

Technische Grundlagen der Informatik

Vorrechenübung 19.11.09



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Was gibt es heute?

- Prof. Koch ist weiterhin erkrankt, voraussichtlich ab Montag wieder da
- Auf JEDEN FALL findet am Montag eine Vorlesung über Verilog statt
- Heute: Vorrechenübung, klausurähnliche Aufgaben
- Nicht nur stumpfes Vorrechnen, aktive Teilnahme, stellen Sie Fragen!
- Lösungen teilweise nur auf Tafel/Overhead

Aufgabe 1: Funktionen

- Realisieren Sie eine Schaltung, die aus einer 4-Bit Binärzahl den entsprechenden Wert des Exzess-3-Codes generiert.
 - Geben Sie minimierte Gleichungen in disjunktiver Form an. Verwenden Sie zur Minimierung K-Diagramme.
 - Verwenden Sie zur Implementierung nur NOR-Gatter. Es sind alle Variablen und Ihre Komplemente verfügbar.
-
- Binär → Exzess-3-Code
 - 0000 → 0011
 - 0001 → 0100
 - 0010 → 0101
 - 0011 → 0110
 - 0100 → 0111
 - 0101 → 1000
 - 0110 → 1001
 - 0111 → 1010
 - 1000 → 1011
 - 1001 → 1100

