Digitaltechnik
Übung 10

Matteo Dietz
mdietz@student.ethz.ch

Polybox

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

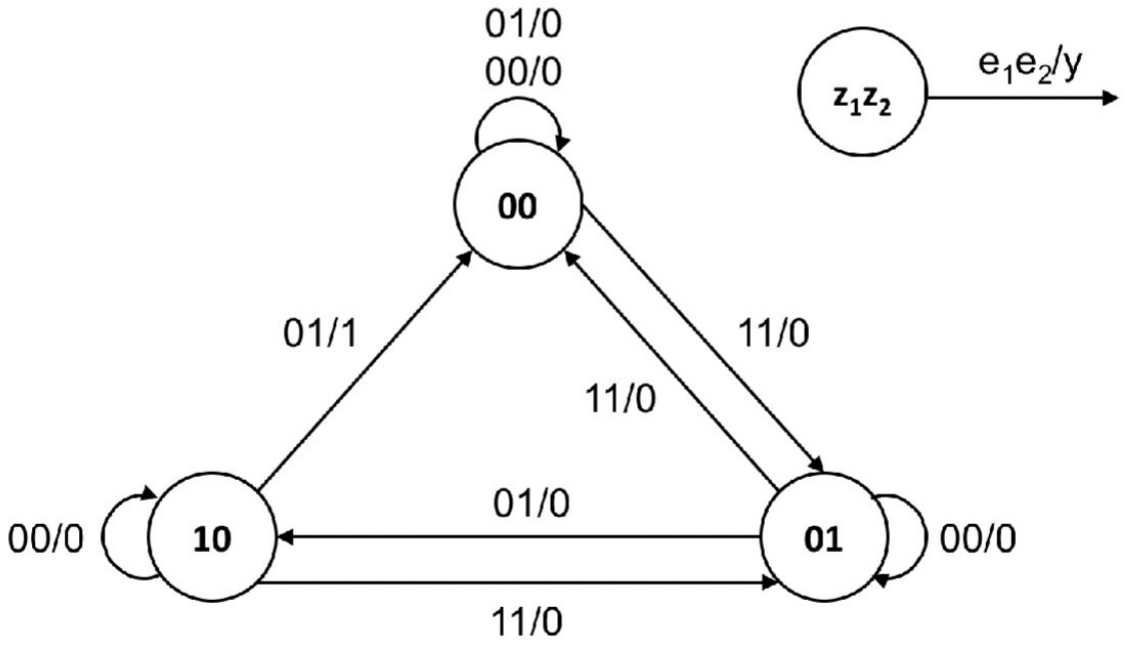
Ablauf:

- Besprechung Zwischenprüfung 3
- Speicher
- Tipps für die Vorbereitung & Prüfung
- Kahoot

Test 3

A.1) Automatentyp: Mealy *(0.5 Punkte)*
Begründung: Ausgang vom aktuellen Zustand und von Eingängen abhängig. *(0.5 Punkte)*

A.2)



e_1	e_2	z_{1n}	z_{2n}	z_{1n+1}	z_{2n+1}	y
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	0	1	1	X	X	X
0	1	0	0	0	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	X	X	X
1	0	0	0	X	X	X
1	0	0	1	X	X	X
1	0	1	0	X	X	X
1	0	1	1	X	X	X
1	1	0	0	0	1	0
1	1	0	1	0	0	0
1	1	1	0	0	1	0
1	1	1	1	X	X	X

Test 3

- Bildet grösstmögliche Päckchen in den Karnaugh Diagrammen!
- Vergesst Subscript n nicht bei den Gleichungen!

