

# Digitaltechnik

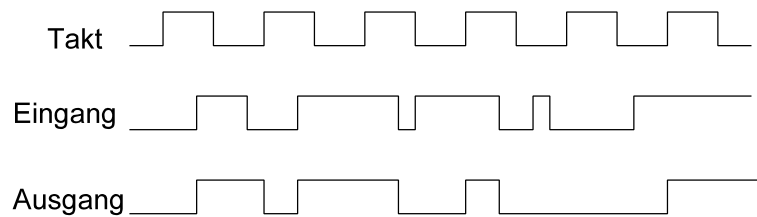
## Übungsaufgaben

### Sequentielle Schaltungen

– Mit Musterlösungen –

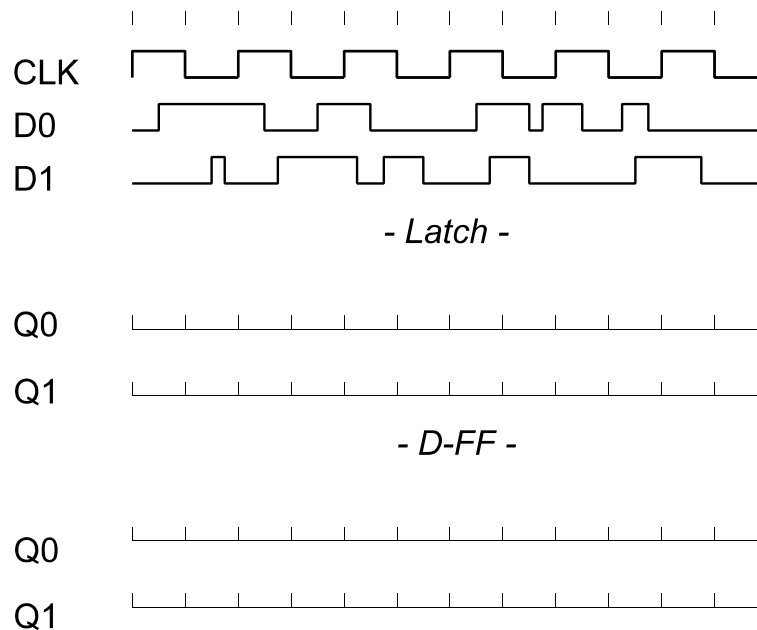
1. 2. 12

1. An einem Schaltkreis, der ein Register enthält, messen Sie die in Abb. 1 gezeigte Signalfolge. Handelt es sich dabei um ein D-Flipflop- oder um ein Latch-Register? (Kurze Begründung.)



**Abb. 1**

2. An den Eingängen eines 2-Bit-Registers liegen Impulse gemäß Abb. 2 an.
- Zeichnen Sie die ausgangsseitigen Impulsfolge ein, wenn es sich um ein Latch-Register handelt.
  - Zeichnen Sie die ausgangsseitigen Impulsfolge ein, wenn es sich um ein D-Flipflop-Register handelt.



**Abb. 2**

3. Wir bearbeiten nochmals die Aufgabe mit den Lüftern (Abb. 3). Die kombinatorischen Verknüpfungen wurden bereits entworfen; die ursprünglichen Fehleranzeigesignale für die grüne und die rote LED sind also gegeben. Jetzt soll die Fehleranzeige solange gehalten werden (auch dann, wenn der jeweilige Fehler zeitweise wieder verschwindet (Aussetzer)), bis der Servicetechniker einen entsprechenden Schalter betätigt (Kontakt SRVCR). Der Kontakt wirkt aktiv Low.

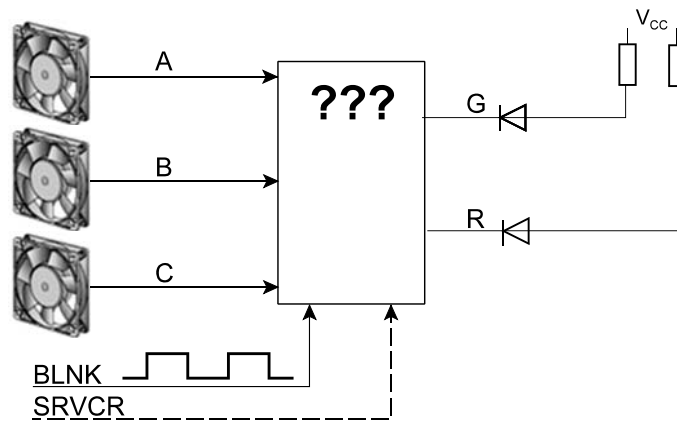


Abb. 3

4. An den Eingängen eines JK-Flipflop messen Sie den Signalverlauf gemäß Abb. 4. Was erwarten Sie am Ausgang? Es handelt sich um ein flankengesteuertes JK-Flipflop, das auf die Taktrückflanke (High-Low) schaltet (z. B. um den Typ 74x112).

