

# Rechnerarchitektur

Sequenzielle Logik II

Univ.-Prof. Dr.-Ing. Rainer Böhme

Wintersemester 2021/22 · 3. November 2021

# Gliederung heute

1. **Schaltwerke**
2. Systematischer Entwurf synchroner Schaltwerke
3. Optional: Realisierung mit Transistoren

(Forts.)

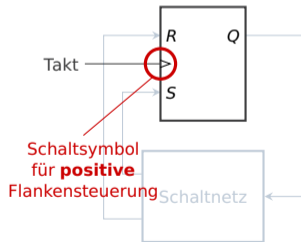
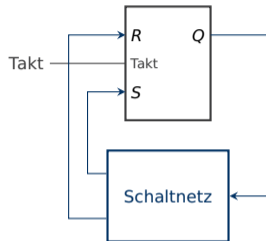
# Flankengesteuertes RS-Flipflop (W)

## Nachteil getakteter RS-Flipflops

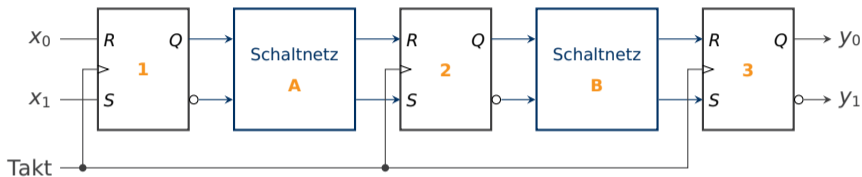
- mehrere Zustandsänderungen in einer Taktphase möglich
- erschwert kontrollierte Rückkopplung über Schaltnetz

## Alternative: Flankensteuerung

- Übernahme der Eingabewerte zu einem bestimmten Zeitpunkt
- Varianten für positive (steigende) oder negative (fallende) Flanke



# Realisierung einer „Pipeline“ (W)



- parallele Speicherung/Verarbeitung unterschiedlicher Daten
- schrittweise **Weitergabe** der Ergebnisse bei Taktflanke
- Ergebnis liegt (im Beispiel) nach drei Takten am Ausgang an.
- **Voraussetzung:** Die maximale Verzögerung aller Schaltnetze – **der kritische Pfad** – ist kürzer als ein Taktzyklus.

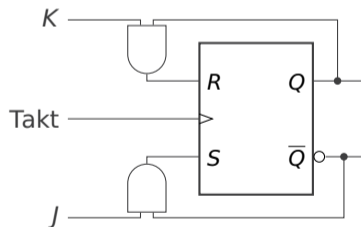
# JK-Flipflop

(Abk. für Jump/Kill oder Initialen von Jack Kilby)

## Idee

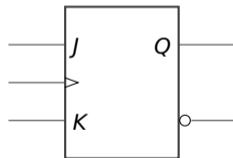
Nutzung der nicht benötigten Eingangskombination (1,1) zur Invertierung von  $Q$  (engl. *toggle*)

## Realisierung



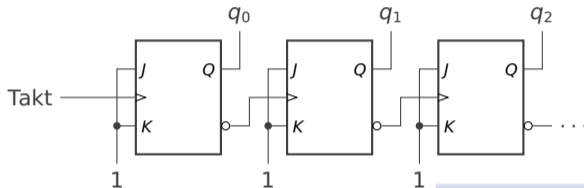
$J$	$K$	$Q'$
0	0	$Q$
0	1	0
1	0	1
1	1	$\bar{Q}$

## Symbol

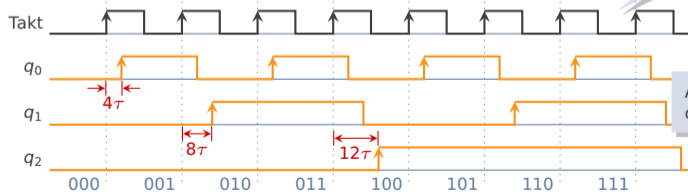


# Asynchroner 3-Bit-Zähler

Zählt positive Taktflanken in Binärdarstellung ( $q_2, q_1, q_0$ )



Zeitverhalten



Die **maximale Taktfrequenz** hängt von Verzögerung  $\tau$  ab

Anzahl der Flipflops definiert **Reichweite**

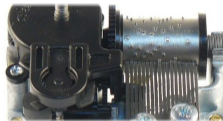
# Wichtige Unterscheidung

## Sequenzielle Logik



### Asynchrone Schaltwerke

- Steuerung durch Änderung der Eingangssignale
- Zeitpunkt stabiler Ausgangssignale abhängig von Gatterlaufzeit
- Entwurf aufwändig
- Schaltungen sehr schnell



