Technische Informatik
Automaten

Thorsten Thormählen

10. Dezember 2023

Teil 7, Kapitel 4

Dies ist die Druck-Ansicht.

Aktiviere Präsentationsansicht

Steuerungstasten

→ nächste Folie (auch Enter oder Spacebar).
← vorherige Folie
d schaltet das Zeichnen auf Folien ein/aus
p wechselt zwischen Druck- und Präsentationsansicht
CTRL + vergrößert die Folien
CTRL - verkleinert die Folien
CTRL 0 setzt die Größenänderung zurück

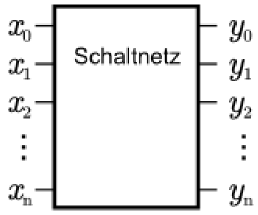
Notation

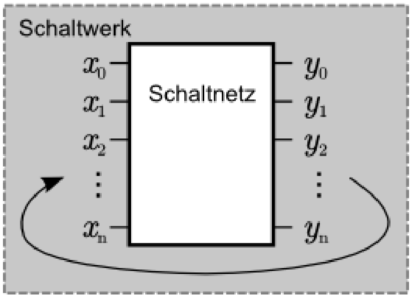
Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	a, b, x, y
Funktionen	aufrecht	$\mathrm{f}, \mathrm{g}(x), \max(x)$
Vektoren	fett, Elemente zeilenweise	$\mathbf{a}, \mathbf{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^\top,$ $\mathbf{B} = (x, y, z)^\top$
Matrizen	Schreibmaschine	$\mathtt{A}, \mathtt{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, B = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}^2, \mathbb{R}^3$

Inhalt

Schaltnetze und Schaltwerke

- Ohne Rückkopplung (Schaltnetz)
  - Werte an den Ausgängen sind nur abhängig von den Eingängen
  - Solche Schaltungen verhalten sich immer gleich (sind zustandslos) und sind durch ihre Schaltfunktion eindeutig beschrieben
  - Es ist jedoch nicht möglich, etwas zu speichern
- Mit Rückkopplung (Schaltwerk)
  - Werte an den Ausgängen sind abhängig von den Eingängen und den vorherigen Ausgangswerten
  - Das Zeitverhalten muss genau betrachtet werden
  - Die vorherigen Ausgangswerte können als Zustand der Gatter interpretiert werden. Abhängig vom Zustand verhalten sich die Gatter anders (zustandsabhängige Schaltfunktion)
  - Es wird möglich, Zustände zu speichern





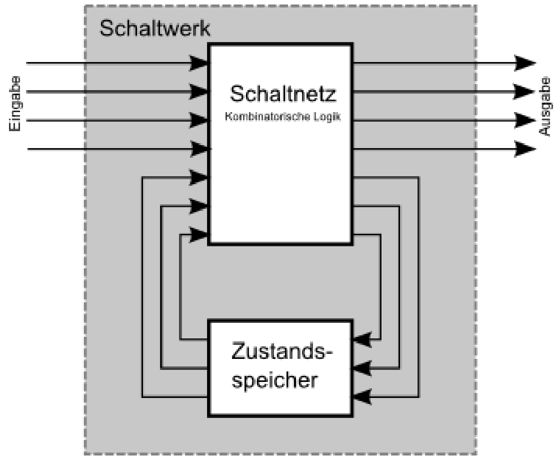
Schaltwerke

Im vorangegangenen Kapitel wurde gezeigt, dass durch Rückkopplung Speicherelemente (Flip-Flops, Register) realisiert werden können, die einen Zustand speichern

Demnach kann ein Schaltwerk auch als eine Kombination aus einem Schaltnetz und einem Zustandsspeicher dargestellt werden

Die Ausgabe eines Schaltwerks kann abhängig vom aktuellen Zustand sein

Abhängig von den Eingangswerten kann sich der Zustand eines Schaltwerks ändern



Zur Beschreibung der zustandsabhängigen Schaltfunktion können *endliche Automaten* verwendet werden

