Digitaltechnik
Übung 9

Matteo Dietz
mdietz@student.ethz.ch

Polybox

- https://polybox.ethz.ch/index.php/s/VehRU12QqdJv98i

Administratives

- Heute letzte „normale“ Übungsstunde
- Nächste Woche Test 3 (Automaten)

Ablauf:

- Repetition Automaten
- Umwandlung Mealy <-> Moore
- Automaten Don‘t Cares
- Zähler und Frequenzteiler
- Prüfungsaufgaben
- Fragen

Theorie – Kochrezept

Vorgehensweise:

1. Zustandsmenge bestimmen, daraus folgt die Anzahl der Zustandsvariablen und der erforderlichen D-Flipflops (mit n-FF, 2^n Zustände speichern)
2. Definition der Ein- und Ausgangsvariablen, Kodierung
3. Darstellung der Zustandsfolge in einem Zustandsdiagramm
4. Aufstellung der Zustandsfolgetabelle
5. Bestimmung der minimierten Ausgangs- und Übergangs-funktionen mit der Hilfe von KV-Diagrammen
6. Prüfung auf unbenutzte Zustände
7. Konstruktion des Schaltplans anhand der Schaltfunktion

Prüfungsaufgabe (HS 2020)

Als erste Aufgabe bei Ihrem neuen Arbeitsgeber müssen Sie den Automaten im Kern einer Kaffeemaschine programmieren, die Espresso (gemahlene Bohnen + 1 dl Wasser), Kaffee (gemahlene Bohnen + 2 dl Wasser) und Cappuccino (gemahlene Bohnen + 2 dl Wasser + 1 dl Milch) vorbereitet, wie in Abbildung A1 gezeigt. Dieser Automat muss die verschiedenen Phasen der Vorbereitung dieser Getränke regeln können. Wenn ein Getränk ausgewählt wurde (Signal **ES** für Espresso, **KA** für Kaffee und **CAP** für Cappuccino) müssen zuerst die Kaffeebohnen gemahlen werden (Signal **BM**). Dann muss das Wasser erwärmt und hinzugefügt werden (Signal **WH**). Nur 1 dl Wasser kann gleichzeitig behandelt werden: um einen Kaffee oder ein Cappuccino vorzubereiten, müssen zwei Wasserdosen nacheinander bearbeitet werden. Im Fall von Cappuccinos muss auch die Milch erhitzt werden. Sie wird nach dem Wasser in einem weiteren Schritt hinzugefügt (Signal **MH**).

Es wird angenommen, dass diese Kaffeemaschine aus 3 Teilen besteht, die von einem einzigen Automaten gesteuert werden: ein erster um die Bohnen zu mahlen, ein anderer, der sich mit dem Wasser beschäftigt und ein letzter, der sich um die Milch kümmert. Wenn diese Kaffeemaschine im Ruhezustand ist, also, wenn es kein Eingangssignal (**KE**) gibt, muss sie nichts machen (Signal **MN**). Wenn ein Getränk bereit ist, muss es von der Maschine angezeigt werden (Signal **GB**), damit der Benutzer weiss, dass er seine Tasse wegnehmen kann. Um diesen Automaten zu entwerfen müssen noch die folgenden Bedingungen berücksichtigt werden:

- Bohnen müssen immer zuerst gemahlen werden, dann muss Wasser (eine oder zwei Dosen) und schlussendlich eventuell Milch hinzugefügt werden. Diese Sequenz bleibt unverändert;
- die Kaffeemaschine kann nur ein einziges Getränk (**ES**, **KA**, oder **CAP**) gleichzeitig vorbereiten. Wenn ein Getränk gewählt worden ist, kann kein anderes selektiert werden, bevor das erste fertig ist;
- wenn ein Getränk selektiert worden ist, dann bleibt das entsprechende Signal (**ES**, **KA**, **CAP**) solange gespeichert, bis die Kaffeemaschine in ihren Ruhezustand zurückkehrt;
- wenn ein Getränk bereit ist, muss die Kaffeemaschine darauf hinweisen (**GB**), bevor sie wieder im Ruhezustand ist.

1. Was sind die Eingänge und die Ausgänge dieses Automaten? Verwenden Sie die folgenden Variablen: BM, CAP, ES, GB, KA, KE, MH, MN und WH. (2 Punkte)

Eingänge: ES, KA, CAP, KE

Ausgänge: MN, BM, WH, MH, GB

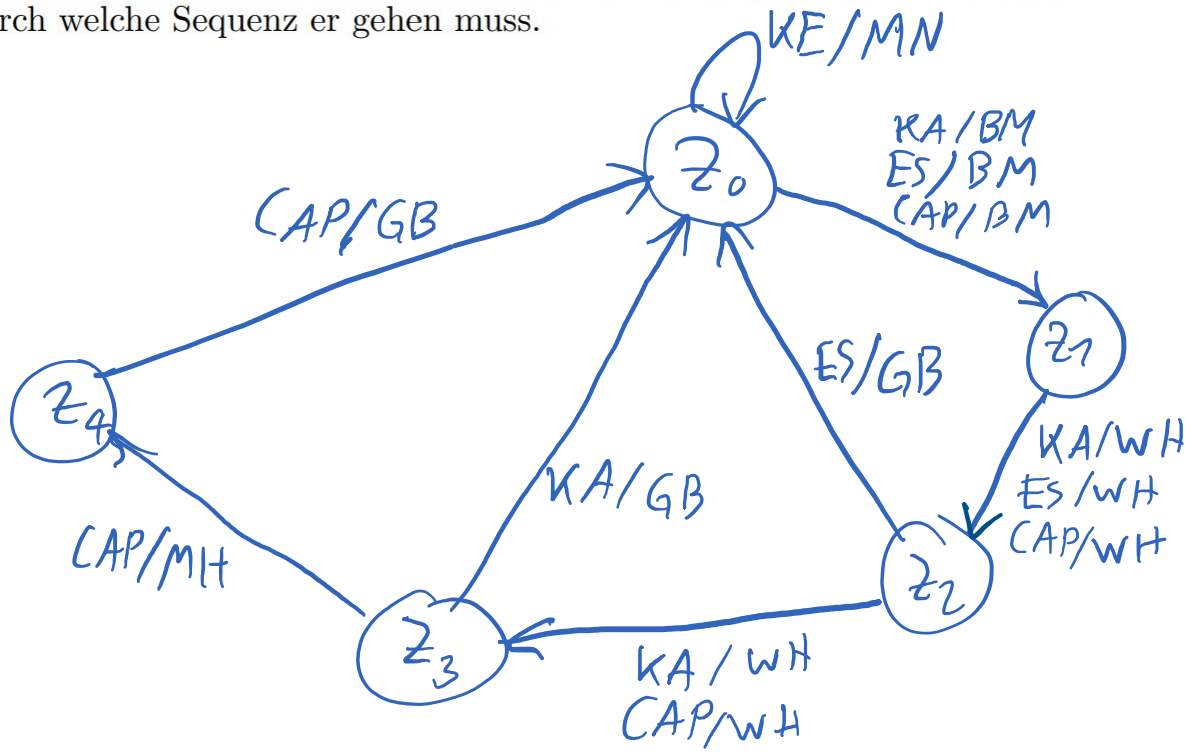
2. Wie viele Bits brauchen Sie, um die Eingangs- und Ausgangsvariablen zu kodieren? (1 Punkt)

Eingänge: 2 Bits, weil 4 Variablen

Ausgänge: 3 Bits, weil 5 Variablen

Prüfungsaufgabe (HS 2020)

3. Zeichnen Sie das Zustandsdiagramm für diese Anlage als **Mealy**-Automat mit der minimalen Anzahl an Zuständen. Definieren Sie N Zustände Z_0 bis Z_{N-1} und beschreiben Sie deren Funktionalität ganz kurz. Als Eingangs- und Ausgangsvariablen benutzen Sie die Abkürzungen aus Aufgabe 1 (BM, CAP, ES, $\cdots$). (5 Punkte)
Hinweis: Nachdem ein Getränk ausgewählt wurde, bleibt das entsprechende Signal (ES, KA oder CAP) bis zum Ende der Vorbereitung dieses Getränks gespeichert. Damit weiss der Automat, durch welche Sequenz er gehen muss.



Z_0 : Ruhezustand
Z_1 : Bohnen werden gemahlen
Z_2 : erste 1dl Wasser erhitzt
Z_3 : zweiten 1dl **Wasser** erhitzt.
Z_4 : Milch wird erhitzt

Prüfungsaufgabe (HS 2020)

4. Wie viele Flipflops benötigen Sie minimal, um diesen Automaten zu realisieren? (1 Punkt)

3 FF, weil 5 Zustände

5. Wie viele Zeilen und Spalten hätte die *komplette* Zustandsfolgetabelle? Begründen Sie. (1 Punkte)

$2^5 = 32$ Zeilen $= 2^{e+m}$

$e + 2m + b = 2 + 2 \cdot 3 + 3 = 11$ Spalten

b: Anzahl Ausgänge

e: Anzahl Eingänge

m: Anzahl Zustandsvariablen = Anzahl Flipflops ≠ Anzahl Zustände

Theorie – Umwandlung

- Moore zu Mealy:

1. Ausgänge von Folgezuständen auf Kanten schreiben.
2. Ausgänge bei Zuständen entfernen.

- Mealy zu Moore: Meist mehr interne Zustände nötig

1. Ausgänge in Knoten schreiben, an denen Kante endet.
2. Knoten mit mehr als einem Ausgang multiplizieren → neu kodieren.
3. Eingehende Kanten entsprechend der Ausgänge auf neue Knoten umhängen.
4. Ausgehende Kanten für alle neue Knoten kopieren.

