

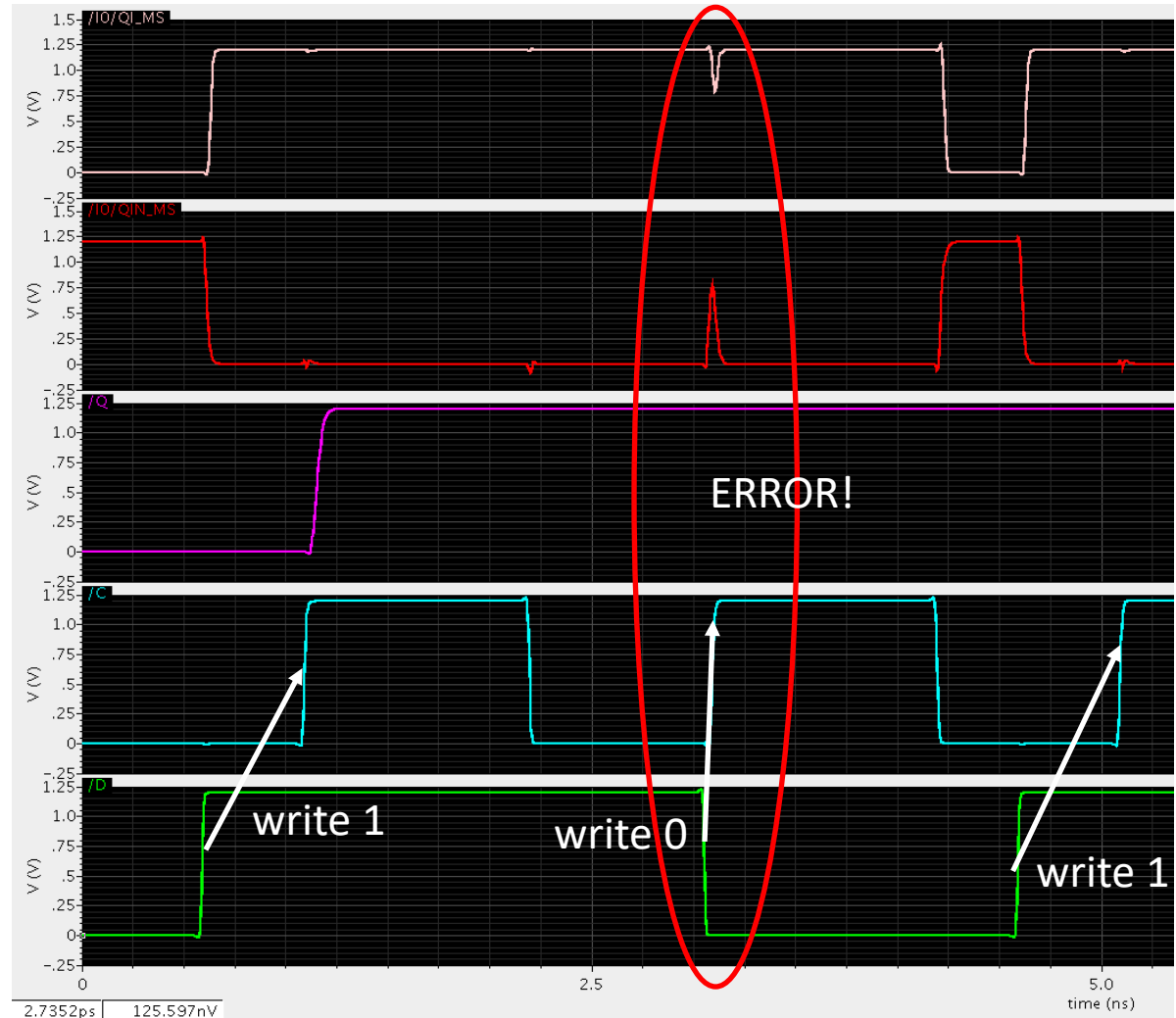
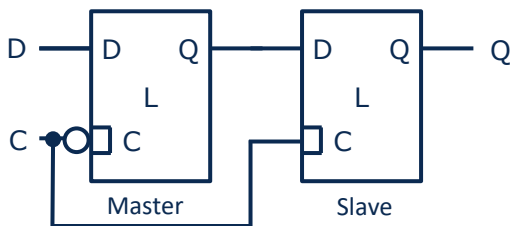
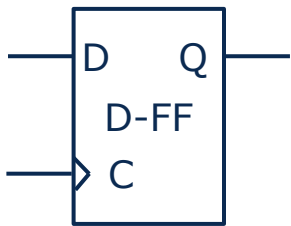
# Timing von CMOS Logik

- Timing bezeichnet das Zeitverhalten von Signalen in digitalen Schaltungen
- Betrachtung von
  - Zeitpunkten und Zeitdifferenzen von Signalwechseln und Verzögerungszeiten
  - Anstiegszeiten (Transition Time) bei Signalwechseln
- Analyse des Timings zur Einhaltung von Randbedingungen (Constraints)
- Korrekte Timing Analyse ist notwendig zum Sicherstellen von
  - Performanz
  - Funktionalität

Master  
Keeper

Master  
Latch Node  
(inverted)

D-FlipFlop

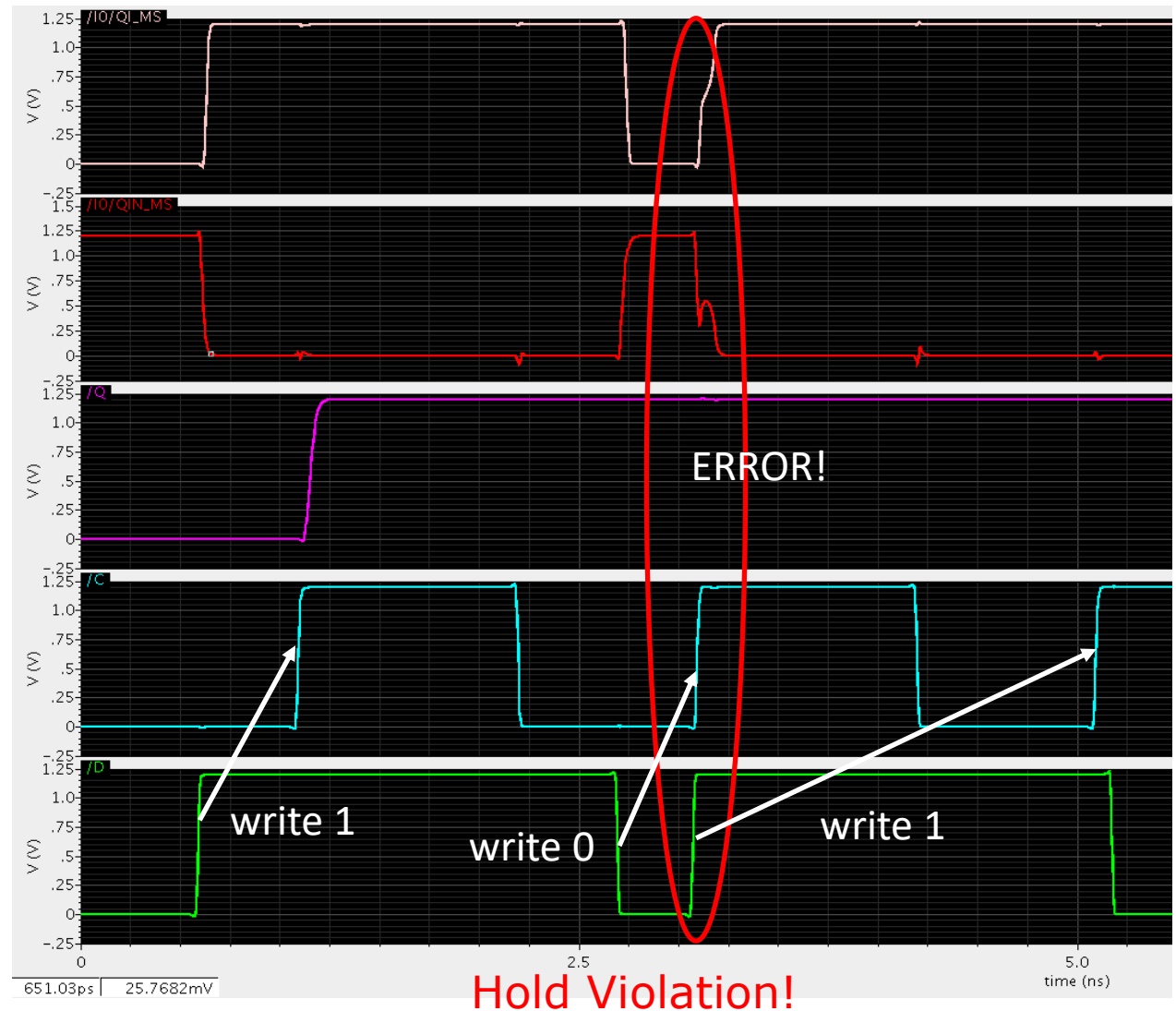
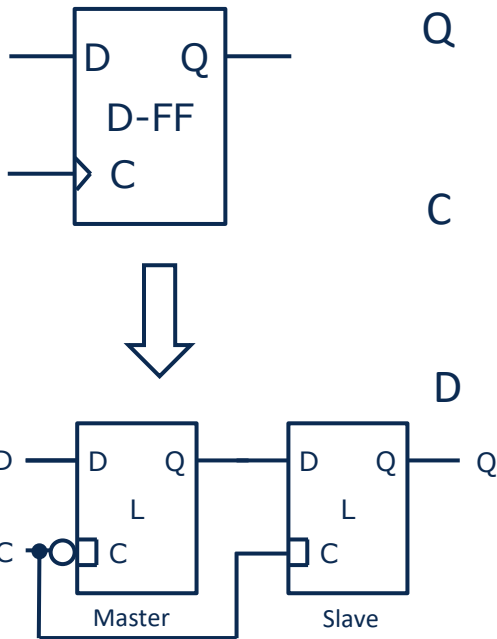


**Setup Violation!**

Master Keeper

Master Latch  
Node (inverted)

D-FlipFlop

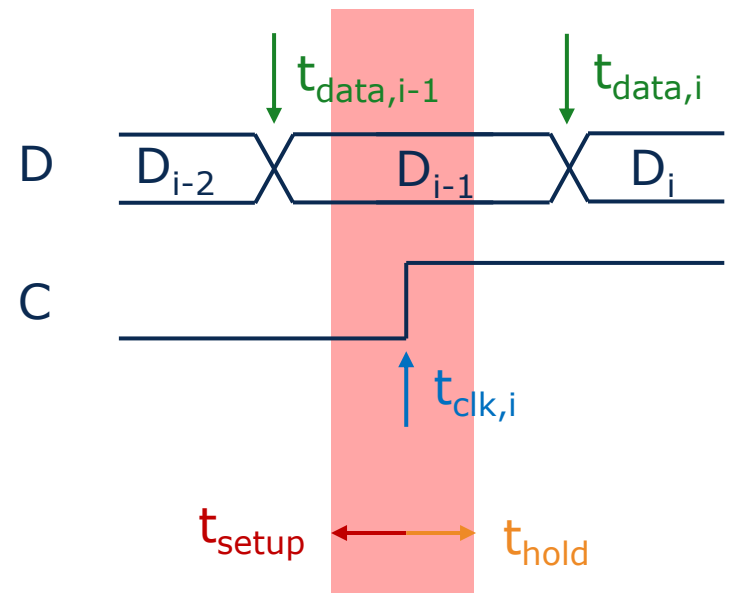


- Sequentielle Elemente (z.B. FlipFlop, Latch, ClockGate, ...)
- Kritisch ist das **Ende der transparenten Phase** des empfangenden Latches (Master-Latch im FlipFlop)
- Clock Arrival Time:  $t_{clk,i}$
- Data Arrival Time:  $t_{data,i}$

- Setup Constraint:

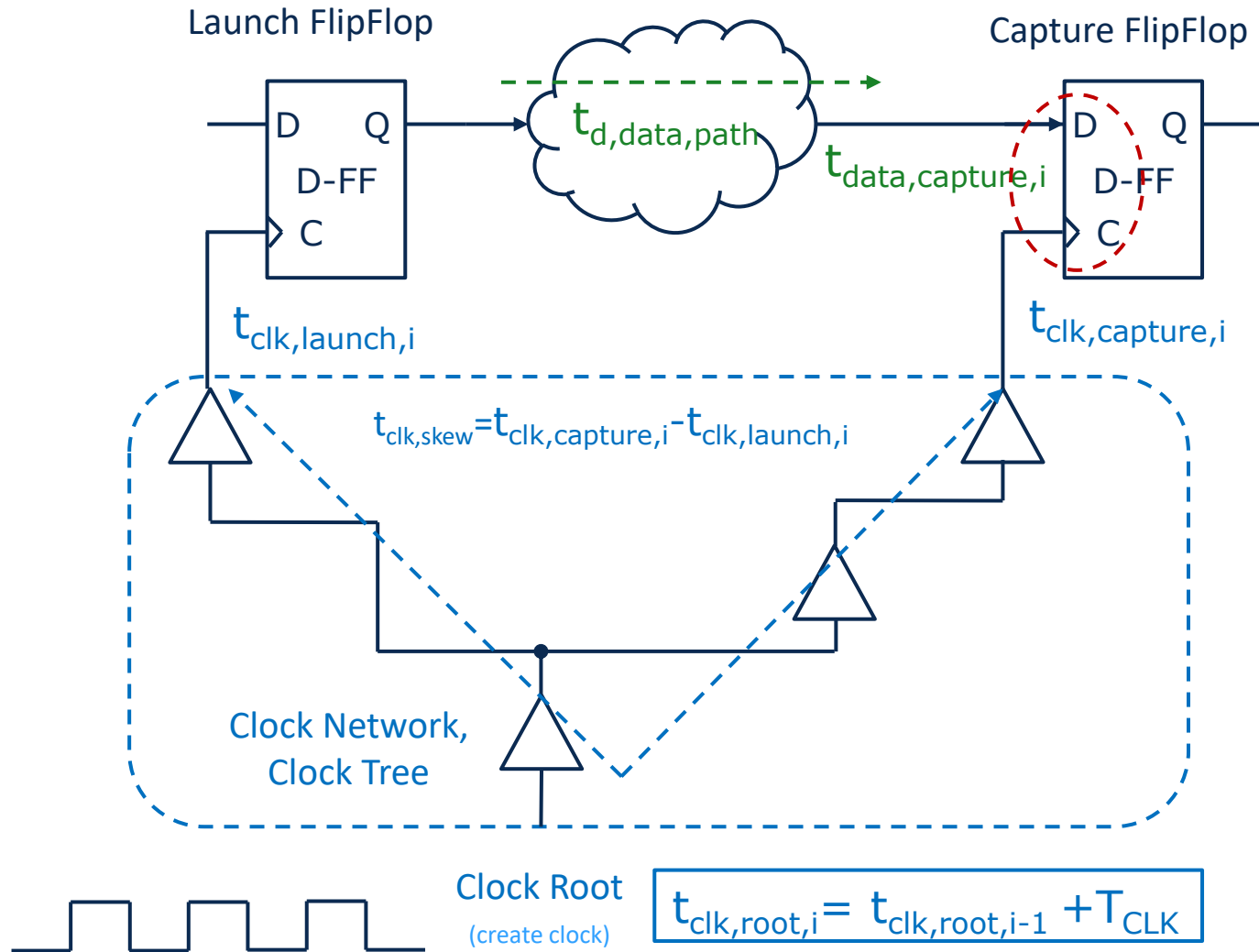
$$t_{clk,i} - t_{data,i-1} > t_{setup}$$