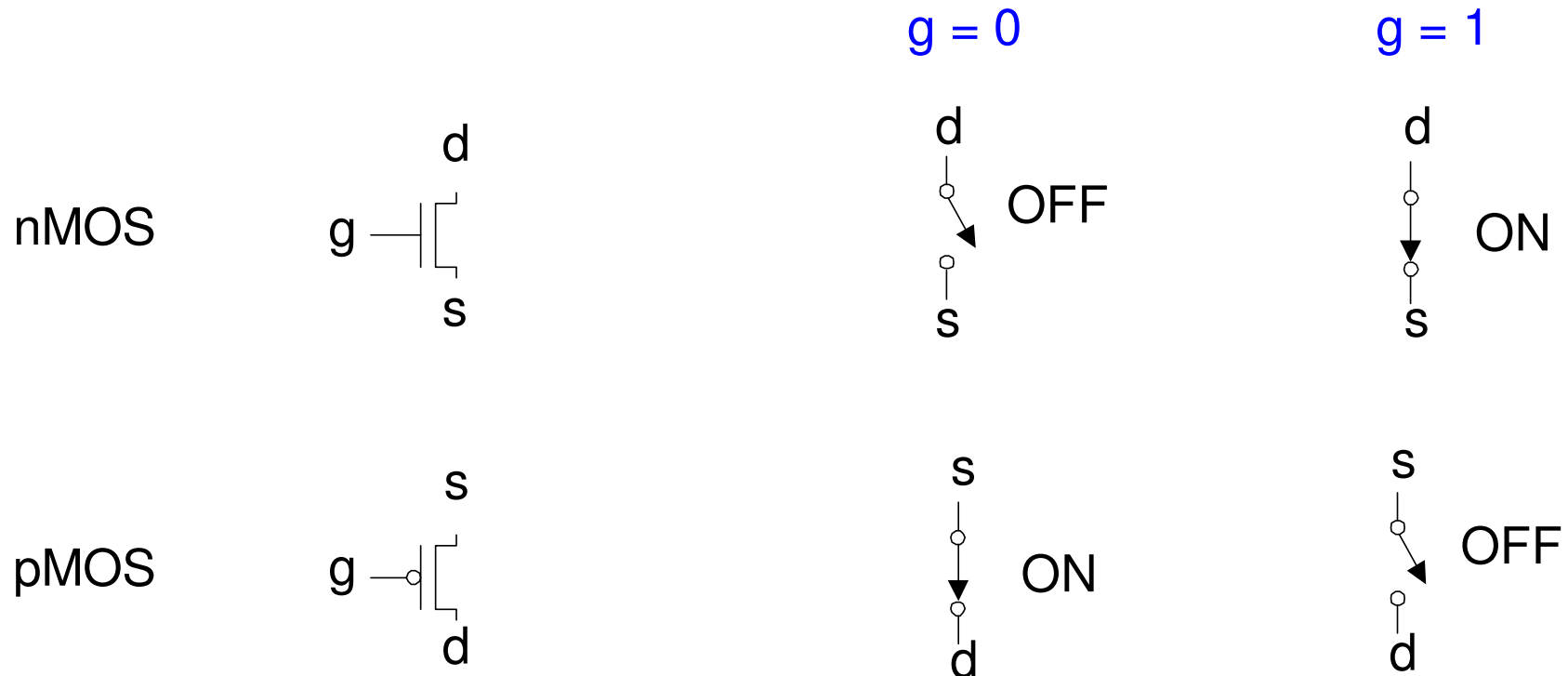
Aufgabe 2: Glitches

- Gegeben ist die folgende Gleichung:

$$F = \overline{D}BA + DC + DB + C\overline{B}A$$

- Zeichnen Sie einen Schaltplan für die Schaltung, die diese Funktion implementiert. Als Eingänge stehen nur die Variablen zur Verfügung, nicht ihre Komplemente. Es stehen Ihnen beliebige Gatter zur Verfügung.
- Ermitteln Sie mit Hilfe eines K-Digramms, ob die Schaltung potentiell Glitches enthält.
- Simulieren Sie den Übergang $ABC\overline{D} \rightarrow \overline{A}BC\overline{D}$, jedes Gatter hat eine Ausbreitungsverzögerung von 5ns.
- Geben Sie eine glitchfreie Funktion an.

Aufgabe 3: nMOS und pMOS



nMOS Transistoren leiten 0'en **gut** zwischen S und D weiter
 pMOS Transistoren leiten 1'en **gut** zwischen S und D weiter

Aufgabe 3: nMOS und pMOS

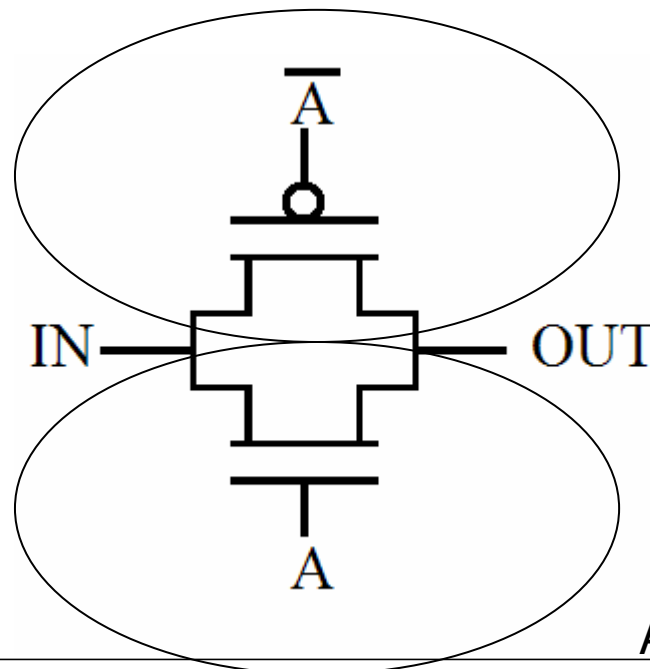
- Realisieren Sie die folgenden Funktionen mit nMOS und pMOS Transistoren
- Es stehen Ihnen als Eingänge die Variablen, ihre Komplemente sowie logisch 0 und 1 zur Verfügung.

$$F = (\overline{A}BC + D)E$$

$$F = \overline{(AC + BD)(E + F)}$$

Transmission Gate

- Besteht aus einem nMOS und einem pMOS Transistor
- Leitet deshalb sowohl die 0 als auch die 1
- Wirkt wie ein Schalter