Endliche Automaten

Ein *endlicher Automat* (engl. Finite State Machine, FSM) ist definiert durch

- eine endliche Menge $\mathcal{A}$ von Eingabesymbolen $a_i \in \mathcal{A}$ (Alphabet)
- eine endliche Menge $\mathcal{S}$ von Zuständen $s_i \in \mathcal{S}$
- einen Anfangszustand $s_0 \in \mathcal{S}$
- eine Zustandsübergangsfunktion $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

Des Weiteren kann er umfassen

- eine endliche Menge $\mathcal{B}$ von Ausgabesymbolen $b_i \in \mathcal{B}$
- eine Ausgabefunktion $\lambda : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{B}$

Bei deterministischen Automaten erfolgen die Zustandsübergänge deterministisch (nicht zufällig)

Endliche Automaten können durch Zustandsübergangsgraphen dargestellt werden

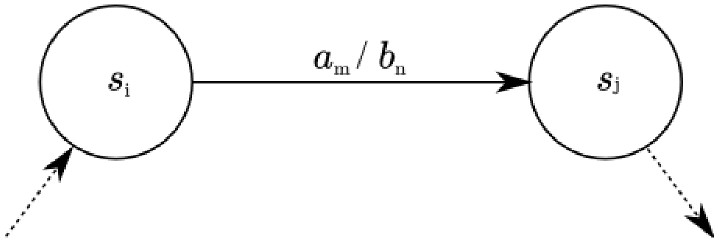
Zustandsübergangsgraphen

In einem Zustandsübergangsgraphen wird jeder Zustand $s_i \in \mathcal{S}$ als ein Kreis dargestellt

Die möglichen Zustandsübergänge werden mit Pfeilen gekennzeichnet

Jeder Pfeil wird mit dem zugehörigen Eingabesymbol $a_m \in \mathcal{A}$ beschriftet, für das dieser Zustandsübergang auftritt

Außerdem (abgetrennt durch einen "/") kann jeweils das Ausgabesymbol $b_n \in \mathcal{B}$ angegeben werden



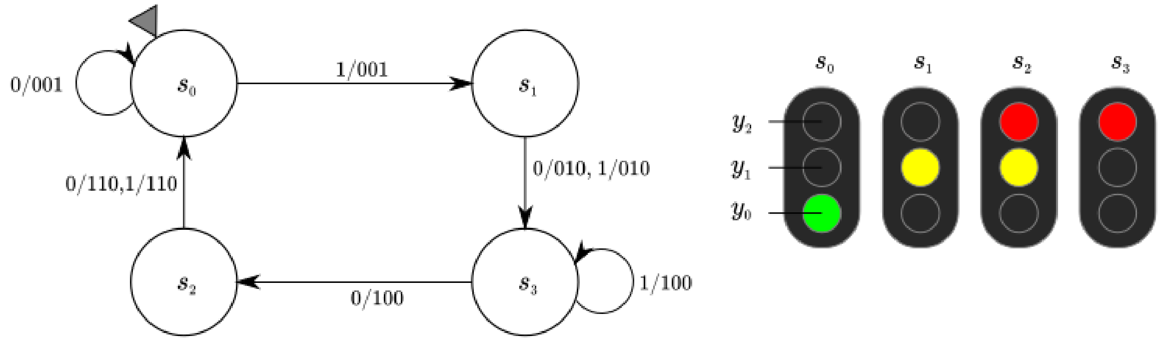
Zustandsübergangsgraphen

Beispiel: Ampelschaltung

- Es soll eine Schaltung für eine Fußgängerampelanlage erstellt werden. Es wird dabei die Ampel, die den Autoverkehr regelt, betrachtet (nicht die Fußgängerampel)
- Die Ampel reagiert auf das Drücken eines Ampelknopfs durch einen Fußgänger:
  a_0 = 0 bedeutet der Ampelknopf wurde nicht gedrückt
  a_1 = 1 bedeutet der Ampelknopf wurde gedrückt
- Im Anfangszustand $s_0 \in \mathcal{S}$ ist die Ampel grün
- Wurde der Ampelknopf gedrückt, soll die Ampel zunächst auf gelb und dann auf rot schalten
- Es wird weiterhin davon ausgegangen, dass das durch den Ampelknopf gesteuerte Eingabesymbol automatisch, nachdem der Fußgänger genug Zeit hatte die Straße zu überqueren, von a_1 nach a_0 wechselt
- Die Ampel soll dann zunächst gelb-rot zeigen und schließlich wieder grün

