

Lösung 1

Es kommt darauf an, wie sich der Ausgang verhält, wenn der Takt aktiv (= High) ist. Beim Latch folgt der Ausgang dem Eingang an, 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.

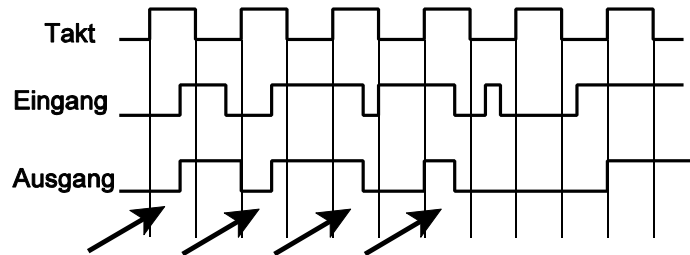


Abb. 1

Lösung 2

So geht es garantiert schief ... Das Problem ergibt sich dann, wenn eine Änderung des asynchronen Eingangssignals (SENSOR_INPUT) mit eine Low-High-Taktfanke zusammentrifft. Es kann dann sein, daß das eine Flipflop die neue und das andere die alte Signalbelegung übernimmt. Die weitere Schaltung würde dann mit zwei unterschiedlichen Werten desselben Signals arbeiten. Hinzu kommt die Gefahr metastabiler Zustände. Abhilfe: das Signal an einer einzigen Stelle synchronisieren (eine wichtige Regel beim Schaltungsentwurf!). Hierzu kann z. B. ein weiteres Flipflop vorgeschaltet werden (Abb. 2).

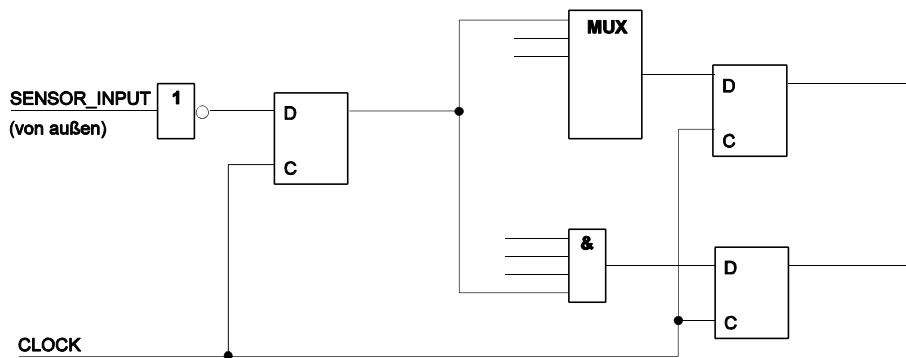


Abb. 2

Lösung 3

Gehen wir die Takte einzeln durch (Abb. 3). Wichtig sind die High-Low-Flanken.

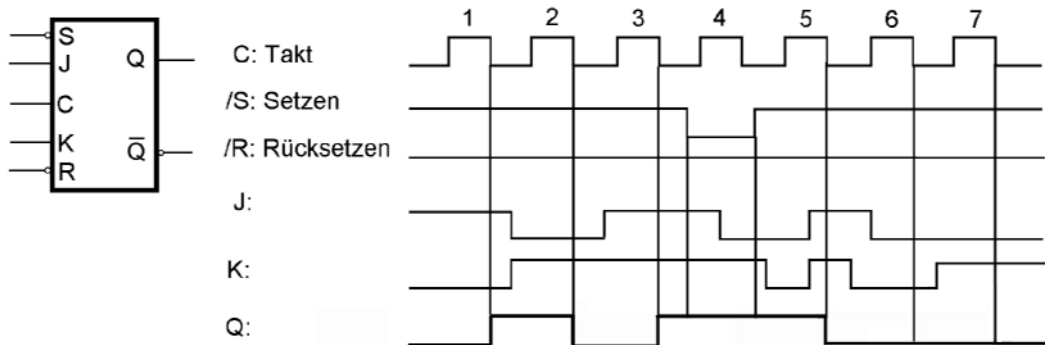


Abb. 3

- 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 4

Außer den angegebenen brauchen wir keine weiteren Bauelemente (Abb. 4). Mit dem Decoder 74x138 wird die Bitadresse decodiert. Jedem Flipflop ist ein 2-zu-1-Multiplexer vorgeschaltet, der vom angeschlossenen Decoderausgang zwischen Selbsthaltung und Übernahme der Dateneingangsbelegung umgeschaltet wird. (Daß der Decoder negierte Ausgänge hat, schadet nichts – man muß halt den MUX richtig herum anschließen...). Zur Ladeerlaubnissteuerung (Signal LD) wird die UND-Verknüpfung der Erlaubnissignale im 138 ausgenutzt.