Übernahme von  $D_{i-1}$  mit Taktflanke  $i$

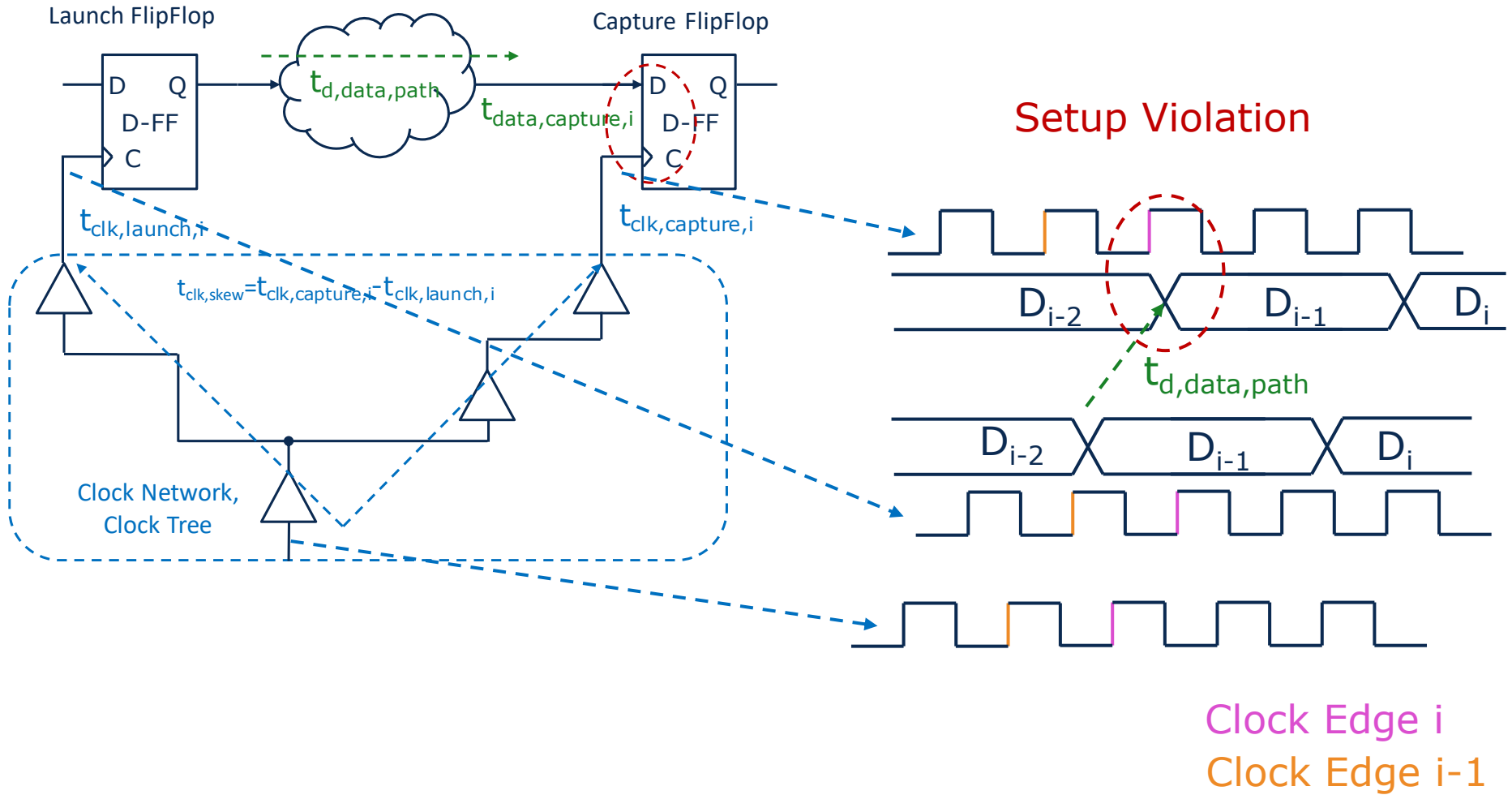


- Hold Constraint:

$$t_{data,i} - t_{clk,i} > t_{hold}$$

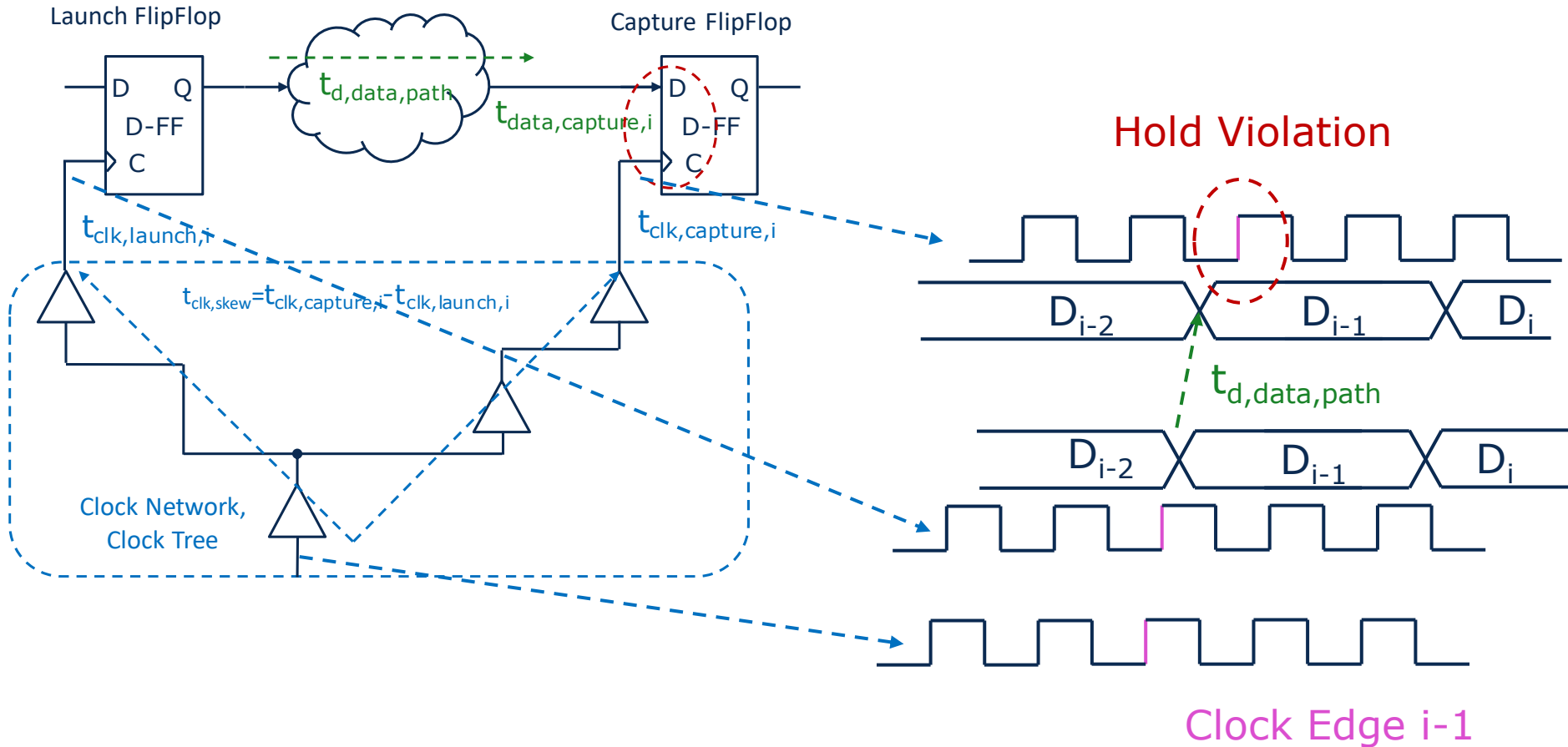


- Setup Constraint:
  - $t_{\text{clk,capture},i} - t_{\text{data,capture},i-1} > t_{\text{setup}}$
  - $t_{\text{clk,capture},i} - (t_{\text{clk,launch},i-1} + t_{\text{d,data,path}}) > t_{\text{setup}}$
  - $t_{\text{clk,skew}} + T_{\text{CLK}} - t_{\text{d,data,path}} > t_{\text{setup}}$
- Die Setup Bedingung ist an **2 Taktflanken (i-1, i)** geknüpft
- → Setup Verletzungen limitieren die maximale Taktfrequenz  $1/T_{\text{CLK}}$
- Setup ist kritisch bei:
  - Langsamem Datenpfad ( $t_{\text{d,data,path}}$  groß)
  - Kleinem Clock Skew ( $t_{\text{clk,skew}} = t_{\text{clk,capture},i} - t_{\text{clk,launch},i}$  klein)

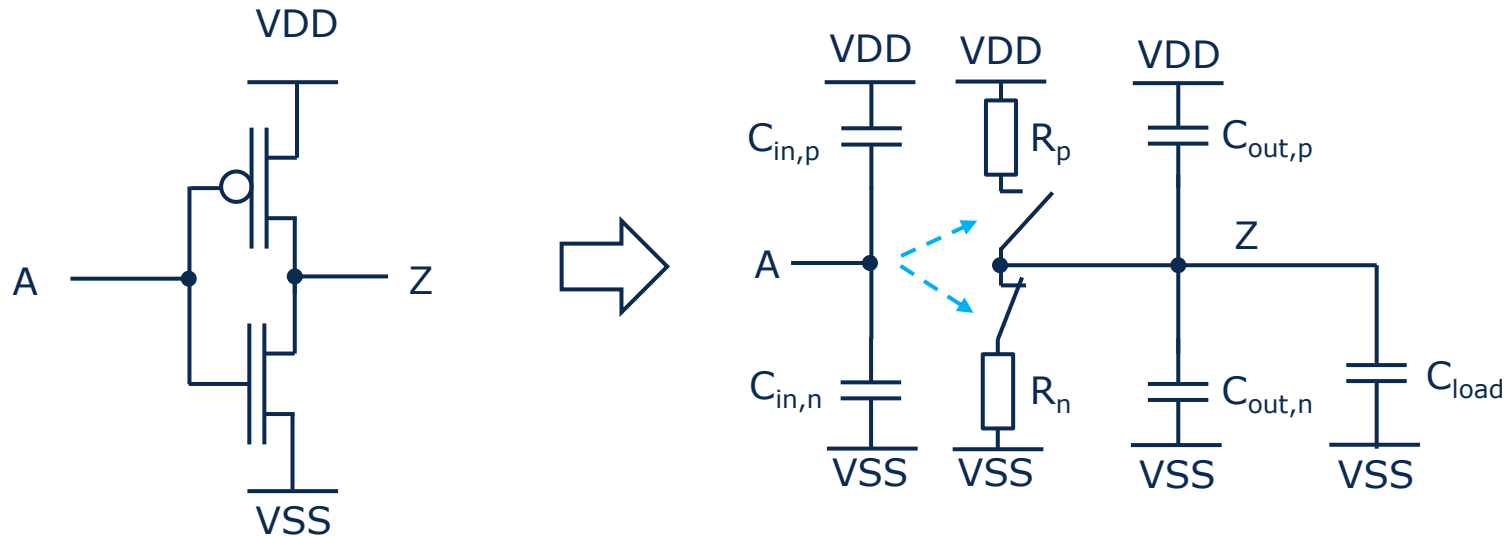




- Hold Constraint:
  - $t_{data,capture,i} - t_{clk,capture,i} > t_{hold}$
  - $(t_{clk,launch,i} + t_{d,data,path}) - t_{clk,capture,i} > t_{hold}$
  - $t_{d,data,path} - t_{clk,skew} > t_{hold}$
- Die Hold Bedingung ist an **1 Taktflanke (i)** geknüpft
- Hold Verletzungen sind **unabhängig von der Taktfrequenz**  $1/T_{CLK}$
- Hold ist kritisch bei:
  - Schneller Datenpfad ( $t_{d,data,path}$  klein)
  - Großer Clock Skew ( $t_{clk,skew} = t_{clk,capture,i} - t_{clk,launch,i}$  klein)

