

Prof. Dr. Christoph Scholl
Tobias Seufert

Freiburg, 28. Juni 2023

Technische Informatik Musterlösung zu Übungsblatt 10

Hinweis: Auf diesem Blatt befindet sich eine „Bonusaufgabe“. Diese Aufgabe zählt nicht in die Gesamtheit der Aufgaben, bei sinnvoller Bearbeitung wird sie jedoch zur Menge der sinnvoll bearbeiteten Aufgaben gerechnet.

Aufgabe 1 (3 Punkte)

Geben Sie einen möglichst kleinen Schaltkreis an, der zu einer n -Bit-Zweierkomplementzahl deren Betrag berechnet, das heißt, entwerfen Sie einen Schaltkreis zu der booleschen Funktion

$$abs_n : \mathbb{B}^n \rightarrow \mathbb{B}^n, (a_{n-1}, \dots, a_0) \mapsto (s_{n-1}, \dots, s_0)$$

mit $\langle s_{n-1}, \dots, s_0 \rangle = |[a_{n-1}, \dots, a_0]|$.

Bestimmen Sie die Kosten Ihrer Schaltkreisrealisierung.

Hinweis: Berücksichtigen Sie die Definition des Betrags einer Zahl r mit:

$$|r| = \begin{cases} r, & \text{falls } r \geq 0 \\ -r, & \text{falls } r < 0. \end{cases}$$

Aus der Vorlesung bekannte Schaltkreise (wie Addierer, Inkrementer) dürfen Sie verwenden.

Kann bei der Berechnung des Betrags einer n -Bit-Zweierkomplement-Zahl ein Übertrag auftreten? Wenn ja, in welchem Fall? Ergänzen Sie Ihre Schaltung gegebenenfalls um eine Überlauferkennung!

Lösung:

Siehe Abbildung 1.

$$C(SK) = C(INC_{n-1}) + (n-1) \cdot C(XOR) = (n-1) \cdot C(HA) + (n-1) = 3n - 3$$

Bewertung: Umsetzung [2], Kosten [1]

Aufgabe 2 (3 Punkte)

Eine alternative, vom Carry-Ripple-Addierer inspirierte Möglichkeit, eine ALU zu entwerfen, ist in Abbildung 2 skizziert. Dabei gibt es eine Basiszelle „CR-ALU“, die jeweils nur ein einziges Bit des Ergebnisses r und ein eventuell notwendiges Carry-Bit berechnet.

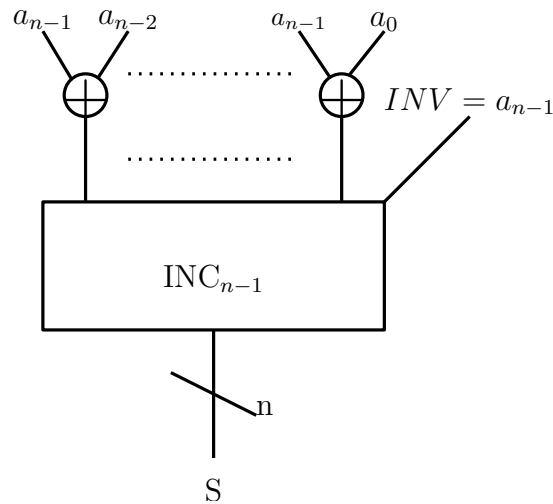


Abbildung 1: Schaltkreis BETRAG_n

Geben Sie eine Realisierung der „CR-ALU“-Basiszelle an, so dass über den *select*-Eingang mit $m = 3$ Bit dieselben Operationen realisiert werden wie in der aus der Vorlesung bekannten ALU. Sie müssen die Schaltung nicht unbedingt zeichnen. Eine ausführliche und nachvollziehbare Beschreibung ist ausreichend.

Hinweis: Wiederholen Sie Kapitel 3.6, Folie 10ff – Option 2. Beachten Sie auch, dass das select-Signal mit den richtigen Eingangssignalen s_0, s_1, s_2 verbunden werden muss.

Durch welche Schaltung muss dabei das Signal c_{-1} berechnet werden?

