

# **DIGITALE SCHALTUNGEN**



Robert Wille (robert.wille@jku.at)

Andreas Grimmer (andreas.grimmer@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

# INHALT DER VORLESUNG



## ■ Grundlagen

- ☐ Beschreibungen über „0“ und „1“ (Boolesche Algebra)
- ☐ Beschreibungen von Schaltungen

## ■ Rechnen

- ☐ Darstellung von Zahlen
- ☐ Digitale Schaltungen für Addition, Subtraktion, Multiplikation

## ■ Speichern

- ☐ Sequentielle Schaltungen
- ☐ Speicherelemente

## ■ Steuern

- ☐ Endliche Automaten
- ☐ Synthese von Steuerwerken

## ■ Entwerfen

- ☐ Synthese von allgemeinen Schaltungen
- ☐ Logikminimierung

# INHALT DER VORLESUNG



## ■ Grundlagen

- ☐ Beschreibungen über „0“ und „1“ (Boolesche Algebra)
- ☐ Beschreibungen von Schaltungen

## ■ Rechnen

- ☐ Darstellung von Zahlen
- ☐ Digitale Schaltungen für Addition, Subtraktion, Multiplikation

## ■ Speichern

- ☐ Sequentielle Schaltungen
- ☐ Speicherelemente

## ■ Steuern

- ☐ **Endliche Automaten**
- ☐ **Synthese von Steuerwerken**

## ■ Entwerfen

- ☐ Synthese von allgemeinen Schaltungen
- ☐ Logikminimierung

# STEUERN



Robert Wille (robert.wille@jku.at)

Andreas Grimmer (andreas.grimmer@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

# ENDLICHE AUTOMATEN UND DEREN UMSETZUNG



Robert Wille (robert.wille@jku.at)

Andreas Grimmer (andreas.grimmer@jku.at)

Institut für Integrierte Schaltungen

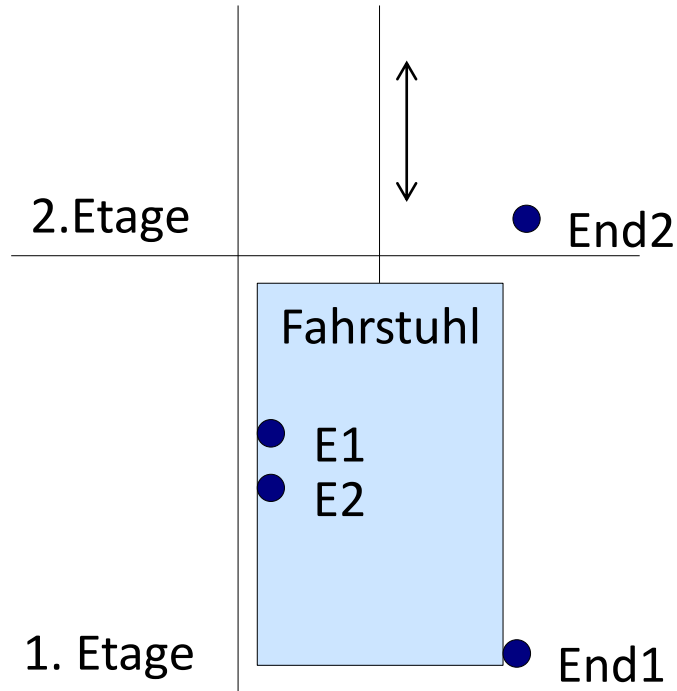
Abteilung für Schaltkreis- und Systementwurf

# ENDLICHER AUTOMAT

- Repräsentiert die Funktion einer sequentiellen Schaltung
- Mathematische Darstellung:  $A(I, S, s_0, d, O)$ 
  - ☐  $I$ ...Menge der Eingaben (inputs)
  - ☐  $S$ ...Menge der Zustände (states)
  - ☐  $s_0$ ...Initialzustand (initial state)
  - ☐  $d$ ...Zustandsübergangsfunktion (transition relation)
  - ☐  $O$ ...Menge der Ausgaben (outputs)

# AUTOMATEN – BEISPIEL #1

## Fahrstuhlsteuerung



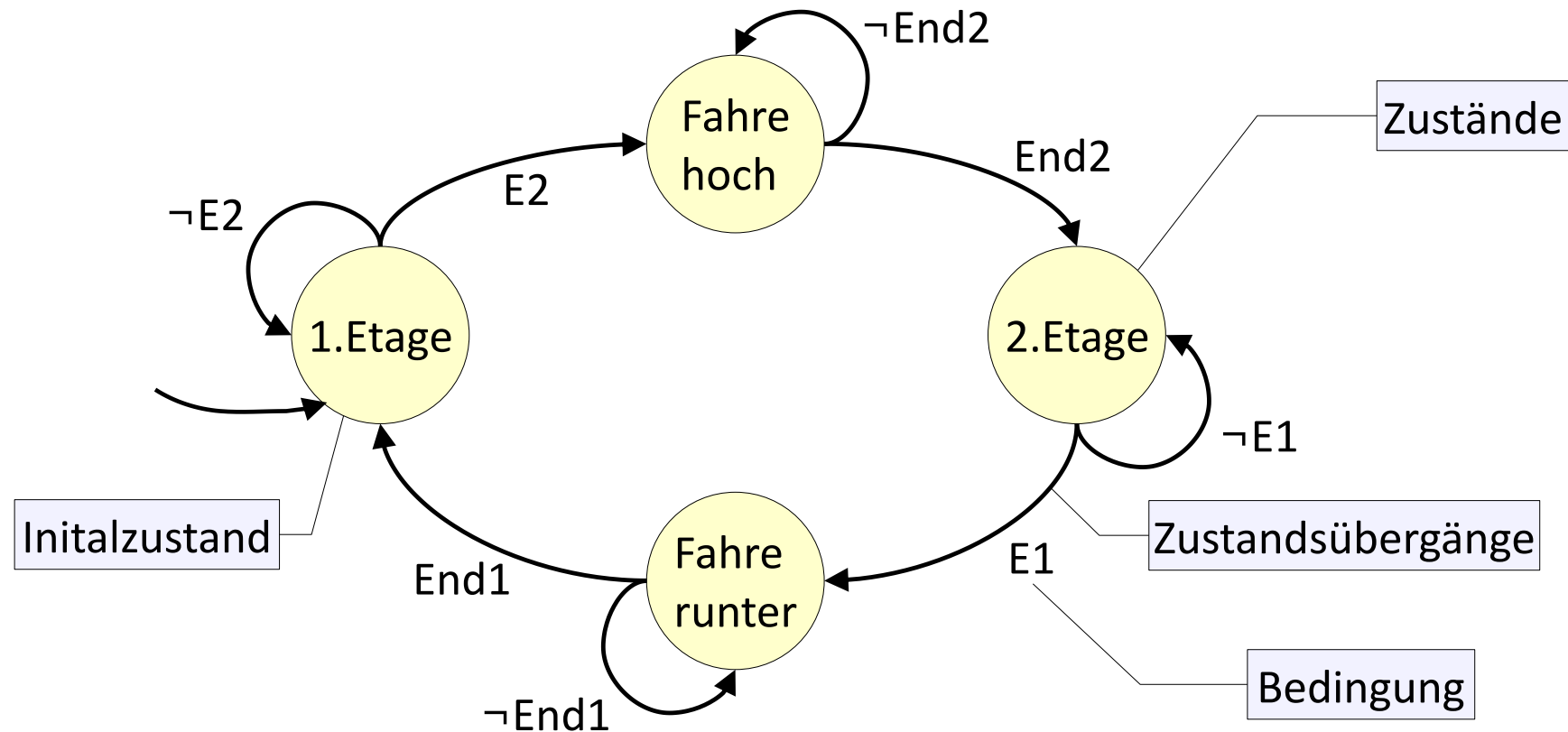
● Schalter

- Fahrstuhl zw. 2 Etagen
- Knöpfe für Etagenwahl
  - ☐ E1
  - ☐ E2
- Endschalter
  - ☐ End1, wenn Fahrstuhl in der 1. Etage angekommen ist
  - ☐ End2, wenn Fahrstuhl in der 2. Etage angekommen ist