z_{1n+1}

z_1z_2 \ e_1e_2	00	01	11	10
00	0	0	0	X
01	0	1	0	X
11	X	X	X	X
10	1	0	0	X

z_{2n+1}

z_1z_2 \ e_1e_2	00	01	11	10
00	0	0	1	X
01	1	0	0	X
11	X	X	X	X
10	0	0	1	X

y_n

z_1z_2 \ e_1e_2	00	01	11	10
00	0	0	0	X
01	0	0	0	X
11	X	X	X	X
10	0	1	0	X

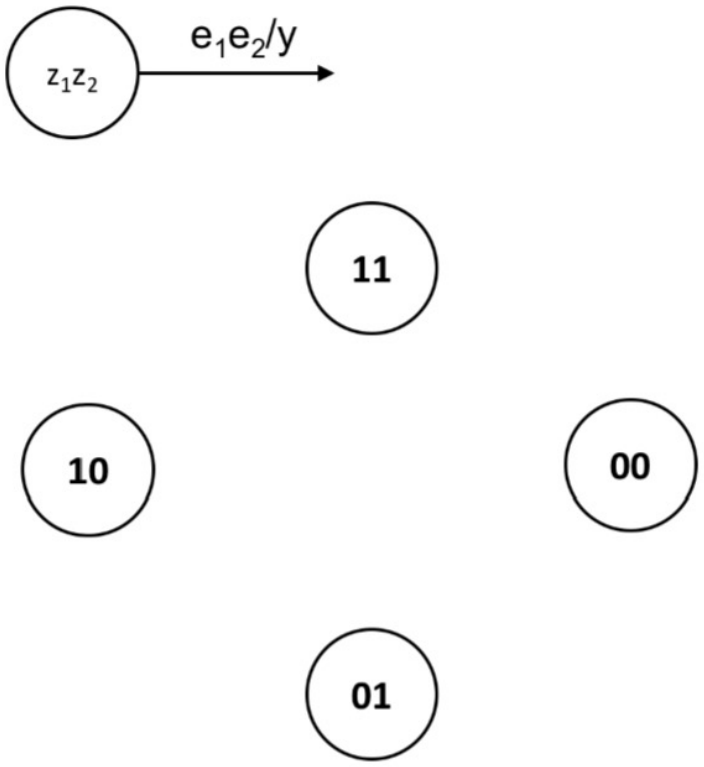
$$z_{1(n+1)} = ((\overline{e_2} \wedge z_1) \vee (\overline{e_1} \wedge e_2 \wedge z_2))_n$$

$$z_{2(n+1)} = ((\overline{e_2} \wedge z_2) \vee (e_1 \wedge \overline{z_2}))_n$$

