12.2 Flipflops

RS-Flipflops

Analyse von Schaltplänen

SP = ($\vec{X}_n$, G, typ, in, out, $\vec{Y}_m$) mit

G *nicht notwendigerweise* azyklisch.

Bemerkung:

Flipflop wird im folgenden mit FF abgekürzt.

Interessant für uns sind:

Belegungen $a = (a_1, \ldots, a_n) \in \mathbf{B}^n$ und Abbildungen $\Phi_a : S \rightarrow \{0,1\}$, die den Signalen *stabile* Belegungen zuordnen:

Interessant für uns sind:

Belegungen $a = (a_1, \ldots, a_n) \in \mathbf{B}^n$ und Abbildungen

$\Phi_a : S \rightarrow \{0,1\}$, die den Signalen *stabile* Belegungen zuordnen:

$\Phi_a(s) = 0,$ falls $(0, s) \in E$

$\Phi_a(s) = 1,$ falls $(1, s) \in E$

$\Phi_a(s) = a_i,$ falls $(x_i, s) \in E$

Interessant für uns sind:

Belegungen $a = (a_1, \ldots, a_n) \in \mathbf{B}^n$ und Abbildungen

$\Phi_a : S \to \{0,1\}$, die den Signalen *stabile* Belegungen zuordnen:

$\Phi_a(s) = 0,$ falls $(0, s) \in E$

$\Phi_a(s) = 1,$ falls $(1, s) \in E$

$\Phi_a(s) = a_i,$ falls $(x_i, s) \in E$

Für $m \in M$, $in(m) = s_1 \ldots s_k$, $out(m) = t_1 \ldots t_l$

$(\Phi_a(t_1), \ldots, \Phi_a(t_l)) = g_{typ(m)}(\Phi_a(s_1), \ldots, \Phi_a(s_k))$

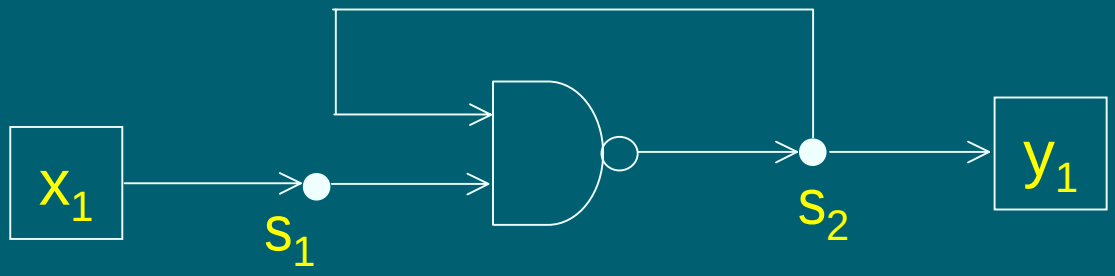
Interessant (ff)

Es ist möglich,

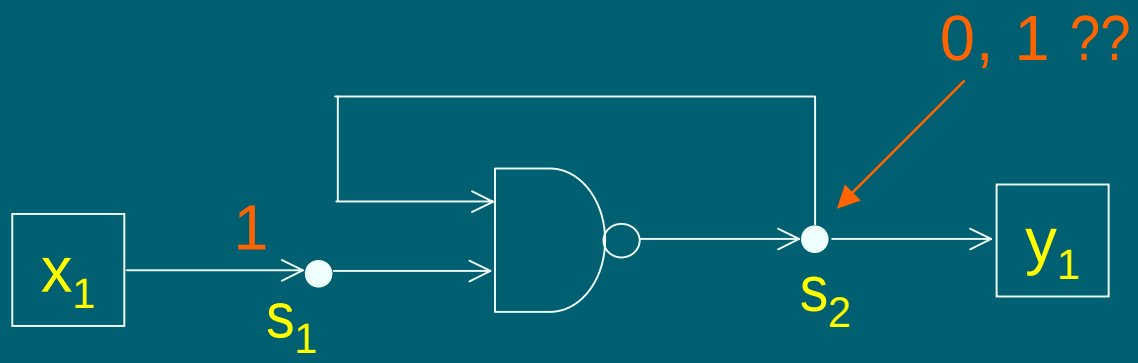
dass es zu einer Eingangsbelegung a

- *keine* stabile Signalbelegung Φ_a gibt
- *mehrere* stabile Signalbelegungen Φ_a gibt

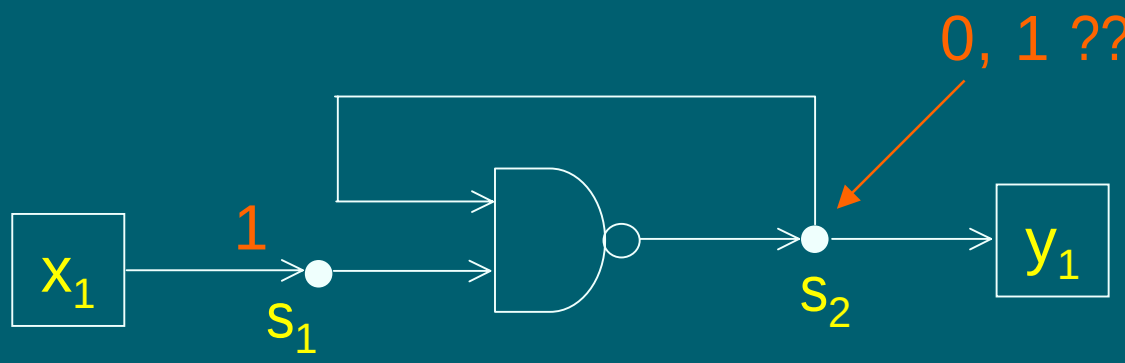
Beispiel 1



Beispiel 1



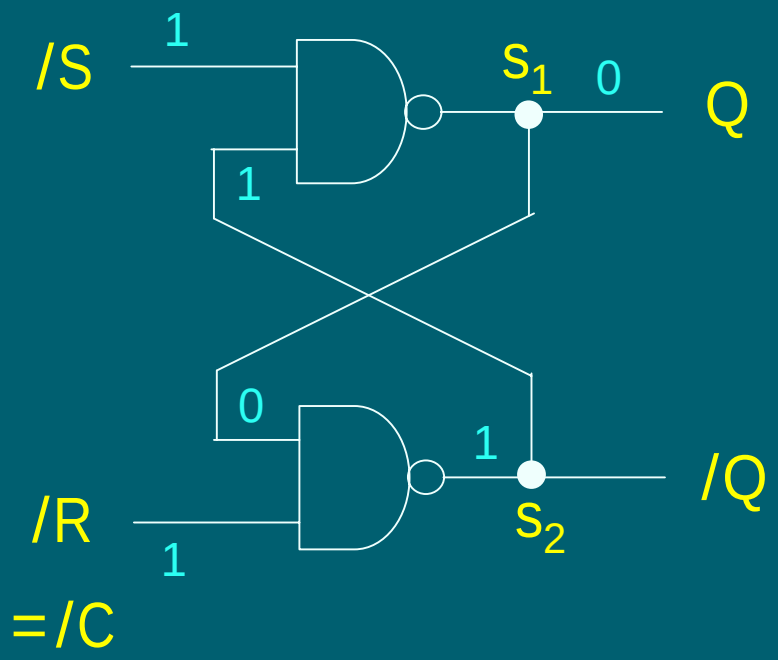
Beispiel 1



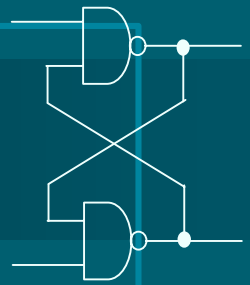
Für a = 1 existiert keine stabile Signalbelegung

(siehe s_2 !)