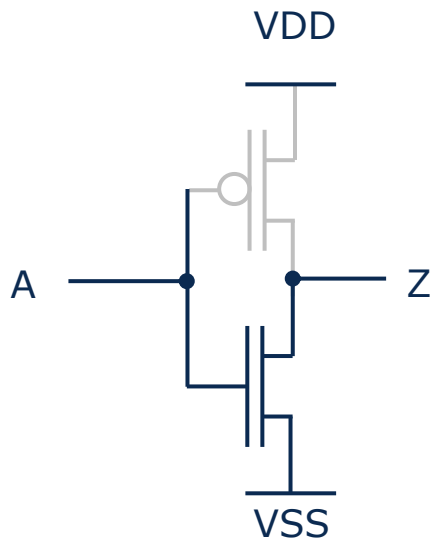


- RC Schaltungsmodell zur Berechnung der Verzögerungszeit

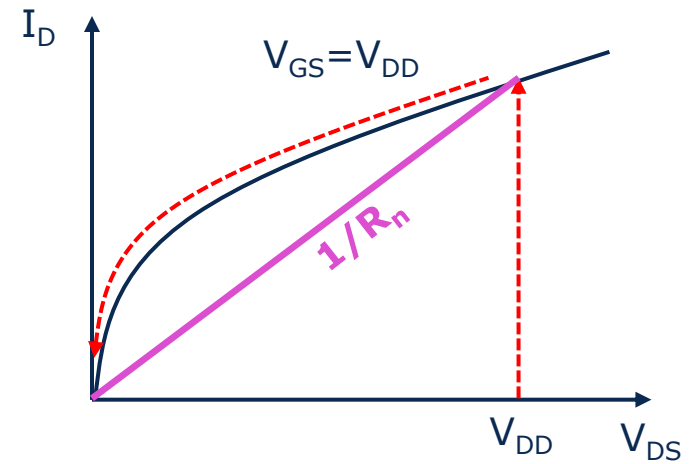


- Effektiver Schalterwiderstand von NMOS und PMOS
- Parasitäre Kapazitäten  $C_{in}$ ,  $C_{out}$ ,  $C_{load}$
- Annahme: Umschalten der Eingänge **VOR** dem Umladen der Ausgänge → Vernachlässigung des Miller-Effekts

- Berechnung des mittlerer Schalterwiderstandes
- Annahme: sofortiges Umschalten von A



Beispiel: NMOS Pull-Down Z: 1→0

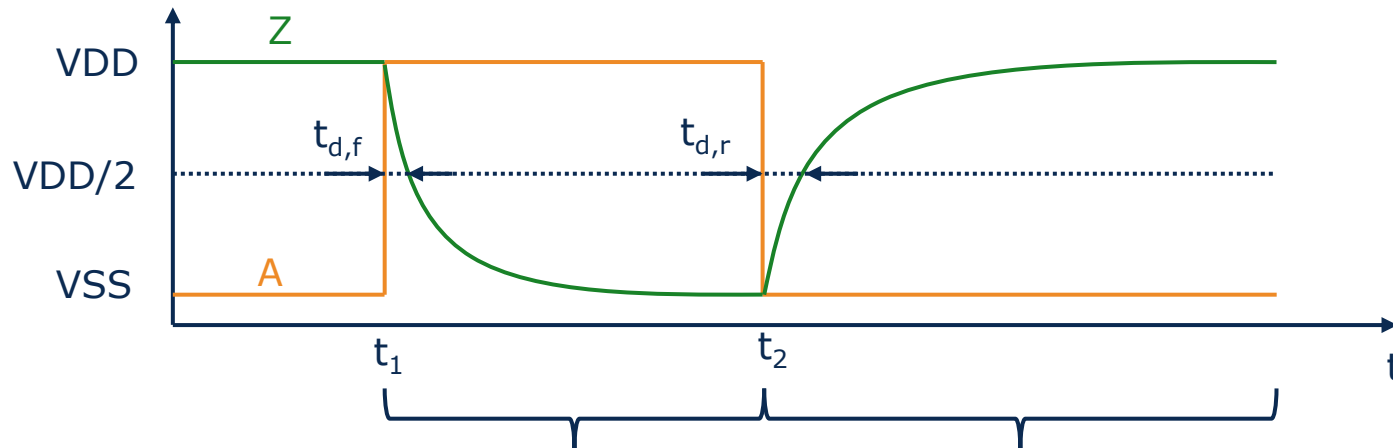


- Effektiver Schalterwiderstand

$$- R_n = \frac{V_{DD}}{\frac{K P_N \cdot W}{2 \cdot L} \cdot (V_{DD} - V_{th,N})^2 \cdot (1 + \lambda_N (V_{th,N}))} = R'_n \cdot \frac{L}{W}$$

- Analoge Betrachtung für PMOS bei Z: 0→1

- Delay Berechnung: RC-System erster Ordnung



$$V_Z(t) = V_{DD} \cdot e^{-\frac{t-t_1}{R_n(C_{out}+C_{load})}}$$

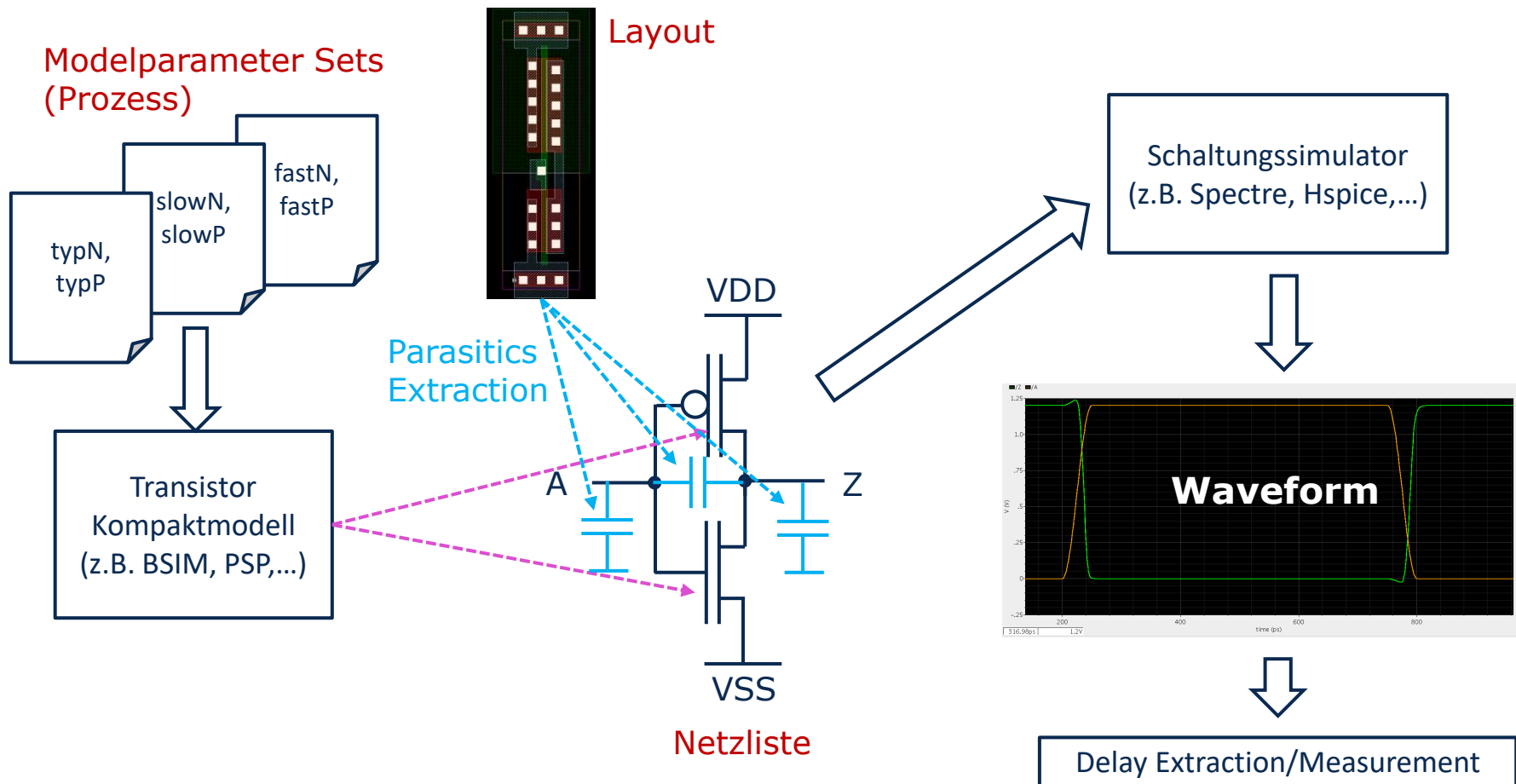
$$V_Z(t) = V_{DD} \cdot \left(1 - e^{-\frac{t-t_2}{R_p(C_{out}+C_{load})}}\right)$$

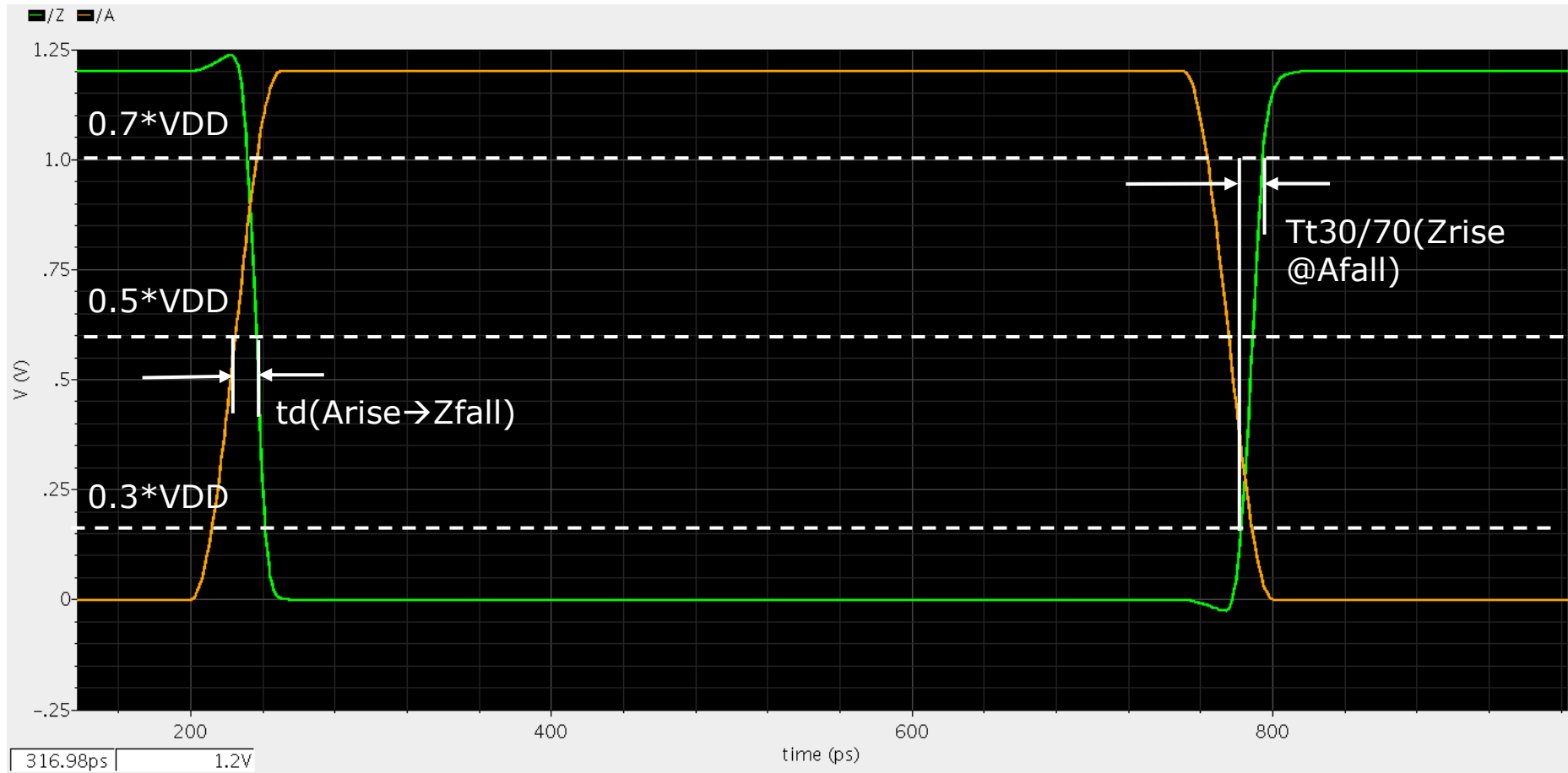
$$t_{d,f} = \ln(2) \cdot R_n(C_{out} + C_{load})$$