$$y_n = (\overline{e_1} \wedge e_2 \wedge z_1)_n$$

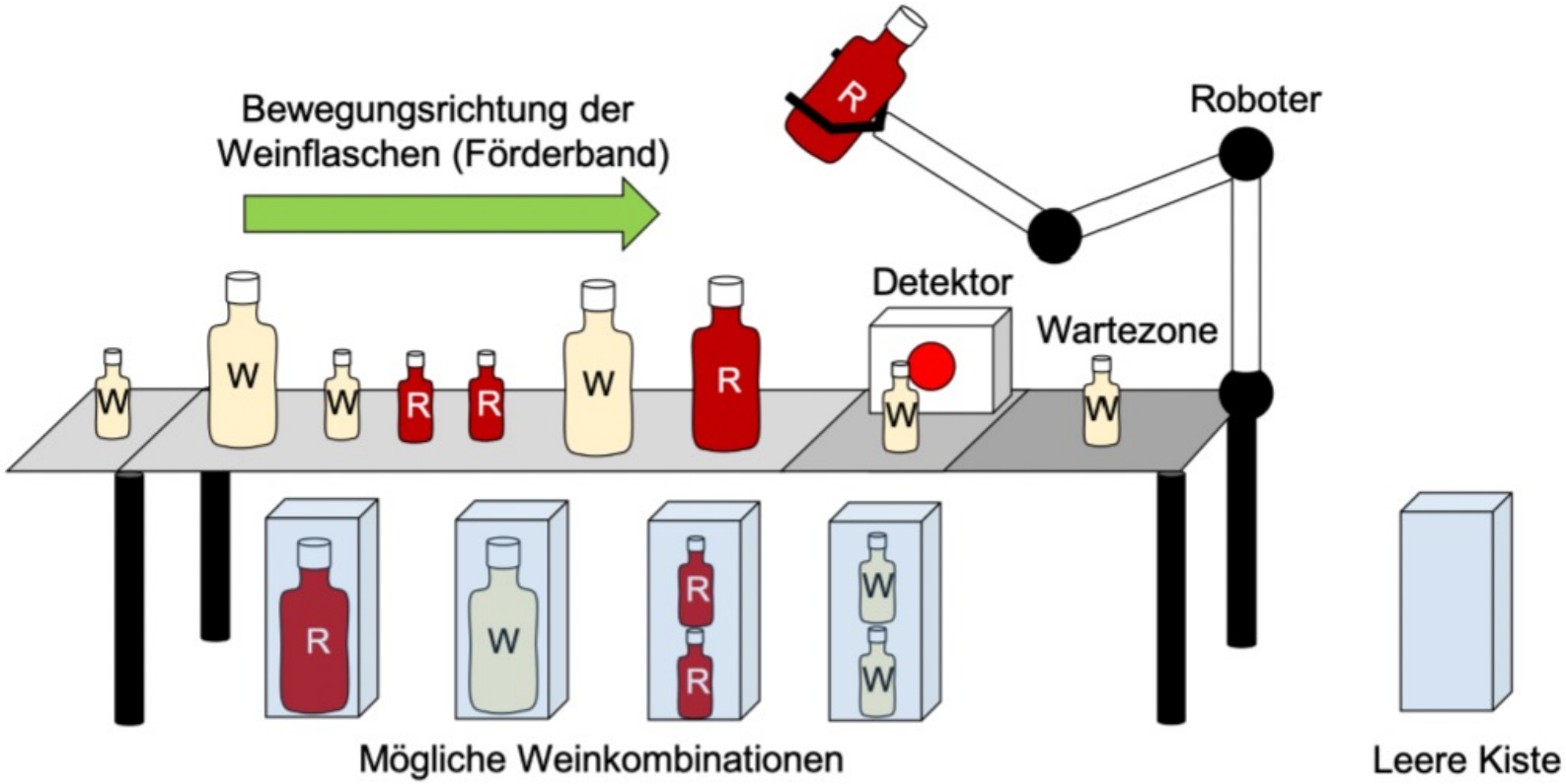
Test 3

A.5) Ergänzen Sie auf den Zustandsgraphen für den Automaten. Zeichnen Sie nur die Kanten, die von diesem "don't care" Zustand anfangen sowie die entsprechenden Eingangs- und Ausgangsvariablen. (4 Punkte)



$$
\begin{aligned}
z_{1(n+1)} &= ((\overline{e_2} \wedge z_1) \vee (\overline{e_1} \wedge e_2 \wedge z_2))_n \\
z_{2(n+1)} &= ((\overline{e_2} \wedge z_2) \vee (e_1 \wedge \overline{z_2}))_n \\
y_n &= (\overline{e_1} \wedge e_2 \wedge z_1)_n
\end{aligned}
$$

Test 3


Bewegungsrichtung der
Weinflaschen (Förderband)
Roboter
Detektor
Wartezone
Mögliche Weinkombinationen
Leere Kiste

Ein Winzer hat eine neue Anlage bestellt, um die Weinflaschen, die er auf seinem Weinberg produziert, in Kisten zu sortieren und zu packen. In seinem Keller wird Rot- und Weisswein erzeugt. Beide Weinsorten können entweder in kleinen (0.5 Liter) oder in grossen (1 Liter) Flaschen abgefüllt werden. Da alle Kisten die gleiche Grüsse haben, werden immer zwei kleine oder eine grosse Flasche(n) zusammengepackt. Rot- und Weisswein werden nie gemischt. Dem Winzer wurde die Anlage in Abbildung B1 vorgeschlagen, der von Ihrer Firma verkauft wird. Als Ingenieur(in) haben Sie jetzt die Aufgabe, einen Automaten zu entwerfen, um diese Anlage zu steuern. Er besteht aus einem Detektor, einem Roboter, einer Wartezone und einer Kontrollstation.

Die Weinflaschen befinden sich zuerst auf einem Förderband und kommen vor dem Detektor vorbei, der die folgenden Signale generiert: **ND**, wenn nichts detektiert wird, **KR** für kleine Rotwein Flaschen, **KW** für kleine Weisswein Flaschen, **GR** für grosse Rotwein Flaschen und **GW** für grosse Weisswein Flaschen. Wenn der Automat eine grosse Flasche detektiert, schickt er dem Roboter ein **PF** Signal: der Roboter packt diese grosse Flasche in eine Weinkiste. Wenn eine kleine Flasche detektiert wird, prüft zuerst der Automat, ob es schon eine kleine Flasche der gleichen Farbe in der Wartezone gibt:

- Wenn ja, dann empfängt der Roboter ein Signal **P2F** und packt die beiden kleinen Flaschen der gleichen Farbe in eine einzige Kiste zusammen. Der Roboter kann die Weinsorten automatisch unterscheiden.
- Wenn nicht, dann wird die detektierte kleine Flasche vom Roboter mit dem **WZ** Befehl in die Wartezone übertragen.

Maximal können sich zwei kleine Weinflaschen von verschiedener Farbe gleichzeitig in der Wartezone befinden. Wenn der Roboter ein **MN** Signal kriegt, macht er nichts und wartet auf den nächsten Befehl.