Beispiel 2



Beispiel 2 (ff)

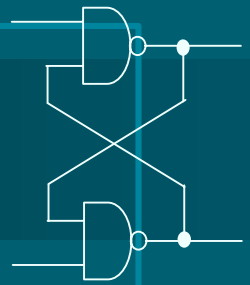


Für Eingangsbelegung /S = 1, /R = 1 gibt es zwei stabile Belegungen $\Phi_{(1,1)}$ und $\Phi'_{(1,1)}$:

$$\Phi_{(1,1)}(s_1) = 0, \quad \Phi_{(1,1)}(s_2) = 1 \quad \text{und}$$

$$\Phi'_{(1,1)}(s_1) = 1, \quad \Phi'_{(1,1)}(s_2) = 0$$

Beispiel 2 (ff)



Für Eingangsbelegung /S = 1, /R = 1 gibt es zwei stabile Belegungen $\Phi_{(1,1)}$ und $\Phi'_{(1,1)}$:

$$\Phi_{(1,1)}(s_1) = 0, \quad \Phi_{(1,1)}(s_2) = 1 \quad \text{und}$$

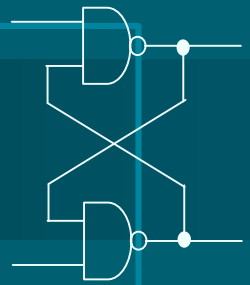
$$\Phi'_{(1,1)}(s_1) = 1, \quad \Phi'_{(1,1)}(s_2) = 0$$

Diese Schaltung nennt man RS-Flipflop,

bei Signalbelegung Φ spricht man von Zustand Q = 0,

bei Φ' von Zustand Q = 1

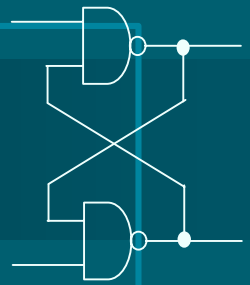
Frage:



Bei Eingangsbelegung /S = 1, /C = 1 hat das RS-FF 2 stabile Zustände.

Wie kann man von einem Zustand in einen anderen kommen?

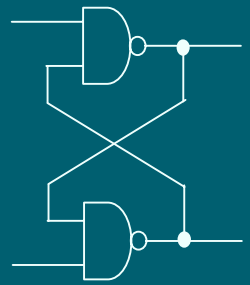
Übergang



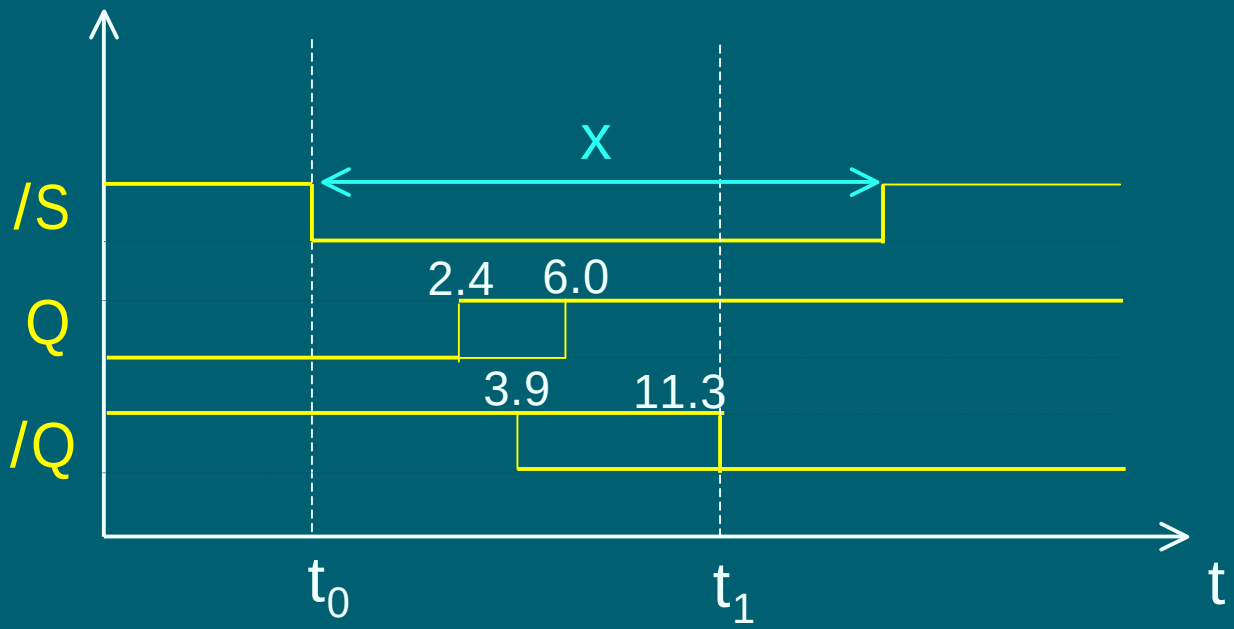
Zustand Q = 0 $\Rightarrow$ Zustand Q = 1:

- Senke /S zur Zeit t_0 ab und hebe zu $t_0 + x$ wieder an
  (einen solchen Signalverlauf nennt man *Puls*)
- Nach Zeit $t_{P/SQ}$ ist Q = 1
- Nach Zeit $t_{P/S/Q}$ ist /Q = 0
- Wähle x so, dass kein Spike entsteht!

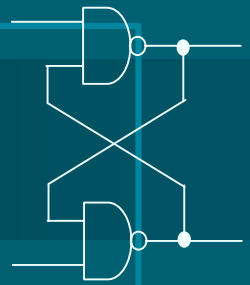
Übergang - graphisch



NAND	min	max
t_{PLH}	2.4	6.0
t_{PHL}	1.5	5.3



Spikefreier Übergang



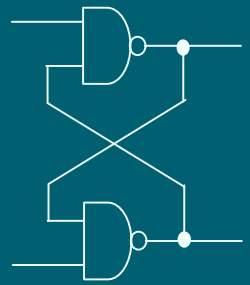
Nach den Regeln des spikefreien Umschaltens von Gattern entsteht kein Spike, falls

$(t_0 + x) - (t_0 + 11.3) \geq 11.0 \Leftrightarrow \underline{x \geq 22.3 \text{ ns}}$

Wechsel von Zustand Q = 1 zu Zustand Q = 0

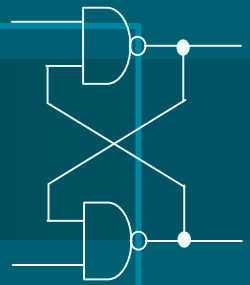
aus Symmetriegründen analog.

Symbole und Bezeichnungen



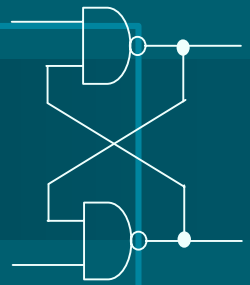
Symbol	Bezeichnung	min	max
x	Pulsweite	22.3	
$t_{P/SQ}$	Verzögerungszeit von /S bis Q	2.4	6.0
$t_{P/S/Q}$	Verzögerungszeit von /S bis /Q	3.9	11.3
$t_{P/RQ}$	Verzögerungszeit von /R bis Q	3.9	11.3
$t_{P/R/Q}$	Verzögerungszeit von /R bis /Q	2.4	6.0

Weitere Bezeichnungen



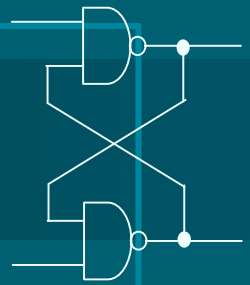
- Umschalten des FF in Zustand Q = 1 heißt Setzen (set).
- Umschalten des FF in Zustand Q = 0 heißt Zurücksetzen (reset).
- /S heißt Set-Signal.
- /R = /C heißt Reset- oder Clear-Signal.

