Kapitel 4 – Sequentielle Logik

1. Speichernde Elemente
2. **Sequentielle Schaltkreise**
3. Entwurf sequentieller Schaltkreise
4. SRAM
5. Anwendung: Datenpfade von ReTI

Albert-Ludwigs-Universität Freiburg

Prof. Dr. Christoph Scholl

Institut für Informatik

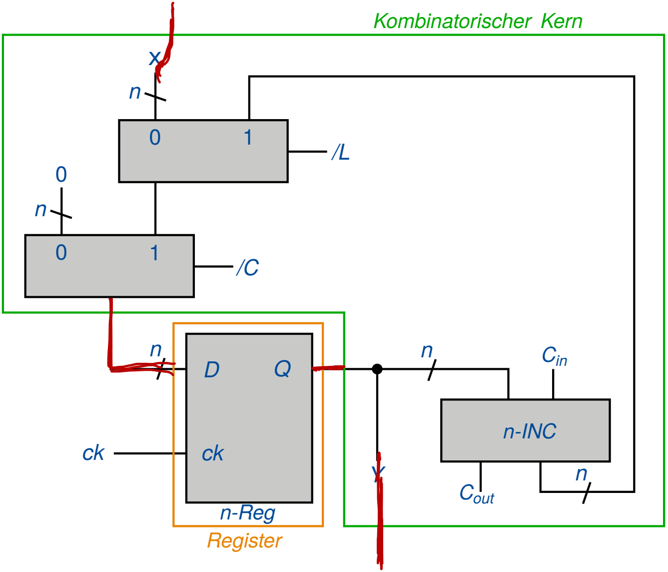
WS 2015/16

Sequentielle Schaltkreise

- Im Folgenden werden keine allgemeinen Schaltpläne mehr analysiert, sondern sogenannte Schaltwerke (auch (synchrone) sequentielle Schaltkreise genannt).
- Diese bestehen aus einem Register und einem (kombinatorischen) Schaltkreis (auch kombinatorischer Kern genannt).
- Im Gegensatz zu (kombinatorischen) Schaltkreisen können Schaltwerke (= sequentielle Schaltkreise) Zyklen enthalten. Die Zyklen müssen aber durch Flipflops des Registers gehen.
- Der Zustand eines Schaltwerkes ist gegeben durch die im Register gespeicherten Werte.
- Schaltwerke (= sequentielle Schaltkreise) entsprechen endlichen Zustandsautomaten.

Schaltkreise	Schaltwerke
(keine Zyklen, keine speichernde Elemente)	(Zyklen auf FFs beschränkt, keine Zyklen im kombinatorischen Kern)
andere Begriffe: kombinatorische Schaltkreise	sequentielle Schaltkreise

Beispiel: Zähler als sequentieller Schaltkreis



Endliche Zustandsautomaten

- Endliche Zustandsautomaten (Finite State Machines, FSMs) sind ein Formalismus, um sequentielles (zeitabhängiges) Verhalten zu spezifizieren.
  - Mealy- und Moore-Automaten
  - (In der theoretischen Informatik werden endliche Automaten mit akzeptierenden Zuständen betrachtet. Diese sind mit FSMs verwandt, aber nicht identisch.)
- Aus einer FSM-Spezifikation kann der sequentielle Schaltkreis hergeleitet werden (Sequentielle Synthese).

Halbautomat

Definition

Das Quadrupel $H = (I, S, S_0, \delta)$ heißt deterministischer, endlicher Halbautomat. Dabei bezeichnet:

- I eine endliche Menge von erlaubten Eingabesymbolen ("Eingabealphabet"), (bei uns meistens $I \subseteq \{0,1\}^n$)
- S eine endliche Menge von Zuständen, (bei uns meistens $S \subseteq \{0,1\}^m$)
- $S_0 \subseteq S$ ist eine endliche Menge von erlaubten Anfangszuständen,
- $\delta : S \times I \rightarrow S$ eine Übergangsfunktion.

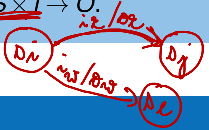
$s_i \xrightarrow{i_k} s_j$ $\quad \delta(s_i, i_k) = s_j$

Mealy- und Moore-Automat

Definition

Ein Mealy-Automat $M = (I, O, S, S_0, \delta, \lambda)$ ist ein endlicher deterministischer Halbautomat H erweitert um:

- eine endliche Menge O von Ausgabesymbolen („Ausgabealphabet”),
  (bei uns meistens Teilmenge von $\{0,1\}^k$)
- eine Ausgabefunktion $\lambda : S \times I \to O$.