Diese Umwandlung ist immer möglich, aber meistens werden mehr Zustände benötigt.

Theorie – Umwandlung Zeitverhalten

- Das Zeitverhalten der Ausgänge verändert sich bei der Umwandlung!!

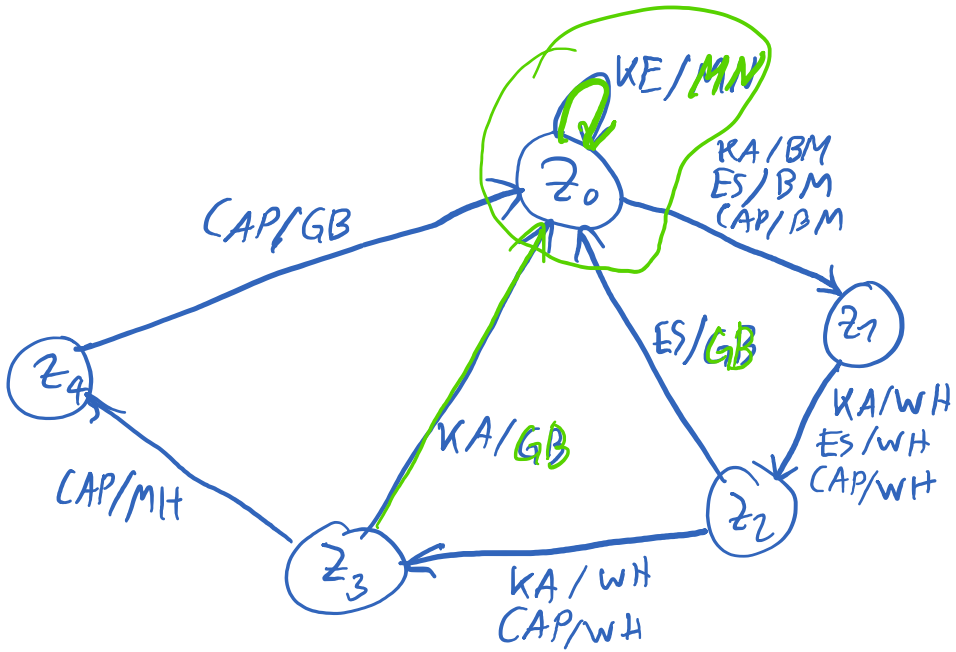
Mealy: Eingangsveränderungen beeinflussen den Ausgang sofort

Moore: Eingangsveränderungen haben erst bei Taktflanke Einfluss (weniger Störungsanfällig)

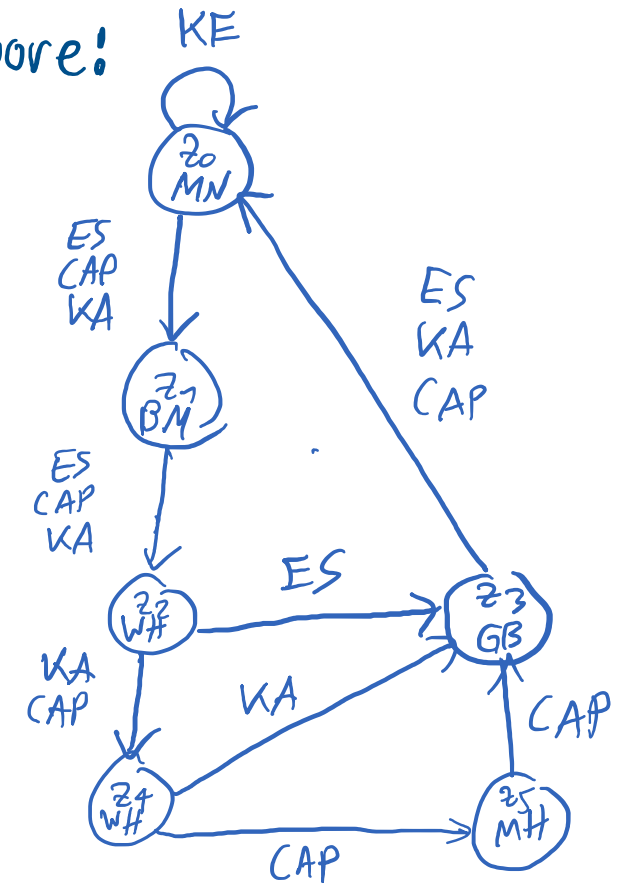
Prüfungsaufgabe (HS 2020) (Fortsetzung von vorher)

6. Zeichnen Sie jetzt das Zustandsdiagramm für diese Anlage als **Moore**-Automat mit der minimalen Anzahl M an Zuständen Z_0 bis Z_{M-1}. Beschreiben Sie jeden Zustand ganz kurz. Es gelten die gleichen Bedingungen wie beim **Mealy**-Automaten. (5 Punkte)

Mealy:



Moore!



Z_0 : Ruhezustand

Z_1 : Bohnen werden gemahlen

Z_2 : 1dl Wasser erhitzt

Z_3 : Getränk ist bereit

Z_4 : 2. 1dl Wasser erhitzt

Z_5 : Milch wird hinzugefügt

Theorie – Don‘t Care Zustände

- Wenn es unbenutzte Zustände im Automaten gibt (don‘t care Zustände), so müssen wir dieses separat betrachten
- Was passiert wenn Automat trotzdem in diesen Zustand kommt?
- Kommt er aus diesem Zustand selber wieder raus
- Wenn nicht: Gleichungen ergänzen (Terme hinzufügen)

Prüfungsaufgabe (HS2020)

Teilaufgabe 2: Analyse des Zustandsdiagramms eines Automaten

Bemerkung: Die Aufgaben in diesem Teil sind unabhängig von den Aufgaben in Teilaufgabe 1.

Ein Automat ist durch das Zustandsdiagramm in Abbildung A2 gegeben. Dabei bezeichnet $\{e_1, e_0\}$ die Eingänge, $\{y\}$ den Ausgang und $\{z_1, z_0\}$ die Zustände.

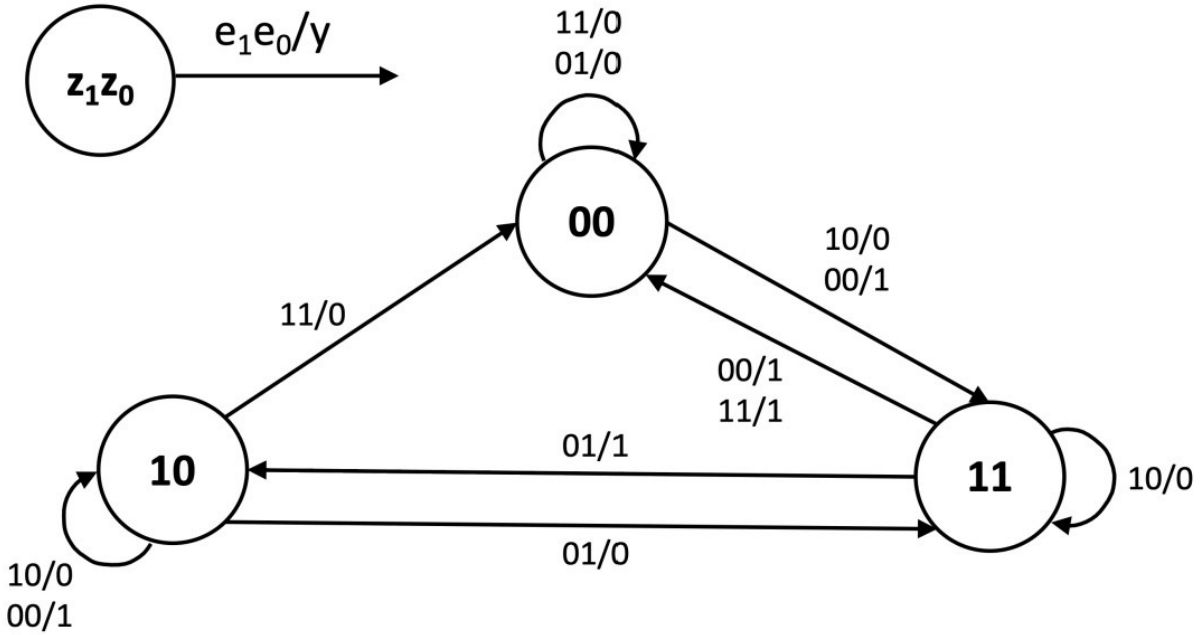


Abbildung A2: Zustandsdiagramm eines Automaten

Prüfungsaufgabe (HS 2020)

Ein Zustand $\{z_1, z_0\}=\{0,1\}$ ist bei diesem Automaten nicht benutzt ("don't care" Zustand).