# AUTOMATEN – BEISPIEL #2

## Zustandsdiagramm

System befindet sich immer in exakt einem Zustand

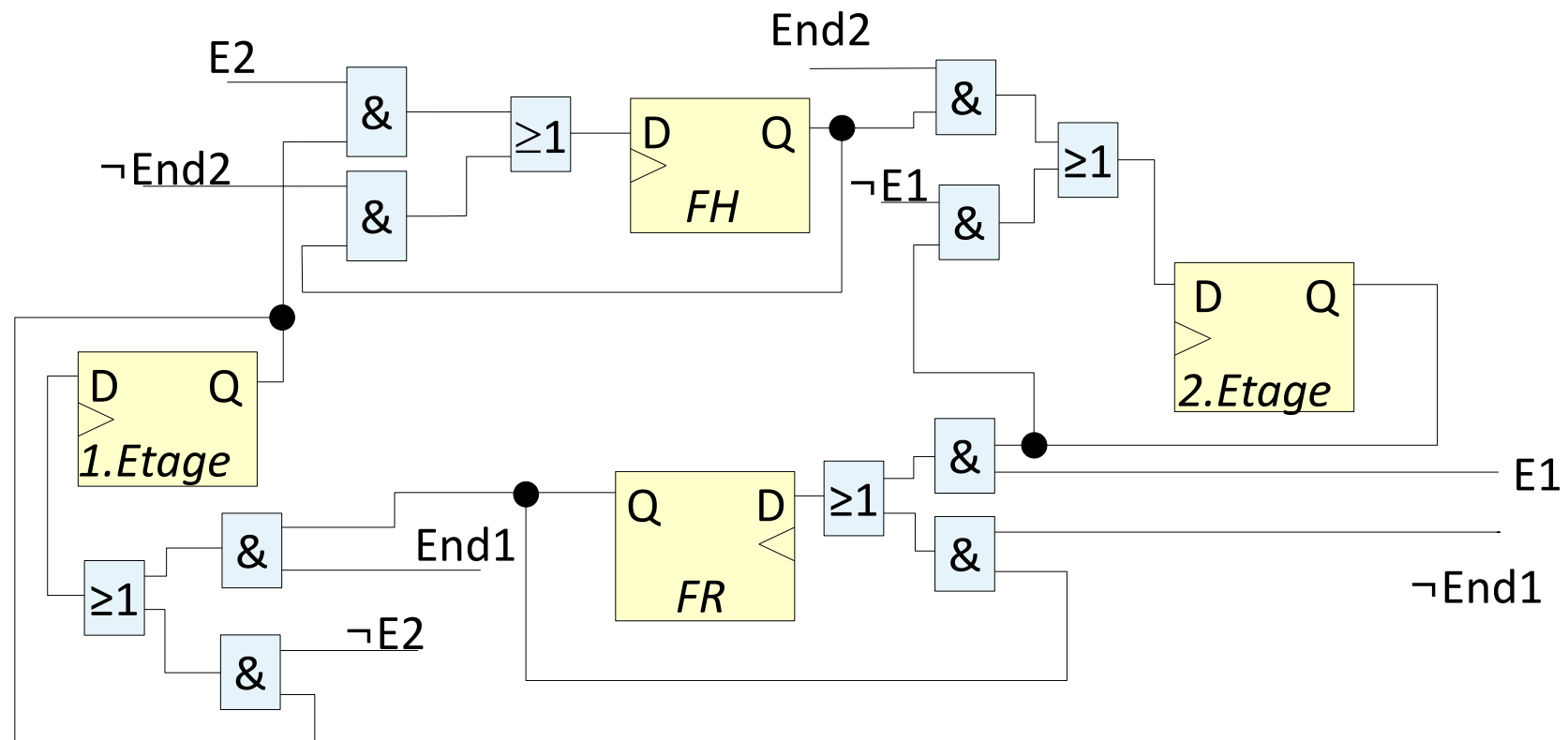




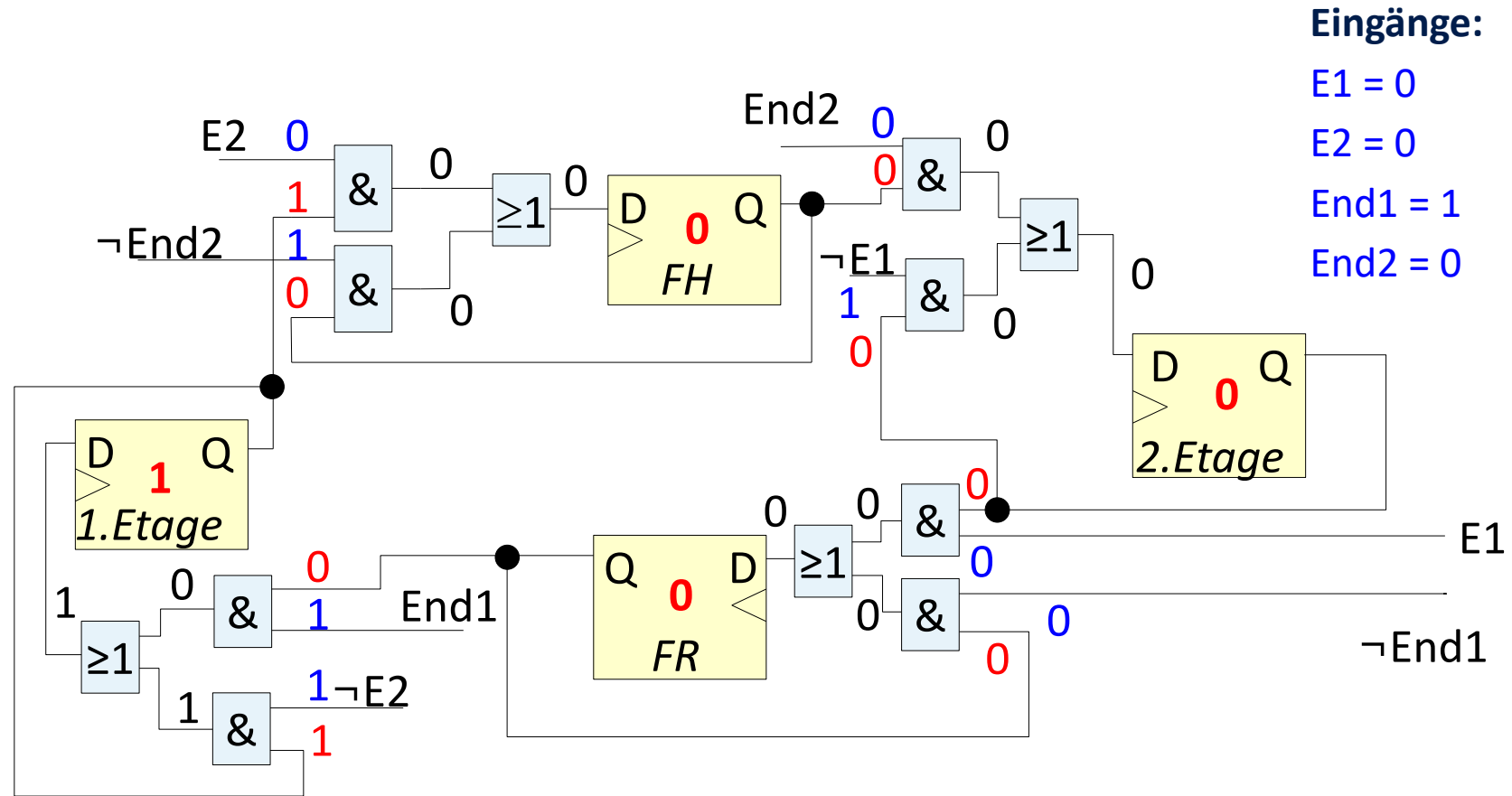
# AUTOMATEN – IMPLEMENTIERUNG

- Naheliegende Realisierung des Automaten:
  - ☐ Jeder Zustand entspricht einem Flipflops
  - ☐ Alle FF arbeiten mit dem gleichen Takt
  - ☐ Genau ein FF ist 1, alle anderen 0  
(*One-Hot* oder 1-aus-N Kodierung)
  - ☐ 1 „wandert“ zwischen den Flipflops
  - ☐ Wechsel nur wenn entsprechende Bedingung erfüllt ist  
(Und-Gatter als Tor-Schaltung)
- Andere Realisierungen möglich

