Sequenzielle Schaltungen (1)

Sequenzielle Schaltung, Schaltwerk:

- Schaltung, deren Ausgänge sowohl von den momentan anliegenden als auch von früheren Eingangsbelegungen abhängen, beispielsweise $y = f(x, y)$
- Wesentliche Elemente einer CPU wie Register, Zähler oder Schieberegister werden durch sequentielle Schaltungen realisiert

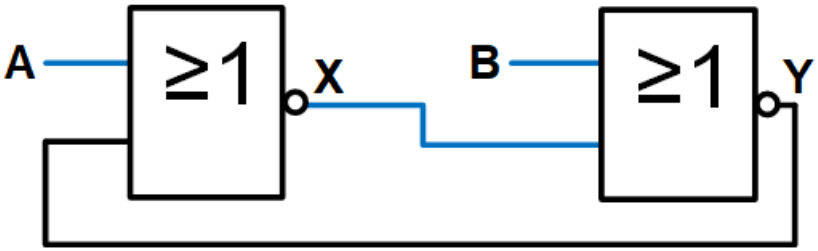
Flip-Flop (Basiseinheit einer sequenziellen Schaltung)**:**

- Eine Box mit 2 oder 3 Eingängen und 2 Ausgängen, wobei der 2. Ausgang in der Regel das Komplement des 1. Ausgangs darstellt.

Eigenschaften:

- sind bi-stabil, d. h. verfügen über zwei stabile Zustände
  (im Gegensatz zum Monoflop mit nur einem stabilen Zustand)
- haben ein „Gedächtnis“, da Ausgang nicht nur abhängig vo aktuellen Eingangsbelegungen sondern *auch von früheren* Eingangsbelegungen
- kleinste Speicherzelle, speichern *genau eine* Informationseinheit (1 Bit)
- Ausdruck „Flip-Flop“ ist lautnachahmend. Er stammt aus einer Zeit, in der sie durch elektromagnetische Relais realisiert wurden (z.B. Relaisrechner Z3)

Sequenzielle Schaltungen (2a)

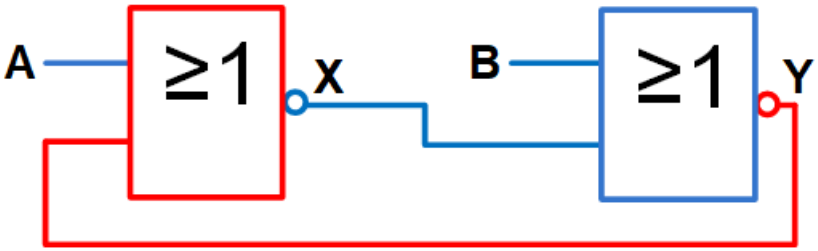


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2b)

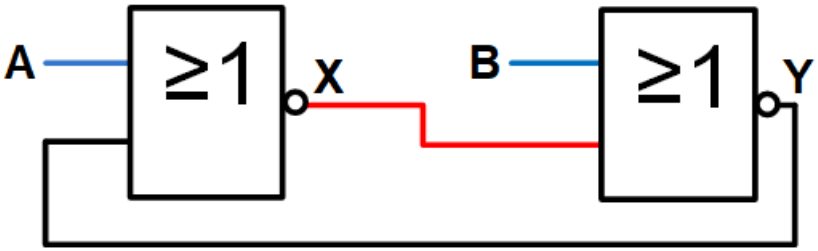


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2c)

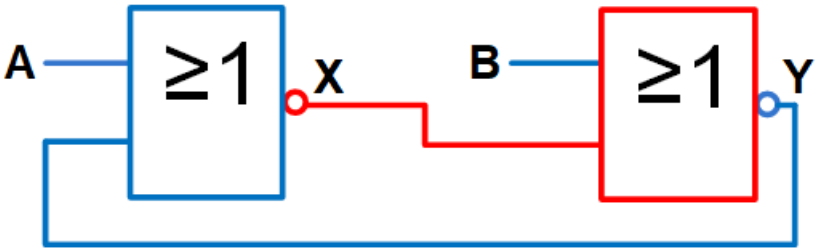


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2d)

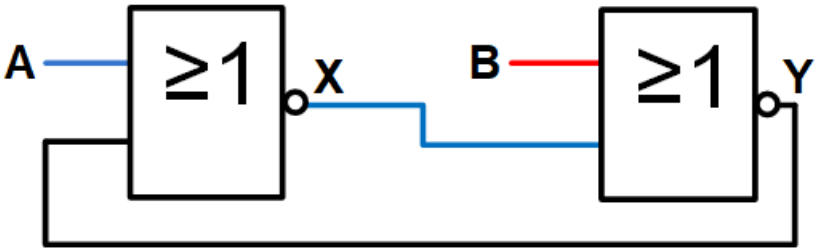


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2e)

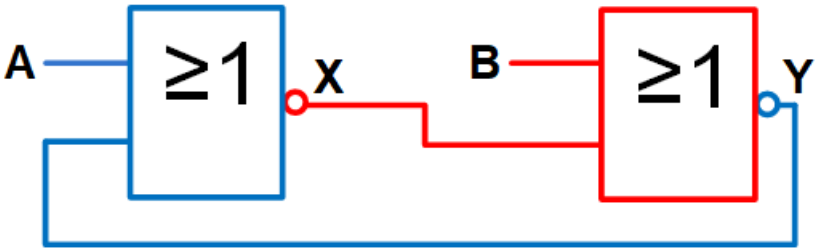


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2f)

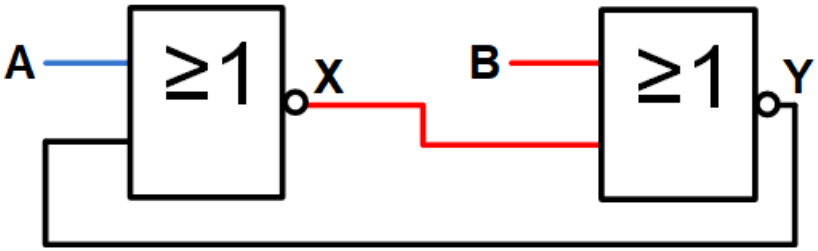


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2g)

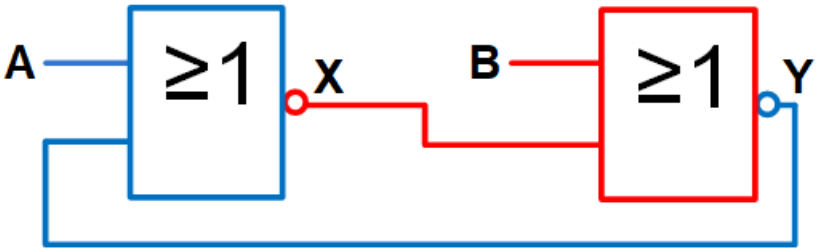


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2h)

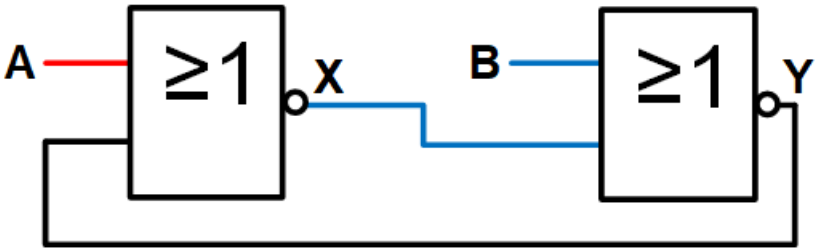


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2i)

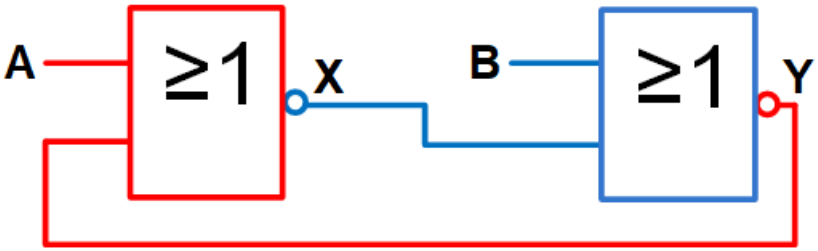


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2j)

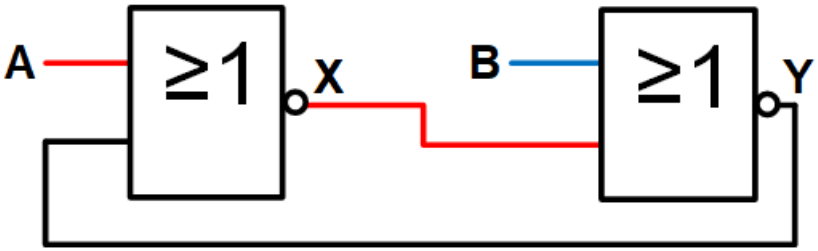


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2k)

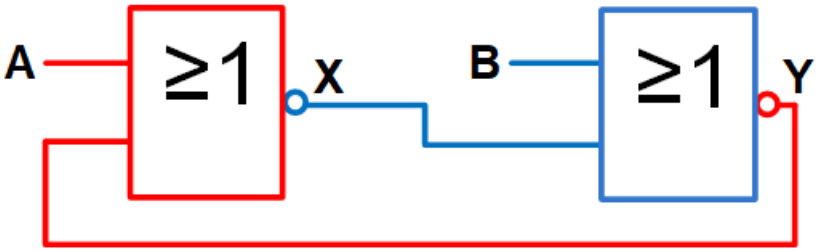


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2l)

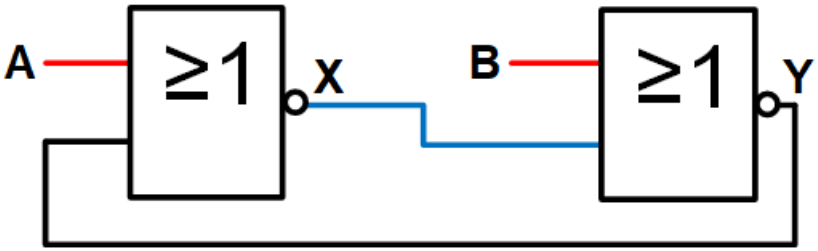


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2m)

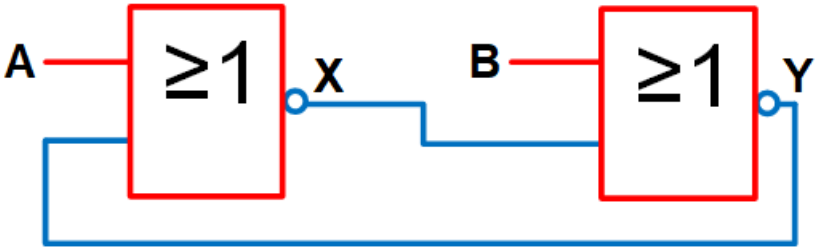


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2n)

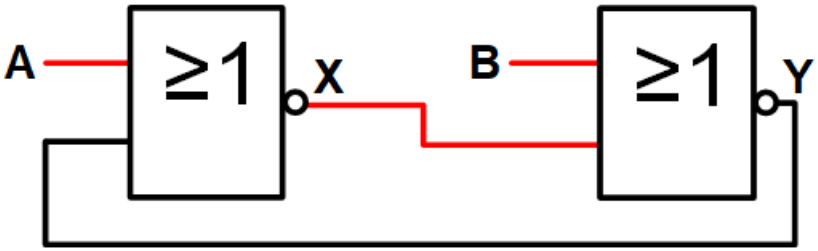


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0 (?)

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2o)

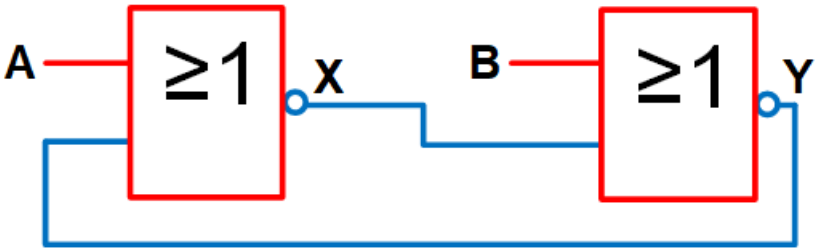


Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0 (?)
1	1	1	

Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2p)



Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0 (?)
1	1	1	0 (?)

Wahrheitstabelle des NOR-Gatters

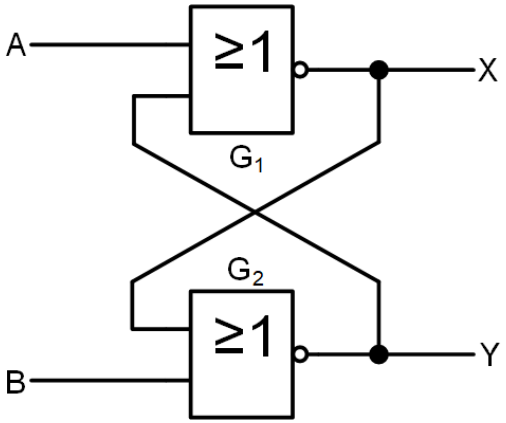
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (2)

Zustandstabelle

Eingänge			Ausgänge
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0 (?)
1	1	1	0 (?)

Flip-Flop aus kreuzgekoppelten NOR-Gattern



Wahrheitstabelle des NOR-Gatters

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Sequenzielle Schaltungen (3)

Zustandstabelle

Eingänge			Ausgänge	
A (R)	B (S)	X_{alt} (Q_{alt})	X_{neu} (Q_{neu})	
0	0	0	0	keine Änderung
0	0	1	1	
0	1	0	1	setzen
0	1	1	1	
1	0	0	0	rücksetzen
1	0	1	0	
1	1	0	0	unbe-stimmt
1	1	1	0	

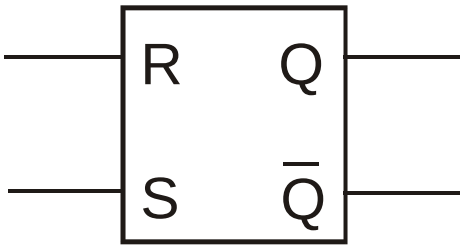
Unbestimmt:

- Implementierungsabhängig (NAND / NOR)
- Gleichzeitig sezuen und rücksetzen nicht möglich

Zustandsfolgetabelle

Eingänge		Ausgänge		Beschreibung
R	S	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	keine Änderung
0	1	1	0	setzen
1	0	0	1	rücksetzen
1	1	X	X	verboten

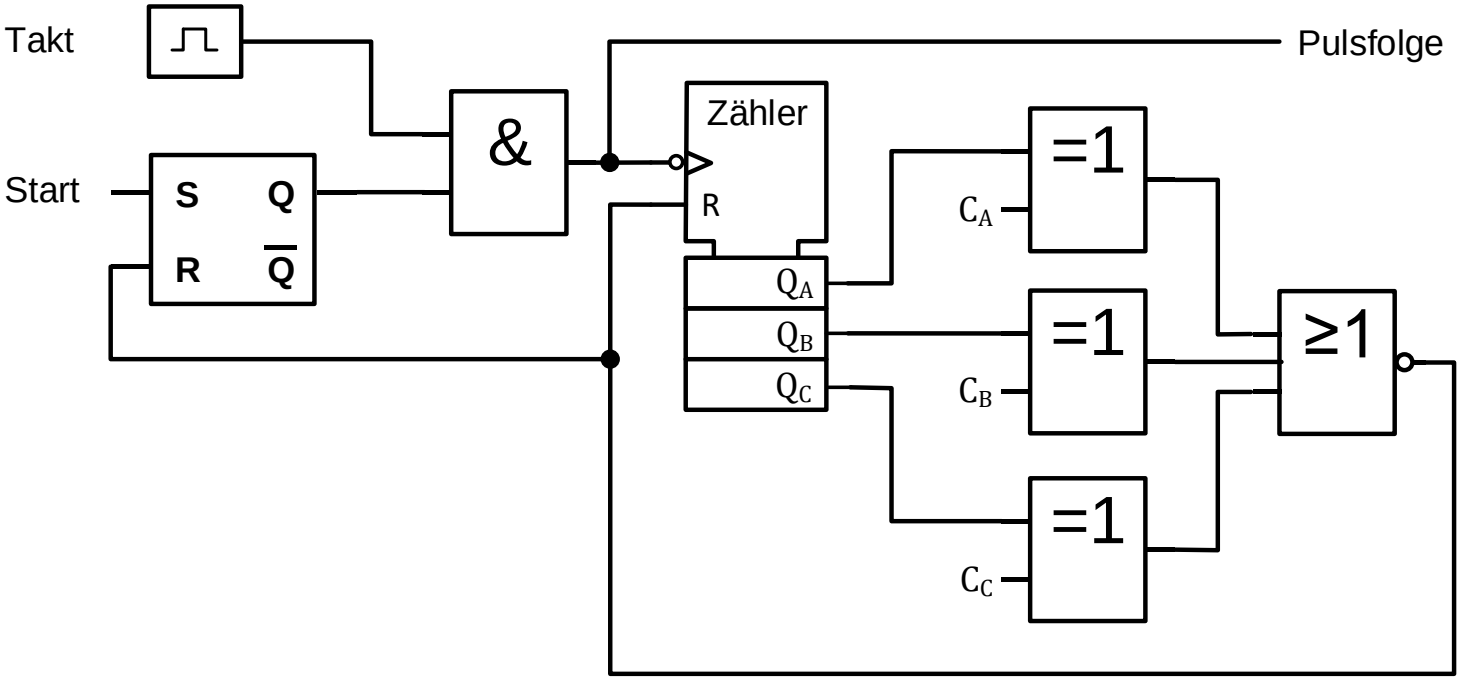
Darstellung des RS-Flip-Flops



Sequenzielle Schaltungen (4)

Beispiel:

Impulsfolgengenerator, realisiert mit Hilfe eines RS Flip-Flops



Sequenzielle Schaltungen (5)

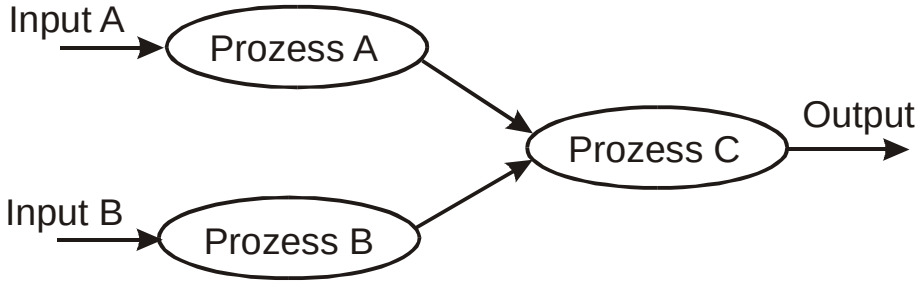
Getaktete Flip-Flops:

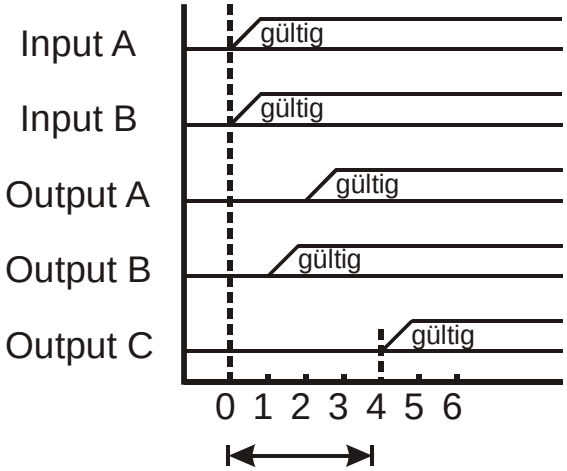
Ein zusätzliches Taktsignal, geliefert von einem Taktgeber wie z.B. einer Uhr, ermöglicht erst die Reaktion auf anliegende Inputwerte.