Lösung:

„CR-ALU“ entspricht einer „normalen“ ALU wie in der Vorlesung definiert für die Wortbreite 1 [0.5], allerdings mit folgenden Änderungen:

- bei A_n handelt es sich jetzt um einen Volladdierer FA (1-Bit Addierer) [0.5]
- das Carry c wird direkt an den Volladdierer übergeben [0.5]
- um das Carry an die nächste Zelle weiterzuleiten, muss CR-ALU in Kontrast zur ALU um einen Carry-Ausgang erweitert werden [0.5]

c_{-1} kann mit der wegfallenden Schaltung zur Carry-Generierung aus der ALU berechnet werden. [1]

Korrekturhinweis: Die Schaltung muss nicht explizit gezeichnet werden. Wie hier reicht eine Beschreibung aus, sofern sie eindeutig nachvollziehbar ist.

Aufgabe 3 (2 + 1 + 1 + 1 Punkte)

Ihnen gefällt das in der Vorlesung vorgestellte RS-Flipflop nicht, weswegen Sie beschließen, sich einen anderen, aber relativ ähnlichen Schaltkreis aus OR- und NOT Bausteinen anzusehen (siehe Abbildung 4).

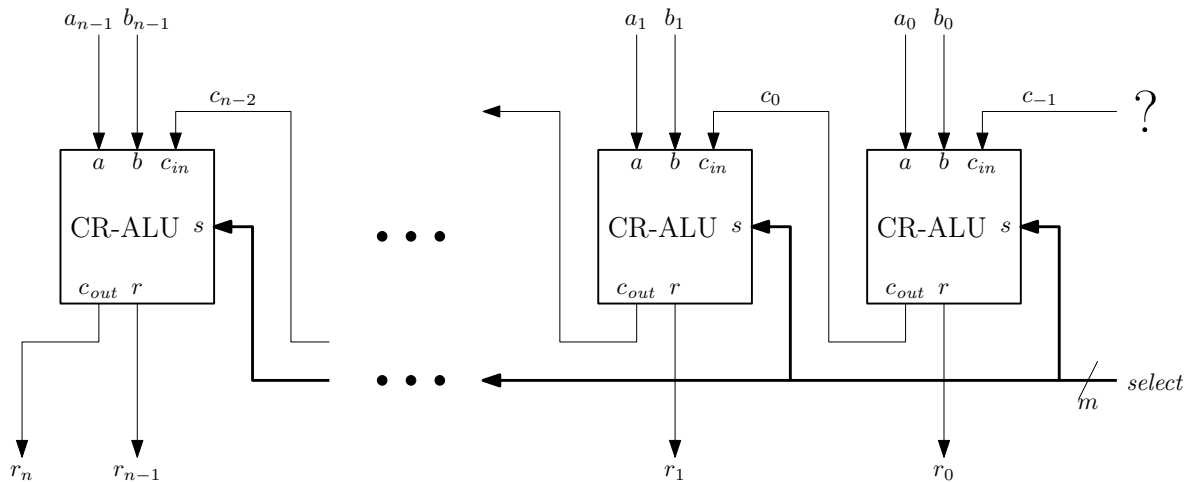


Abbildung 2: Eine vom Carry-Ripple-Addierer inspirierte ALU-Realisierung

- Welche stabilen Belegungen gibt es für diesen Schaltkreis? Geben Sie dazu die Werte für a , b , c und d an.
- Geben Sie an, bei welcher Eingangsbelegung ein gespeicherter Wert gehalten wird und durch welche Eingangsbelegungen ein neuer Wert gespeichert werden kann.
- Sind a und b active-low oder active-high? Begründen Sie!
- Welche der stabilen Belegungen ergibt bei der Verwendung als speicherndes Element keinen Sinn? Was ist der Grund dafür?

Lösung:

- Es gibt fünf stabile Belegungen:

- 1) $a = 0, b = 0, c = 0, d = 1$
- 2) $a = 0, b = 0, c = 1, d = 0$
- 3) $a = 0, b = 1, c = 1, d = 0$
- 4) $a = 1, b = 0, c = 0, d = 1$
- 5) $a = 1, b = 1, c = 0, d = 0$

Bewertung: pro falscher oder nicht aufgezählter Belegung [-0.5] (keine Minuspunkte), Belegung gilt nur als korrekt, wenn alle Werte für a , b , c und d angegeben sind.