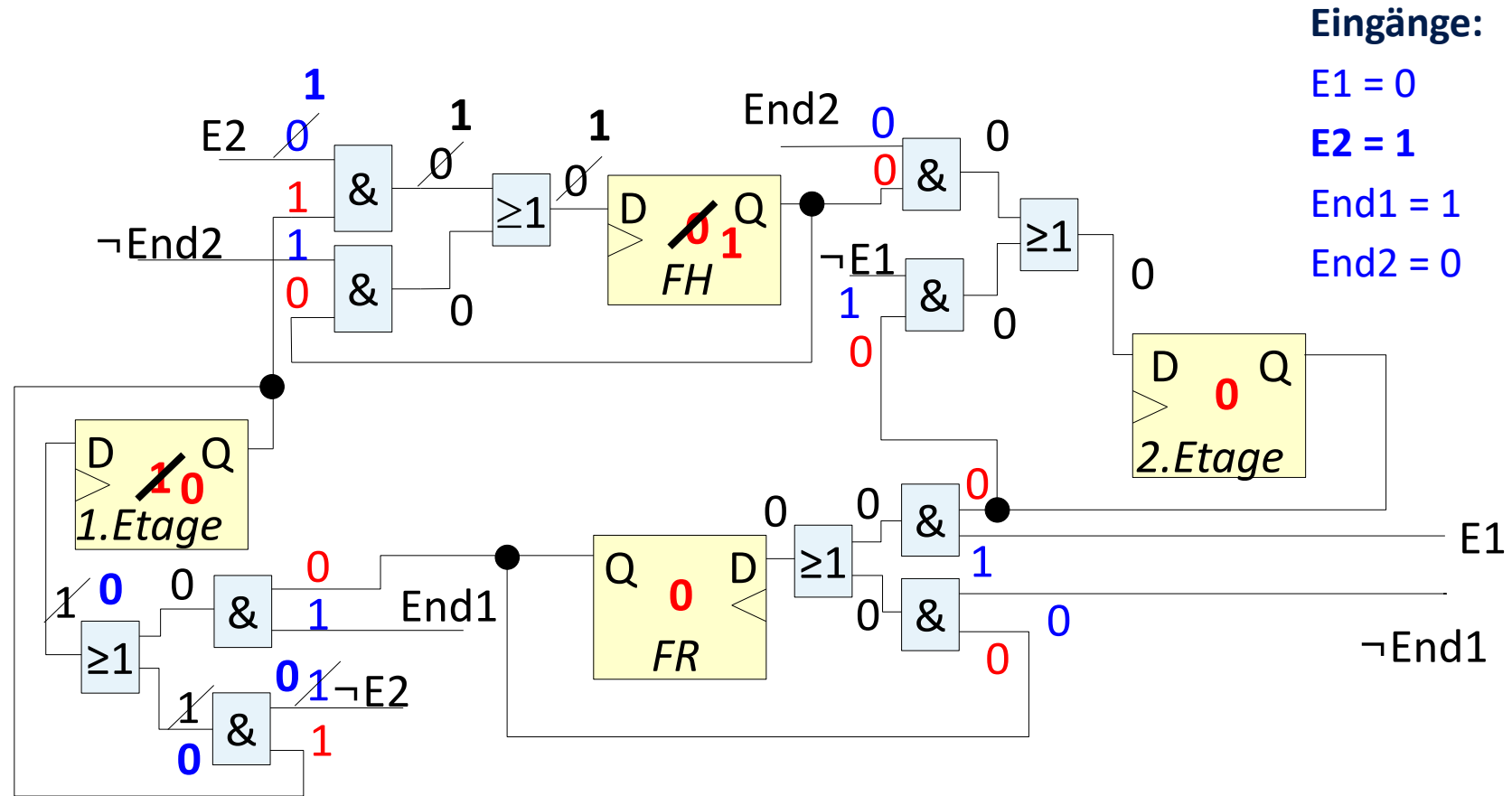
# ONE-HOT KODIERUNG



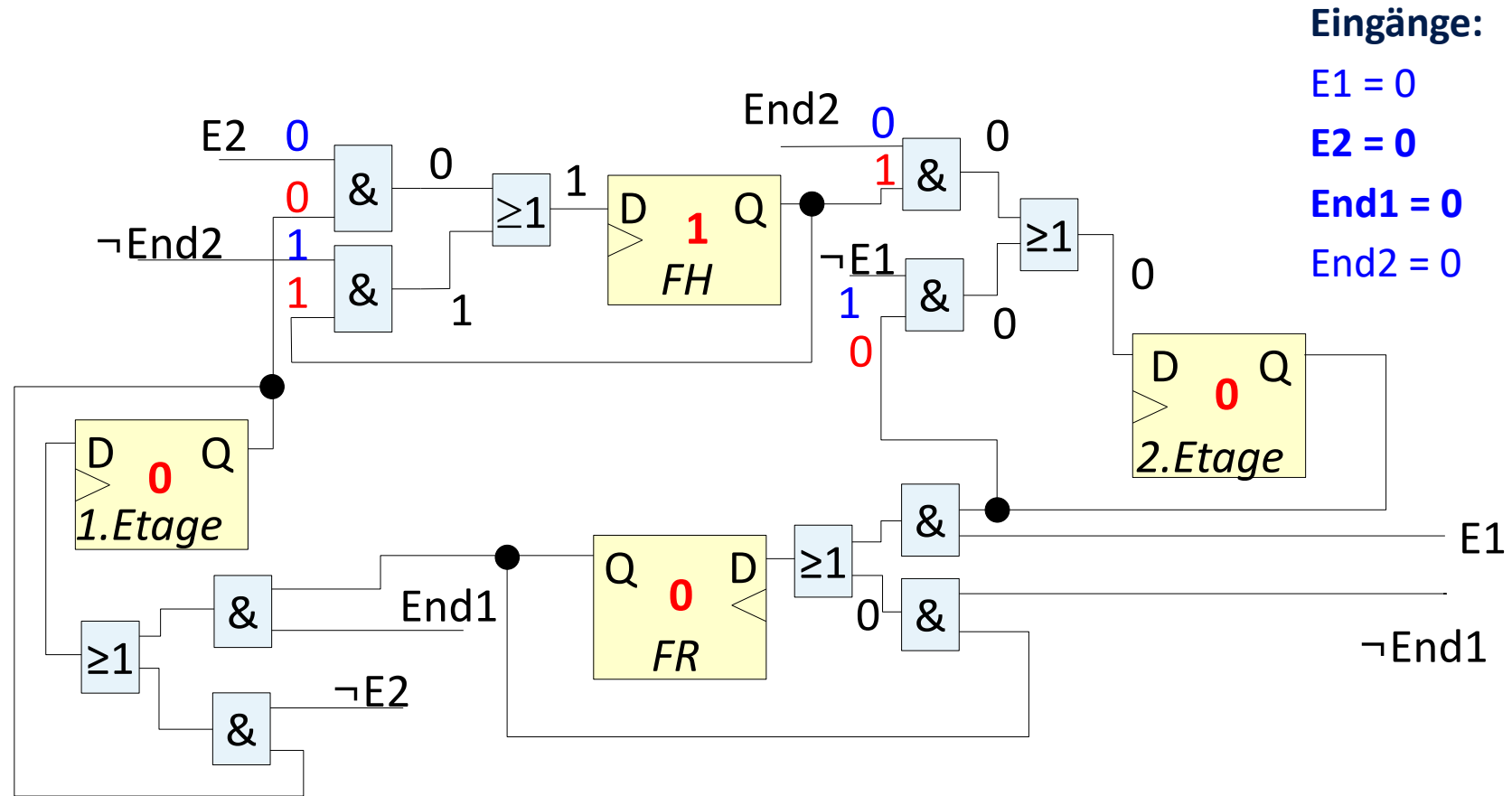
# ONE-HOT KODIERUNG – BEISPIEL #1



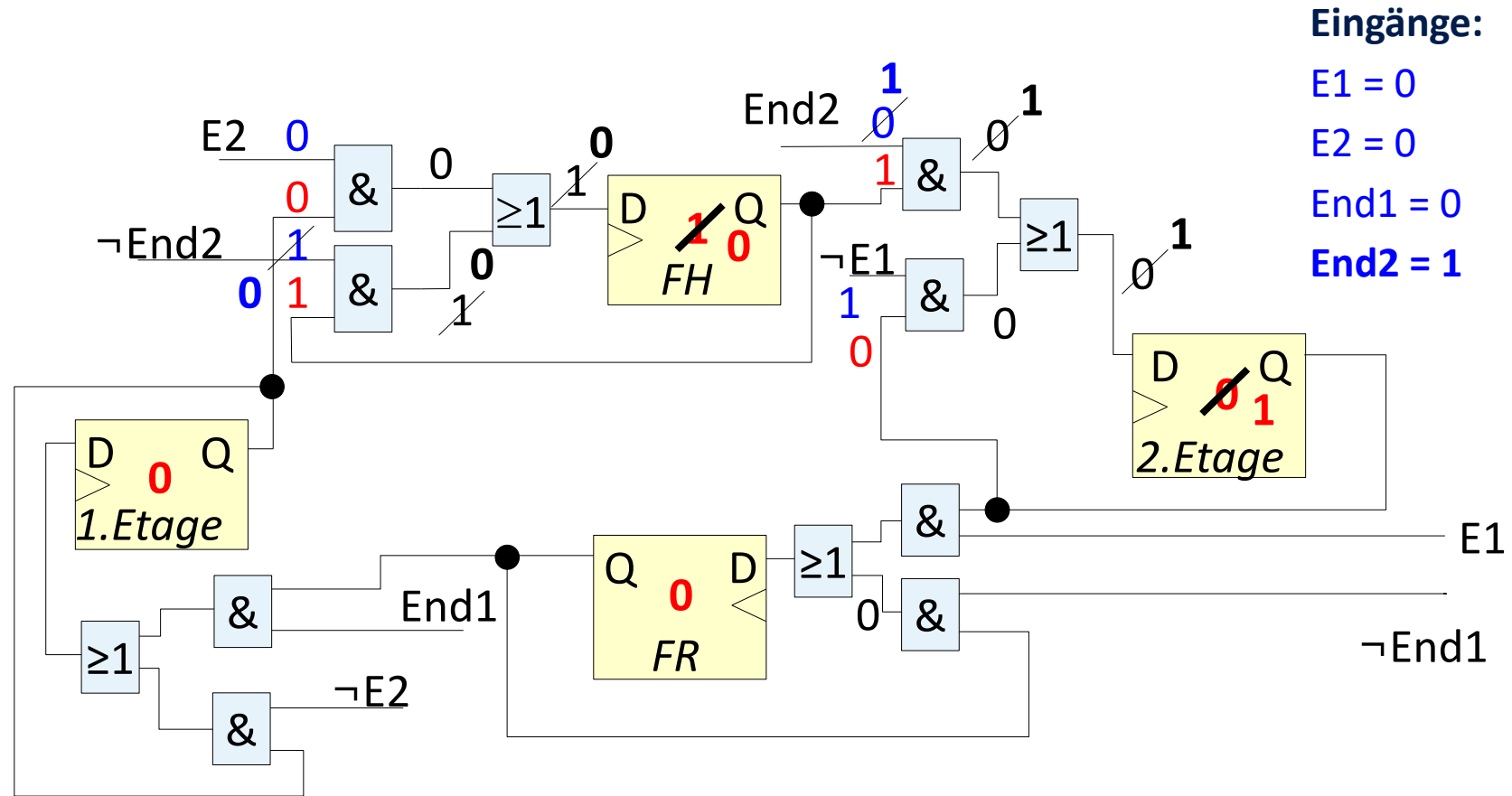