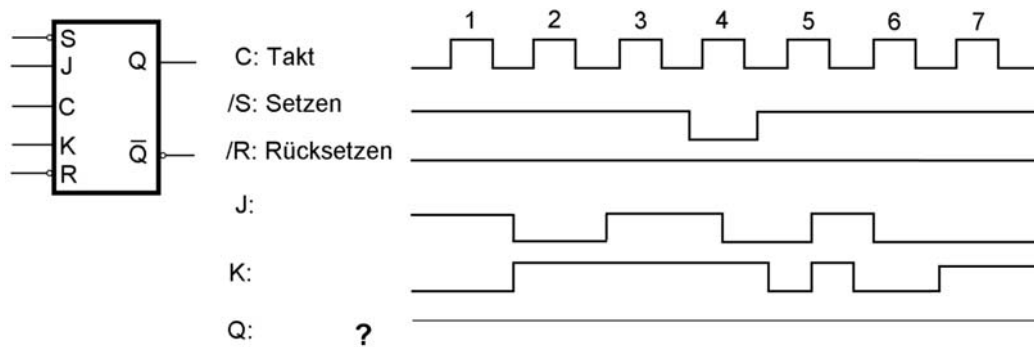


Abb. 4

5. Entwerfen Sie eine Zusatzbeschaltung, die ein T-Flipflop in ein JK-Flipflop umwandelt (Abb. 5).

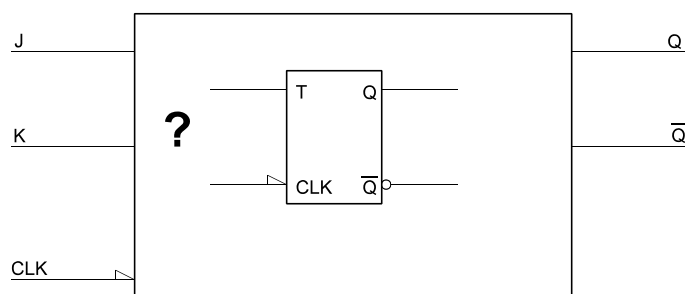


Abb. 5

6. Einem Mikrocontroller soll ein Fehler-Flipflop vorgeschaltet werden (Abb. 6. Dieses soll durch ein impulsförmiges Fehlersignal ERROR gesetzt und durch ein programmseitig schaltbares Signal ERROR\_RESET gelöscht werden. Flipfloptyp: D-Flipflop 7474. Geben Sie zwei Schaltungen an, die ein jeweils anderes Schaltverhalten verwirklichen:
- Das Flipflop soll gesetzt bleiben, falls während des programmseitigen Löschens (also bei aktivem ERROR\_RESET).
  - Das Flipflop soll durch Erregen von ERROR\_RESET unter allen Umständen gelöscht werden, auch wenn währenddessen das ERROR-Signal erregt wird.

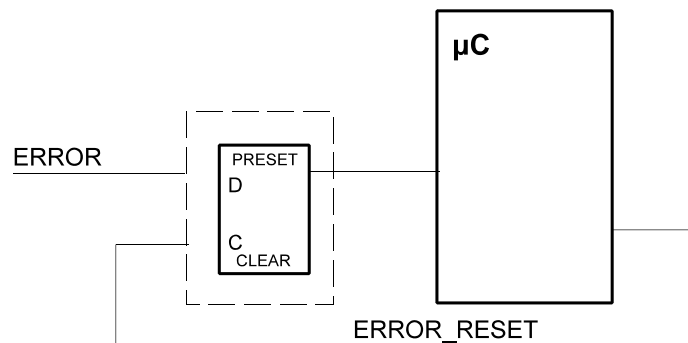


Abb. 6

7. Denksportaufgabe: Abb. 7 zeigt ein JK-Flipflop vom Typ 74x109. Können Sie es sich denken, weshalb man den K-Eingang invertiert ausgelegt hat?

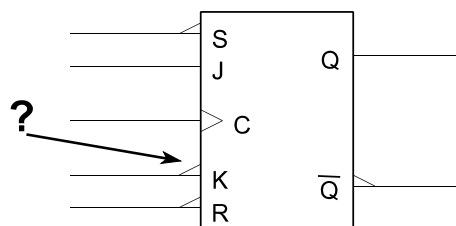


Abb. 7

8. Abb. zeigt einen Schaltungsausschnitt. Wird diese Schaltung immer zuverlässig arbeiten? Erläutern Sie kurz, welches Problem Sie sehen und schlagen Sie ggf. eine Änderung vor.

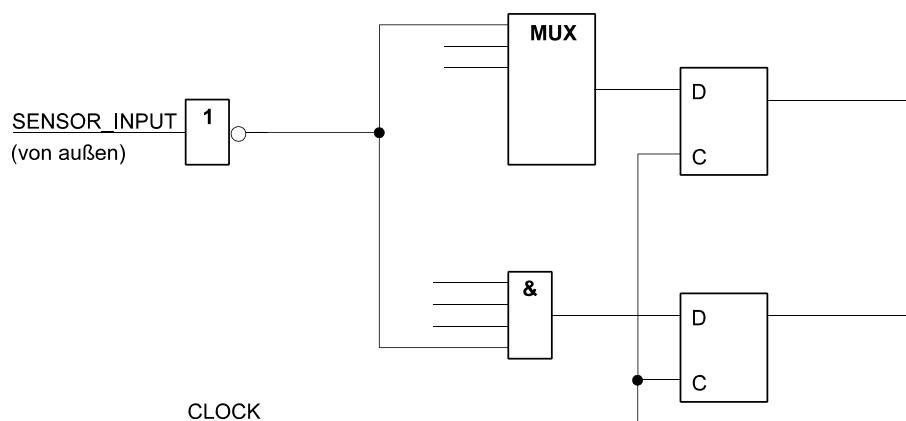


Abb. 8

9. Entwerfen Sie ein vollsynchrones 4-Bit-Register mit den Funktionen gemäß der folgenden Tabelle (Abb. 9). Grundlage: D-Flipflops sowie beliebige Gatter. Es genügt, eine Bitposition sowie ggf. erforderliche zentrale Schaltmittel darzustellen. Vorrangregeln (wenn zwei oder mehr Steuersignale gleichzeitig aktiv sind):

- CLR dominiert über alle anderen Funktionen.
- LD dominiert über INV.