### Synchrone Schaltwerke

- Steuerung durch zentralen Takt
- Ein- und Ausgangssignale zu festen Zeitpunkten
- systematischer Entwurf
- kritischer Pfad bestimmt maximale Taktfrequenz

# Gliederung heute

1. Schaltwerke
2. **Systematischer Entwurf synchroner Schaltwerke**
3. Optional: Realisierung mit Transistoren

(Forts.)

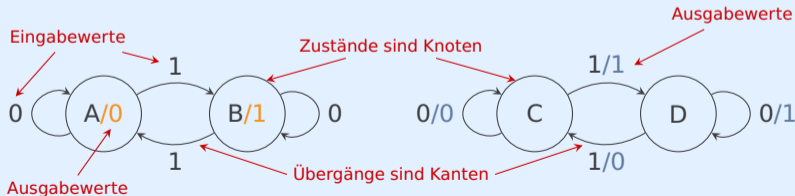


# Zustandsautomat

## Modell

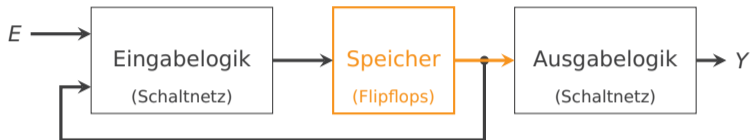
- Zu jedem diskreten Zeitpunkt  $t$  befindet sich der Automat in genau einem Zustand  $S_t$  aus der endlichen Zustandsmenge  $\mathbb{S}$ .
- Zustandsübergänge sind eine Funktion von  $S$  und Eingabe  $\mathbf{x}$ .
- Die **Ausgabe**  $\mathbf{y}$  hängt ab von:
  - entweder **nur** dem Zustand → Moore-Automat
  - oder von Zustand **und** Eingabe → Mealy-Automat

## Darstellung als gerichtete zyklische Graphen

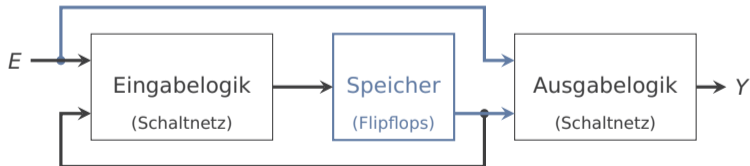


# Blockschaltbilder

## Moore-Automat



## Mealy-Automat



# Vorgehensweise beim Entwurf

1. Erstellung eines **Zustandsdiagramms**
2. Erstellung einer **Zustandstabelle**
3. Wahl einer **binären Zustandskodierung** und Ableitung der **binären Zustandstabelle**
4. Auswahl der Flipflop-Typen und Ermittlung der **Ansteuerlogik** für jeden Zustandsübergang
5. Ermittlung der **Ausgabegleichungen**
6. **Minimierung** der Ansteuer- und Ausgabegleichungen
7. **Realisierung** des Schaltwerks

# Beispiel: Entwurf eines synchronen Schaltwerks

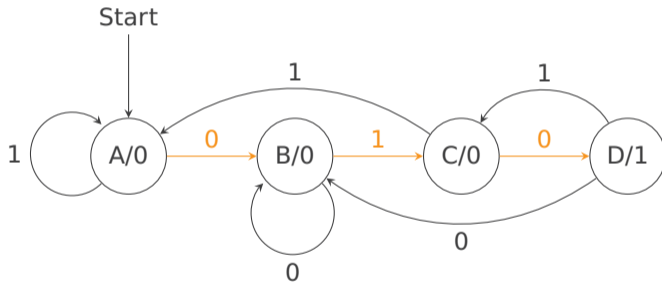
## Aufgabenstellung

*„Entwerfen Sie ein synchrones Schaltwerk zur Erkennung der Sequenz 010 im binären Eingabestrom  $x(t)$ . Die Ausgabe  $y$  nehme den Wert 1 an, sobald im Eingabestrom die Sequenz 010 erkannt wurde. Sonst sei  $y = 0$ .“*

## Vorgehensweise

- ausführliches Vorgehen für einen Moore-Automaten
- Betrachtung der Unterschiede im Falle eines Mealy-Automaten

