

Digitaltechnik

Wintersemester 2021/2022

7. Übung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Prof. Dr.-Ing. Thomas Schneider, M.Sc. Daniel Günther, M.Sc. Amos Treiber
LÖSUNGSVORSCHLAG

KW49

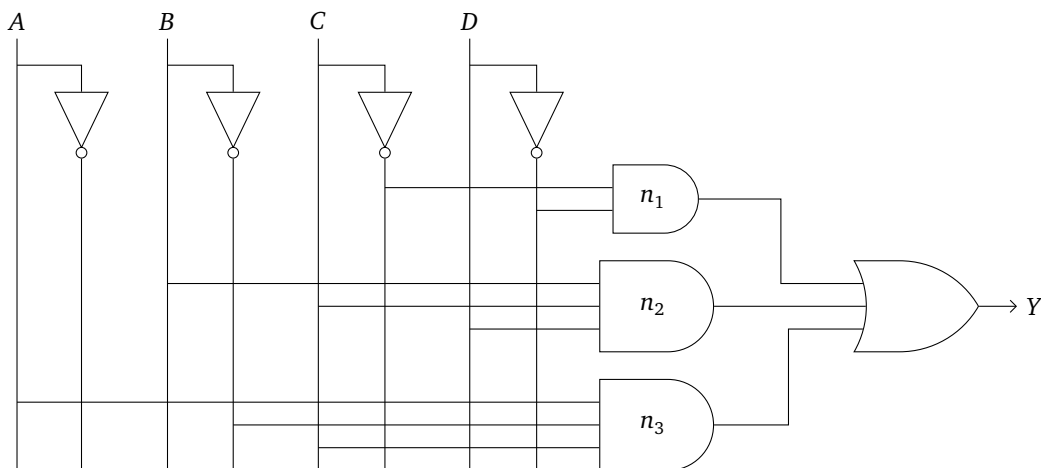
Bitte bearbeiten Sie die Übungsblätter bereits im Voraus, sodass Sie Ihre Lösungen zusammen mit Ihren Kommilitonen und Tutoren während der wöchentlichen Übungsstunde diskutieren können.

Mit der angegebenen Bearbeitungszeit für die einzelnen Aufgaben können Sie Ihren Leistungsstand besser einschätzen.

Übung 7.1 Störimpulse – Wiederholung

[20 min]

Gegeben sei die folgende Schaltung

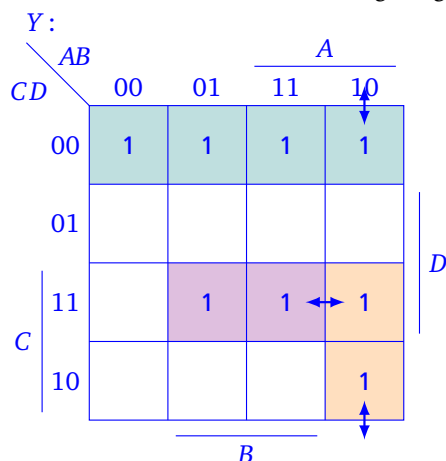


Gehen Sie davon aus, dass jedes Gatter eine Verzögerungszeit von 5 ns hat.