pMOS, leitet 1 durch
wenn $A=1$

nMOS, leitet 0 durch
wenn $A=1$

$A=0$: beide Transistoren gesperrt

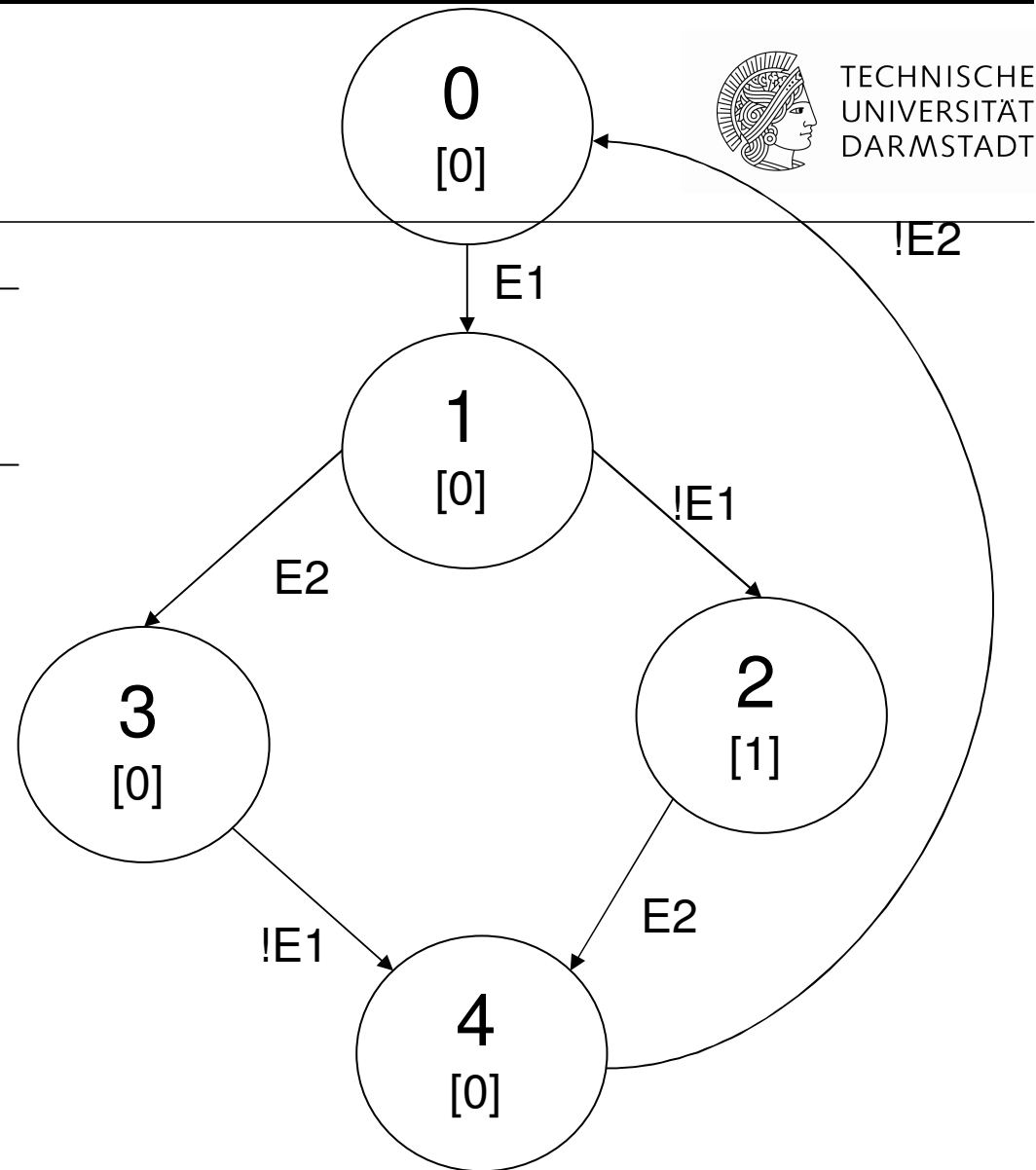
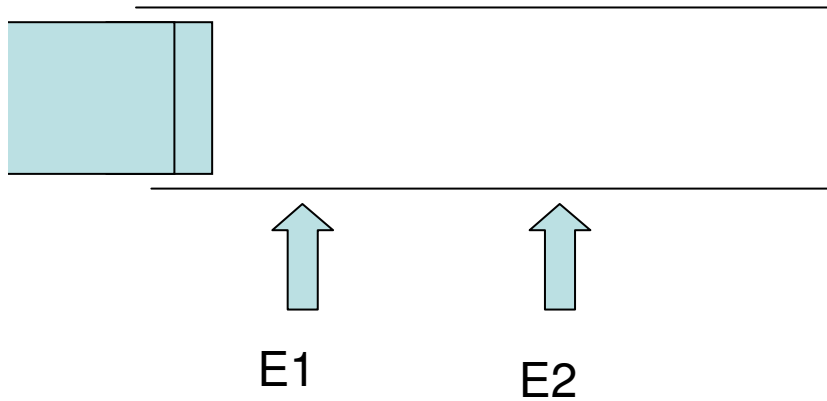
Aufgabe 4: Automat