Signal	Funktion
LD	Laden
INV	Invertieren des Inhaltes (aus 0 wird 1, aus 1 wird 0)
CLR	Löschen (alle Stellen = 0)
–	Datenbelegung halten

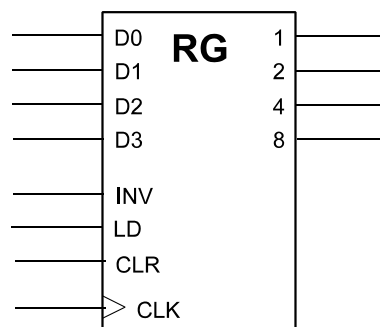


Abb. 9

10. Entwerfen Sie ein vollsynchrones 4-Bit-Register mit den Funktionen gemäß der folgenden Tabelle (Abb. 10). Grundlage: D-Flipflops sowie beliebige Gatter. Es genügt, eine Bitposition sowie ggf. erforderliche zentrale Schaltmittel darzustellen. Vorrangregeln: keine.

Signal	Funktion
LD	Laden
LDM	Laden maskiert. Nur die Bitpositionen laden, deren Maskenbit mit 1 belegt ist. Alle anderen so lassen, wie sie sind
TGM	Umschalten maskiert (Toggle). Nur die Bitpositionen ändern (von 0 nach 1 oder von 1 nach 0), deren Maskenbit mit 1 belegt ist. Alle anderen so lassen, wie sie sind
CLR	Löschen (alle Stellen = 0)
–	Datenbelegung halten

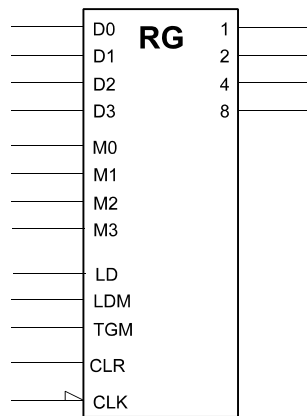


Abb. 10

11. Entwerfen Sie einen Zustandsautomaten (State Machine), der das folgende Zustandsdiagramm durchläuft (Abb. 11). Anfangszustand: Z1. Eingangssignale: T und W. Codierung: OHE. Flipfloptyp: D-FF. Um das Anfangsrücksetzen müssen Sie sich nicht kümmern.

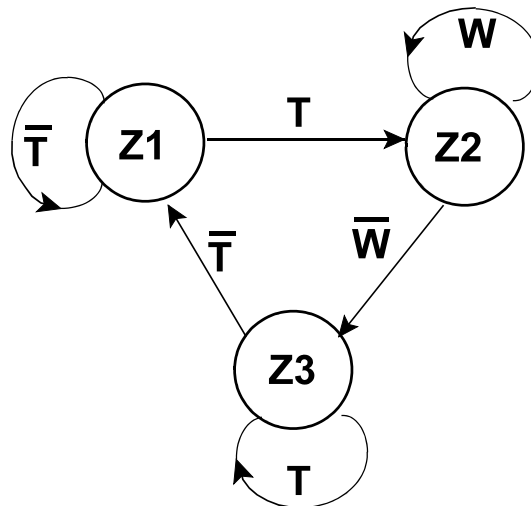


Abb. 11

12. Welches Impulsdiagramm ergibt sich aus der folgenden Schaltung (Abb. 12)? Es genügt, einen Schieberegister-Umlauf darzustellen.

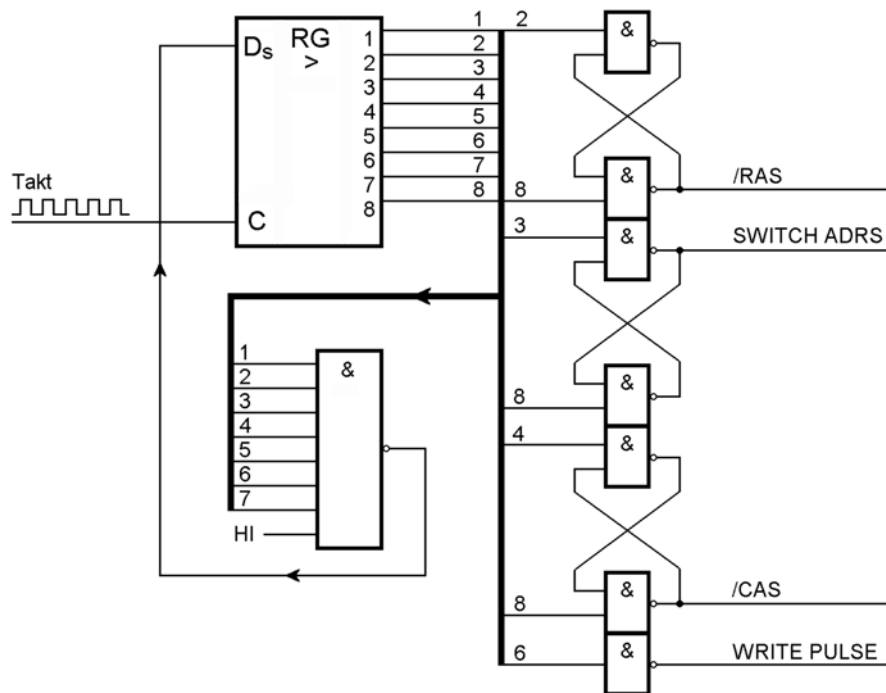


Abb. 12

13. Entwerfen Sie einen Johnsonzähler (mit D-Flipflops), der modulo 6 zählt (Abb. 13) und erweitern Sie ihn so, daß er alle 6 Stellungen im 1-aus-n-Code ausgibt (das ist die Grundlage für den elektronischen Würfel der ersten Übungen (s. dort Aufgabe 3)). Es genügt, die Schaltgleichungen der Decodierung anzugeben.