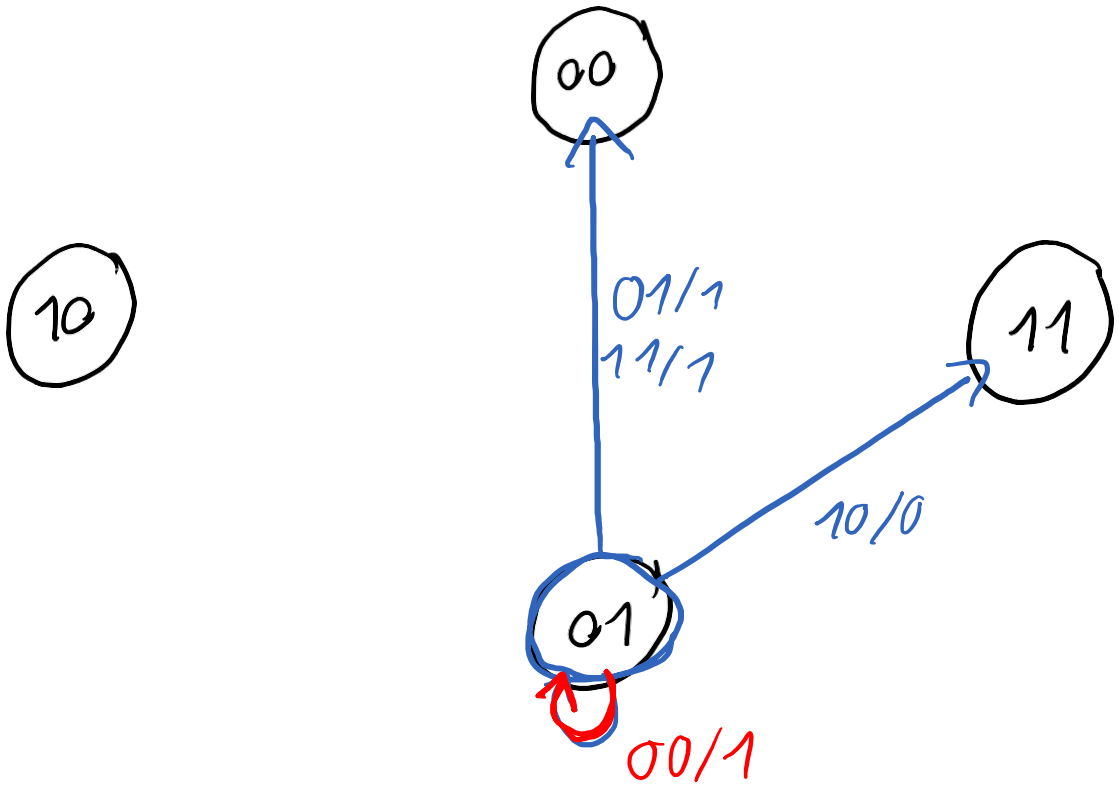
11. Ergänzen Sie das Zustandsdiagram auf dem Lösungsblatt für den Fall, in dem der Automat aus irgendeinem Grund in diesen unbenutzten Zustand gelandet ist. Zeichnen Sie nur die Kanten, die vom $\{0,1\}$-Zustand starten sowie die entsprechenden Eingangs- und Ausgangsvariablen. (1 Punkt)

$z_1 z_0 = 01$

$$
\begin{aligned}
z_{1(n+1)} &= ((\overline{e_0} \wedge \overline{z_0}) \vee (e_1 \wedge \overline{e_0}) \vee (\overline{e_1} \wedge e_0 \wedge z_1))_n \\
z_{0(n+1)} &= ((\overline{e_0} \wedge \overline{z_1}) \vee (e_1 \wedge \overline{e_0} \wedge z_0) \vee (\overline{e_1} \wedge e_0 \wedge z_1 \wedge \overline{z_0}))_n \\
y_n &= ((\overline{e_1} \wedge \overline{e_0}) \vee (e_0 \wedge z_0))_n
\end{aligned}
$$

$e_0 e_1 \in \{(0,0),(0,1),(1,0),(1,1)\}$

→ in Gleichungen einsetzen



Prüfungsaufgabe (HS 2020) (Fortsetzung von vorher)

12. Wie muss die Gleichung für $z_{1(n+1)}$ minimal modifiziert werden, damit der Automat, wenn er zuerst im $\{0,1\}$-Zustand ist, unabhängig von der Eingangskombination, immer in einen der benutzten Zustände wechselt? Die Ergebnisse der Eingangskombinationen, die schon zu einem Zustandswechsel führen, sollten nicht modifiziert werden. Verwenden Sie dafür das Karnaugh-Diagramm auf dem Lösungsblatt, wo Sie zuerst die nötigen Päckchen bilden werden. Geben Sie nachher die neue Gleichung für $z_{1(n+1)}$. (2 Punkte)

Vorher:

z_{1n+1}

z_1z_0 \ e_1e_0	00	01	11	10
00	1	0	0	1
01	X	X	X	X
11	a	1	0	1
10	1	1	0	1

Was müssen wir ändern?

damit der Automat, falls er im Zustand 01 ist, bei Eingang den Zustand Wechselt.

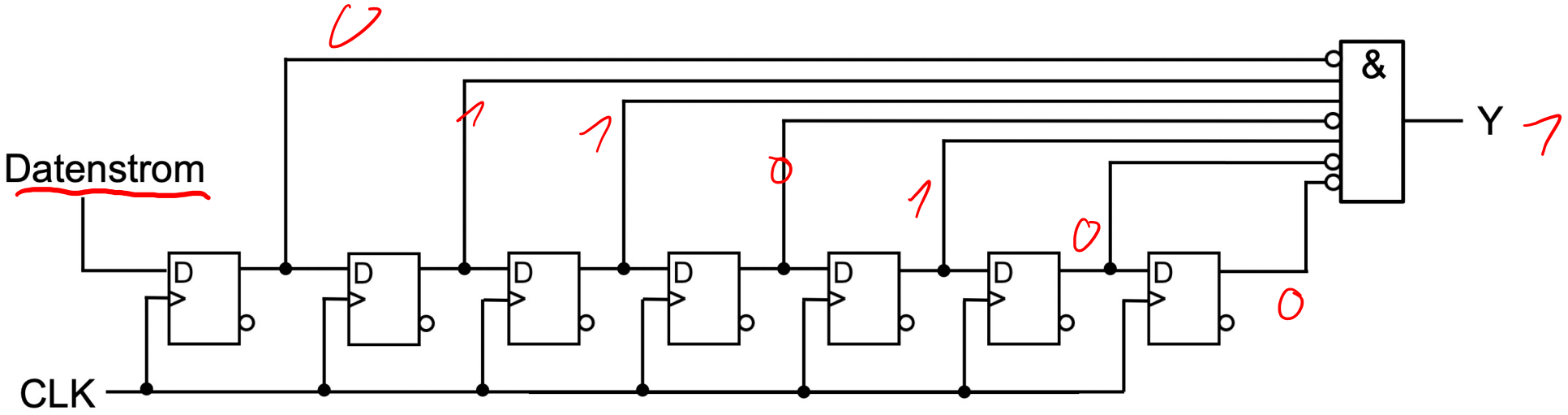
z_{1n+1}

z_1z_0 \ e_1e_0	00	01	11	10
00	1	0	0	1
01	1	X	X	X
11	a	1	0	1
10	1	1	0	1

$z_{1n+1} = ((\bar{e}_0 \wedge \bar{z}_0) \vee (e_1 \wedge \bar{e}_0) \vee (\bar{e}_1 \wedge e_0 \wedge z_1) \vee (\bar{e}_0 \wedge \bar{z}_1))_n$

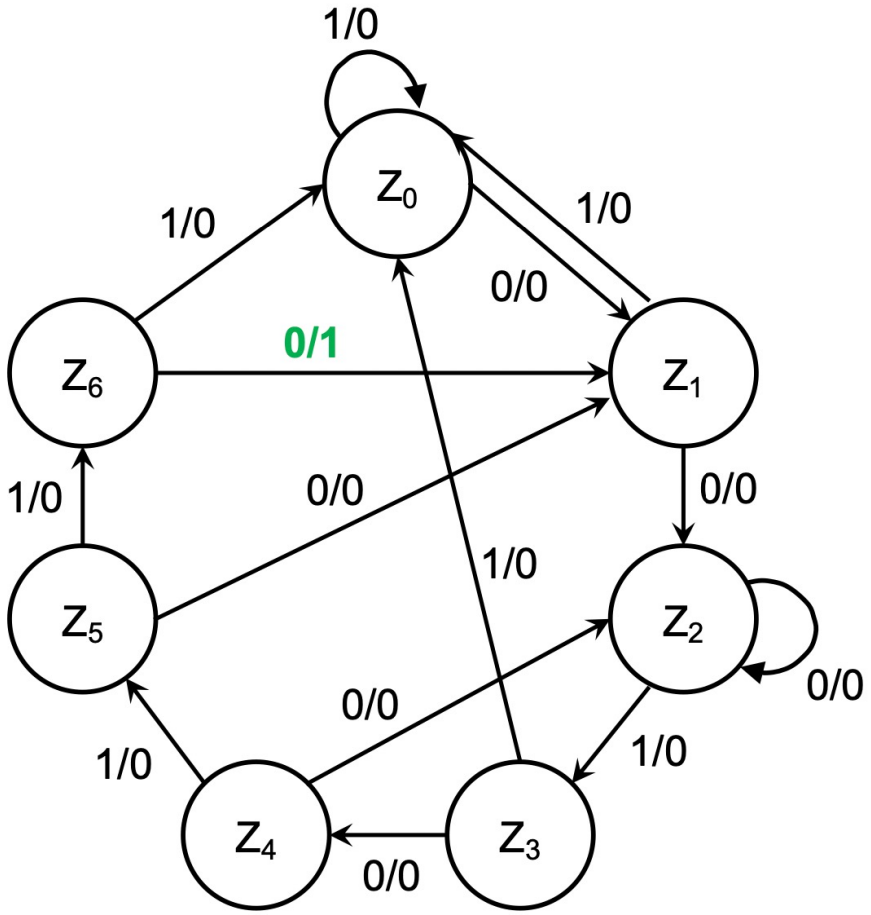
Theorie – Folgedetektor

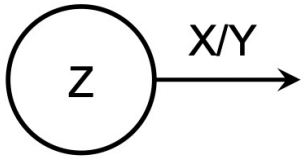
- Ziel: Entwurf eines Automaten, der eine bestimmte Sequenz erkennt und in diesem Fall Y = 1 ausgibt. Ansonsten Y = 0. z.B. 0010110
- 1. Möglichkeit: (Einfach zu entwerfen, aber nicht minimale Anzahl FF)