Idee:

Kontrolle erhalten darüber, dass Schaltnetze (bzw. die durch Schaltnetze realisierten Prozesse) nur auf gültige Inputwerte reagieren.

Beispiel:

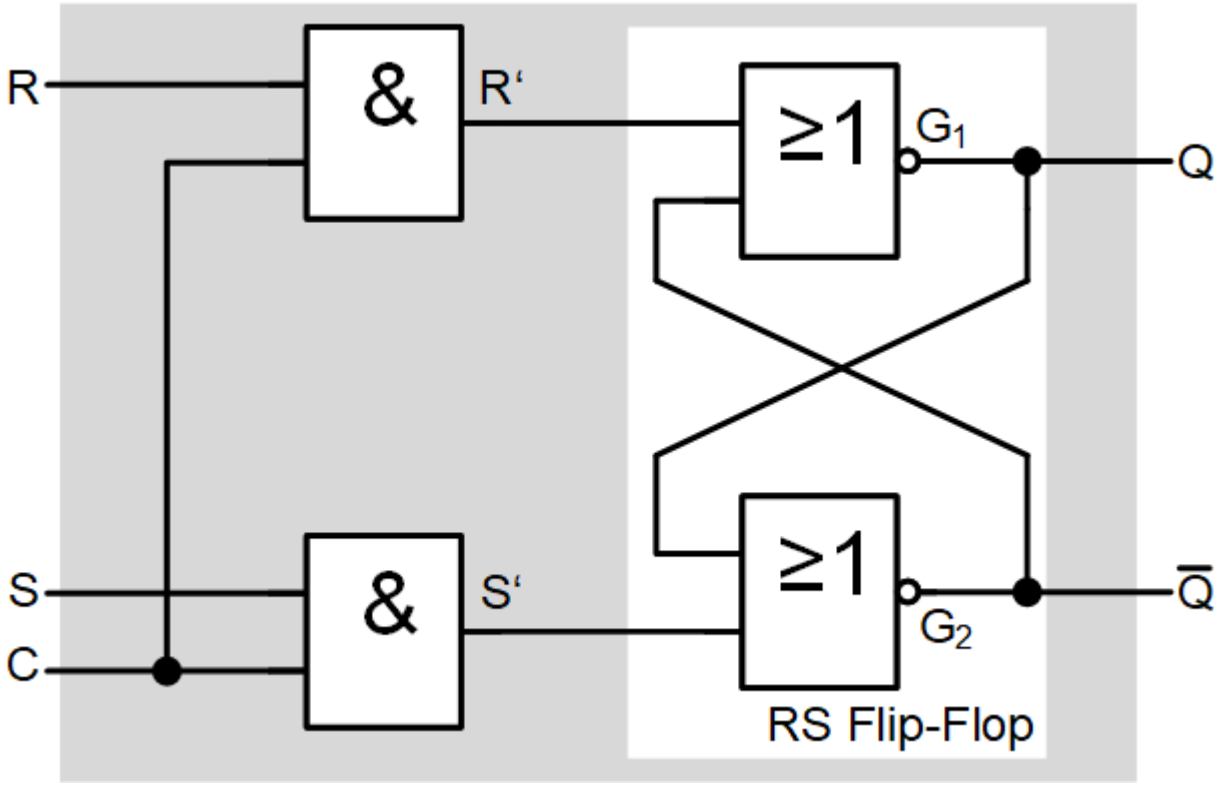




Verzögerung, bis Output C gültig ist

Sequenzielle Schaltungen (6)

Getaktetes RS-Flip-Flop:.



R- oder S-Eingang des „inneren“ Flip-Flops werden nur dann aktiviert, wenn am Eingang C (Clock, Takt) eine 1 anliegt.
Das Flip-Flop ist somit taktzustandsgesteuert.

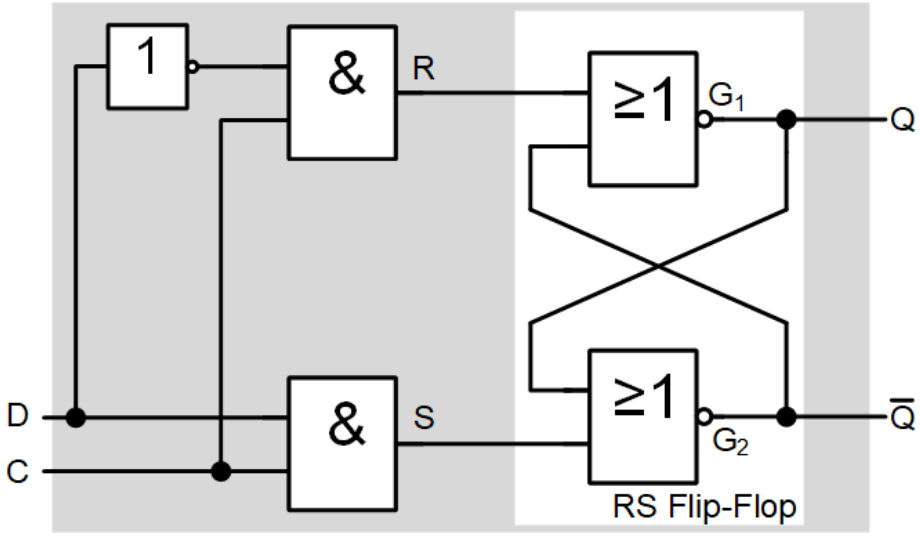
Sequenzielle Schaltungen (7)

D Flip-Flop

Zustandstabelle

Eingänge			Ausgänge
C	D	Q_{alt}	Q_{neu}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

D-Flip-Flop aus Gattern



Sequenzielle Schaltungen (8)

D Flip-Flop

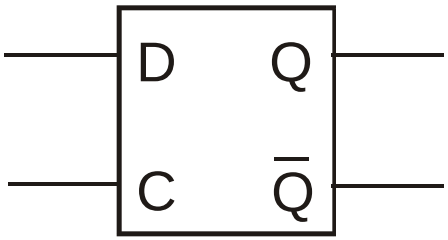
Zustandstabelle

Eingänge			Ausgänge	
C	D	Q_{alt}	Q_{neu}	
0	0	0	0	keine Änderung
0	0	1	1	
0	1	0	0	keine Änderung
0	1	1	1	
1	0	0	0	D über-nehmen
1	0	1	0	
1	1	0	1	D über-nehmen
1	1	1	1	

Zustandsfolgetabelle

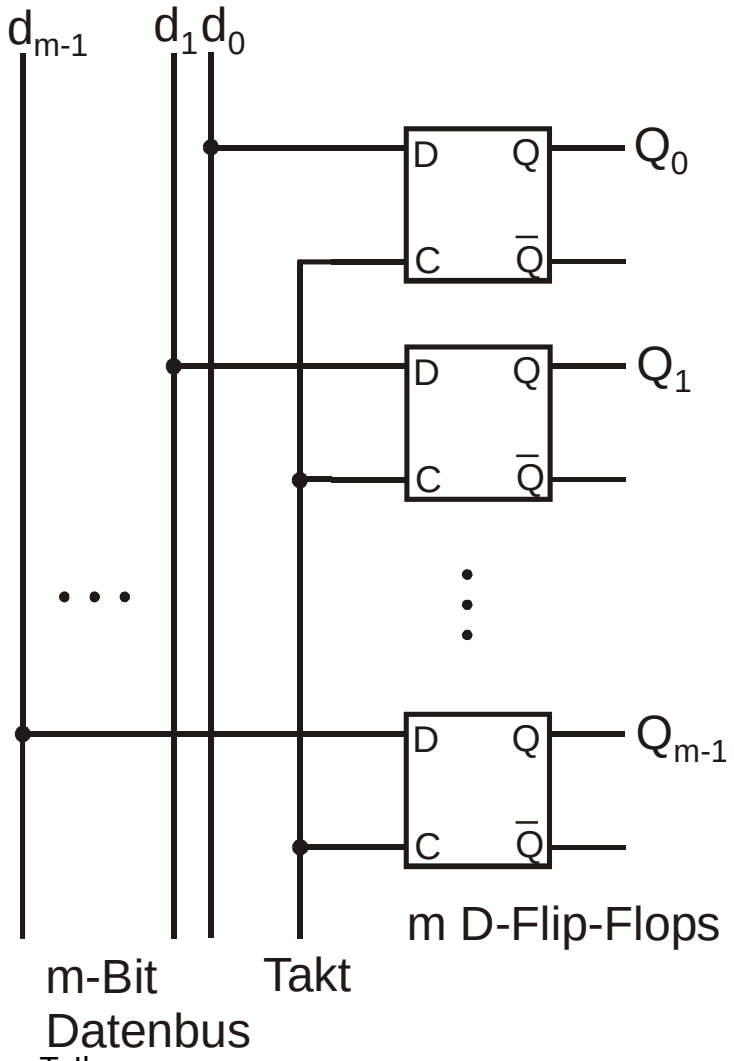
Eingänge		Ausgänge		Beschreibung
C	D	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	keine Änderung
0	1	Q	$\overline{Q}$	Keine Änderung
1	0	0	1	D übernehmen
1	1	1	0	D übernehmen

Darstellung des D-Flip-Flops



Sequenzielle Schaltungen (9)

Applikationsbeispiel 1:

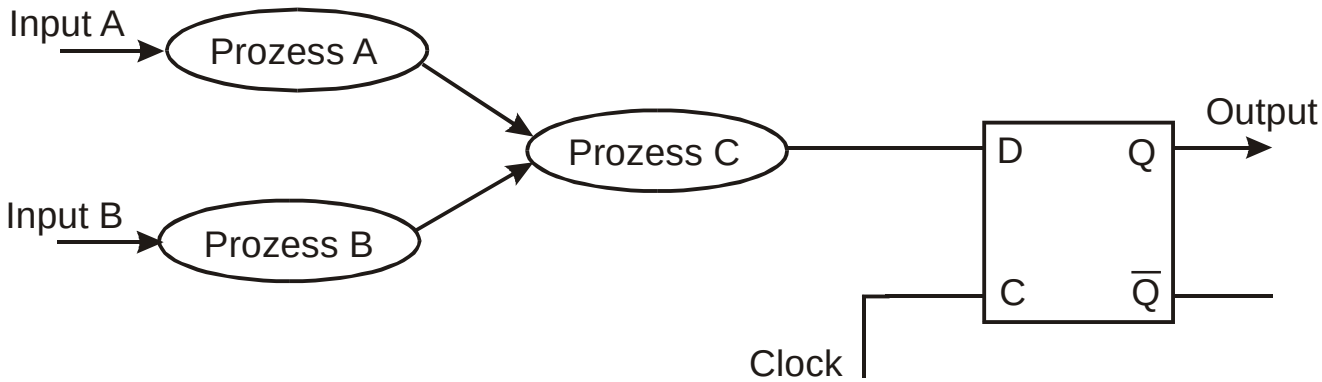


Latch (Zwischenspeicher):

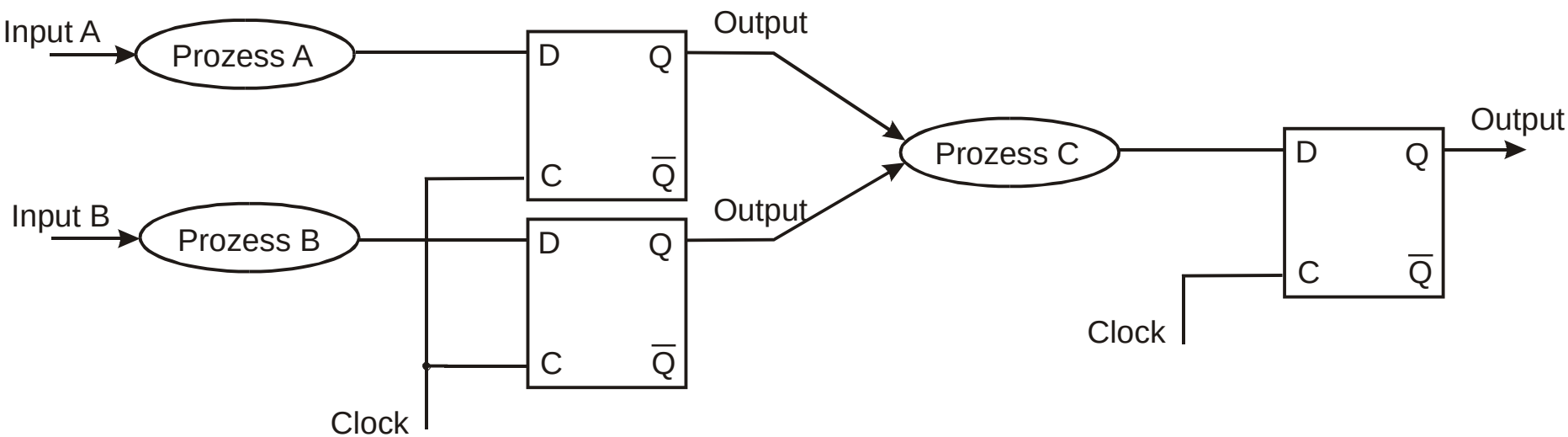
Daten werden nur dann übernommen und zwischengespeichert, wenn an der Clock-Leitung logisch 1 anliegt.