# Schritt 1: Zustandsdiagramm



## Schritt 2: Zustandstabelle

Die Zustandstabelle enthält für jeden (symbolischen) Zustand  $S \in \mathbb{S}$

- **Folgezustand**  $S'$  in Abhängigkeit von der Eingabe
- **Ausgabe**

(im Bsp.:  $x$ )

(im Bsp.:  $y$ )

$S$	$x$	$S'$	$y$
A	0	B	0
A	1	A	0
B	0	B	0
B	1	C	0
C	0	D	0
C	1	A	0
D	0	B	1
D	1	C	1

# Schritt 3: binär kodierte Zustandstabelle

Umkodierung von  $S$  in **binäre Zustände**  $Q \subseteq \{0, 1\}^k$   
mit  $k = \lceil \log_2 |S| \rceil$ .

$S$	$x$	$S'$	$y$		$q_1$	$q_0$	$x$	$q'_1$	$q'_0$	$y$
A	0	B	0		0	0	0	0	1	0
A	1	A	0		0	0	1	0	0	0
B	0	B	0		0	1	0	0	1	0
B	1	C	0	→	0	1	1	1	0	0
C	0	D	0		1	0	0	1	1	0
C	1	A	0		1	0	1	0	0	0
D	0	B	1		1	1	0	0	1	1
D	1	C	1		1	1	1	1	0	1

# Schritt 4: Flipflop-Typen und Ansteuerlogik

Darstellung aller Zustandsübergänge  $Q_i \rightarrow Q'_i$  über Flipflops

**Ansteuerungstabelle** für zwei flankengesteuerte **JK-Flipflops**

$q_1$	$q_0$	$x$	$q'_1$	$q'_0$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	1	0	d	1	d
0	0	1	0	0	0	d	0	d
0	1	0	0	1	0	d	d	0
0	1	1	1	0	1	d	d	1
1	0	0	1	1	d	0	1	d
1	0	1	0	0	d	1	0	d
1	1	0	0	1	d	1	d	0
1	1	1	1	0	d	0	d	1

Der Schaltungsaufwand hängt von der Wahl der Flipflop-Typen ab.



# Schritt 5: Ausgabegleichungen

Beim Moore-Automat kann die binäre Zustandstabelle auf alle eindeutigen Zeilen von  $Q$  (ohne  $Q'$ ) reduziert werden.

$q_1$	$q_0$	$y$
0	0	0
0	1	0
1	0	0
1	1	1

Daraus lassen sich Boolesche Funktionen als Ausgabegleichungen bestimmen:

$$y = q_0 \cdot q_1$$

(in diesem Beispiel bereits minimal)

# Schritt 6: Minimierung der Ansteuergleichungen

- Flipflop des niederwertigen Bits der Zustandskodierung  $q_0$

$$J_0 = \bar{x} \quad K_0 = x$$

(direkt aus der Ansteuerungstabelle, siehe Schritt 4)