Zustandsübergangsgraphen

Beispiel: Ampelschaltung



Menge der Eingabesymbole $\mathcal{A} = \{a_0, a_1\}$ binär kodiert mit $\{0, 1\}$

Menge der Zustände $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$

Menge der Ausgabesymbole $\mathcal{B} = \{b_0, b_1, b_2, b_3\}$ binär kodiert mit $y_2 y_1 y_0$ zu $\{001, 010, 110, 100\}$

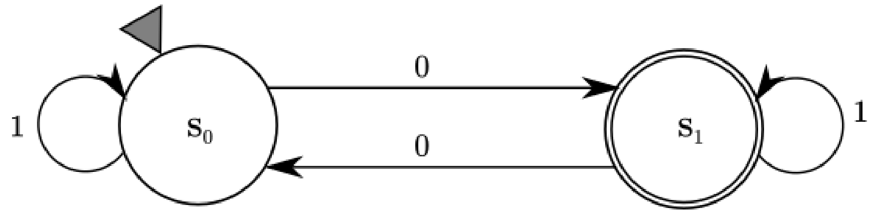
Es gibt $|\mathcal{S}| \cdot |\mathcal{A}| = 4 \cdot 2 = 8$ mögliche Zustandsübergänge

Transduktoren und Akzeptoren

- Transduktoren
  - Enthält ein Automat eine Ausgabefunktion λ wird er als Transduktor bezeichnet
  - Transduktoren sind Automaten, die aus Sequenzen von (binären) Eingabesymbolen, Sequenzen von (binären) Ausgabesymbolen erzeugen
  - Diese "übersetzen" die Eingabe in eine Ausgabe
- Akzeptoren
  - Akzeptoren sind Automaten, die bestimmt Sequenzen von Eingabewörtern akzeptieren oder verwerfen
  - Dazu wird eine Teilmenge $\mathcal{F} \subset \mathcal{S}$ der möglichen Zustände zu akzeptierenden Zuständen erklärt, so genannte "Finalzustände"
  - Endet ein Automat nach Abarbeiten einer Sequenz von Eingabesymbolen in einem Finalzustand, wird die Eingabe akzeptiert, sonst wird sie verworfen
  - Akzeptoren erzeugen somit keine Sequenz von Ausgabesymbolen

Akzeptoren

- Finalzustände werden im Zustandsübergangsgraphen als doppelt gestrichene Kreise dargestellt
- Der Anfangszustand wird mit einem auf den Zustand zeigendem Dreieck gekennzeichnet
- Beispiel für einen Akzeptor:
  - Ausgehend vom Anfangszustand s_0 würde der dargestellte Automat alle Eingabesequenzen akzeptieren, die eine ungerade Anzahl an Nullen enthalten



  - Menge der Eingabewörter $\mathcal{A} = \{0, 1\}$
  - Menge der Zustände $\mathcal{S} = \{s_0, s_1\}$
  - Menge der Finalzustände $\mathcal{F} = \{s_1\}$