- Es soll die Steuerung eines Paketsortierers entwickelt werden.
- Auf einem Förderband liegen die Pakete. Im Abstand von 20cm sind 2 Lichtschranken angebracht, die eine logische 1 liefern wenn sie unterbrochen sind.
- Sind die Pakete $< 20\text{cm}$, so soll der Ausgang A, der eine Weiche steuert, logisch 1 sein. Ist das Paket $> 20\text{cm}$, soll $\text{OUT} = 0$ sein. Es kann davon ausgegangen werden, dass die Pakete immer einen Abstand von 30cm haben und dass die Maschine immer fehlerfrei arbeitet.

Aufgabe 4: Automat

- Geben Sie den Zustandsübergangsgraph an, verwenden Sie eine binäre Codierung.
- Geben Sie die Zustandsübergangstabelle an
- Geben Sie die Ausgangstabelle an
- Geben Sie Zustandsübergangs- und Ausgangsgleichungen an

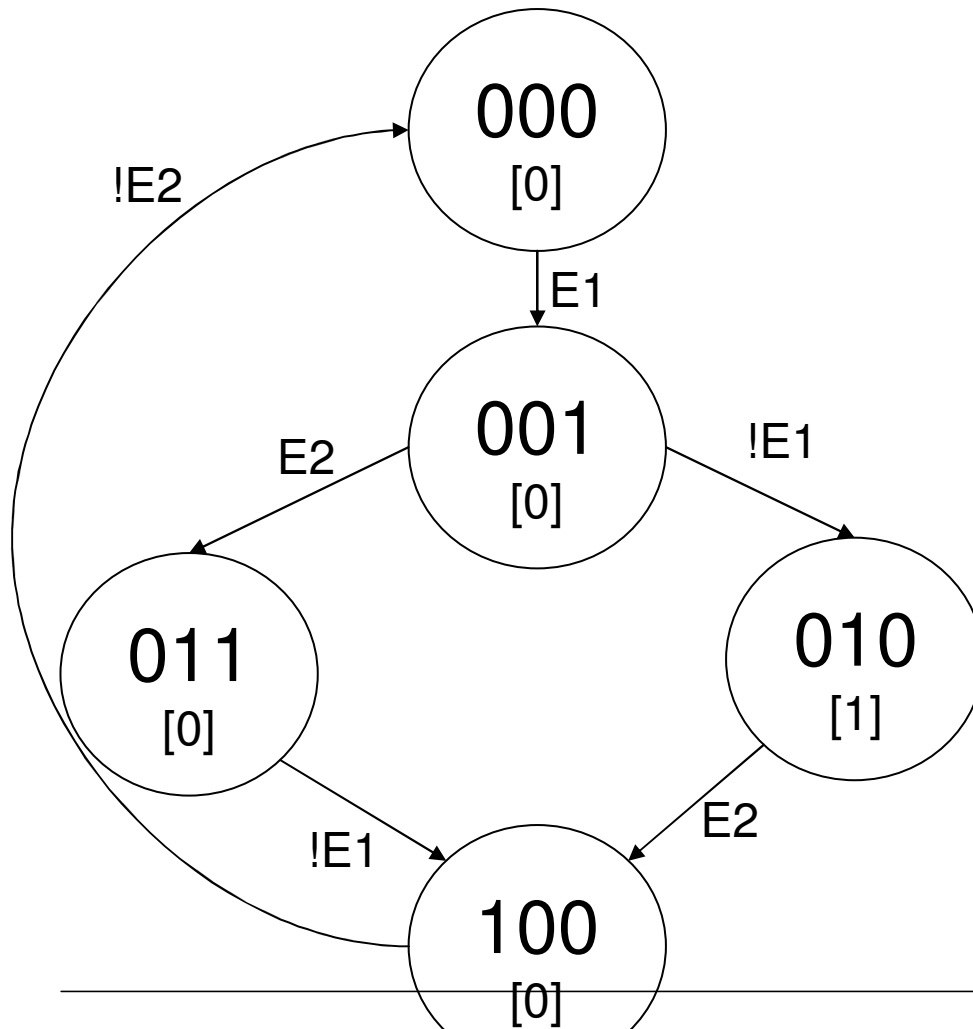
Zustandsgraph



Implementierung

- Moore-Automat, es wäre auch Mealy möglich
- 5 Zustände → 3 Flip-Flops benötigt
- Automat arbeitet immer korrekt → Ungültige Eingaben (z.B. 2 Pakete im Abstand von weniger als 20cm) können zur Vereinfachung ausgeschlossen werden → Don't Care

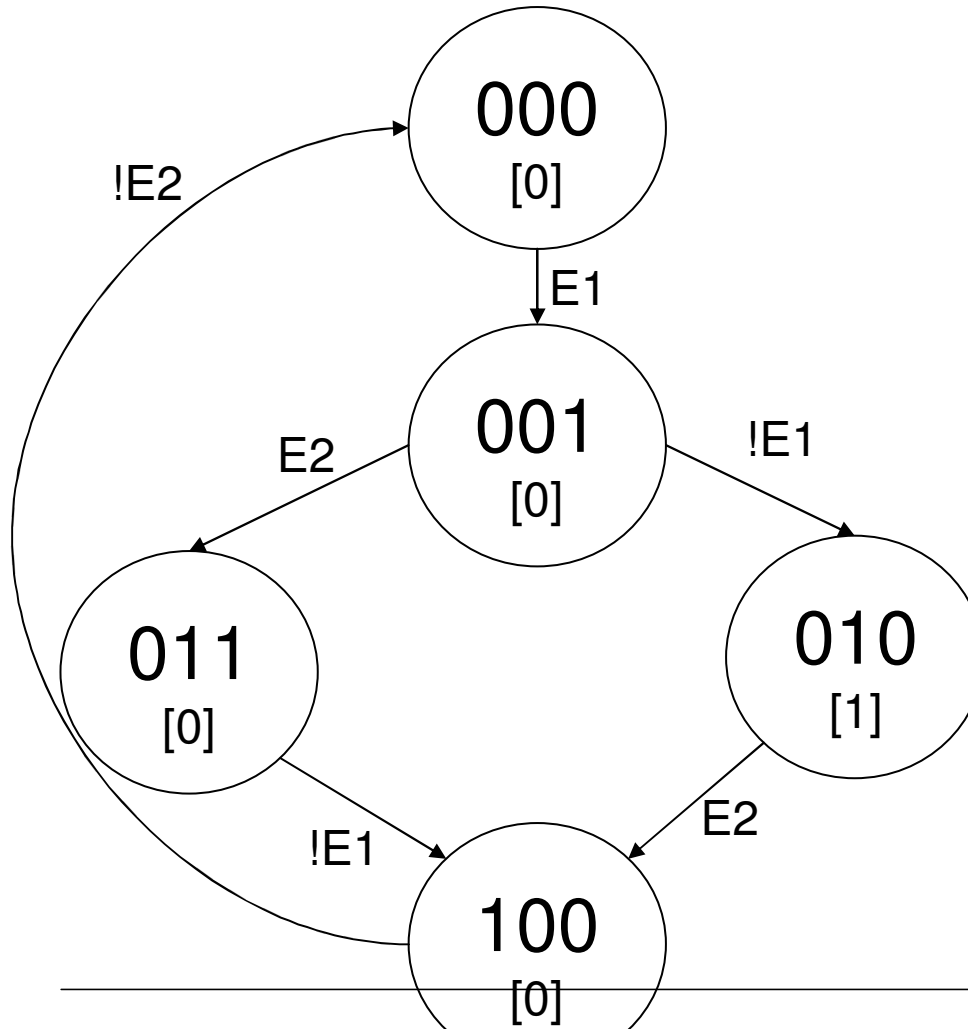
Zustandsübergangstabelle (1)