Theorie – Folgedetektor

- 2. Möglichkeit: Mealy Automat





Z_0: Anfangszustand, nichts detektiert

Z_1: 0 detektiert

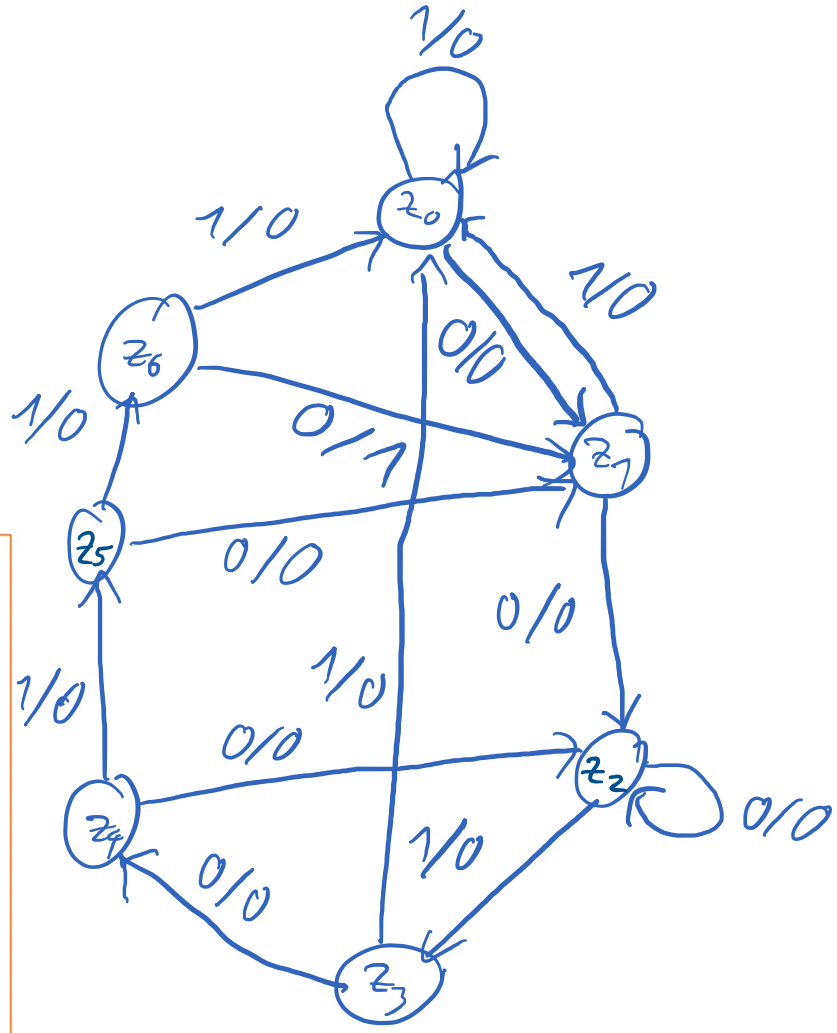
Z_2: 00 detektiert

Z_3: 001 detektiert

Z_4: 0010 detektiert

Z_5: 00101 detektiert

Z_6: 001011 detektiert

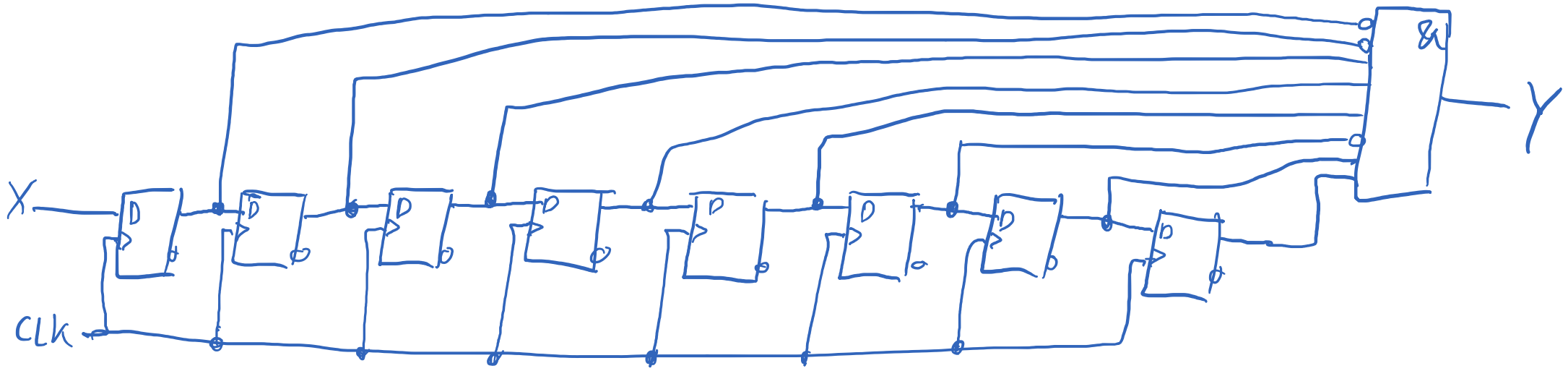


Prüfungsaufgabe (HS 2020)

Teilaufgabe 3: Folgedetektor

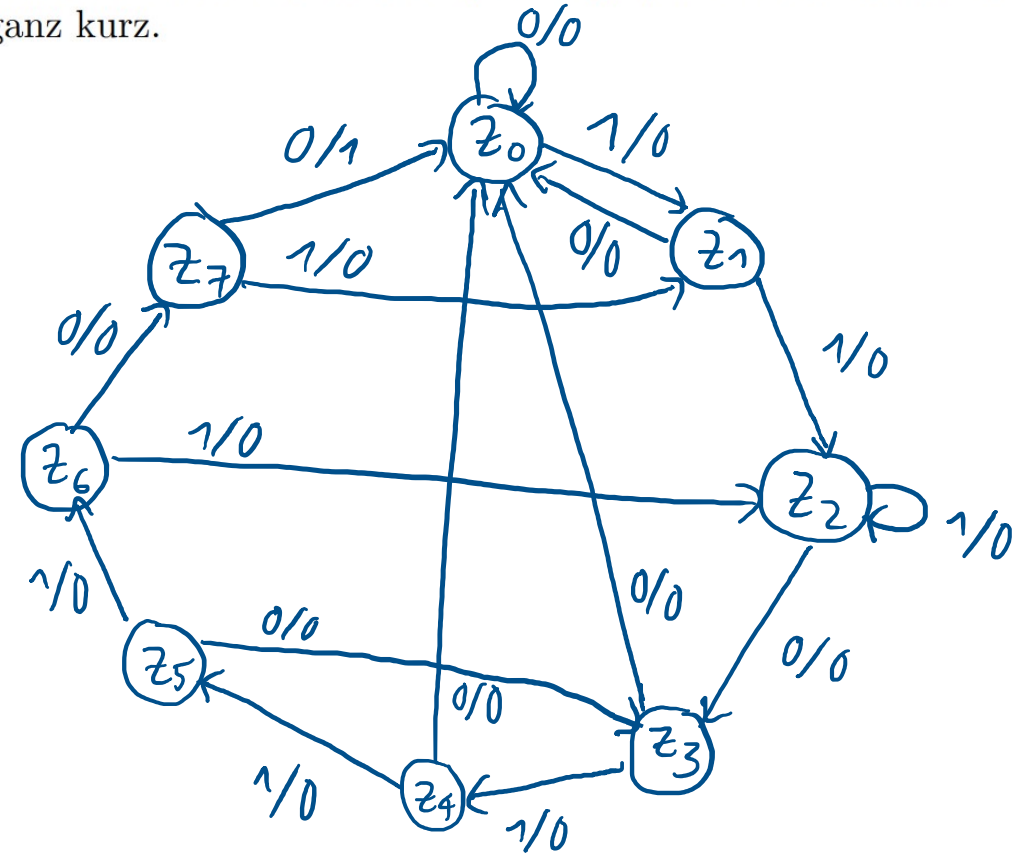
Das Ziel dieser Teilaufgabe ist, einen Folgedetektor zu entwerfen, der eine 1 als Ausgang ausgibt, wenn er die folgende Bitsequenz anerkannt hat: 11011100.

