

Rechnerstrukturen

2. Grundlagen

© 1997 Peter Sturm, Universität Trier

Inhalt

- ◆ Elektronische Schalter
- ◆ Elementare Gatterfunktionen
- ◆ Schaltnetze
- ◆ Schaltwerke

2.2

© 1997 Peter Sturm, Universität Trier

Motivation

- ◆ Unterscheidung von zwei Zuständen
 - Strom / kein Strom
 - Spannung / keine Spannung
 - Positiv / Negativ geladen
 - Reflektierend / nicht reflektierend
 - ...
- ◆ Technische Umsetzung
 - Mechanisch
 - Elektromechanisch
 - Elektronisch
 - Licht
- ◆ Abbildung auf
 - binäre Zahlen
 - Wahrheitswerte
 - Zeichen



2.3

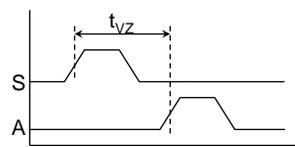
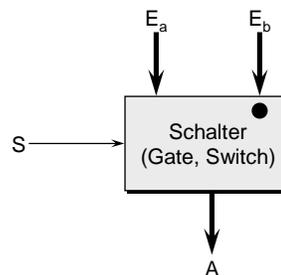
© 1997 Peter Sturm, Universität Trier

Elektronische Schalter

- ◆ Elementar Wechselschalter


```

            if S then
              A := Ea
            else
              A := Eb
            
```
- ◆ Vereinfachungen
 - Ein Eingang konstant 0 oder 1
- ◆ Verzögerungszeit t_{vZ}
 - wird durch die eingesetzte Technologie bestimmt
 - wenige Nanosekunden typisch

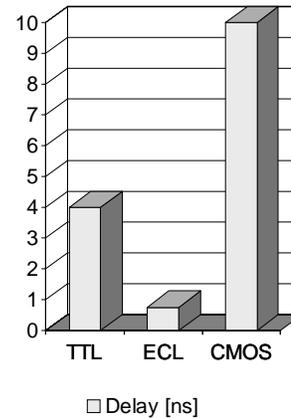


2.4

© 1997 Peter Sturm, Universität Trier

Realisierungsvarianten

- ◆ Transistor-Transistor-Logik (TTL)
- ◆ Emitter-Coupled-Logik (ECL)
- ◆ Metal Oxid Semiconductor (MOS)
- ◆ Unterschiede
 - Verzögerungszeit
 - Versorgungsspannung
 - 5 V (TTL)
 - 2.8 - 3.3 V (MOS)
 - Integrationsdichte
 - 0.35 μm - 0.25 μm (MOS)
 - 0.18 μm demnächst (MOS)
(vgl. Haar = 100 μm)
- ◆ Materialien
 - Silicium, Aluminium
 - in Zukunft ev. GaAs, Kupfer

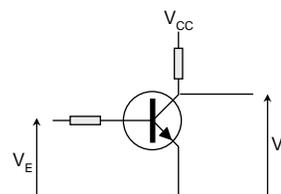
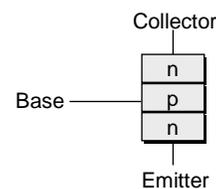


© 1997 Peter Sturm, Universität Trier

2.5

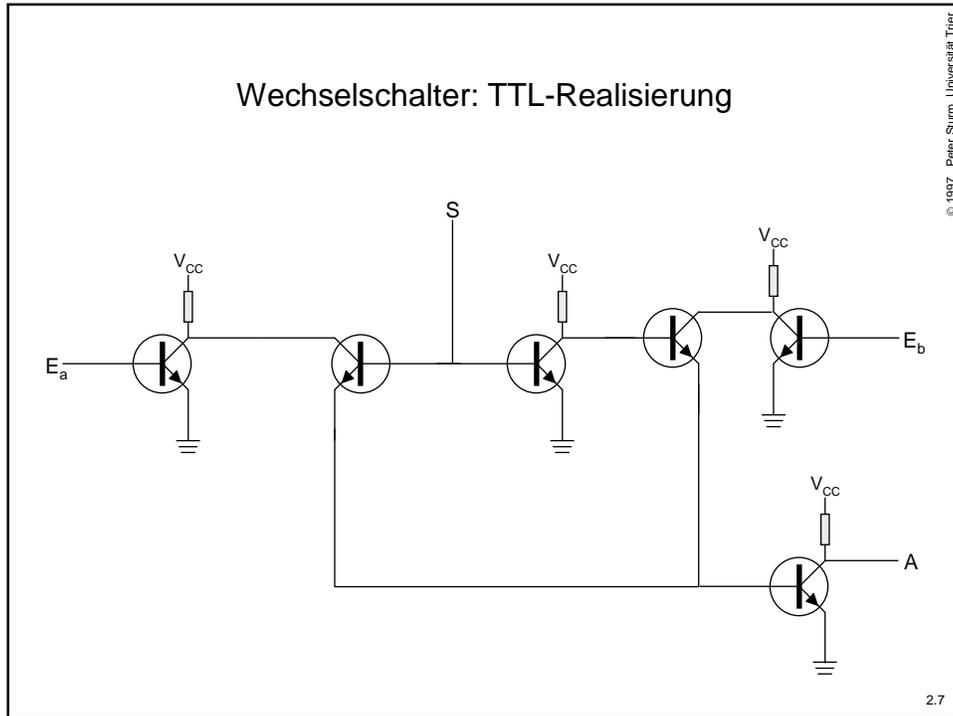
Bipolare Transistoren

- ◆ npn- und pnp-Transistor
 - Dotierung von Silicium
- ◆ Schaltverhalten
 - $V_E = 0 \text{ V}$, $V_A = 3 \text{ V}$
 - $V_E = 3 \text{ V}$, $V_A = 0.2 \text{ V}$
- ◆ Hohe Ströme
 - p-Kanal muß mit Elektronen gefüllt werden
 - Wärmeentwicklung
 - Dicke der Leiterbahnen
- ◆ Geringe Kapazitäten
- ◆ Hohe Schaltgeschwindigkeit



© 1997 Peter Sturm, Universität Trier

2.6



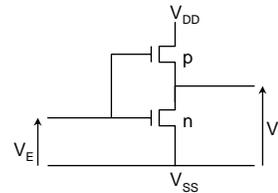
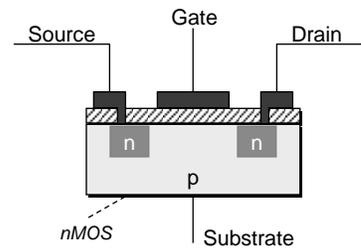
TTL-Familien

74xx	Standard TTL	30 mW/Gate	9 ns t_{VZ}
74Lxx	Low-power TTL	1	33
74Hxx	High-speed TTL	22	6
74Sxx	Schottky TTL	20	3
74LSxx	Low-power Schottky TTL	2	9
74ASxx	Advanced Schottky TTL	20	1.6
74ALSxx	Advanced Low-power Schottky TTL	1.3	5
74Fxx	Fast TTL	5	3

© 1997 Peter Sturm, Universität Trier
2.8

Feldeffekttransistoren

- ◆ FETs
 - NMOS (n-Kanal)
 - PMOS (p-Kanal)
 - CMOS (complementary MOS)
 - NMOS und PMOS paarweise
- ◆ Schaltverhalten
 - $V_E = 0V, V_A = V_{DD}$
 - $V_E = V_{DD}, V_A = 0V$
- ◆ Längere Verzögerungszeit
 - Kapazitives Element
- ◆ Geringe Spannungswerte
- ◆ Extrem geringe Ströme



© 1997 Peter Sturm, Universität Trier

2.9

Wechselschalter: CMOS-Realisierung

© 1997 Peter Sturm, Universität Trier

2.10

State of the Art

- ◆ z.B. DEC Alpha 21264
 - 0.35µm CMOS-Technik
 - 3.02 cm²
 - 15200000 Transistoren
 - 600 MHz Taktfrequenz
 - 72 W Stromverbrauch

0.35 µm = 1 cm

49 km

Belichten Ätzen Bedampfen, Dotieren

© 1997 Peter Sturm, Universität Trier

2.11

Tri-State

- ◆ Direkte Verschaltung von Ausgängen kritisch
- ◆ Zugang mehrerer Elemente zu einem gemeinsamen Bus
- ◆ Dritter, hochohmiger Zustand der Ausgänge
 - Steuerung durch zusätzlichen Eingang (CS = Chip Select)

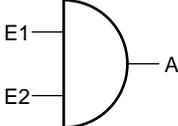
© 1997 Peter Sturm, Universität Trier

2.12

Die Grundgatter

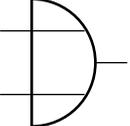
◆ UND

E1	E2	A
0	0	0
0	1	0
1	0	0
1	1	1



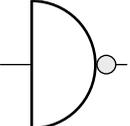
◆ ODER

E1	E2	A
0	0	0
0	1	1
1	0	1
1	1	1



◆ NICHT

E1	A
0	1
1	0



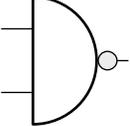
© 1997 Peter Sturm, Universität Trier

2.13

NAND, NOR, XOR

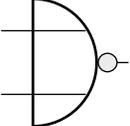
◆ NAND

E1	E2	A
0	0	1
0	1	1
1	0	1
1	1	0



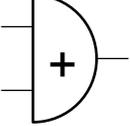
◆ NOR

E1	E2	A
0	0	1
0	1	0
1	0	0
1	1	0



◆ XOR

E1	E2	A
0	0	0
0	1	1
1	0	1
1	1	0



© 1997 Peter Sturm, Universität Trier

2.14

Äquivalenz

- ◆ Ist XOR ein Grundgatter?
- ◆ Wieviele Grundgatter braucht man minimal, um beliebige boolesche Ausdrücke zu beschreiben?

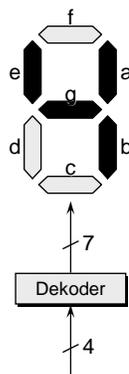
Schaltnetze

- ◆ Schaltung aus logischen Grundgattern mit
 - n Eingängen
 - m Ausgängen
 - Rückkopplungsfrei
- ◆ 2^n Eingangskombinationen
- ◆ m boolesche Ausdrücke über $E_1 \dots E_n$
- ◆ Wahrheitstabelle



Wahrheitstabelle 7-Segment-Dekoder

- ◆ 4 Eingänge
 - Binärzahl $e_3 e_2 e_1 e_0$
- ◆ 7 Ausgänge
 - Austeuerung der Segment a bis g



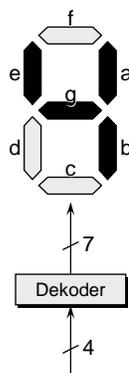
e_3	e_2	e_1	e_0	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

2.18

© 1997 Peter Sturm, Universität Trier

Wahrheitstabelle 7-Segment-Dekoder

- ◆ 4 Eingänge
 - Binärzahl $e_3 e_2 e_1 e_0$
- ◆ 7 Ausgänge
 - Austeuerung der Segment a bis g



e_3	e_2	e_1	e_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

2.19

© 1997 Peter Sturm, Universität Trier

7-Segment-Dekoder (cont.)

- ◆ Wie erhält man boolesche Ausdrücke für a bis g?

$$a(e_3, e_2, e_1, e_0) =$$

$$b(e_3, e_2, e_1, e_0) =$$

⋮

$$g(e_3, e_2, e_1, e_0) =$$

2.20

© 1997 Peter Sturm, Universität Trier

7-Segment-Dekoder (cont.)