- Flipflop des höherwertigen Bits der Zustandskodierung  $q_1$

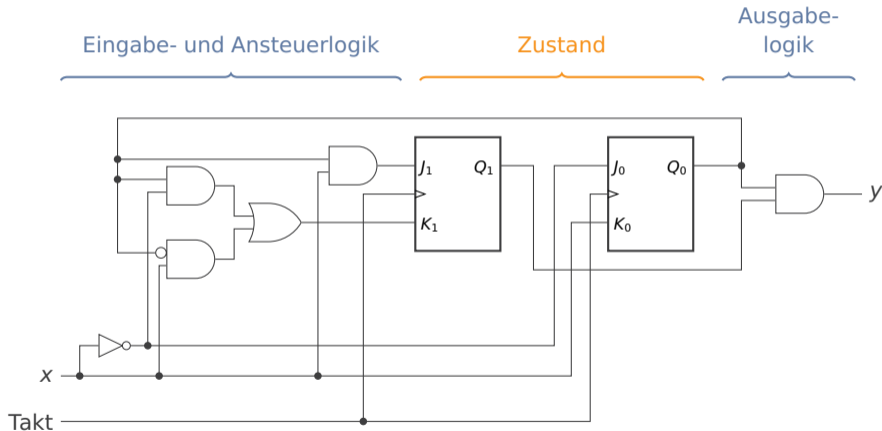
$$J_1 = q_0 \cdot x$$

	$q_1$			
	0	<b>d</b>	<b>d</b>	0
$x$	0	<b>d</b>	<b>d</b> 1	
	$q_0$			

$$K_1 = q_0 \cdot \bar{x} + \bar{q}_0 \cdot x$$

	$q_1$			
	<b>d</b>	0	<b>1</b> <b>d</b>	
$x$	<b>d</b> 1	0	<b>d</b>	
	$q_0$			

# Schritt 7: Realisierung des Schaltwerks



# Beispiel: Entwurf eines synchronen Schaltwerks

## Aufgabenstellung

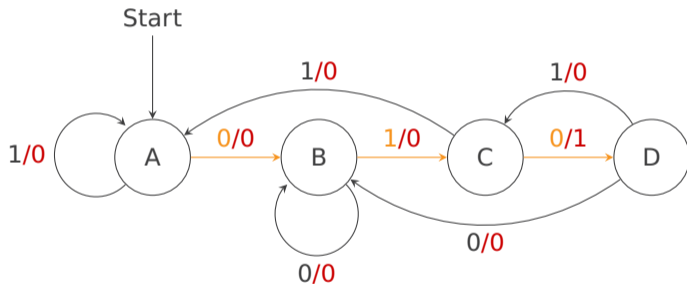
*„Entwerfen Sie ein synchrones Schaltwerk zur Erkennung der Sequenz 010 im binären Eingabestrom  $x(t)$ . Die Ausgabe  $y$  nehme den Wert 1 an, sobald im Eingabestrom die Sequenz 010 erkannt wurde. Sonst sei  $y = 0$ .“*

## Vorgehensweise

- ausführliches Vorgehen für einen Moore-Automaten
- **Betrachtung der Unterschiede im Falle eines Mealy-Automaten**

# Schritt 1: Zustandsdiagramm

bei Modellierung als **Mealy-Automat**



→ Markierung der Kanten (statt Knoten) mit Ausgabe  $y$

# Schritte 2 & 3: Zustandstabellen

Die prinzipielle Vorgehensweise ist analog zum Moore-Automat.

Beim **Mealy**-Automat ändert sich jedoch die Ausgabe  $y$  **im gleichen Takt**, indem sich die Eingabe  $x$  ändert.

$S$	$x$	$S'$	$y$		$q_1$	$q_0$	$x$	$q'_1$	$q'_0$	$y$
A	0	B	0		0	0	0	0	1	0
A	1	A	0		0	0	1	0	0	0
B	0	B	0		0	1	0	0	1	0
B	1	C	0	→	0	1	1	1	0	0
C	0	D	1		1	0	0	1	1	1
C	1	A	0		1	0	1	0	0	0
D	0	B	0		1	1	0	0	1	0
D	1	C	0		1	1	1	1	0	0

# Schritte 4 bis 6: Logik, Minimierung

- Ermittlung der **Ansteuerlogik**

(hier im Beispiel unverändert zum Moore-Automat)

- Ermittlung der **Ausgabegleichungen**

Abhängig von Zustand  $Q$  und Eingabe  $x$

$$y = \overline{q_0} \cdot q_1 \cdot \overline{x}$$

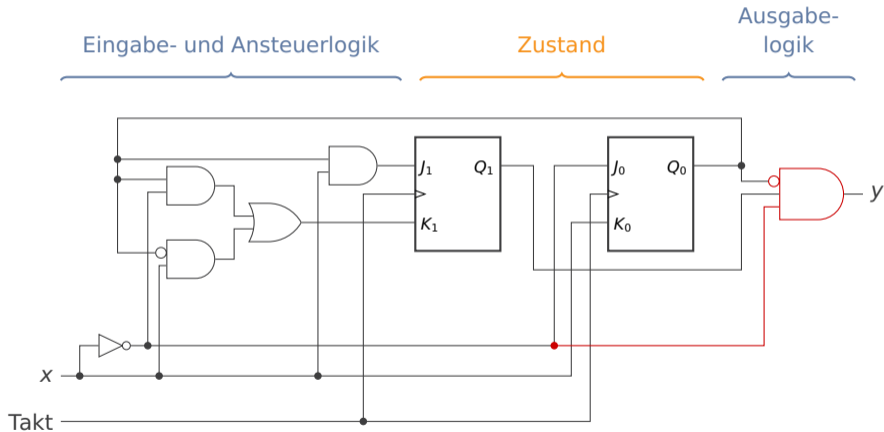
$q_1$	$q_0$	$x$	$y$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

- **Minimierung**

(hier im Beispiel unverändert bzw. bereits minimal)

# Schritt 7: Realisierung des Schaltwerks

(jetzt als **Mealy**-Automat)





# Hörsaalfragen



24 82 94 16

Welche Eigenschaften treffen eher auf Moore- oder Mealy-Automaten zu ?