Test 3

B.1) Erläutern Sie den grundsätzlichen Unterschied zwischen einem Moore- und einem Mealy-Automaten (max. 2 Sätze). (2 Punkte)

B.2) Was sind die Eingänge und die Ausgänge dieses Automaten? Verwenden Sie folgende Variablen: GR, GW, KR, KW, MN, ND, PF, P2F und WZ. (4 Punkte)

Eingänge:

Ausgänge:

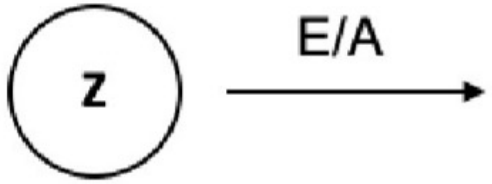
B.3) Wie viele Bits brauchen Sie, um die Eingangs- und Ausgangsvariablen zu kodieren? (2 Punkte)

Bits für die Eingänge:

Bits für die Ausgänge:

Test 3

B.4) 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 Frage B.2 (GR, GW, KR, $\cdots$). (10 Punkte)



Test 3

B.5) Wie viele Flipflops benötigen Sie minimal, um diesen Automaten zu realisieren? (2 Punkte)

B.6) Wie viele Zeilen und Spalten hätte die *komplette* Zustandsfolgetabelle? Begründen Sie. (4 Punkte)

Anzahl an Zeilen:

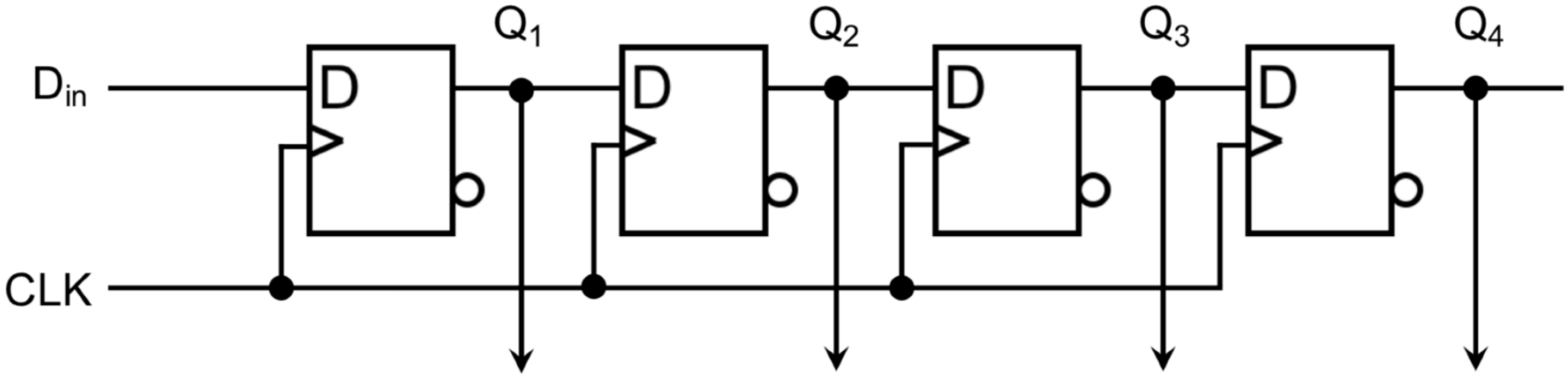
Begründung:

Anzahl an Spalten:

Begründung:

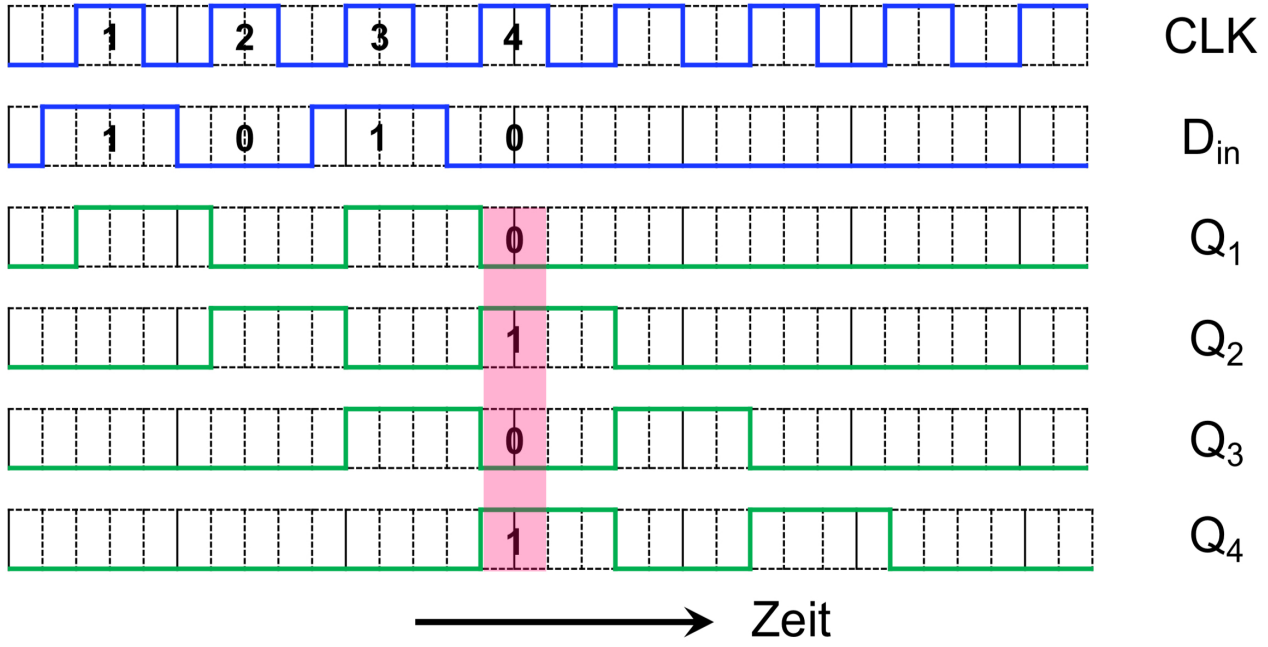
Schieberegister

- n = Anzahl Flipflops
  - n Bits können gespeichert werden
  - n Taktflanken um Bits einzulesen
- Nachteil: Flipflops sind teuer und platzintensiv



Schieberegister

- Charakteristische Gleichungen:
  - Allgemein für ein D-Flipflop: $Q_{n+1}=D_n$
  - Für das Schieberegister: $Q_{in+1}=Q_{i-1n}$ oder $Q_{in+i}=D_n$ (i=1,2,3,4)
- Zeitverhalten beim Einlesen der Datensequenz 1010:



Speichermedien und -funktionen

- Als Speichermedium können **Magnetbänder**, **Plattenlaufwerke** oder **Halbleiter** verwendet werden.

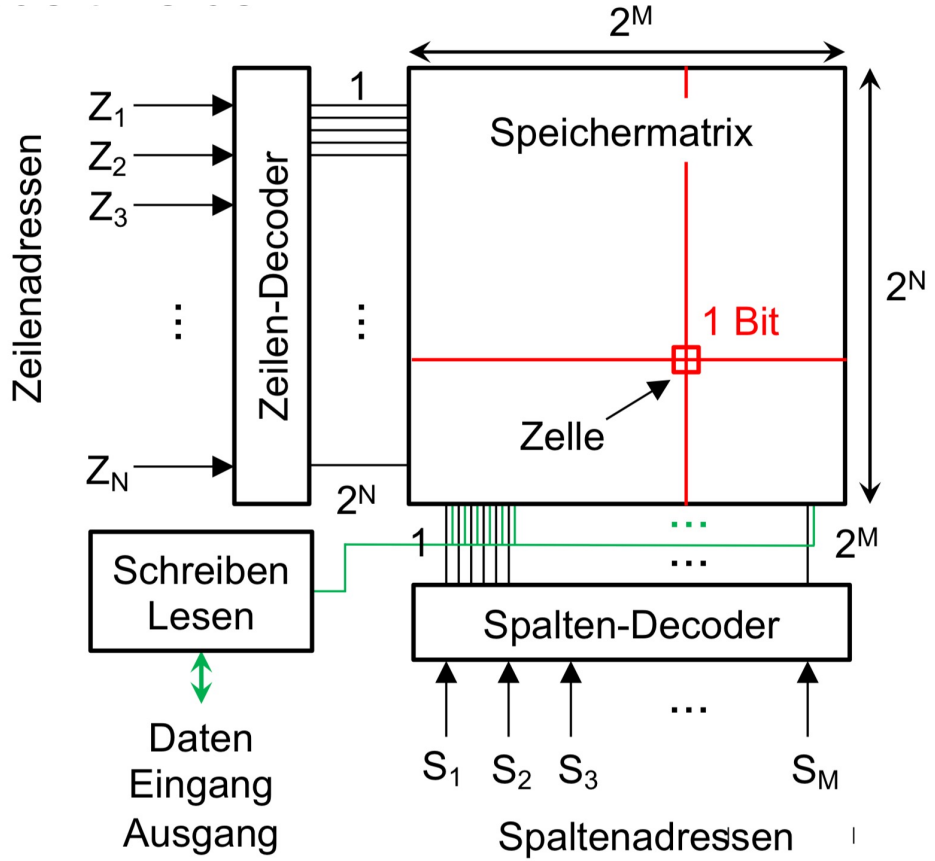
- Speicherfunktionen:
  - **ROM** = Read Only Memory (nur lesen)
  - **RAM** = Random Access Memory (lesen und schreiben)
  - **Permanente** (nichtflüchtige) oder **temporäre** (flüchtige) Speicherung