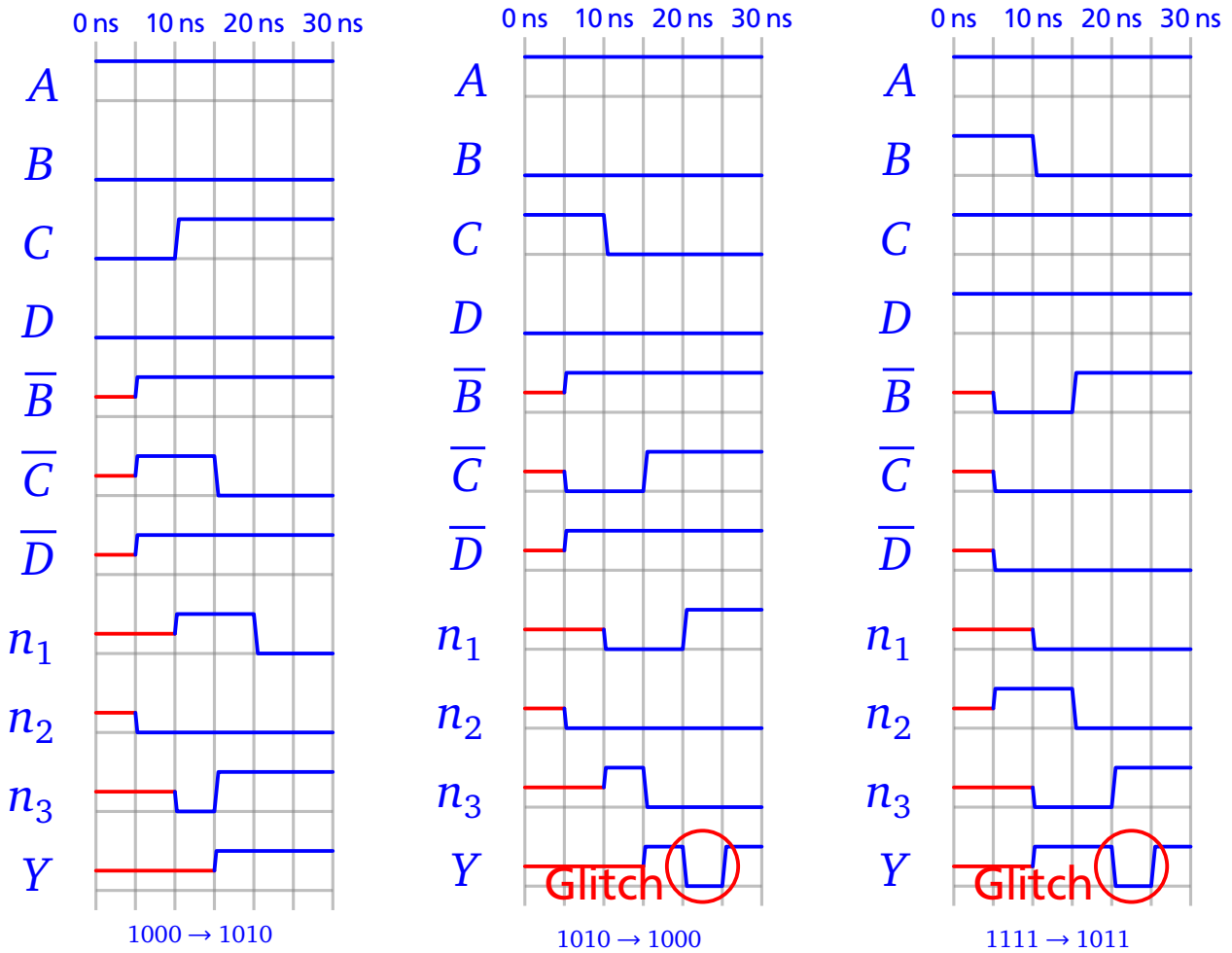
Übung 7.1.1 Störimpulse erkennen

Identifizieren Sie die kritischen Eingangstransitionen (mit nur einer geänderten Variable) im Karnaugh Diagramm bei denen Störimpulse auftreten können.

Verifizieren Sie anhand von Timing-Diagrammen, ob bei diesen Transitionen tatsächlich Störimpulse auftreten.