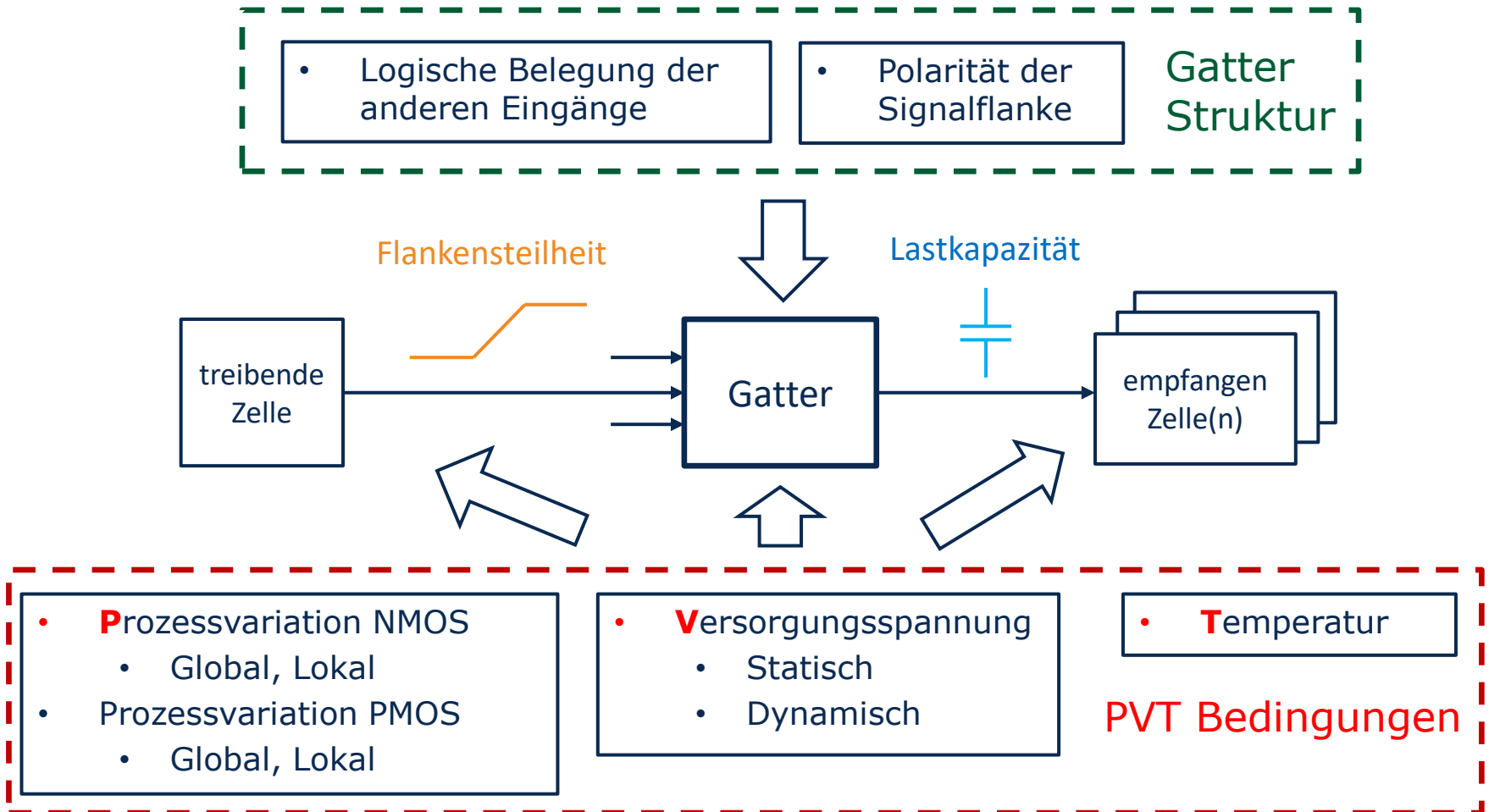
$$t_{d,r} = \ln(2) \cdot R_p(C_{out} + C_{load})$$

→ Details siehe Vorlesung  
digitale Schaltungstechnik

- Bestimmung des Timings durch Schaltungssimulation
- Nutzung von Kompaktmodellen für die Transistoren (z.B. BSIM, PSP)
- Extraktion von parasitären Layout-Kapazitäten und -Widerständen

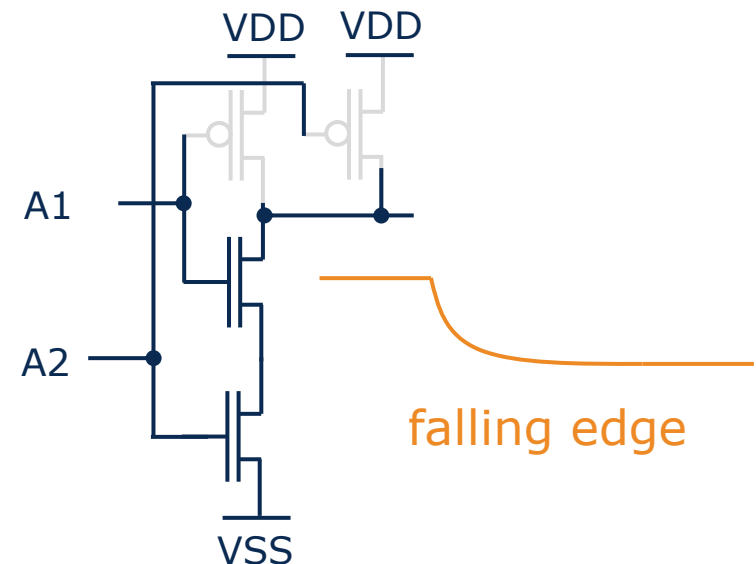
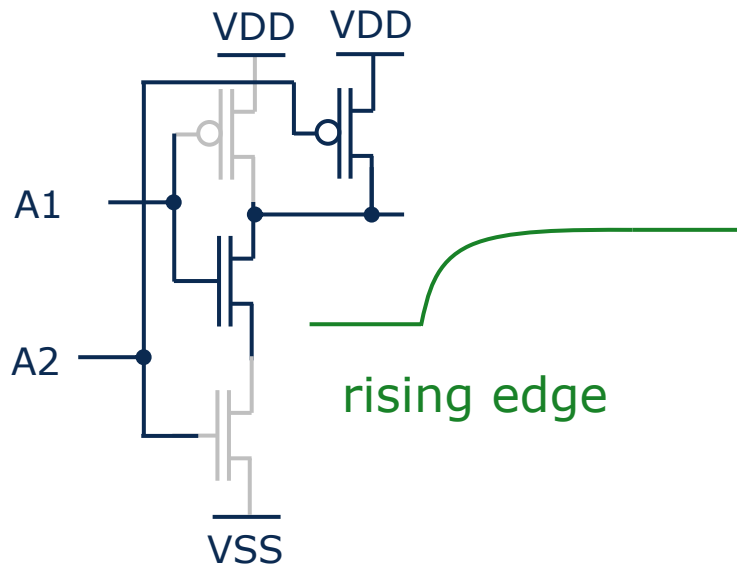






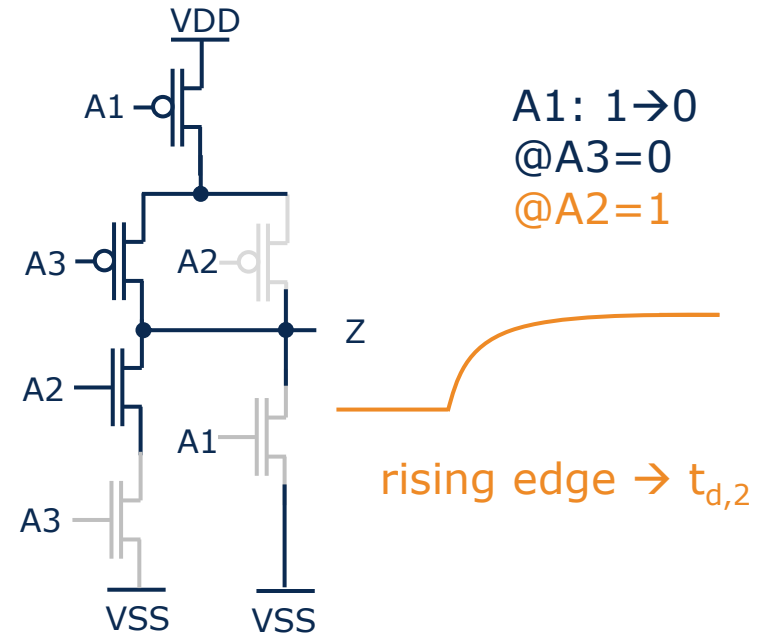
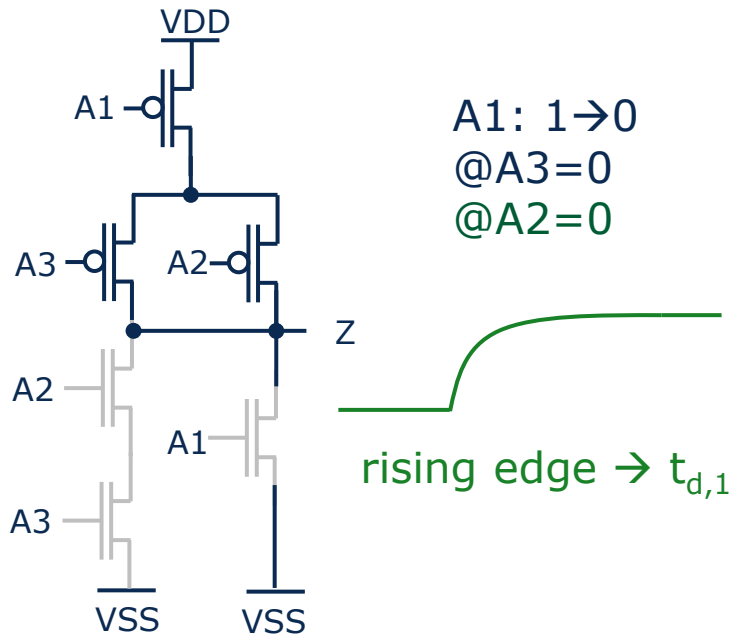


- In CMOS Logik sind am Schalten von steigender und fallender Taktflanke unterschiedliche Bauelemente in unterschiedlichen Verschaltungen beteiligt (Pull-up- und Pull-down-Netzwerk)



- Berücksichtigung im Design Flow:**
  - Modellierung individueller Delays in pro Signalflanke

- Unterschiedliche Pull-Up oder Pull-Down Treiberstärken durch andere Inputs
- Beispiel: AOI12:



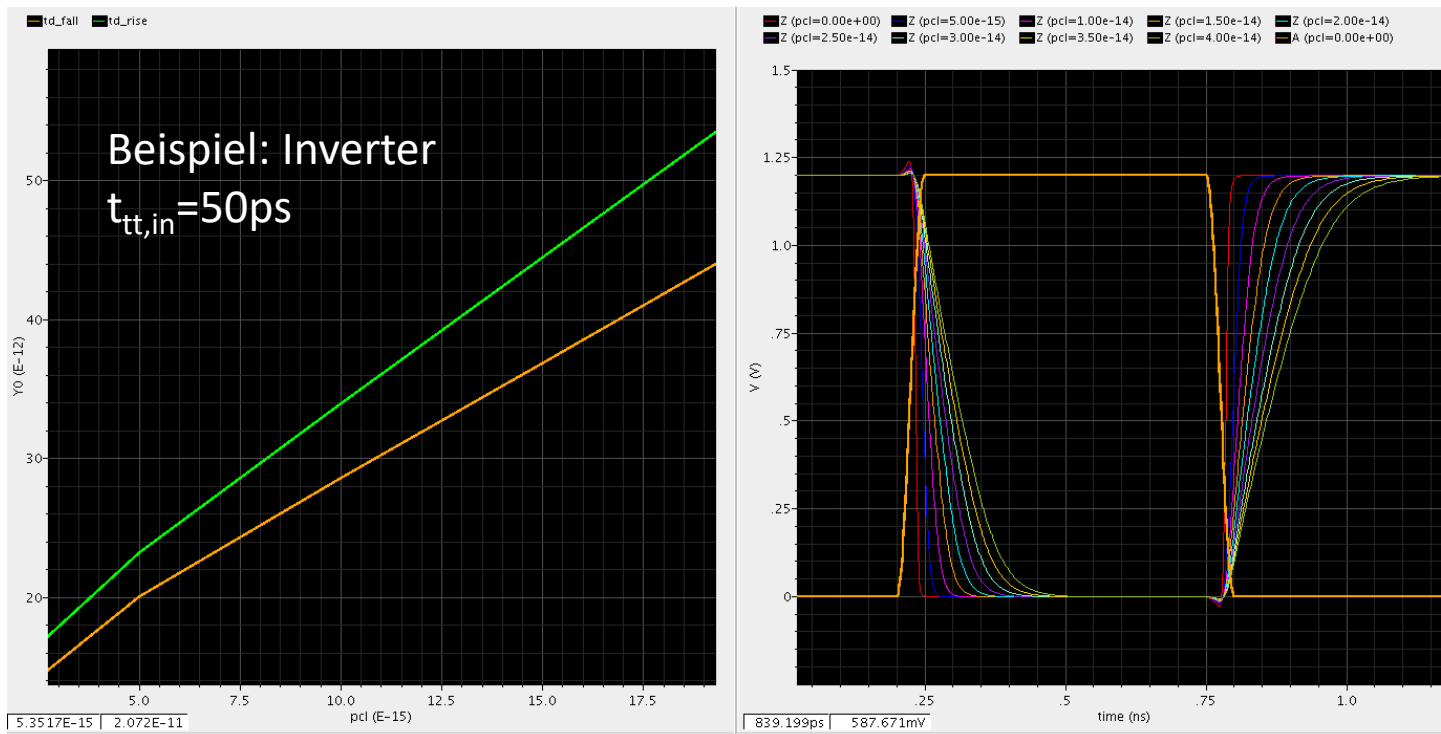
- **Berücksichtigung im Design Flow:**
  - Modellierung des Delays von Timing Arcs in Abhängigkeit der Logikpegel anderer Eingänge