1. Schnelle Reaktion auf Veränderungen der Eingabesignale
2. Geringere Anzahl von Flipflops, wenn Zustände durch Übergänge mit verschiedenen Ausgaben erreicht werden können
3. Zum Entwurf beliebiger Schaltwerke geeignet
4. Geringer Schaltungsaufwand der Ausgabelogik
5. Asynchrone Störungen der Eingabesignale wirken sich niemals auf die Ausgabe aus.

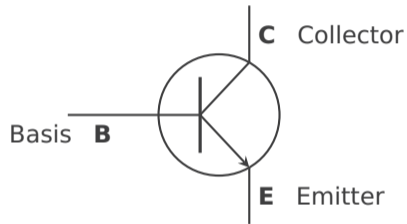
Zugang: <https://arsnova.uibk.ac.at> mit Zugangsschlüssel **24 82 94 16**. Oder scannen Sie den QR-Kode.

# Gliederung heute

1. Schaltwerke
2. Systematischer Entwurf synchroner Schaltwerke
3. **Optional: Realisierung mit Transistoren**

(Forts.)

# Transistor

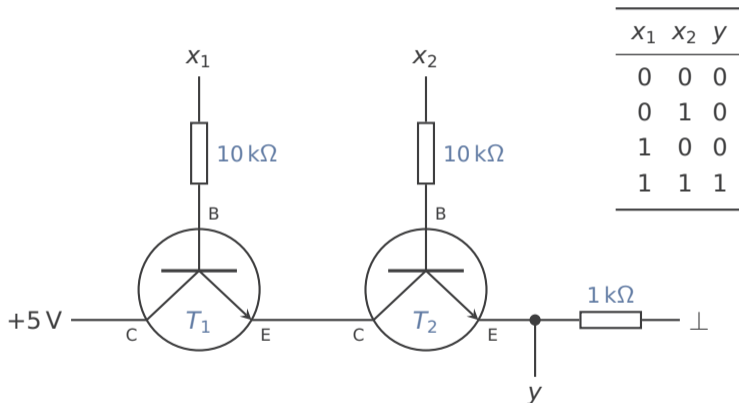


## Transistor (engl. transfer resistor)

Strom fließt von C nach E genau dann wenn ein vernachlässigbar kleiner Steuerstrom von B nach E fließt.

# Gatter-Schaltung

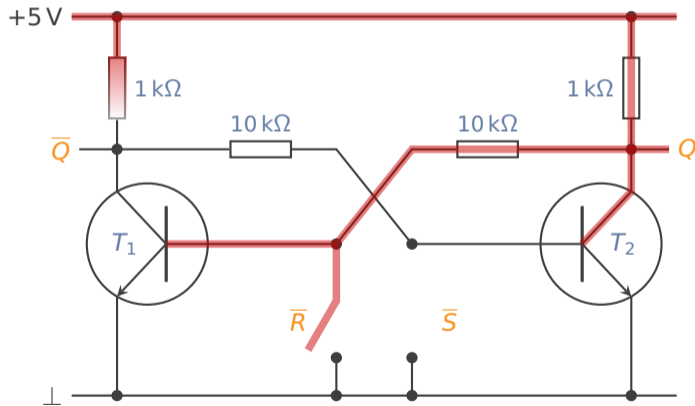
Realisierung eines AND-Gatters



# Flipflop-Schaltung

(engl. auch "latch")

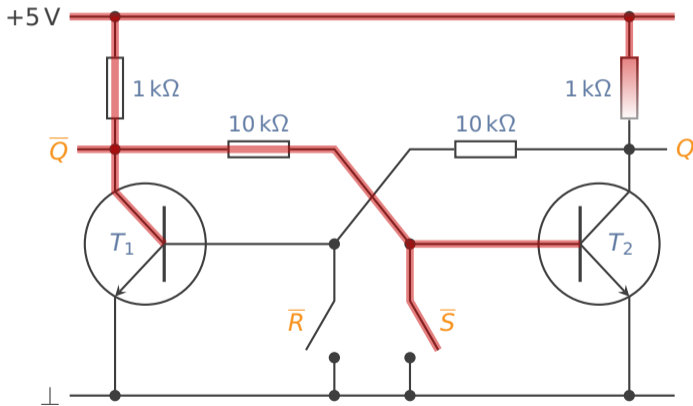
Realisierung mit zwei Bipolartransistoren