Sequenzielle Schaltungen (10)

Applikationsbeispiel 2a:

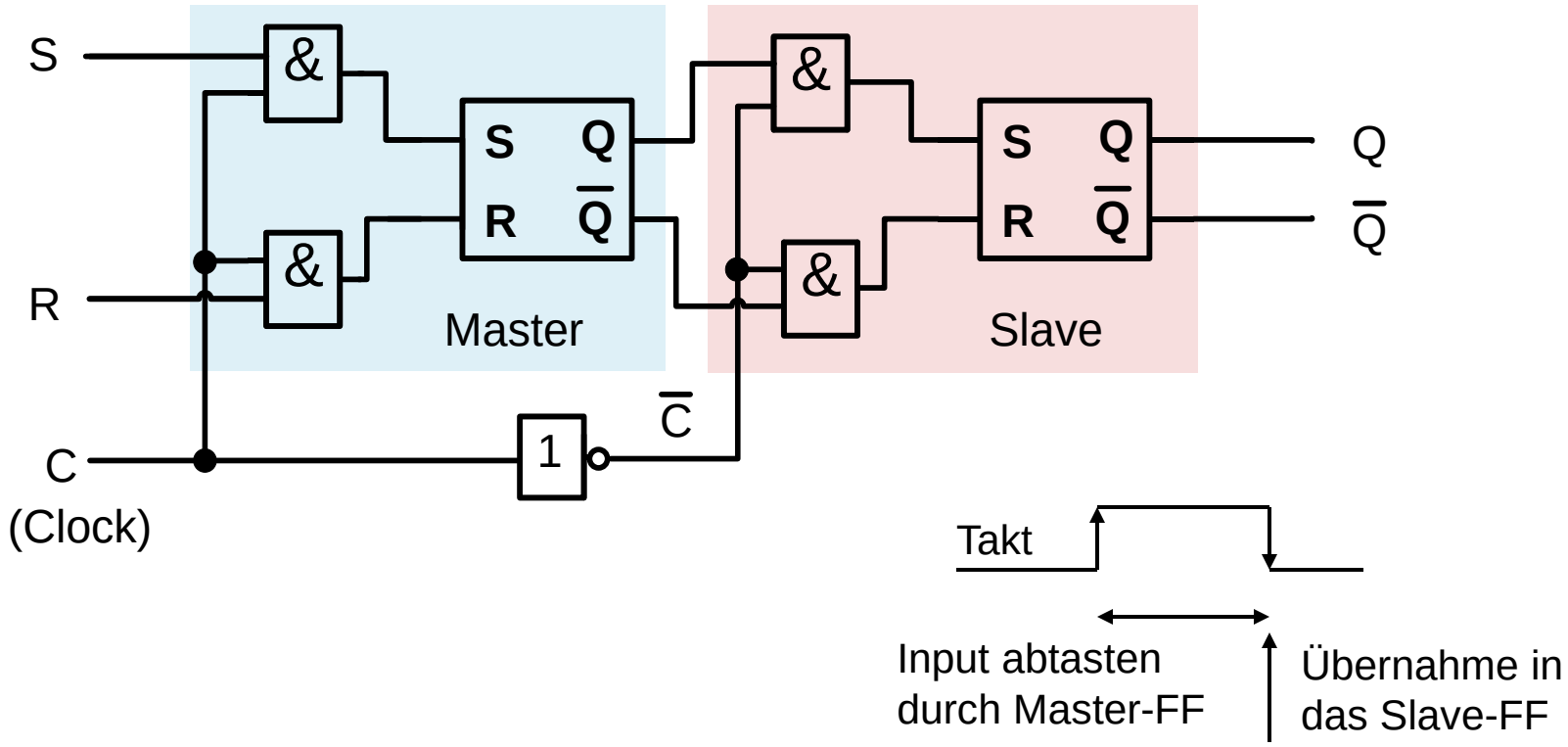


Applikationsbeispiel 2b:



Sequenzielle Schaltungen (11)

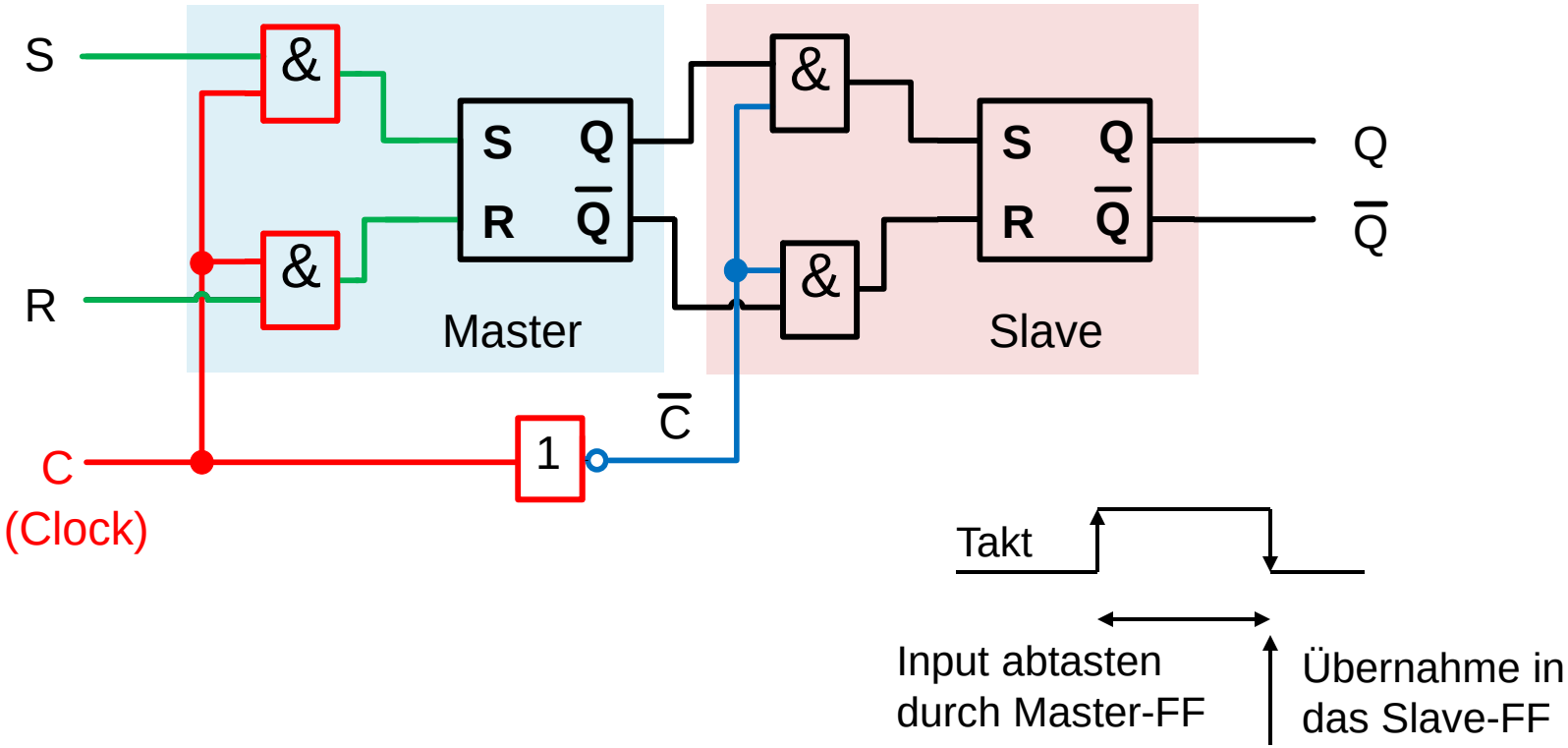
RS Master-Slave Flip-Flop:



Vorteil: Vermeiden des „Durchbrechens“ von Eingangsinformationen zum Ausgang, da Eingangsinformationen erst bei inaktivem Taktpegel in das Slave-FF übernommen werden.

Sequenzielle Schaltungen (11a)

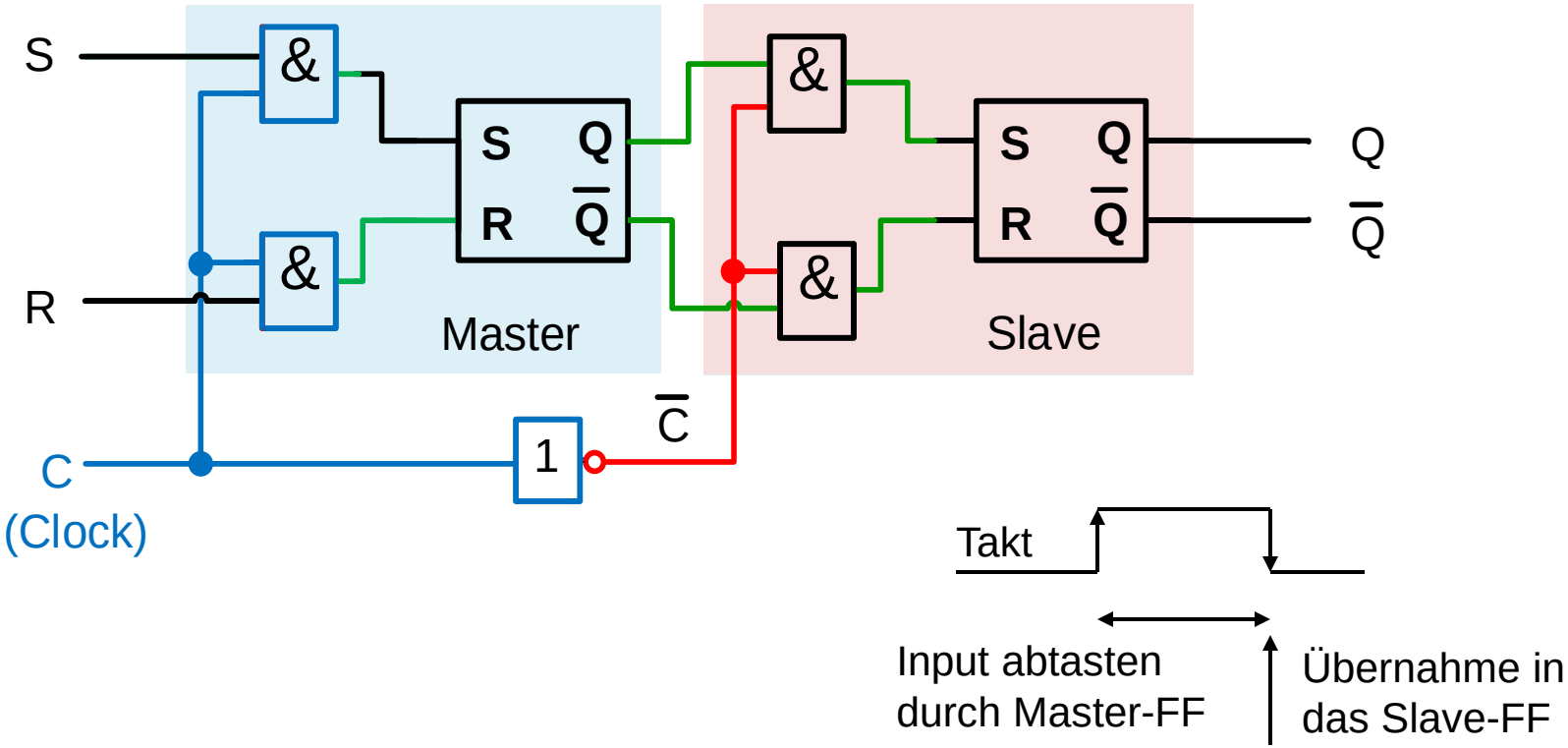
RS Master-Slave Flip-Flop:



Vorteil: Vermeiden des „Durchbrechens“ von Eingangsinformationen zum Ausgang, da Eingangsinformationen erst bei inaktivem Taktpegel in das Slave-FF übernommen werden.

Sequenzielle Schaltungen (11b)

RS Master-Slave Flip-Flop:



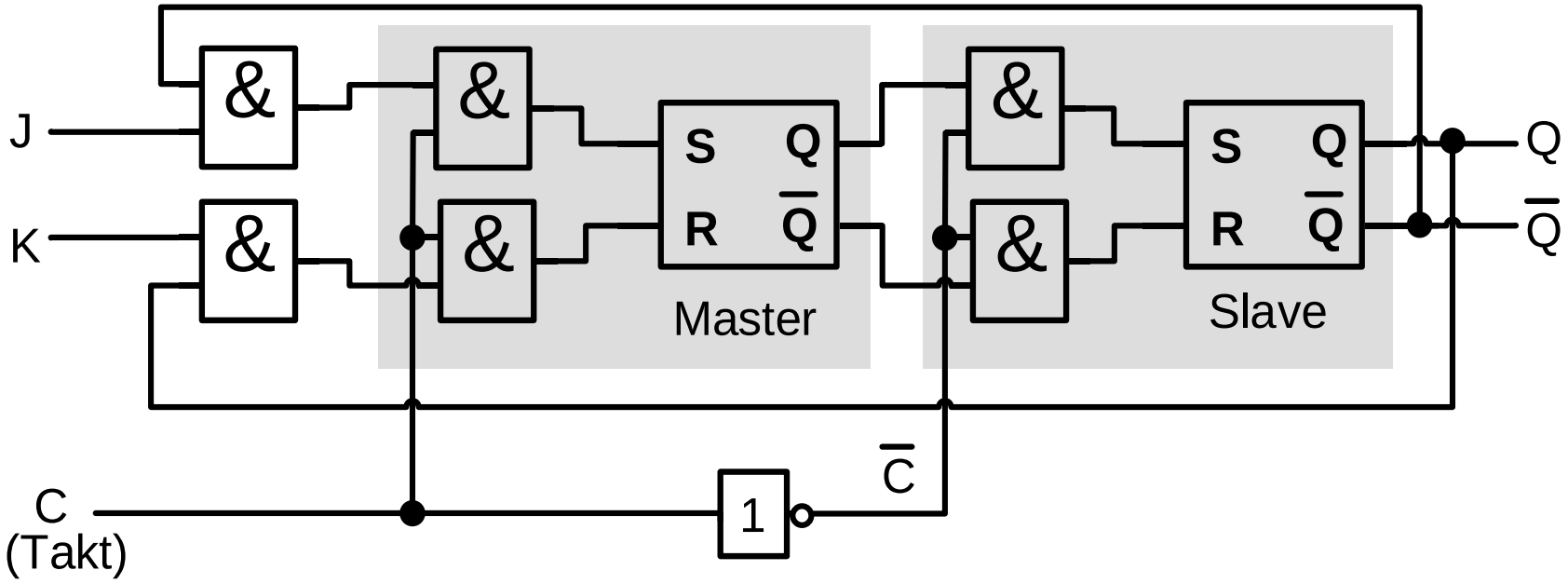
Vorteil: Vermeiden des „Durchbrechens“ von Eingangsinformationen zum Ausgang, da Eingangsinformationen erst bei inaktivem Taktpegel in das Slave-FF übernommen werden.

Sequenzielle Schaltungen (12)

JK Flip-Flop:

gewünschte Funktionsweise: getaktetes RS Flip-Flop mit $S = J\overline{Q}$ und $R = KQ$

Schaltung des JK-FF:



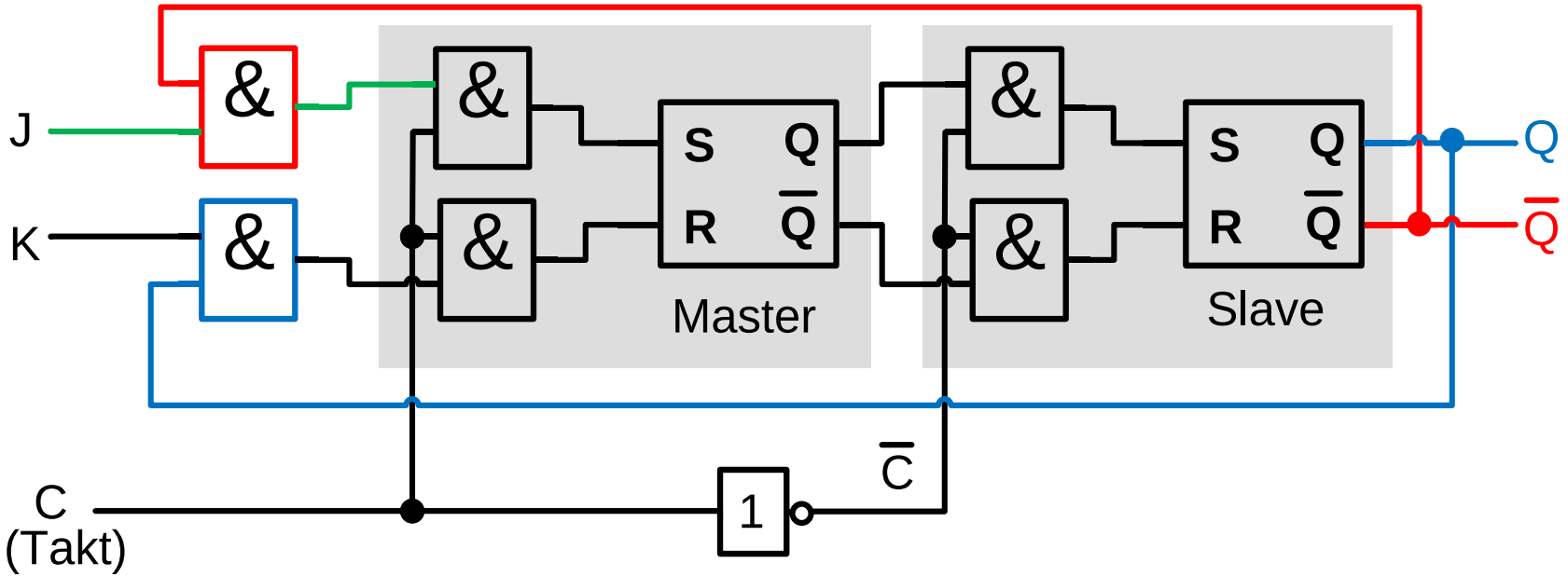
Vorteil: das Problem mit dem verbotenen Zustand ist beseitigt.