- Direkte Abhängigkeit des Delays von der Lastkapazität

$$C_{load} \uparrow \rightarrow t_d \uparrow$$

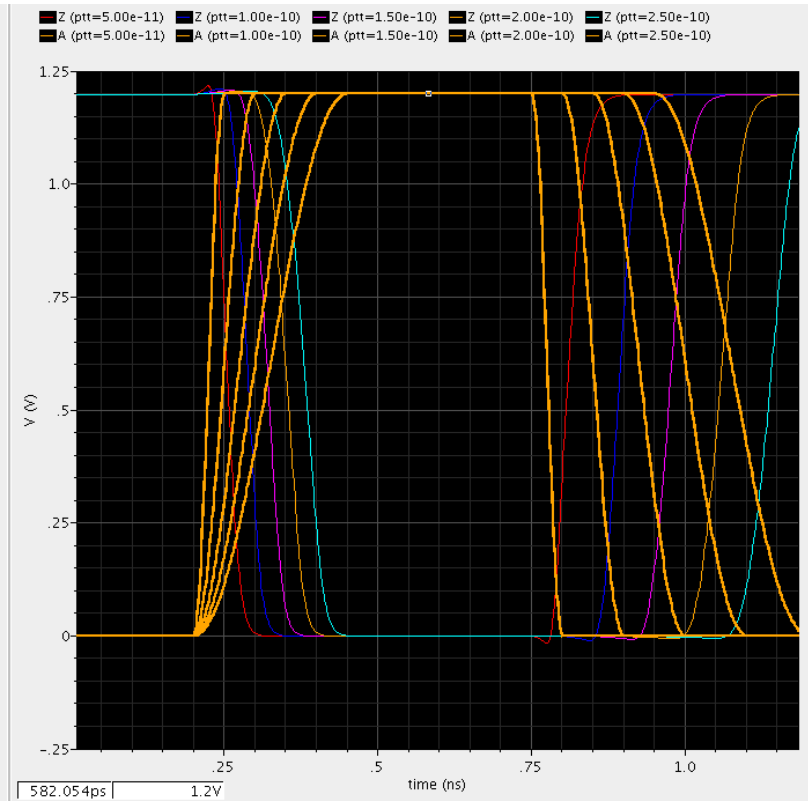
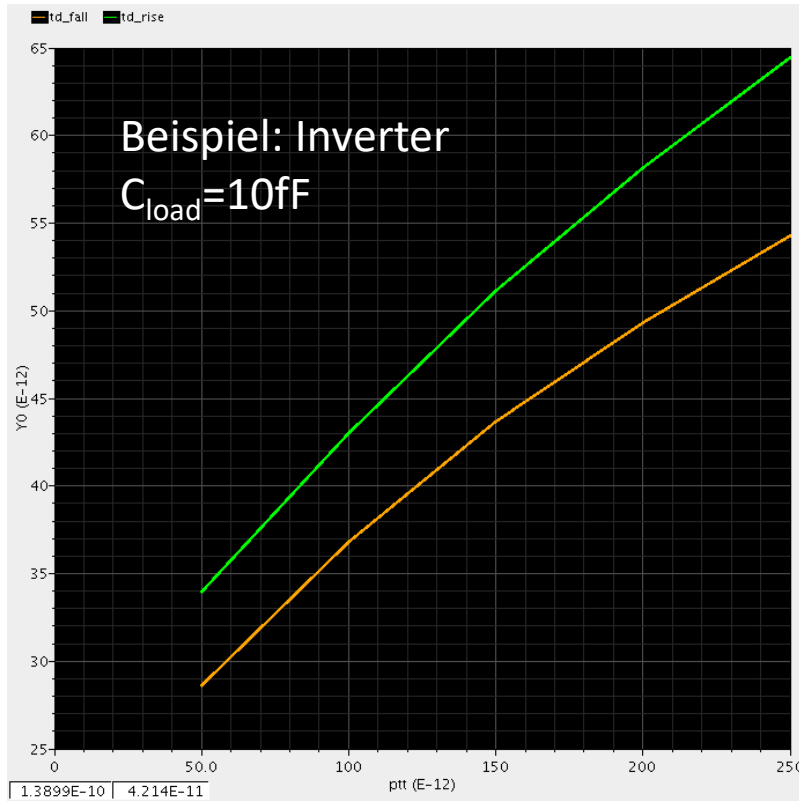
$$t_{d,f} = \ln(2) \cdot \frac{V_{DD}}{\frac{KP_{N/P}}{2} \cdot \frac{W}{L} \cdot (V_{DD} - V_{th,N})^2 \cdot (1 + \lambda_N(V_{th,N}))} \cdot (C_{out} + C_{load})$$

(analog für  $t_{d,r}$ )



- Berücksichtigung im Design Flow:**
  - Modellierung des Delays  $t_d = f(C_{load})$  und Output Transition Time  $t_{tt,out} = g(C_{load})$ , z.B. als Lookup-Tabelle

- Direkte Abhängigkeit des Delays von der Flankensteilheit (Transition Zeit  $t_{tt,in}$ ) der Eingangsflanke



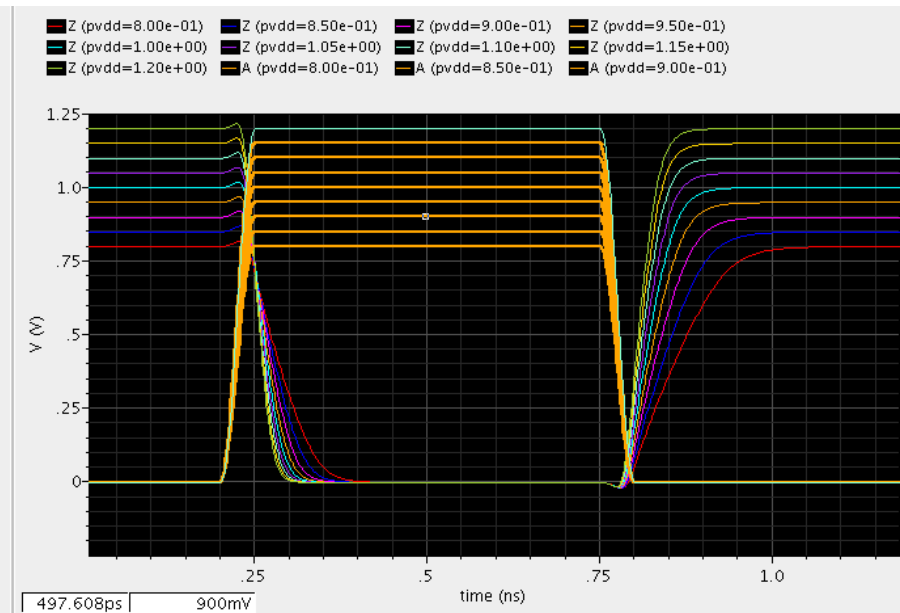
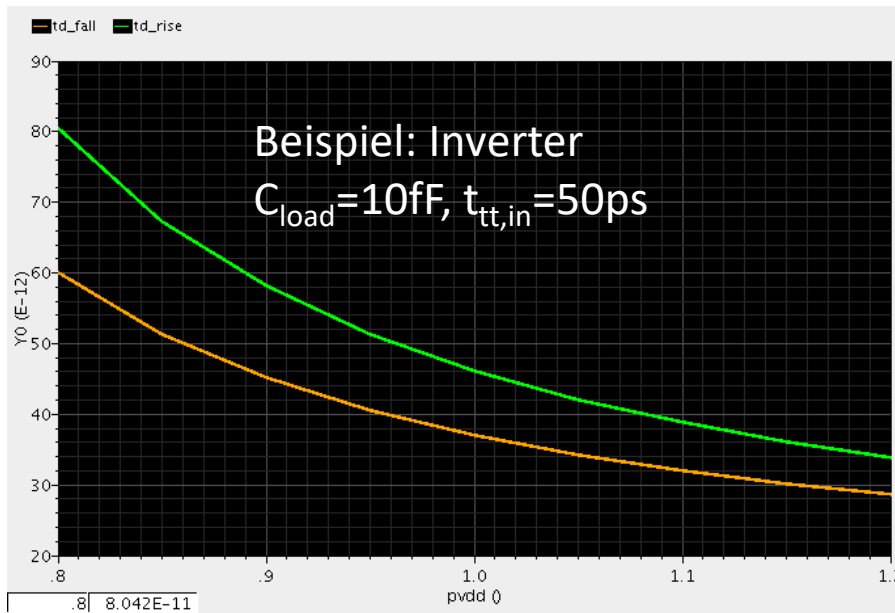
- **Berücksichtigung im Design Flow:**
  - Modellierung des Delays  $t_d = f(t_{tt,in})$  und  $t_{tt,out} = g(t_{tt,in})$ , z.B. als Lookup-Tabelle

- Abhängigkeit des Delays von der statischen VDD

$$V_{DD} \downarrow \rightarrow t_d \uparrow$$

$$t_{d,f} = \ln(2) \cdot \frac{K_{P_{N/P}} \cdot \frac{W}{L} \cdot V_{DD}}{2 \cdot (V_{DD} - V_{th,N})^2 \cdot (1 + \lambda_N(V_{th,N}))} \cdot (C_{out} + C_{load})$$

(analog für  $t_{d,r}$ )



- Berücksichtigung im Design Flow:**
  - Bestimmung des Timings durch Schaltungssimulation im Bereich der Versorgungsspannungen des Zielsystems (z.B. 1.2V  $\pm$  10%)  $\rightarrow$  **PVT Corner**

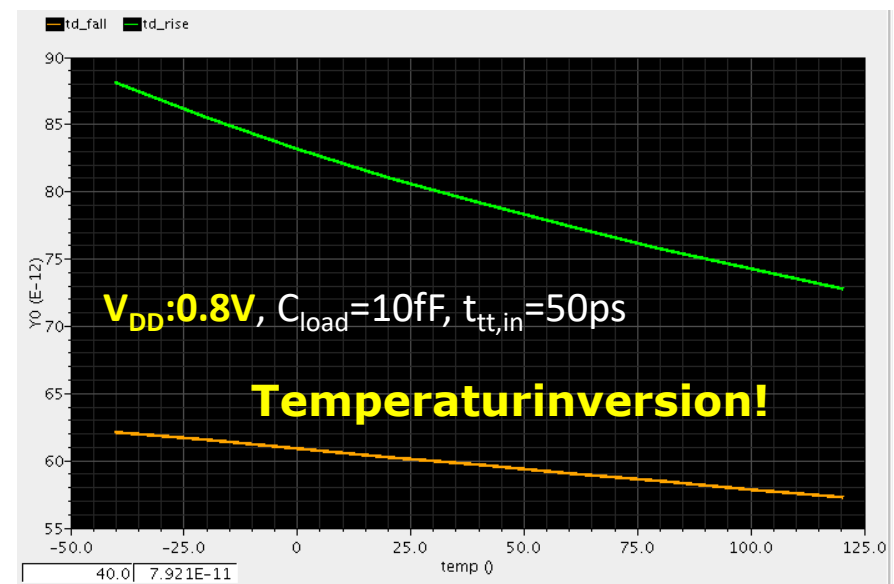
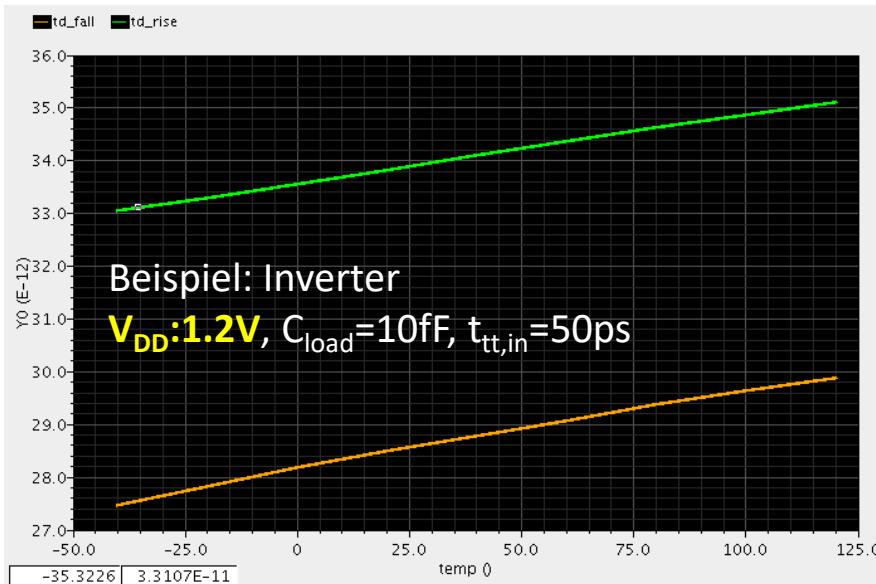
- Temperatur beeinflusst maßgeblich die Schwellspannung  $V_{th}$  und die Leitfähigkeit des Kanals ( $K_P$ ,  $\lambda$ )

$$T \uparrow \rightarrow V_{th} \downarrow$$

$$T \uparrow \rightarrow K_P \downarrow$$

$$t_{d,f} = \ln(2) \cdot \frac{K_{P,N/P}}{2} \cdot \frac{W}{L} \cdot \frac{V_{DD}}{(V_{DD} - V_{th,N})^2} \cdot (1 + \lambda_N(V_{th,N})) \cdot (C_{out} + C_{load})$$

(analog für  $t_{d,r}$ )

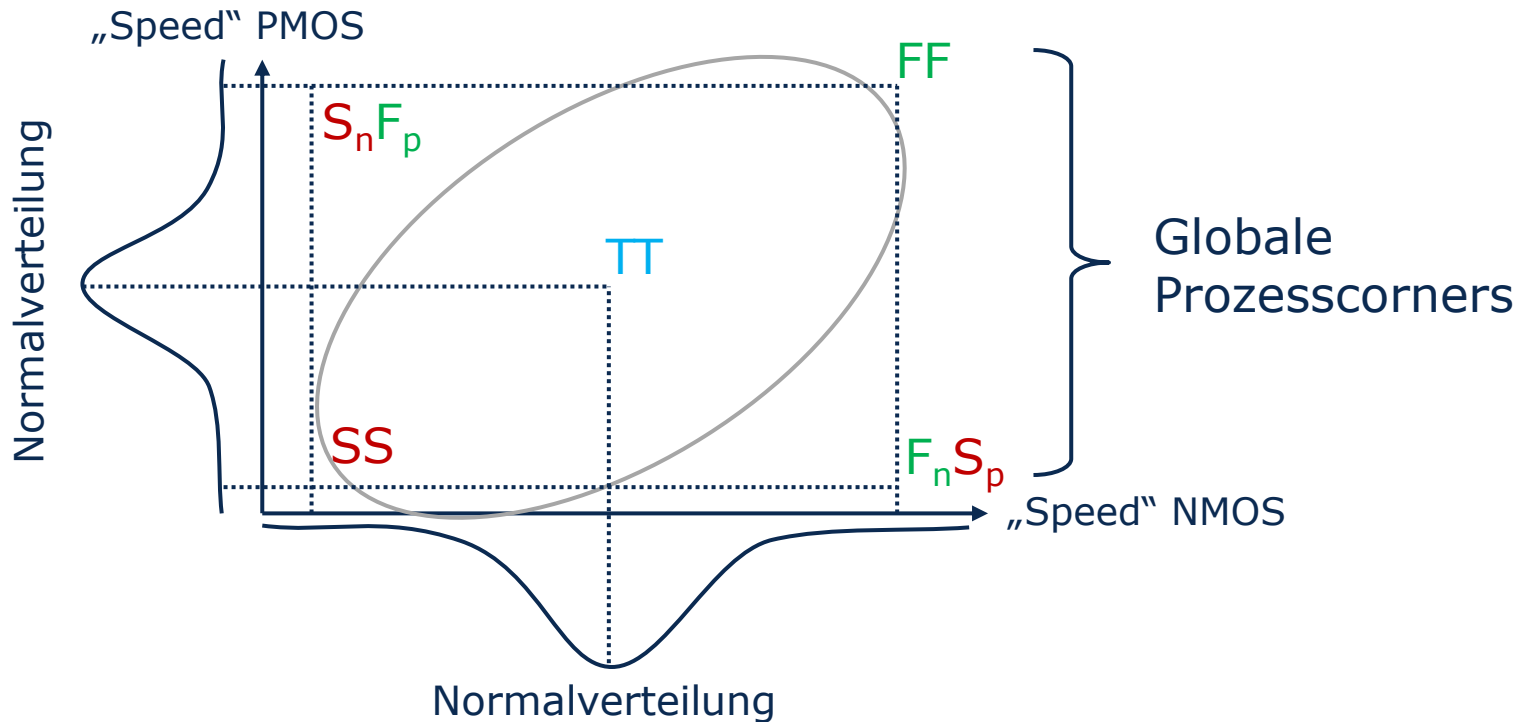


## Berücksichtigung im Design Flow:

- Bestimmung des Timings durch Schaltungssimulation im Temperaturbereich des Zielsystems (z.B.  $-40^{\circ}C$  bis  $125^{\circ}C$ ) → **PVT Corner**