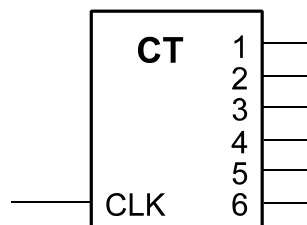


Abb. 13

14. Entwerfen Sie einen Kabeltester. Er soll Kabel mit maximal acht Einzelleitungen prüfen können (vgl. die übliche Netzwerkkabel). Ein solcher Tester (Abb. 14) besteht aus einem Stimulusgenerator, einer Anzeigeeinheit und einem Vergleicher. Der Stimulusgenerator wirkt so, daß er zunächst eine 1 auf die erste Leitung gibt, dann eine 1 auf die zweite Leitung usw.
- Entwerfen Sie einen Stimulusgenerator, der ein derartiges Prüfmuster zyklisch abgibt. Bauelemente: Flipflops, Register, Gatter.
  - Entwerfen Sie einen Vergleicher, der kontrolliert, ob das Prüfmuster am anderen Ende tatsächlich ankommt. Er soll eine Gesamt-Fehleranzeige ERROR erregen (ERROR = 0, wenn alles o.k., ansonsten = 1).

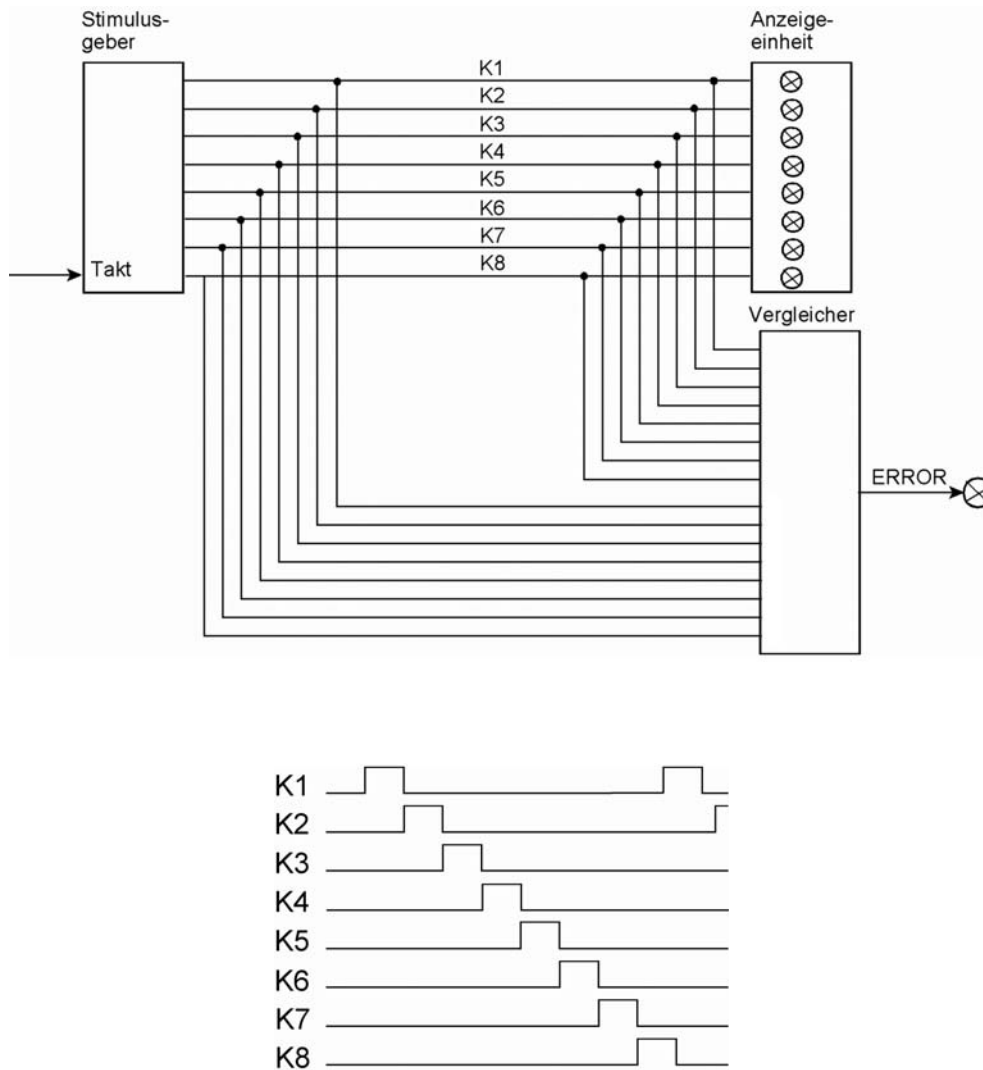


Abb. 14

15. Entwerfen Sie eine Zählerschaltung mit drei T-Flipflops C, B, A, die gemäß der folgenden Tabelle zyklisch zählt (von Stellung 6 wieder nach Stellung 1). Beim Einschalttrücksetzen wird die Stellung 1 eingestellt (darin müssen Sie sich nicht kümmern). Es genügt die Angabe der Schaltgleichungen ohne Minimierung.