- ◆ Gatterschaltung:

2.22

© 1997 Peter Sturm, Universität Trier

Disjunktive und konjunktive Normalformen

- ◆ Disjunktive Normalform
 - Summe von Produkten (minterme)
- ◆ Konjunktive Normalform
 - Produkt von Summen (maxterme)
- ◆ Minterm
 - 1 Zeile in Wahrheitstabelle
 - Eingänge mit 1: e
 - Eingänge mit 0: e negiert
- ◆ Maxterm
 - 0 Zeile in Wahrheitstabelle
 - Eingänge mit 1: e negiert
 - Eingänge mit 0: e

e2	e1	e0	a
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

2.24

© 1997 Peter Sturm, Universität Trier

Disjunktive und konjunktive Normalformen

- ◆ Disjunktive Normalform
 - Summe von Produkten (minterme)
- ◆ Konjunktive Normalform
 - Produkt von Summen (maxterme)
- ◆ Minterm
 - 1 Zeile in Wahrheitstabelle
 - Eingänge mit 1: e
 - Eingänge mit 0: e negiert
- ◆ Maxterm
 - 0 Zeile in Wahrheitstabelle
 - Eingänge mit 1: e negiert
 - Eingänge mit 0: e

$$\bar{e}_2 e_1 \bar{e}_0 + \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 + e_2 e_1 \bar{e}_0$$

e2	e1	e0	a
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

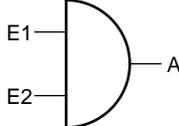
$$(e_2 + e_1 + e_0) \cdot (e_2 + e_1 + \bar{e}_0) \cdot (\bar{e}_2 + e_1 + \bar{e}_0) \cdot (\bar{e}_2 + \bar{e}_1 + \bar{e}_0)$$

2.25

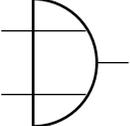
© 1997 Peter Sturm, Universität Trier

Boolesche Algebra

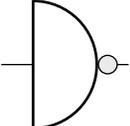
- ◆ Dualität zwischen
 - Gatterschaltungen
 - Booleschen Ausdrücken
- ◆ Positive Logik
 - 0 = False
 - 1 = True
- ◆ Gesetze



$A = E1 \cdot E2$



$A = E1 + E2$



$A = \bar{E}$

© 1997 Peter Sturm, Universität Trier

2.26

Gesetze

- ◆ Operationen mit 0 und 1
 - $X + 0 = X, X + 1 = 1$
 - $X \cdot 0 = 0, X \cdot 1 = X$
- ◆ Idempotenz
 - $X + X = X$
 - $X \cdot X = X$
- ◆ Komplementärgesetz
 - $X + \bar{X} = 1$
 - $X \cdot \bar{X} = 0$

© 1997 Peter Sturm, Universität Trier

2.27

Gesetze (cont.)

◆ Kommutativitätsgesetz

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

◆ Assoziativitätsgesetz

$$(X + Y) + Z = X + (Y + Z)$$

$$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$$

◆ Distributivgesetz

$$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$$

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

2.28

© 1997 Peter Sturm, Universität Trier

Gesetze (cont.)

◆ Vereinfachungsgesetze

$$X \cdot Y + X \cdot \bar{Y} = X, (X + Y) \cdot (X + \bar{Y}) = X$$

$$X + X \cdot Y = X, X \cdot (X + Y) = X$$

$$(X + \bar{Y}) \cdot Y = X \cdot Y, (X \cdot \bar{Y}) + Y = X + Y$$

◆ DeMorgan's Gesetz

$$\overline{X + Y + \dots + Z} = \bar{X} \cdot \bar{Y} \cdot \dots \cdot \bar{Z}$$

$$\overline{X \cdot Y \cdot \dots \cdot Z} = \bar{X} + \bar{Y} + \dots + \bar{Z}$$

2.29

© 1997 Peter Sturm, Universität Trier

Umformung und Minimierung

- ◆ Gründe
 - Begrenzungen bei der Schaltungstiefe
 - Minimaler Materialeinsatz
 - Bestimmter Gattervorrat
 - Elektrische Eigenschaften
 - Platzbeschränkungen
 - ...
- ◆ Algebraische Umformungen
- ◆ Algorithmische Verfahren
 - Karnaugh-Diagramme (1-4 Eingangsvariablen, 1 Ausgang)
 - Quine-McCluskey-Verfahren (n Eingänge, 1 Ausgang)
 - Bündelminimierung (n Eingänge, m Ausgänge)

2.30

© 1997 Peter Sturm, Universität Trier

Karnaugh-Diagramme

- ◆ Graphische Methode
- ◆ max. 4 Eingänge ABCD
- ◆ Übertragung der Wahrheitstabelle
 - Position im Diagramm entspricht ABCD als Binärzahl

A	B	C	a
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

= (2, 3, 4, 6) =

		A	
		0	1
B	0	0	2
	1	1	3

		A			
		00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5

		A			
		00	01	11	10
C	0	0	1	1	1
	1	0	1	0	0

2.31

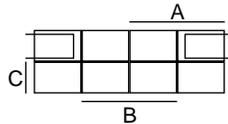
© 1997 Peter Sturm, Universität Trier

Minimierung im Karnaugh-Diagramm

- ◆ **Legende = Gray-Code**
 - benachbarte Zeilen und Spalten ändern sich nur in einem Bit
- ◆ **Zusammenfassen von Gruppen zu 1, 2, 4 oder 8 Einsen**
 - 1 = keine Minimierung
 - 2 = Term mit 2 aus 3 Eingängen
 - 4 = Term mit 1 aus 3 Eingängen
 - 8 = Funktion konstant 1
- ◆ **Diagramm als Torus auffassen!**

		A			
		00	01	11	10
0		0	1	1	1
1		0	1	0	0
		B			

$$= \bar{A}B + B\bar{C} + A\bar{C}$$



2.32

© 1997 Peter Sturm, Universität Trier

4 Eingänge

- ◆ **Zusammenfassen von Gruppen zu 1, 2, 4, 8 oder 16 Einsen**
 - 1 = keine Minimierung
 - 2 = Term mit 3 aus 4 Eingängen
 - 4 = Term mit 2 aus 4 Eingängen
 - 8 = Term mit 1 aus 4 Eingängen
 - 16 = Funktion konstant 1
- ◆ **Torus**

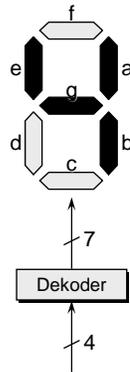
		A			
		00	01	11	10
00		0	4	12	8
01		1	5	13	9
11		3	7	15	11
10		2	6	14	10
		B			

2.33

© 1997 Peter Sturm, Universität Trier

Minimierung 7-Segment-Dekoder

- ◆ 4 Eingänge
 - Binärzahl e3 e2 e1 e0
- ◆ 7 Ausgänge
 - Austeuerung der Segment a bis g

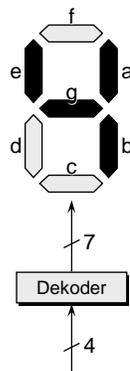


e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

2.34

Minimierung 7-Segment-Dekoder (Don't Care)

- ◆ 4 Eingänge
 - Binärzahl e3 e2 e1 e0
- ◆ 7 Ausgänge
 - Austeuerung der Segment a bis g



e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

2.39

Bemerkungen

- ◆ Bei bis zu vier Eingangsvariablen ideale Minimierungstechnik
- ◆ 5 und 6 Eingänge theoretisch auch möglich
 - 5: zwei übereinander liegende 4er-Diagramme
 - 6: vier übereinander liegende 4er-Diagramme
 - insgesamt aber praktisch nicht handhabbar
- ◆ Konjunktive Minimalform ebenfalls möglich
 - Zusammenfassen der 0-Gruppen

© 1997 Peter Sturm, Universität Trier

2.44

Quine-McCluskey-Methode

- ◆ Algorithmisches Verfahren
 - beliebig viele Eingänge
 - 1 Ausgang
- ◆ 3 Schritte
 - Initialisierung
 - Ermittlung der Primimplikanten
 - Ermittlung der minimalen Überdeckung
- ◆ Beispielfunktion
 - Segment a

e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

© 1997 Peter Sturm, Universität Trier

2.45

1. Initialisierung

- ◆ Sortierung der Minterme aufsteigend nach Anzahl 1

0000	(0)
0001	(1)
0010	(2)
0100	(4)
1000	(8)
0011	(3)
1001	(9)
1010	(10)
0111	(7)
1101	(13)

2.46

© 1997 Peter Sturm, Universität Trier

2. Primimplikanten

- ◆ Vergleich jedes Element einer Gruppe mit allen Elementen der nächsten Gruppe
- ◆ Übernahme in die nächste Spalte, wenn nur in einer Position verschieden
- ◆ Gewählte Zeilen markieren (ok)

0000	(0)	ok	000-	(0,1)
			00-0	(0,2)
0001	(1)	ok	0-00	(0,4)
0010	(2)	ok	-000	(0,8)
0100	(4)	ok		
1000	(8)	ok	00-1	(1,3)
			-001	(1,9)
0011	(3)	ok	001-	(2,3)
1001	(9)	ok	-010	(2,10)
1010	(10)	ok	100-	(8,9)
			10-0	(8,10)
0111	(7)	ok		
1101	(13)	ok	0-11	(3,7)
			1-01	(9,13)

2.47

© 1997 Peter Sturm, Universität Trier

Abbruchkriterium

- ◆ Markierung mit der nächsten Spalte wiederholen
- ◆ Bis auf eine Position gleich (- = -)
- ◆ Abbruch, wenn keine weitere Spalte entsteht

000-	(0,1)	ok	00--	(0,1,2,3)	*
00-0	(0,2)	ok	-00-	(0,1,8,9)	*
0-00	(0,4)	*	-0-0	(0,2,8,10)	*
-000	(0,8)	ok			
00-1	(1,3)	ok			
-001	(1,9)	ok			
001-	(2,3)	ok			
-010	(2,10)	ok			
100-	(8,9)	ok			
10-0	(8,10)	ok			
0-11	(3,7)	*			
1-01	(9,13)	*			

2.48

© 1997 Peter Sturm, Universität Trier

3. Minimale Überdeckung

- ◆ Sammeln der mit * markierten Primimplikanten:

0-00	(0,4)
0-11	(3,7)
1-01	(9,13)
00--	(0,1,2,3)
-00-	(0,1,8,9)
-0-0	(0,2,8,10)

	0	1	2	3	4	7	8	9	A	D
0-00	X				X					
0-11				X	X					
1-01								X	X	
00--	X	X	X	X						
-00-	X	X					X	X		
-0-0	X	X					X	X		

- ◆ Nur einmal markierte Spalten suchen
 - essentielle Primimplikanten
 - Zusätzliche Spalten streichen

	0	1	2	3	4	7	8	9	A	D
0-00	X				X					
0-11				X	X					
1-01								X	X	
00--	X	X	X	X						
-00-	X	X					X	X		
-0-0	X	X					X	X		

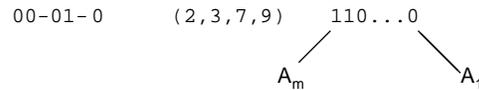
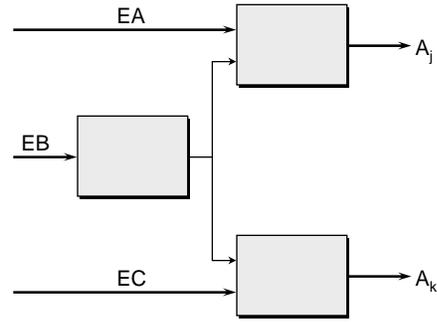
- ◆ Schritt auf unmarkierten Spalten wiederholen

2.49

© 1997 Peter Sturm, Universität Trier

Bündelminimierung

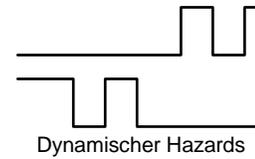
- ◆ Anwendbar auf n Eingänge und m Ausgänge
- ◆ Nutzung gemeinsamer Teilsummen in verschiedenen Ausgangsfunktionen
- ◆ Multilevel-Logik
- ◆ Ansatz
 - Erweiterung der Zeileninformation
 - ansonsten Quine-McCluskey



© 1997 Peter Sturm, Universität Trier
2.50

Hazards

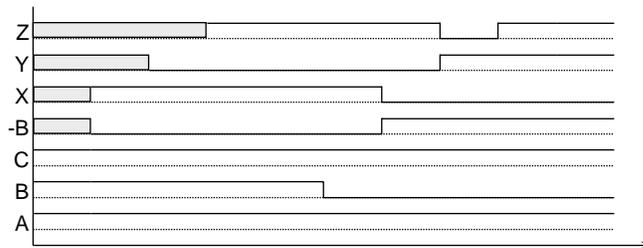
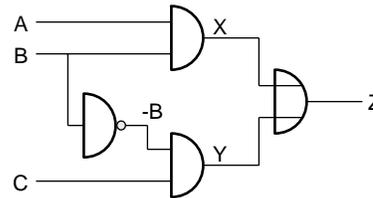
- ◆ Ungewollte Wechsel an einem Ausgang
 - Unterschiedliche Verzögerungszeiten der Gatter
 - Unterschiedliche Gatteranzahl zwischen Eingängen und Ausgang
- ◆ Statischer Hazard
 - Einmaliger Wechsel
- ◆ Dynamischer Hazard
 - Mehrfacher Wechsel



