multiplexer (binär codierte Funktionsauswahl); b) mit Datenselektor (Funktionsauswahl 1 aus  $n$ ). Die Funktionssignale F1, F2 usw. können Eingänge sein, aber auch Ausgänge anderer Flipflops oder Ausgänge von Verknüpfungsschaltungen, beispielsweise von Zählnetzwerken. Nähere Erläuterung im Text.

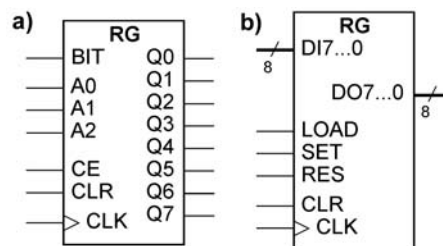
- Funktionsauswahl über Multiplexer. Sind die SELECT-Eingänge mit Nullen belegt, so geschieht nichts, und die Datenbelegung bleibt erhalten (Selbsthaltung). Ansonsten wählt jede SELECT-Belegung eine bestimmte Funktion aus.
- Funktionsauswahl über Datenselektor. Die Auswahleingänge SEL1, SEL2 usw. werden im 1-aus- $n$ -Code angesteuert. Sind alle Auswahleingänge mit Nullen belegt, so wird über das NOR-Gatter die Rückführung aktiviert (Selbsthaltung).

7: Das sind zumeist konjunktive Verknüpfungen, wobei die Signale, die jeweils mit Vorrang wirken sollen, invertiert angeschlossen werden (Inhibition).



**Abb. 4.61** Vollsynchrones Mehrfunktionsregister mit DE-Flipflops. Das Erlaubnissignal CE ist dann zu aktivieren, wenn eine Funktion auszuführen ist.

Die Abb. 4.62 bis 4.64 veranschaulichen das Prinzip des vollsynchronen Mehrfunktionsregisters anhand von zwei Beispielen. Beide Register haben acht Bitpositionen.



**Abb. 4.62** Zwei Beispiele von Mehrfunktionsregistern. a) Einzelbitzugriff; b) selektives Setzen und Löschen.

#### Beispiel 1: Einzelbitzugriff

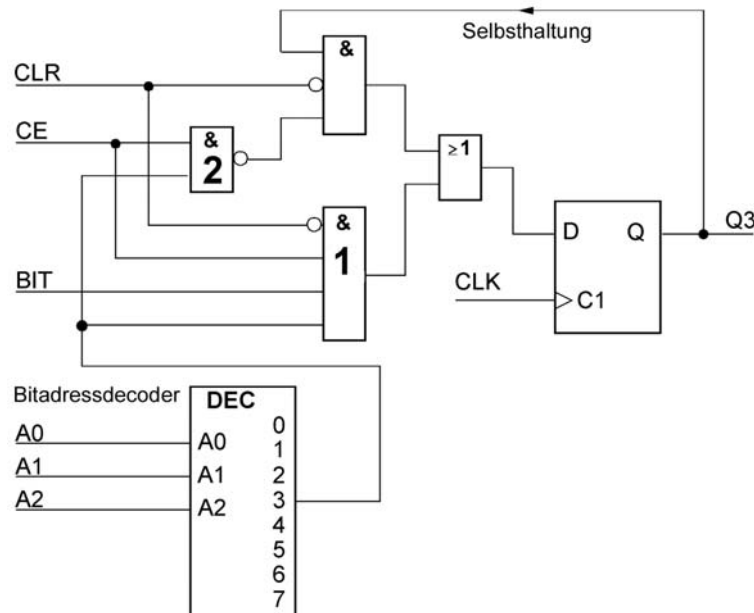
Über die Adresseingänge A2..A0 kann jede der acht Bitpositionen Q7..Q0 ausgewählt werden. Bei aktivem Erlaubniseingang CE wird die Belegung des Dateneingangs BIT in das ausgewählte Flipflop übernommen. Der Löscheinang CLR bewirkt ein synchrones Löschen des Registers. Diese LösCHFunktion wirkt dominierend. Die Selbsthaltebedingung ist wirksam, wenn CLR und CE beide inaktiv sind oder wenn CE aktiv, die betreffende Bitposition aber nicht ausgewählt ist.

#### Beispiel 2: selektives Setzen und Löschen

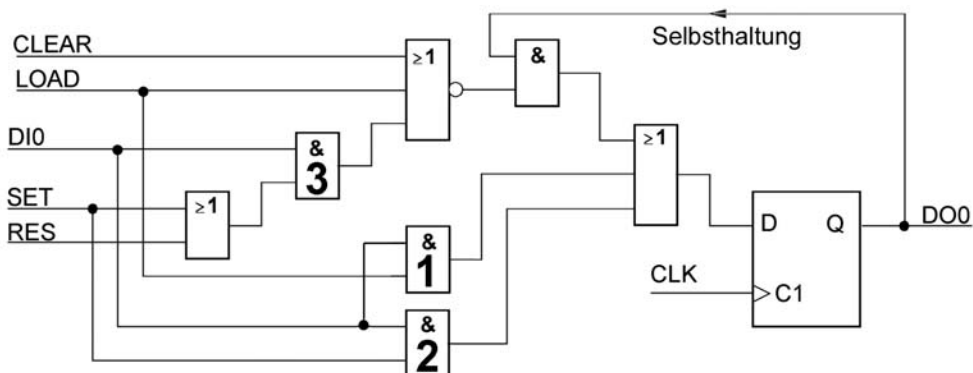
Es gibt vier Funktionen: Gesamtlöschen (CLEAR), Laden (LOAD), selektives Setzen (SET) und selektives Löschen (RES). Beim selektiven Setzen und Löschen werden die Dateneingänge ausgewertet. Ist ein Dateneingang mit Eins belegt, so wird die betreffende Bitposition gesetzt oder gelöscht. Ansonsten bleibt sie unverändert<sup>8)</sup>. In der Schaltung wurde

8: Vgl. die Funktionen der E-A-Ports verschiedener Mikrocontroller.

– der Überschaubarkeit wegen – auf eine Vorrangregelung verzichtet; es darf somit jeweils nur eines der Steuersignale aktiv sein. Ist LOAD aktiv, wird die Belegung des jeweiligen Dateneingangs übernommen. Die Selbsthaltebedingung ist wirksam, wenn CLEAR und LOAD sowie die selektiven Funktionen SET und RES inaktiv sind. Eine selektive Funktion ist dann aktiv (also in der betreffenden Bitposition wirksam), wenn der zugehörige Dateneingang mit Eins belegt ist. Dann wird die Selbsthaltung getrennt. Ist RES aktiv, so liegt am D-Eingang des Flipflops eine Null an, ist SET aktiv, eine Eins.



**Abb. 4.63** Register mit Einzelbitzugriff. Jede Bitposition ist an den entsprechenden Ausgang des gemeinsamen Bitadressdecoders angeschlossen. Das Beispiel zeigt die Bitposition 3. 1 - Eingangsbelegung übernehmen; 2 - Selbsthaltung trennen, wenn Bitposition ausgewählt.



**Abb. 4.64** Register mit selektivem Setzen und Löschen. Das Beispiel zeigt die Bitposition 0. 1 - Laden; 2 - Setzen; 3 - Selbsthaltung trennen, wenn SET oder RES aktiv.