Weitere Bezeichnungen (ff)



- Weil /R, /S durch Absenken aktiviert werden, nennt man sie active low.
- Signalnamen von active-low-Signalen beginnen in der Regel mit **/**.

Flackern



Zu vermeiden ist:

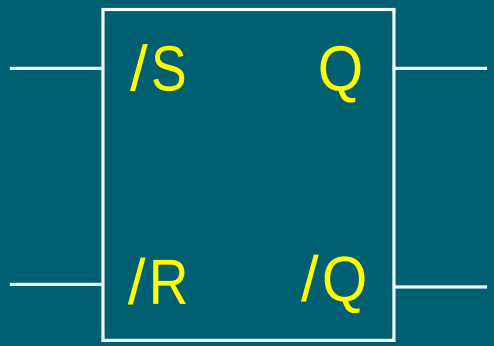
- /S, /R beide aktiv, d.h. /S = 0, /R = 0

  → Q = 1, /Q = 1

- Wenn dann
  - /S, /R gleichzeitig inaktiv und
  - Gatter gleich schnell schalten

→ *Flackern* = metastabiler Zustand

Schaltsymbol für RS-FF

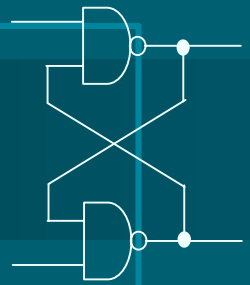


Beispiel:

FAST-Baustein 74F74 enthält 2 FF,

die als RS-FF nutzbar sind

Nachteil von RS-FF



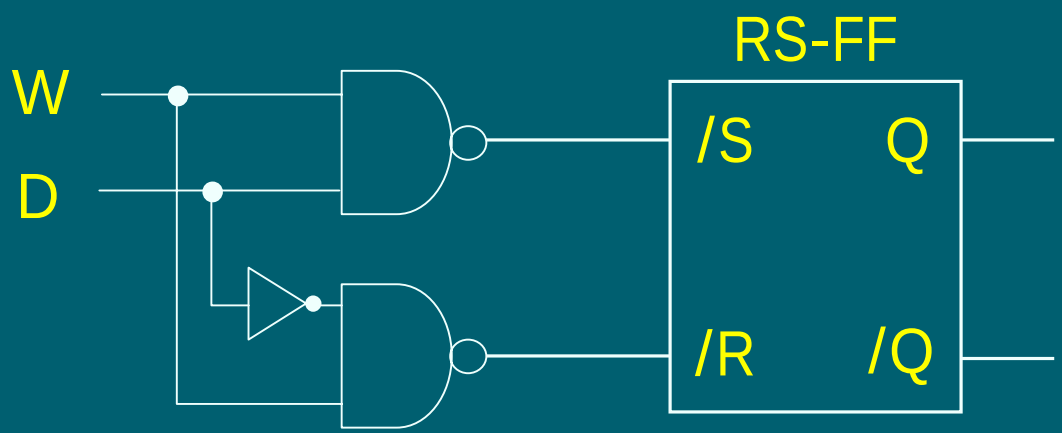
Beim Speichern eines Wertes 0 oder 1 muss man den Wert kennen:

- 0 → Aktiviere /R
- 1 → Aktiviere /S

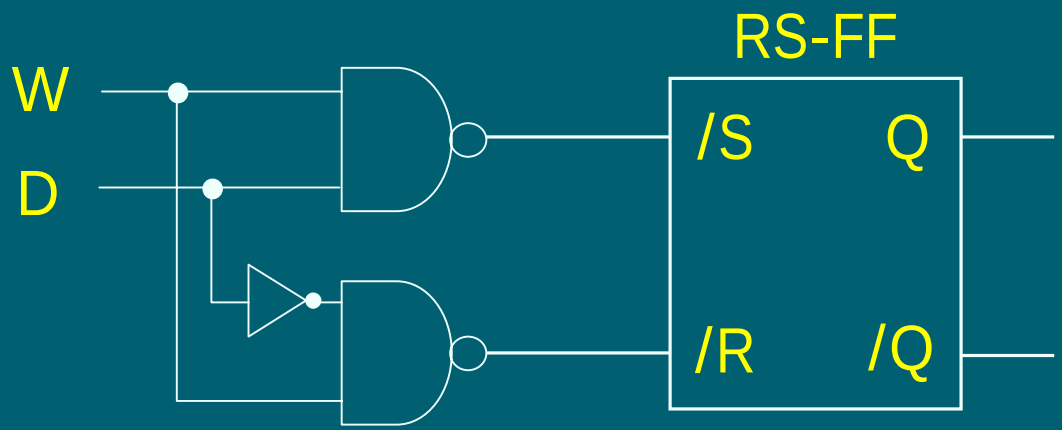
Ziel:

Speichern *unbekannter* Werte

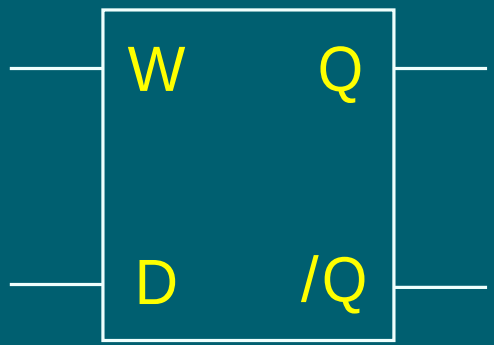
D-Latch - graphisch



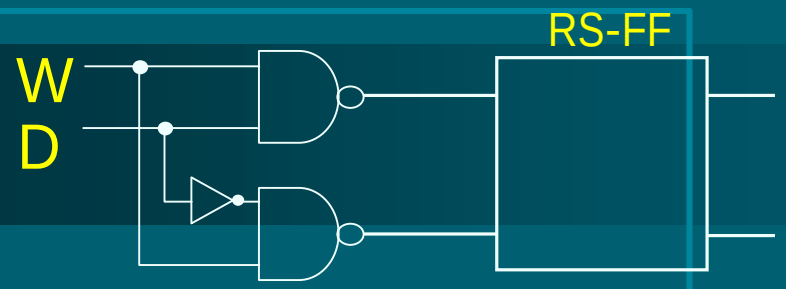
D-Latch - graphisch



Symbol:



D-Latch

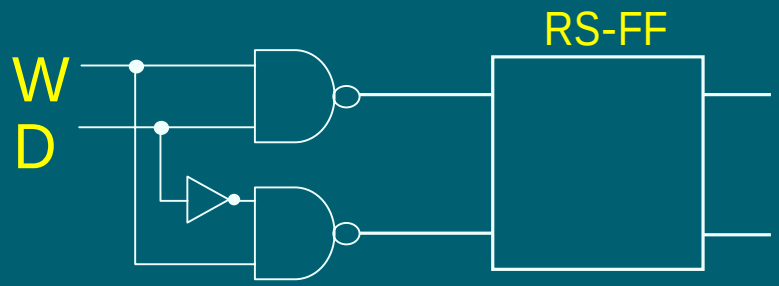


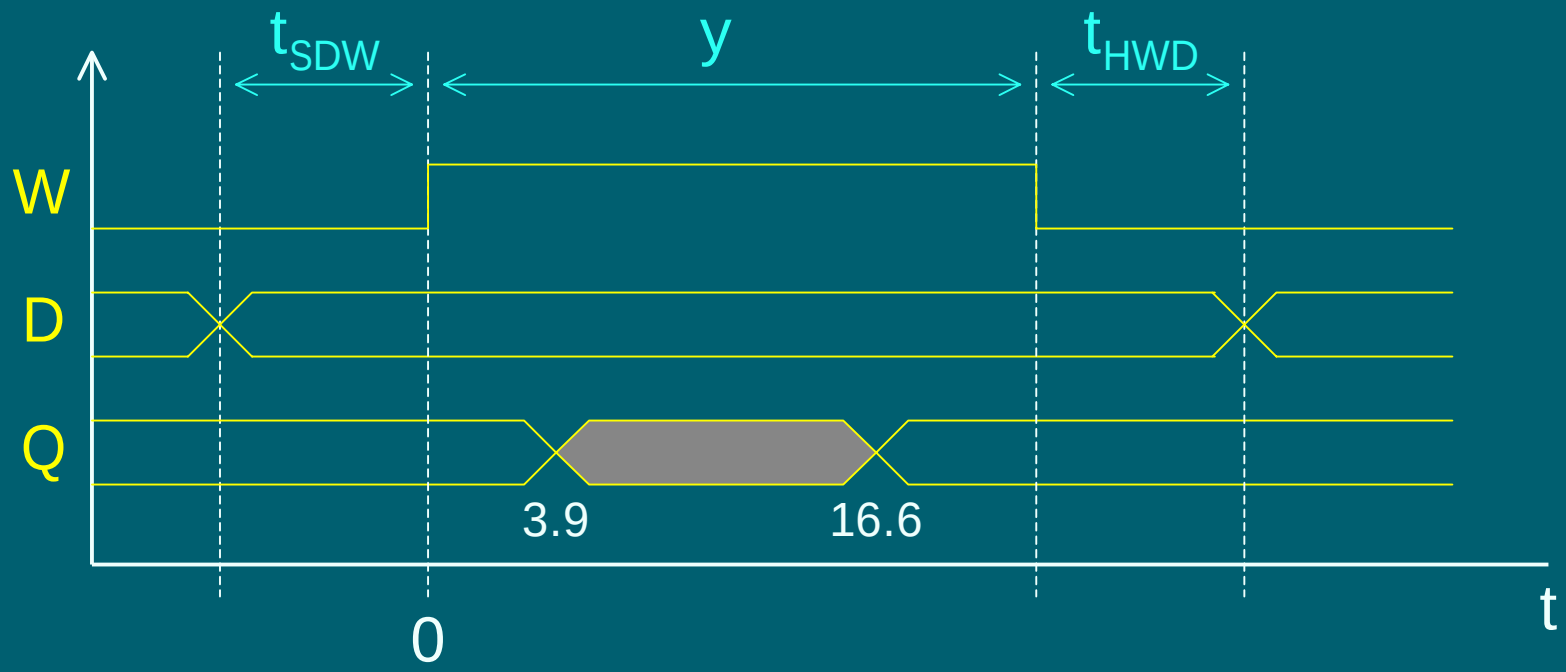
W ist active high.

$W = 0 \Rightarrow$ /S, /R inaktiv

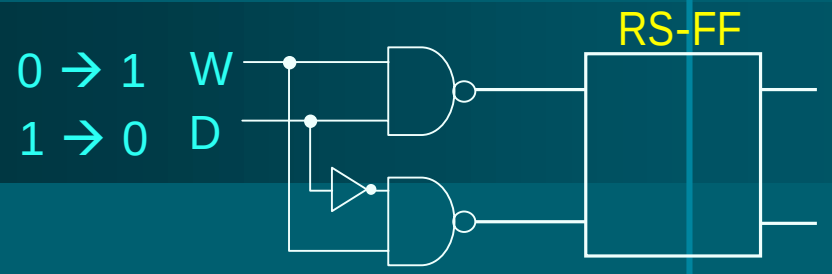
$W = 1 \Rightarrow$
- /S aktiv, falls $D = 1$
- /R aktiv, falls $D = 0$

Timing-Diagramm



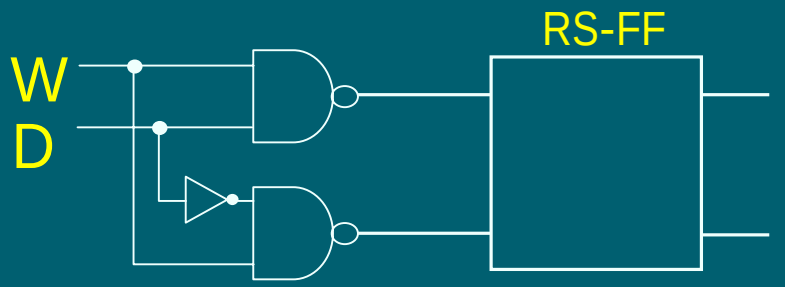


Timing-Bedingungen für das D-Latch



- W muss beim Schreiben lange genug 1 sein, um minimale Pulsweite x des RS-FFs zu garantieren
- Vor W: 0 → 1 werden Daten für Zeit t_{SDW} stabil gehalten, um Spikes auf /S, /R zu vermeiden
- Nach W: 1 → 0 werden Daten für Zeit t_{HWD} stabil gehalten, um Spikes auf /S, /R zu vermeiden

Symbole und Bezeichnungen



Symbol	Bezeichnung	min	max
y	Pulsweite des Schreibpulses	25.2	
t_{SDW}	Setupzeit von D bis W	16.3	
t_{HWD}	Holdzeit von W nach D	11.0	
t_{PWQ}	Verzögerungszeit von W bis Q	3.9	16.6
(t_{PDQ}	Verzögerungszeit von D bis Q	3.9	22.6)

Lemma 12.1

Der Schreibvorgang beim D-Latch funktioniert mit den Parameterwerten aus der Tabelle.

Beweis: siehe Übung

Hinweise zum Beweis des Lemmas

<u>Voraussetzung:</u>

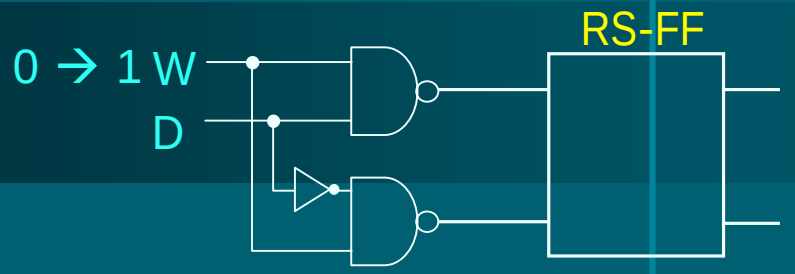
Pulsweite und Verzögerungszeiten für RS-FF

wie in Vorlesung hergeleitet.

Inverter (74F04), NAND (74F00) mit folgenden Werten:

	NAND		NOT	
	min	max	min	max
t_{PLH}	2.4	6.0	2.4	6.0
t_{PHL}	1.5	5.3	1.5	5.3

Hinweise ff
Setupzeit t_{SDW} mit Fallunterscheidung



D : 0 → 1 :

- Kein Spike an oberem NAND
- unteres NAND: $\overline{D}$: 1 → 0 nach (1.5, 5.3)
  W : 0 → 1 mind. 11 ns danach

→ $t_{SDW} \geq 16.3$ ns

D : 1 → 0 :

- oberes NAND: W : 0 → 1 mind. 11 ns danach
  → $t_{SDW} \geq 11$ ns
- unteres NAND: kein Spike möglich

→ $t_{SDW} \geq 16.3$ ns

Weitere Eigenschaften eines D-Latches

- Das D-Latch heißt transparent, wenn das Schreibsignal aktiv ist.
- Hält man W lange aktiv und ändert D zur Zeit t, dann ändert sich Q zur Zeit $t + t_{PDQ}$.

- Das D-Latch ist pulsgesteuert (Schreibpuls W)

Beispiel:

FAST-Baustein 74F73 enthält 8 D-Latches.