© 1997 Peter Sturm, Universität Trier
2.51

Statischer Hazard

- ◆ Nur 1-Bit-Wechsel
- ◆ Beispiel
 - $Z = AB + \bar{B}C$
 - Wechsel 111 nach 011 (ABC)



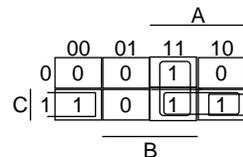
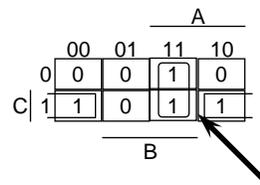
2.52

© 1997 Peter Sturm, Universität Trier

Eliminierung statischer Hazards

- ◆ Grundlage Karnaugh-Diagramm
- ◆ Merkmal für 1-Hazard
 - Wechsel des Primimplikanten bei 1-Bitwechsel der Eingabe
 - Lösung: Redundante Implikanten
- ◆ Merkmal für 0-Hazard
 - Karnaugh-Diagramm für konjunktive Normalform
 - Analog 1-Hazard
- ◆ Eliminierung statischer Hazards auch in mehrstufigen Schaltungen möglich

$$Z = AB + \bar{B}C$$

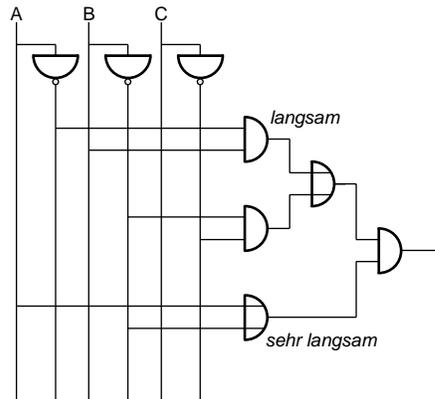


2.53

© 1997 Peter Sturm, Universität Trier

Dynamischer Hazard

- ◆ Beispiel
 - $(\bar{A}B + \bar{B}C)(A + \bar{B})$
 - Wechsel 000 nach 010 (ABC)
- ◆ Grund
 - Unterschiedlich lange Wege von einem Eingang zu einem Ausgang
- ◆ Eliminierung schwierig

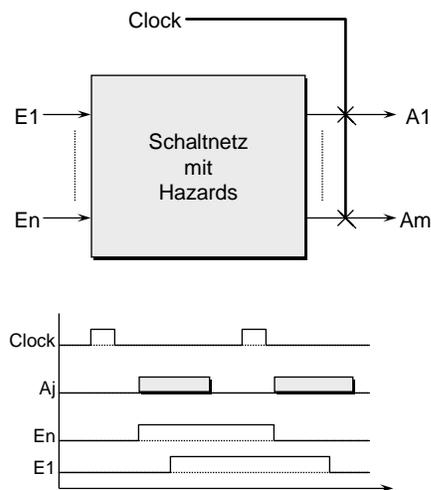


© 1997 Peter Sturm, Universität Trier

2.54

Bemerkungen

- ◆ Beschränkte Möglichkeiten
 - Nur 1-Bit-Wechsel
 - Multilevel-Logik schwierig
- ◆ Schaltungen frei von statischen und dynamischen Hazards
 - 2-stufige Logik
 - Fan-In/Fan-Out-Problem
- ◆ Gängiger Ansatz
 - Auferzwungene Taktung
 - Periode länger als maximaler Hazard

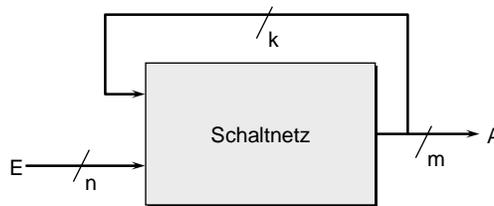


© 1997 Peter Sturm, Universität Trier

2.55

Schaltwerke

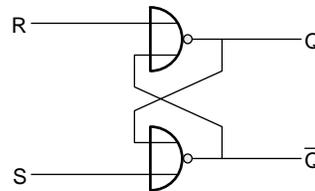
- ◆ Kombinatorische Schaltnetze
 - Ausgang hängt nur von den Eingängen ab
 - Unterschiedliche Laufzeiten (Hazard-Problematik)
- ◆ Schaltwerk
 - Ausgang hängt von den Eingängen und den vorherigen Ausgaben ab
- ◆ Aspekte
 - Speichern möglich
 - Synchron / Asynchron
 - Schwingungen
 - Metastabilität
 - Selbsttaktung
- ◆ Elementarbausteine
 - Latch
 - Flip-Flop



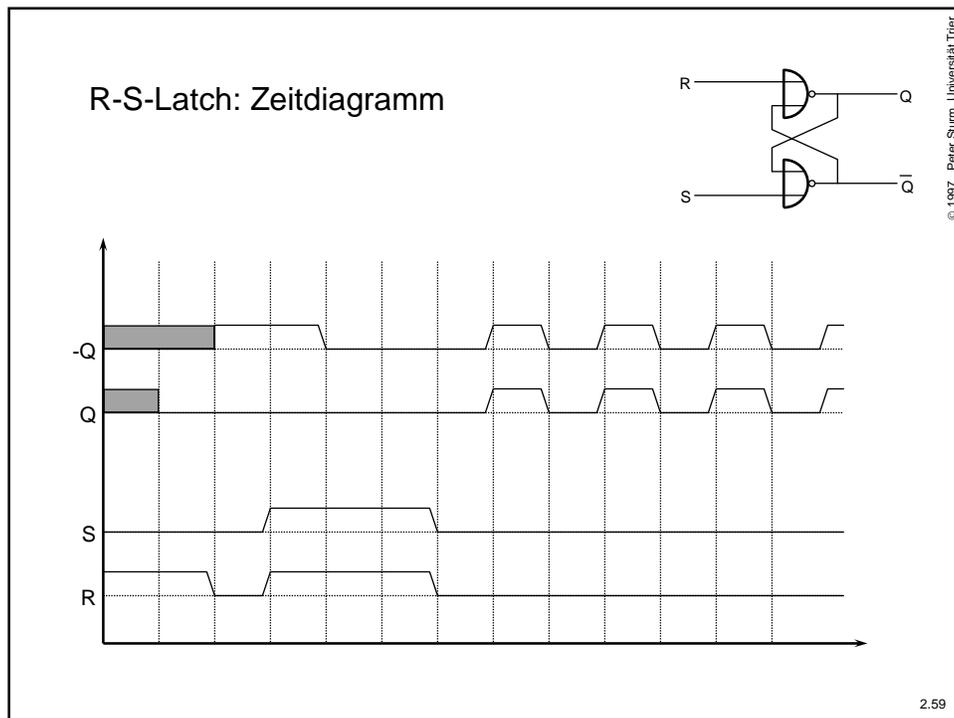
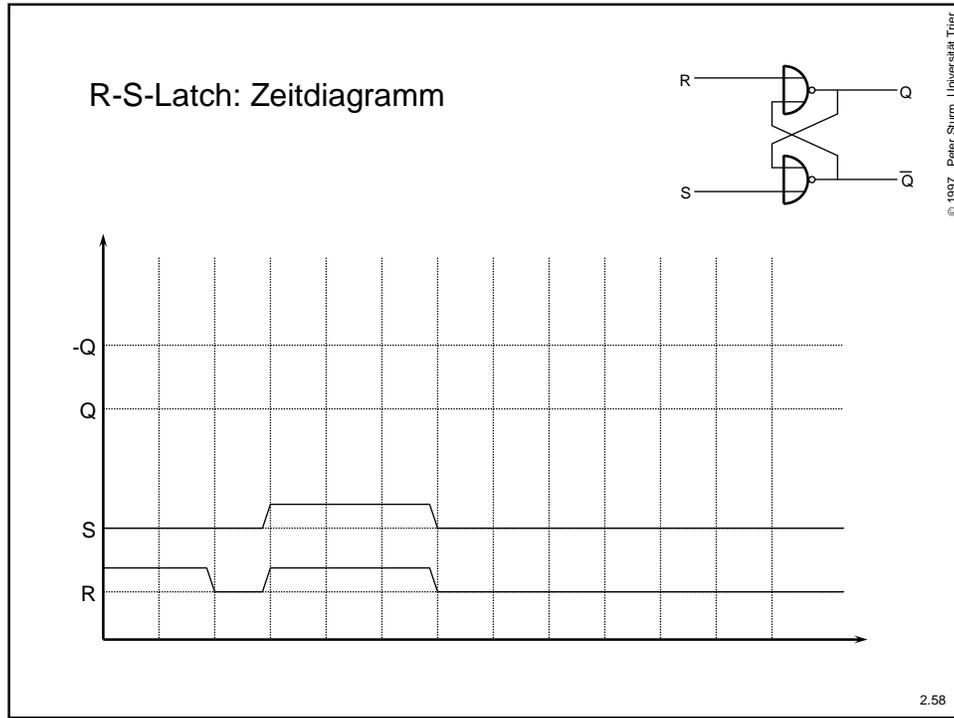
© 1997 Peter Sturm, Universität Trier
2.56

Elementarspeicher: R-S-Latch

- ◆ Kreuzverschaltete NOR-Gatter
 - R: Reset
 - S: Set
 - Ausgang Q mit Komplement
- ◆ Ausgang zurücksetzen
 - R=1, S=0
- ◆ Ausgang setzen
 - R=0, S=1
- ◆ Wert speichern
 - R=0, S=0



© 1997 Peter Sturm, Universität Trier
2.57



R-S-Latch: Charakteristische Gleichung

S(t)	R(t)	Q(t)	Q(t+Δ)	
0	0	0	0	Hold
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	X	Invalid
1	1	1	X	