- Bei $a = b = 0$ wird der aktuelle Wert gehalten, mit $a = 0, b = 1$ wird c auf 1 und d auf 0 gesetzt und mit $a = 1, b = 0$ wird c auf 0 und d auf 1 gesetzt.

Bewertung: richtig oder falsch, Belegung um Wert zu halten [0.5], Belegung um Wert zu speichern [0.5], keine Begründung/Erläuterung gefragt/notwendig.

- a und b sind active-high, da sie durch das Heben auf 1 aktiviert werden.

Bewertung: richtig oder falsch (falsche/keine Begründung \rightarrow [0])

- Die Belegung $a = 1, b = 1$ ergibt keinen Sinn, da es bei gleichzeitigem Absenken von a und b zu Flackern kommen kann.

Bewertung: Belegung erkannt [0.5], Begründung [0.5]

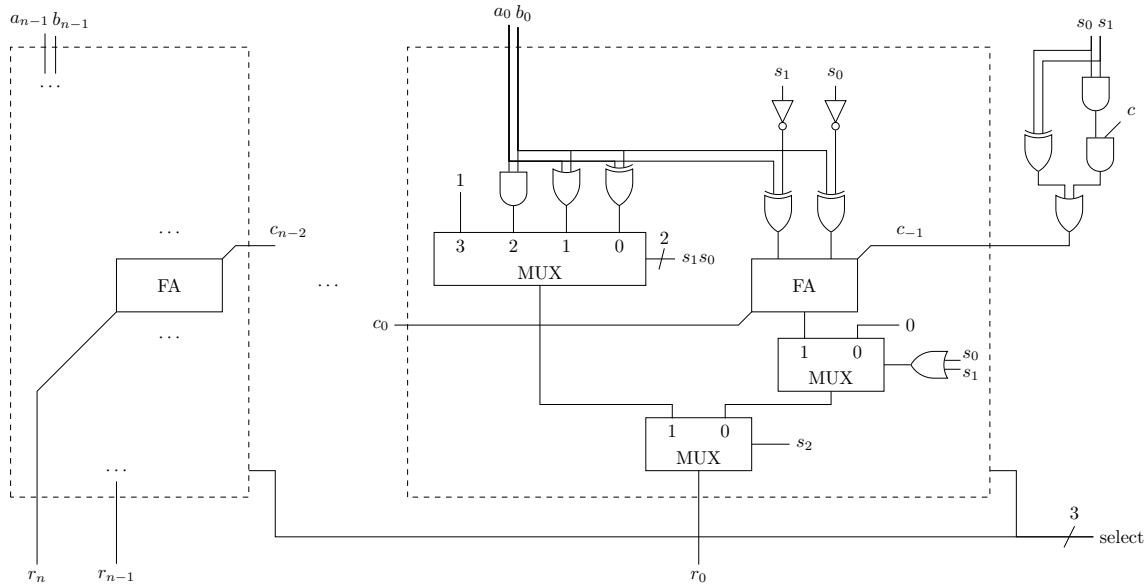


Abbildung 3: CR_ALU Details.

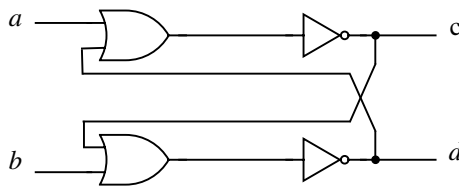


Abbildung 4: Kein RS-Flipflop

Aufgabe 4 (Bonusaufgabe: 3 Punkte)

Betrachten Sie das in der Vorlesung vorgestellte D-Flip-Flop in Abbildung 5.

Betrachten Sie nun einen Zeitraum von zwei Takten. Während dieser Zeitspanne ändert sich der Dateneingang D von 0 auf 1 (siehe Timingdiagramm in Abbildung 6). Vervollständigen Sie das Timingdiagramm aus Abb. 6 für die Signale $/S_2$, Q_1 , $/Q_1$, Q_2 , $/Q_2$, Q_3 und $/Q_3$, wobei die Q_i s für die entsprechenden Q -Signale des RS-Flip-Flops i stehen (siehe Abb. 5).