Sequenzielle Schaltungen (12a)

JK Flip-Flop:

gewünschte Funktionsweise: getaktetes RS Flip-Flop mit $S = J\overline{Q}$ und $R = KQ$

Schaltung des JK-FF:



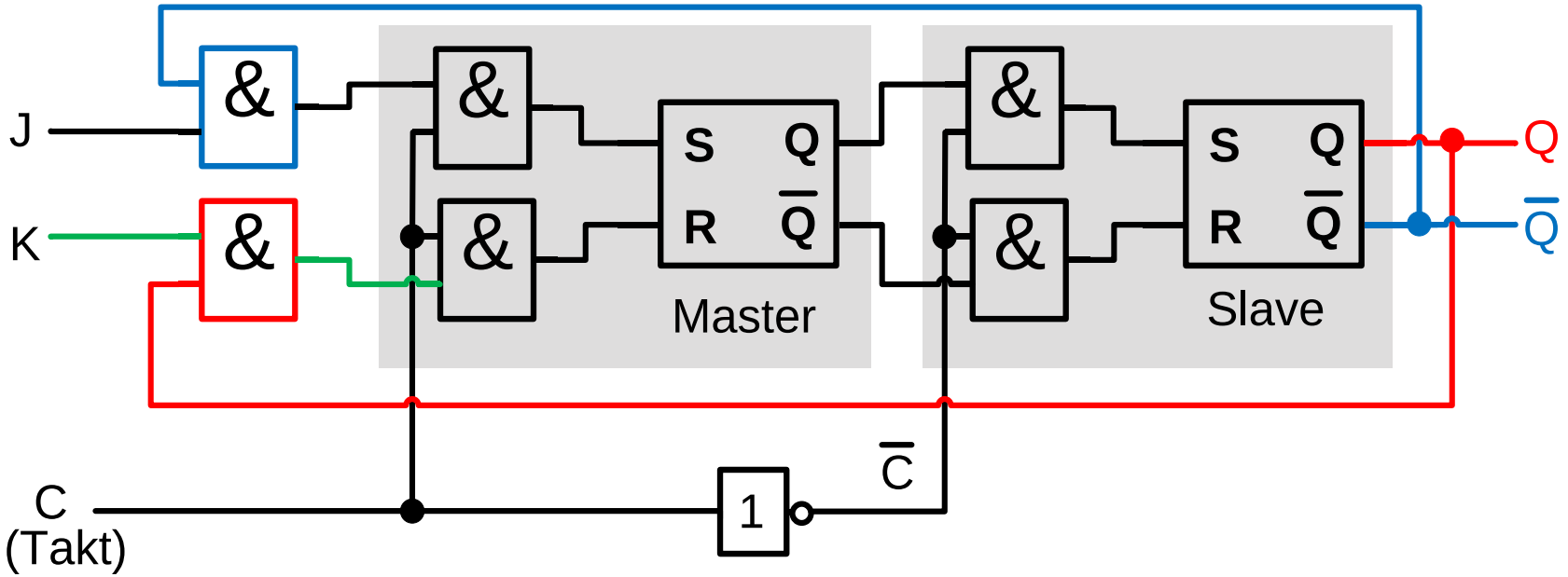
Vorteil: das Problem mit dem verbotenen Zustand ist beseitigt.

Sequenzielle Schaltungen (12b)

JK Flip-Flop:

gewünschte Funktionsweise: getaktetes RS Flip-Flop mit $S = J\overline{Q}$ und $R = KQ$

Schaltung des JK-FF:



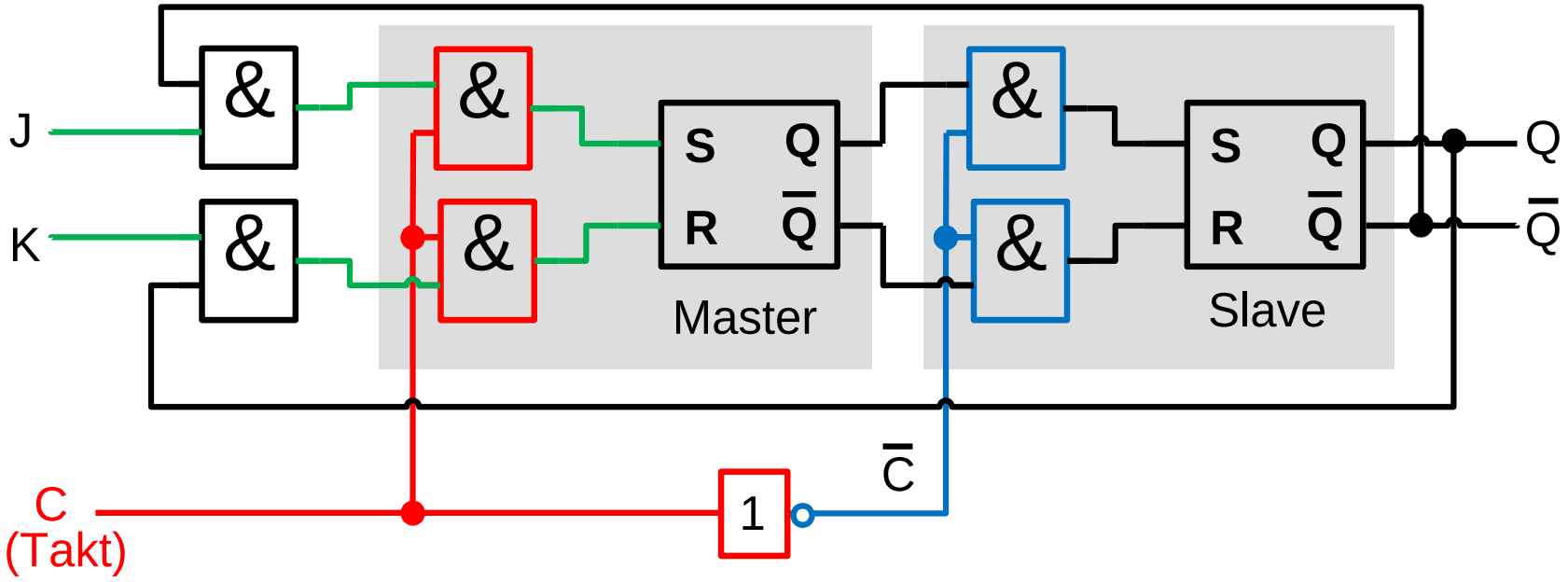
Vorteil: das Problem mit dem verbotenen Zustand ist beseitigt.

Sequenzielle Schaltungen (12c)

JK Flip-Flop:

gewünschte Funktionsweise: getaktetes RS Flip-Flop mit $S = J\overline{Q}$ und $R = KQ$

Schaltung des JK-FF:



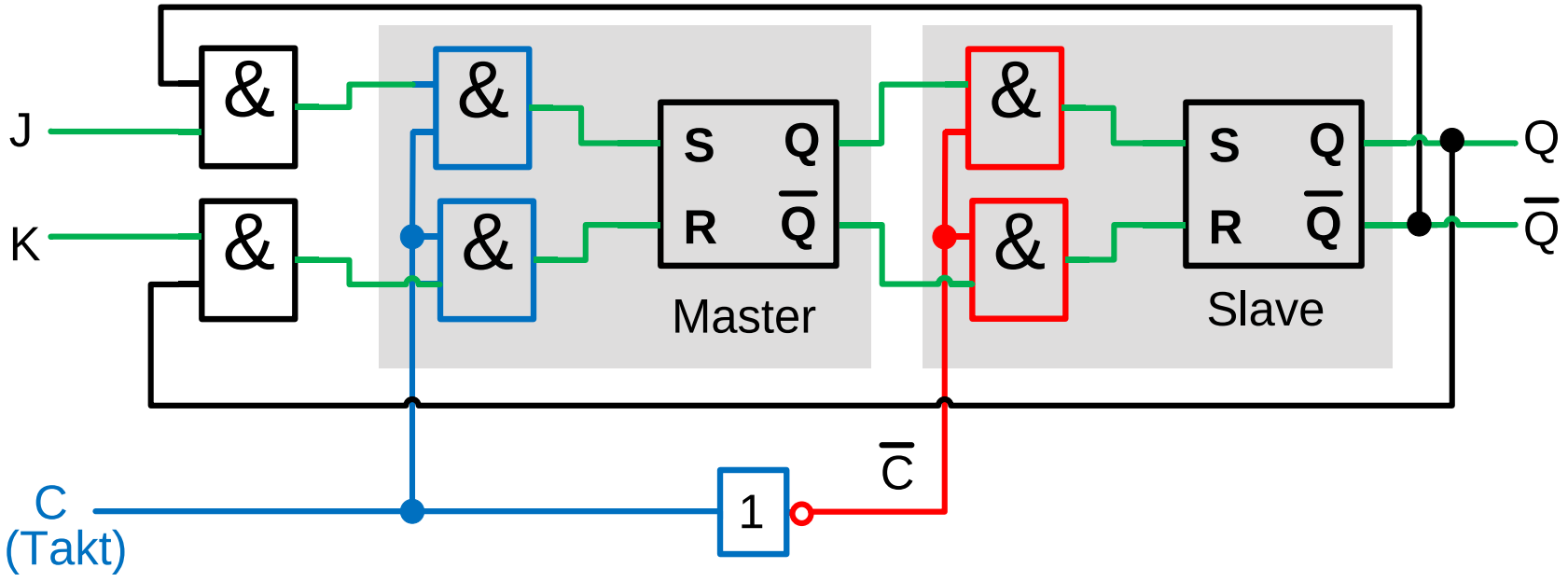
Vorteil: das Problem mit dem verbotenen Zustand ist beseitigt.

Sequenzielle Schaltungen (12d)

JK Flip-Flop:

gewünschte Funktionsweise: getaktetes RS Flip-Flop mit $S = J\overline{Q}$ und $R = KQ$

Schaltung des JK-FF:



Vorteil: das Problem mit dem verbotenen Zustand ist beseitigt.

Sequenzielle Schaltungen (13)

JK Flip-Flop

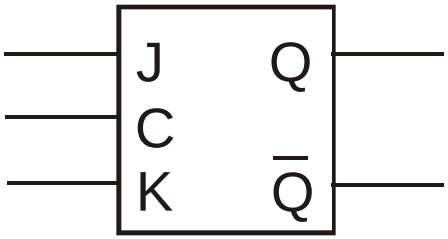
Zustandstabelle

Eingänge			Ausgänge	
J	K	Q_{alt}	Q_{neu}	
0	0	0	0	keine Änderung
0	0	1	1	
0	1	0	0	rück-setzen
0	1	1	0	
1	0	0	1	setzen
1	0	1	1	
1	1	0	1	toggle
1	1	1	0	

Zustandsfolgetabelle

Eingänge		Ausgänge		Beschreibung
J	K	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	keine Änderung
0	1	0	1	rücksetzen
1	0	1	0	setzen
1	1	$\overline{Q}$	Q	toggle

Darstellung des JK-Flip-Flops

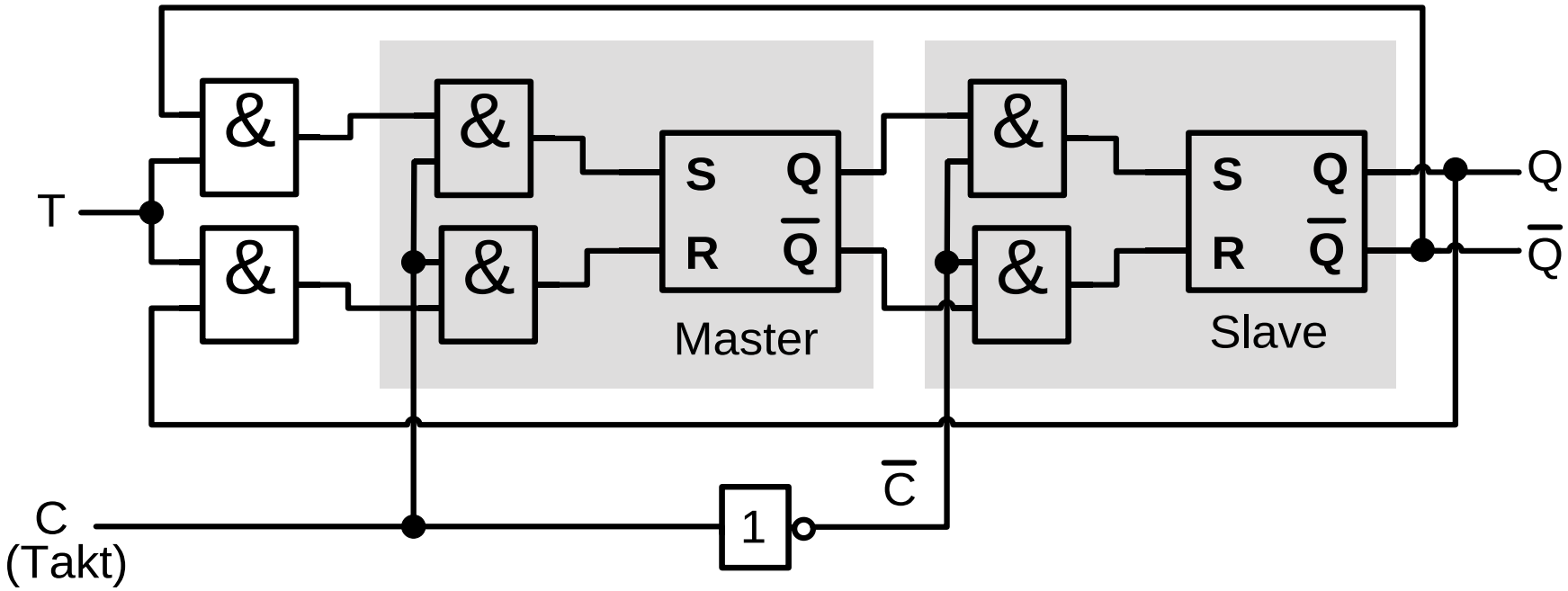


Sequenzielle Schaltungen (14)

T Flip-Flop:

JK-Flip Flop mit $T = J = K$

Schaltung des T-FF:



J- und K-Eingang sind zu einem einzigen Eingang verbunden.

Sequenzielle Schaltungen (15)

T Flip-Flop

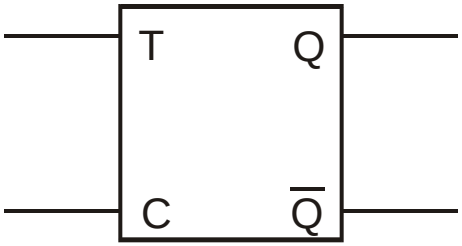
Zustandstabelle

Eingänge		Ausgänge	
T	Q_{alt}	Q_{neu}	
0	0	0	keine Änderung
0	1	1	
1	0	1	toggle
1	1	0	

Zustandsfolgetabelle

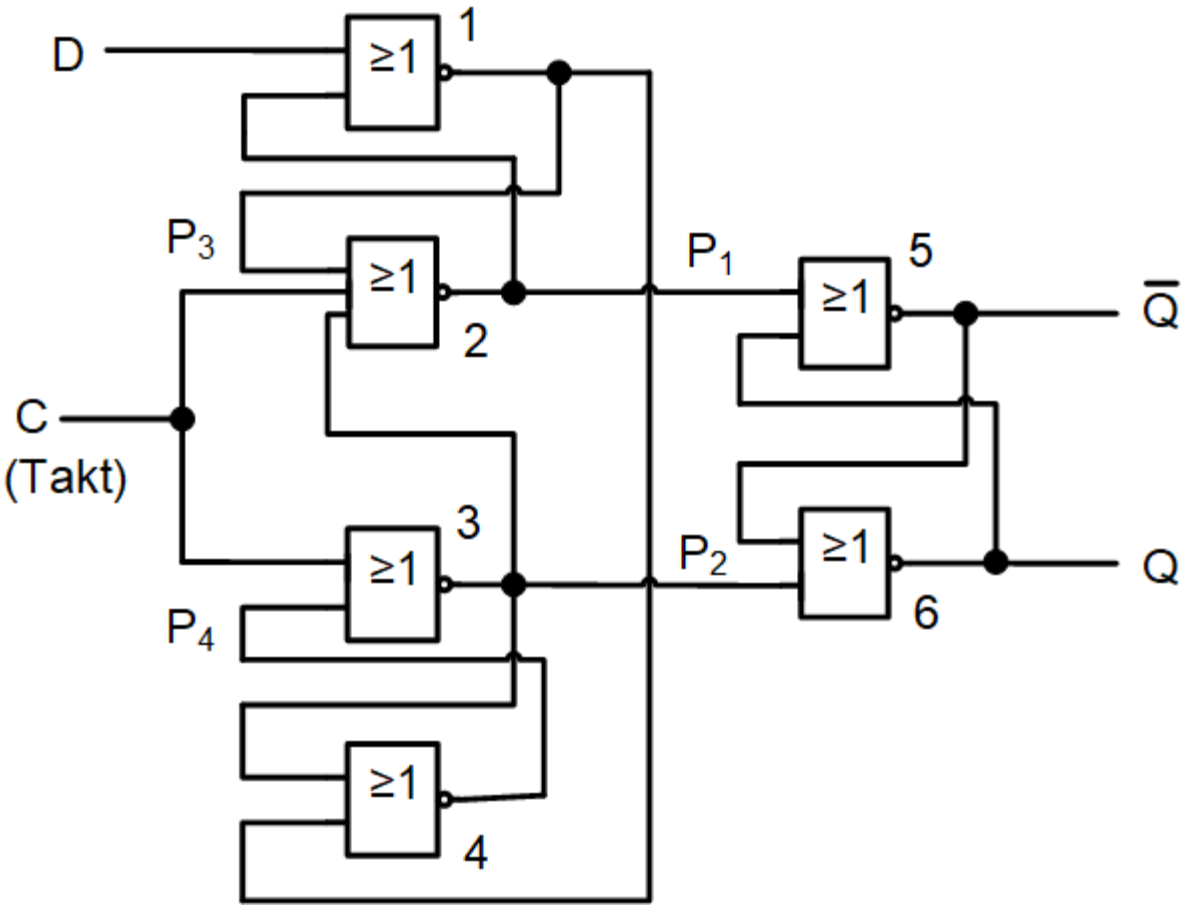
Eingänge	Ausgänge		Beschreibung
T	Q	$\overline{Q}$	
0	Q	$\overline{Q}$	keine Änderung
1	$\overline{Q}$	Q	toggle