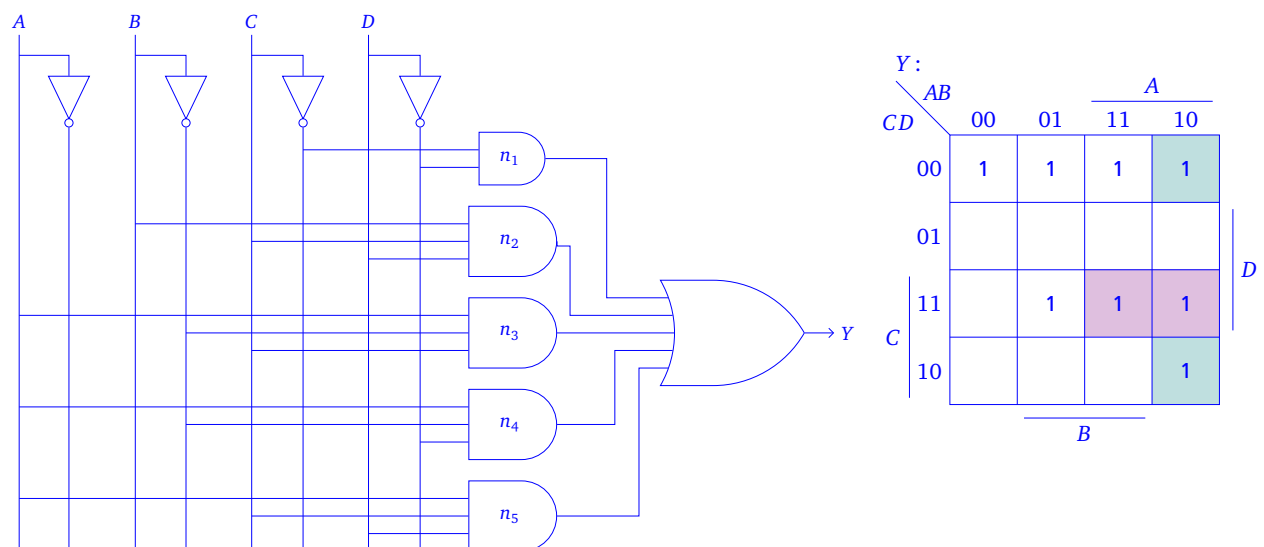
$$Y = \overline{C} \overline{D} + B C D + A \overline{B} C$$



Übung 7.1.2 Störimpulse beheben

Geben Sie eine funktional äquivalente Schaltung ohne Störimpulse an. Versuchen Sie dies mit möglichst wenigen Modifikationen der bestehenden Schaltung zu erreichen.

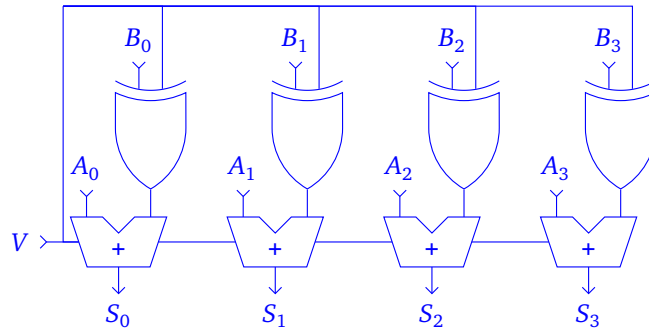
$$Y = \overline{C} \overline{D} + B C D + A \overline{B} C + \overline{A} \overline{B} \overline{D} + A C D$$



Übung 7.2.1 Ripple-Carry Addierer / Subtrahierer

Entwerfen Sie die Schaltung eines Ripple-Carry Addierers / Subtrahierers. Die beiden 4-bit breiten Zahlen A und B werden addiert, wenn am Eingang V eine 0 anliegt und subtrahiert, wenn am Eingang V eine 1 anliegt. Verwenden Sie dafür Volladdierer und die in der Vorlesung vorgestellten Gatter.