# ONE-HOT KODIERUNG – BEISPIEL #2



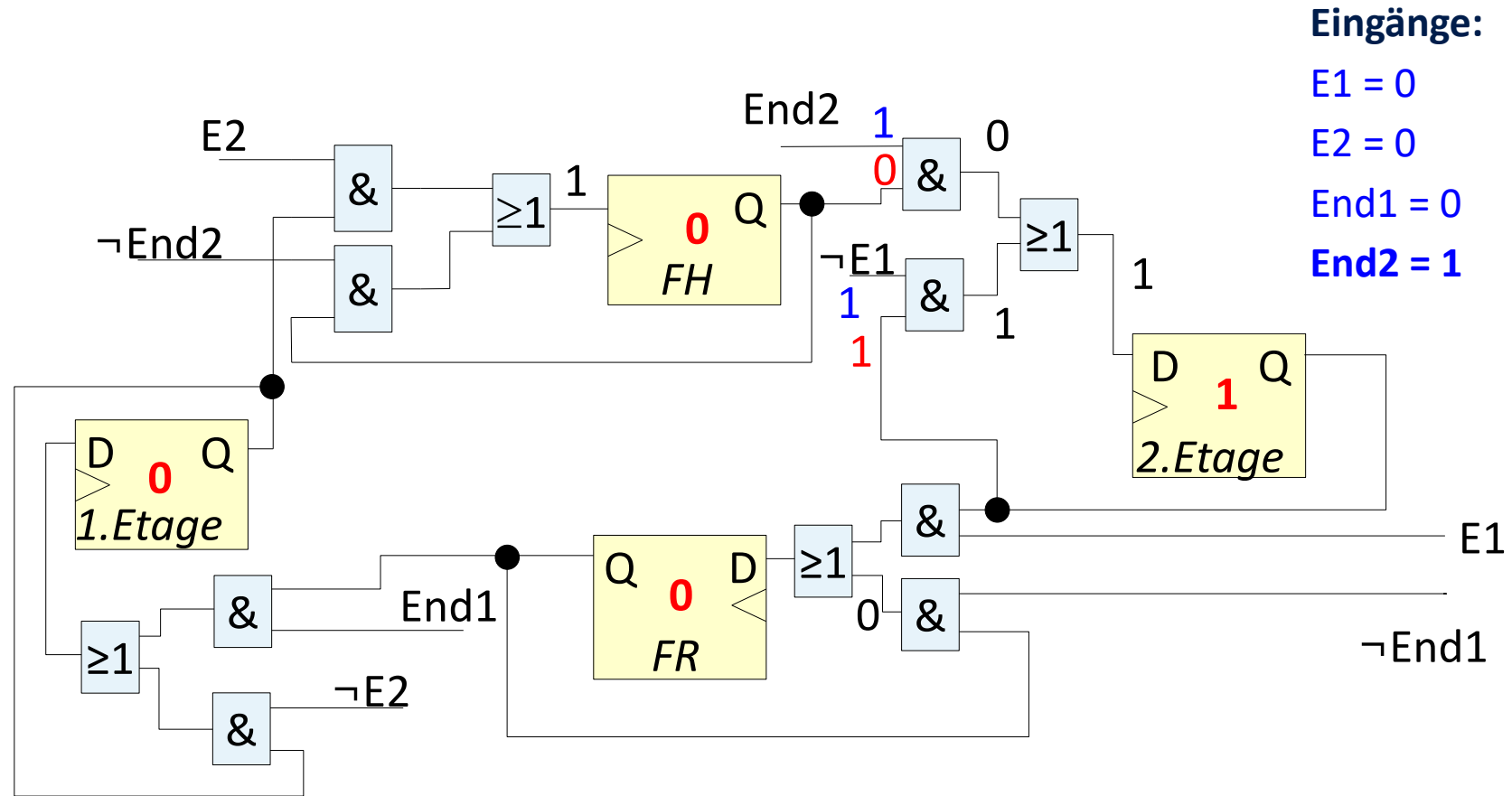
# ONE-HOT KODIERUNG – BEISPIEL #3



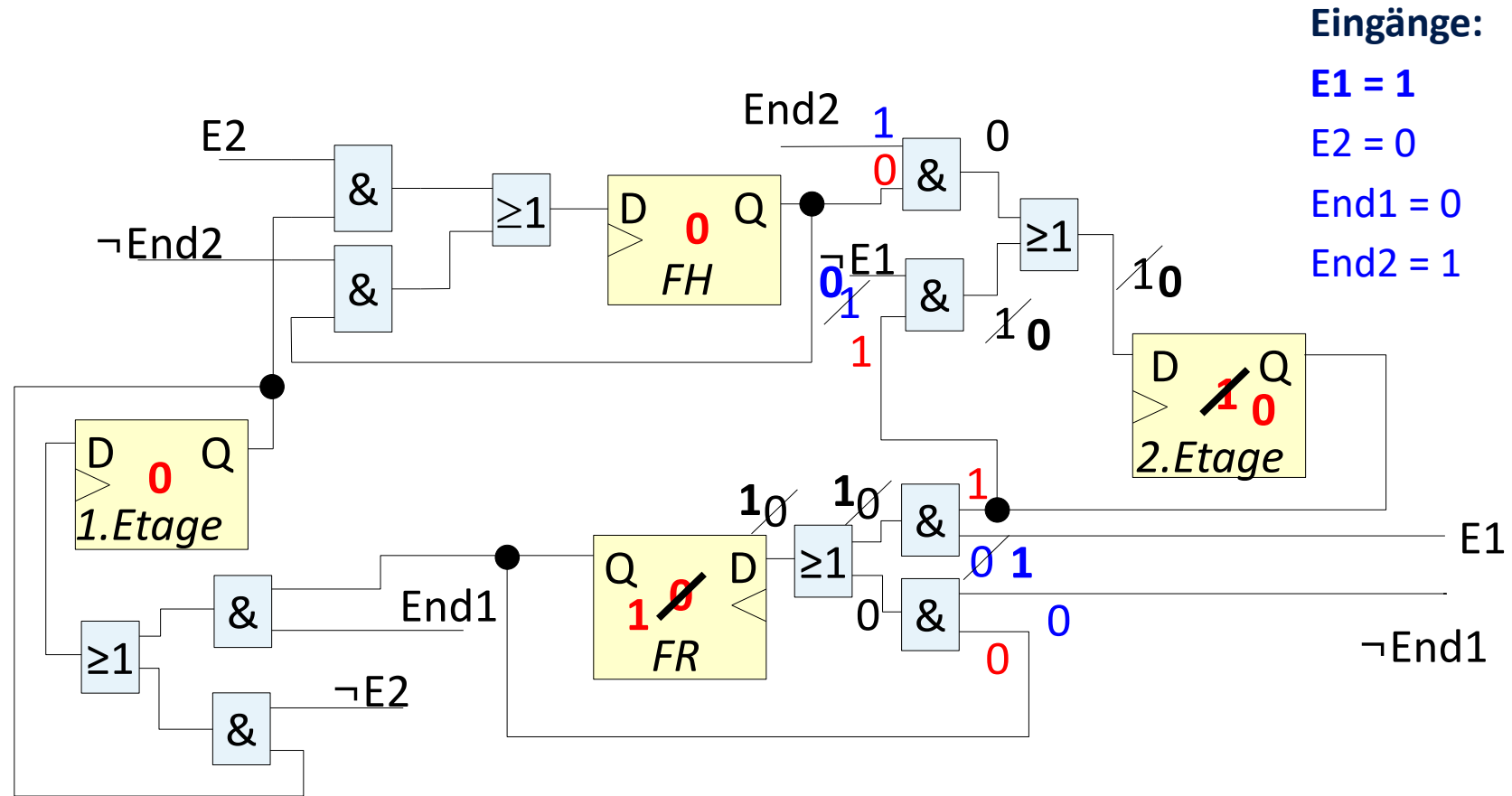