Ergänzen Sie dabei zunächst die Werte für die Signale vor dem ersten Flankenanstieg der $Clock$. Sie können davon ausgehen, dass sich alle Werte zu Beginn bereits eingestellt haben. Dabei sei der Zustand des RS-Flip-Flops 3 auf „ $Q = 1$ “, d.h. es gilt vor dem ersten betrachteten Flankenwechsel von $Clock$: $Q_3 = 1$ und $/Q_3 = 0$.

Geben Sie danach für jedes Signal einen schematischen Verlauf in der Zeit an – Sie brauchen hier *keine* exakte Timinganalyse vornehmen, es sollte aber klar werden, wie sich die Signale untereinander beeinflussen. Die Werte für $Clock$ und D sind bereits für den gesamten Zeitraum, den Sie betrachten sollen, angegeben (vergessen Sie dabei nicht das Stück nachdem $Clock$ zum zweiten Mal auf 0 gesetzt wird).

Sie dürfen davon ausgehen, dass nach allen Änderungen von D und $Clock$ genügend Zeit vergeht,

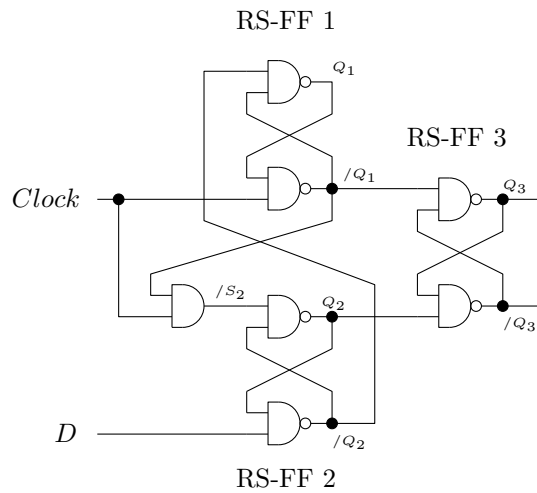


Abbildung 5: D-Flipflop

so dass alle Schaltvorgänge der Gatter bis zur nächsten Änderung auf *D* bzw. *Clock* vollständig abgeschlossen sind.

Eine Vorlage des Timingdiagramms in PDF-Format finden Sie bei den Vorlesungsmaterialien unter “Zusatzmaterial“.