# Flipflop-Schaltung

(engl. auch "latch")

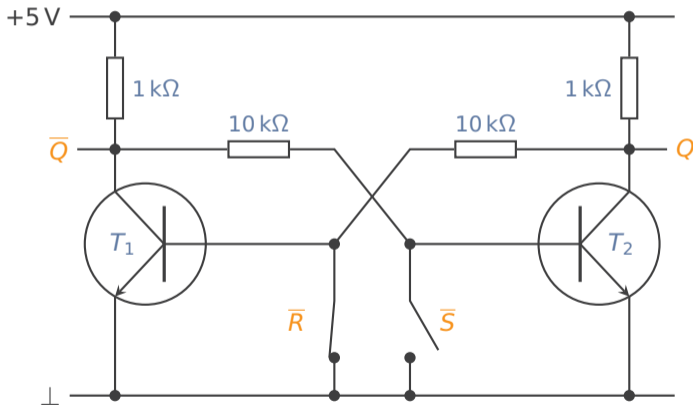
Realisierung mit zwei Bipolartransistoren



# Flipflop-Schaltung

(engl. auch "latch")

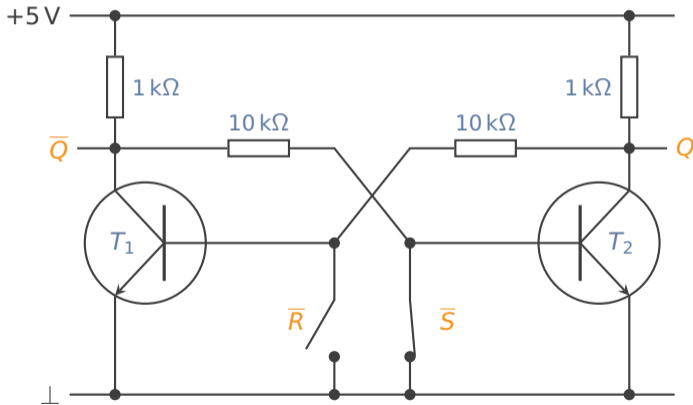
Realisierung mit zwei Bipolartransistoren



# Flipflop-Schaltung

(engl. auch "latch")

Realisierung mit zwei Bipolartransistoren





# Flipflops als integrierte Schaltungen

Realisierung in frühen integrierten Schaltkreisen

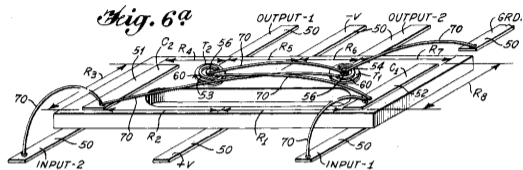
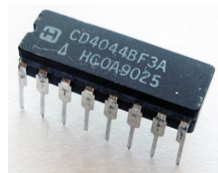
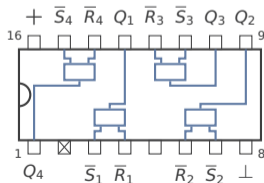


Fig. 4. Germanium flip-flop using mesa transistors, bulk resistors, diffused capacitors, and air isolation of the components. From US Patent 3,138,743.

Standardbauteil 4044 mit Vierfach-RS-Flipflop (engl. Quad RS Latches)



Bildquellen: J. Kilby, IEEE, West Florida Components

# Syllabus – Wintersemester 2021/22

06.10.21	1. Einführung	
13.10.21	2. Kombinatorische Logik I	
20.10.21	3. Kombinatorische Logik II	
27.10.21	4. Sequenzielle Logik I	
03.11.21	5. Sequenzielle Logik II	
10.11.21	6. Arithmetik I	
17.11.21	7. Arithmetik II	später →
24.11.21	8. Befehlssatzarchitektur (ARM) I	<b>inday students</b>
01.12.21	9. Befehlssatzarchitektur (ARM) II	
15.12.21	10. Ein-/Ausgabe	
12.01.22	11. Prozessorarchitekturen	
19.01.22	12. Speicher	
26.01.22	13. Leistung	
<b>02.02.22</b>	<b>Klausur (1. Termin)</b>	