D-Flipflop

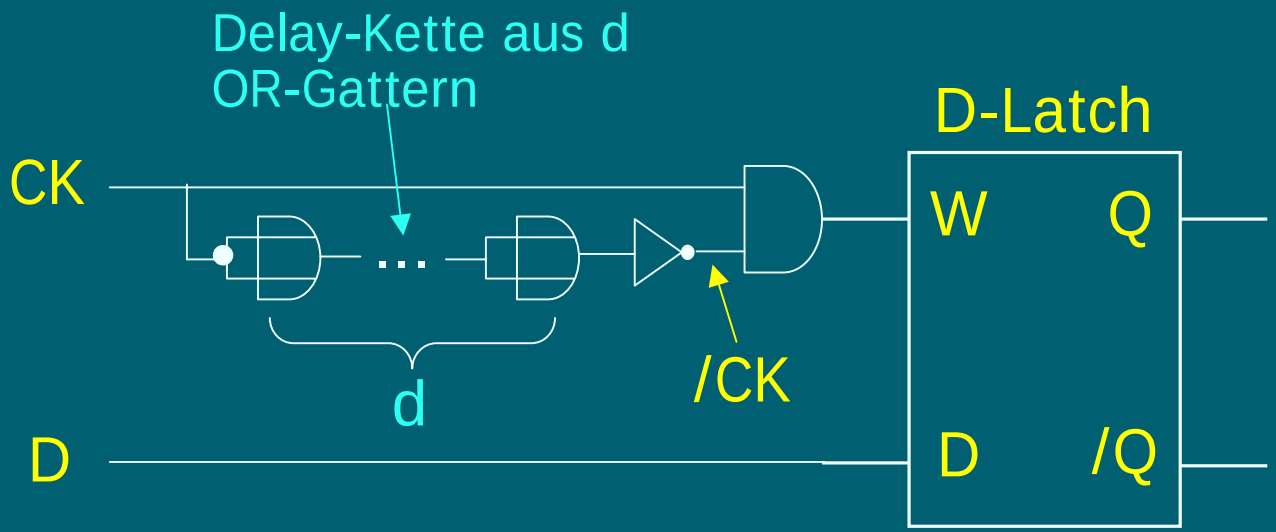
Taktflankengesteuerte Flipflops wie das D-Flipflop übernehmen Daten zu einem bestimmten Zeitpunkt (kein transparenter Modus!), nämlich bei der steigenden Flanke des sog. Clocksignals

Vorteil:

Daten müssen lediglich bei der steigenden Taktflanke stabil sein (zzgl. Setup- und Holdzeit)

D-Flipflop aus D-Latch

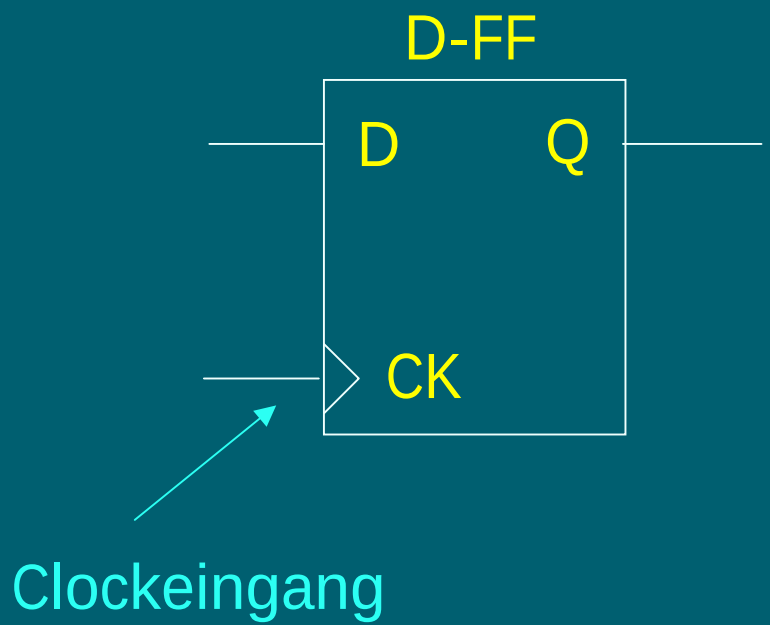
Prinzipielle Möglichkeit, ein D-FF aus einem D-Latch zu konstruieren:



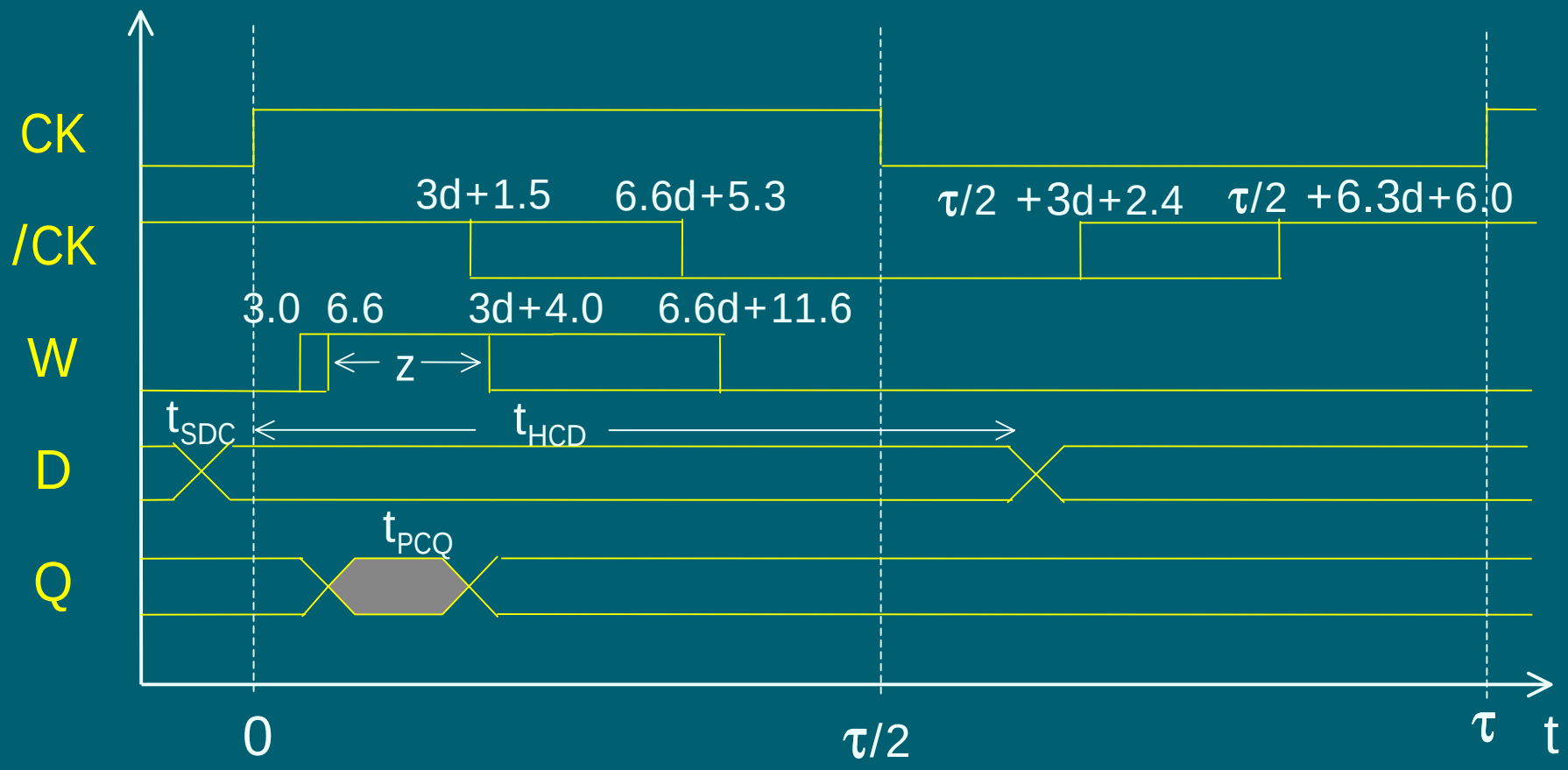
Idee:

Erzeuge mit Delay-Kette aus Clocksignal einen Schreibpuls

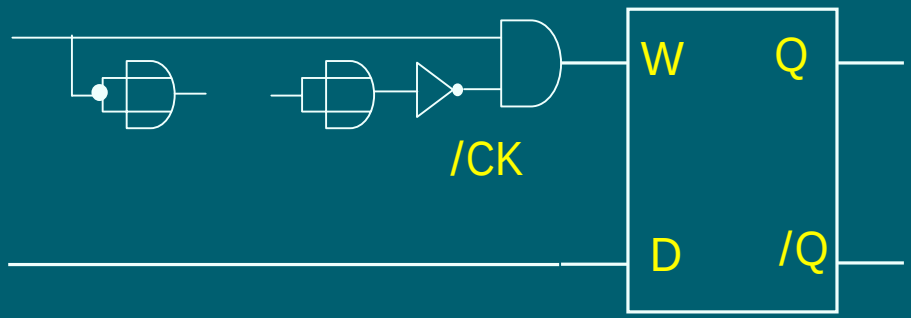
D-Flipflop: Symbol



Timing-Diagramm



Timing: Berechnungen



Verzögerungszeiten der Gatter:

	AND		OR		NOT	
	min	max	min	max	min	max
t_{PLH}	3.0	6.6	3.0	6.6	2.4	6.0
t_{PHL}	2.5	6.3	3.0	6.3	1.5	5.3

Timing: Berechnungen (ff)

CK : 0 → 1 zum Zeitpunkt t = 0

→ W : 0 → 1 bei

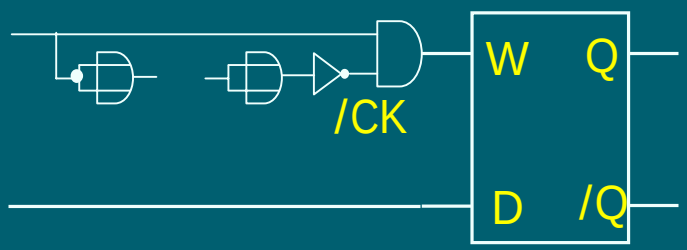
t_1 = (3.0, 6.6), da noch /CK = 1

→ /CK : 1 → 0 bei

t_2 = d • (3.0, 6.6) + (1.5, 5.3)

→ W : 1 → 0 bei

$t_3 = t_2 + (2.5, 6.3) = d \bullet (3.0, 6.6) + (4.0, 11.6)$



	AND		OR		NOT	
	min	max	min	max	min	max
t_{PLH}	3.0	6.6	3.0	6.6	2.4	6.0
t_{PHL}	2.5	6.3	3.0	6.3	1.5	5.3

Timing: Berechnungen (ff)

→ Breite des Schreibpulses

$z = \min(t_3) - \max(t_1) = 3d + 4.0 - 6.6 = 3d - 2.6$

Es muss gelten:

$3d - 2.6 \geq 25.2$ (= min. Schreibpulsweite D-Latch)

→ $d \geq 10$

Analyse mit folgenden Parametern:

- steigende Flanke von CK bei $t = 0$
- Daten stabil ab $- t_{SDC}$
- Daten stabil bis t_{HCD}
- Neue Daten am Ausgang zur Zeit t_{PCQ}

Symbole und Bezeichnungen

Symbol	Bezeichnung	min	max
t_{SDC}	Setupzeit von D bis CK	13.3	
t_{HCD}	Holdzeit von CK nach D	88.6	
t_{PCQ}	Verzögerungszeit von CK bis Q	6.9	23.2
ν	Clockfrequenz (MHz)		7.2

Berechnungen zum D-Flipflop

Für D-Latch:

$t_{SDW} \geq 16.3$, $t_{HWD} \geq 11.0$, $t_{PWQ} = (3.9, 16.6)$

→ $t_{SDC} \geq 16.3 - \min(t_1) = 16.3 - 3.0 = 13.3$

→ $t_{HCD} \geq \max(t_3) + 11.0 = 6.6d + 11.6 + 11.0 = 88.6$

bei d = 10

$t_{PCQ} = t_1 + t_{PWQ} = (3.0, 6.6) + (3.9, 16.6)$

$= (6.9, 23.2)$

Berechnungen zum D-Flipflop (ff)

Sei CK ein periodisches Signal mit Clockperiode τ ns.

Clockfrequenz von CK ist $\nu = 1/\tau$.

Voraussetzung:

CK ist symmetrisch, d.h. steigt und fällt alle $\tau/2$ ns.

Achtung!

- Übergang CK : 1 → 0 darf nicht zu früh kommen

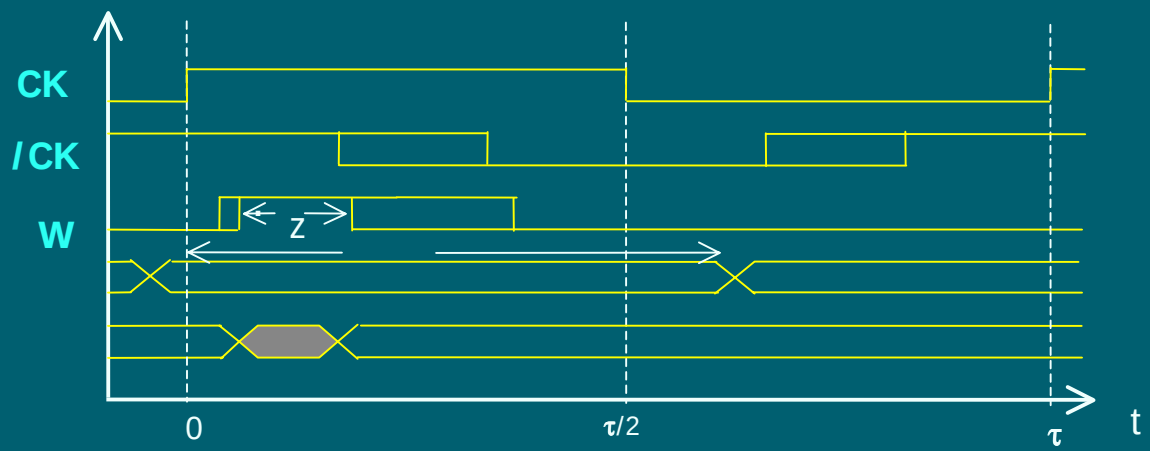
  wegen Schreibpulsweite:

  W sinkt wegen CK zur Zeit $t_4 = \frac{\tau}{2} + (2.5, 6.3)$

  → $\min(t_4) - \max(t_1) = \frac{\tau}{2} + 2.5 - 6.6 \geq 25.2$

  → $\frac{\tau}{2} \geq 29.3 \Rightarrow \tau \geq 58.6$

Mindeszeit
Schreibpuls D-Latch



Achtung! (2)

- Übergang CK : 0 $\rightarrow$ 1 nicht bevor /CK = 1 !