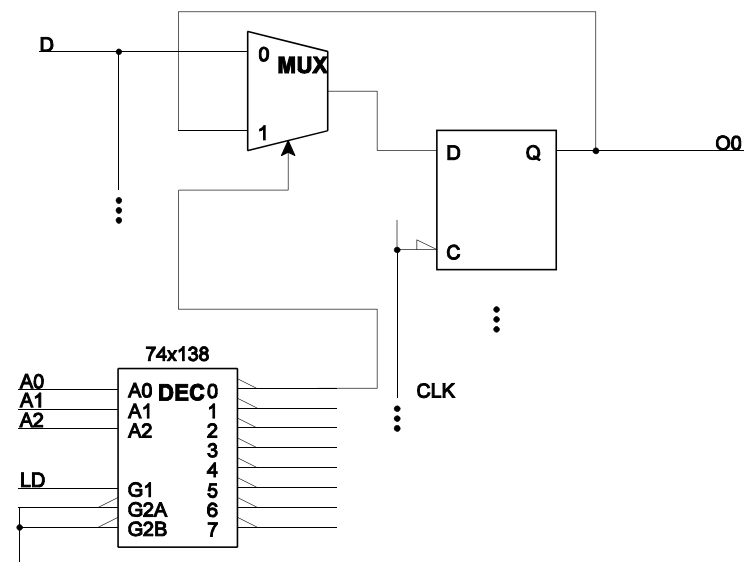


Abb. 4

Lösung 5

Die Beschaltung der D-Flipflops muß aus elementaren Gattern aufgebaut werden (Abb. 5). Die D-Flipflops sind zwischen folgenden Funktionen umzuschalten:

1. Halten, wenn nichts zu tun ist ($LOAD = 0$ und $DIAG = 0$). Diese Bedingung kann z. B.m it einem NOR-Gatter erkannt werden.
2. Datenübernahme, wenn $LOAD = 1$ und $DIAG = 0$. Diese Bedingung wird it einem UND-Gatter erkannt.
3. Diagnosefunktion (Wechseln oder Halten), wenn $DIAG = 1$. Halten (bei Dateneingang = 0) heißt Rückführung des Flipflop-Ausgangs, Wechseln (bei Dateneingang =1) heißt Rückführung des invertierten Flipflop-Ausgangs. Wir brauchen also eine Umsteuerung zwischen Durchreiche und Negation. Die naheliegende Lösung: eine Antivalenverknüpfung von Flipflopausgang und Dateneingang (XOR-Gatter).

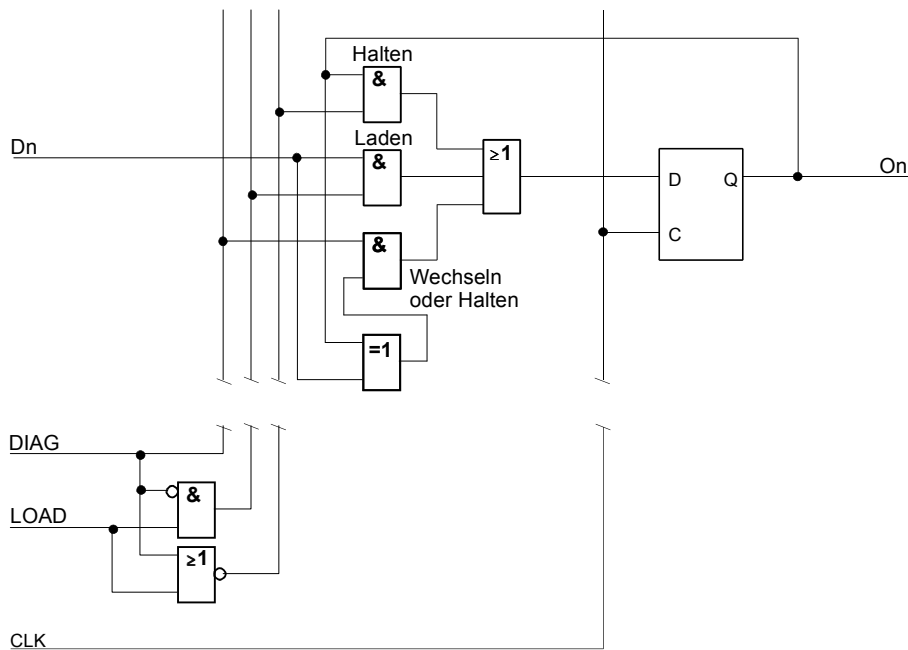


Abb. 5

Lösung 6

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'	Änderungen zwischen Q und Q' (= 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 Karnaugh-Plan kommt man auf das gleiche Ergebnis). Abb. 6 zeigt die zugehörige Schaltung.