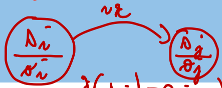


$\lambda(s_i, i_k) = o_k$
$\delta(s_i, i_k) = s_j$

Definition

Ein Moore-Automat $M = (I, O, S, S_0, \delta, \lambda)$ ist ein endlicher, deterministischer Halbautomat H erweitert um:

- eine endliche Menge O von Ausgabesymbolen,
- eine Ausgabefunktion $\lambda : S \to O$.



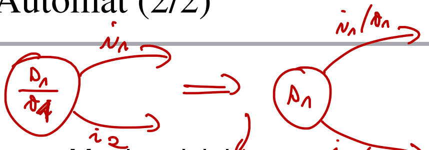
$\lambda(s_i) = o_i, \ \lambda(s_j) = o_j$

Mealy- vs. Moore-Automat (1/2)

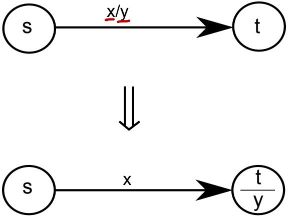
- Beim Mealy-Automaten ist:
  - die Ausgabe abhängig vom aktuellen Zustand und der aktuellen Eingabe,
  - der Folgezustand abhängig vom aktuellen Zustand und der aktuellen Eingabe.
- Ein Moore-Automat ist ein spezieller Mealy-Automat, bei dem die Ausgabe nur vom aktuellen Zustand und nicht von der Eingabe abhängt.
- Moore- und Mealy-Automaten kann man ineinander überführen.

UNI FREIBURG

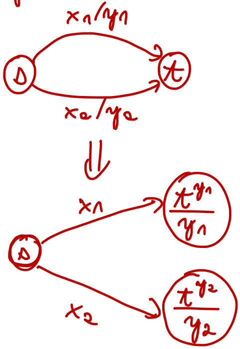
Mealy- vs. Moore-Automat (2/2)



- Überführung Moore → Mealy: trivial
- Überführung Mealy → Moore:
  Grundidee:"Ziehe Ausgabe in den Zustand"



genauer:



UNI FREIBURG

a) Zustands- und Ausgangstafel:

Eingabe-Symbol → x | Zustand → *state* | Folgezustand → *next-state* | y ← Ausgabe

x	*state*	*next-state*	y
1	s_1	s_1	0
0	s_1	s_2	1
-	s_2	s_1	0

← Abkürzung für 2 Zeilen

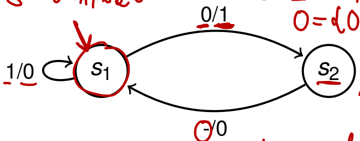
0   1

b) Flusstafel:

	$x=0$	$x=1$
s_1	$s_2, 1$	$s_1, 0$
s_2	$s_1, 0$	$s_1, 0$

c) Zustandsdiagramm:

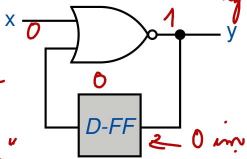
$S = \{s_1, s_2\}$ $S_0 = \{s_1\}$ $I = \{0,1\}$ $O = \{0,1\}$



$\delta(s_1, 0) = s_2$
$\lambda(s_1, 0) = 1$
usw.

← 0 oder 1, „egal welche Eingabe"

d) Sequentieller Schaltkreis: (Zustandskodierung: $s_1 \mathrel{\hat=} 0$, $s_2 \mathrel{\hat=} 1$)

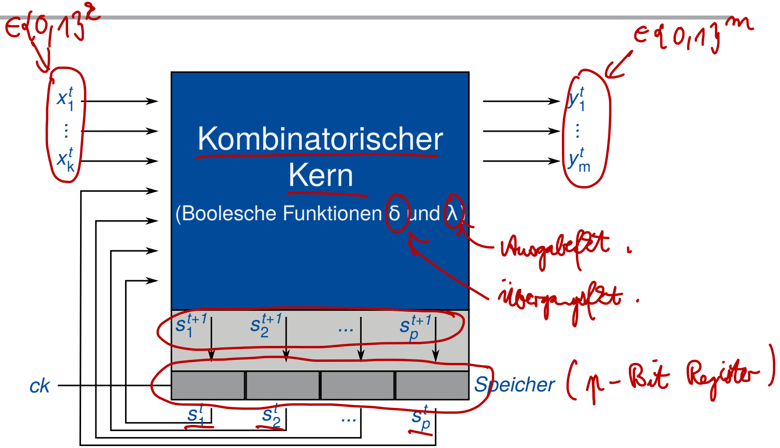


← 0 im D-FF $\mathrel{\hat=} s_1$
1 im D-FF $\mathrel{\hat=} s_2$

- Im Folgenden: Weg von *c*) zu *d*)