# ONE-HOT KODIERUNG – BEISPIEL #4



# ONE-HOT KODIERUNG – BEISPIEL #5

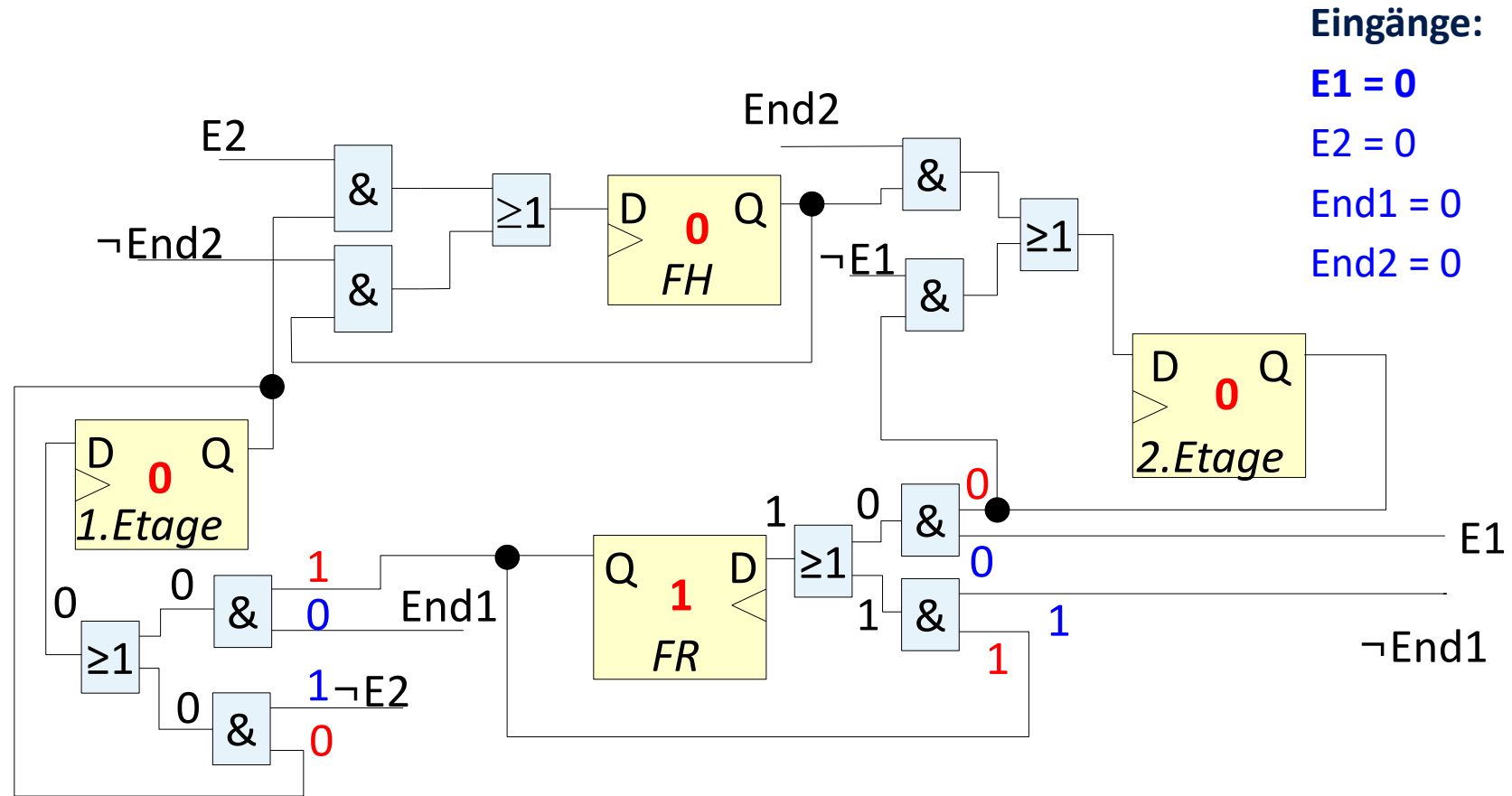


# ONE-HOT KODIERUNG – BEISPIEL #6

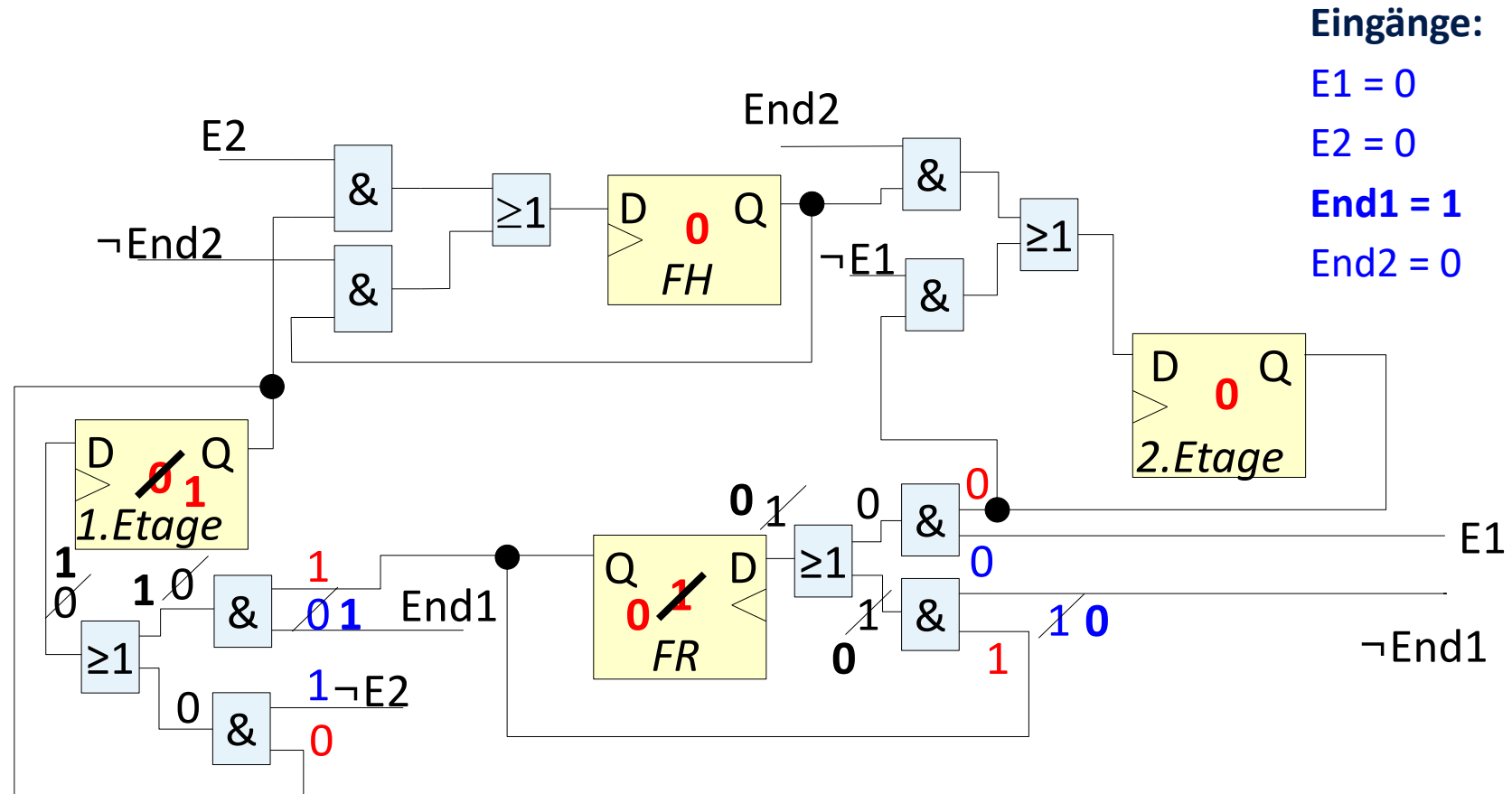




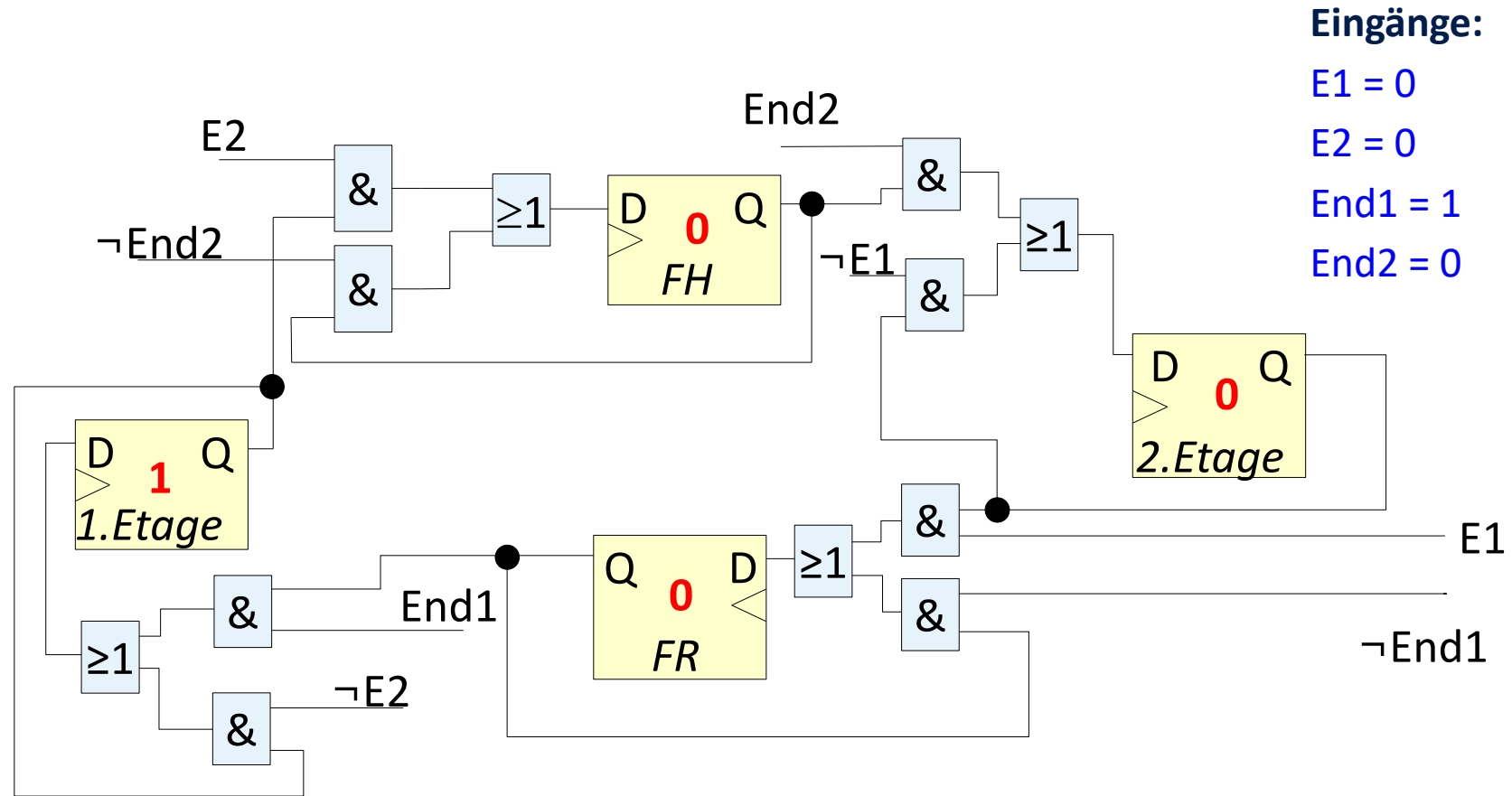
# ONE-HOT KODIERUNG – BEISPIEL #7



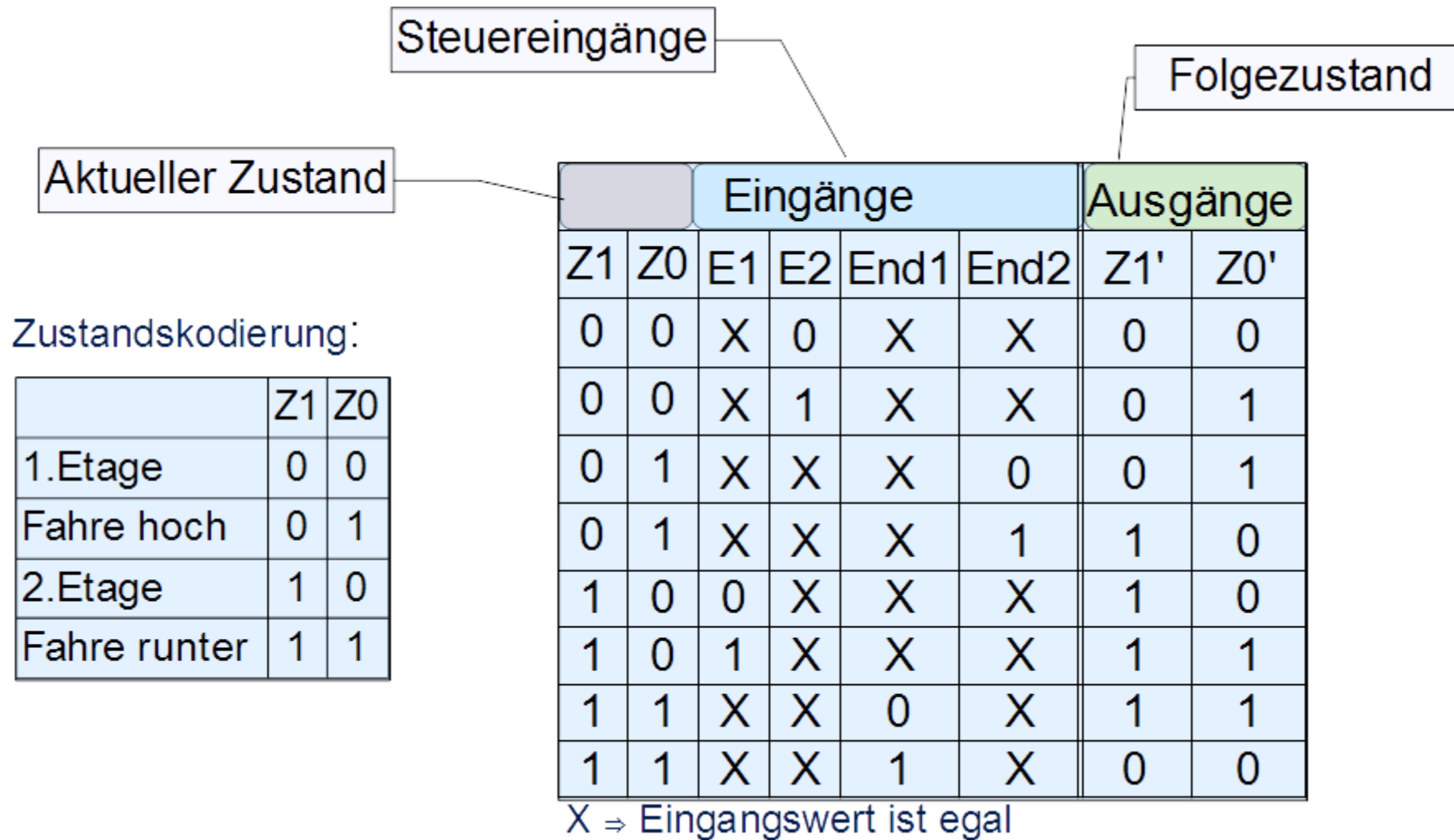