Mealy und Moore-Automaten

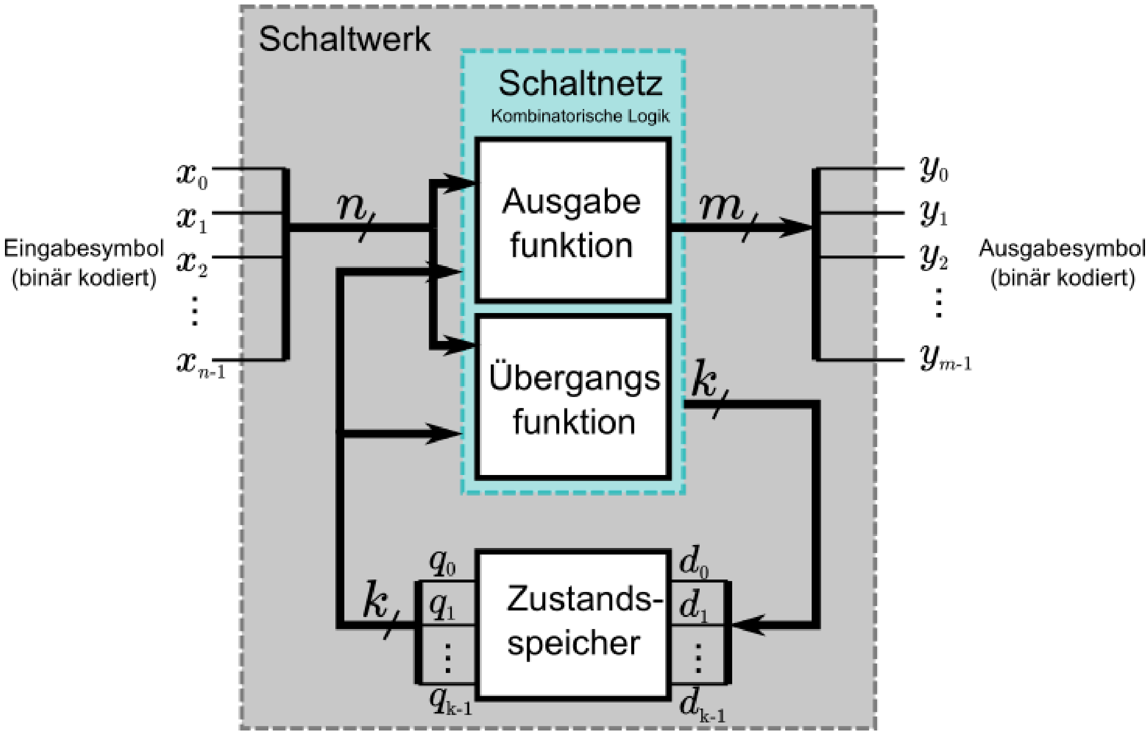
Mealy-Automaten

Bei einem Mealy-Automaten ist die Ausgabefunktion $\lambda : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{B}$ vom aktuellen Zustand und der Eingabe abhängig