→ Kante steht für abkürzend für 2 Kanten (ein Mal Eingabe 0, ein Mal mit Eingabe 1)

Sequentielle Schaltkreise allgemein



$$
\begin{aligned}
y_i^t &= \lambda_i(x_1^t, x_2^t, \ldots, x_k^t, s_1^t, s_2^t, \ldots, s_p^t) \\
s_i^{t+1} &= \delta_i(x_1^t, x_2^t, \ldots, x_k^t, s_1^t, s_2^t, \ldots, s_p^t)
\end{aligned}
$$

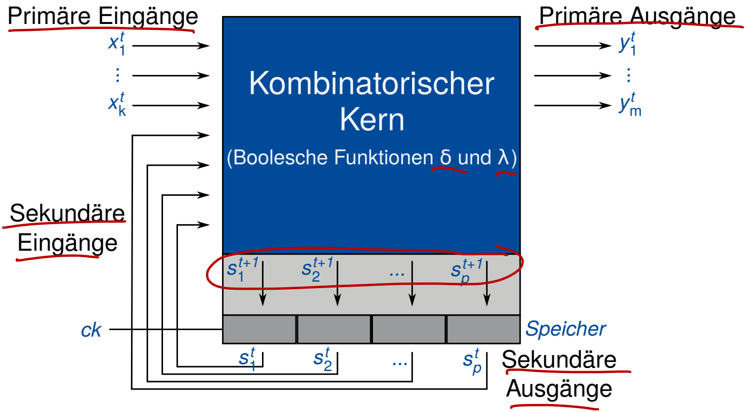
Die Belegung $S^t = (s_1^t, \ldots, s_p^t)$ der Flipflops im Register heißt Zustand des sequentielle Schaltkreises zum Zeitpunkt t.

Kombinatorischer Kern

- Der kombinatorische Kern hat vier Arten von Ein- und Ausgängen:
  - Primäre Eingänge bekommen Werte „von außen".
  - Primäre Ausgänge liefern Werte „nach außen".
  - Sekundäre Eingänge sind mit den Datenausgängen der Flipflops im Register verbunden. Auf diese Weise kann der aktuelle Zustand des Schaltkreises in Funktionen δ und λ berücksichtigt werden.
  - Sekundäre Ausgänge sind mit den Dateneingängen der Flipflops verbunden. Durch sie wird der nächste Zustand des Schaltkreises spezifiziert.

UNI FREIBURG
11

Primäre und sekundäre Ein- und Ausgänge



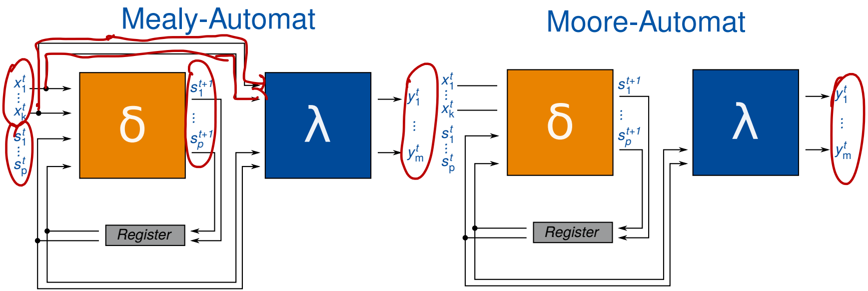
$$y_i^t = \lambda_i(x_1^t, x_2^t, \ldots, x_k^t, s_1^t, s_2^t, \ldots, s_p^t)$$

$$s_i^{t+1} = \delta_i(x_1^t, x_2^t, \ldots, x_k^t, s_1^t, s_2^t, \ldots, s_p^t)$$

UNI FREIBURG

12

Sequentielle Schaltung für einen FSM



- Eingabevektor: $X^t = (x_1^t, x_2^t, \dots, x_k^t)$
- Ausgabevektor: $Y^t = (y_1^t, y_2^t, \dots, y_m^t)$
- Zustandsvektor: $S^t = (s_1^t, s_2^t, \dots, s_p^t)$
- Ausgabefunktion (Mealy): $Y^t = \lambda(X^t, S^t)$
- Übergangsfunktion: $S^{t+1} = \delta(X^t, S^t)$
- Ausgabefunktion (Moore): $Y^t = \lambda(S^t)$

13

UNI FREIBURG