Stellung	C	B	A
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1

## Musterlösungen

### Lösung 1

Es kommt darauf an, wie sich der Ausgang verhält, wenn der Takt aktiv (= High) ist. Beim Latch folgt der Ausgang dem Eingang auch, beim D-Flipflop entspricht er der Eingangsbelegung zur Zeit der Low-High-Flanke des Taktsignals, ändert sich also später nicht. Hier ist zu erkennen, daß der Ausgang dem Eingang nachfolgt (Abb. 1). Also ein Latch.

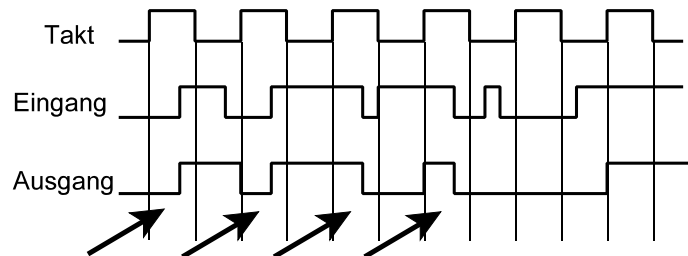


Abb. 1

### Lösung 2

Beim Latch ist der Signalverlauf an den Dateneingängen bei aktivem Takt (High) 1:1 zu übernehmen. Ist der Takt Low, so hält der Ausgang die Eingangsbelegung zur Zeit der High-Low-Flanke des Taktsignals. Beim D-Flipflop sind nur die Low-High-Flanken des Taktsignals von Bedeutung. Dort die Eingangsbelegung abnehmen und bis zur nächsten Low-High-Flanke des Taktes halten (Abb. 2).

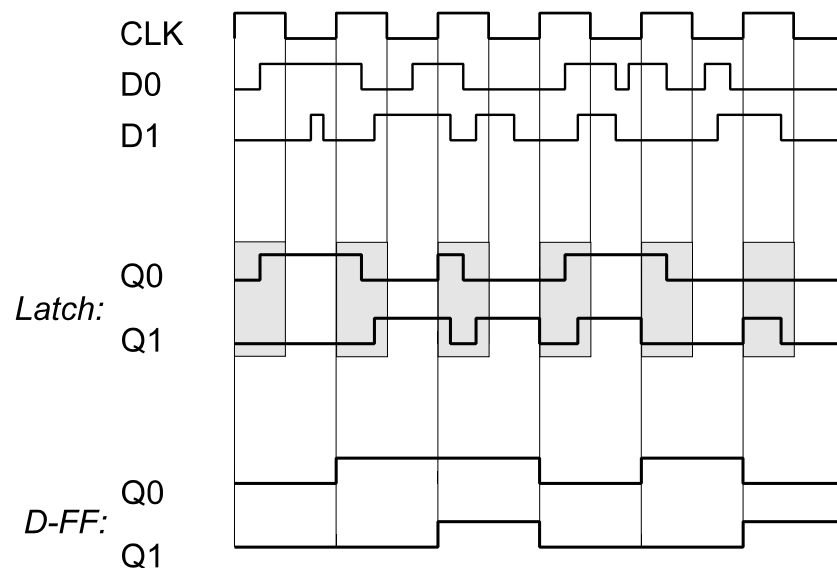


Abb. 2

### Lösung 3

Die Fehlersignale müssen in RS-Latches gehalten werden (Abb. 3). Bei der Anzeige muß das Blinken Vorrang haben.

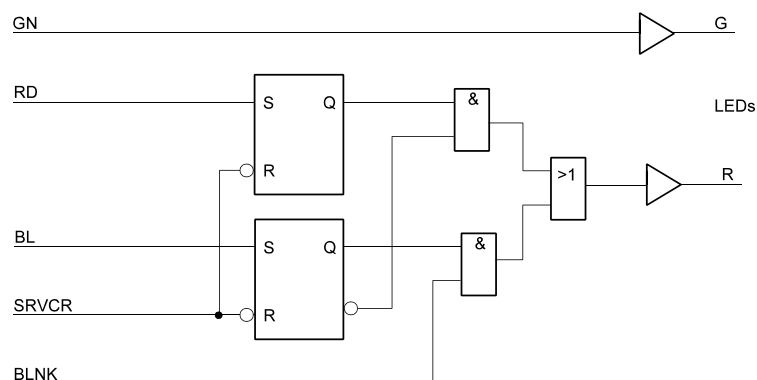
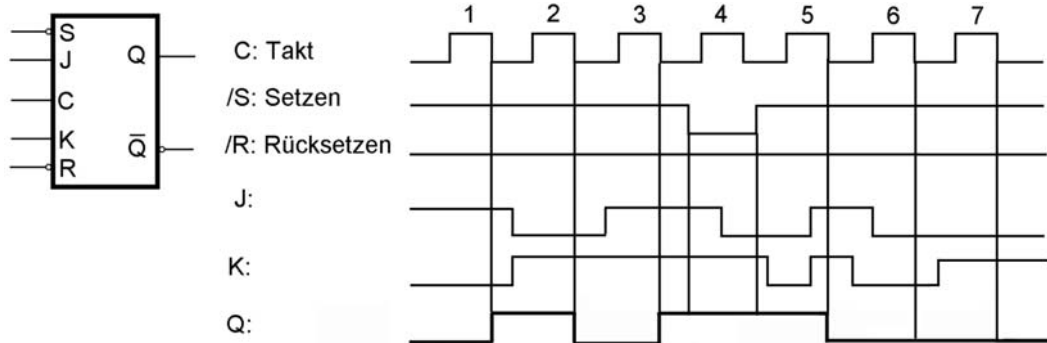


Abb. 3



**Lösung 4**

Gehen wir die Takte einzeln durch (Abb. 4). Wichtig sind die High-Low-Flanken.