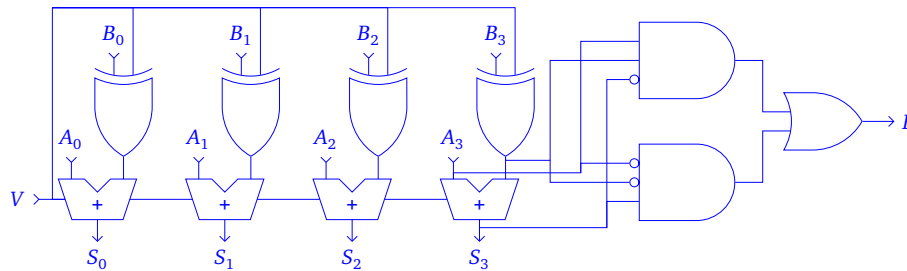
Bei gesetztem V wird der Eingang B invertiert und auf das Ergebnis 1 addiert. Die bedingte Invertierung kann bitweise mittels XOR Gattern realisiert werden und das erste Carrybit zur Addition von 1 verwendet werden.



Übung 7.2.2 Überlauf- und Unterlauferkennung

Erweitern Sie Ihre Schaltung um ein Ausgangsbit F , welches angibt, ob bei der Addition / Subtraktion ein Überlauf / Unterlauf aufgetreten ist.

Ein Überlauf bzw. Unterlauf tritt genau dann auf, wenn das Vorzeichenbit von A und B identisch sind und das Vorzeichen des Ergebnis S dazu invers ist. Es gilt also $F = A_3 \cdot B_3 \cdot \overline{S_3} + \overline{A_3} \cdot \overline{B_3} \cdot S_3$.



Übung 7.2.3 Analyse des Zeitverhaltens

Berechnen Sie die Ausbreitungsverzögerung t_{pd} und Kontaminationsverzögerung t_{cd} Ihres 4 Bit breiten Addierers / Subtrahierers mit Überlauf- und Unterlauferkennung. Gehen Sie dabei davon aus, dass die verwendeten Volladdierer wie in der Vorlesung vorgestellt umgesetzt wurden. Benutzen Sie dafür die folgenden Zeitcharakteristiken:

- | | | | |
|---------------------------------|---------------------------------|--------------------------------|---------------------------------|
| a) $t_{cd,NOT} = 10 \text{ ps}$ | c) $t_{cd,AND} = 20 \text{ ps}$ | e) $t_{cd,OR} = 20 \text{ ps}$ | g) $t_{cd,XOR} = 30 \text{ ps}$ |
| b) $t_{pd,NOT} = 20 \text{ ps}$ | d) $t_{pd,AND} = 40 \text{ ps}$ | f) $t_{pd,OR} = 40 \text{ ps}$ | h) $t_{pd,XOR} = 50 \text{ ps}$ |