		S			
		00	01	11	10
Q(t)	0	0	0	X	1
	1	1	0	X	1
		R			

$$Q(t + \Delta) = S(t) + \bar{R}(t) \cdot Q(t)$$

2.60

© 1997 Peter Sturm, Universität Trier

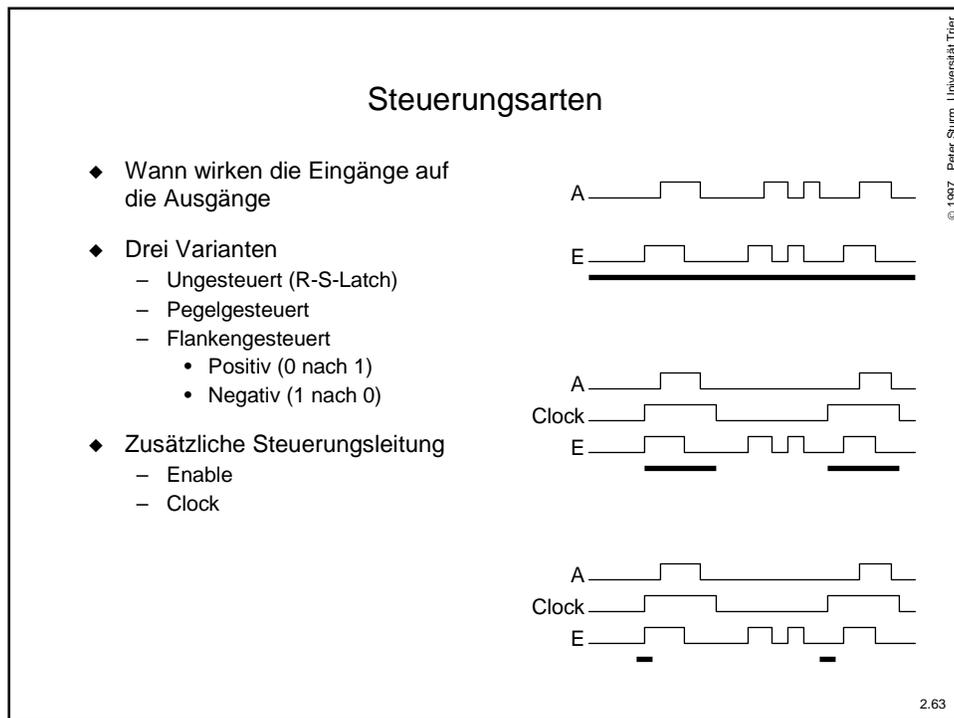
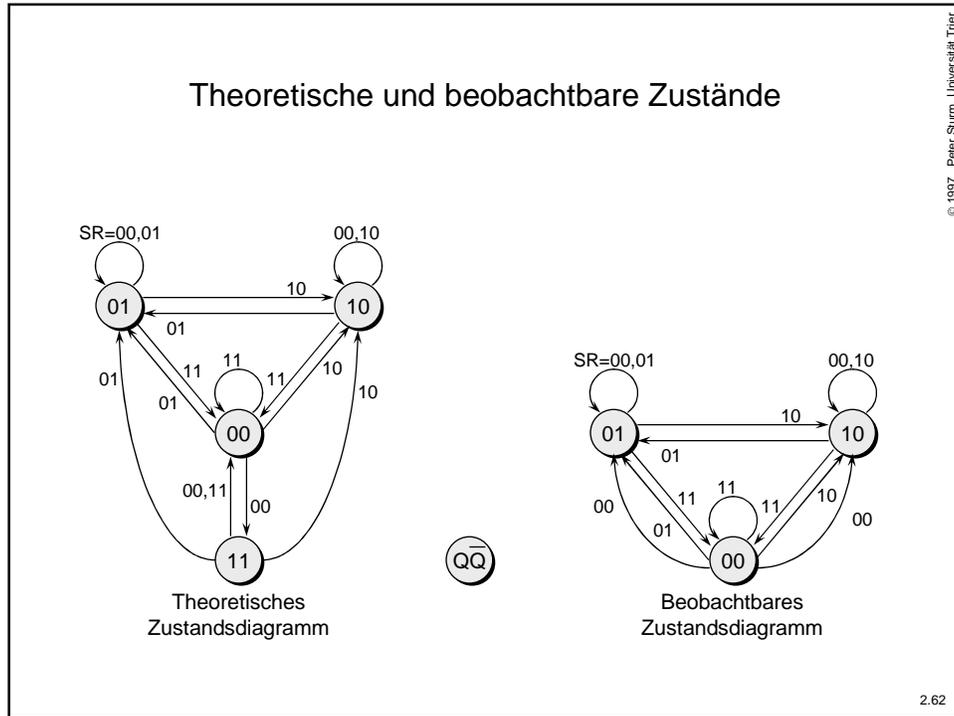
Probleme

- ◆ Verbotene Eingangskombination
 - R=S=1
 - Beide Ausgänge sind 0
- ◆ Verbotener Übergang
 - R und S gleichzeitig von 1 nach 0
 - Oszillation der Ausgänge
- ◆ Wie lang kann die Oszillation dauern?
 - Theoretisch?
 - Praktisch?
- ◆ Race-Condition

S	R	Q
0	0	Speichern
0	1	0
1	0	1
1	1	Instabil

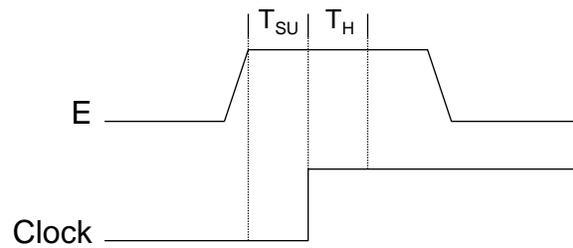
2.61

© 1997 Peter Sturm, Universität Trier



Flankensteuerung

- ◆ Stabiler Eingang in einem Zeitfenster vor dem Flankenwechsel
 - Setup-Zeit: T_{SU}
 - Hold-Zeit: T_H
- ◆ Verhalten ansonst undefiniert
- ◆ Typische Werte (TTL)
 - $T_{SU} = 20\text{ns}$
 - $T_H = 5\text{ns}$

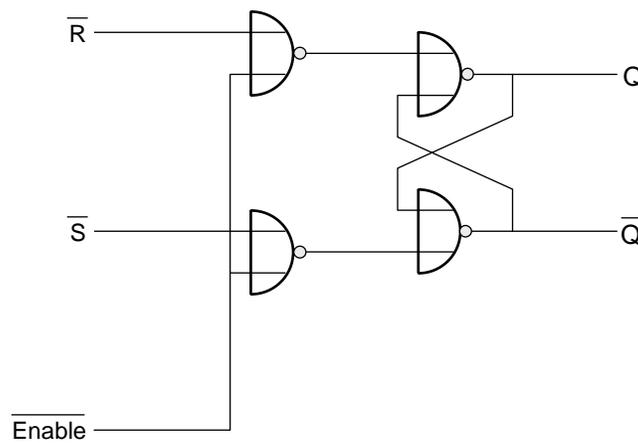


2.64

© 1997 Peter Sturm, Universität Trier

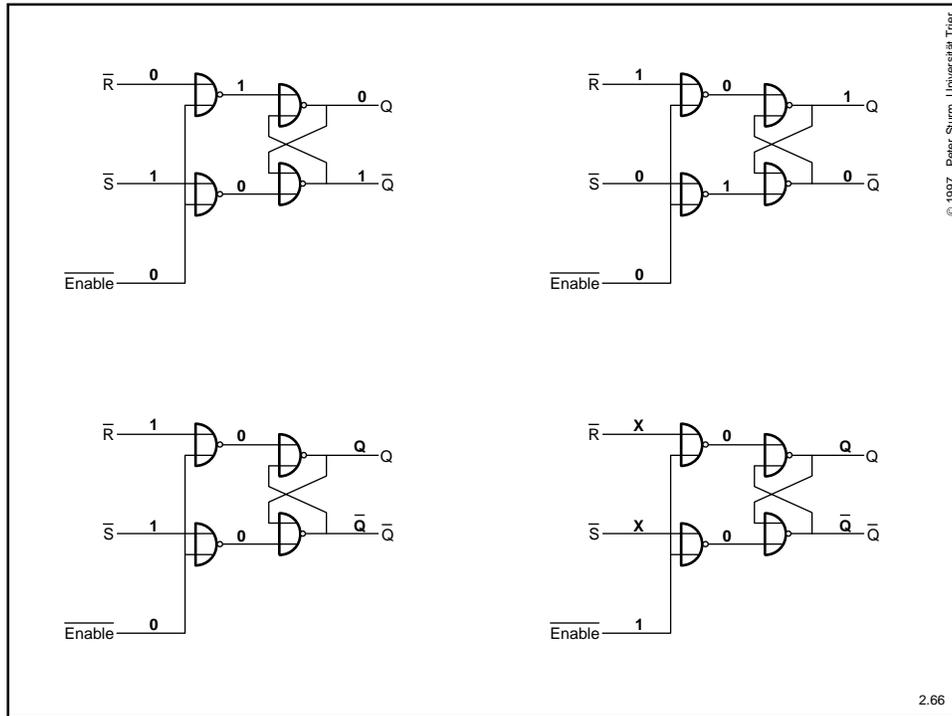
Pegelgesteuertes R-S-Latch

- ◆ Steuereingang Enable



2.65

© 1997 Peter Sturm, Universität Trier



Latch vs. Flip-Flop

<ul style="list-style-type: none"> ◆ Latch <ul style="list-style-type: none"> - Ungesteuert - Pegelgesteuert ◆ Änderung der Ausgänge bei Änderung der Eingänge 	<ul style="list-style-type: none"> ◆ Flip-Flop <ul style="list-style-type: none"> - Positiv flankengesteuert - Negativ flankengesteuert - Master/Slave ◆ Änderung der Ausgänge wird durch den Steuerungseingang getriggert
---	--

Clock

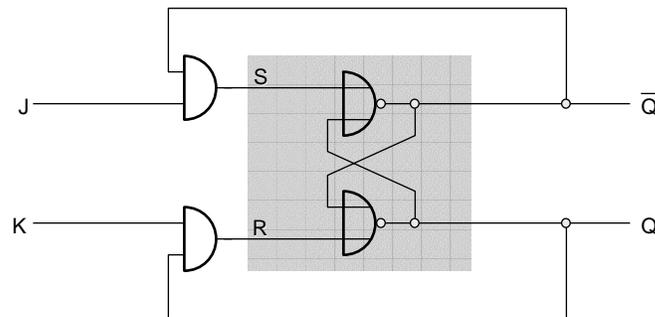
Clock

© 1997 Peter Sturm, Universität Trier

2.67

JK-Latch

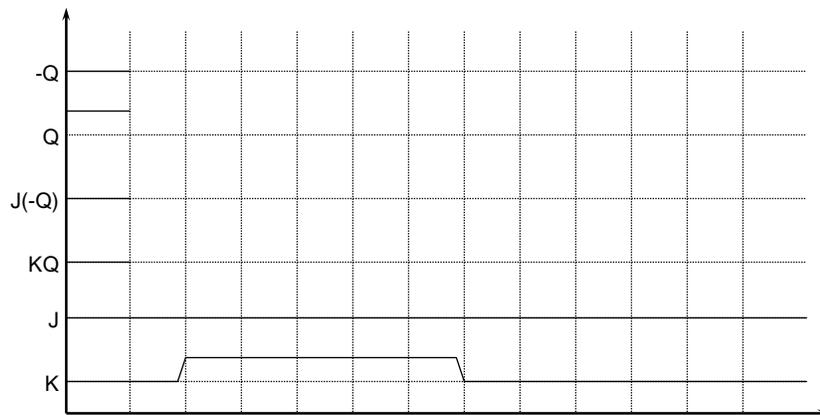
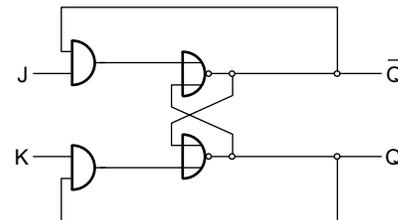
- ◆ Erweiterung eines R-S-Latch
 - Ungültige Eingabe $R=S=1$ verhindern
- ◆ Was passiert bei $J=K=1$?