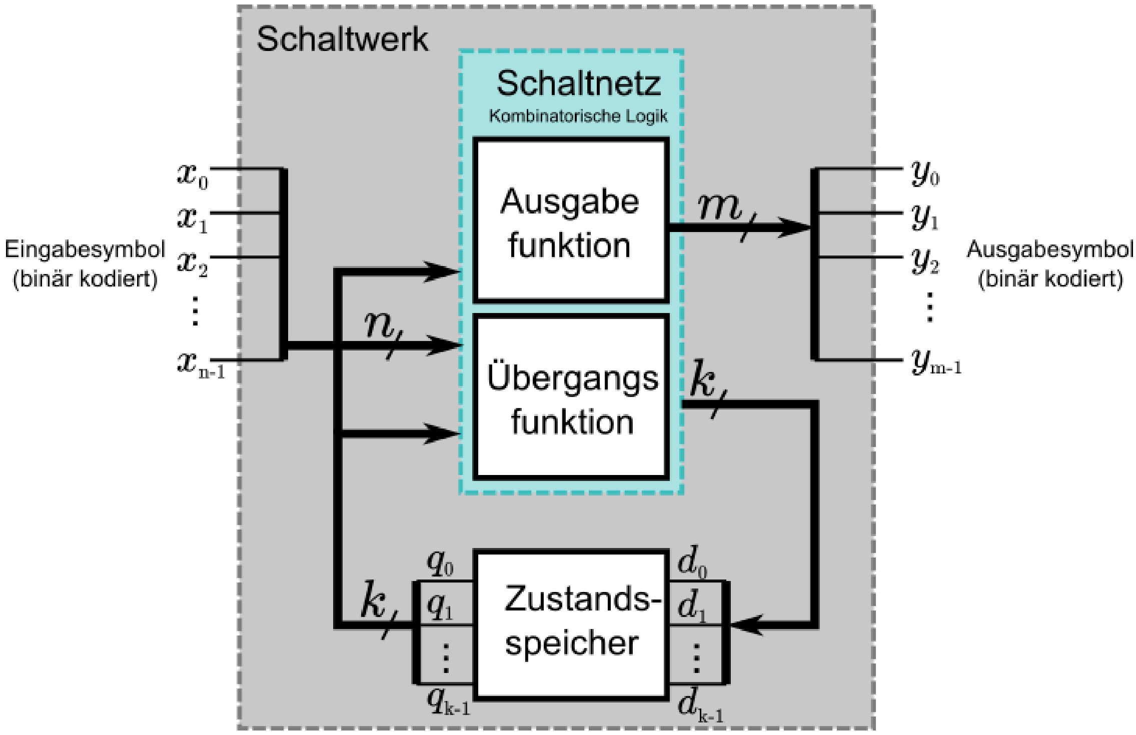
Moore-Automaten

Bei einem Moore-Automaten ist die Ausgabefunktion $\lambda : \mathcal{S} \rightarrow \mathcal{B}$ nur vom aktuellen Zustand abhängig

Mealy-Automat

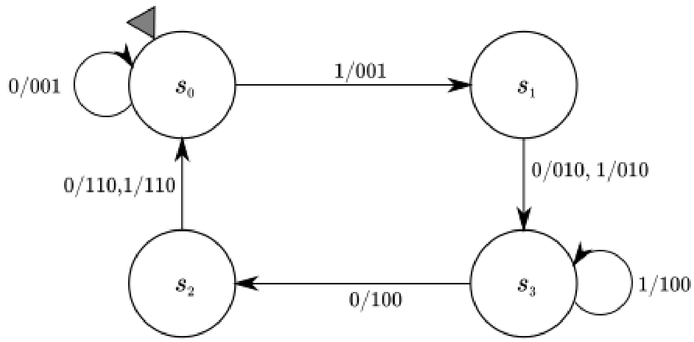


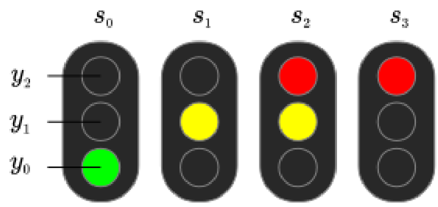
Moore-Automat



Vom Zustandsübergangsgraphen zum Schaltwerk

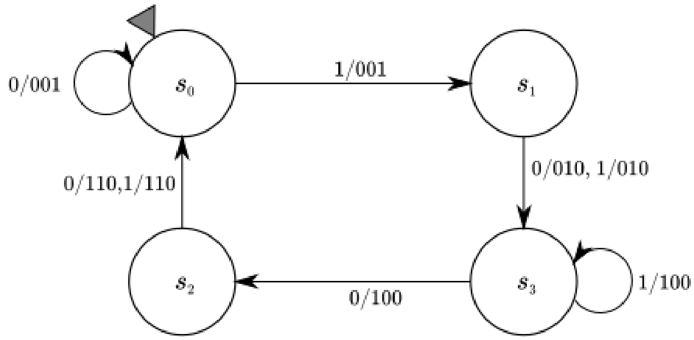
Beispiel 1: Der Zustandsübergangsgraph der Ampelschaltung soll in ein Schaltwerk umgesetzt werden

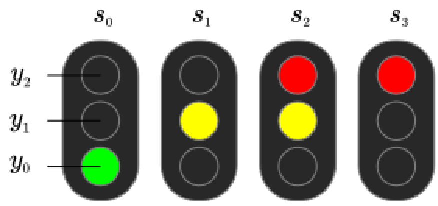




- Zur Erstellung des Schaltwerks kann die Ausgabefunktion, die Übergangsfunktion und der Zustandsspeicher separat betrachtet werden
- Für gewünschte Ausgabe- und Übergangsfunktion kann direkt aus dem Zustandsübergangsgraphen abgelesen und als Wahrheitstafel dargestellt werden
- Zur Minimierung können anschließend z.B. KV-Diagramme oder das Quine-McCluskey-Verfahren verwendet werden

Vom Zustandsübergangsgraphen zum Schaltwerk





Handelt es sich um einen Moore- oder Mealy-Automaten?