- Der CMOS Fertigungsprozess unterliegt **zufälligen Schwankungen**
- → Die Eigenschaften von NMOS und PMOS Transistoren unterliegen Schwankungen (z.B.  $W$ ,  $L$ ,  $V_{th}$ ,  $\mu$ ,  $C_{ox}$ )
- NMOS und PMOS Transistoren werden in **unterschiedlichen** Herstellungsschritten gefertigt → Schwankungen sind unabhängig voneinander bzw. nur schwach korreliert.
  
- **Globale** Variationen:
  - **Alle Transistoren eines Typs** auf dem Wafer/Chip gleichermaßen beeinflusst
- **Lokale** Variationen:
  - Un- bzw. schwach korrelierter Mismatch **zwischen Transistoren auf einem Wafer/Chip**

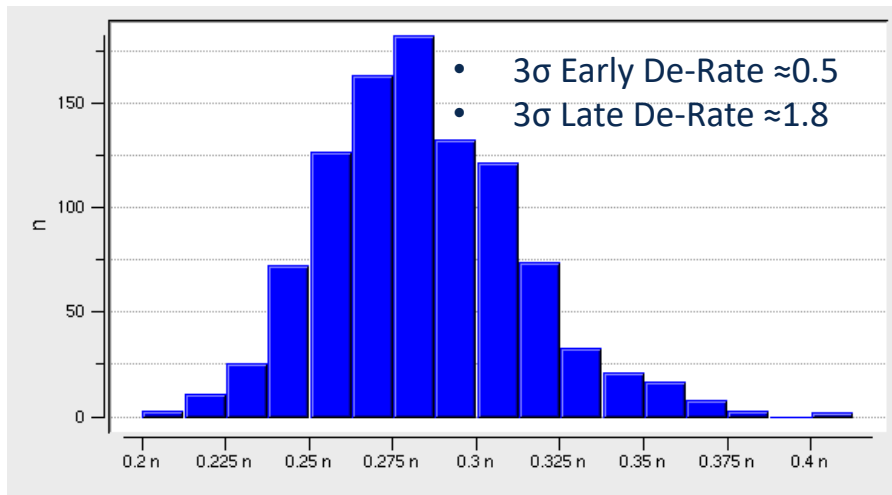
- Unabhängige, globale Variation der NMOS und PMOS Parameter
- Definition von Prozess Corners an den  $\pm 3\sigma$  Grenzen der Parameterverteilungen



- **Berücksichtigung im Design Flow:**
  - Bestimmung des Timings durch Schaltungssimulation in den globalen Corners des Prozesses (z.B. TT, SS, FF) → **PVT Corner**



- Bestimmung der lokalen Delay Variabilität Monte-Carlo Simulation
  - Extraktion der Delay Statistiken der Standardzelle (können unsymmetrisch sein!)
- Lokale Variabilität ist kritisch bei:
  - Modernen Technologien mit kleinen Strukturgrößen
  - Kleinen Versorgungsspannungen



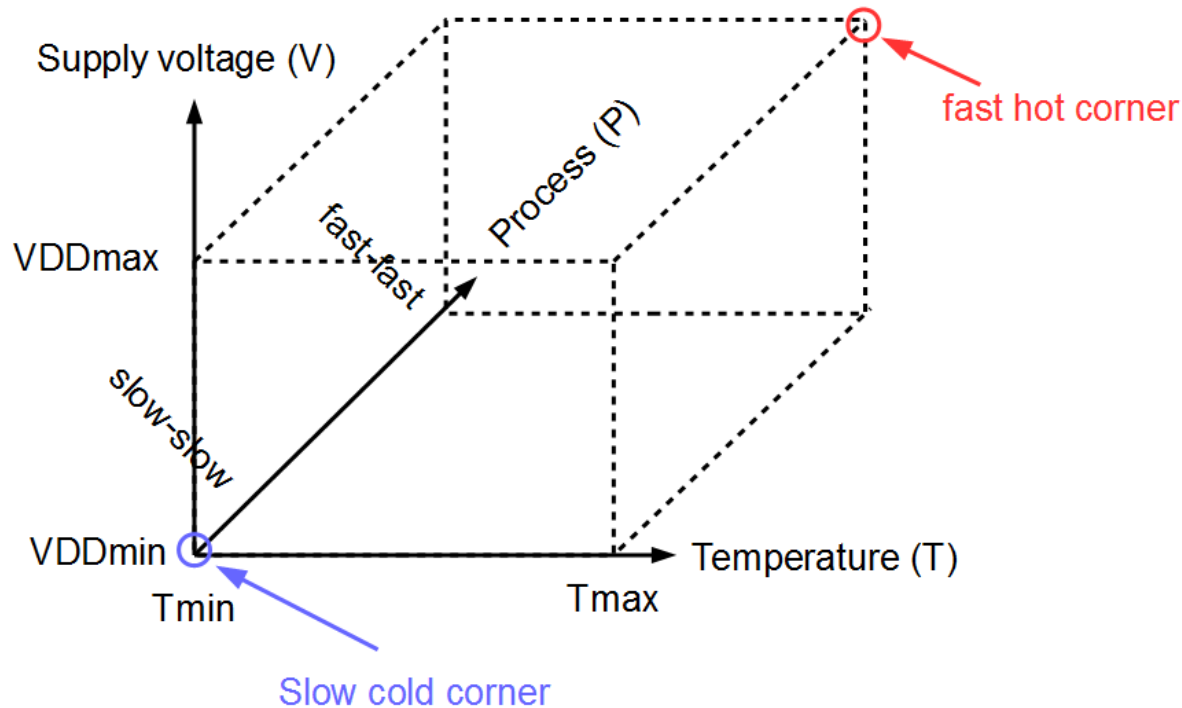
- Beispiel: Clock Buffer bei **sehr kleiner**  $V_{DD}$
- → Unterschiedliche Instanzen der **gleichen** Standardzelle können auf einem Chip Delays im Bereich  $0.5 \cdot t_d \dots 1.8 \cdot t_d$  aufweisen

- **Berücksichtigung im Design Flow:**
  - Extraktion der Delay Variabilität (Standardabweichungen für Early/Late Timings), Design Views: AOCV Tabellen, LVF .lib Files
  - De-Rating in der Timing Analyse

- Ein CMOS Gatter hat **einen weiten Bereich** an Delay Zeiten
- ↓
- Die **korrekte** Timing Analyse ist notwendig
- ↓
- Wie wird dies bei der Implementierung berücksichtigt?

- Definition von Analysepunkte an den Ecken (Corner) des Parameterbereichs
- **PVT Corner:**
  - **Prozess** (NMOS,PMOS), Backend (extraction)
    - z.B. tc (TT, typical backend RC), wc (SS, worst RC), bc (FF, best RC)
  - **Versorgungsspannung**
    - Spannungsbereich laut Chip Spezifikation
    - z.B.  $1.0V \pm 10\% \rightarrow (0.9V; 1.0V; 1.1V)$
  - **Temperatur**
    - Temperaturbereich laut Chip Spezifikation
    - z.B.  $-40^{\circ}C \dots 125^{\circ}C \rightarrow (-40^{\circ}C; 25^{\circ}C; 125^{\circ}C)$
- **Charakterisierung** der Standardzellen in den Corners
  - Simulation der Delays in Abhängigkeit von Struktur, Transition Zeit, Lastkapazität) [→ siehe Abschnitt Standardzellbibliotheken](#)

- Jede **Corner** stellt einen best/worst-case für eine **Performance** dar
  - **Performances:** z.B. Delay (Timing), Power
- Beispiel:
  - Worst-case Delay (Timing) in Temperaturinversion: **slow cold corner**
  - Worst-case Power: **fast hot corner**



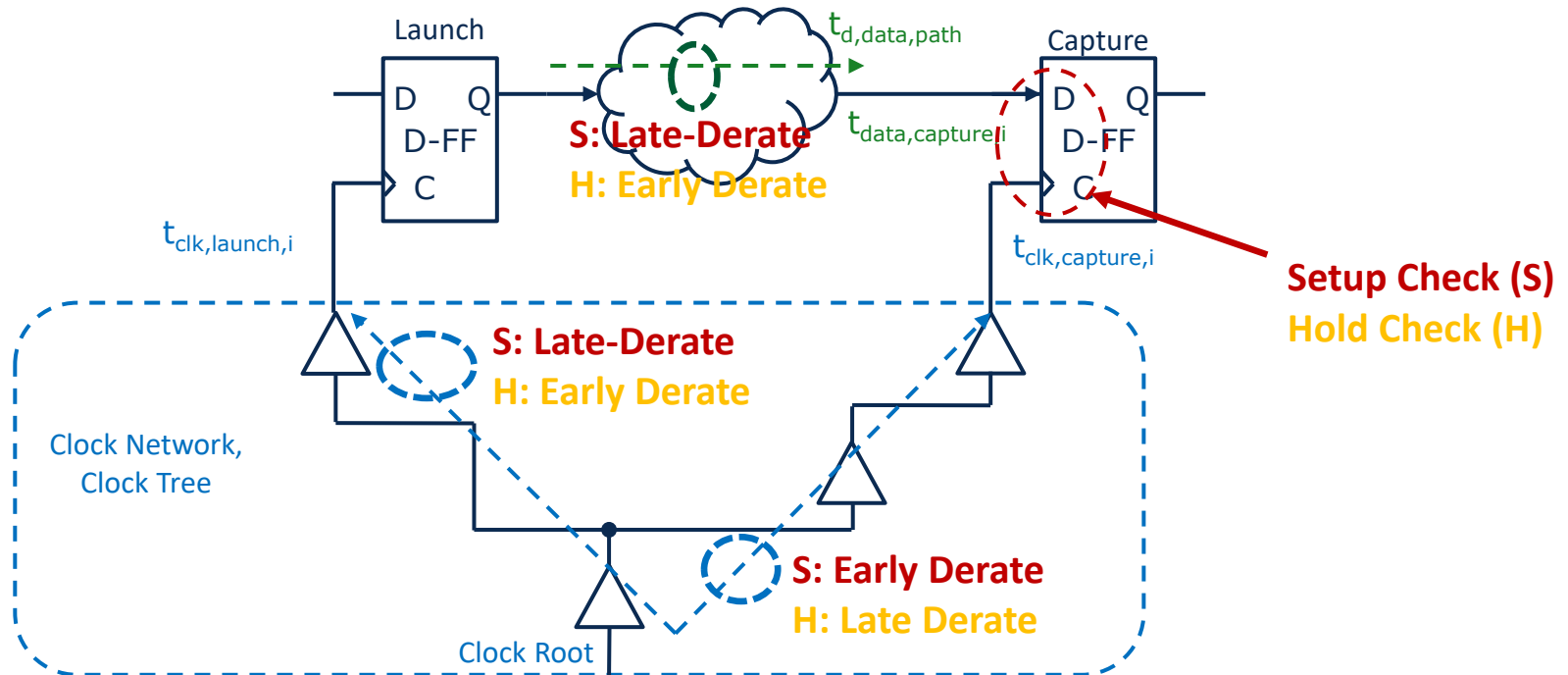
- 28nm CMOS Standardzellenbibliothek HPSNLIB
- Prozess und Backend Parasitics:
  - wc (SS und worst RC), tc (TT und typical RC), bc (FF und best RC)
- VDD Corners:
  - 1.0V  $\pm$  10%, 1.1V  $\pm$  5% 0.8V  $\pm$  5%
- Temperatur:
  - -40°C bis 125°C
- Standardzellen charakterisiert in **15 Corners**
- 3 typical, 6 worst, 6 best