A	B	C	E1	E2	A+	B+	C+
0	0	0	0	0	0	0	0
0	0	0	0	1	X	X	X
0	0	0	1	0	0	0	1
0	0	0	1	1	X	X	X
0	0	1	0	0	0	1	0
0	0	1	0	1	X	X	X
0	0	1	1	0	0	0	1
0	0	1	1	1	0	1	1
0	1	0	0	0	0	1	0
0	1	0	0	1	1	0	0
0	1	0	1	0	X	X	X
0	1	0	1	1	X	X	X
0	1	1	0	0	X	X	X
0	1	1	0	1	1	0	0
0	1	1	1	0	X	X	X
0	1	1	1	1	0	1	1



Zustandsübergangstabelle (2)



A	B	C	E1	E2	A+	B+	C+
1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	X	X	X
1	0	0	1	1	X	X	X
1	0	1	0	0	X	X	X
1	0	1	0	1	X	X	X
1	0	1	1	0	X	X	X
1	0	1	1	1	X	X	X
1	1	0	0	0	X	X	X
1	1	0	0	1	X	X	X
1	1	0	1	0	X	X	X
1	1	0	1	1	X	X	X
1	1	1	0	0	X	X	X
1	1	1	0	1	X	X	X
1	1	1	1	0	X	X	X
1	1	1	1	1	X	X	X

Ausgangstabelle

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	X
1	1	0	X
1	1	1	X

$$OUT = B\bar{C}$$

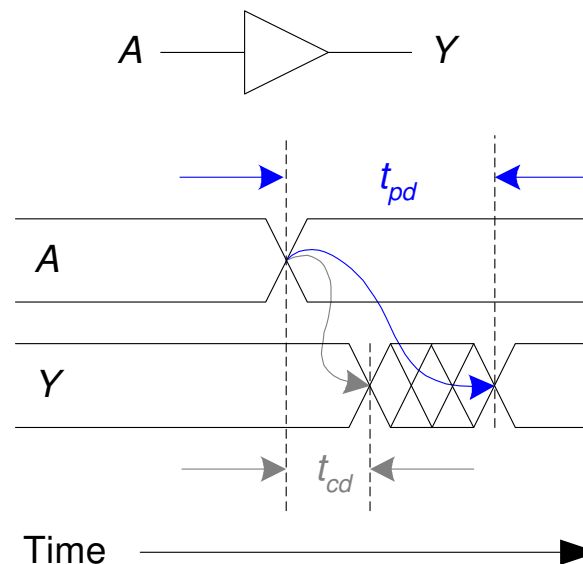
Erklärung Boole'sche Algebra

- Vereinfachungsregel: $A + AB = A$

- Vereinfachungsregel: $A + \overline{A}B = A + B$

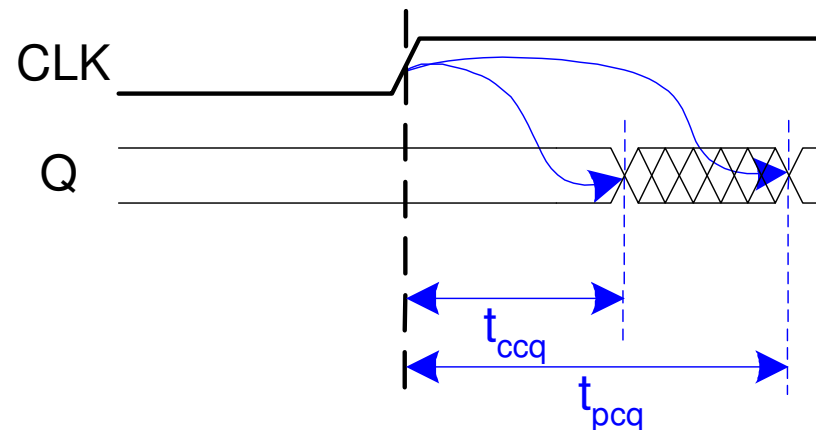
Timing (kombinatorische Gatter)

- Gatter benötigen Zeit, um einen Wechsel der Eingangssignale zu verarbeiten
- t_{pd} : Propagation Delay, Zeit, die ein Gatter höchstens benötigt
- t_{cd} : Contamination Delay, Zeit, die ein Gatter mindestens benötigt