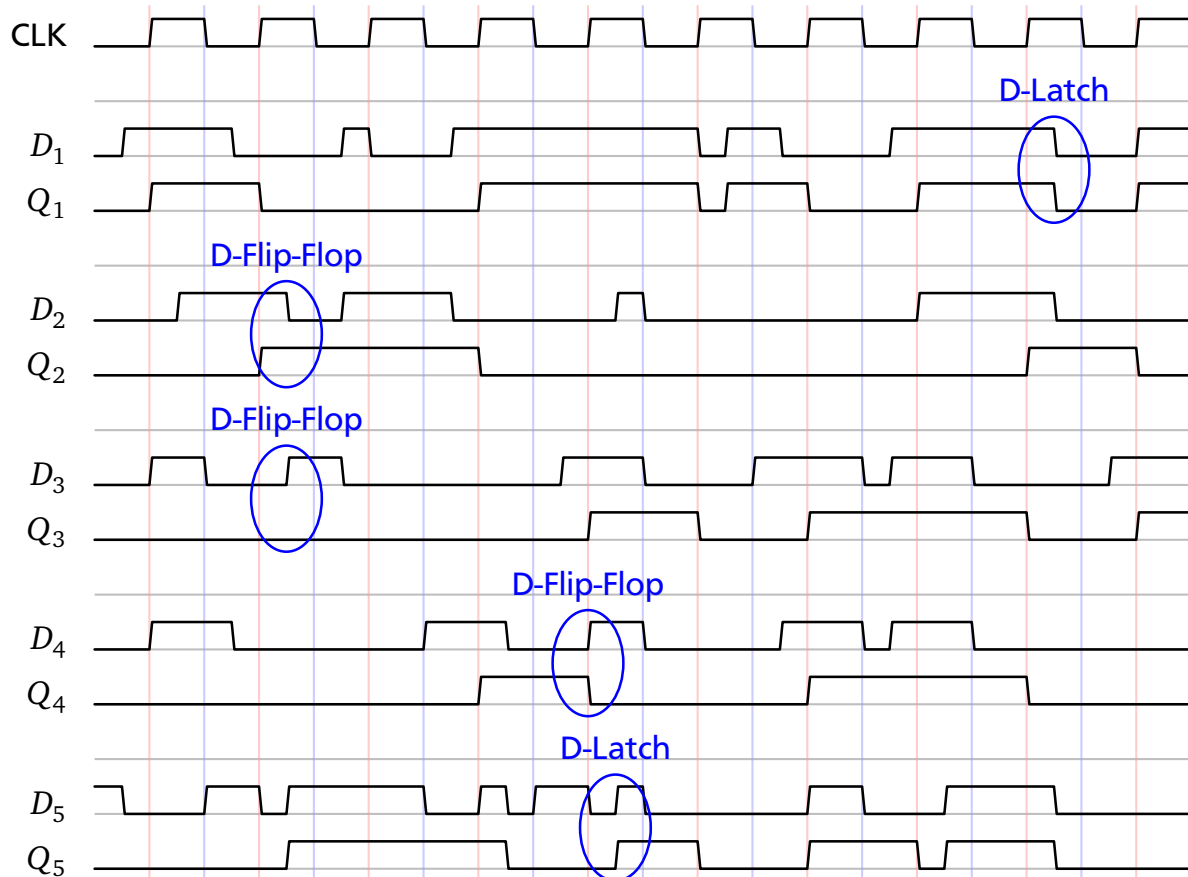
Der kritische Pfad des Volladdierers verläuft vom Eingang A_i bzw. B_i über 1 XOR, 1 AND und 1 OR Gate zum Ausgang C_i . Somit ergibt sich eine Ausbreitungsverzögerung von $t_{pd,VA} = t_{pd,XOR} + t_{pd,AND} + t_{pd,OR} = 50 \text{ ps} + 40 \text{ ps} + 40 \text{ ps} = 130 \text{ ps}$. Für den kritischen Pfad ist insbesondere noch der Pfad zwischen C_{i-1} nach C_i relevant, welcher eine Ausbreitungsverzögerung von $t_{pd,VA,C} = t_{pd,AND} + t_{pd,OR} = 40 \text{ ps} + 40 \text{ ps} = 80 \text{ ps}$ hat, sowie der Pfad zwischen C_{i-1} nach S mit einer Ausbreitungsverzögerung von $t_{pd,VA,S} = t_{pd,XOR} = 50 \text{ ps}$.

Der kürzeste Pfad des Volladdierers verläuft vom Carrybit C_{i-1} über ein XOR Gatter zum Ausgang S_i . Damit ergibt sich eine Kontaminationsverzögerung von $t_{cd,VA} = t_{cd,XOR} = 30 \text{ ps}$.