7. Zeichnen Sie auf dem Lösungsblatt den Schaltplan dieses Folgedetektors, wenn Sie eine beliebige Anzahl an D-Flipflops und ein einziges UND Gatter mit mehreren Eingängen zur Verfügung haben. Diese Eingänge können invertiert werden aber müssen immer mit dem Ausgang eines D-Flipflops verbunden sein. Die Bits werden ein nach dem anderen in den Folgedetektor eingelesen. Verwenden Sie die Variablen X und Y, um den Eingang und Ausgang des Detektors zu beschreiben. (4 Punkte)



Prüfungsaufgabe (HS 2020)

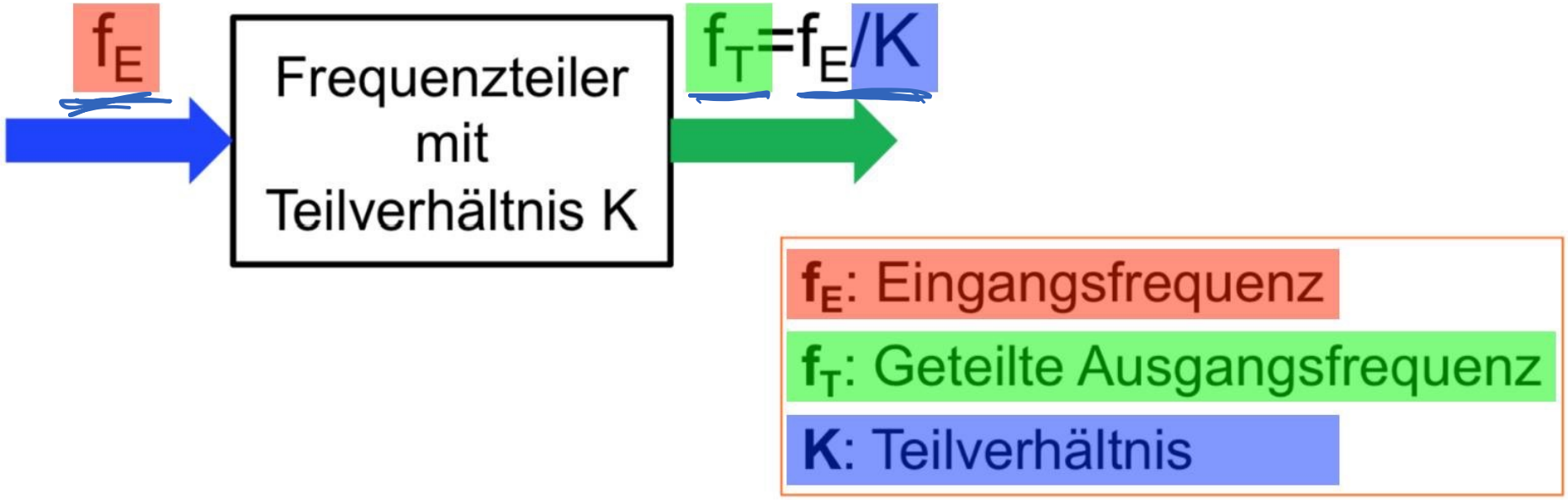
8. Dieser Folgedetektor muss jetzt als **Mealy**-Automat entworfen werden. Zeichnen Sie das entsprechende Zustandsdiagramm mit der minimalen Anzahl an Zuständen auf dem Lösungsblatt. Definieren Sie N Zustände Z_0 bis Z_{N-1} und beschreiben Sie deren Funktionalität ganz kurz. (4 Punkte)



z_n : n Bit(s) wurden schon anerkannt

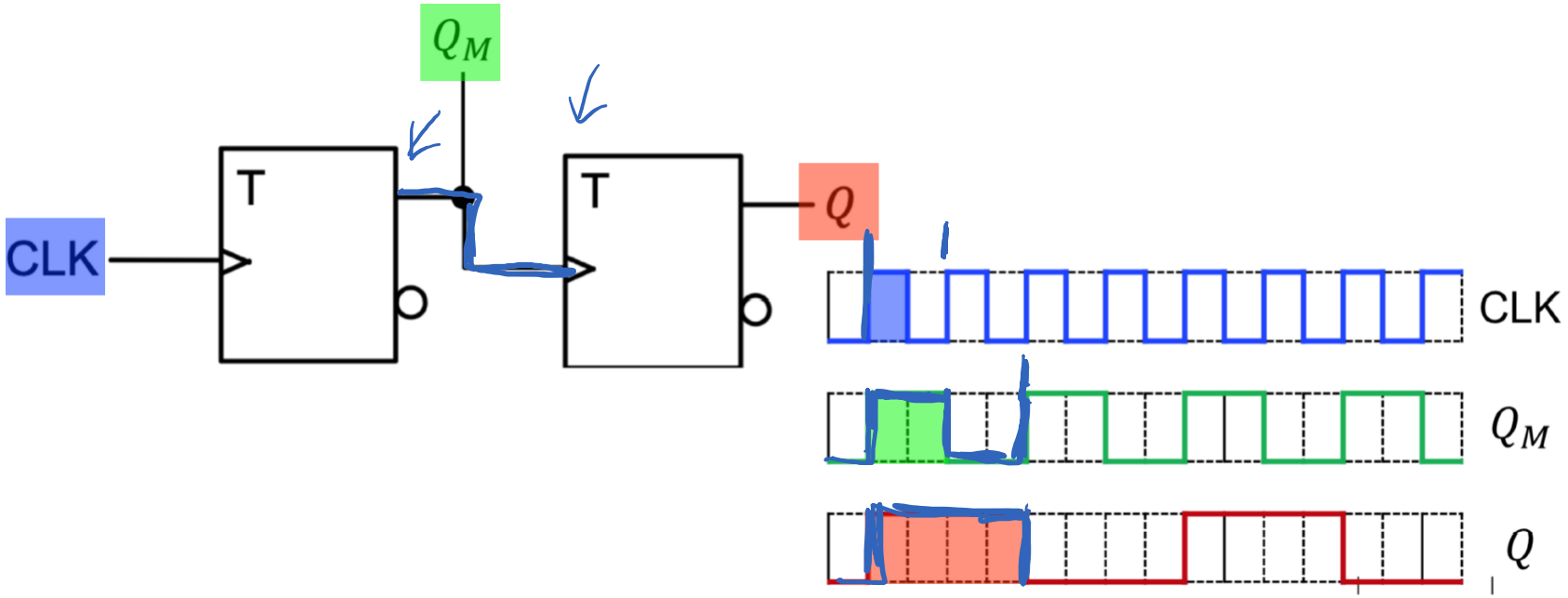
Theorie – Frequenzteiler

- Um Frequenzen zu reduzieren
- Periode T eines Taktsignals verlängern



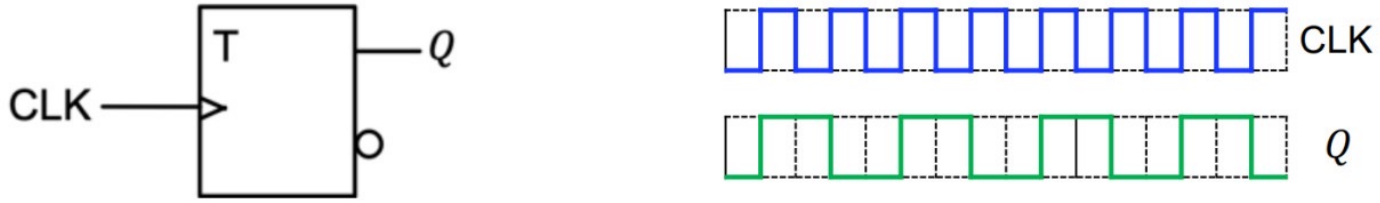
Theorie – Frequenzteiler

- Mithilfe von Flipflops realisieren
- **T-FF** oder **JK-FF** mit J=K=1
- n Flipflops führen zu einer Reduktion von 2^n