2.68

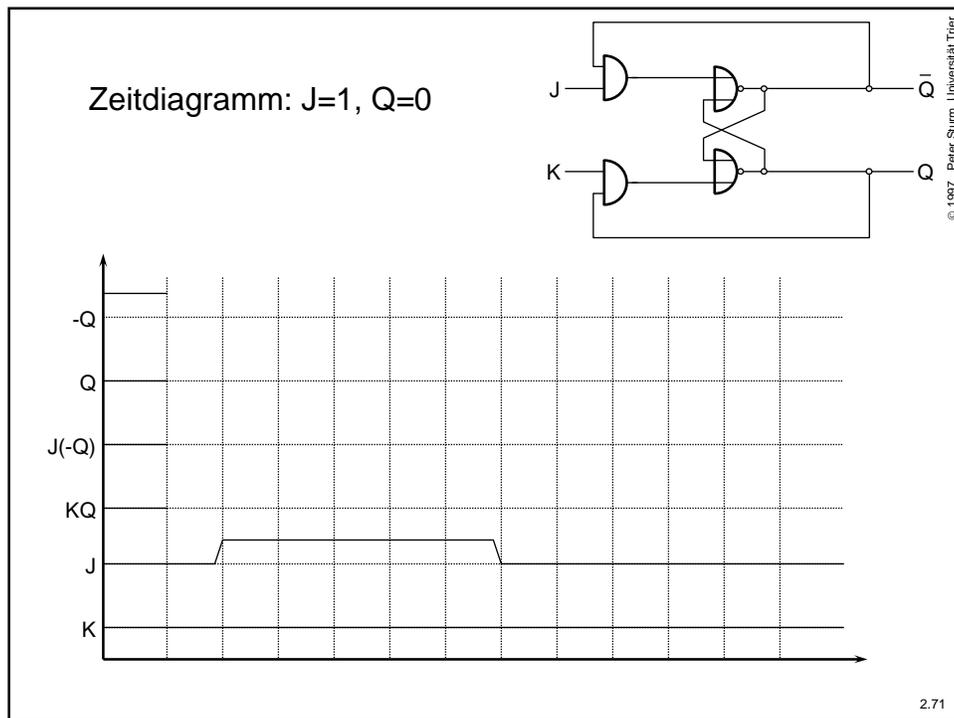
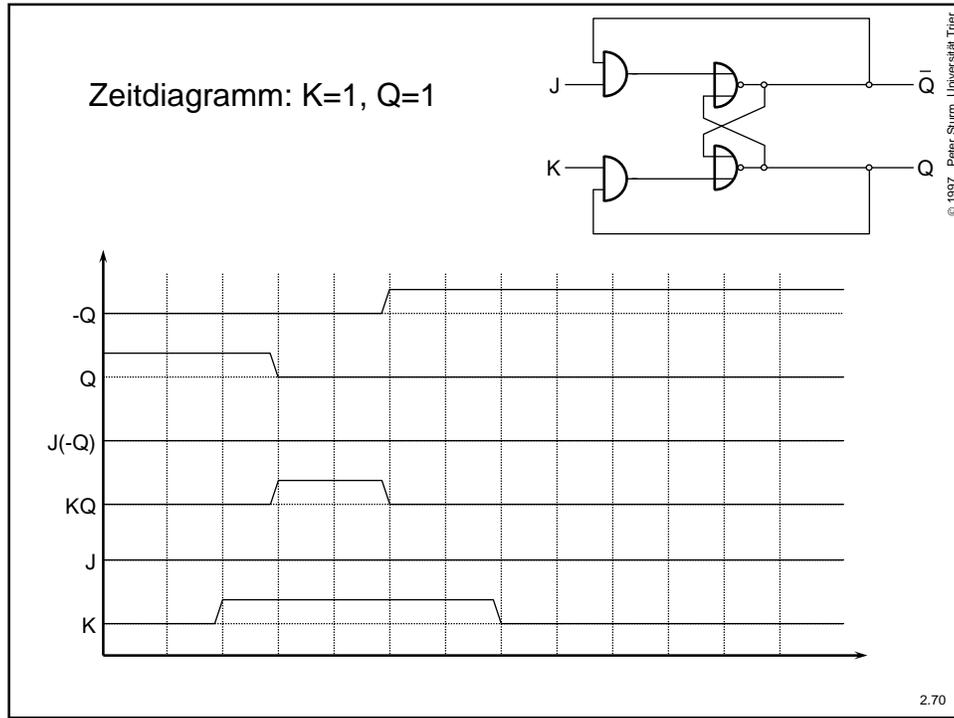
© 1997 Peter Sturm, Universität Trier

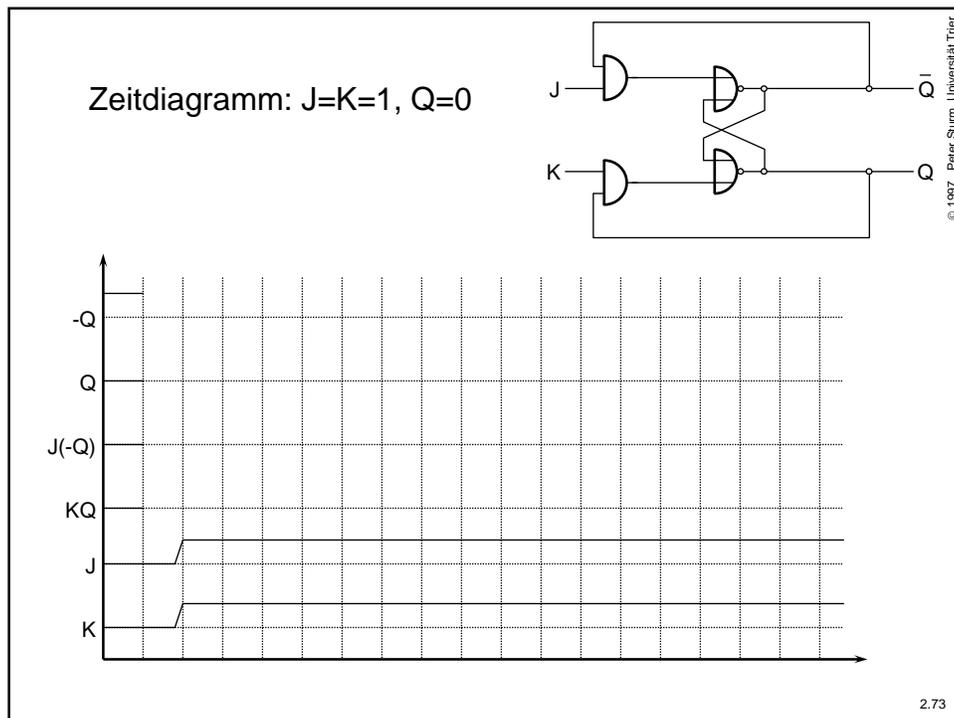
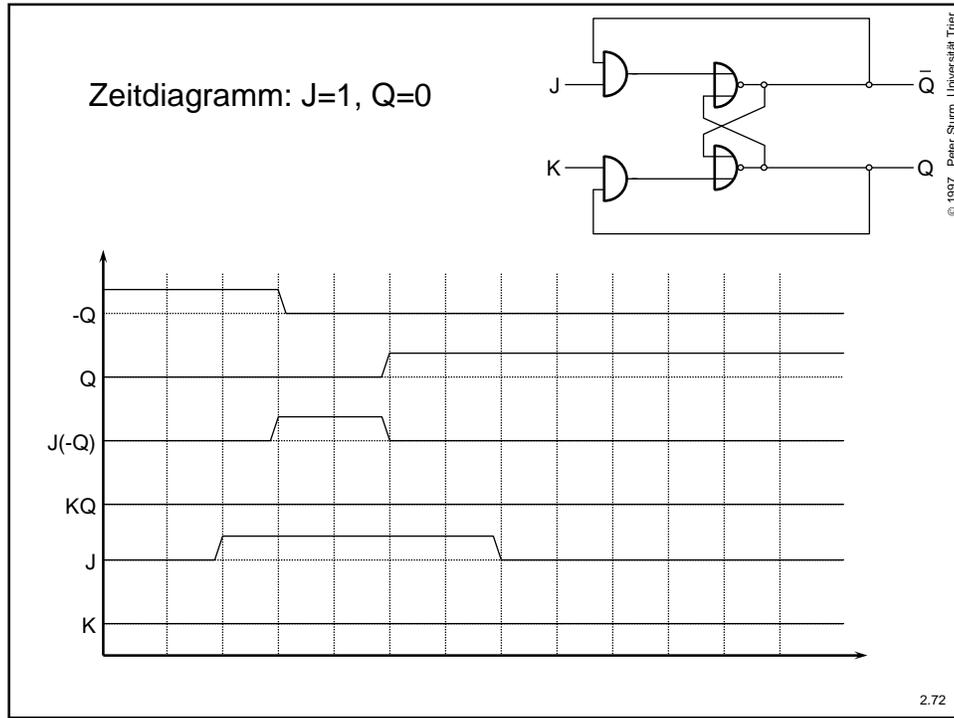
Zeitdiagramm: $K=1, Q=1$

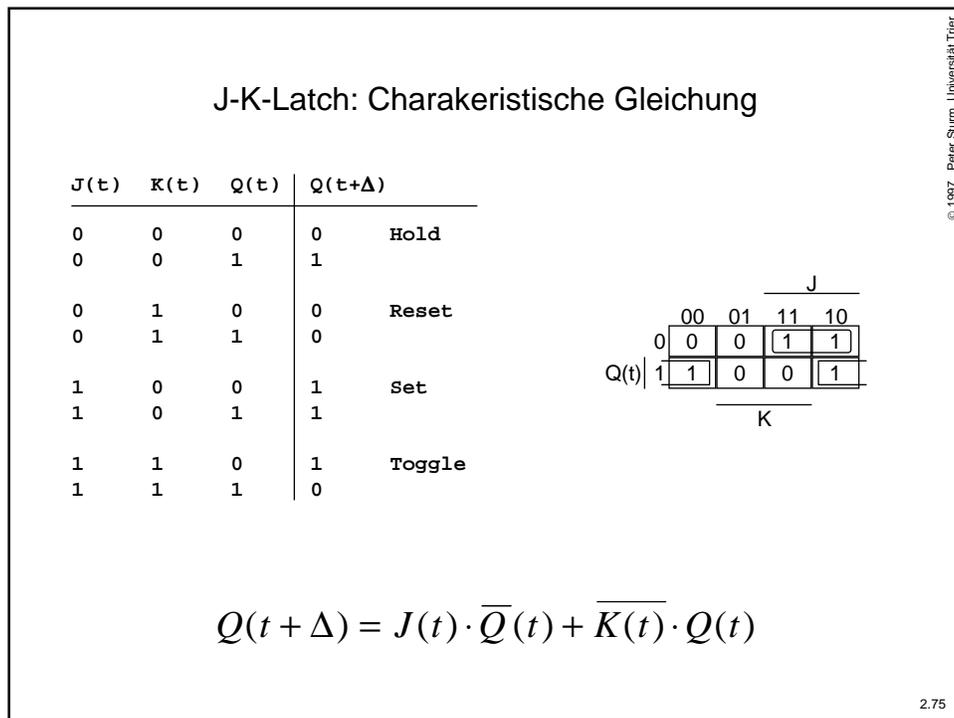
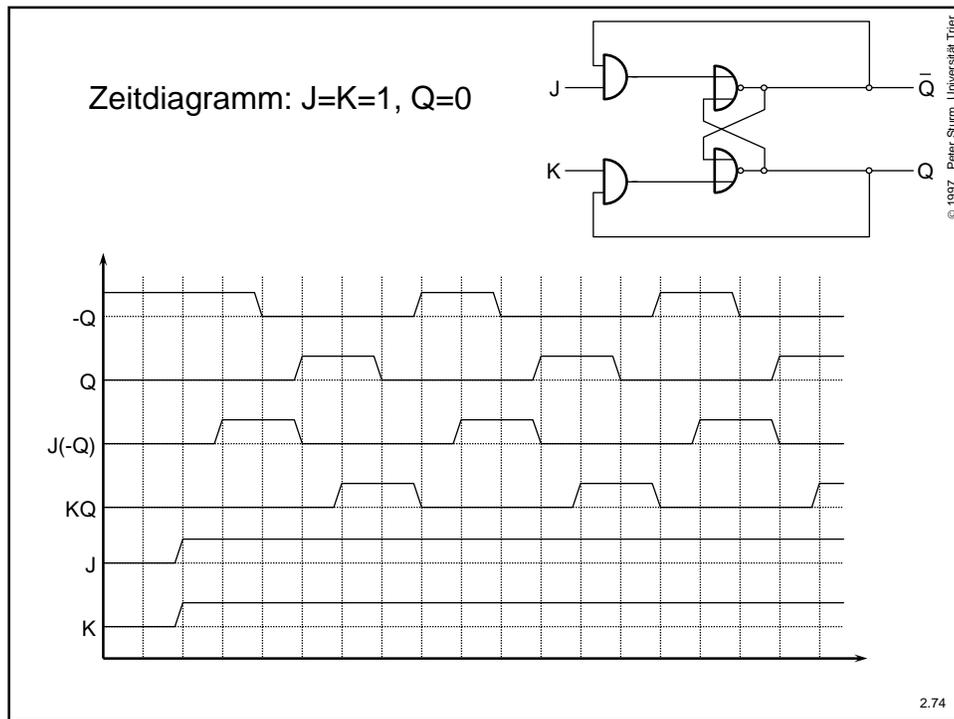