Offensichtlich verläuft der kritische Pfad des Addierers/Subtrahierers vom Eingang B_0 über 1 XOR Gatter, 4 Volladdierer, 1 Inverter, 1 AND Gatter und 1 OR Gatter zum Ausgang F . Damit ergibt sich die eine Ausbreitungsverzögerung von $t_{pd} = t_{pd,XOR} + t_{pd,VA} + 2t_{pd,VA,C} + t_{pd,VA,S} + t_{pd,NOT} + t_{pd,AND} + t_{pd,OR} = 50 \text{ ps} + 130 \text{ ps} + 2 \cdot 80 \text{ ps} + 50 \text{ ps} + 20 \text{ ps} + 40 \text{ ps} + 40 \text{ ps} = 490 \text{ ps}$. Der kürzeste Pfad des Addierers/Subtrahierers verläuft vom Eingang V über einen Volladdierer zum Ausgang S_0 . Damit ergibt sich eine Kontaminationsverzögerung von $t_{cd} = t_{cd,VA} = 30 \text{ ps}$.

Übung 7.3.1 Am Schaltverhalten erkennen

Das folgende Timing-Diagramm beschreibt das Schaltverhalten von fünf Speicherelementen. Gegeben ist der Takteingang CLK, der Dateneingang D_i und der Datenausgang Q_i . Geben Sie an, welche Speicherelemente Flip-Flops und welche Latches sind. Markieren Sie Ihre Antwort im Diagramm und begründen Sie diese.



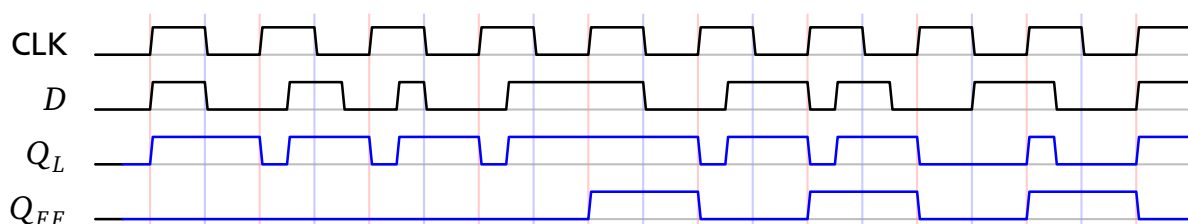
Ein D-Latch ist Takt-phasengesteuert, ein Flip-Flop hingegen Takt-flankengesteuert. Das bedeutet, dass D-Flip-Flops nur zu steigender Taktflanke den Wert des Eingangs auf den Ausgang übernehmen. Ein D-Latch hingegen kann den Ausgang immer ändern – solange der anliegende Takt 1 ist; zu einer fallenden Taktflanke sieht ein Latch noch den alten Wert.

Im Timing-Diagramm ist für jedes Speicherelement eine Stelle markiert, an der die Zuordnung festgemacht werden kann:

- D_1 wechselt mitten in 1-Phase der CLK, was von Q_1 übernommen wird \Rightarrow D-Latch
- D_2/D_3 wechselt mitten in 1-Phase der CLK, ohne dass dies von Q_2/Q_3 übernommen wird \Rightarrow D-Flip-Flop
- D_4 : Zu steigender Taktflanke wird der alte Wert von D_4 übernommen \Rightarrow D-Flip-Flop
- D_5 wechselt mitten in 1-Phase der CLK, was von Q_5 übernommen wird \Rightarrow D-Latch

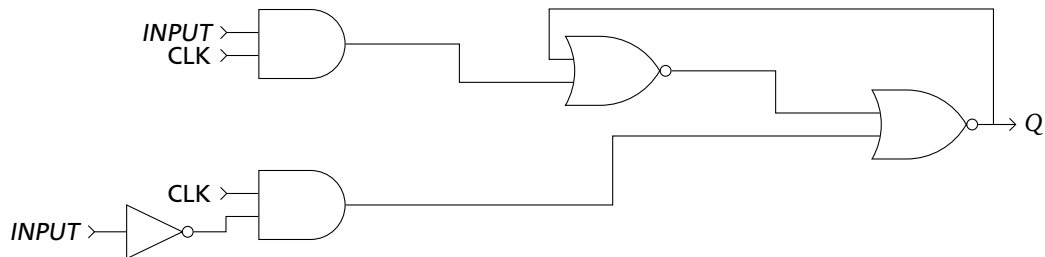
Übung 7.3.2 Schaltverhalten reproduzieren

Ergänzen Sie im folgenden Timing-Diagramm das Schaltverhalten eines D-Latches mit Ausgang Q_L und eines D-Flip-Flops mit Ausgang Q_{FF} . Beide Speicherelemente werden vom gleichen Takt- (CLK) und Datensignal (D) gesteuert.