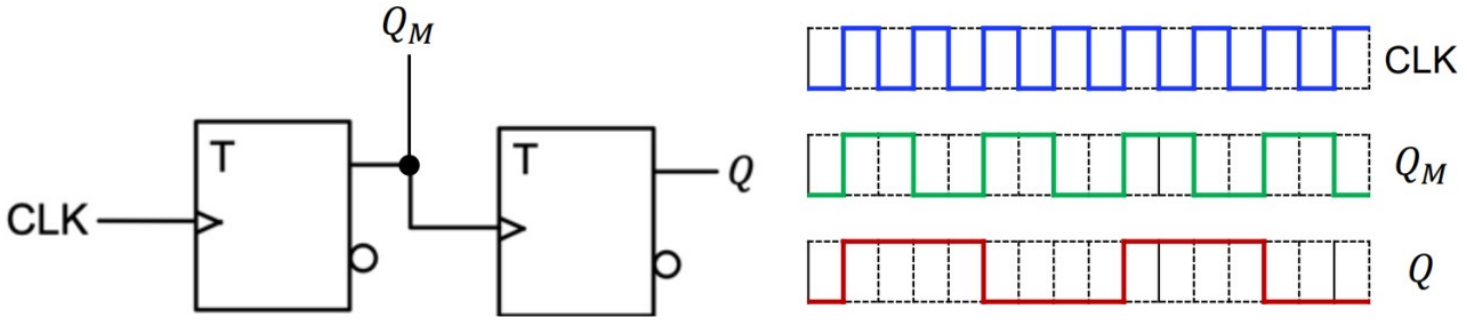


Theorie – Frequenzteiler

Frequenzreduktion um einen Faktor 2

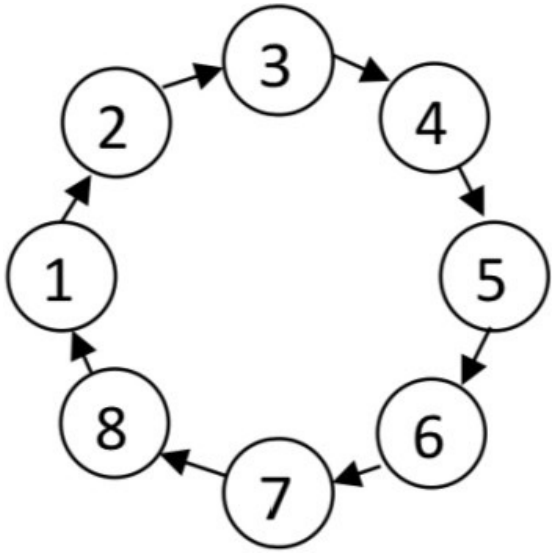


Frequenzreduktion um einen Faktor 4



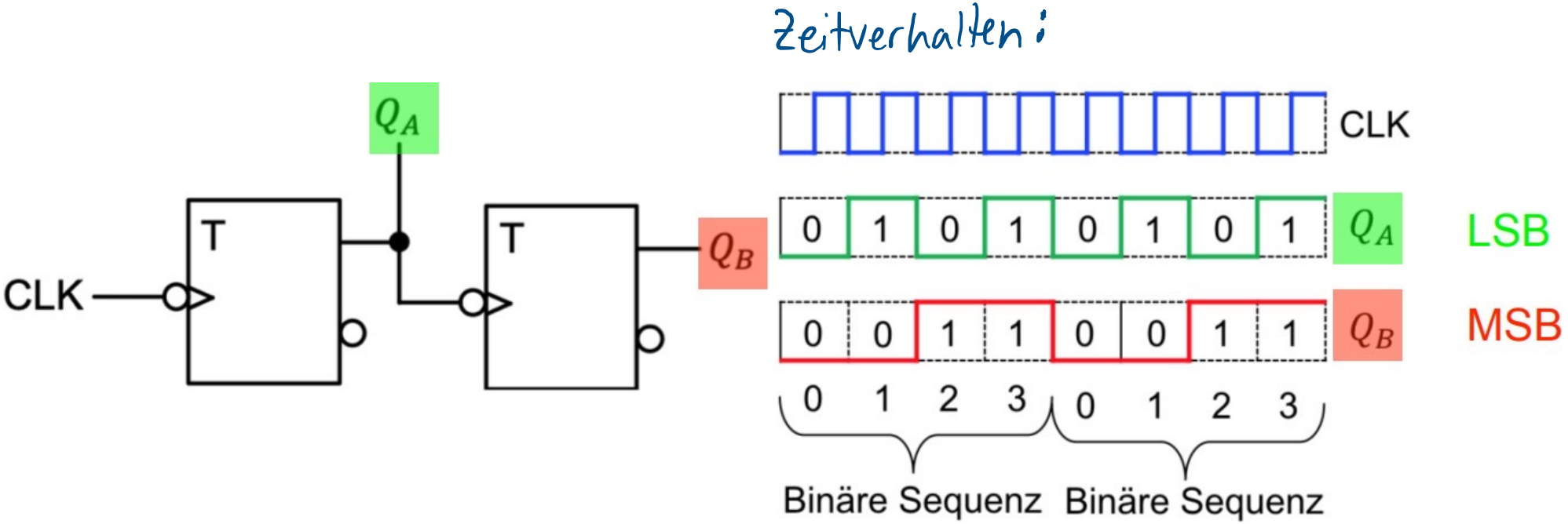
Theorie – Zähler

- Zählen bis zu einer bestimmten Zahl und fangen dann wieder von vorne an
- Vorwärts zählen = +1 in jedem Schritt
- Rückwärts zählen = -1 in jedem Schritt



Theorie – Asynchronzähler bzw. Dualzähler

- Kaskadierung von T-Flipflops
- Mit n Flipflops kann man bis 2^n - 1 zählen



Theorie – Asynchronzähler, Dualzähler

- Vorwärtszähler: FF mit fallender aktiven Taktflanke
- Rückwärtszähler: FF mit steigender aktiven Taktflanke oder einfach statt Q die bitweise invertierten Werte !Q

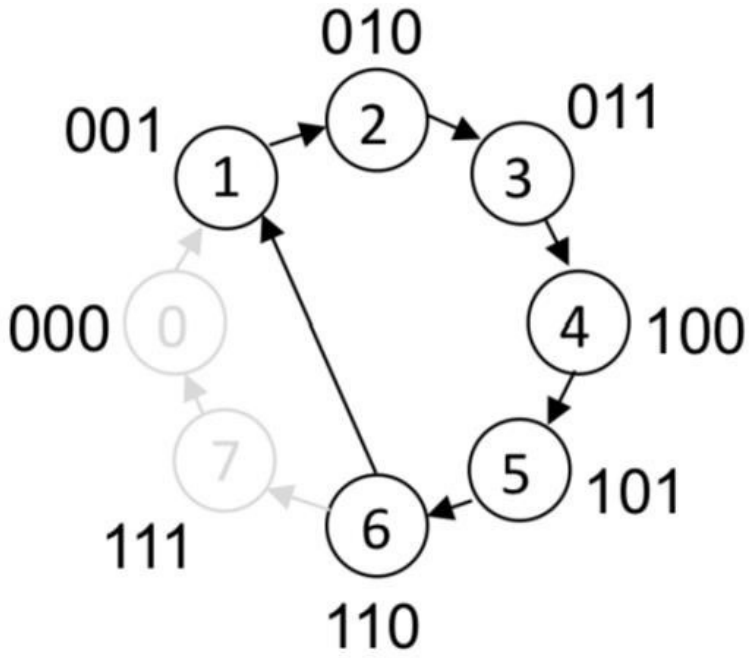
Probleme:

- Verzögerungen der Zustandsänderungen kumulieren sich entlang er Schaltung
- Zeitverzögerung ist bei jedem Zustand anders
- Maximale Taktfrequenz damit jeder mögliche Zustand bei n FF (kurz) auftritt:

$$f_{max} = \frac{1}{\sum_{i=1}^{n} t_{pd,i}}$$

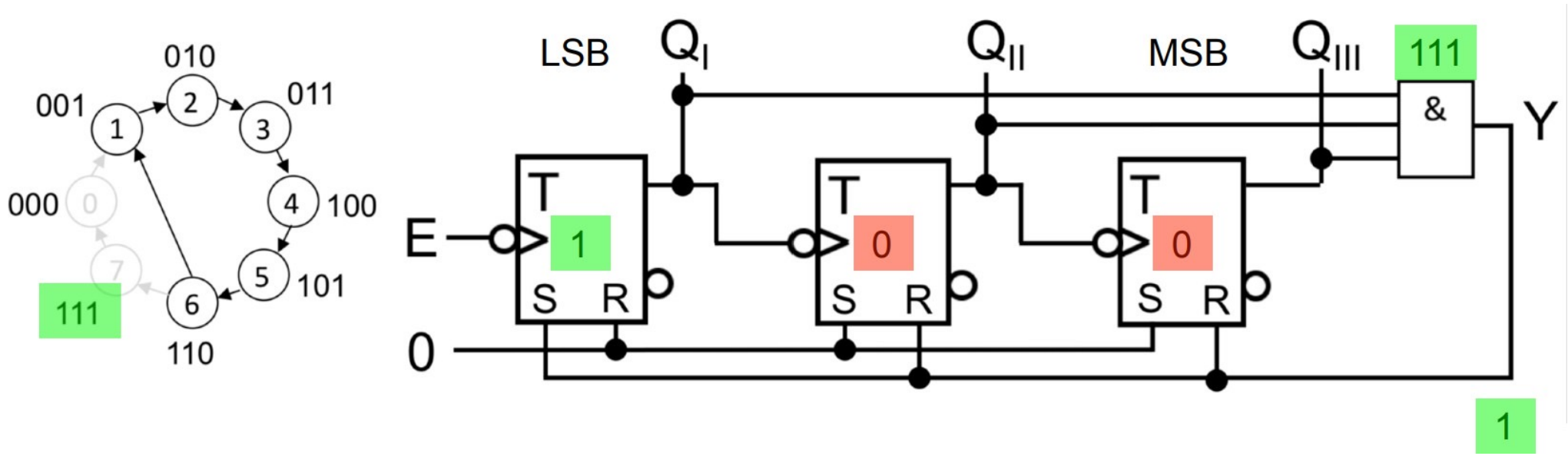
Theorie – Modulo-n Zähler

- Zählt bis n und springt dann zurück auf einen vorgegebenen Startzustand
- Umsetzbar als Asynchronzähler und Synchronzähler