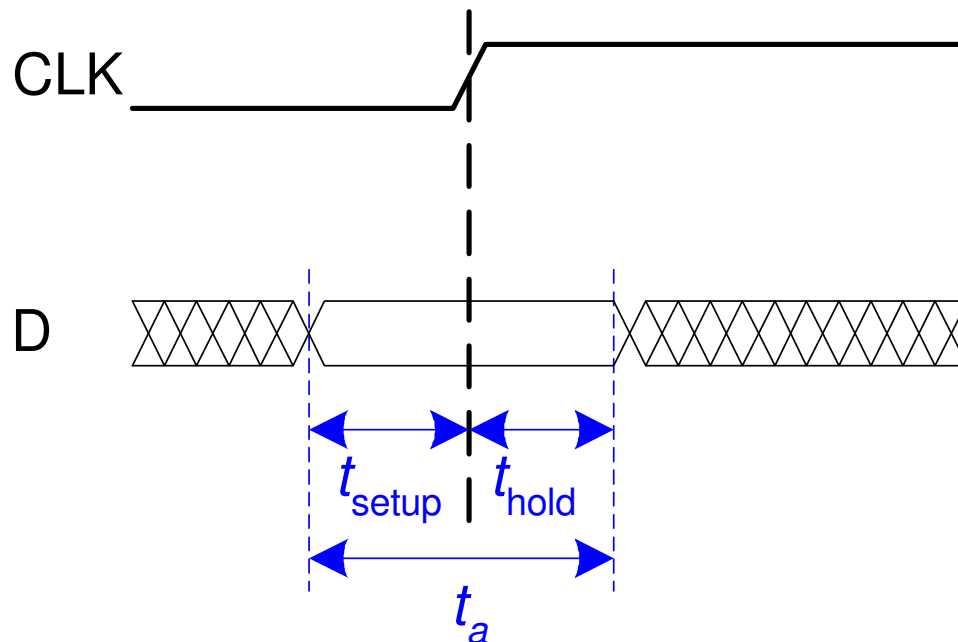
Timing (Flip-Flops)

- Flip-Flop benötigen Zeit, um einen Wechsel der Eingangssignale (nach der Taktflanke) zu verarbeiten
- t_{pcq} : Propagation Delay, Zeit, die ein FF höchstens benötigt
- t_{ccq} : Contamination Delay, Zeit, die ein FF mindestens benötigt

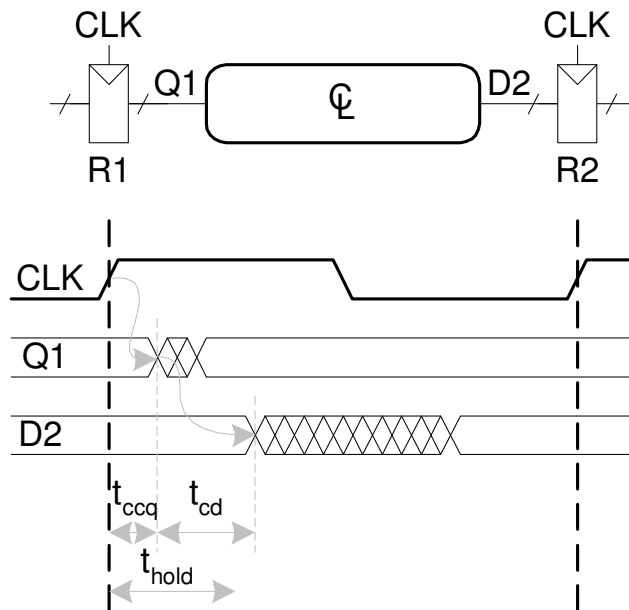


Timing (Flip-Flops 2)