Darstellung des T-Flip-Flops



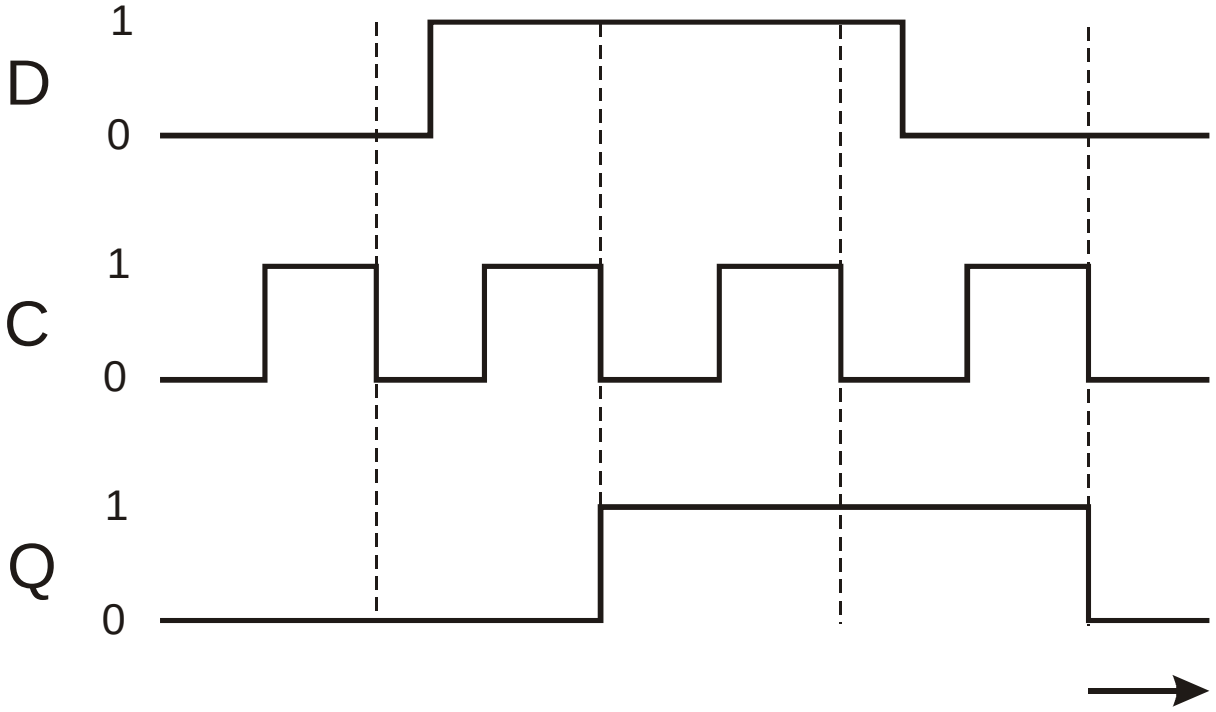
Sequenzielle Schaltungen (16)

Negativ flankengetriggertes D-Flip-Flop:



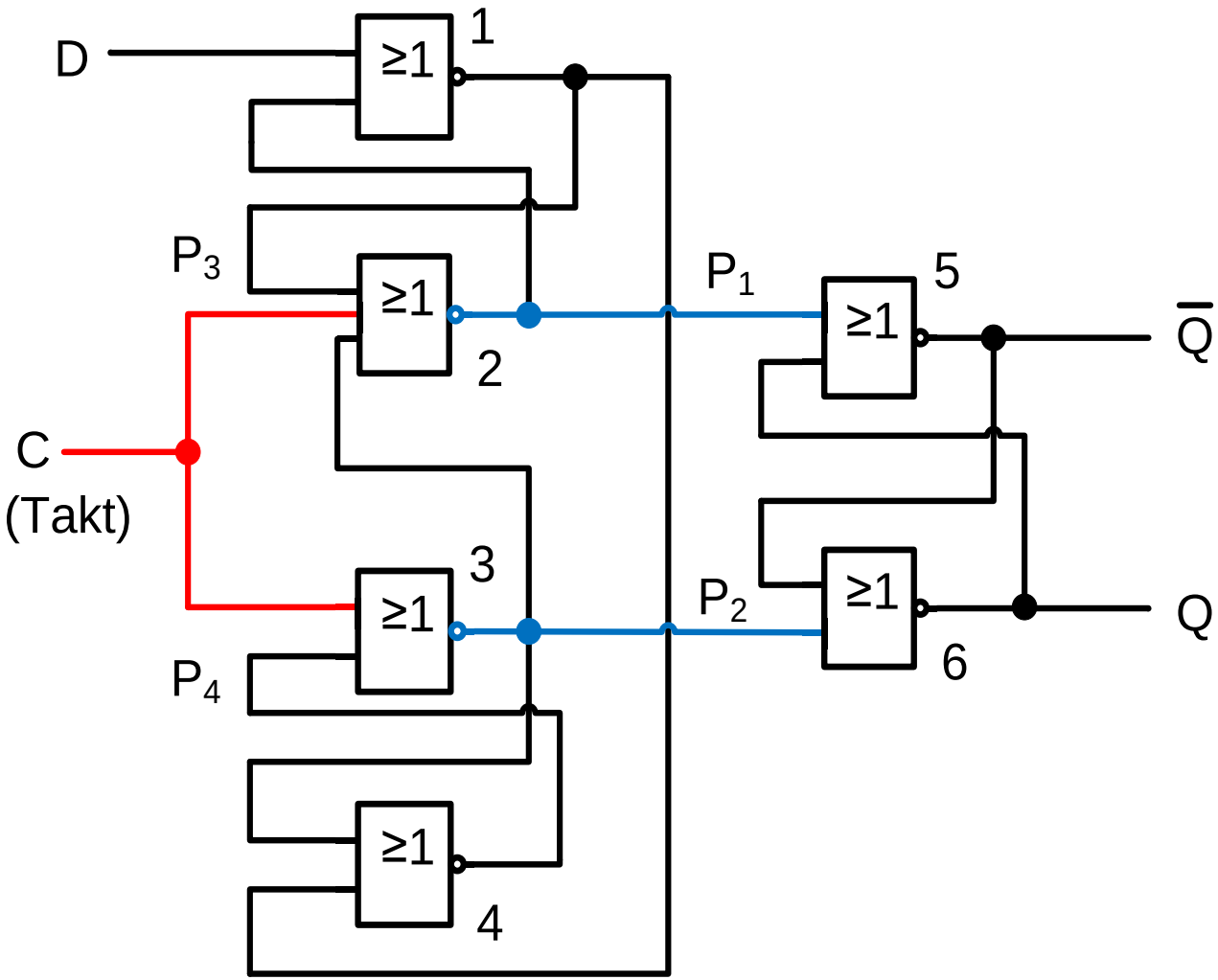
Sequenzielle Schaltungen (16a)

Negativ flankengetriggertes D-Flip-Flop:



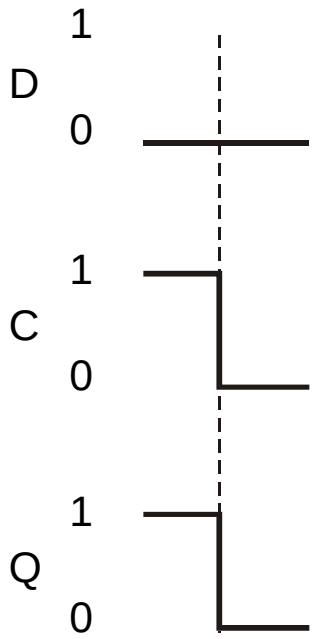
Sequenzielle Schaltungen (16b)

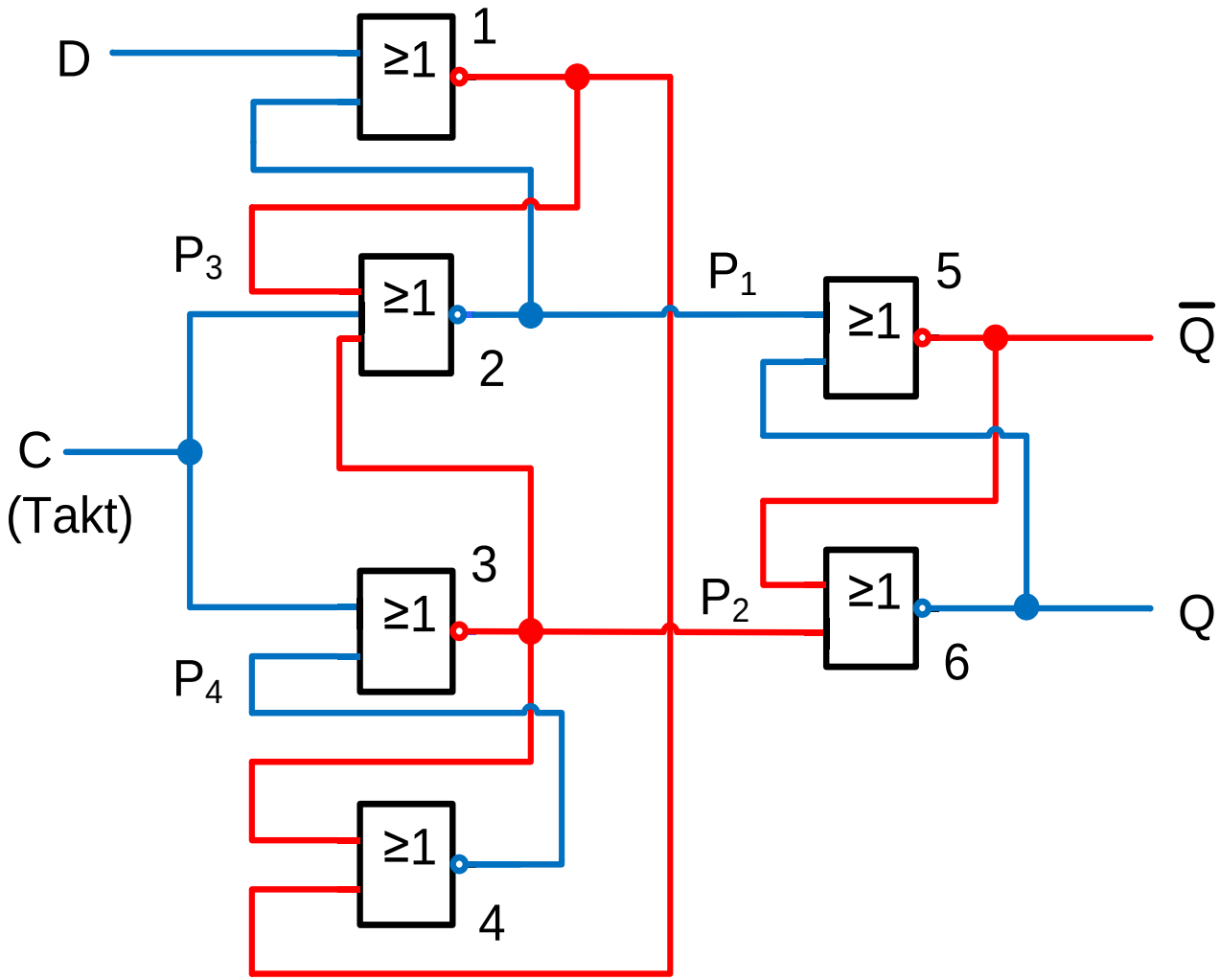
Negativ flankengetriggertes D-Flip-Flop:



Sequenzielle Schaltungen (16c)

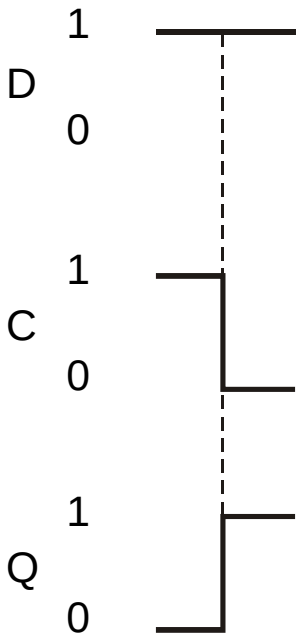
Negativ flankengetriggertes D-Flip-Flop:

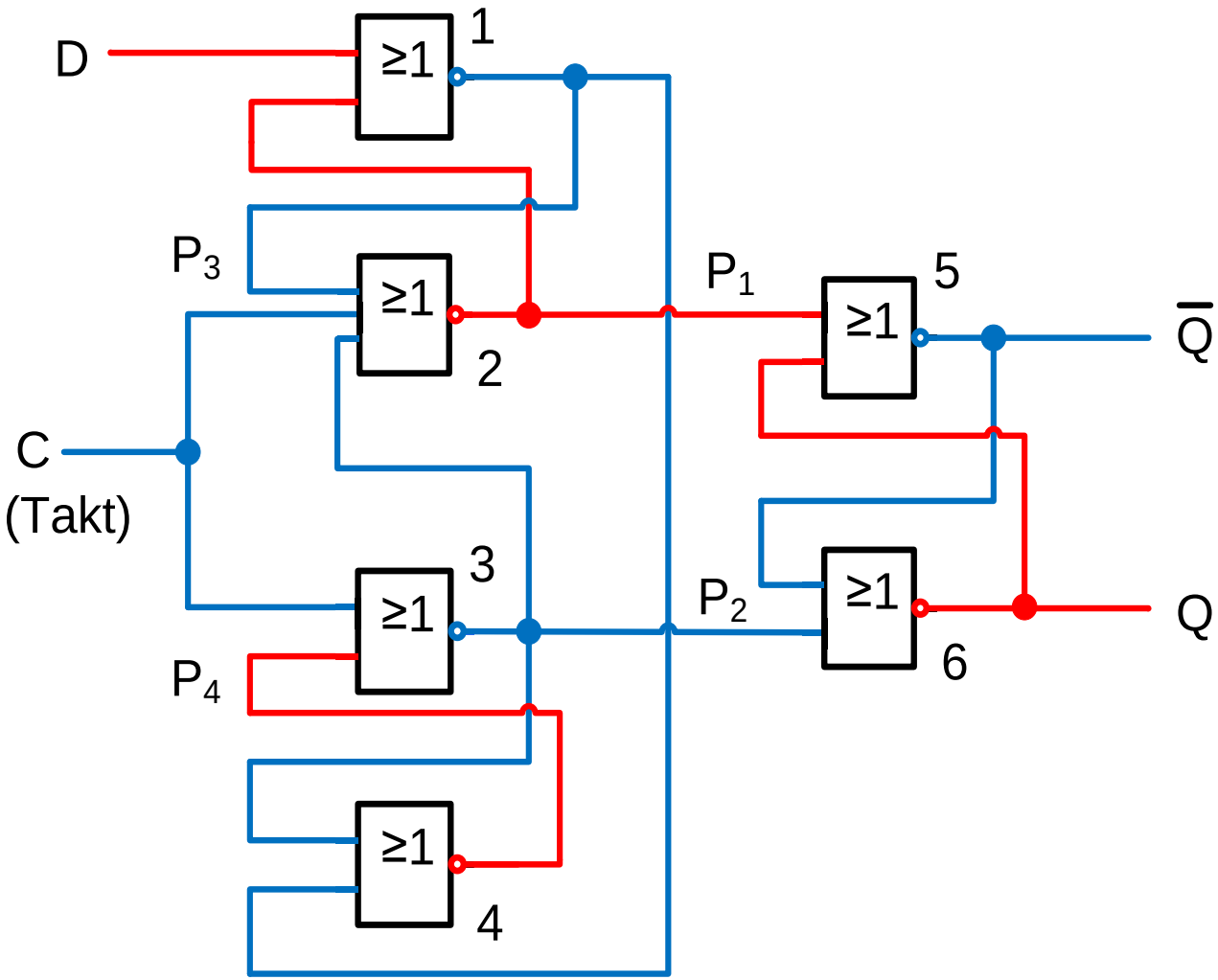




Sequenzielle Schaltungen (16d)