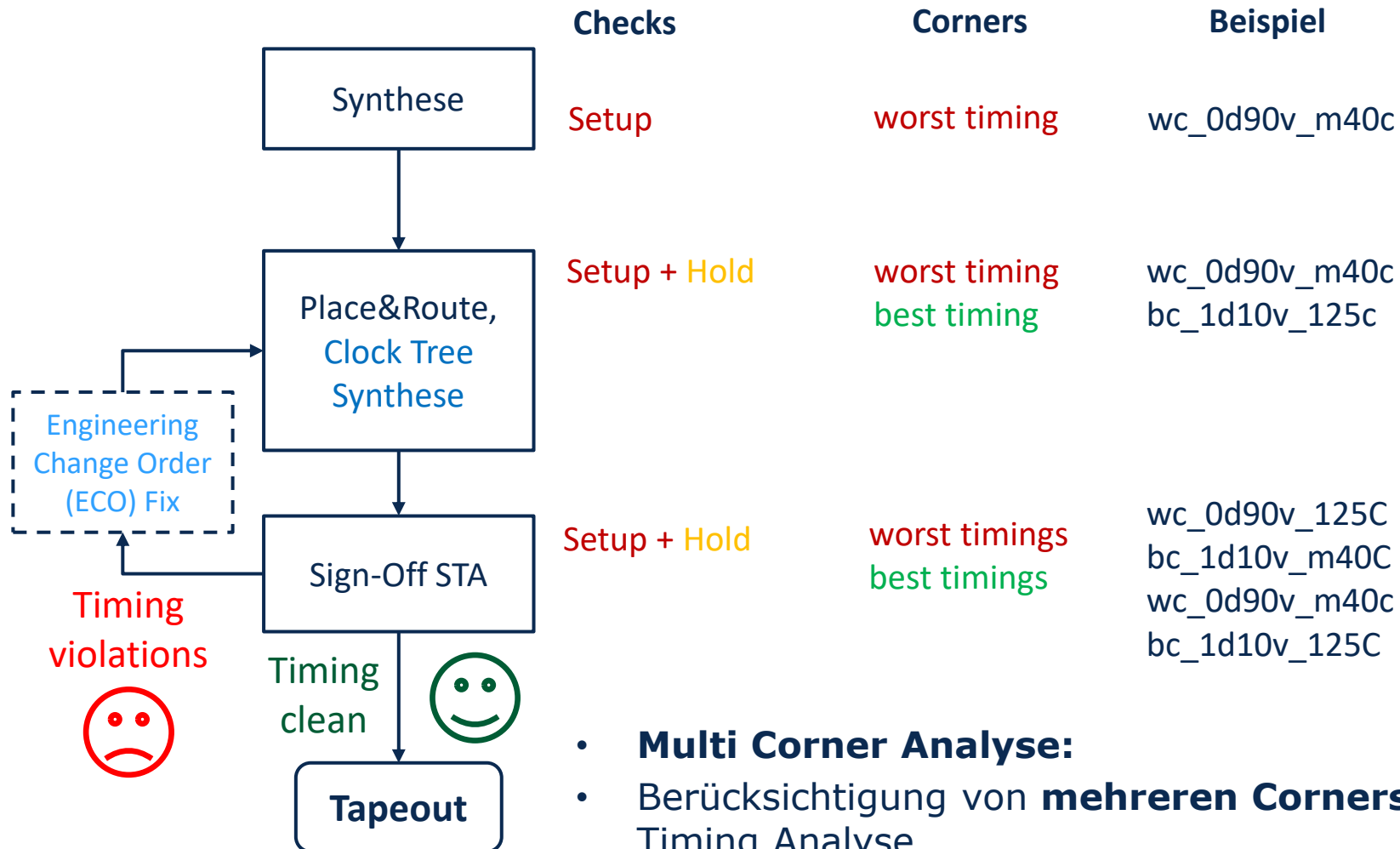
```

hpsnlib_g28_9t_RVT_bc_0d84V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_0d84V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d10V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d10V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d155V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d155V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_tc_0d80V_25C.nldm.lib
hpsnlib_g28_9t_RVT_tc_1d00V_25C.nldm.lib
hpsnlib_g28_9t_RVT_tc_1d10V_25C.nldm.lib
hpsnlib_g28_9t_RVT_wc_0d76V_125C.nldm.lib
hpsnlib_g28_9t_RVT_wc_0d76V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_wc_0d90V_125C.nldm.lib
hpsnlib_g28_9t_RVT_wc_0d90V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_wc_1d045V_125C.nldm.lib
hpsnlib_g28_9t_RVT_wc_1d045V_m40C.nldm.lib

```

- Berücksichtigung von lokaler Variabilität (On-Chip Variation (OCV)) durch Early/Late De-Rating
- Relative Variation  $d \cdot t_d$  mit  $d < 1$  für **Early** und  $d > 1$  für **Late**
- Separate Berücksichtigung für Setup und Hold Timing Checks

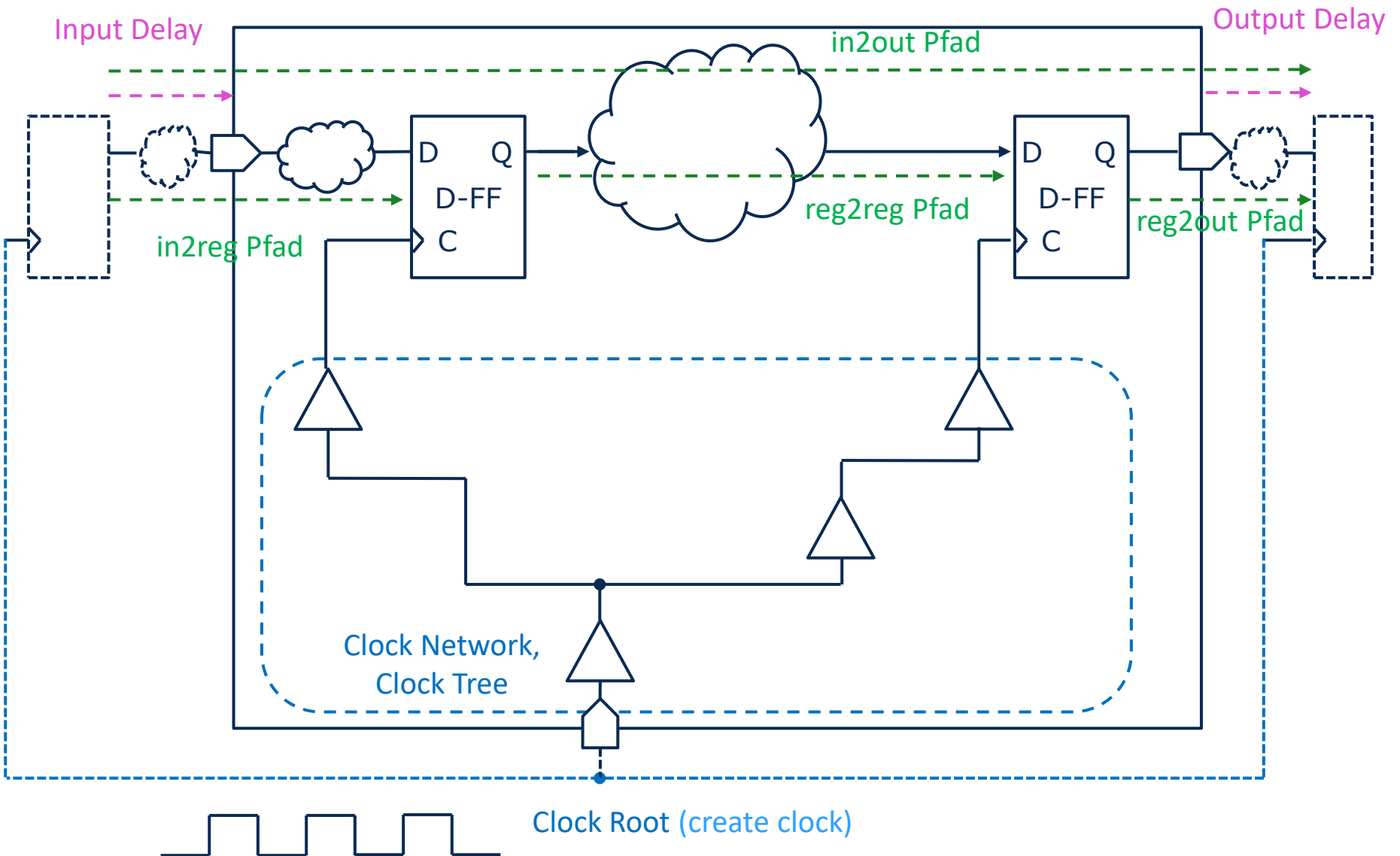




- **Multi Corner Analyse:**
- Berücksichtigung von **mehreren Corners** bei der Timing Analyse
- → Details siehe Vorlesung VLSI Prozessorentwurf

# Constraining und Timing-Signoff

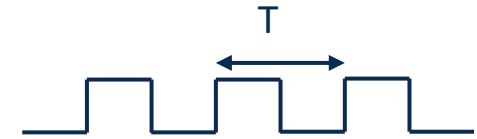




- Constraints stellen Randbedingungen für die Timing-Analyse dar

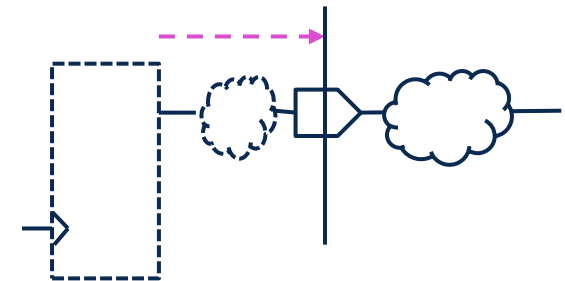
- **Taktperiode Clock Period**

- Zusätzlich können separat `clk_high` und `clk_low` Zeiten und Jitter berücksichtigt werden



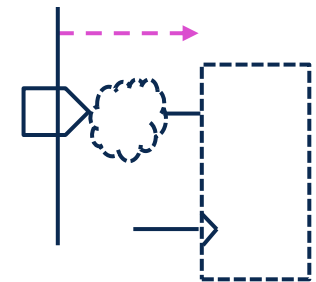
- **Input Delay**

- Verzögerungszeit VOR den INPUTs des constrainten Designs
- Started (launch) bei idealem Taktsignal