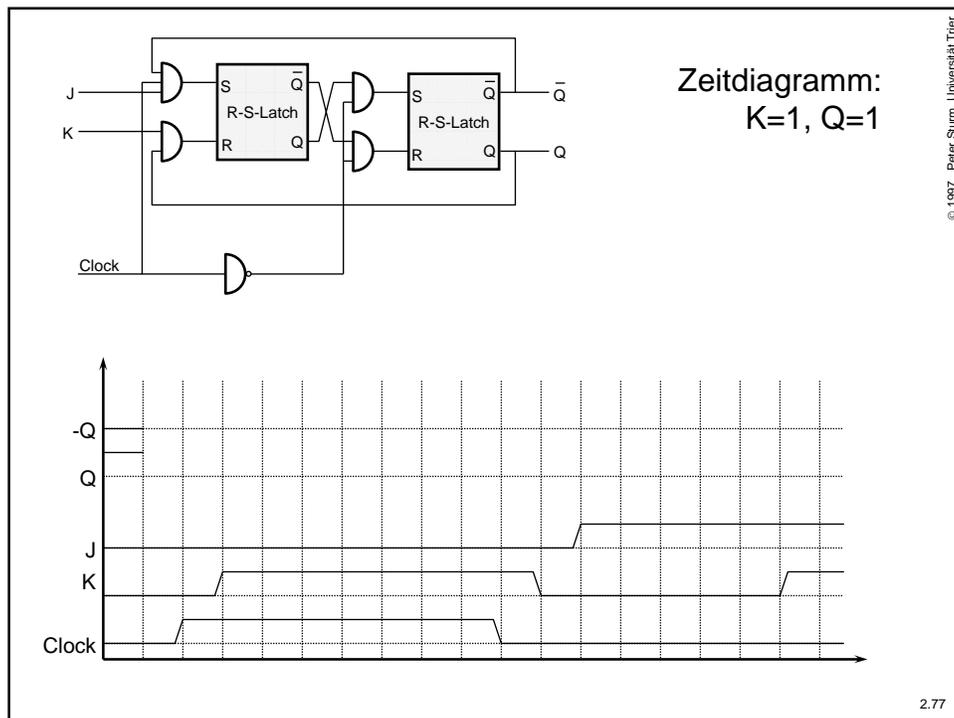
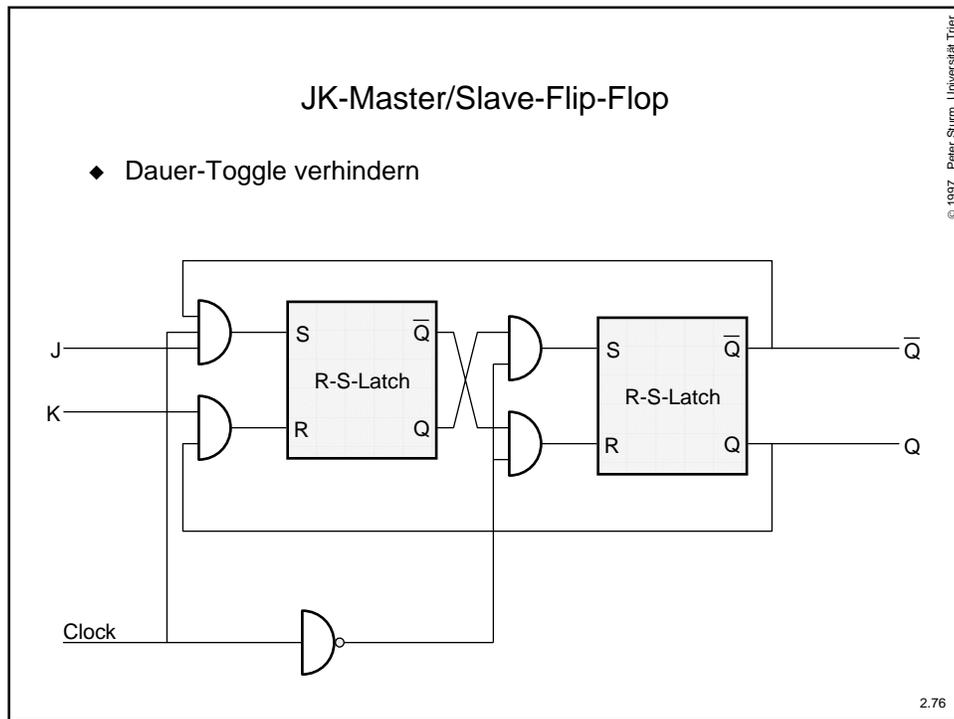
2.69

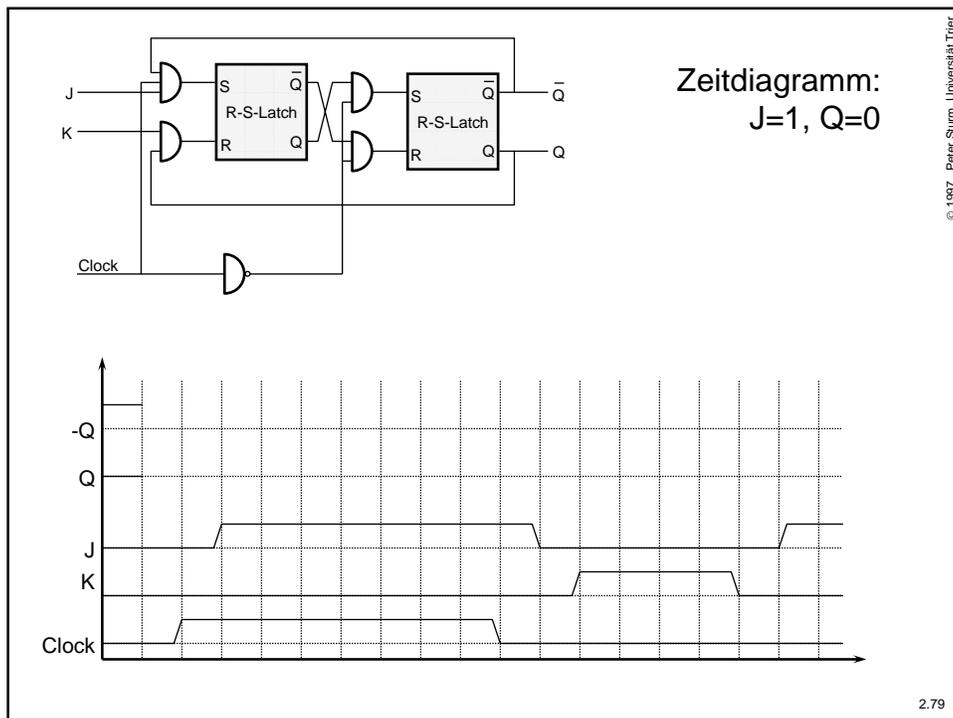
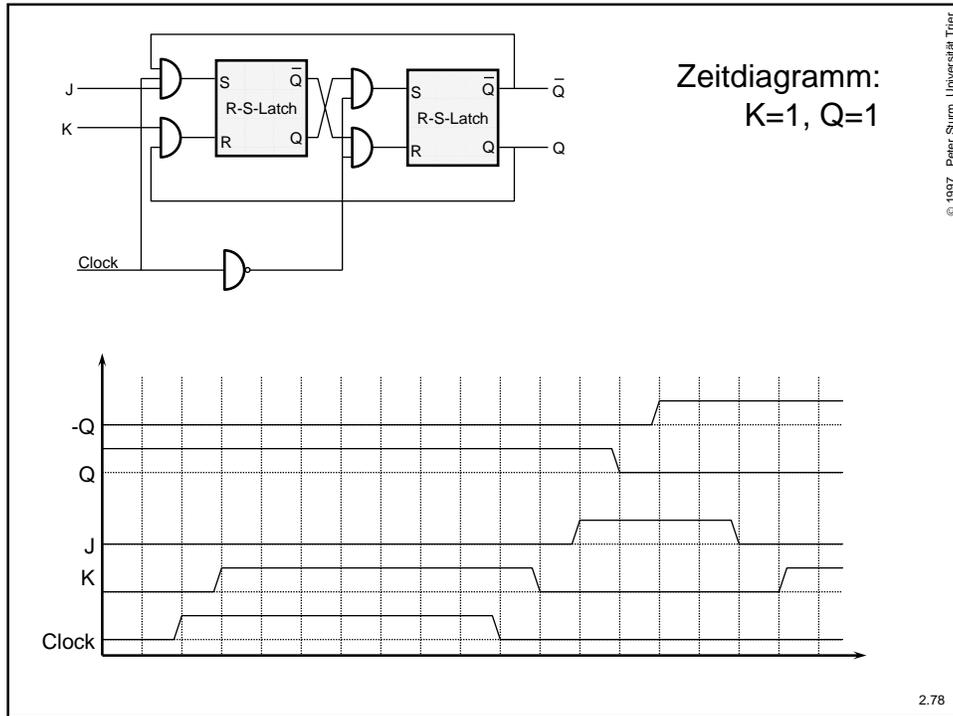
© 1997 Peter Sturm, Universität Trier

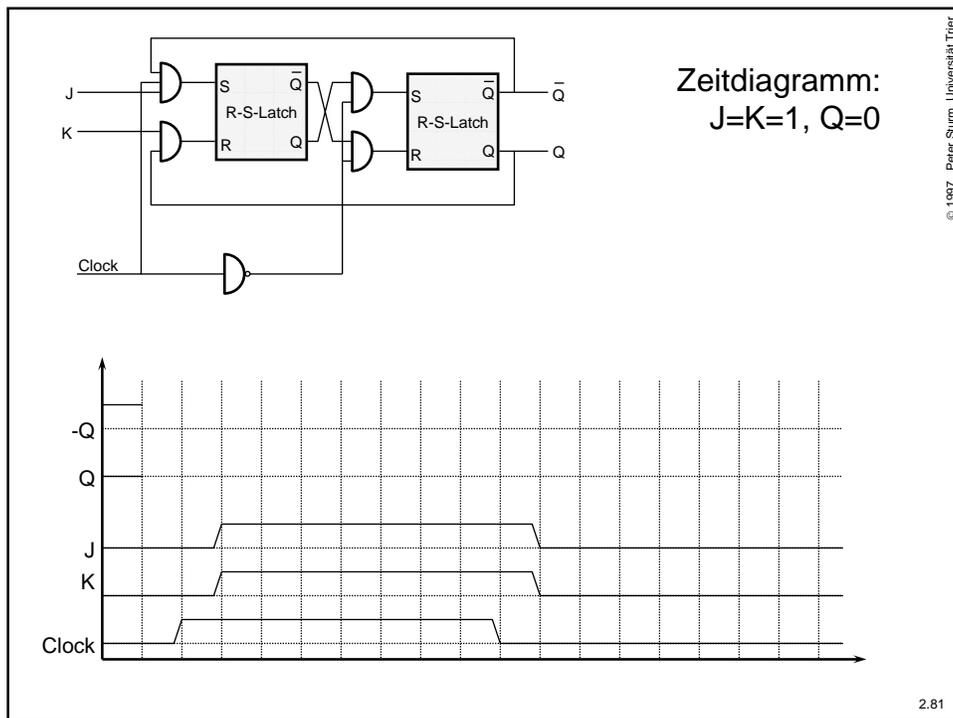
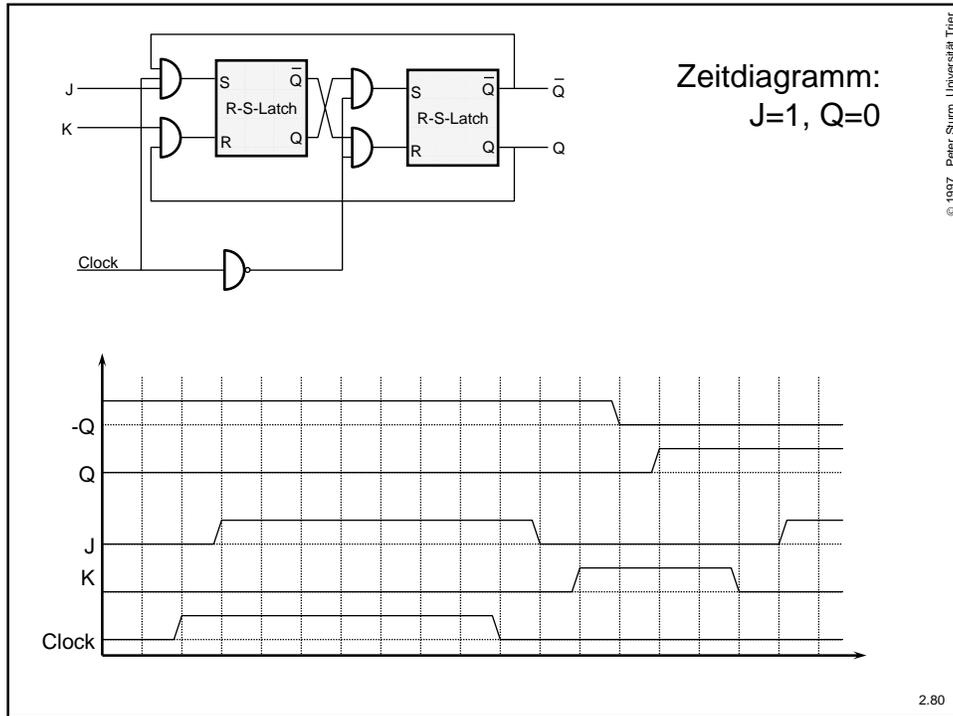