## ONE-HOT KODIERUNG – BEISPIEL #8



## ONE-HOT KODIERUNG – BEISPIEL #9

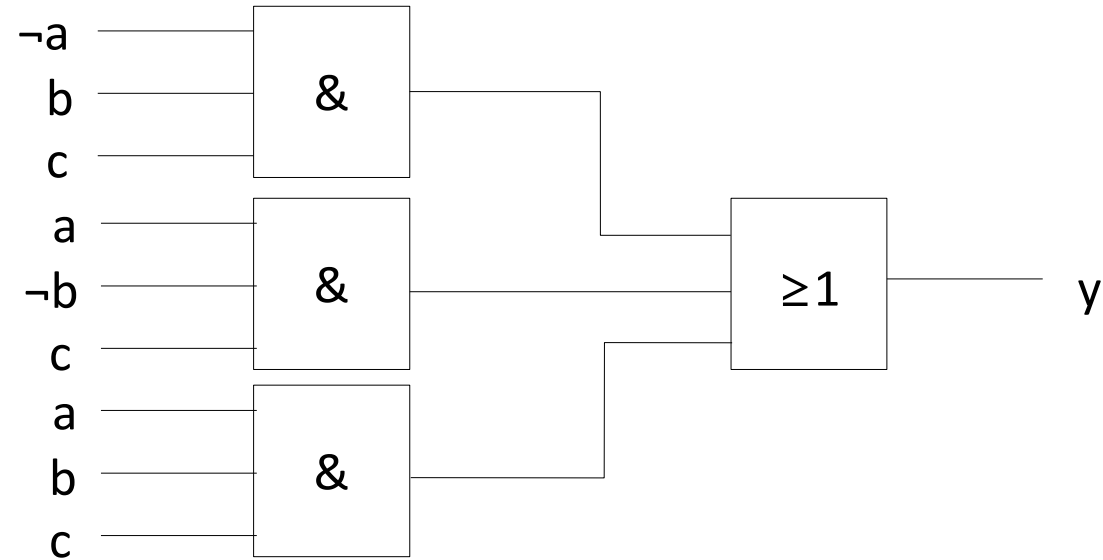
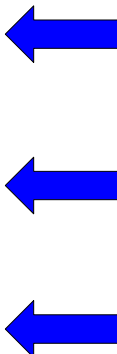