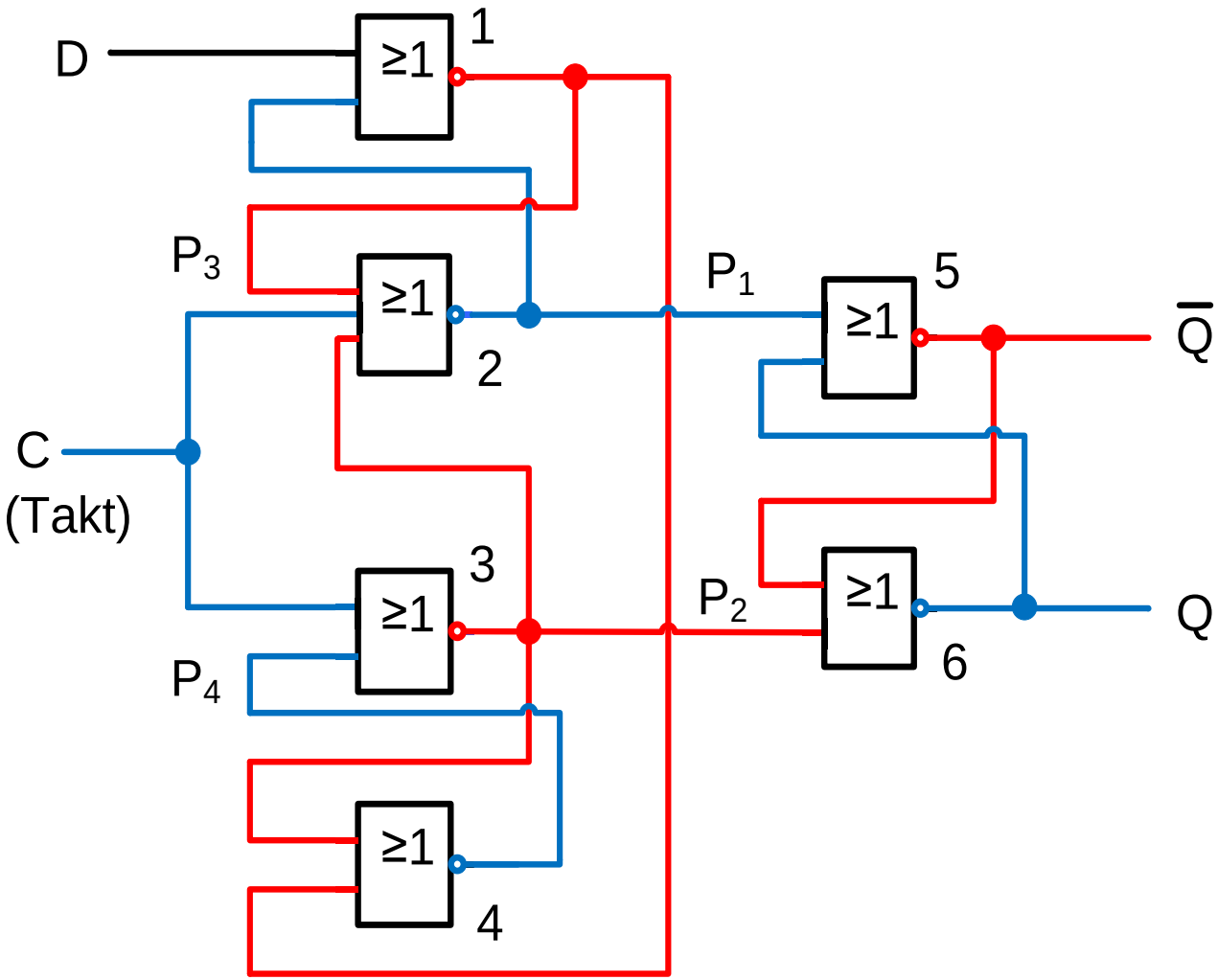
Negativ flankengetriggertes D-Flip-Flop:





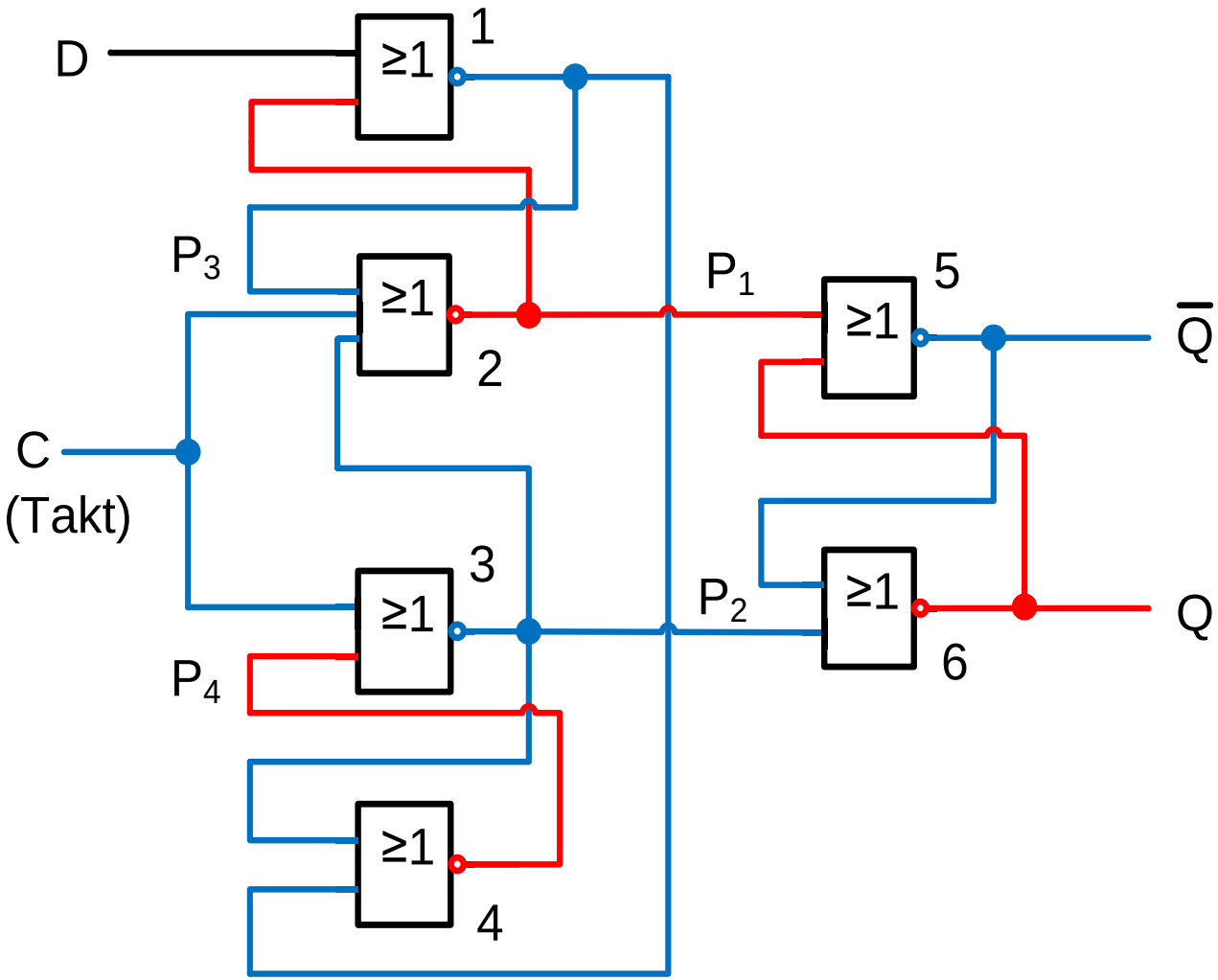
Sequenzielle Schaltungen (16e)

Negativ flankengetriggertes D-Flip-Flop:



Sequenzielle Schaltungen (16f)

Negativ flankengetriggertes D-Flip-Flop:



Sequenzielle Schaltungen (17)

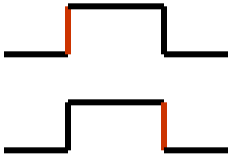
Taktung von Flip-Flops:

Bestimmt den Zeitpunkt der Übernahme von Eingangsinformationen

Taktzustandsgesteuert (pegelgesteuert, level sensitive)

An Eingang liegende Informationen werden während der gesamten Dauer des Taktimpulses übernommen.

Taktflankengesteuert (edge triggered)



Einflankensteuerung: Eingangsinformationen werden mit aktiver (positiver oder negativer) Taktflanke übernommen

Zweiflankensteuerung: Eingangsinformationen werden während der Impulsdauer des Taktes in den Zwischenspeicher übernommen und mit der aktiven Taktflanke zum Ausgang übernommen (Master-Slave-Prinzip). Es wird die letzte, während der Impulsdauer anliegende Information übernommen

Sequenzielle Schaltungen (18)

Zusammenfassung “Flip-Flops“:

Flip-Flops haben interne Zustände und externe Inputs

- Output abhängig auch vom internen Zustand
- Output abhängig von früheren Inputs („Vorgeschichte“)
- Flip-Flops sind Speicherelemente

Flip-Flops unterscheiden sich auf Grund ihrer

- Eingänge
- Taktung

Eingänge:

- RS
- D
- JK
- T

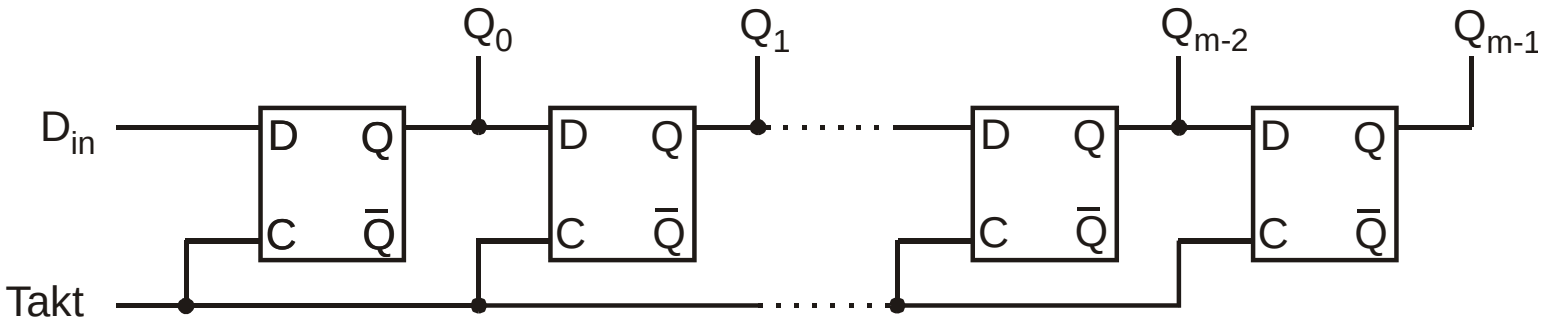
Taktung:

- Ohne Taktung
- taktzustandsgesteuert
- taktflankengesteuert
  - einflankengesteuert
  - zweiflankengesteuert (Master-Slave)

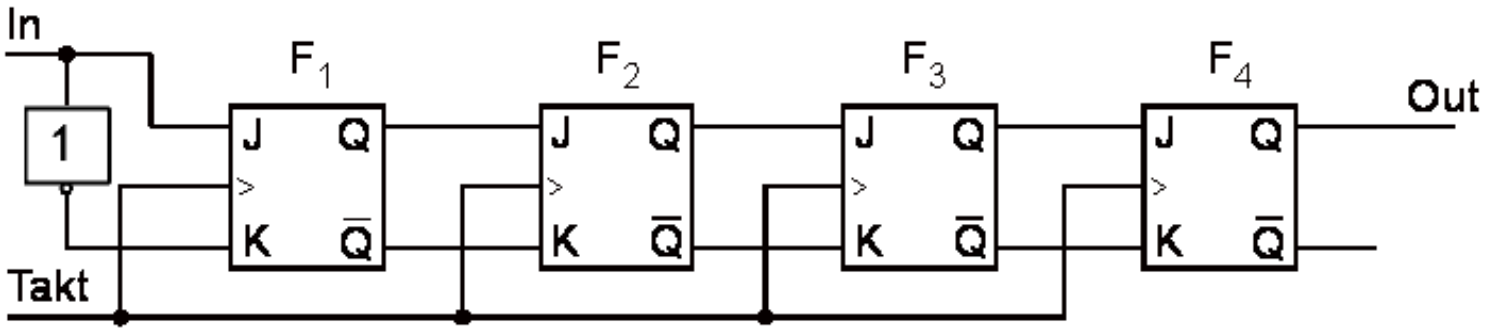
Sequenzielle Schaltungen (19)

Schieberegister (shift register):

Realisierung durch D Flip-Flops:



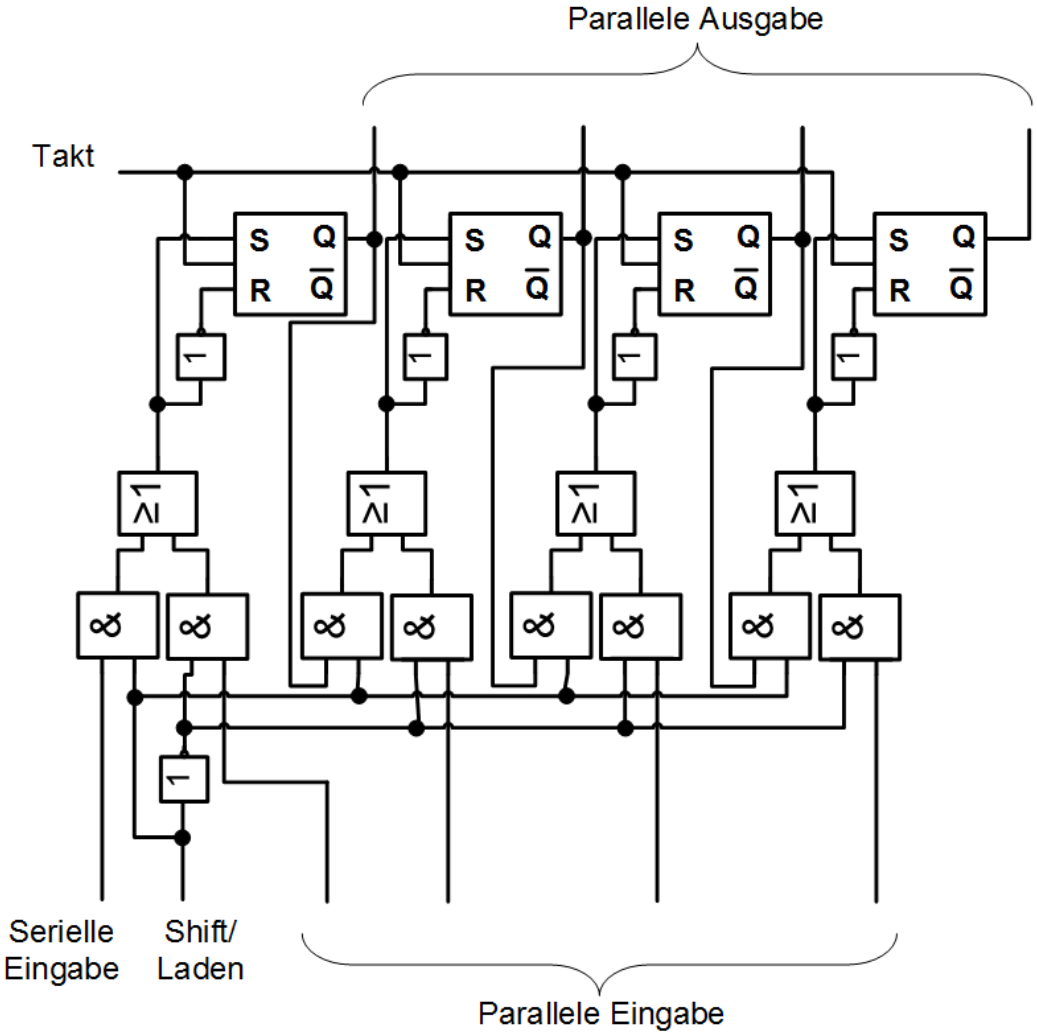
Realisierung durch JK Flip-Flops:



Sequenzielle Schaltungen (20)

Parallel/Schieberegister (parallel / shift register):

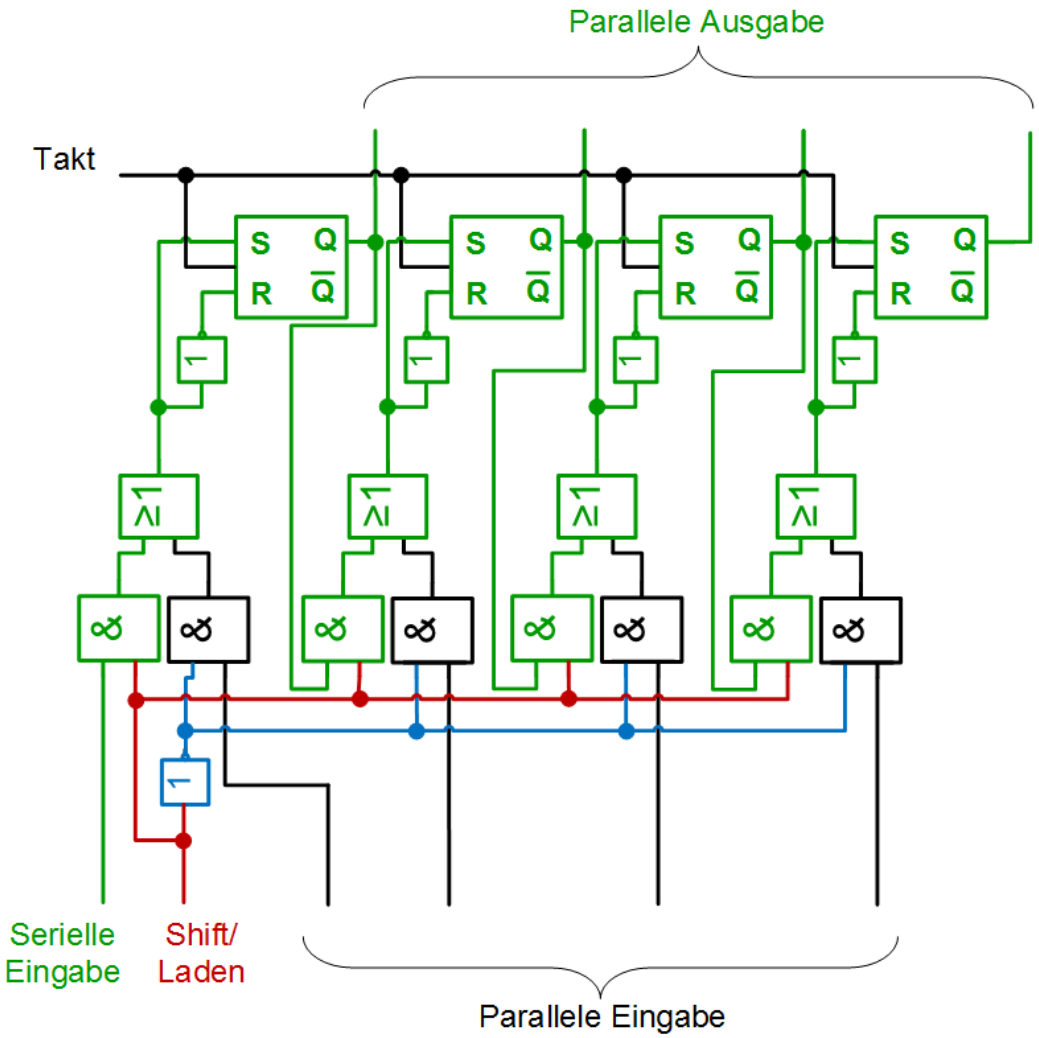
Beispiel mit RS Flip-Flop:



Sequenzielle Schaltungen (20a)

Parallel/Schieberegister (parallel / shift register):

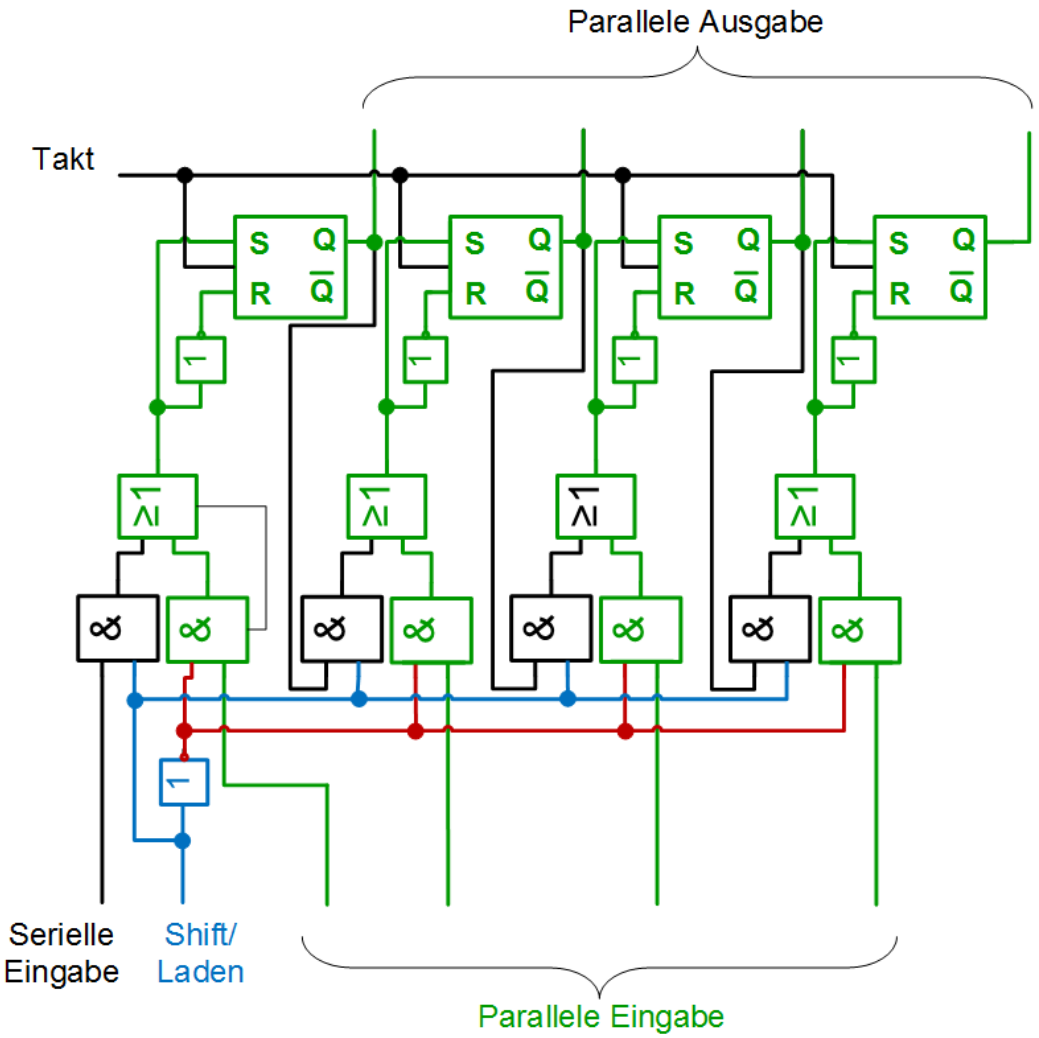
Beispiel mit RS Flip-Flop:



Sequenzielle Schaltungen (20b)

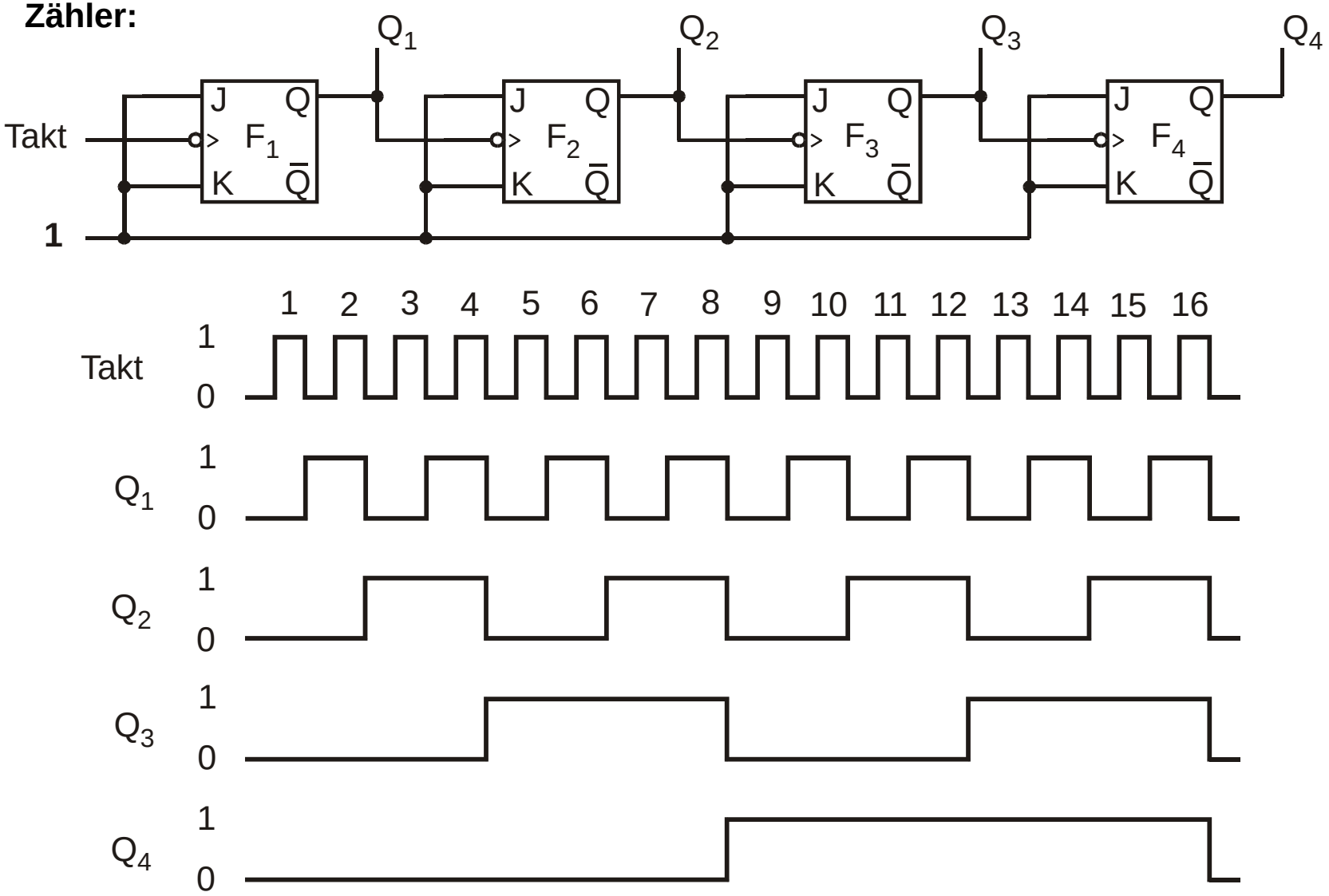
Parallel/Schieberegister (parallel / shift register):

Beispiel mit RS Flip-Flop:



Sequenzielle Schaltungen (21)

Zähler:

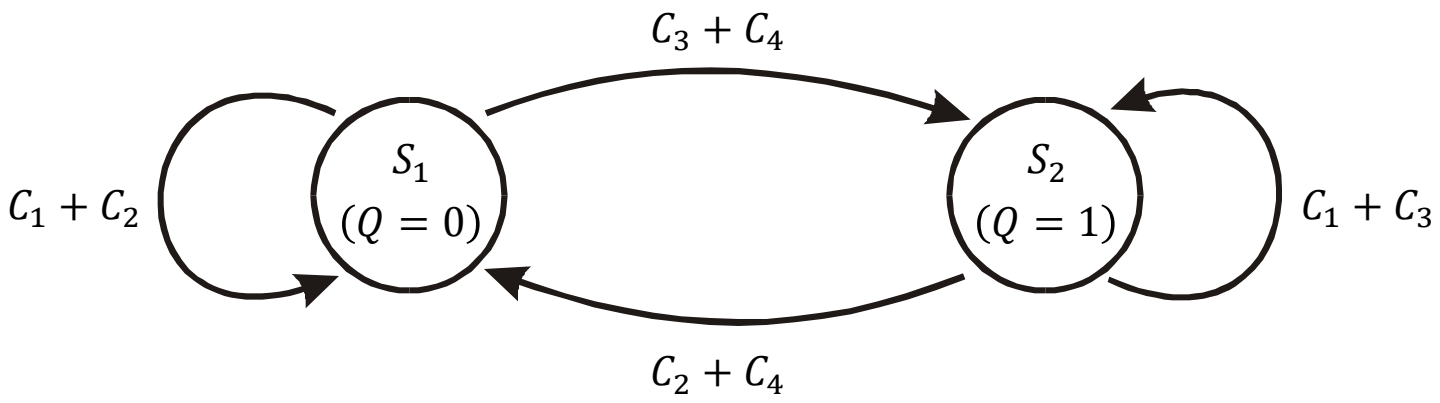


Sequenzielle Schaltungen (22)

Darstellung der Funktion von sequentiellen Schaltungen durch Zustandsdiagramme:

Beispiel JK Flip-Flop:

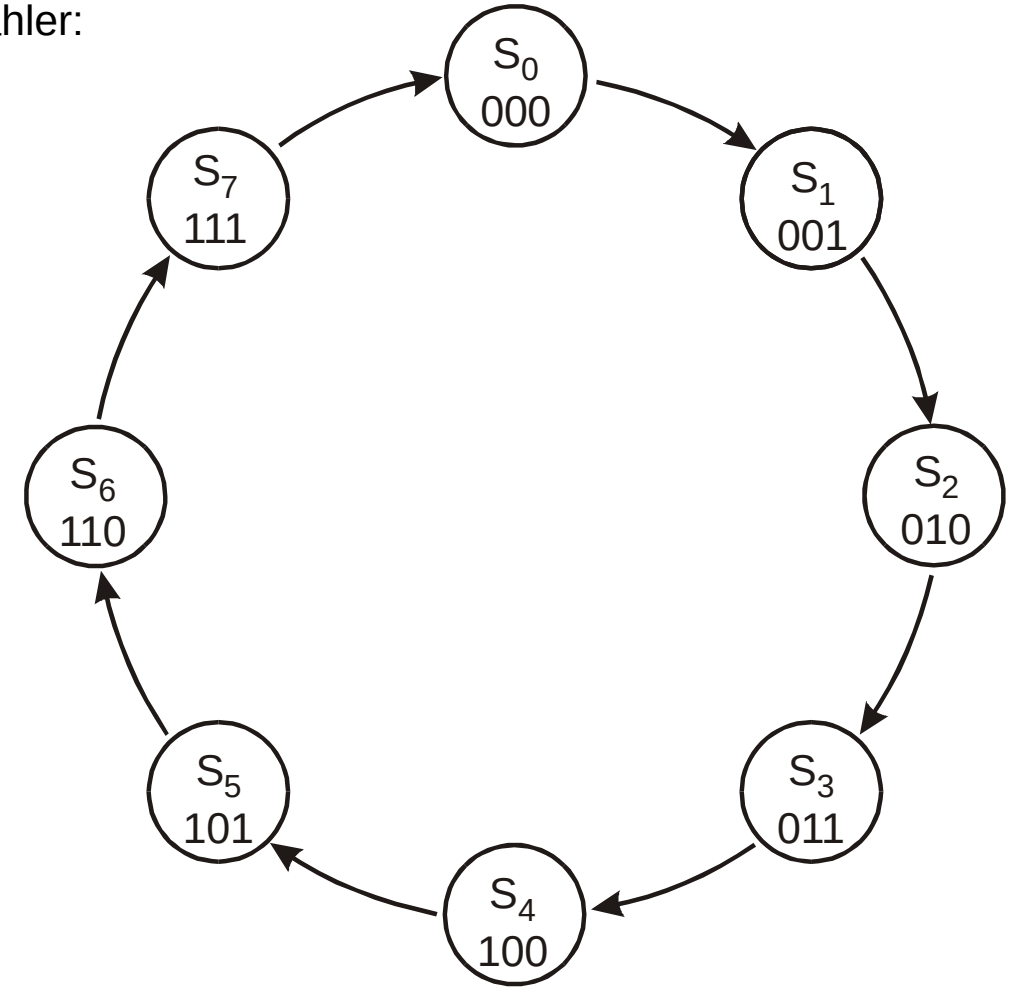
J	K	Q
0	0	C_1
0	1	C_2
1	0	C_3
1	1	C_4



Sequenzielle Schaltungen (23)

Darstellung der Funktion von sequentiellen Schaltungen durch Zustandsdiagramme:

Beispiel 3-Bit Zähler:



Sequenzielle Schaltungen (24)

Beschreibung von sequentiellen Schaltungen mit Hilfe endlicher Automaten

Endlicher Automat

Die Mengen X (Eingabevektor), Y (Ausgabevektor) und Z (Zustandsvektor) sind endlich und nicht leer. Des weiteren gibt es 2 Funktionen f und g, die die Ausgabe- und die Überführungsfunktion beschreiben

Ausgabefunktion $$Y(t_n) = f\big(X(t_n), Z(t_n)\big)$$

Überführungsfunktion $$Z(t_{n+1}) = g\big(X(t_n), Z(t_n)\big)$$

Sequenzielle Schaltungen (25)

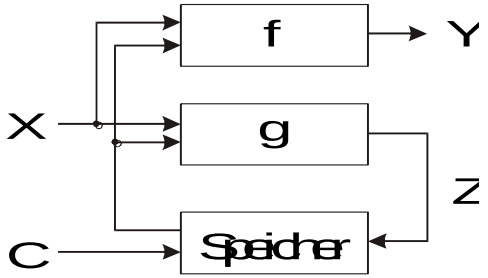
Beschreibung von sequentiellen Schaltungen mit Hilfe endlicher Automaten

Moore-Automat

$$Y = f\big(X, Z(t_n)\big)$$

$$Z(t_{n+1}) = g\big(X, Z(t_n)\big)$$

Ausgabe erfolgt beim Übergang in den neuen Zustand, d.h. Ausgabe Ist gleichzeitig abhängig von der Eingabe.

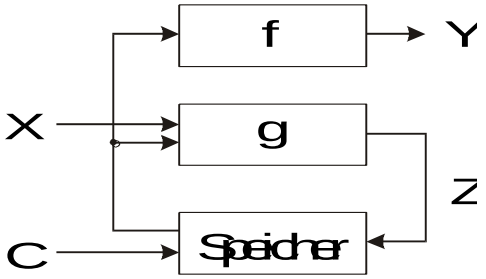


Nealy-Automat

$$Y = f\big(Z(t_n)\big)$$

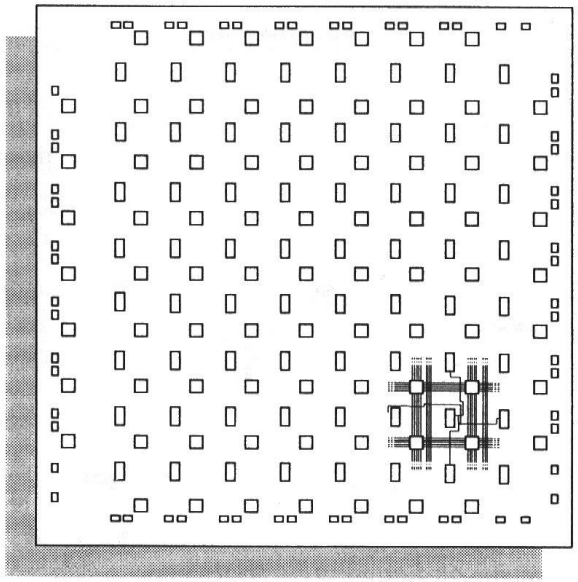
$$Z(t_{n+1}) = g\big(X, Z(t_n)\big)$$

Ausgabe hängt nur vom gegenwärtigen Zustand ab, nicht davon, wie dieser Zustand erreicht wurde. X wirkt erst nach Einem Takt auf den Ausgang.