Antwort: Moore-Automat, da Ausgabefunktion nur abhängig vom aktuellen Zustand

Ausgabefunktion:

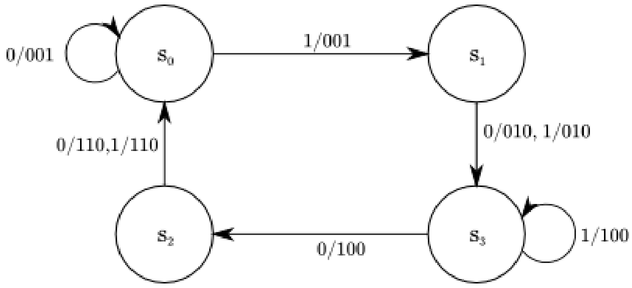
$$y_2 = q_1$$

$$y_1 = q_0 \not\leftrightarrow q_1$$

$$y_0 = \neg q_0 \wedge \neg q_1 = \neg(q_0 \vee q_1)$$

Zustand	q_1	q_0	y_2	y_1	y_0
s_0	0	0	0	0	1
s_1	0	1	0	1	0
s_2	1	0	1	1	0
s_3	1	1	1	0	0

Vom Zustandsübergangsgraphen zum Schaltwerk



Übergangsfunktion:

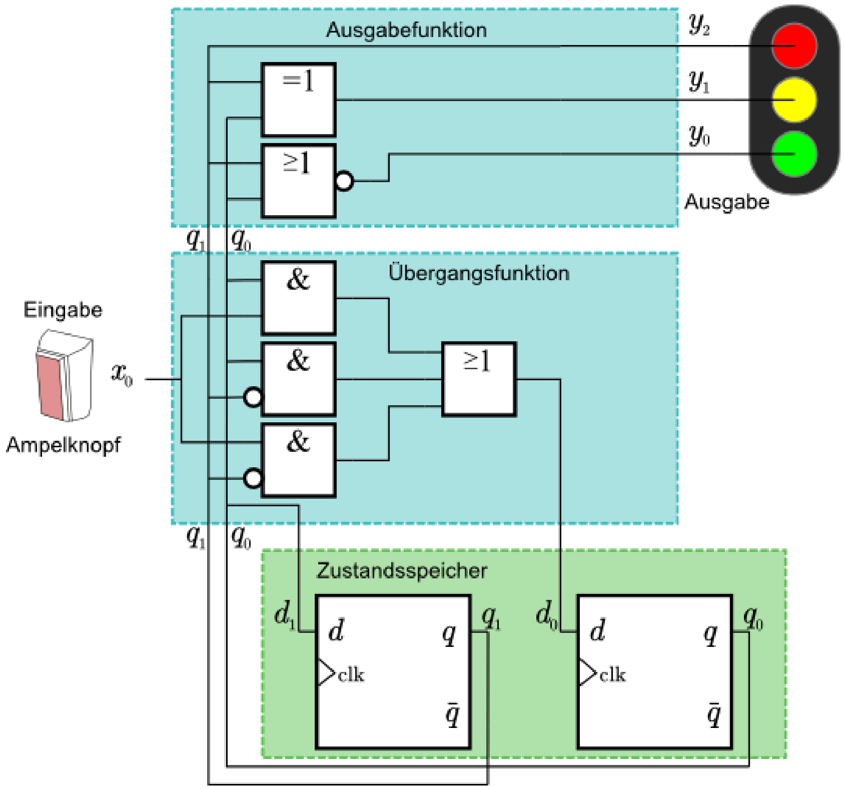
$d_1 = q_0$

Mittels KV-Diagramm oder Quine-McCluskey-Verfahren:

$d_0 = (\bar{q}_1 x_0) \vee (\bar{q}_1 q_0) \vee (q_0 x_0)$

Zustand	q_1	q_0	x_0	d_1	d_0
s_0	0	0	0	0	0
s_0	0	0	1	0	1
s_1	0	1	0	1	1
s_1	0	1	1	1	1
s_2	1	0	0	0	0
s_2	1	0	1	0	0
s_3	1	1	0	1	0
s_3	1	1	1	1	1

Vom Zustandsübergangsgraphen zum Schaltwerk



Vom Zustandsübergangsgraphen zum Schaltwerk

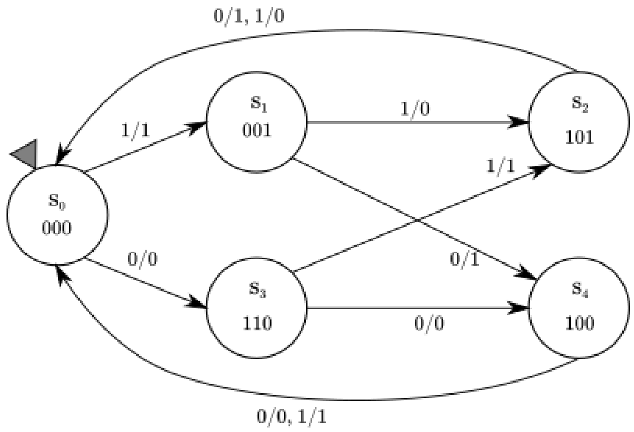
Beispiel 2: Dieser Zustandsübergangsgraph übersetzt einen 3-Bit Binär-Code in einen Gray-Code

Menge der Eingabesymbole $\mathcal{A} = \{0, 1\}$

Menge der Zustände $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4\}$

Menge der Ausgabesymbole $\mathcal{B} = \{0, 1\}$

Anfangszustand s_0



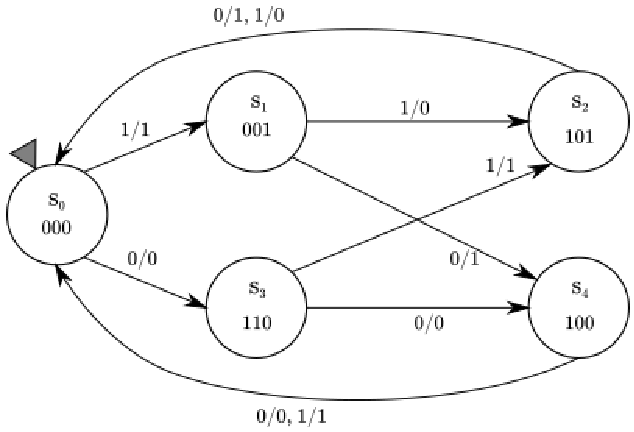
Eingabe	Ausgabe
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

[**Quelle:** basierend auf: D. W. Hoffmann: *Grundlagen der Technischen Informatik*, 2. Auflage; Hanser 2009, Abb. 9.2, S. 298ff]

Vom Zustandsübergangsgraphen zum Schaltwerk

Aufgabe: Entwurf des entsprechenden Schaltwerks

Handelt es sich um einen Moore- oder Mealy-Automaten?

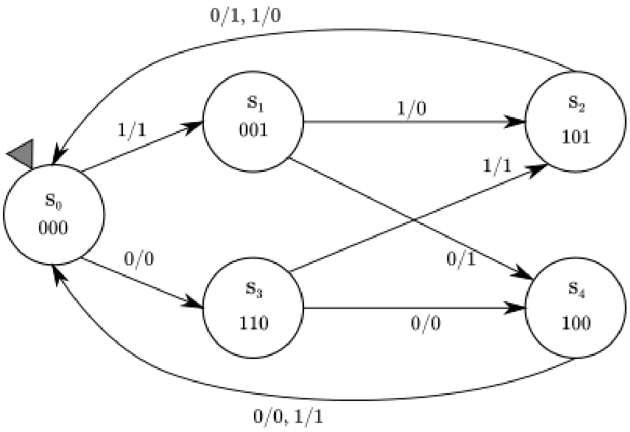


Eingabe	Ausgabe
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Antwort: Mealy-Automat, da Ausgabefunktion abhängig vom aktuellen Zustand und der Eingabe