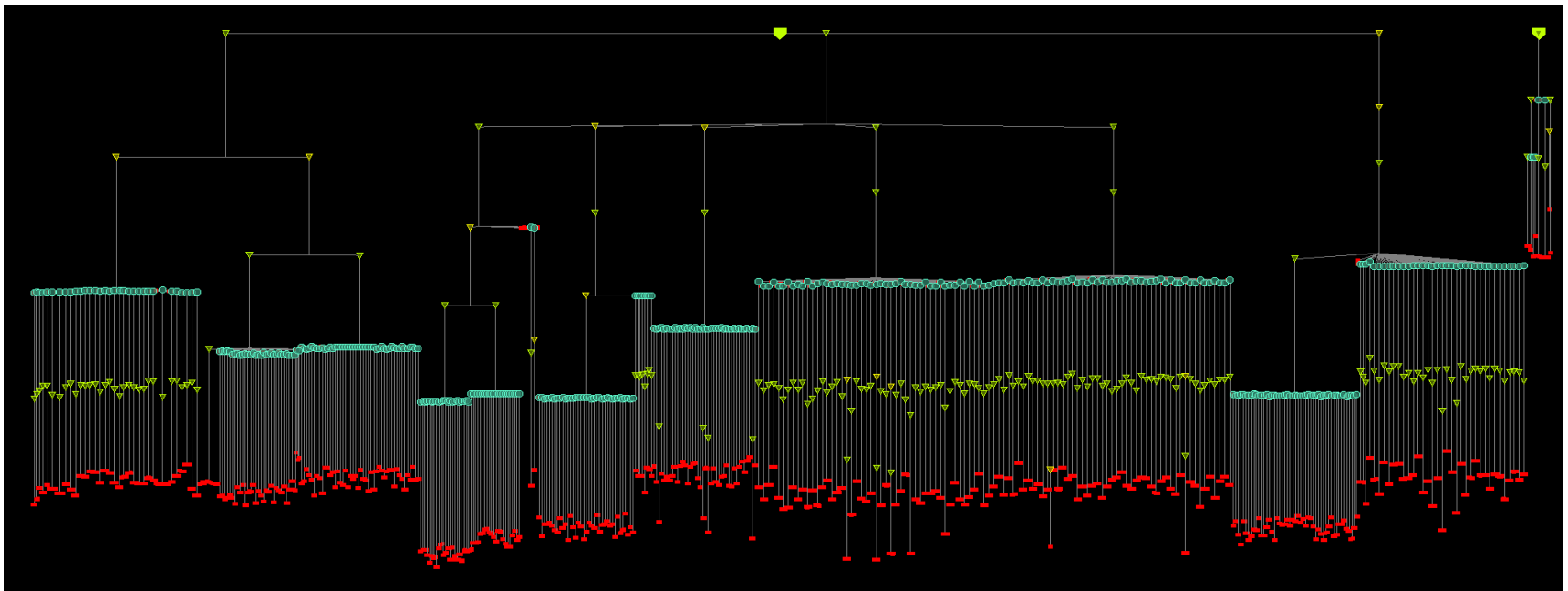
- **Output Delay**

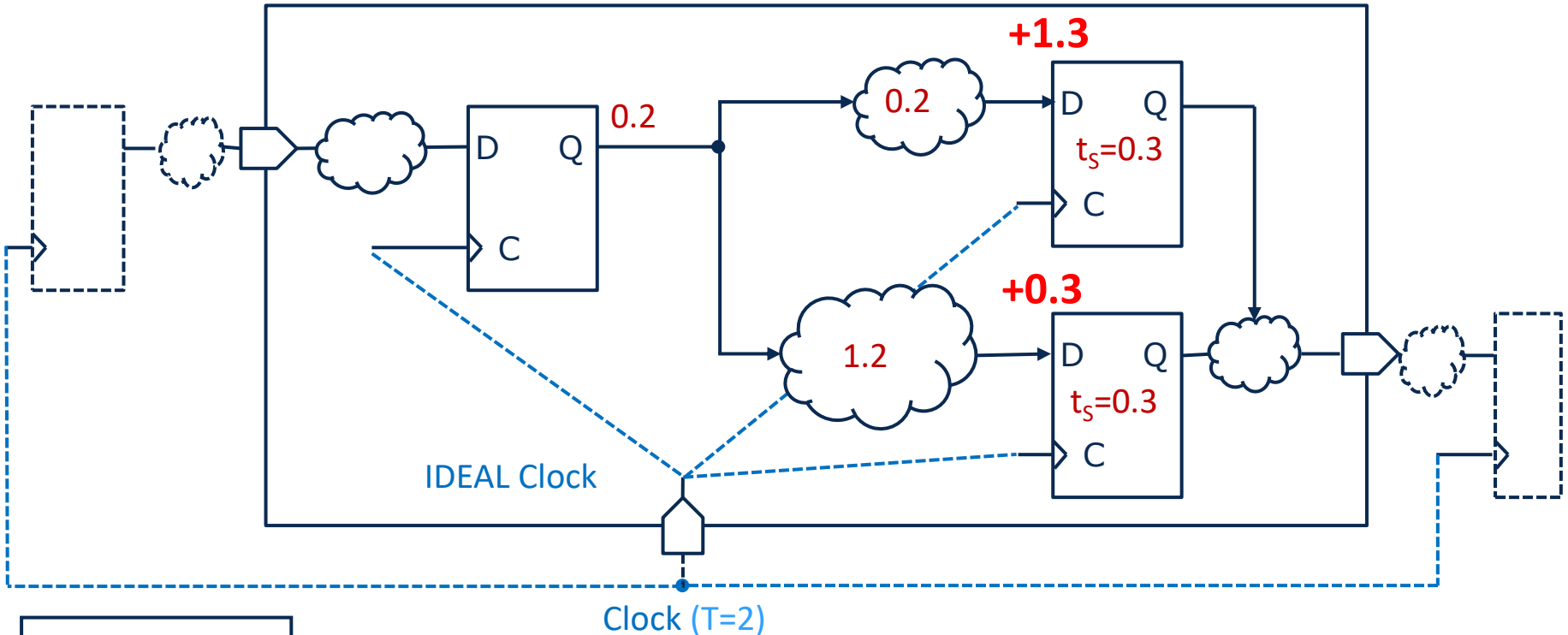
- Verzögerungszeit nach den Outputs des constrainten Designs
- Endpunkt (capture) bei idealem Taktsignal



→ Details siehe Praktikum und  
 Vorlesung VLSI-Prozessorwurf

- FlipFlops im Layout sind räumlich verteilt und benötigen verteilte Taktsignale
- Clock-Buffer Schaltungen müssen vom P&R Tool eingefügt werden
- Individuelle Endpunkte haben unterschiedliches Clock-Delay bezogen auf den **Startpunkt des Clock Trees** (idealer clock)
- → Balancierter Taktbaum
- Tiefe des Baums ( in ns) kann per Constraining angegeben werden. Wichtig z.B. bei Berücksichtigung von Input und Output Delays der Schaltung
- Beispiel Clock Tree Visualisierung:





Legende:

worst delay

best delay

Setup slack

Hold slack

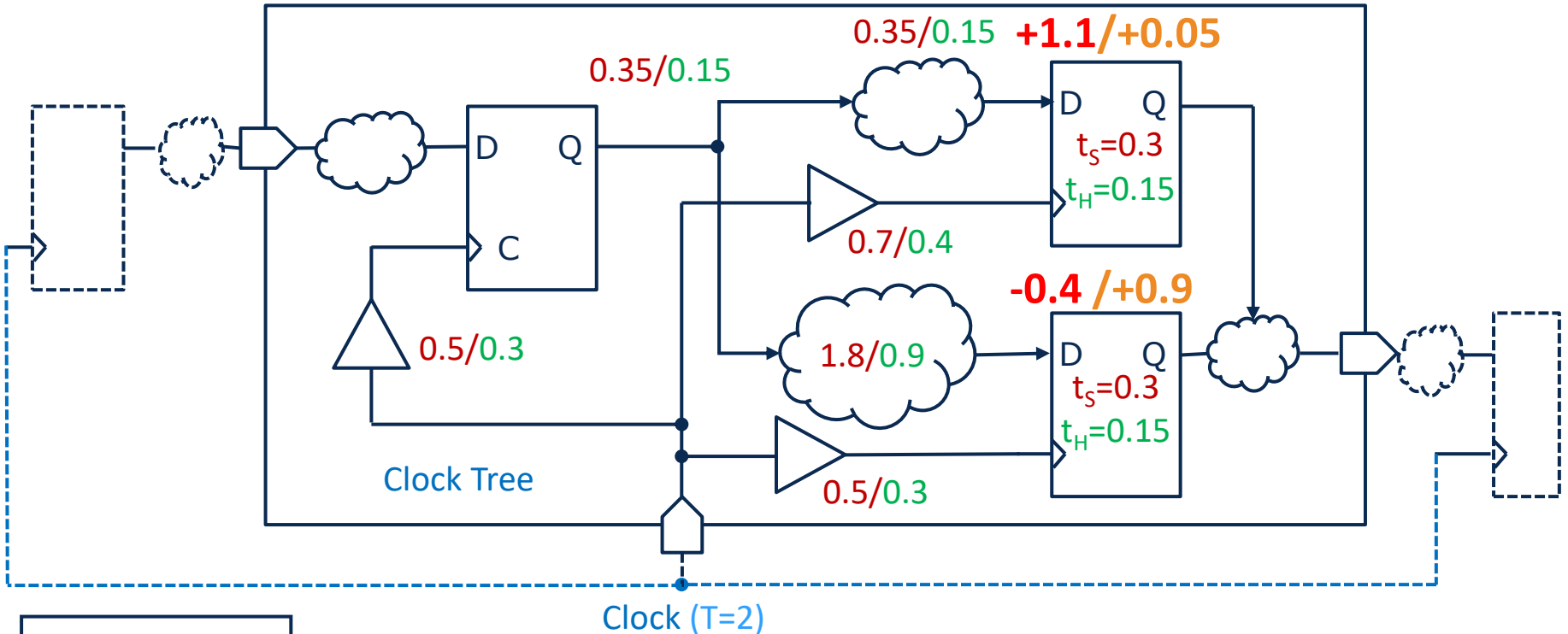
Zeiteinheit: [ns]

Setup Slack: Required Time - Arrival Time - FlipFlop Setup Time

Bsp FF1:  $2 - 0.2 - 1.2 - 0.3 = +0.3$

**Alle Setup Slacks positiv**  
**Synthese Netzliste Timing Clean!**





Legende:

worst delay

best delay

Setup slack

Hold slack

Zeiteinheit: [ns]

Größere Delays durch P&R Parasitic Kapazitäten der Leitungen)

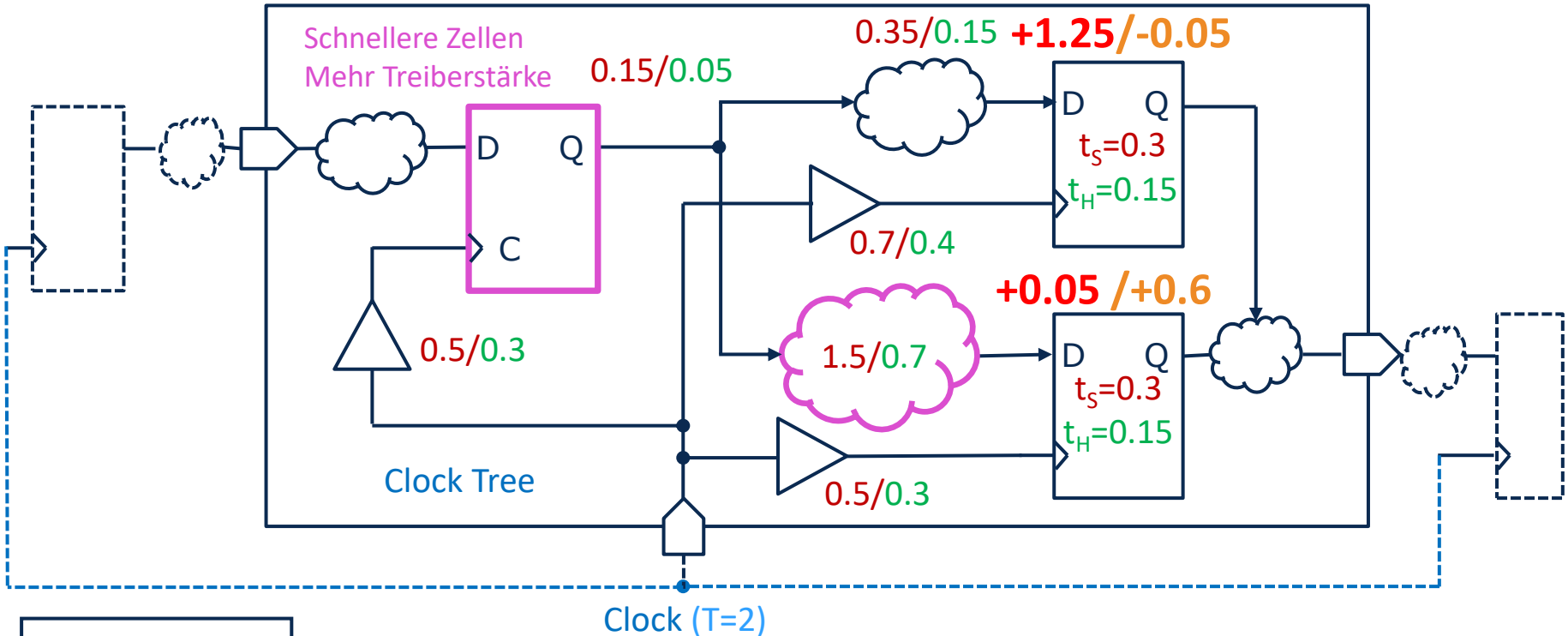
Setup Slack FF1:  $2 - 0.3 - 1.8 - 0.3 = -0.4$

Hold Slack FF2:  $0.30 + 0.15 + 0.15 - 0.4 - 0.15 = +0.05$

**Setup Violation!!**

**Hold Clean**





**Legende:**  
 worst delay (red)  
 best delay (green)  
 Setup slack (red)  
 Hold slack (orange)  
 Zeiteinheit: [ns]

Größere Delays durch P&R Parasitic Kapazitäten der Leitungen)

**Setup Slack** FF1:  $2 - 0.15 - 1.5 - 0.3 = +0.05$

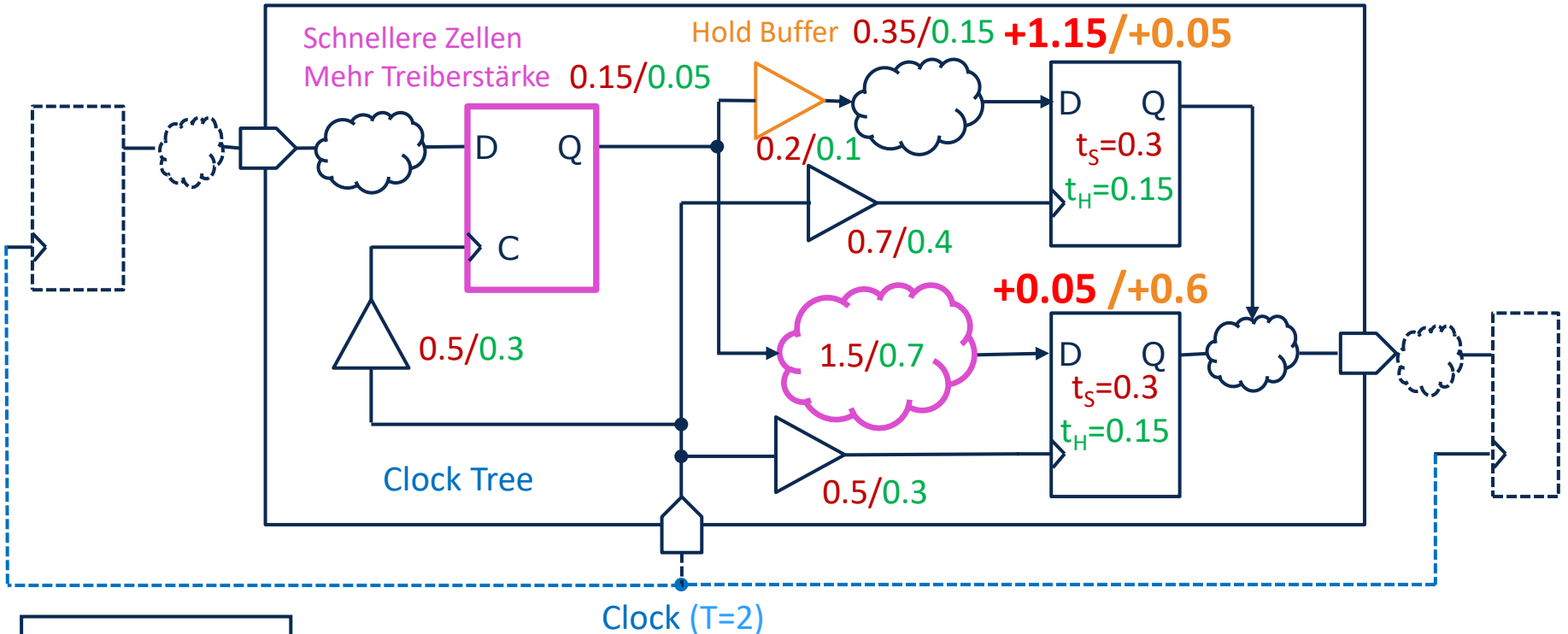
**Hold Slack** FF2:  $0.30 + 0.05 + 0.15 - 0.4 - 0.15 = -0.05$

**Setup Clean**



**Hold Violation!!!**





Legende:

worst delay

best delay

Setup slack

Hold slack

Zeiteinheit: [ns]

Größere Delays durch P&R Parasitic Kapazitäten der Leitungen)

Setup Slack FF1:  $2 - 0.15 - 1.5 - 0.3 = +0.05$

Hold Slack FF2:  $0.30 + 0.05 + 0.1 + 0.15 - 0.4 - 0.15 = +0.05$

**Setup Clean**

**Hold Clean**