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

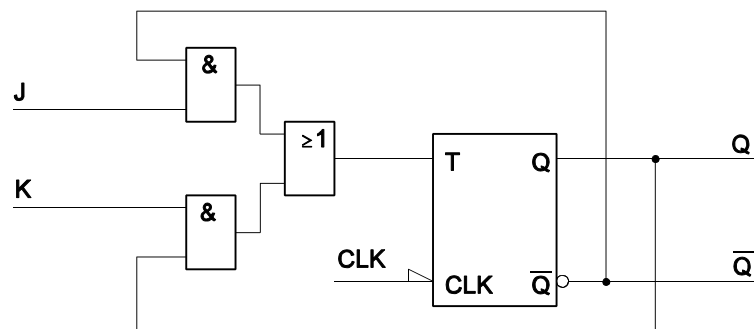


Abb. 6

Lösung 7

Beim asynchronen Binärzähler halbieren sich die Anforderungen an die Taktfrequenz mit jeder Stufe (1. Stufe: Eingangsfrequenz, 2. Stufe: 1/2 Eingangsfrequenz, 3. Stufe: 1/4 Eingangsfrequenz usw.).

Lösung 8

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. 7).

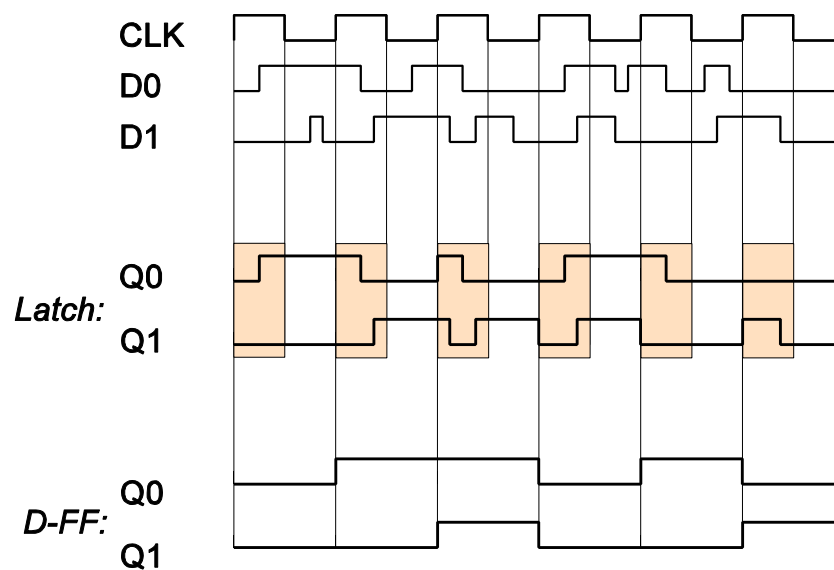


Abb. 7

Lösung 9

Die Tabelle in der Aufgabenstellung ist doch hinreichend suggestiv – oder? Man sieht sofort, daß ein 1-aus-8-Multiplexer je Bitposition genügt. Abb. 8 zeigt die Schaltung am Beispiel der Bitposition 3.

Steuersignale			Funktion	Schaltungslösung
CTL2	CTL1	CTL0		
0	0	0	Laden eines 16-Bit-Operanden	Dateneingang auswählen
0	0	1	Laden mit Festwert 0	Festwert 0 auswählen
0	1	0	Laden mit Festwert -1	Festwert 1 auswählen
0	1	1	Linksschieben	den Ausgang des jeweils benachbarten niederwertigen Flipflops auswählen
1	0	0	Rechtsschieben	den Ausgang des jeweils benachbarten höherwertigen Flipflops auswählen
1	0	1	Rechtsschieben arithmetisch	in Bitposition 15 die Vorzeichenbelegung halten (Rückführung); ansonsten den Ausgang des jeweils benachbarten höherwertigen Flipflops auswählen
1	1	0	Belegung wechseln	den invertierten Flipflop-Ausgang auswählen
1	1	1	nichts tun (halten)	den Flipflop-Ausgang auswählen (Rückführung)