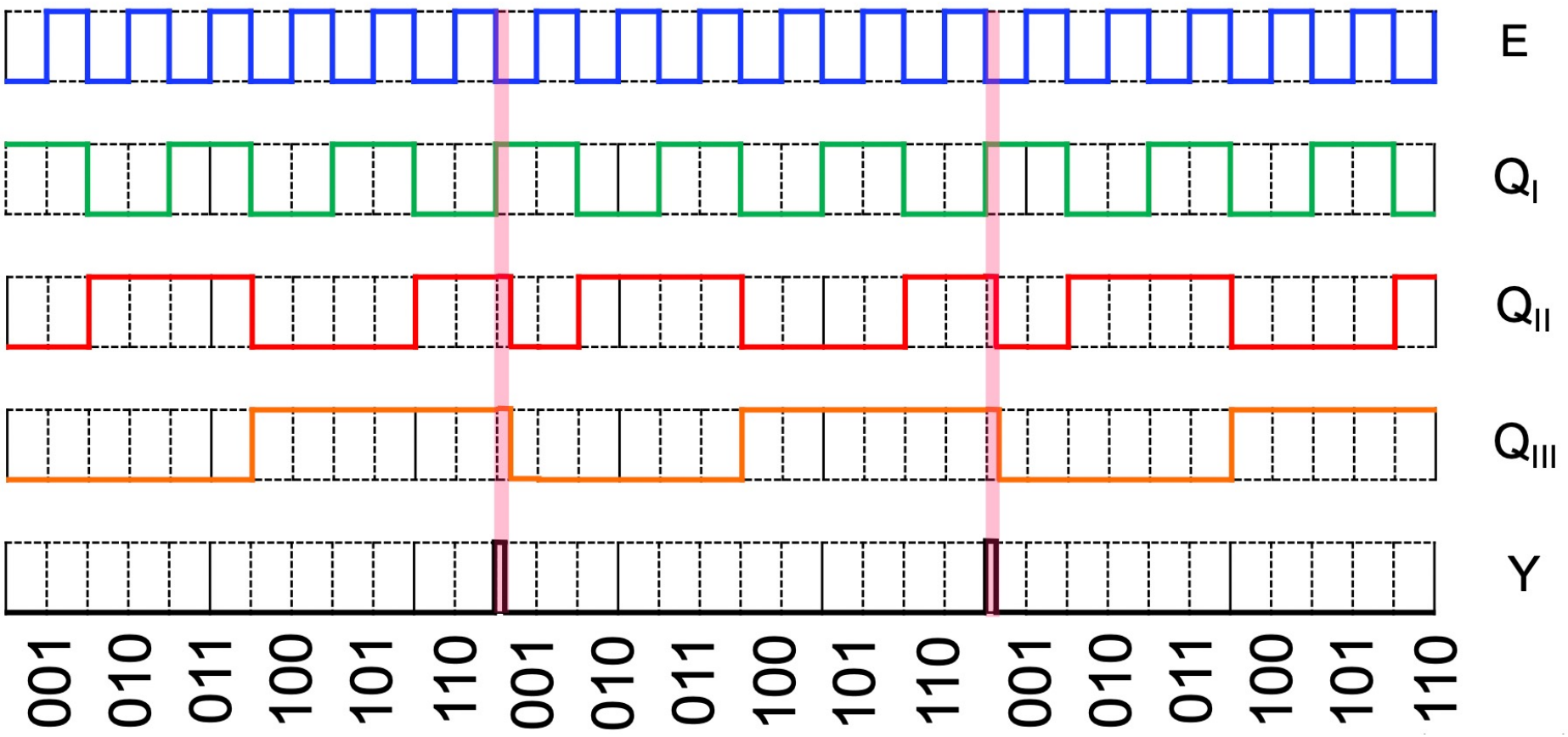


Theorie – Modulo-n Asynchronzähler

- Umsetzung mit Dualzähler
- Zustand n+1 mit kombinatorischen Schaltung detektieren
- Mit diesem Ergebnis Zähler zum gewünschten Zustand zurücksetzen
- Mithilfe von Set und Reset Eingängen

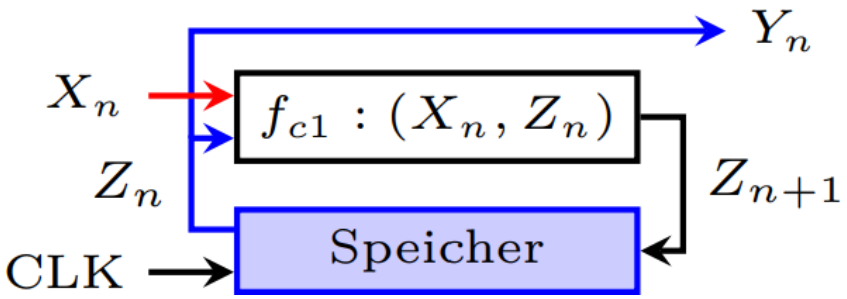


Theorie – Zeitverhalten Modulo-6 Asynchronzähler



Theorie – Modulo-n Synchronzähler

- Alle FF haben dasselbe Taktsignal (schalten gleichzeitig)
- Sind Medwedjew-Automaten (Ausgänge entsprechen den Zuständen)

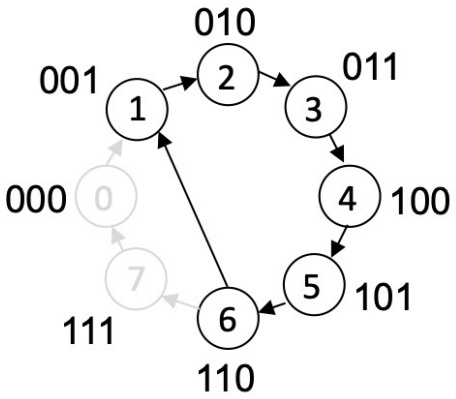


- Entwerfen wie Automaten

Zustandsgraph ⟶ Folgezustandstabelle ⟶ Karnaugh Diagramme ⟶ Schaltplan

Theorie – Modulo-6 Synchronzähler

Zustandsgraph → Folgezustandstabelle → Karnaugh Diagramme → Schaltplan



Zustandsgraph

Q_{3n}	Q_{2n}	Q_{1n}	Q_{3n+1}	Q_{2n+1}	Q_{1n+1}
0	0	0	X	X	X
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	X	X	X

Folgezustandstabelle

Theorie – Modulo-6 Synchronzähler

Zustandsgraph → Folgezustandstabelle → Karnaugh Diagramme → Schaltplan

Q_{3n+1}

Q_1 \ Q_3Q_2	00	01	11	10
0	X	0	0	1
1	0	1	X	1

Q_{2n+1}

Q_1 \ Q_3Q_2	00	01	11	10
0	X	1	0	0
1	1	0	X	1

$$Q_{3n+1} = (Q_{2n} \wedge Q_{1n}) \vee (Q_{3n} \wedge \overline{Q_{2n}})$$

$$Q_{2n+1} = (\overline{Q_{3n}} \wedge \overline{Q_{1n}}) \vee (\overline{Q_{2n}} \wedge Q_{1n})$$

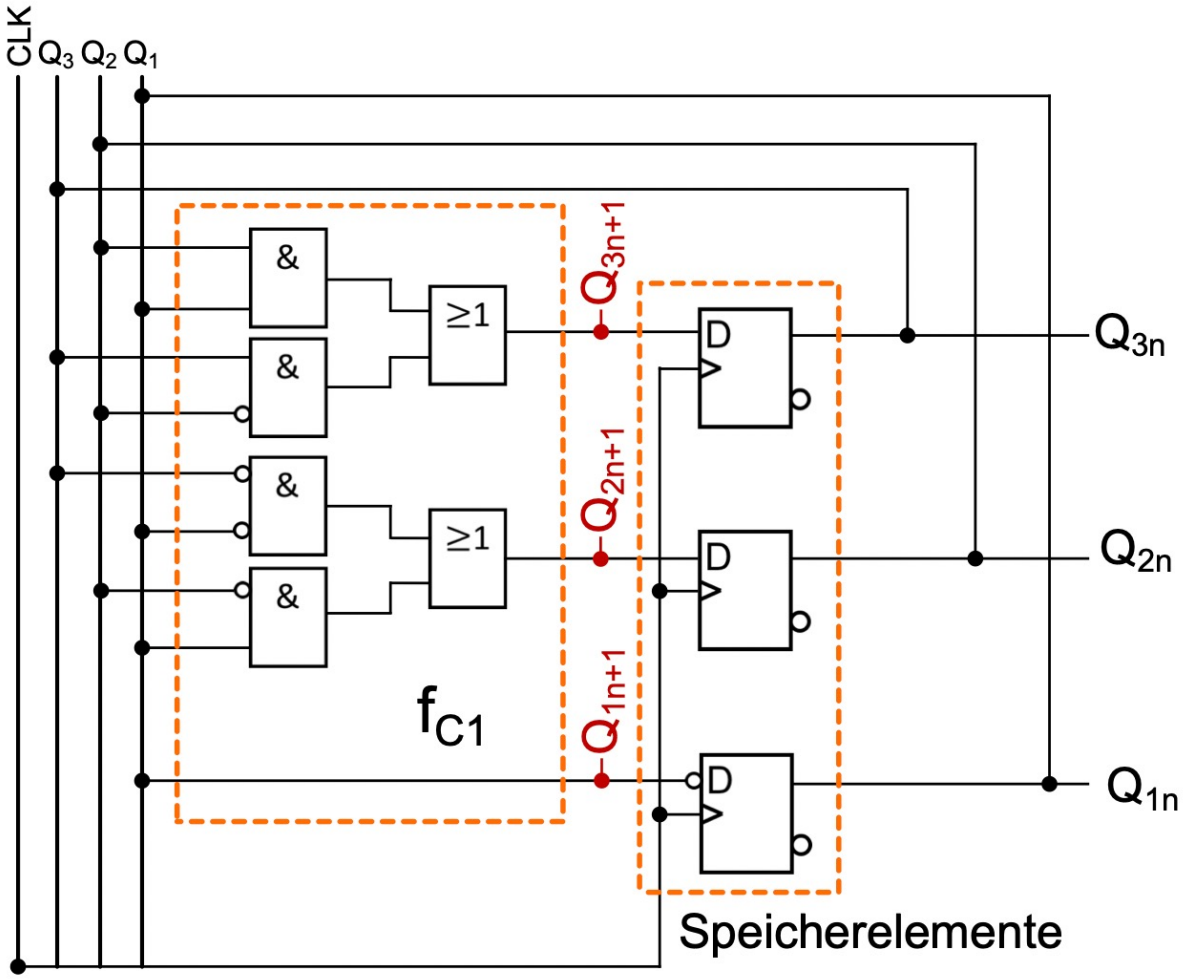
Q_{1n+1}

Q_1 \ Q_3Q_2	00	01	11	10
0	X	1	1	1
1	0	0	X	0

$$Q_{1n+1} = \overline{Q_{1n}}$$

Theorie – Modulo-6 Synchronzähler

Zustandsgraph → Folgezustandstabelle → Karnaugh Diagramme → Schaltplan



Theorie – Karnaughdiagramme & Flipflops

- D-FF:

➔ Karnaughdiagramme „normal“

- JK-FF: $Q_{in+1} = ((J_i \wedge \overline{Q_i}) \vee (\overline{K_i} \wedge Q_i))_n$