**Abb. 4**

- 1)  $J = 1, K = 0$ , also  $Q = 1$ .
- 2)  $J = 0, K = 1$ , also  $Q = 0$ .
- 3)  $J = 1, K = 1$ , also ändert sich  $Q$ . Es ergibt sich  $Q = 1$ .
- 4) Setzen (S) ist aktiv und dominiert über die Taktauswertung.  $Q = 1$ .
- 5)  $J = 1, K = 1$ , also ändert sich  $Q$ . Es ergibt sich  $Q = 0$ .
- 6)  $J = 0, K = 0$ , also tut sich nichts;  $Q$  bleibt = 0.
- 7)  $J = 0, K = 1$ , also  $Q = 0$ .

**Lösung 5**

Um die JK-Funktion herbeizuführen, steht uns nur die T-Funktion zur Verfügung. Bei  $T = 1$  ändert sich die Belegung von  $Q$  mit dem nächsten Takt, bei  $T = 0$  wird sie gehalten. Betrachten wir die Automatentabelle eines JK-Flipflops:

J	K	Q	Q <sup>1</sup>	Änderungen zwischen Q und Q <sup>1</sup> (= T-Funktion)
0	0	0	0	-
0	0	1	1	-
0	1	0	0	-
0	1	1	0	1
1	0	0	1	1
1	0	1	1	-
1	1	0	1	1
1	1	1	0	1

Eine Q-Änderung (= T-Funktion) ist also bei folgenden Belegungen erforderlich:

J	K	Q
0	1	1
1	0	0
1	1	0
1	1	1

Die ursprüngliche Schaltfunktion (aus der Wahrheitstabelle bzw. Belegungsliste abzulesen):

$$T = \bar{J}KQ \vee \bar{J}\bar{K}\bar{Q} \vee JK\bar{Q} \vee JKQ$$

Ersichtlicherweise (z. B. Vorgehen mittels Quine-McCluskey, 1. Schritt) kann man die erste mit der vierten und die zweite mit der dritten Belegung zusammenfassen (mittels KV-Diagramm kommt man auf das gleiche Ergebnis). Abb. 5 zeigt die zugehörige Schaltung.

$$T = J\bar{Q} \vee KQ$$

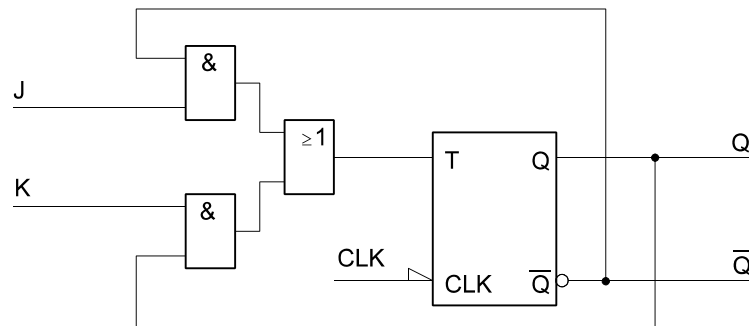


Abb. 5

### Lösung 6

Wichtig ist, daß das Signal, das jeweils vorrangig (dominierend) wirken soll, an den PRESET- bzw. CLEAR-Eingang des Flipflops angeschlossen wird (Abb. &).

- das Flipflop soll gesetzt bleiben, falls während des programmseitigen Löschs (also bei aktivem CLEAR) ERROR aktiviert wird. D. h., ERROR soll über ERROR\_RESET dominieren. Also ERROR an den PRESET-Eingang; Rücksetzen durch Übernehmen einer Null mit ERROR\_RESET als Taktimpuls.
- das Flipflop soll durch Erregen von CLEAR unter allen Umständen gelöscht werden, auch wenn währenddessen das ERROR-Signal erregt wird. D. h., ERROR\_RESET soll über ERROR dominieren. Also ERROR\_RESET an den CLEAR-Eingang; Setzen durch Übernehmen einer Eins mit ERROR als Taktimpuls.

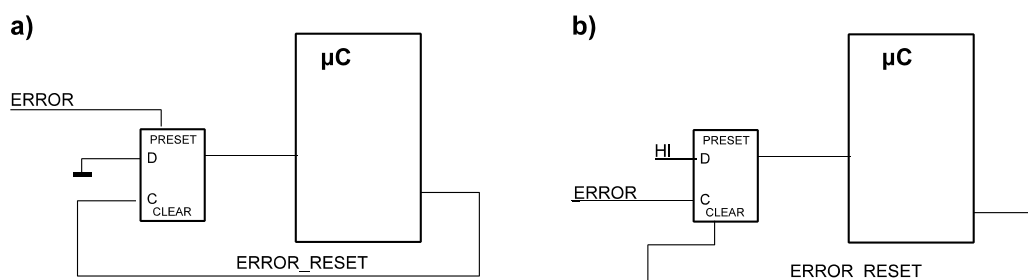


Abb. 6

### Lösung 7

Kann sein, daß für einen eingangsseitigen Negator kein Platz mehr war – oder daß man einen Design-Fehler als Feature verkauft. Es hat aber auch etwas für sich (Abb. 7): um ein D-Flipflop zu bauen, genügt es, J und  $\bar{K}$  einfach zusammenzuschalten.

