Timing von CMOS Logik

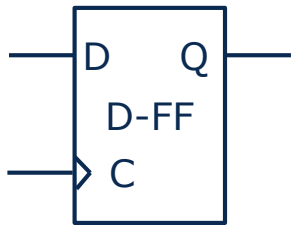
- Timing bezeichnet das Zeitverhalten von Signalen in digitalen Schaltungen
- Betrachtung von
  - Zeitpunkten und Zeitdifferenzen von Signalwechseln und Verzögerungszeiten
  - Anstiegszeiten (Transition Time) bei Signalwechseln
- Analyse des Timings zur Einhaltung von Randbedingungen (Constraints)
-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

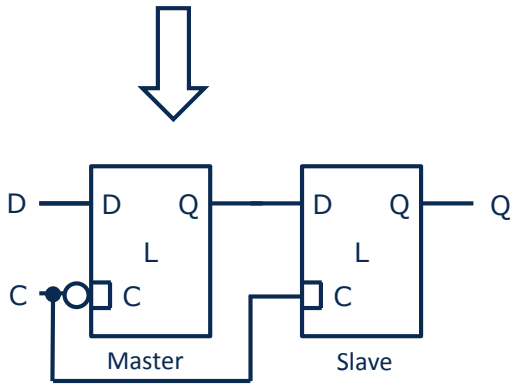
Beispiel: Simulation D- FlipFlop

Master Keeper

Master Latch Node (inverted)