➔ Karnaughdiagramme mit Felder für Q und !Q

Q_{1n+1}

Q_1 \ Q_3Q_2	00	01	11	10
0				
1				

Q_1 \ Q_3Q_2	00	01	11	10
0				
1				

Q_{2n+1}

Q_{3n+1}

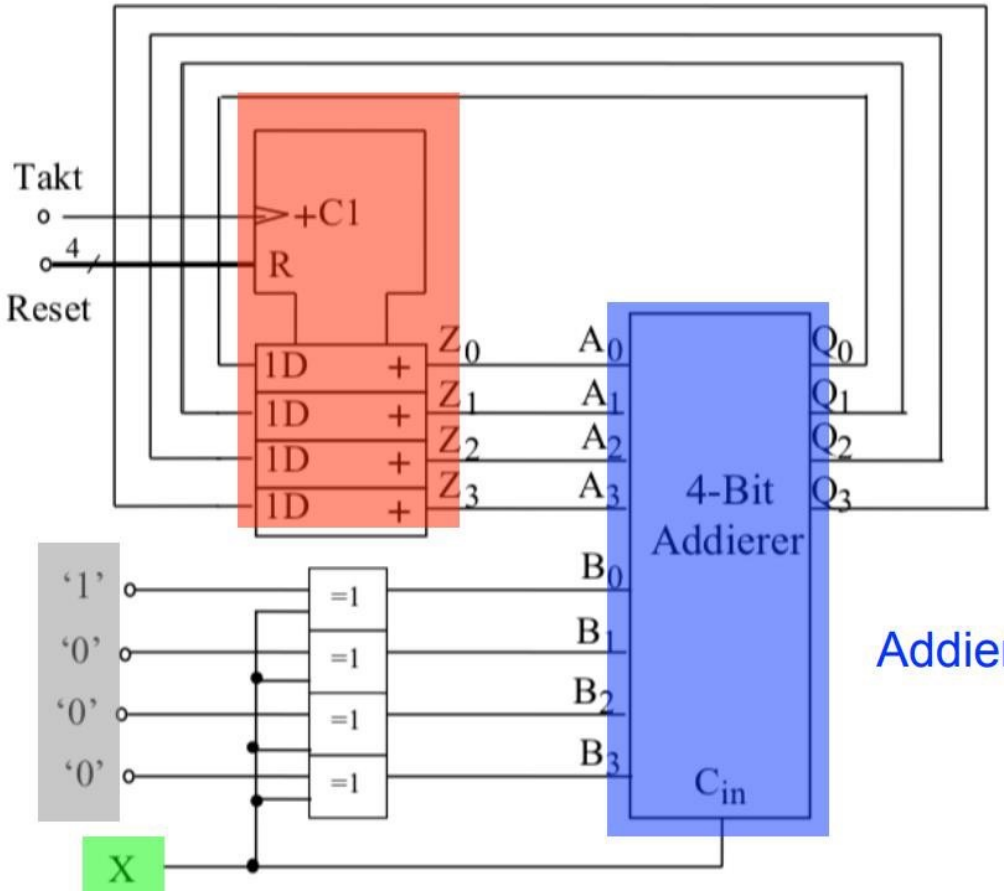
Q_1 \ Q_3Q_2	00	01	11	10
0	0	1	1	0
1	0	0	X	X

Theorie – Vorwärts-Rückwärtszähler (= Reversible Zähler)

- Ziel: Entwurf eines Zählers mit zwei möglichen Bewegungsrichtungen (+1/-1)
- D-FF mit Addierer kombinieren vgl. Addierer/Subtrahierer Übung 6

- X = 0: Addition
- X = 1: Subtraktion

4 D-Flipflops mit gleichem CLK



1 wird addiert (subtrahiert)

Addierer

Prüfungsaufgabe (FS 2020)

Teilaufgabe 3: Asynchronzähler mit T-Flipflops

Das Ziel dieser Teilaufgabe ist, einen asynchronen Binärzähler zu entwerfen, der in der Regel von 0 bis 15 zählt ausser wenn ein Eingangssignal **IN**=1 ist. In diesem Fall muss er bei 5 neu starten, sobald er 12 erreicht. Alle Zahlen sind als Dualzahlen kodiert. Ihnen stehen 1 T-Flipflop mit einem asynchronen Setz (S) Eingang, 1 T-Flipflop mit einem asynchronen Rücksetz (R) Eingang sowie 4 normale T-Flipflops ohne zusätzliche asynchrone Eingänge zur Verfügung. Diese Bauelemente sind in Abbildung D3 gezeigt.

Beim T-Flipflop mit dem asynchronen Setz Eingang wird der Ausgang Q sofort 1 wenn S=1, unabhängig vom Taktsignal. Beim T-Flipflop mit dem asynchronen Rücksetz Eingang wird der Ausgang Q sofort 0 wenn R=1, auch unabhängig vom Taktsignal. Wenn der zu entwerfende Zähler 12 erreicht und **IN**=1, dann muss er sofort bei 5 neu anfangen. Sonst (**IN**=0) zählt er von 0 bis 15 und fängt dann wieder bei 0 an.

Wenn Sie N Flipflops benötigen, um diesen Zähler aufzubauen, dann sollten die Flipflop Ausgänge Q_0 bis Q_{N-1} benannt werden, wobei Q_0 dem 'least significant bit' entspricht und Q_{N-1} dem 'most significant bit'.

Prüfungsaufgabe (FS 2020)

6. Wie viele T Flipflops sind nötig, um diesen Zähler zu realisieren? (1 Punkt)

4 FF, da 16 Zustände

7. Wie viele nicht verwendete Zustände ergeben sich daraus? Begründen Sie Ihre Aussage. (1 Punkt)

0 Zustände sind unbenutzt. (0 bis 15 zählen braucht 16 Zustände)

8. Was ist die logische Bedingung, damit der Zähler in einen anderen Zustand wechselt, wenn er 12 erreicht? Drücken Sie diese Bedingung als Funktion von den Q-Ausgängen und vom **IN** Signal aus. (1 Punkt)

Q_0=LSB, Q_3=MSB

$12_{10} = 1100_2$

$Q_2 \doteq Q_3 \doteq 1$ $Q_1 = Q_0 \doteq 0$

$$IN \wedge \overline{Q_0} \wedge \overline{Q_1} \wedge Q_2 \wedge Q_3 = 1$$

Prüfungsaufgabe (FS 2020)

9. Zeichnen Sie diesen Zähler auf dem Lösungsblatt. Benützen Sie die T-Flipflops aus Abbildung D3 sowie NICHT, UND und ODER Gatter mit einer beliebigen Anzahl an Eingängen. (4 Punkte)

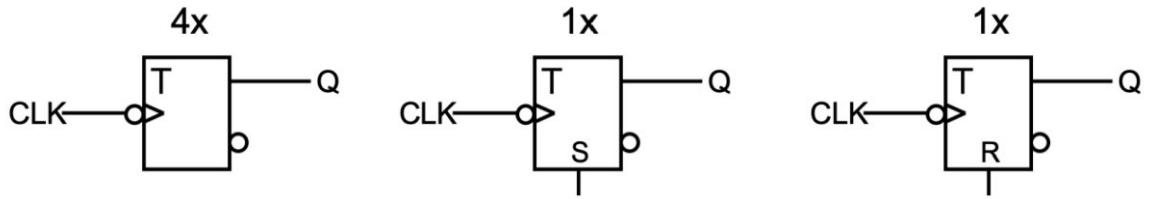
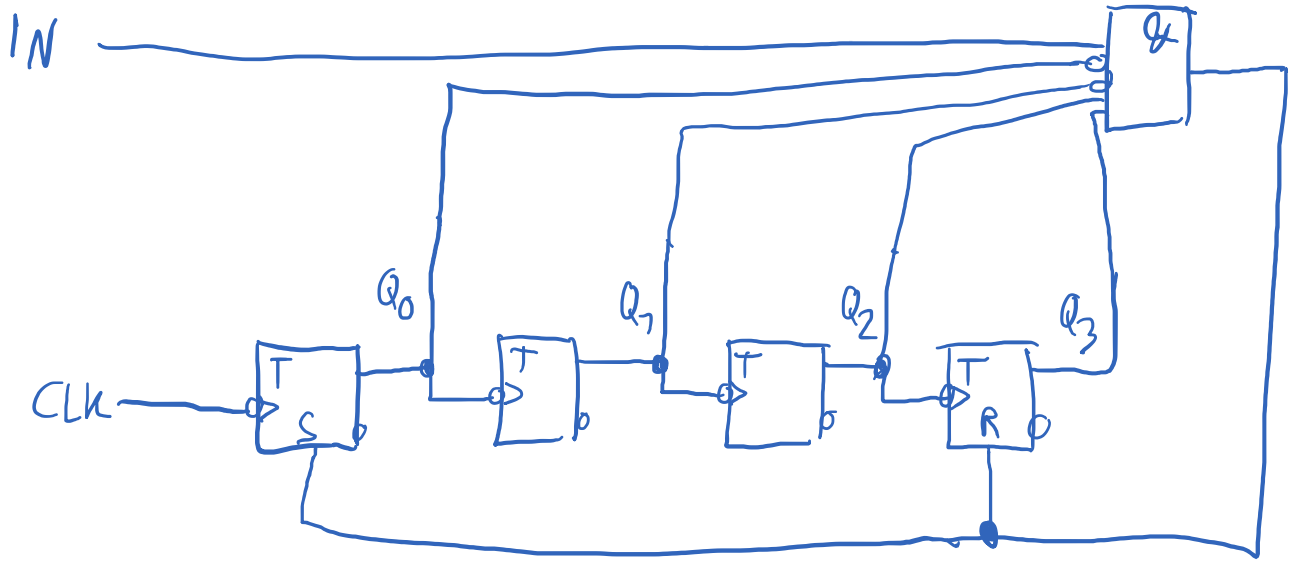


Abbildung D3: (links) Normale T-Flipflops. (mitte) T-Flipflop mit einem asynchronen Setz (S) Eingang. (rechts) T-Flipflop mit einem asynchronen Rücksetz (R) Eingang.



$12_{10} = 1100_2$
reset ↓ ↓ set
$5_{10} = 0101_2$