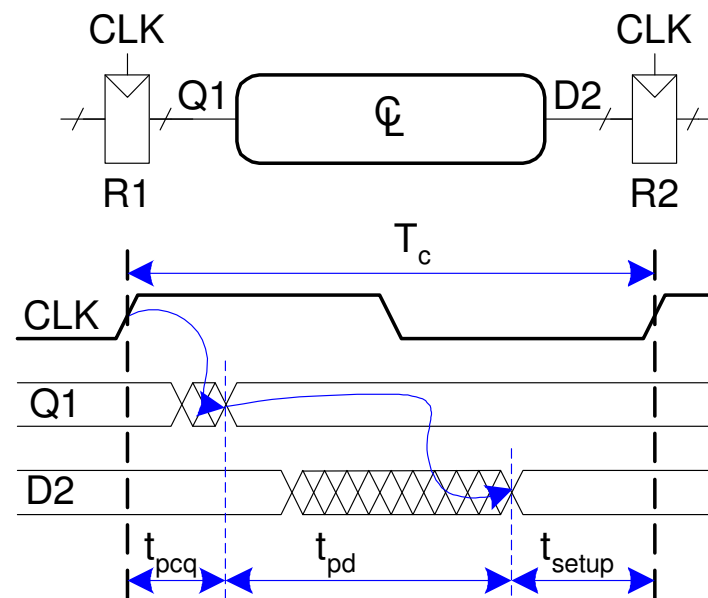
- Um die Taktflanke herum muss das Eingangssignal stabil sein
- t_{setup} : Zeit vor der Taktflanke
- t_{hold} : Zeit nach der Taktflanke



Berechnung Hold und Setupzeit

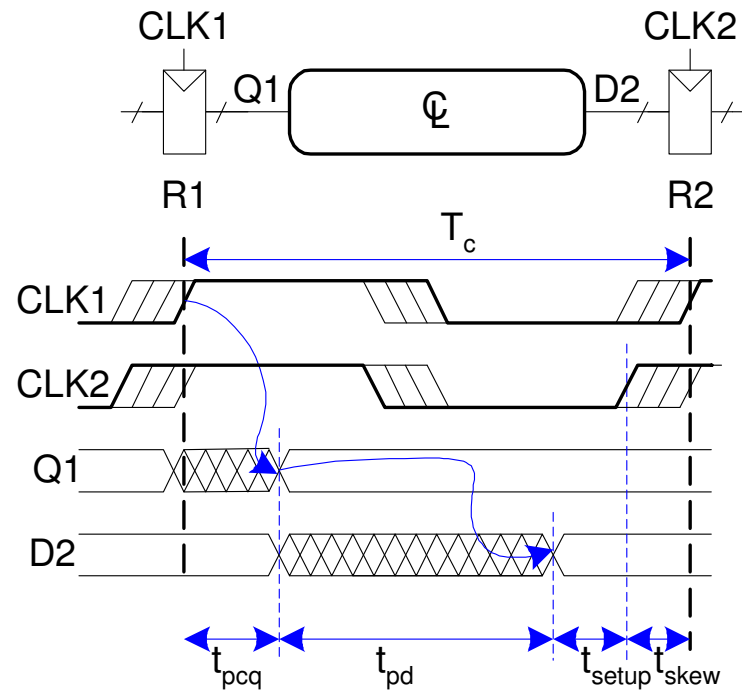


$$t_{hold} < t_{ccq} + t_{cd}$$



$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

Clock-Skew



$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}$$

$$t_{hold} < t_{ccq} + t_{cd} + t_{skew}$$

Zur Klausur

- 15.12.09, 18:00 – 20:00
- Nächste Woche: Liste mit erhaltenen Zulassungsscheinen, sortiert nach PO, bitte überprüfen!
- KEINE Hilfsmittel (Ausnahme: ein Wörterbuch für ausländische Studierende)
- Stift (Kugelschreiber oder Tinte, blau oder schwarz) sowie Lineal natürlich erlaubt
- Studenten- und Lichtbildausweis
- Essen und Trinken OK, aber Rücksicht (keine Popcorntüte 😊)

Noch Fragen?

- Jetzt stellen
- Oder die Sprechstundentermine vor der Klausur nutzen (ruhigere Atmosphäre, viel Zeit für individuelle Betreuung):

24.11.2009	13:00 - 14:00	E302
25.11.2009	10:00 - 11:00	A213
26.11.2009	13:00 - 14:00	E202
01.12.2009	8:30 - 9:30	E102
02.12.2009	12:00 - 13:00	E202
08.12.2009	13:00 - 14:00	A126
10.12.2009	11:30 - 13:30	A313
11.12.2009	12:00 - 13:00	E302