D-FlipFlop

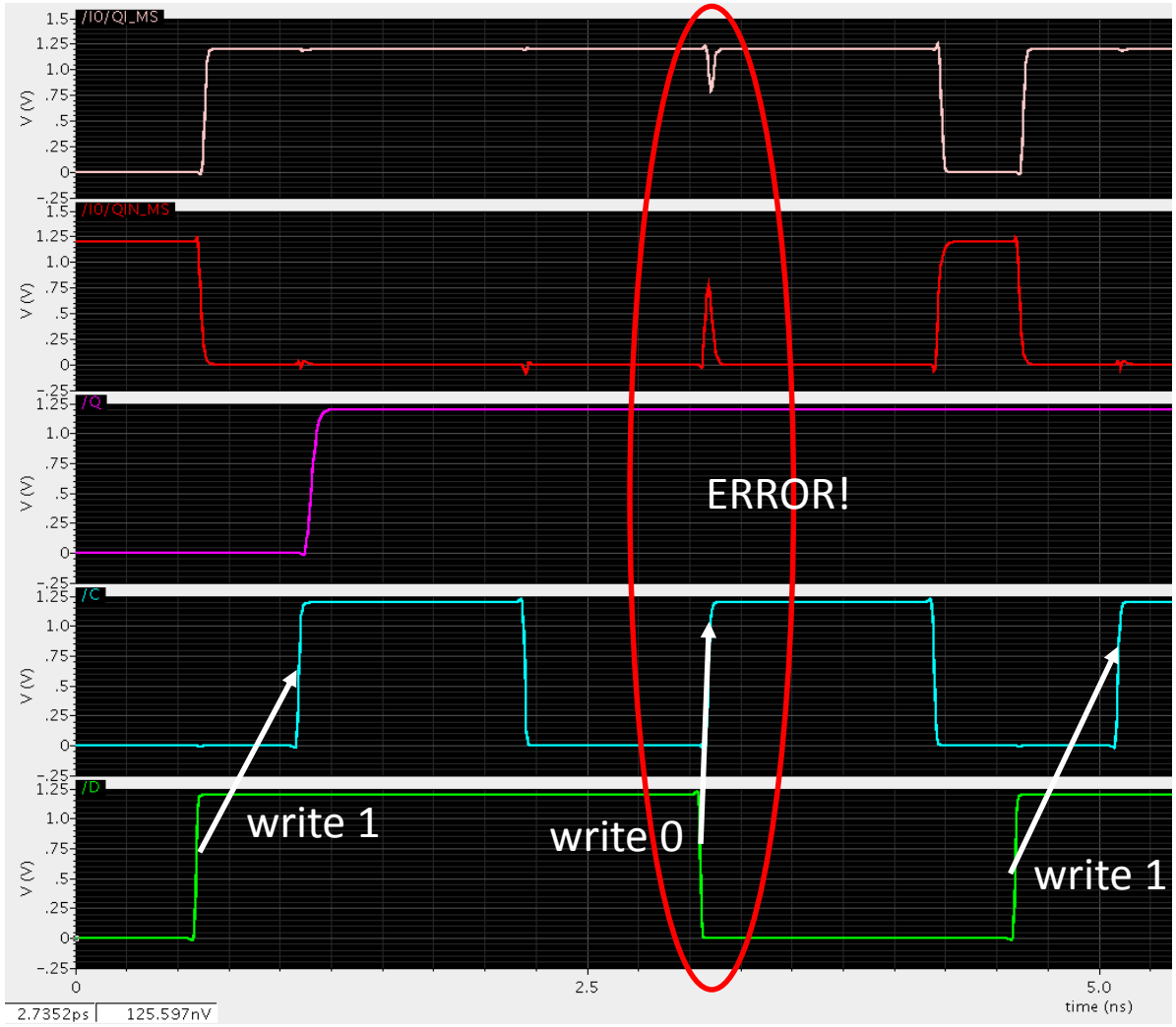




Q

C

D



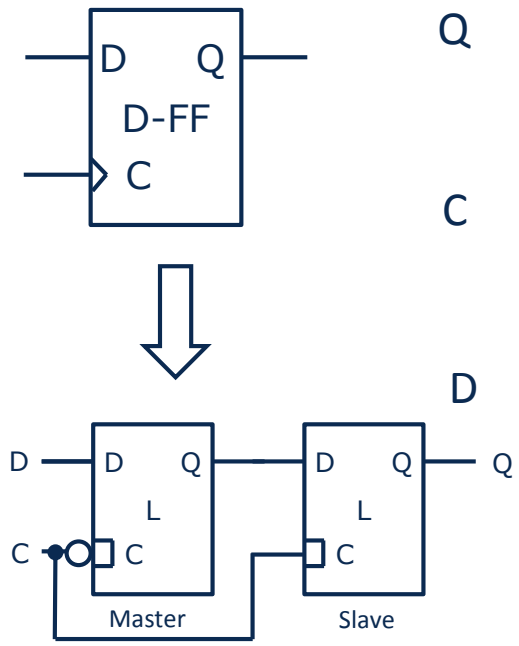
Setup Violation!