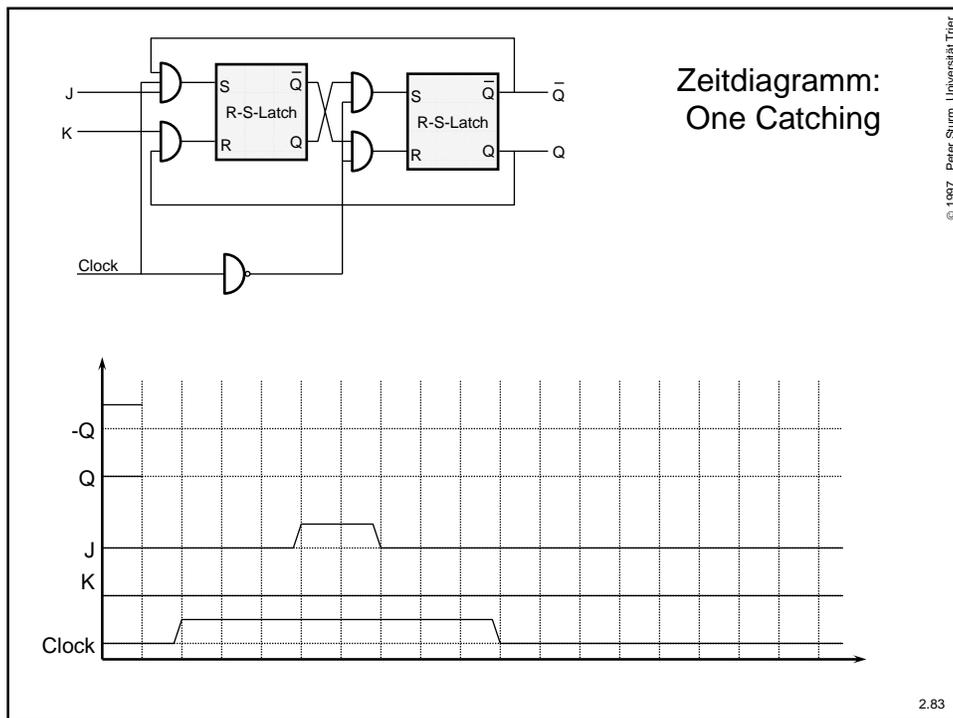
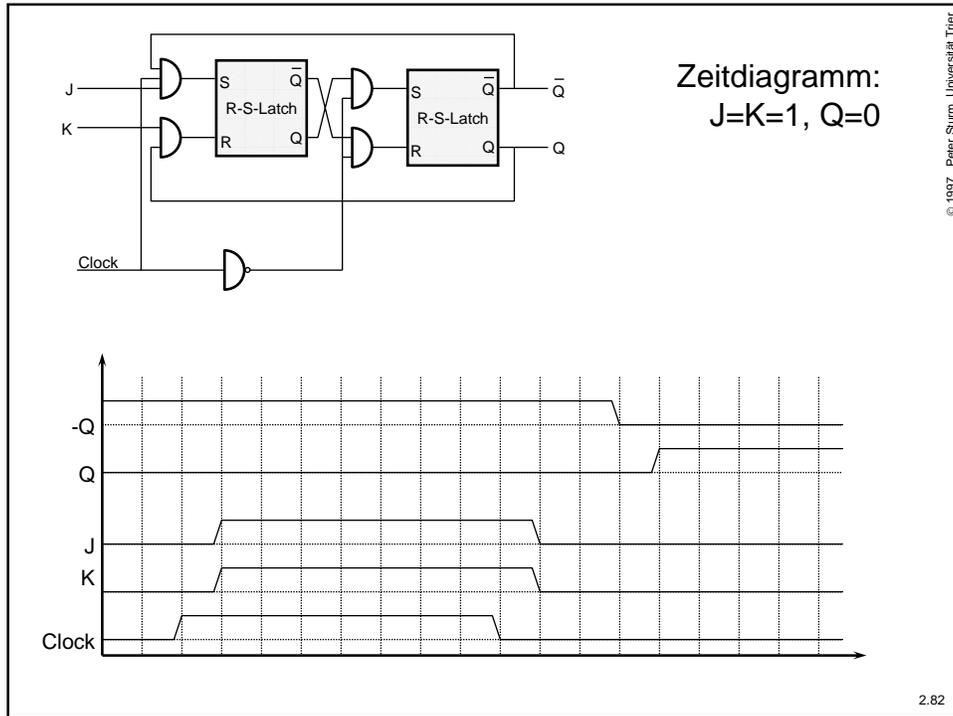


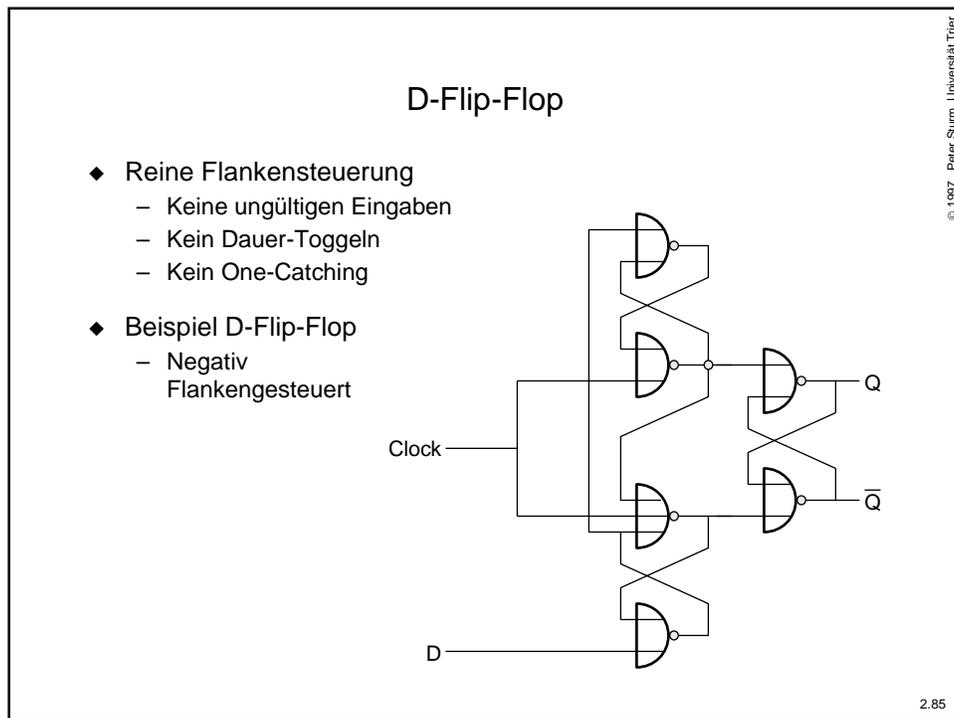
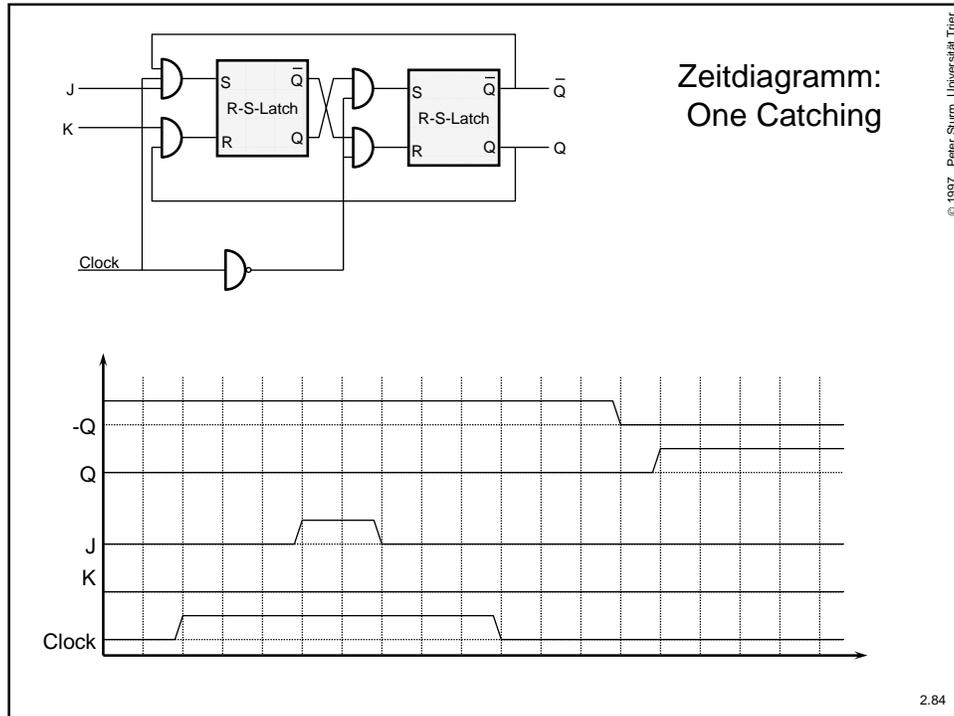


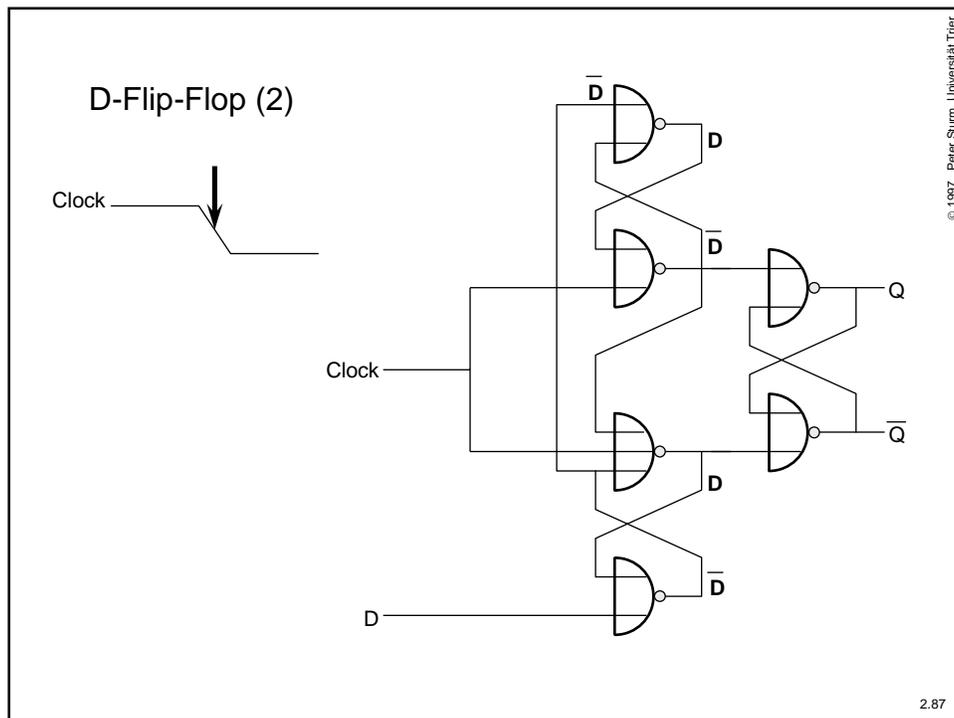
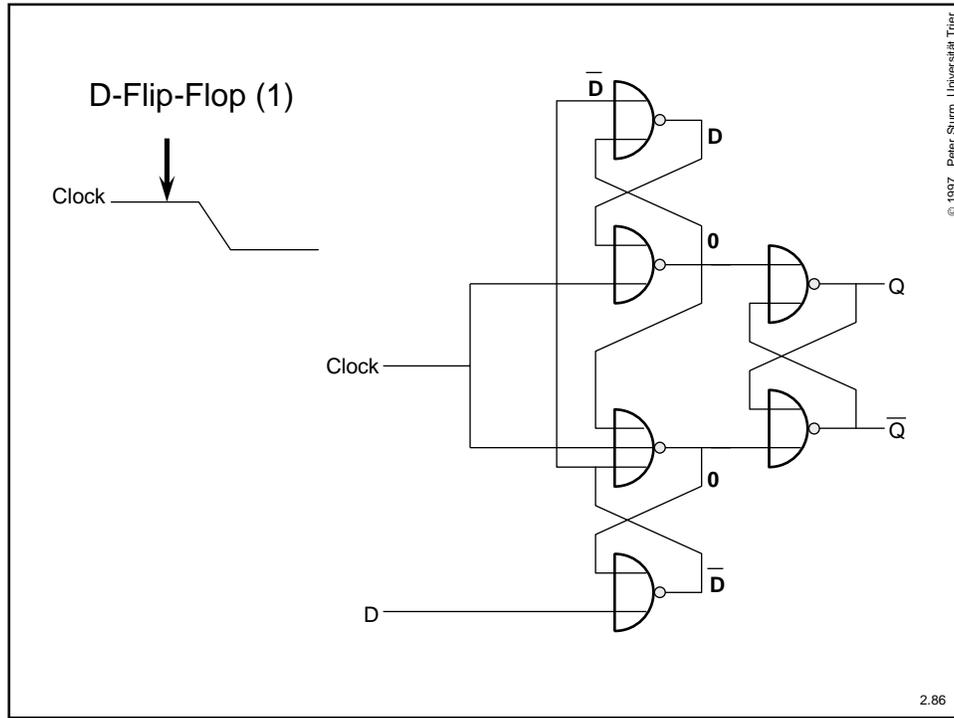