Speicherorganisation

- **Halbleiter Speicherelemente** werden meistens als Zellenarray oder Matrizen organisiert

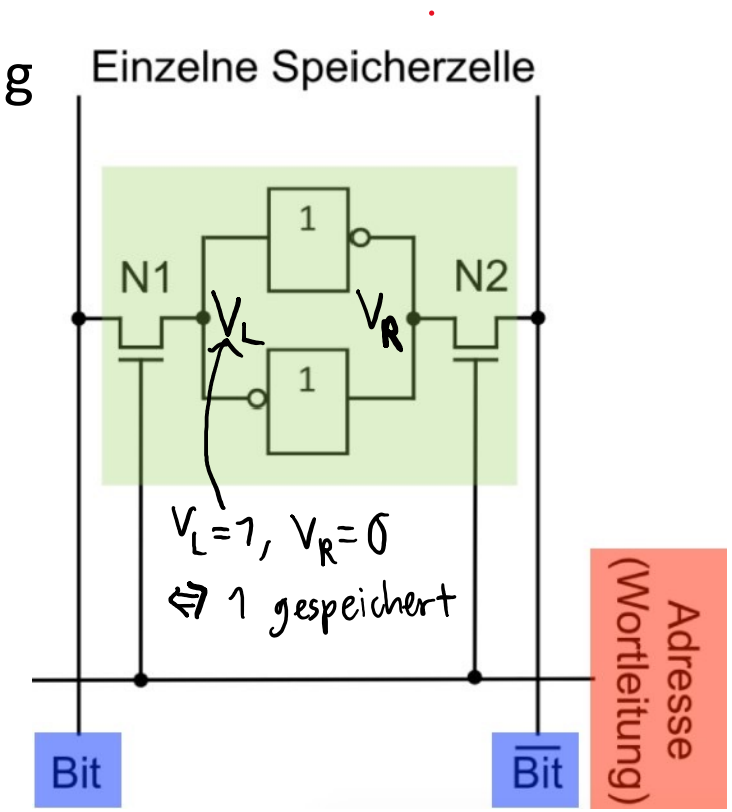
Eigenschaften:

- 2^N Zeilen
- 2^M Spalten
- $2^N \times 2^M$ Bits werden gespeichert
- Lesen und schreiben sind möglich (**RAM**)
- Decoder nötig, um die richtige Zelle (Bit) zu selektieren



SRAM (flüchtig)

- Static random access memory
  - Speichert Daten, solange die Versorgungsspannung angelegt ist.
- Speicherzelle wird angewählt
- Speicherinhalt wird gesetzt/gelesen



- lesen: Bit = 1, !Bit = 1
- 1 schreiben: Bit = 1, !Bit = 0
- 0 schreiben: Bit = 0, !Bit = 1