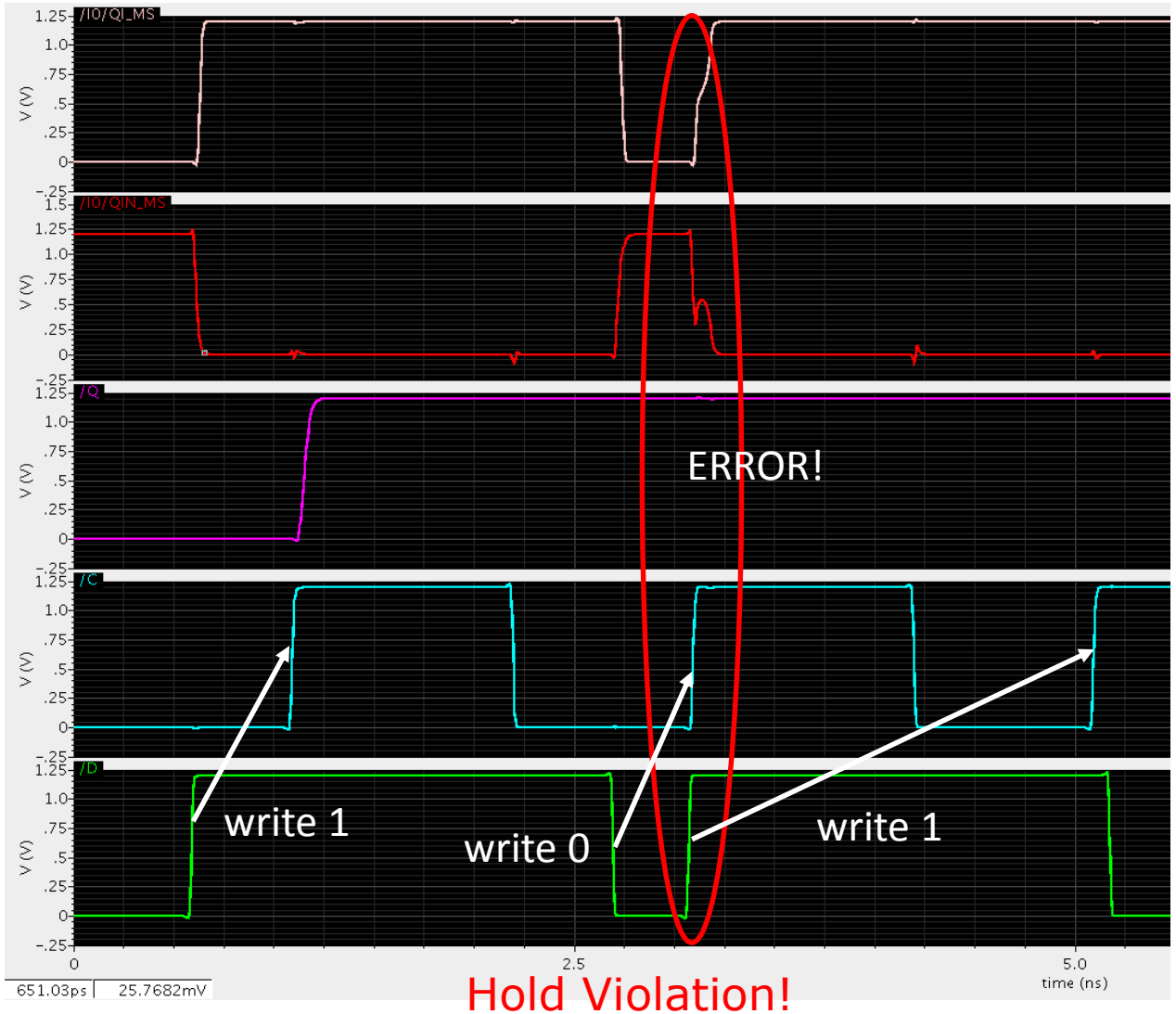
Beispiel: Simulation D-FlipFlop

Master Keeper

Master Latch Node (inverted)

D-FlipFlop





Hold Violation!

Timing Randbedingungen (Constraints)

- 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: $t_{data,i}$

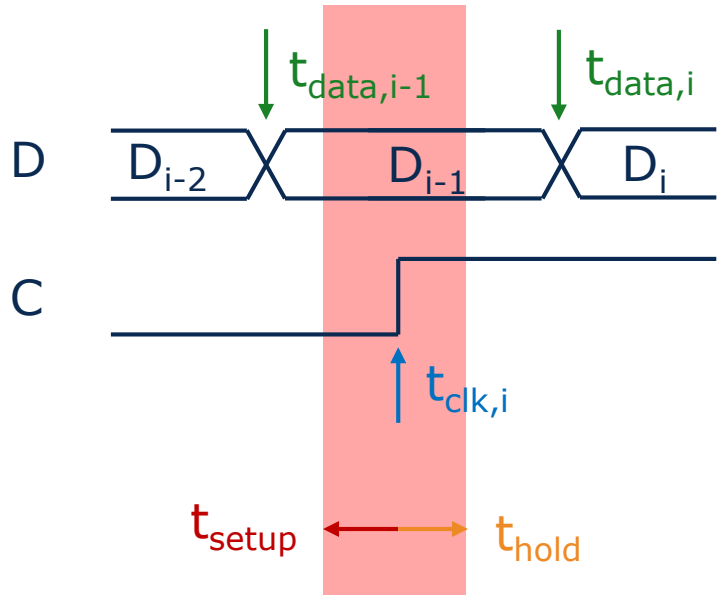
- Setup Constraint:

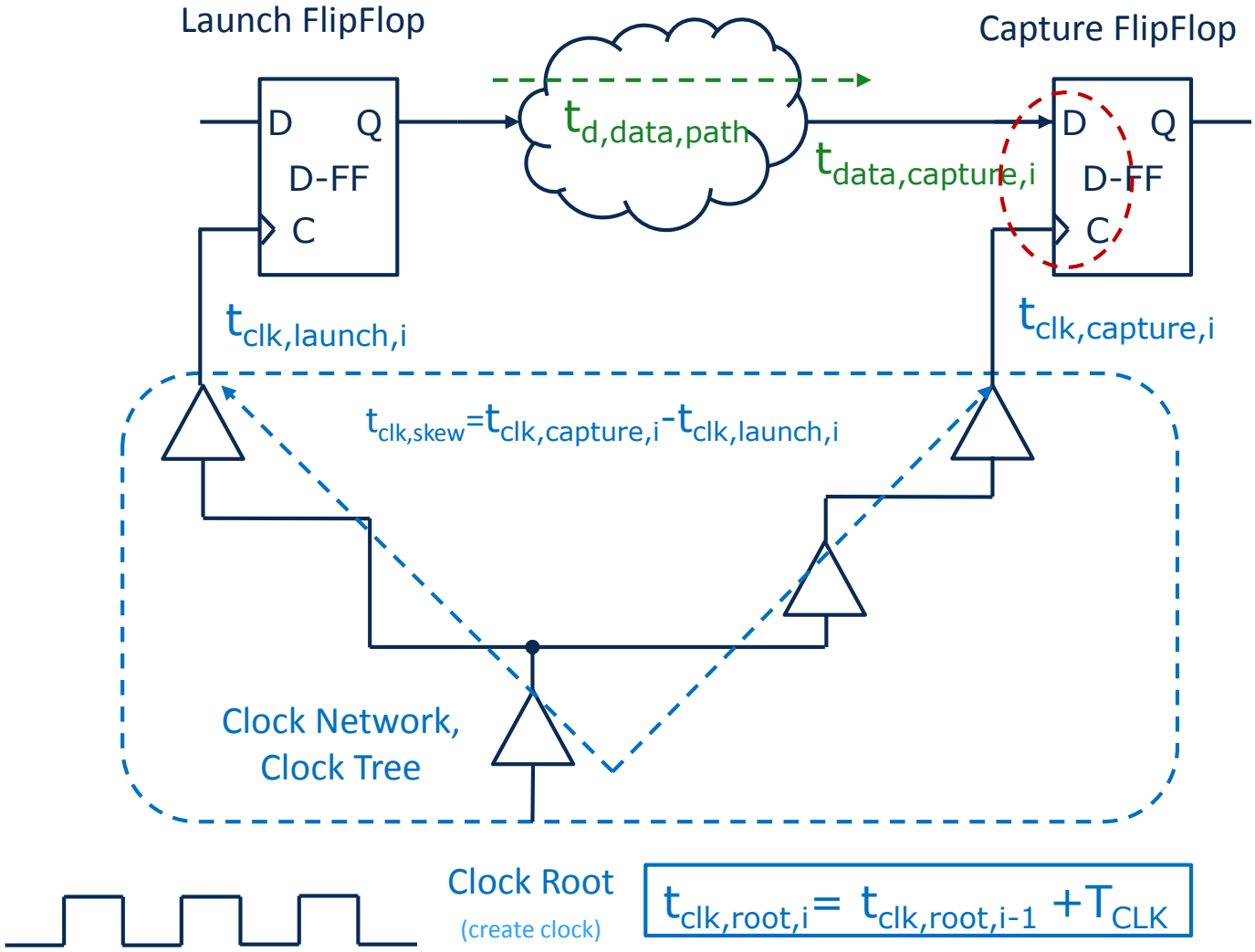
$$t_{clk,i} - t_{data,i-1} > t_{setup}$$