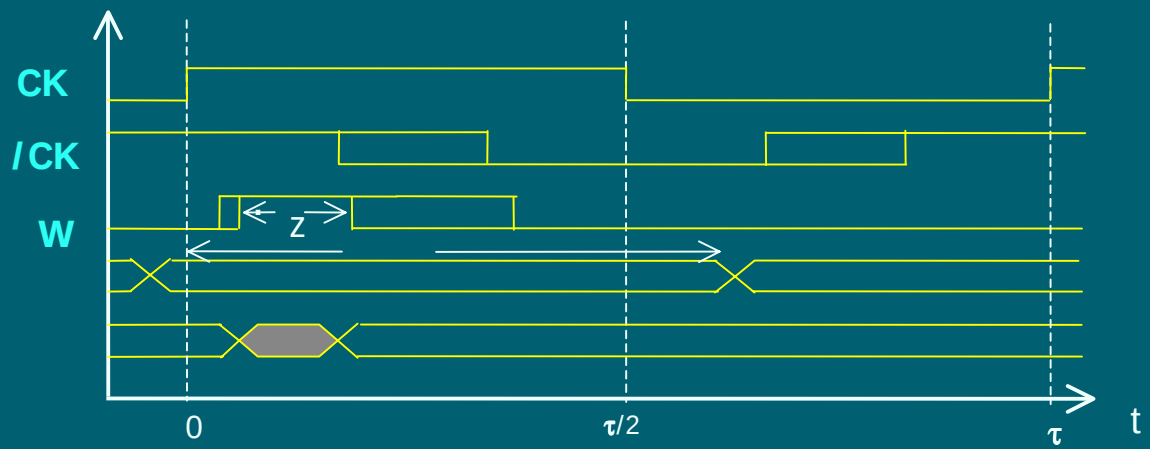
  /CK = 1 nach $t_5 = \frac{\tau}{2} + d \cdot (3.0, 6.3) + (2.4, 6.0)$

  $= (\frac{\tau}{2} + 3.0d + 2.4, \frac{\tau}{2} + 6.3d + 6.0)$

  $\Rightarrow \tau \geq \max(t_5) = \frac{\tau}{2} + 6.3d + 6.0$

  $\Rightarrow \tau \geq 12.6d + 12.0 = 138.0$

  d = 10



Achtung! (3)

Wegen $t_5 - \tau/2 \geq 3.0\text{d} + 2.4 \geq 11.3$

ist spikefreies Umschalten auf jeden Fall garantiert!

$$\Rightarrow \nu \leq \frac{1}{(138.0 \cdot 10^{-9}\text{s})}$$

$$\frac{1}{(138.0 \cdot 10^{-9}\text{s})} \geq 7.2 \cdot 10^{6}\ \text{Hz} = 7.2\ \text{MHz}$$

Bemerkung

D-Flipflop in der hier vorgestellten Realisierung verdeutlicht (nur) das Prinzip !

Reale Flipflops haben $t_{HCD} < t_{PCQ}$

→ siehe Datenblätter

Einfache Bausteine mit Flipflops

Einfache Bausteine mit Flipflops

- **Register**

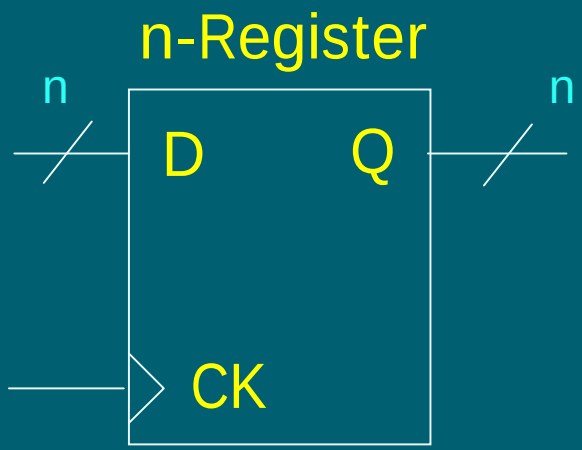
Einfache Bausteine mit Flipflops

- Register
- Zähler

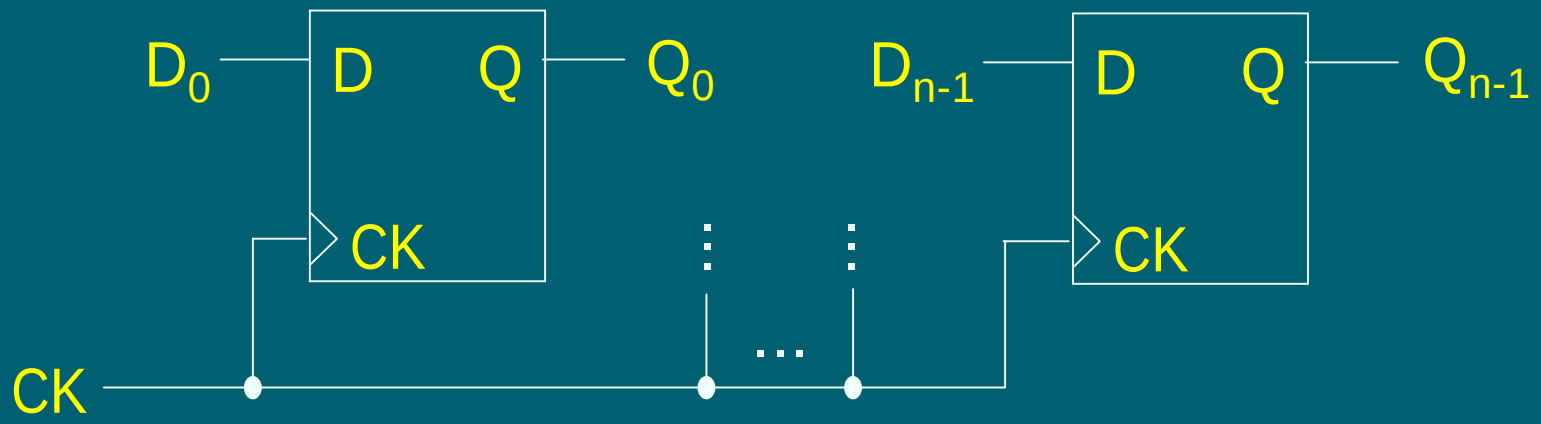
Register

n – Bit Register

= n D-Flipflops mit gemeinsamem Clocksignal



Aufbau eines Registers



(Entsprechend: n - Bit Latch

= n D-Latches mit gemeinsamem Schreibsignal W)

Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

- Dateneingänge $X = (X_{n-1}, \ldots, X_0)$

Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

- Dateneingänge $X = (X_{n-1}, \ldots, X_0)$
- Datenausgänge $Y = (Y_{n-1}, \ldots, Y_0)$

Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

- Dateneingänge $X = (X_{n-1}, \ldots, X_0)$
- Datenausgänge $Y = (Y_{n-1}, \ldots, Y_0)$
- Dateneingang c_{in} für Eingangsübertrag

Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

- Dateneingänge $X = (X_{n-1}, \ldots, X_0)$
- Datenausgänge $Y = (Y_{n-1}, \ldots, Y_0)$
- Dateneingang c_{in} für Eingangsübertrag
- Datenausgang c_{out} für Ausgangsübertrag