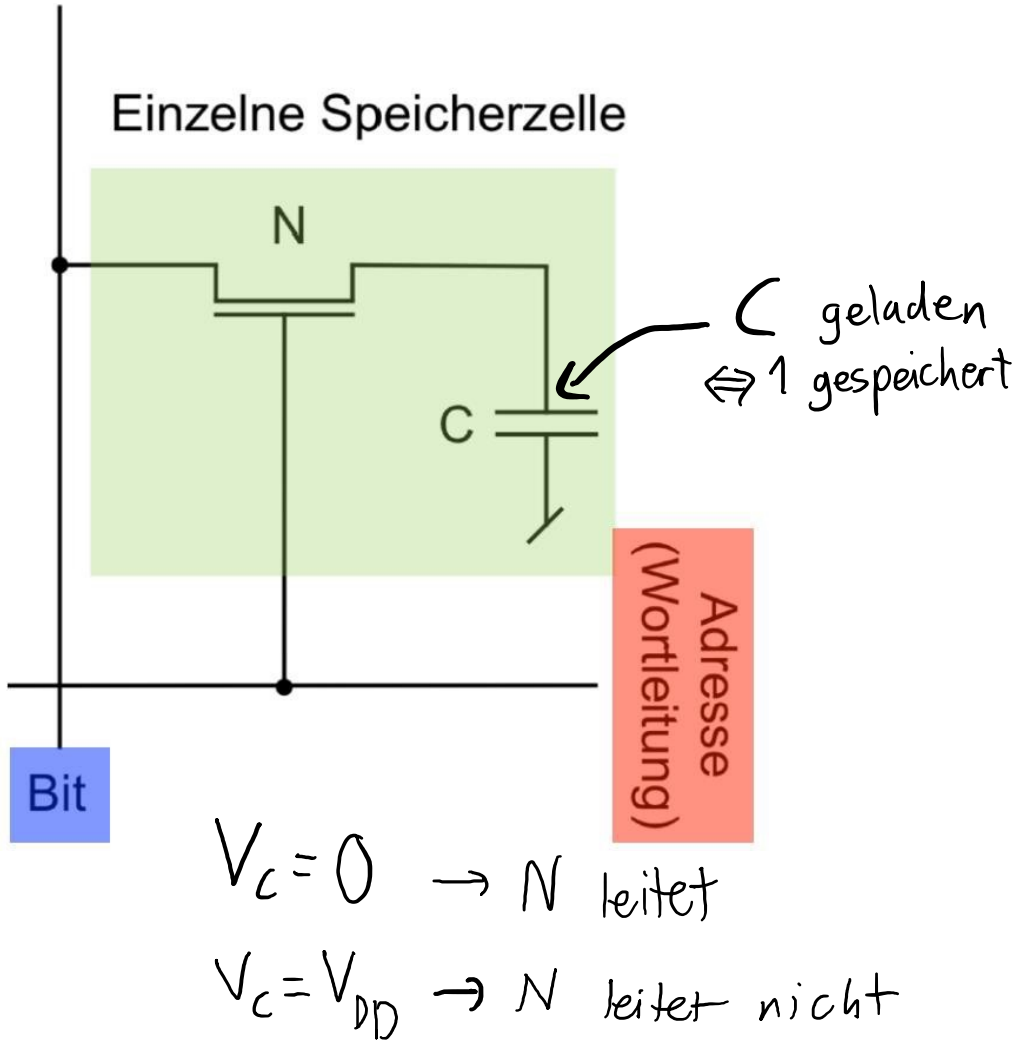
- Schneller Zugriff
- 6 Transistoren

DRAM (flüchtig)

- Dynamic random access memory
  - Datenspeicherung muss periodisch (20ms) wieder aufgefrischt werden.
- Speicherzelle wird angewählt
- Speicherinhalt wird gesetzt/gelesen