- Hold Constraint:

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

Übernahme von D_{i-1} mit Taktflanke i



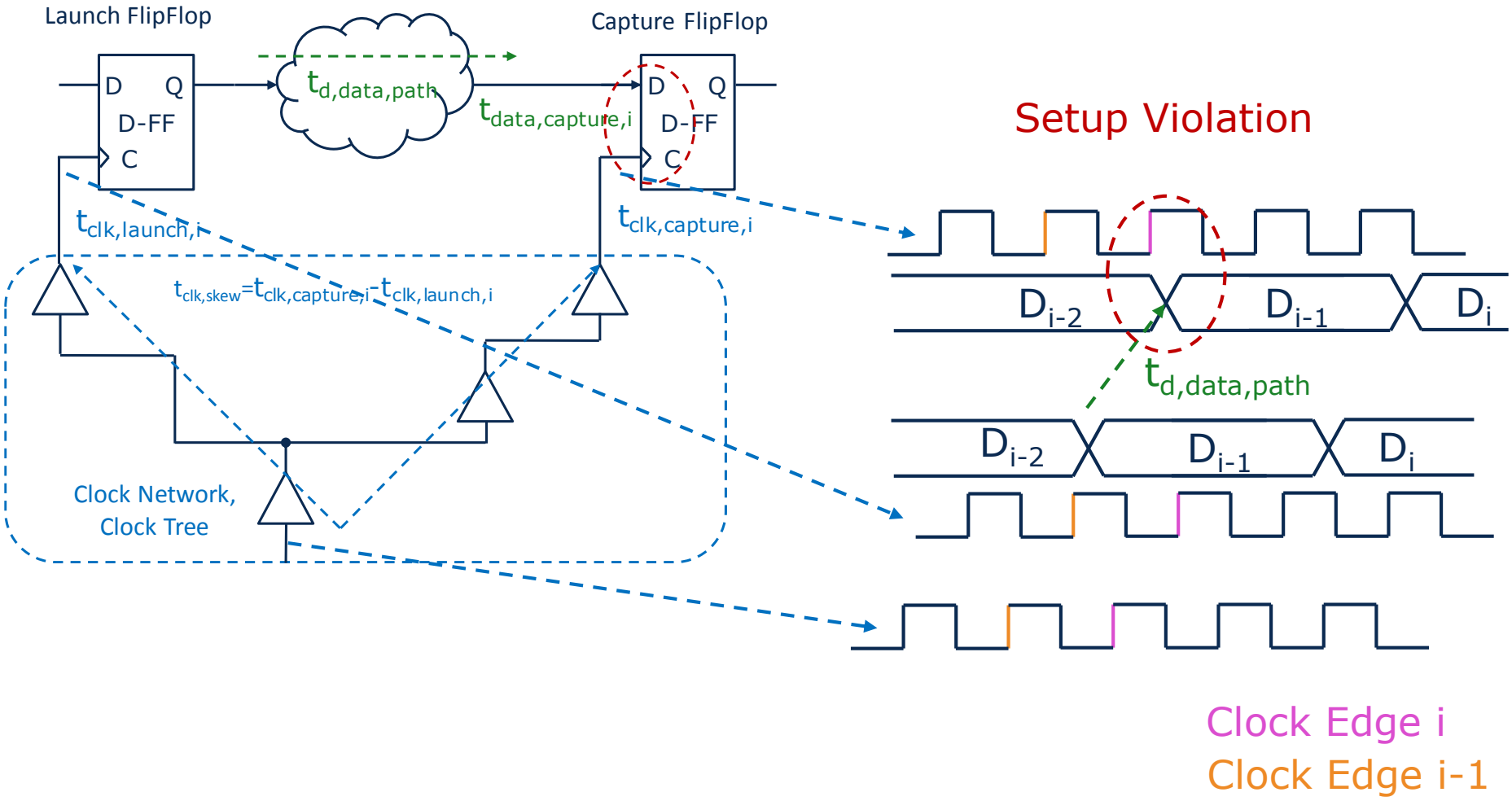

Launch FlipFlop
Capture FlipFlop
D
Q
D-FF
C
D
Q
D-FF
C
$t_{d,data,path}$
$t_{data,capture,i}$
$t_{clk,launch,i}$
$t_{clk,capture,i}$
$t_{clk,skew}=t_{clk,capture,i}-t_{clk,launch,i}$
Clock Network,
Clock Tree
Clock Root
(create clock)
$t_{clk,root,i}= t_{clk,root,i-1} +T_{CLK}$

Setup Constraint

- Setup Constraint:
  - $t_{clk,capture,i} - t_{data,capture,i-1} > t_{setup}$
  - $t_{clk,capture,i} - (t_{clk,launch,i-1} + t_{d,data,path}) > t_{setup}$
  - $t_{clk,skew} + T_{CLK} - t_{d,data,path} > t_{setup}$