# BINÄRE KODIERUNG #1



# SYNTHESE (SIMPEL) – BEISPIEL

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



# BINÄRE KODIERUNG #2

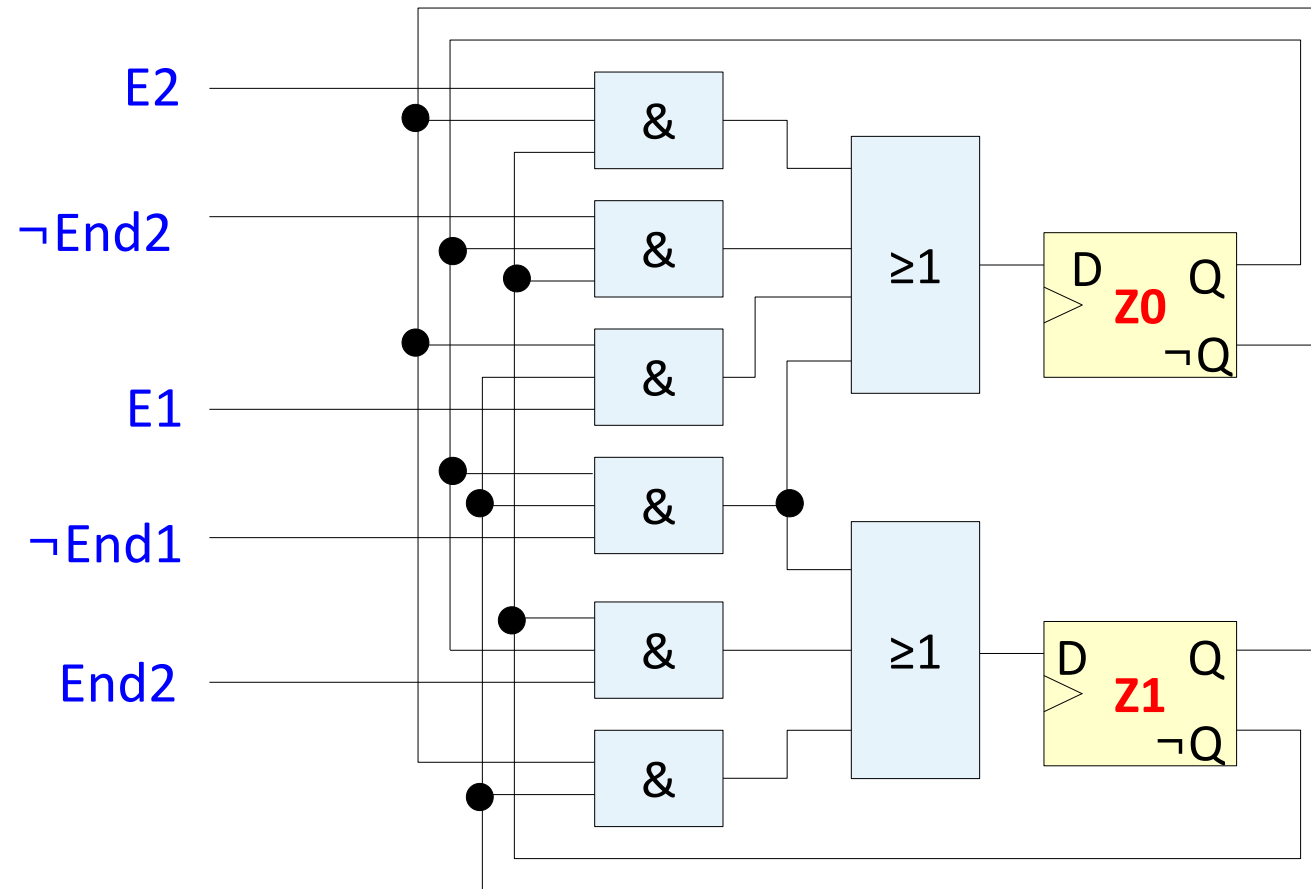
$Z0$  und  $Z1$  bilden zusammen den aktuellen Zustand  
 $Z0'$  und  $Z1'$  bilden zusammen den Folgezustand

$$Z0' = \neg Z1 \neg Z0 E2 + \neg Z1 Z0 \neg End2 + Z1 \neg Z0 E1 + Z1 Z0 \neg End1$$
$$Z1' = \neg Z1 Z0 End2 + Z1 \neg Z0 \neg E1 + Z1 \neg Z0 E1 + Z1 Z0 \neg End1$$

Zusammenfassen der beiden mittleren Produkte für  $Z1'$   
 $Z1' = \neg Z1 Z0 End2 + Z1 \neg Z0 + Z1 Z0 \neg End1$

Eingänge						Ausgänge	
Z1	Z0	E1	E2	End1	End2	Z1'	Z0'
0	0	X	0	X	X	0	0
0	0	X	1	X	X	0	1
0	1	X	X	X	0	0	1
0	1	X	X	X	1	1	0
1	0	0	X	X	X	1	0
1	0	1	X	X	X	1	1
1	1	X	X	0	X	1	1
1	1	X	X	1	X	0	0

# BINÄRE KODIERUNG - IMPLEMENTIERUNG



# **„TAKE-HOME-MESSAGE“**

- Sequentielle Schaltungen lassen sich letztlich wie kombinatorische Schaltungen entwerfen!



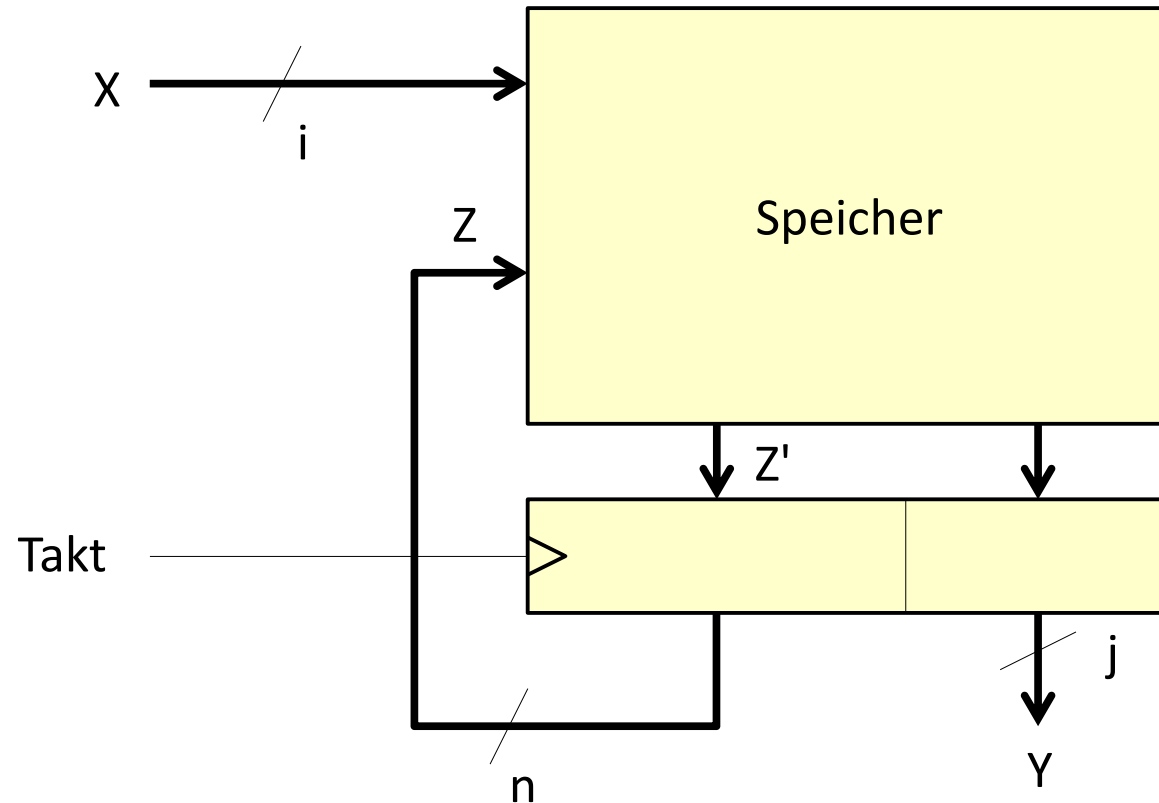
# NACHTEILE DER GATTERREALISIERUNG

- Implementierung ist aufwändig
  - ☐ Ermitteln der Zustandsübergangsgleichungen
  - ☐ Minimierung der resultierenden Schaltung erforderlich
- Struktur der Schaltung korrespondiert nicht mit Automat
  - ☐ Verifikation nicht einfach möglich
  - ☐ Kleine Änderungen am Automaten können große Änderungen an der Schaltung bewirken
- Lösung:  
Strukturiertere Implementierung des Automaten (Mikroprogramm-STW)

# MIKROPROGRAMMIERTE STEUERWERKE

- Strukturierte Implementierung von Automaten
- Realisierung der Zustandsübergangsfunktion durch einen Speicher
- Adresse wird gebildet aus
  - Zustand Z (n Bit)
  - Eingangssignale X (i Bit)
- Ausgang liefert
  - Folgezustand Z' (n Bit)
  - Ausgangssignale Y (j Bit)
- Größe des Speichers:  $2^{(n+i)} * (n+j)$  Bit

# GRUNDSTRUKTUR



Eingänge						Ausgänge	
Z1	Z0	E1	E2	End1	End2	Z1'	Z0'
0	0	X	0	X	X	0	0
0	0	X	1	X	X	0	1
0	1	X	X	X	0	0	1
0	1	X	X	X	1	1	0
1	0	0	X	X	X	1	0
1	0	1	X	X	X	1	1
1	1	X	X	0	X	1	1
1	1	X	X	1	X	0	0