Übung 7.4.1 An Logikgatterschaltung erkennen

Folgende Logikgatterschaltung beschreibt eines der in der Vorlesung vorgestellten Speicherelemente. Um welches Speicherelement handelt es sich und wie funktioniert dieses?

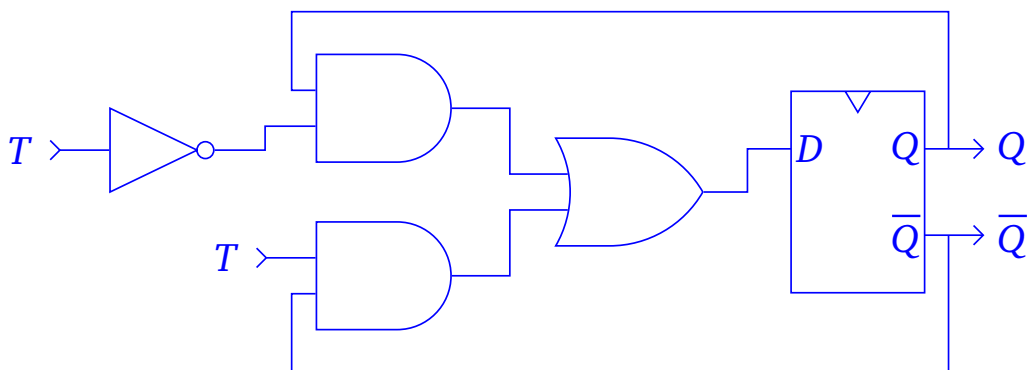


Für $CLK = 0$ wird der Zustand Q gehalten. Für $CLK = 1$ wird der Zustand Q auf $INPUT$ gesetzt. Dies entspricht der Funktionalität von einem D-Latch.

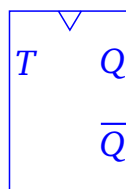
Übung 7.4.2 Toggle-Flip-Flops

Entwerfen Sie ein T-Flip-Flop. Der Zustand Q wechselt bei jeder steigenden Taktflanke genau dann, wenn am Eingang T eine 1 anliegt und behält ansonsten seinen Wert – es gibt kein Datensignal D . Verwenden Sie zur Realisierung ein D-Flip-Flop und primitive Gatter (AND, OR, Inverter).

Die Übergangsgleichung eines T-Flip-Flops ist $Q' = Q \bar{T} + \bar{Q} T$. Somit muss diese Funktion vor den Eingang des verwendeten D-Flip-Flops geschaltet werden.



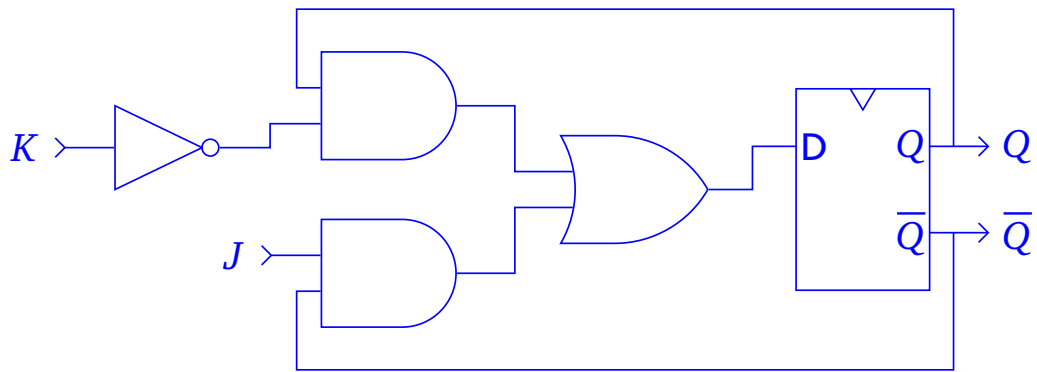
Ein T-Flip-Flop wird mit folgendem Symbol dargestellt:



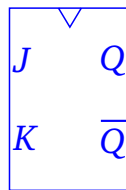
Übung 7.4.3 JK-Flip-Flops

Entwerfen Sie ein JK-Flip-Flop. Am Ausgang Q liegt nach steigender Taktflanke eine 1 an, wenn am Eingang J eine 1 und am Eingang K eine 0 anliegt. Sind die Eingänge $J = 0$ und $K = 1$, so liegt nach der steigenden Taktflanke eine 0 an Q an. Für $J = K = 0$ hält das JK-Flip-Flop seinen vorherigen Zustand. Im Fall $J = K = 1$ wechselt der Zustand von Q zu \bar{Q} . Verwenden Sie zur Realisierung ein D-Flip-Flop und primitive Gatter (AND, OR, Inverter).

Die Übergangsgleichung eines JK-Flip-Flops ist $Q' = Q \bar{K} + \bar{Q} J$. Somit muss diese Funktion vor den Eingang des verwendeten D-Flip-Flops geschaltet werden.

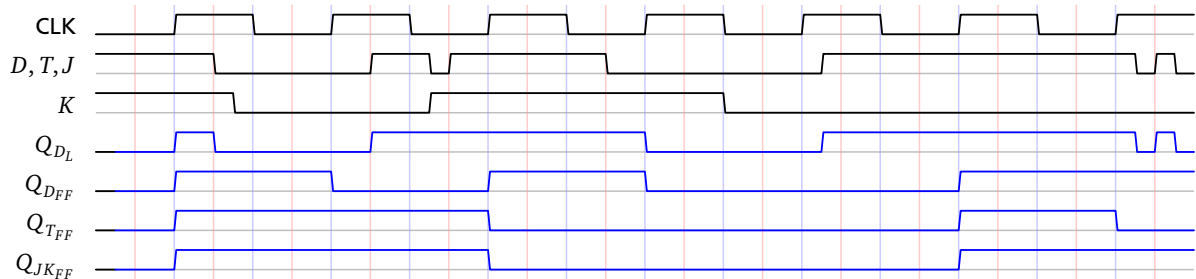


Ein JK-Flip-Flop wird mit folgendem Symbol dargestellt:



Übung 7.4.4 Schaltverhalten reproduzieren

Ergänzen Sie im folgenden Timing-Diagramm das Schaltverhalten eines D-Latches mit Ausgang Q_{DL} , eines D-Flip-Flops mit Ausgang Q_{DFF} , eines T-Flip-Flops mit Ausgang Q_{TFF} und eines JK-Flip-Flops mit Ausgang Q_{JKFF} .



Übung 7.5 2-Bit Zähler EX9-4-1 EX9-4-2

[15 min]

Entwerfen Sie einen Zähler, der eine 2-Bit Zahl B_1B_0 mit jedem Takt um 1 erhöht. Wird die höchste darstellbare Zahl erreicht, fängt der Zähler erneut bei 0 an. Verwenden Sie dafür 2 D-Flip-Flops und primitive Logikgatter. Beachten Sie, dass CLK ausschließlich mit den dafür vorgesehenen Inputs der Flip-Flops verbunden werden soll.

Hinweis: Für die minimalste Lösung wird neben den Flip-Flops nur 1 weiteres Gatter benötigt.