[**Quelle:** basierend auf: D. W. Hoffmann: *Grundlagen der Technischen Informatik*, 2. Auflage; Hanser 2009, Abb. 9.2, S. 298ff]

Vom Zustandsübergangsgraphen zum Schaltwerk

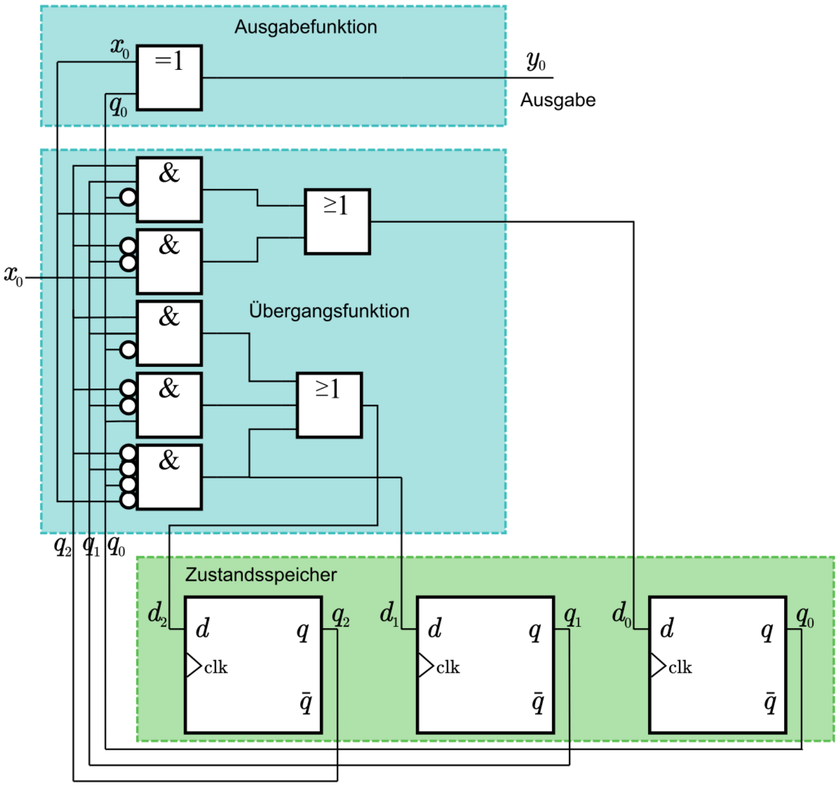


q_2	q_1	q_0	x_0	d_2	d_1	d_0	y_0
0	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1
0	0	1	0	1	0	0	1
0	0	1	1	1	0	1	0
1	0	1	0	0	0	0	1
1	0	1	1	0	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1

Mittels KV-Diagramm oder durch Ablesen und Vereinfachen ergibt sich:

$$\begin{aligned} d_2 &= \bar{q}_2 \bar{q}_1 \bar{q}_0 \bar{x}_0 \vee \bar{q}_2 \bar{q}_1 q_0 \vee q_2 q_1 \bar{q}_0 \\ d_1 &= \bar{q}_2 \bar{q}_1 \bar{q}_0 \bar{x}_0 \\ d_0 &= \bar{q}_2 \bar{q}_1 x_0 \vee q_2 q_1 \bar{q}_0 x_0 \\ y_0 &= x_0 \not\leftrightarrow q_0 \end{aligned}$$

Vom Zustandsübergangsgraphen zum Schaltwerk



Gibt es Fragen?

Anregungen oder Verbesserungsvorschläge können auch gerne per E-mail an mich gesendet werden: [Kontakt]

[Weitere Vorlesungsfolien]