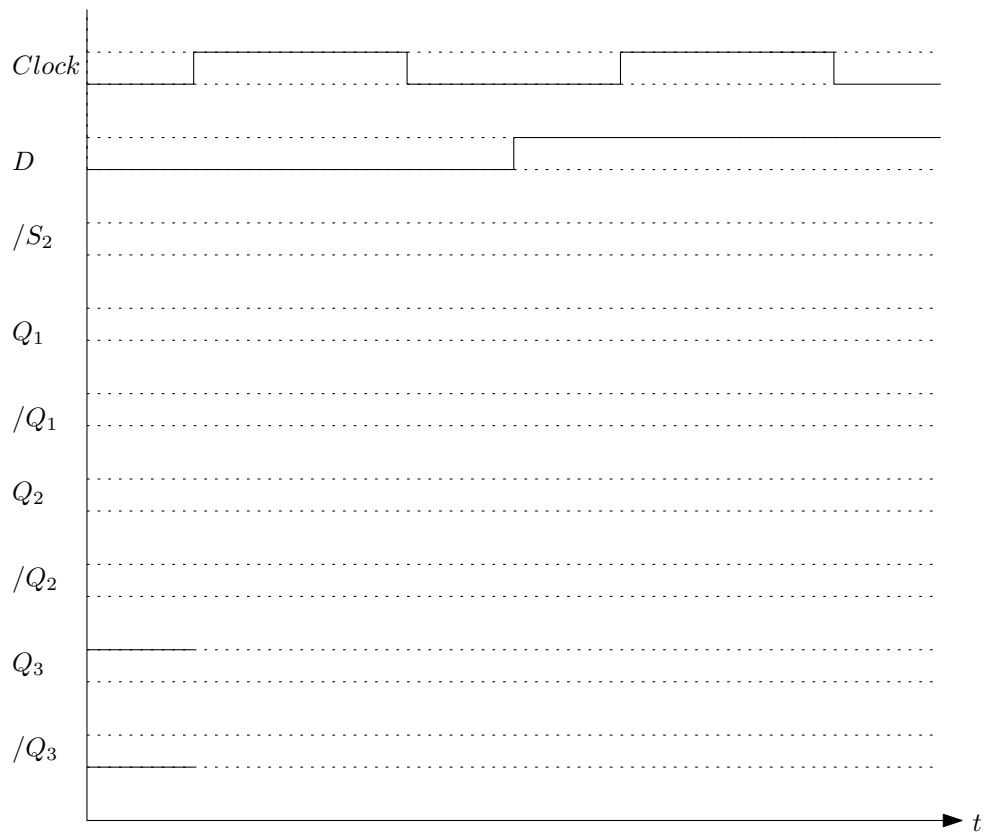


Abbildung 6: Timingdiagramm

Lösung:

a) Zunächst ein paar Beobachtungen:

$$\begin{aligned} /S_1 &= /Q_2, \quad /R_1 = Clock \\ /S_2 &= Clock \ \& \ /Q_1, \quad /R_2 = D \\ /S_3 &= /Q_1, \quad /R_3 = Q_2 \end{aligned}$$

Desweiteren:

$$\begin{aligned} Q_1 &= \overline{/Q_1 \ \& \ /Q_2} \\ /Q_1 &= \overline{Q_1 \ \& \ Clock} \\ Q_2 &= \overline{(Clock \ \& \ /Q_1) \ \& \ /Q_2} \\ /Q_2 &= \overline{Q_2 \ \& \ D} \\ Q_3 &= \overline{/Q_1 \ \& \ /Q_3} \\ /Q_3 &= \overline{Q_3 \ \& \ Q_2} \end{aligned}$$

- **clk : 0 → 1** ⇒ $/S_2 : 0 \rightarrow 1, \ /Q_1 : 1 \rightarrow 1$
 $/S_2 : 0 \rightarrow 1 \Rightarrow Q_2 : 1 \rightarrow 0$
 $Q_2 : 1 \rightarrow 0 \Rightarrow /Q_2 : 1 \rightarrow 1, \ /Q_3 : 0 \rightarrow 1$
 $/Q_3 : 0 \rightarrow 1 \Rightarrow Q_3 : 1 \rightarrow 0$
 $Q_3 : 1 \rightarrow 0 \Rightarrow /Q_3 : 1 \rightarrow 1$
- **clk : 1 → 0** ⇒ $/S_2 : 1 \rightarrow 0, \ /Q_1 : 1 \rightarrow 1$
 $/S_2 : 1 \rightarrow 0 \Rightarrow Q_2 : 0 \rightarrow 1$