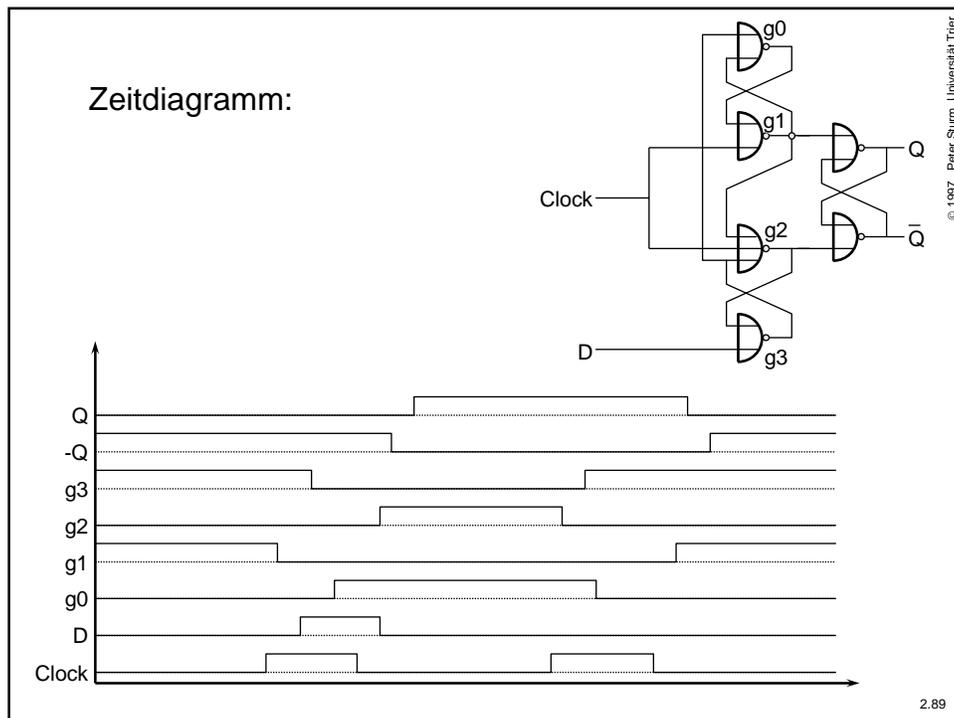
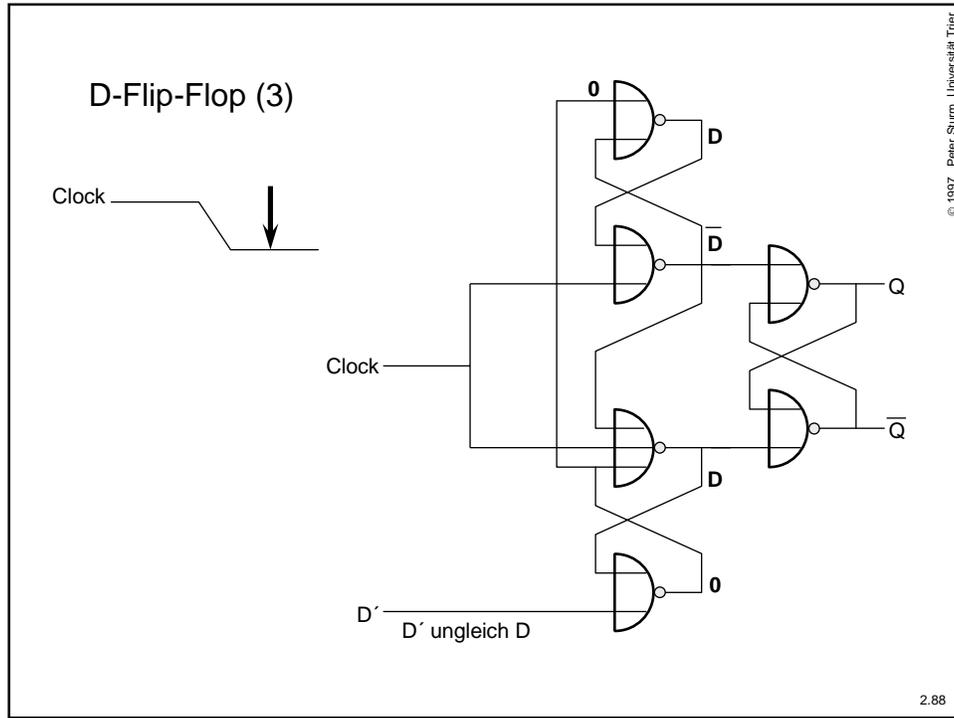












Programmierbare Logik

- ◆ Realisierung von Schaltnetzen und Schaltwerken
- ◆ Aufbau mit Hilfe von TTL- und CMOS-ICs aufwendig
 - Große Anzahl an Bausteinen
 - Hoher Platz- und Stromverbrauch
 - Geringe Integrationsdichte
- ◆ IC mit 1000 AND mit jeweils 2 Eingängen = 3002 Pins
- ◆ “Programmierbare Bausteine”
 - Genügend Eingängen
 - Genügend Ausgängen
 - Ausreichende Programmierbarkeit
- ◆ Zweistufige Normalformen

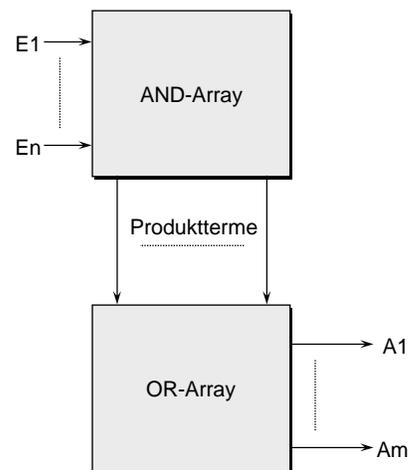


2.90

© 1997 Peter Sturm, Universität Trier

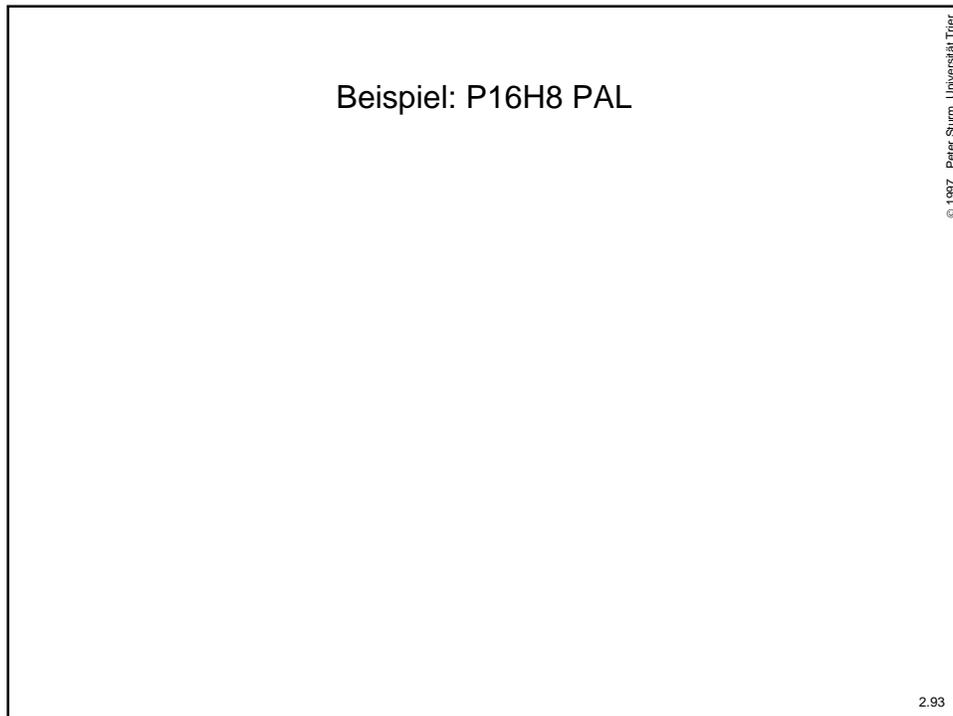
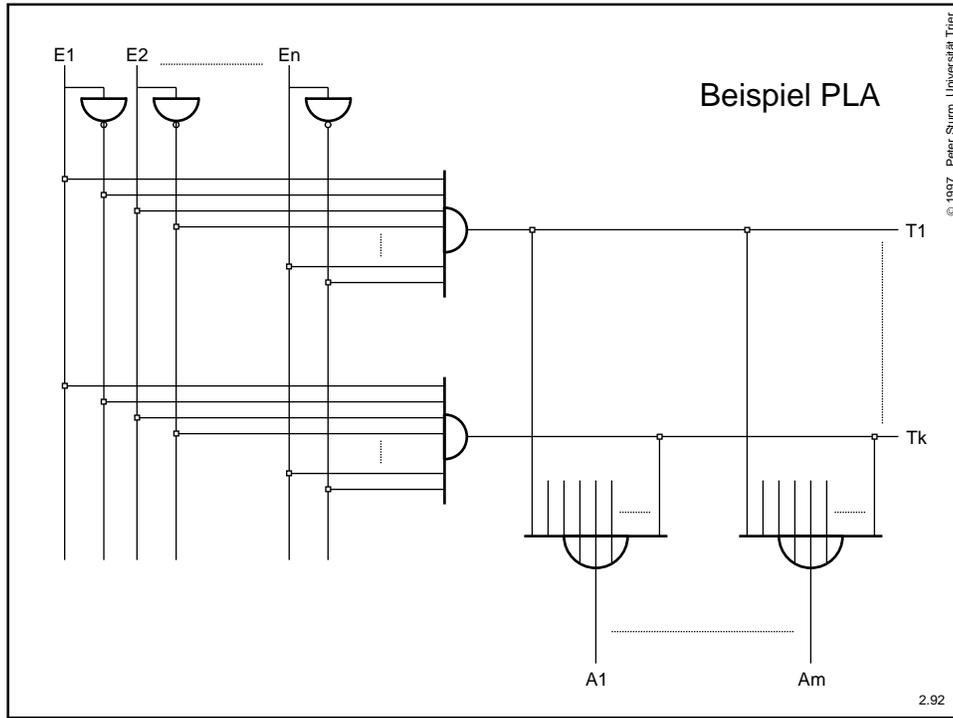
PLA und PAL

- ◆ Programmierbare disjunktive Normalform
 - n Eingängen
 - m Ausgängen
 - k Terme
- ◆ n, m und k vom jeweiligen Baustein abhängig
- ◆ PLA = Programmable Logic Array
 - UND- und ODER-Array programmierbar
- ◆ PAL = Programmable Array Logic
 - Nur UND-Array programmierbar
- ◆ Programmieren
 - Höhere Programmierspannung
 - “Durchbrennen” einer Verbindung



2.91

© 1997 Peter Sturm, Universität Trier



Beispiel: P14H8 PAL

© 1997 Peter Sturm, Universität Trier

2.94

Beispiel: F100 PAL

© 1997 Peter Sturm, Universität Trier

2.95