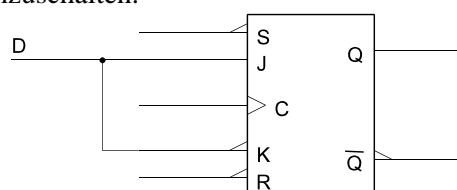


Abb. 7



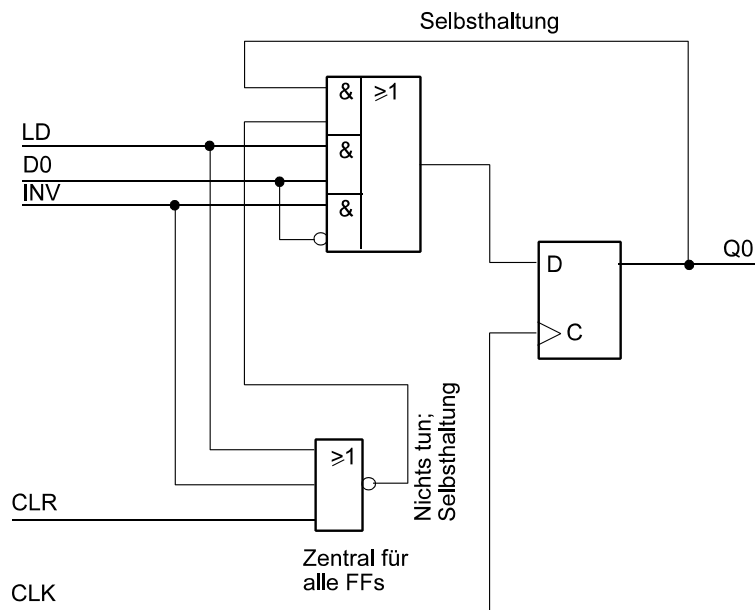


Abb. 9b

Nun sollen die Vorrangregeln berücksichtigt werden. Hierzu werden die UND-Verknüpfungen um die entsprechenden Inhibitionen erweitert; alle Signale, die inaktiv sein müssen, wenn die jeweilige Funktion ausgeführt werden soll, werden jeweils invertiert an die betreffenden UND-Gatter angeschlossen (Abb. 9c).

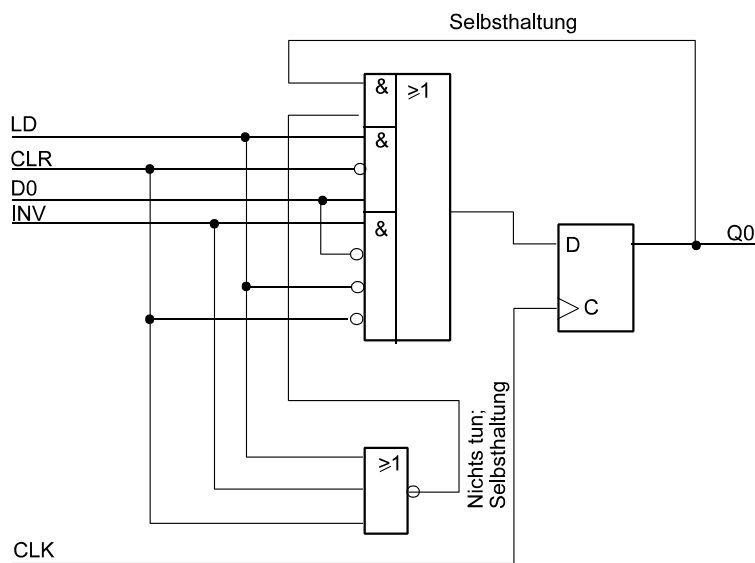


Abb. 9c

### Lösung 10

So kompliziert wird es in der Klausur nicht ... Infolge der Maskierung müssen die Flipflops für die jeweiligen Funktionen einzeln (selektiv) angesteuert werden (Abb. 10). Ist das Maskenbit inaktiv, ist die jeweilige Belegung zu halten (Rückführung).

1 - Selbsthaltung, wenn nichts zu tun; 2 - Laden; 3 - Laden maskiert bei gesetztem Maskenbit (Datenübernahme); 4 - Laden maskiert bei gelöschtem Maskenbit (Selbsthaltung); 5 - Umschalten maskiert bei gesetztem Maskenbit (invertierte Rückführung; Toggle-Funktion); 6 - Umschalten maskiert bei gelöschtem Maskenbit (Selbsthaltung).



Die Schaltgleichungen können aus dem Zustandsdiagramm abgelesen werden:

$$Z1_D = Z3 \cdot \bar{T} \vee Z1 \cdot \bar{T}$$

$$Z2_D = Z1 \cdot T \vee Z2 \cdot W$$

$$Z3_D = Z2 \cdot \bar{W} \vee Z3 \cdot T$$

### Lösung 12

Es handelt sich im Grunde um einen Ringzähler, in dem eine Null umläuft. Die Nullimpulse setzen und löschen die RS-Latches zu verschiedene Taktzeitpunkten (Abb. 12). Es ergibt sich ein Impulsmuster (RAS, SWITCH ARS, CAS, WRITE PULSE), das für die Ansteuerung asynchroner DRAMs typisch ist.

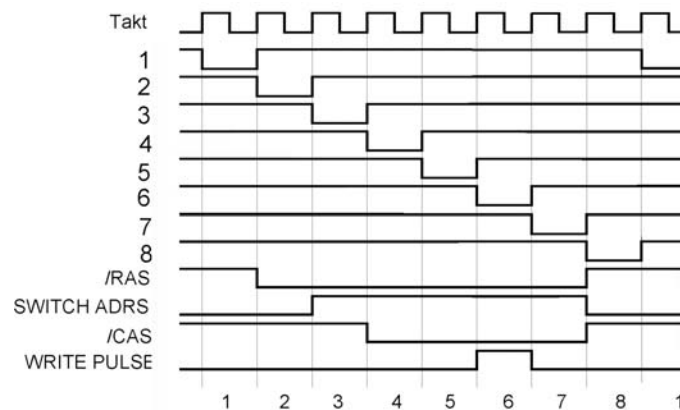


Abb. 12

### Lösung 13

Für einen Johnsonzähler modulo n brauchen wir  $n/2$  Flipflops (hier also drei Stück). Es ist im Grunde ein Schieberegister, dessen letzter Ausgang invertiert auf den Schiebendeingang zurückgeführt ist (Abb. 13). Die nachfolgende Tabelle veranschaulicht die Decodierung.