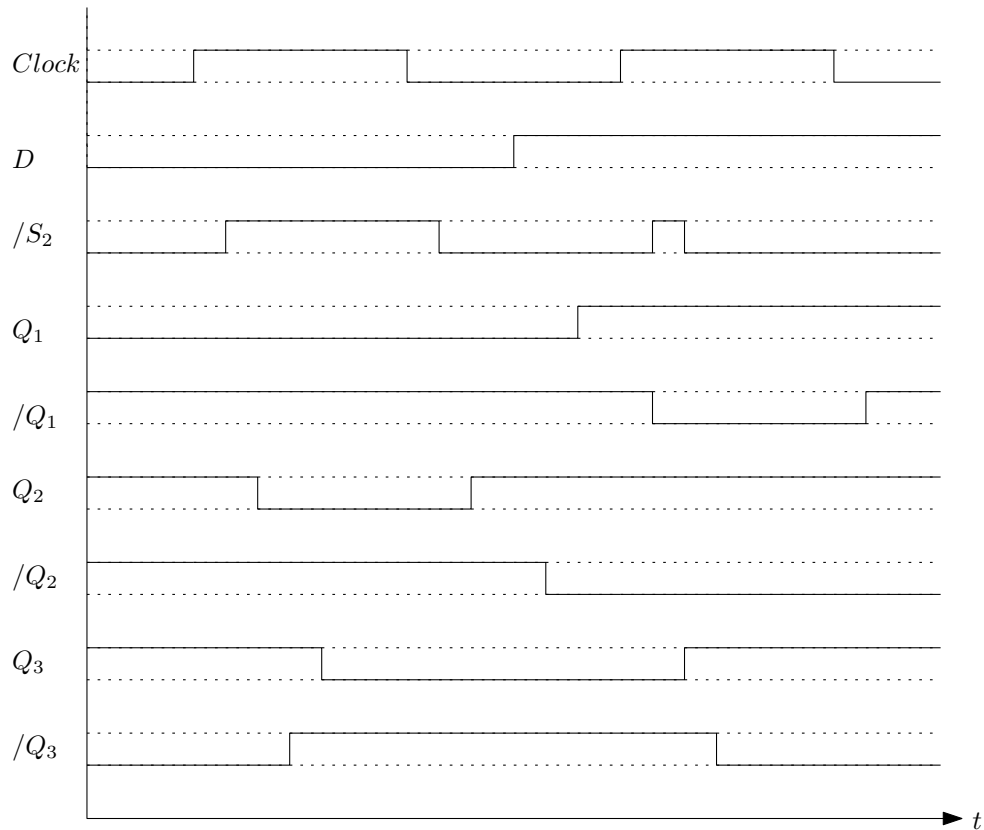


Abbildung 7: Timingdiagramm (ausgefüllt)

$Q_2 : 0 \rightarrow 1 \Rightarrow /Q_2 : 1 \rightarrow 1, /Q_3 : 1 \rightarrow 1$

- $D : 0 \rightarrow 1 \Rightarrow /Q_2 : 1 \rightarrow 0$
 $/Q_2 : 1 \rightarrow 0 \Rightarrow Q_2 : 1 \rightarrow 1, Q_1 : 0 \rightarrow 1$
 $Q_1 : 0 \rightarrow 1 \Rightarrow /Q_1 : 1 \rightarrow 1$
- $clk : 0 \rightarrow 1 \Rightarrow /S_2 : 0 \rightarrow 1, /Q_1 : 1 \rightarrow 0$
 $/S_2 : 0 \rightarrow 1 \Rightarrow Q_2 : 1 \rightarrow 1$
 $/Q_1 : 1 \rightarrow 0 \Rightarrow Q_1 : 1 \rightarrow 1, Q_3 : 0 \rightarrow 1, /S_2 : 1 \rightarrow 0$
 $Q_3 : 0 \rightarrow 1 \Rightarrow /Q_3 : 1 \rightarrow 0$
 $/S_2 : 1 \rightarrow 0 \Rightarrow Q_2 : 1 \rightarrow 1$
 $/Q_3 : 1 \rightarrow 0 \Rightarrow Q_3 : 1 \rightarrow 1$
- $clk : 1 \rightarrow 0 \Rightarrow /S_2 : 0 \rightarrow 0, /Q_1 : 0 \rightarrow 1$
 $/Q_1 : 0 \rightarrow 1 \Rightarrow Q_1 : 1 \rightarrow 1, Q_3 : 1 \rightarrow 1, /S_2 : 0 \rightarrow 0$

Der kurze Puls bei $/S_2$ ist unkritisch, da der andere Eingang ($/Q_2$) des betreffenden Gatters, während dieses Spikes den Kontrollwert 0 hat – das NAND kann also nicht auf 0 schalten.

Punkte: Bearbeiten der Aufgabe [1.5]. Für je korrekt gesetzte Flanke [0.25]; für alle 14 Flanken also insgesamt [3,5]. Folgefehler beachten. Signaleinflüsse untereinander müssen kenntlich sein, sonst insgesamt -1.5.

b) **Überbleibsel aus ursprünglicher Aufgabe. Muss hier nicht beachtet werden.**

Zur Erinnerung die 4 Zustände eines RS-FF:

- $/S = 0, /R = 0 \rightarrow Q = 1, /Q = 1$: „Metastabiler Zustand“ („MZ“)
- $/S = 0, /R = 1 \rightarrow Q = 1, /Q = 0$: „Set“
- $/S = 1, /R = 0 \rightarrow Q = 0, /Q = 1$: „Reset“
- $/S = 1, /R = 1 \rightarrow Q = x, /Q = \bar{x}$: „Wert speichern“ („WS“) Der Wert x hängt davon ab, ob zuvor ein „Set“ oder „Reset“ ausgeführt wurde.