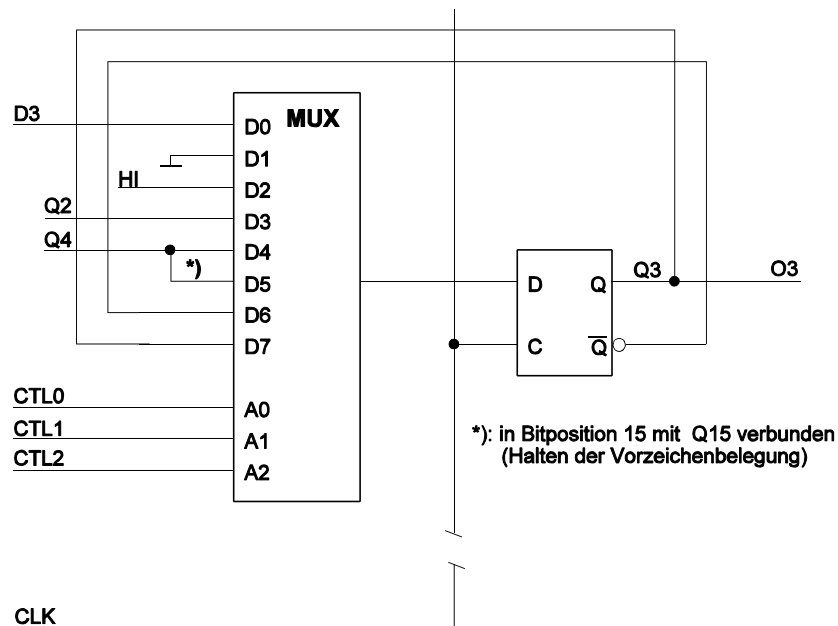


Abb. 8

Aufgabe 10

Wichtig ist, daß das Signal, das jeweils vorrangig (dominierend) wirken soll, an den PRSET- bzw. CLEAR-Eingang des Flipflops angeschlossen wird (Abb. 9).

- a) das Flipflop soll gesetzt bleiben, falls während des programmseitigen Löschsens (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.
- b) 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.

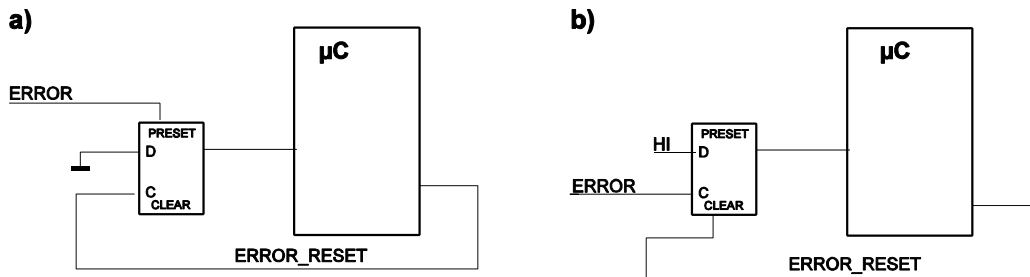


Abb. 9

Lösung 11

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. 10). Es ergibt sich ein Impulsmuster (RAS, SWITCH ARS, CAS, WRITE PULSE), das für die Ansteuerung asynchroner DRAMs typisch ist.

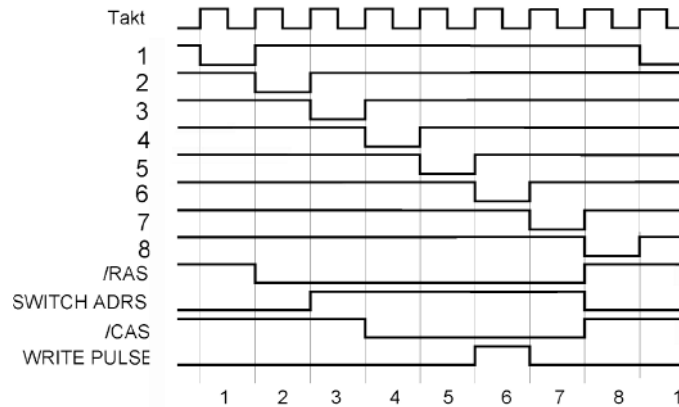


Abb. 10

Lösung 12

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. 11): um ein D-Flipflop zu bauen, genügt es, J und \bar{K} einfach zusammenschalten.

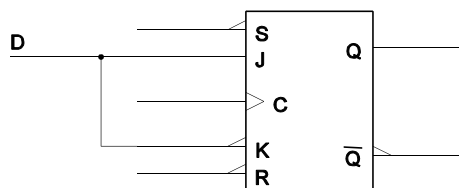


Abb. 11