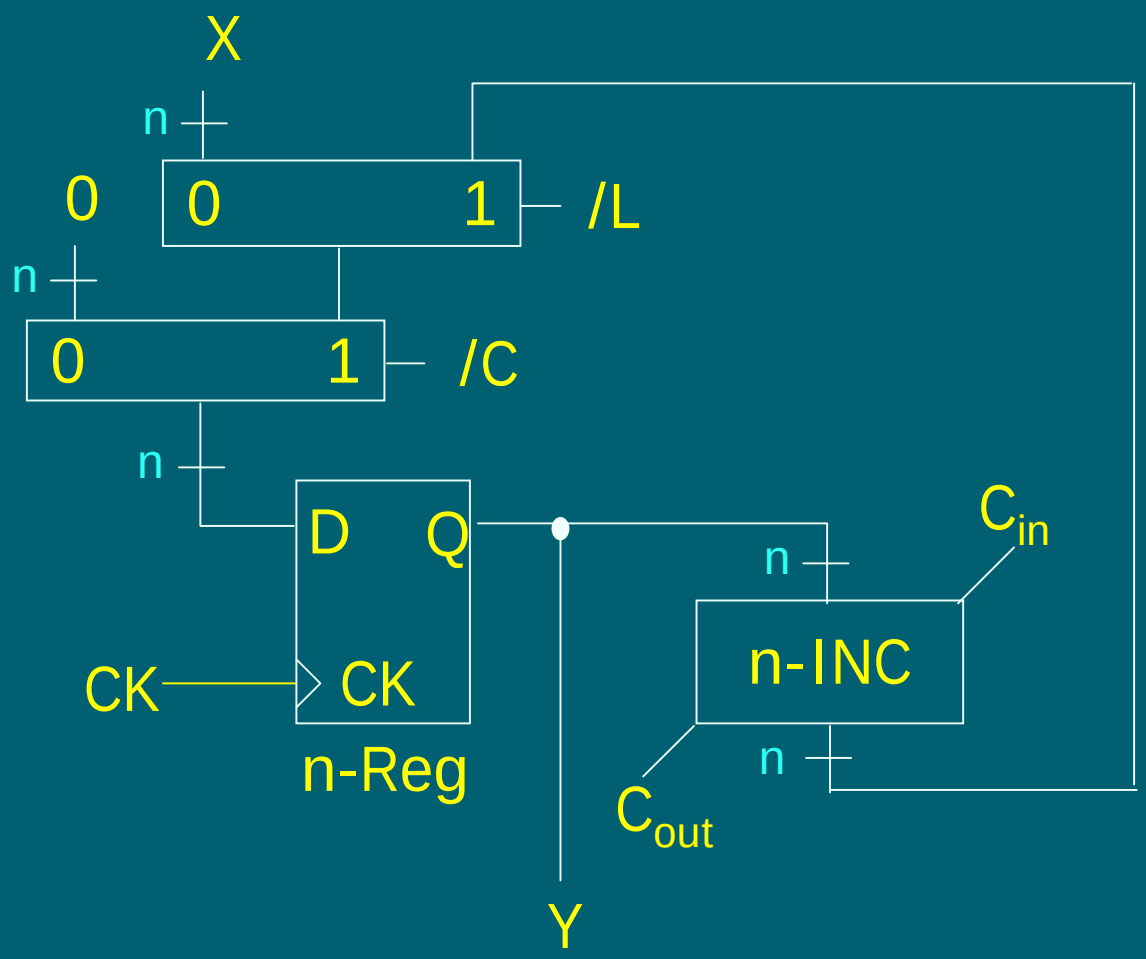
Zähler

Ein **n – Bit Zähler** ist eine Schaltung mit folgenden Ein- und Ausgängen:

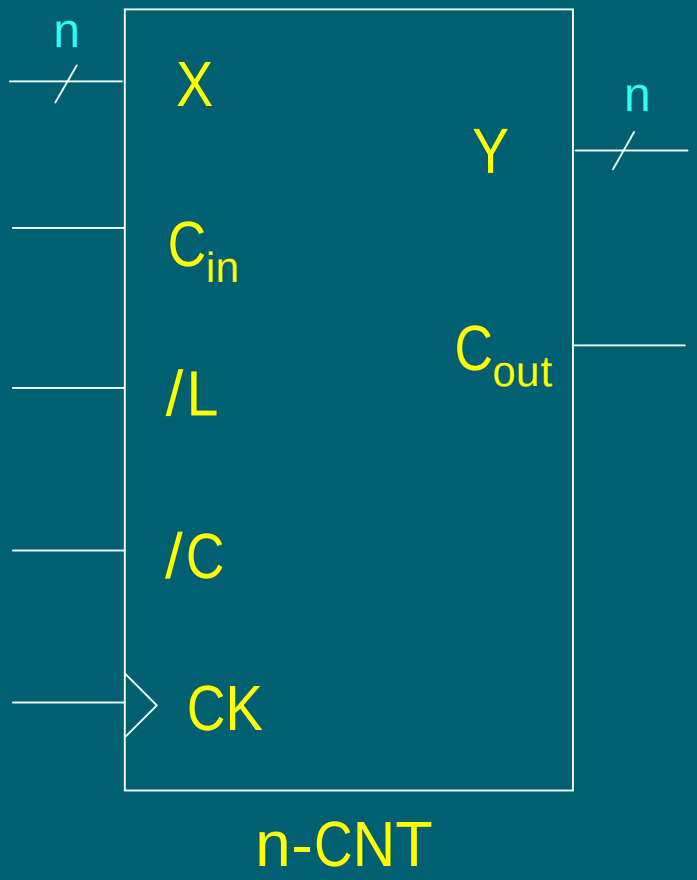
- Dateneingänge $X = (X_{n-1}, \ldots, X_0)$
- Datenausgänge $Y = (Y_{n-1}, \ldots, Y_0)$
- Dateneingang c_{in} für Eingangsübertrag
- Datenausgang c_{out} für Ausgangsübertrag
- Eingänge für Kontrollsignale:

  /C (Clear) , /L (Load) , CK (Clock)

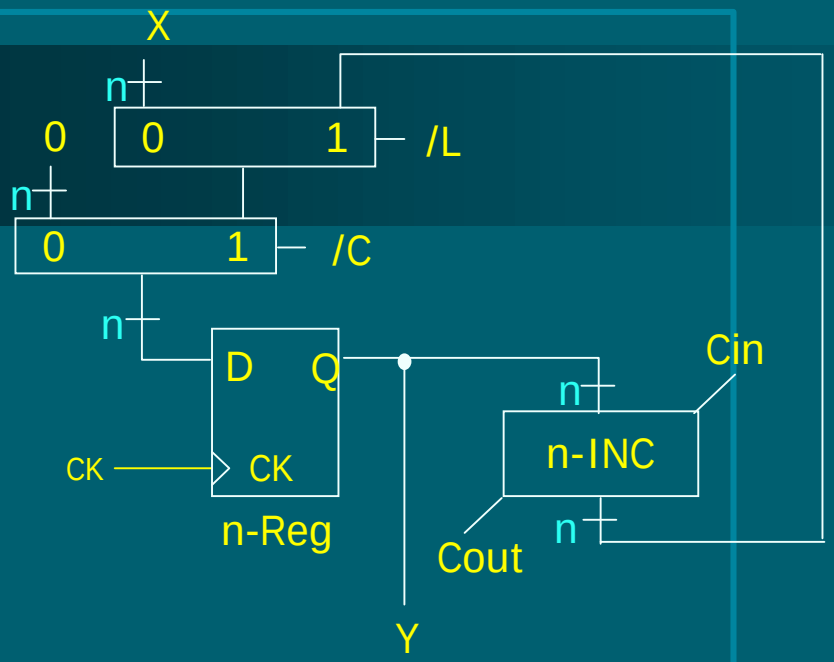
Aufbau eines Zählers



Symbol eines Zählers



Funktionsweise eines Zählers



Ein Zähler speichert ein n-Bit-Wort, das an den Ausgängen Y erscheint. (*Zählerstand*)

Bei jeder steigenden Flanke von CK wird ein neuer Zählerstand Y_{neu} gespeichert. Für Y_{neu} gilt:

$$Y_{neu} = \begin{cases} 0 \ldots 0 & , \text{falls } /C = 0 \\ X & , \text{falls } /C = 1, /L = 0 \\ bin_n((< Y > + C_{in}) \bmod 2^n) & , \text{falls } /C = 1, /L = 1 \end{cases}$$

Funktionsweise eines Zählers (ff)

Ausgangsübertrag C_{out} ermöglicht es, den Zähler zu *kaskadieren*, z.B. aus s n-Bit-Zählern einen $s \cdot n$-Bit-Zähler zu bauen.

Bsp.: Baustein 74F163 ist ein 4-Bit-Zähler.