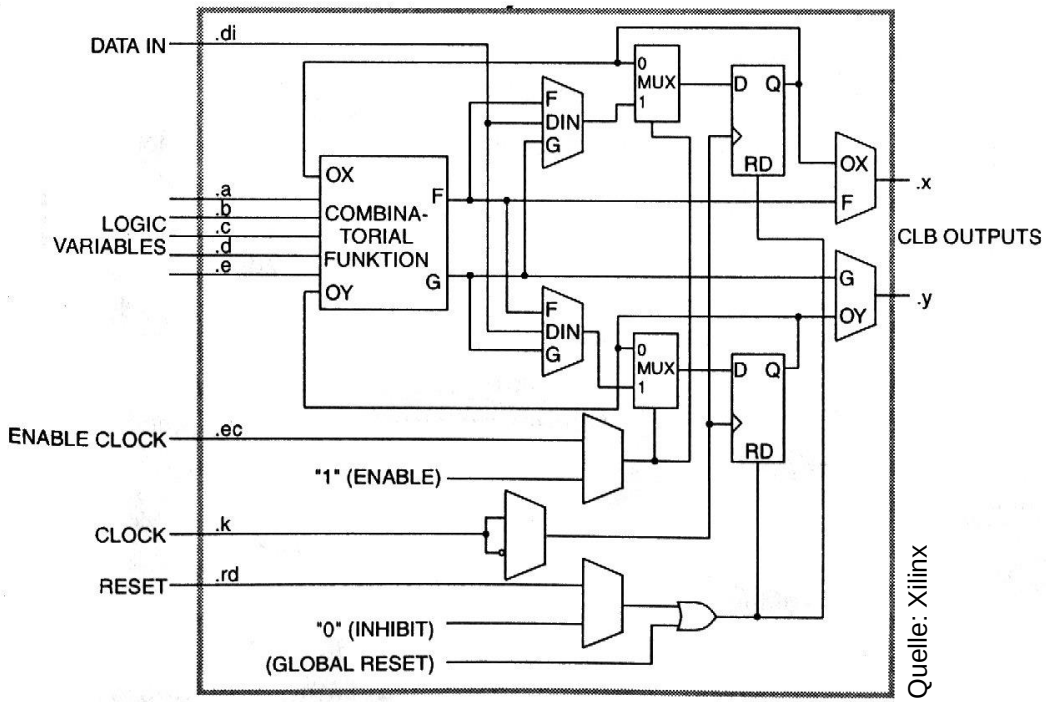


Sequenzielle Schaltungen (26)

Realisierung von sequentiellen Schaltungen mit Hilfe von FPGA



Matrix von CLB‘s



Quelle: Xilinx

Architektur des Xilinx FPGA XC4000

Combinatorial Logic Block (CLB)

Programmierung:

- Logikpläne,
- Endliche Zustandsautomaten (FSM)
- VHDL

Sequenzielle Schaltungen (27)

Test von digitalen Schaltkreisen/Schaltungen

Warum sind Tests notwendig?

- Herstellungsfehler
  - zu viel oder zu wenig Material (z.B. Ätzen Dotieren)
  - mechanische Beschädigungen
  - Justierung von Masken
- Fertigungstoleranzen
  - Streuung des Dotierungsprozesses
  - Fehler beim Ätzen (Überätzen, Unterätzen)
- Statistische Fehler
- Menschliche Fehler

Ziel:

- Funktionssicherheit des Endprodukts Speicher
- hohe Ausbeute, gute Qualität

Sequenzielle Schaltungen (28)

Auftreten von Fehlern

- Permanente Fehler
  - Verbindungsfehler, Defekte, funktionale Entwurfsfehler

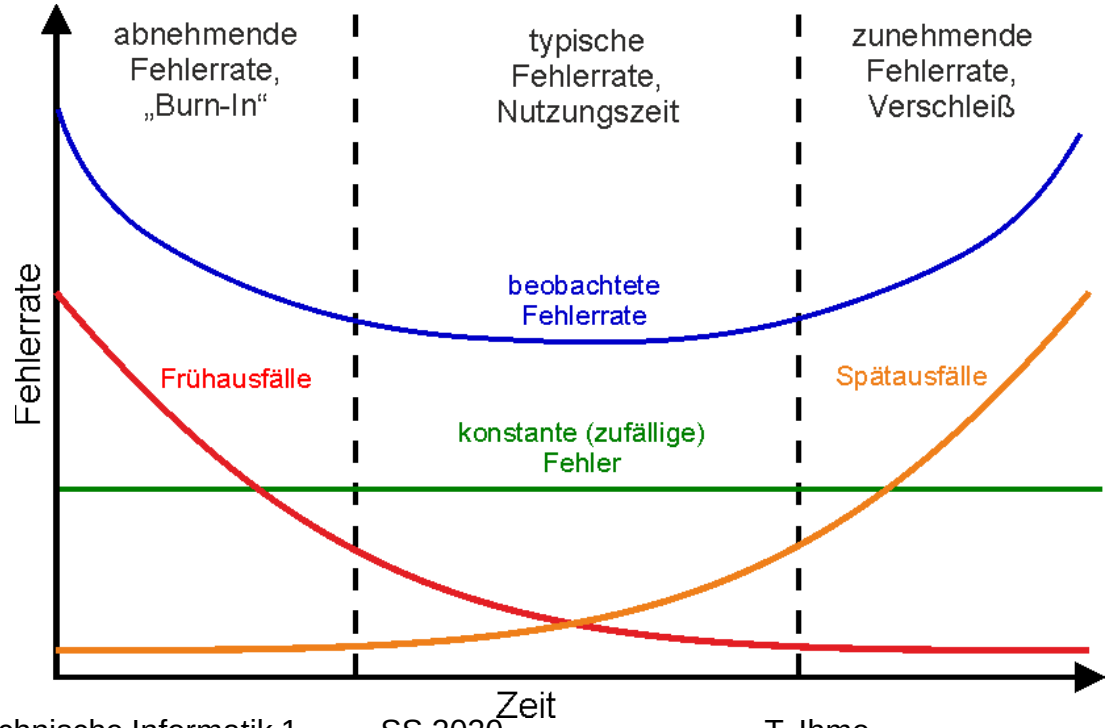
- Nicht-permanente Fehler
  - Strahlung, Alpha-Teilchen
  - Verschmutzung
  - Feuchtigkeit, Temperatur

- Intermittierende Fehler
  - Alterungsprozess
  - Verbindungsfehler
  - zeitkritischer Entwurf

Sequenzielle Schaltungen (29)

Auftreten von Fehlern

- Ausfallwahrscheinlichkeiten („Badewannen-Kurve“)
  - Frühausfälle, reduzierbar durch Burn-In
  - Normaler Betrieb
  - Anstieg am Ende der Lebensdauer durch Alterung



Sequenzielle Schaltungen (30)

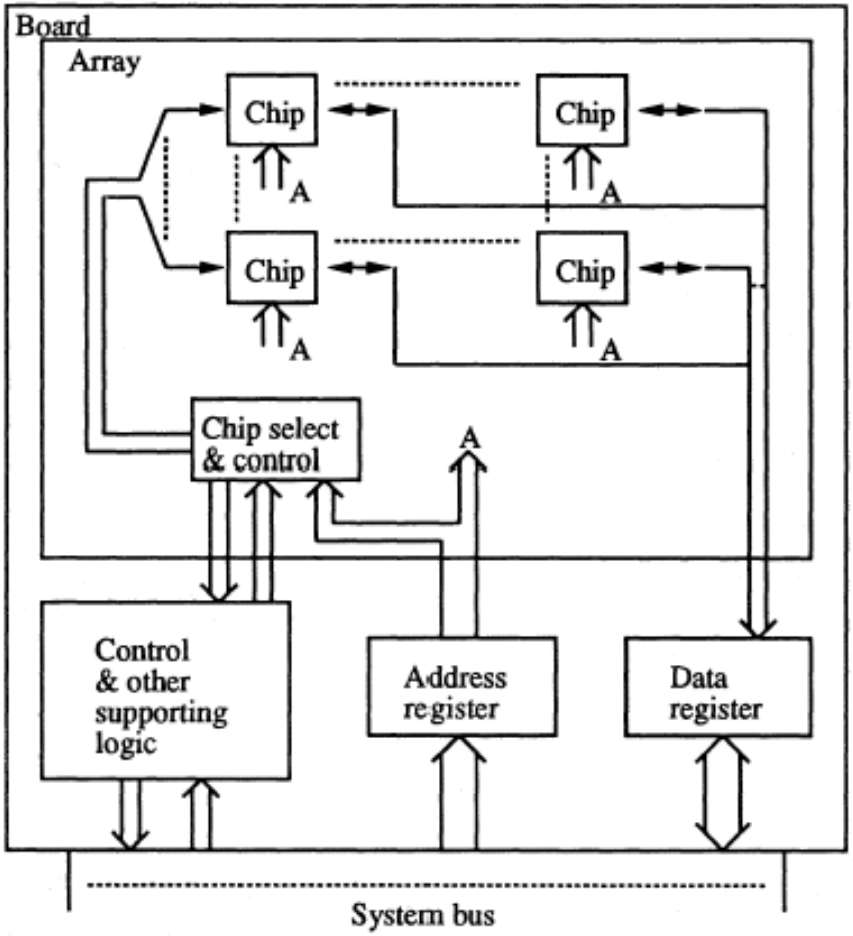
Auftreten von Fehlern

Hierarchien

- Chip
- Board
- System

Testverfahren

- Fehlermodell
- Genauigkeit der Beschreibung
- Datenmenge
- Testzeit



Sequenzielle Schaltungen (31)

Beispiel: Speichertest

Beispiel: Test eines 1kByte - Speichers

- Test 1: Schreiben *eines* Testmusters in jede der 2^{10} Speicherzellen und anschließend wieder herauslesen
  → 2^{10} Tests
- Test 2: Überprüfung *aller* Testmuster für jede der 2^{10} Speicherzellen
  → $2^{10} \cdot 2^{8} = 2^{18}$ Tests

Musterabhängiger Fehler (pattern sensitivity):

Schreiben eines Wortes in Speicherplatz X führt zu einem fehlerhaften Wort in Speicherplatz Y.

- Test 3: Überprüfung auf musterabhängige Fehler
  → $2^{18} \cdot 2^{10} = 2^{28}$ Tests

Sequenzielle Schaltungen (32)

Beispiel: Speichertest

```
Memtest-86 v3.0                  | Pass 10% ###
 Pentium III 449.2MHz            | Test 91% ##################################
 L1 Cache     32K  4403MB/s      | Test #3  [Moving inv, 8 bit pattern, cached]
 L2 Cache    512K   577MB/s      | Testing:   88K -  128M  128M
 Memory      128M   247MB/s      | Pattern:   fefefefe
 Chipset     i440[bz]x

  WallTime   Cached  RsvdMem   MemMap   Cache  ECC  Test  Pass  Errors ECC Errs
 ----------  ------  -------  --------  -----  ---  ----  ----  ------ --------
    0:03:36    128M      76K  e820-Std    on   off   Std     0       1        0
 ------------------------------------------------------------------------------
 Tst  Pass   Failing Address          Good       Bad     Err-Bits  Count Chan
 ---  ----  ----------------------  --------  --------  --------  ----- ----
   3     0  00001d6e1d8 -    29.8MB  fefefefe  fefefeee  00000010      1

 (ESC)exit  (c)configuration  (SP)scroll_lock  (CR)scroll_unlock
```

Beispiel eines Speichertests (Memtest-86 v3.0)
Dieser Fehler wurde durch den Power On Self Test (POST) nicht gefunden.

Sequenzielle Schaltungen (33)

Schlussfolgerung:

Vollständiger Test eines digitalen Systems mit externen Inputs und internen Zuständen ist effektiv nicht möglich

Defekt:

originäre, bei der Herstellung oder später eingetretener Fehlzustand in Hard- bzw. Software einer Komponente eines digitalen Systems

Fehler:

beobachteter Effekt auf Grund eines Defekts

Fehlermodell:

Auflistung der Fehler, die durch Tests gefunden werden sollen

Sequenzielle Schaltungen (34)

Fehlerverhalten

- Z.B. Verhalten so, als ob ständig ein Knoten ständig auf „0“ oder „1“ ist (der Knoten „klebt fest“) → „stuck-at“

Typische Störungen (Fehlermodell):

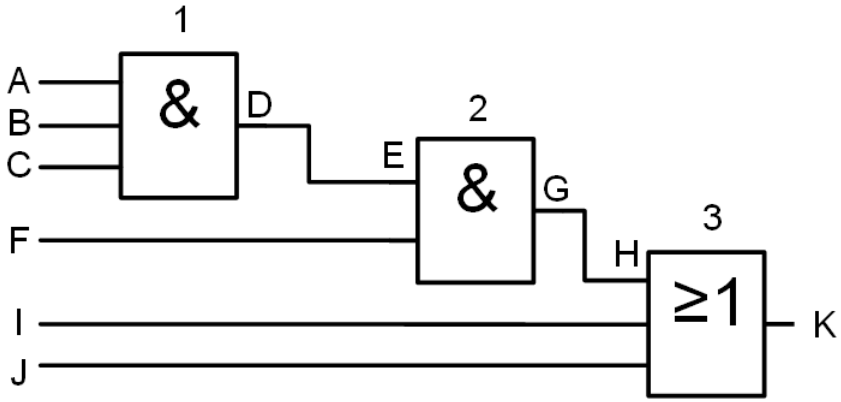
- stuck-at-one (s_a_1)
- stuck-at-zero (s_a_0)
- Überbrückungsfehler (bridging fault)

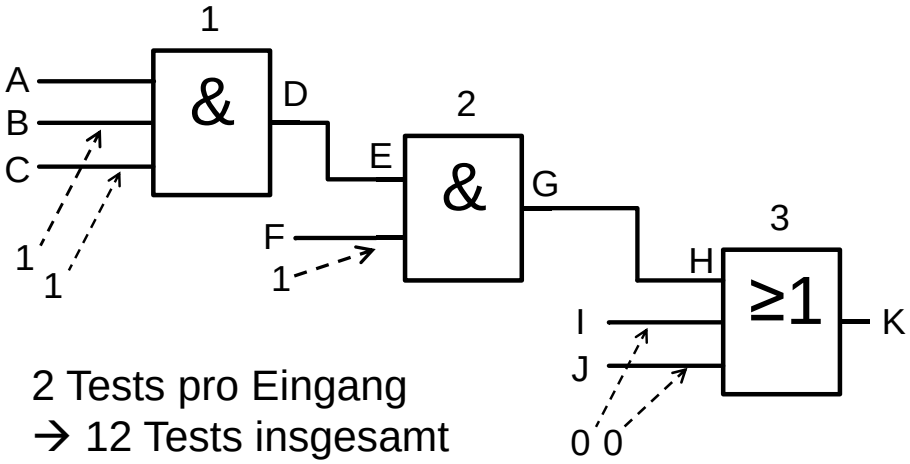
Test auf Stuck-at-Fehler

- Prinzip der Pfadsensitivierung
- Pfad führt über bestimmte Knoten
- Ausgangspegel hängt nur diesem Pfad und dem Eingang ab
- Alle Knoten müssen sich mit 0 und 1 belegen lassen

Sequenzielle Schaltungen (35)

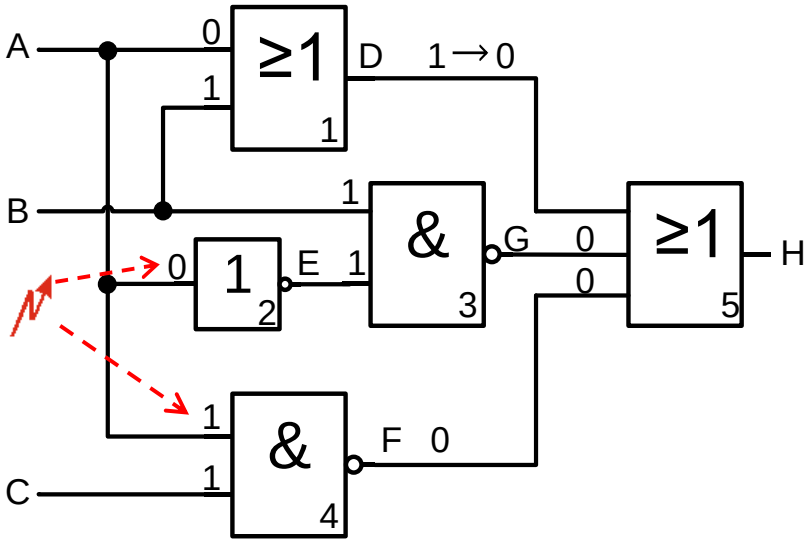
Beispiel einer Pfadsensitivierung (A-K):





2 Tests pro Eingang
→ 12 Tests insgesamt

Beispiel für einen nicht erkennbaren Fehler:



Test mit Pfadsensitivierung (A-H) nicht möglich, da gegensinnige Nebenbedingungen nicht realisierbar (nicht widerspruchsfrei)

Sequenzielle Schaltungen (36)

Zusammenfassung:

Wir haben bisher einige wichtige Schaltungen für den Entwurf eines digitalen Rechners eingeführt.

Die wichtigsten Parameter sind:

- Kosten und Geschwindigkeit
- Testbarkeit
- Zuverlässigkeit

Problem:

Die genannten Kriterien sind nicht widerspruchsfrei

Hauptaufgabe beim Entwurf:

Finden eines (die Applikation) zufriedenstellenden **Trade-offs**