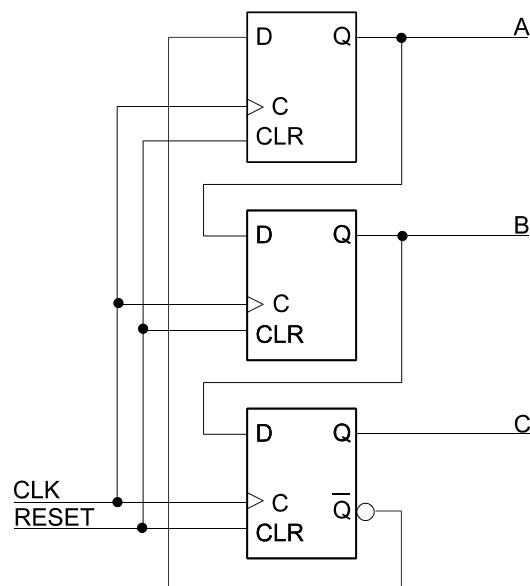


Abb. 13

	A	B	C	Decodierung 1 aus n
1	0	0	0	$\bar{A} \cdot \bar{C}$
2	1	0	0	$A \cdot \bar{B}$
3	1	1	0	$B \cdot \bar{C}$
4	1	1	1	$A \cdot \bar{C}$
5	0	1	1	$\bar{A} \cdot B$
6	0	0	1	$\bar{B} \cdot C$

**Lösung 14**

Die Prüffolge wird von einem Ringzähler erzeugt, der aus acht Flipflops besteht, die über ein NOR-Gatter rückgekoppelt sind (Impulsfolge ähnlich Abb. 12, nur andere Polung). Der Vergleicher besteht aus XOR-Gattern, die disjunktiv verknüpft sind (Abb. 14).

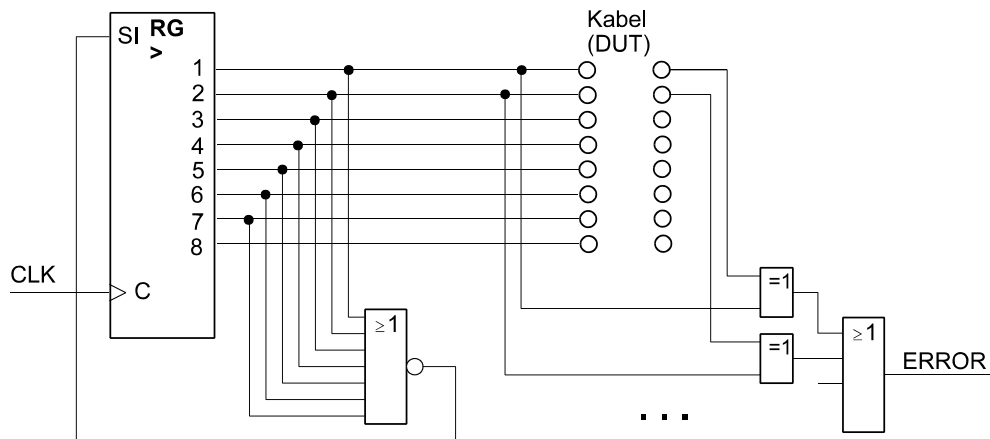


Abb. 14

**Lösung 15**

Prinzip: In jeder Stellung für jedes Flipflop festlegen, wie es mit dem jeweils nächsten Takt zu schalten hat. Die im jeweils nächsten Takt einzunehmende Belegung wird gemäß Zählrichtung aus der Zustandsfolgetabelle abgelesen (vorwärts = von oben nach unten, rückwärts = von unten nach oben). Es ist immer dann eine Eins an den T-Eingang anzulegen, wenn sich mit dem jeweils nächsten Takt die Belegung der Zählstelle ändern soll.

$$T(t) = Q(t) \oplus Q(t + 1)$$

Stellung	C	B	A	T <sub>C</sub>	T <sub>B</sub>	T <sub>A</sub>
1	0	0	1	0	1	0
2	0	1	1	0	0	1
3	0	1	0	1	0	0
4	1	1	0	0	0	1
5	1	1	1	0	1	0
6	1	0	1	1	0	0

Erläuterung am Beispiel der Stelle A:

Stellung	A	T <sub>A</sub>	
1	1	0	weil in Stellung 2 keine Änderung gegenüber Stellung 1
2	1	1	weil in Stellung 3 Änderung von 1 auf 0
3	0	0	weil in Stellung 4 keine Änderung gegenüber Stellung 3
4	0	0	weil in Stellung 5 Änderung von 0 auf 1
5	1	1	weil in Stellung 6 keine Änderung gegenüber Stellung 5
6	1	0	weil in Stellung 1 keine Änderung gegenüber Stellung 6

Die Schaltgleichungen:

$$T_A = \bar{C} \cdot B \cdot A \vee C \cdot B \cdot \bar{A}$$

$$T_B = \bar{C} \cdot \bar{B} \cdot A \vee C \cdot B \cdot A$$

$$T_C = \bar{C} \cdot B \cdot \bar{A} \vee C \cdot \bar{B} \cdot A$$