Allgemein gilt für das D-FF: der Wert, der gespeichert werden soll, steht in RS-FF 3. Solange $Clock = 0$ ist, gilt: $/S_3 = /R_3 = 1$. Damit ist RS-FF 3 in Zustand „WS“. Die anderen beiden RS-FFs sind dafür da, dass RS-FF 3 in den richtigen Zustand („Set“ bzw. „Reset“) gesetzt wird, je nachdem welcher Wert D hat. Wenn $D = 0$ muss auf RS-FF 3 ein „Reset“ ausgeführt werden, bei $D = 1$ ein „Set“.

Fallunterscheidung bzgl. D :

- $D = 0$: Solange $Clock$ genügend lange auf 0 ist (Setup-Zeit), ist RS-FF 2 im „MZ“ und RS-FF 1 im „Reset“ (damit RS-FF 3 im „WS“). Ändert sich $Clock$ auf 1, so geht RS-FF 1 in „WS“ über (speichert also: $Q_1 = 0, /Q_1 = 1$, damit weiterhin $/S_3 = 1$) und RS-FF 2 in geht in „Reset“, da $/S_2$ durch Flankenwechsel 1 wird und D immernoch 0 ist (somit $Q_2 = 0, /Q_2 = 1$ und dadurch: $/R_3 = 0$). RS-FF 3 ist nun in „Reset“, was gewünscht war. Nun muss die Hold-Zeit von D eingehalten werden und wenn schließlich $Clock$ wieder auf 0 geht, geht $/S_2$ wieder auf 0 und RS-FF2 in „MZ“ $\rightarrow Q_2 = 1, /Q_2 = 1$ und RS-FF 1 wieder in „Reset“. Damit ist RS-FF 3 in „WS“ bis $Clock$ wieder ansteigt.
- $D = 1$: Solange $Clock$ genügend lange auf 0 ist (Setup-Zeit), ist RS-FF 2 im „Set“ und RS-FF 1 im „MZ“ (damit RS-FF 3 im „WS“). Ändert sich $Clock$ auf 1, so geht RS-FF 1 in „Set“ über (also: $Q_1 = 1, /Q_1 = 0$, damit $/S_3 = 0$) und RS-FF 2 wechselt kurz in „WS“ und geht schließlich wieder in „Set“ (somit $Q_2 = 1, /Q_2 = 0$ und dadurch: $/R_3 = 1$). RS-FF 3 ist nun in „Set“, was gewünscht war. Nun muss die Hold-Zeit von D eingehalten werden und wenn schließlich $Clock$ wieder auf 0 geht, geht RS-FF1 in „MZ“ $\rightarrow Q_1 = 1, /Q_1 = 1$ und RS-FF in „Set“, damit ist RS-FF 3 in „WS“ bis $Clock$ wieder ansteigt.

Zusammengefasst: D sorgt dafür, dass RS-FF 1 und RS-FF 2 jeweils im „richtigen“ Zustand ist, so dass ein Flankenwechsel an $Clock$ auf 1 den RS-FF 3 richtig einstellt und bei $Clock$ auf 0 RS-FF 3 in „WS“ übergeht.

D	$Clock$	RS-FF 1	RS-FF 2	RS-FF 3
0	0	Reset	MZ	WS
0	$\rightarrow 1$	WS	Reset	Reset
0	$\rightarrow 0$	Reset	MZ	WS
1	0	MZ	Set	WS
1	$\rightarrow 1$	Set	Set	Set
1	$\rightarrow 0$	MZ	Set	WS

Punkte: insgesamt [4 oder 2]; Muss nicht so ausführlich sein; Wichtig zu erkennen (inklusive Erläuterung): RS-FF 3 ist das Flipflop, das den Wert speichert [1/0.5], RS-FF 1 und RS-FF 2 sind für das korrekte Einstellen von RS-FF 3 da [2/1]; Warum der Wert am Ende gespeichert bleibt [1/0.5]

Abgabe: 05. Juli 2023, 13⁰⁰ über das Übungsportal