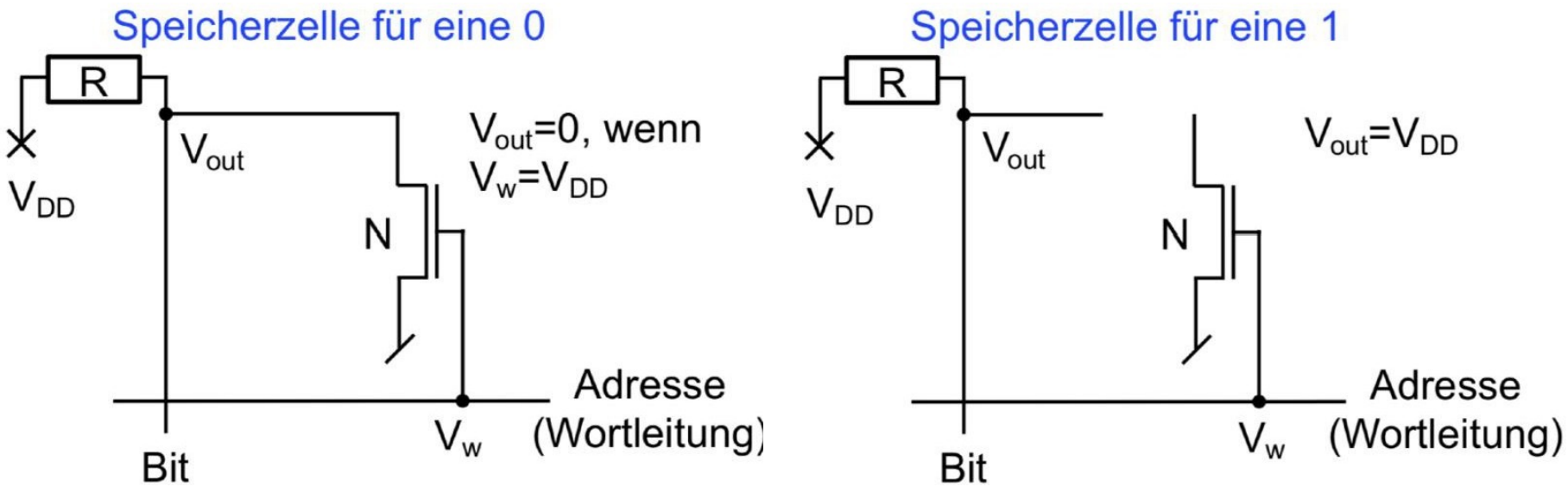
- Daten lessen (0 gespeichert): Bit nicht gesetzt, W = 1
- 1 schreiben: Bit = 1, W = 1

- Langsamer Zugriff
- Platzsparend: 1 Transistor & 1 Kondensator



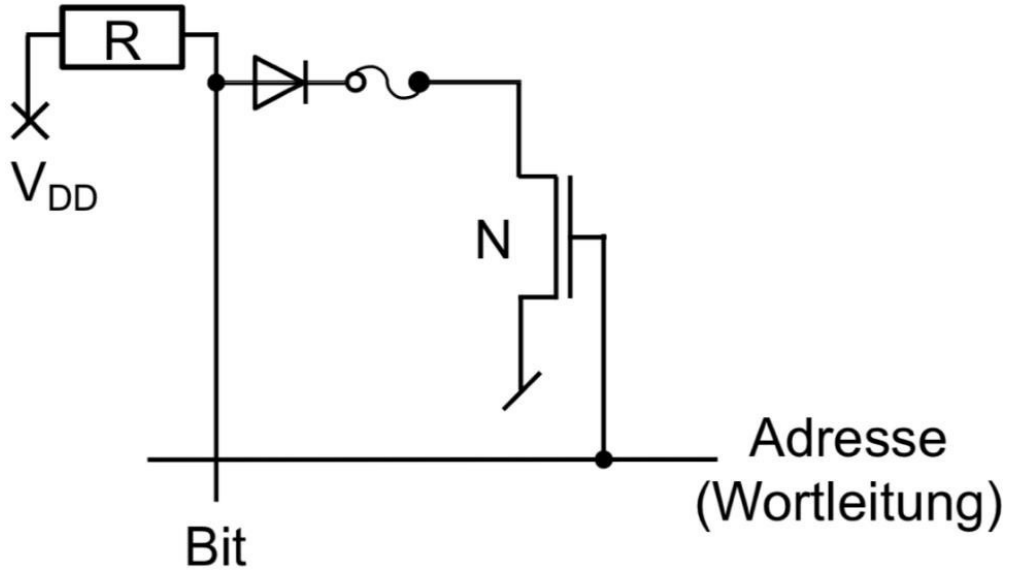
ROM (nichtflüchtig)

- Read only memory
- werden bei Produktion programmiert



PROM (nichtflüchtig)

- Programmierbare ROM
- Als 1 oder 0 programmierbar mittels Schmelzsicherung (fusable link)



Tipps für Vorbereitung

- Möglichst viele alte Prüfungen lösen
- Vor allem Automaten, CMOS Analyse und letzte Aufgabe (Zähler, Folgededektor) lösen
- Selber lösen probieren, nicht einfach die Lösungen abschreiben
- Klärt eure Fragen sofort, nicht auf den letzten Moment warten

Tipps für Prüfung

- Lasst euch nicht stressen, auch wenn diese Prüfung als „Stressprüfung“ bezeichnet wird
- Ihr müsst nicht alles lösen (Kennt eure Stärken und löst diese Aufgaben zuerst)
- Plant in welcher Reihenfolge ihr die Prüfung lösen wollt
- Nehmt eine Uhr mit!