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

Beispiel: Setup Violation

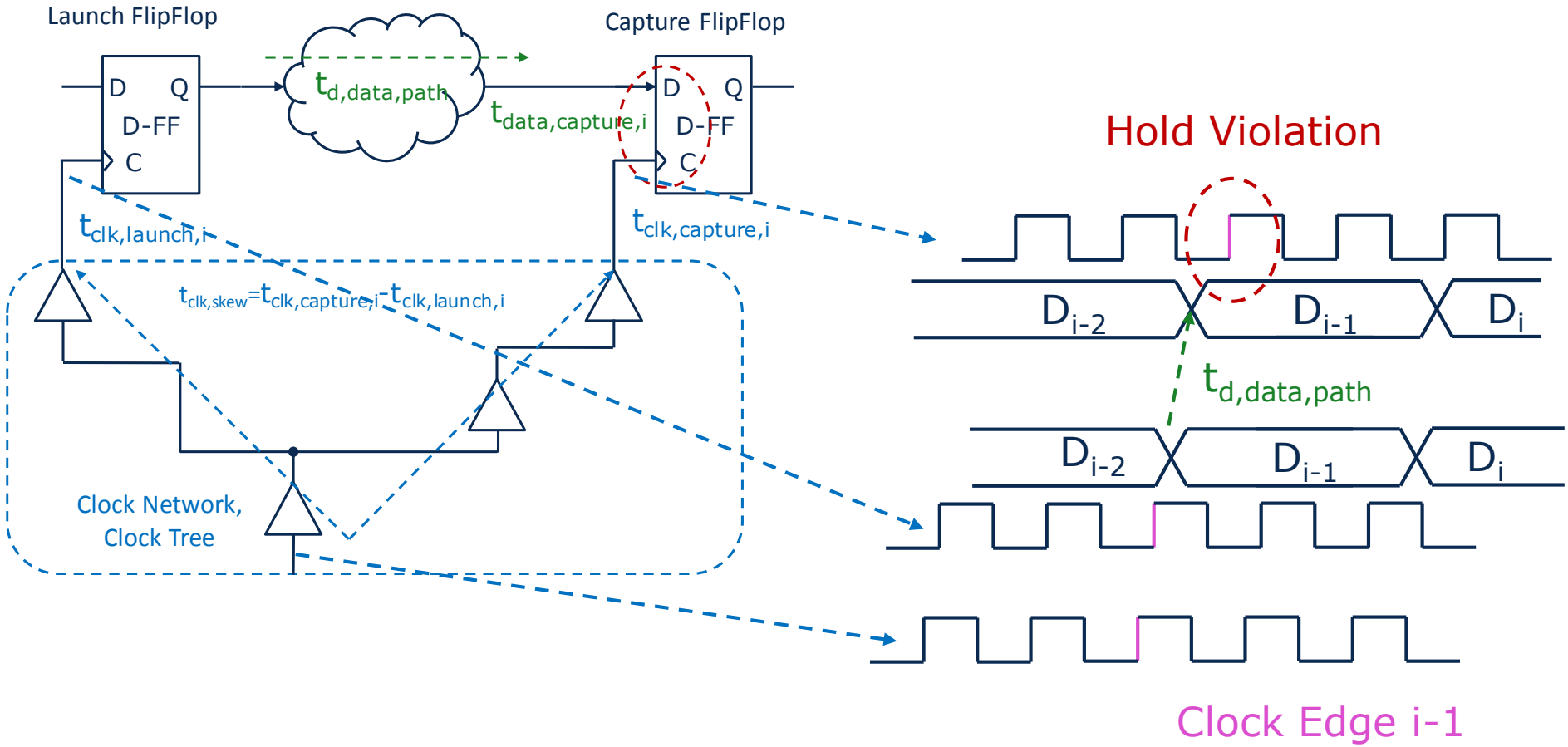


Hold Constraint

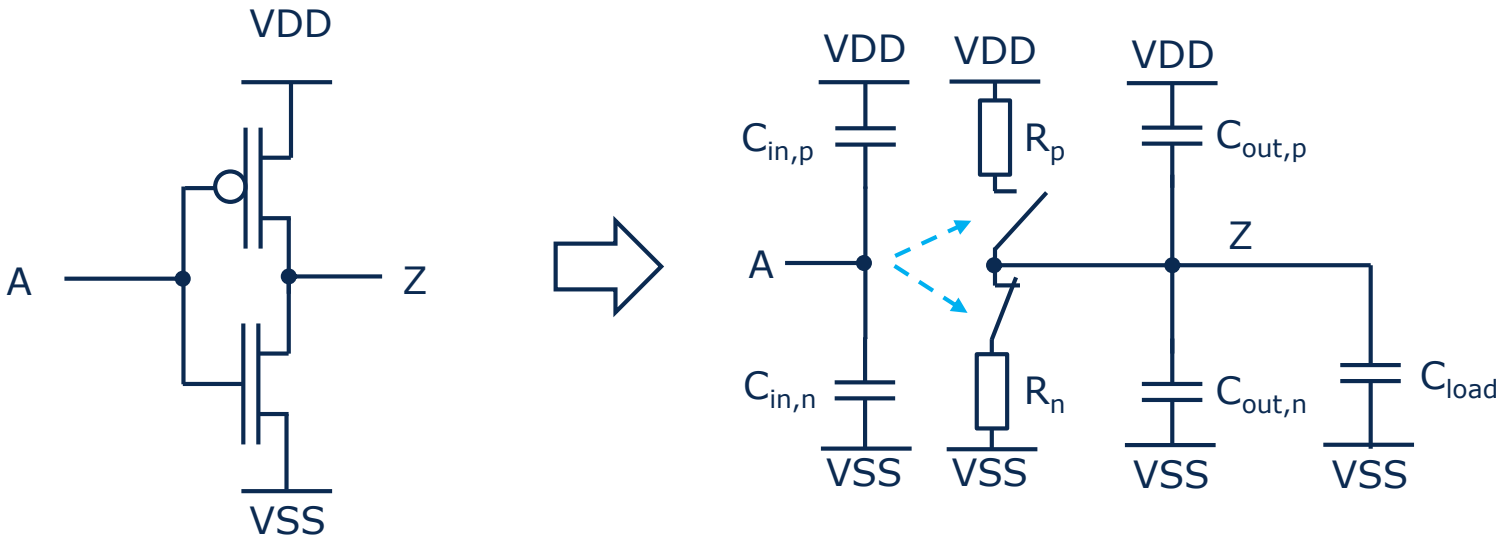
- Hold Constraint:
  - $t_{data,capture,i} - t_{clk,capture,i} > t_{hold}$
  - $(t_{clk,launch,i} + t_{d,data,path}) - t_{clk,capture,i} > t_{hold}$
  - $t_{d,data,path} - t_{clk,skew} > t_{hold}$

  - 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)

Beispiel: Hold Violation

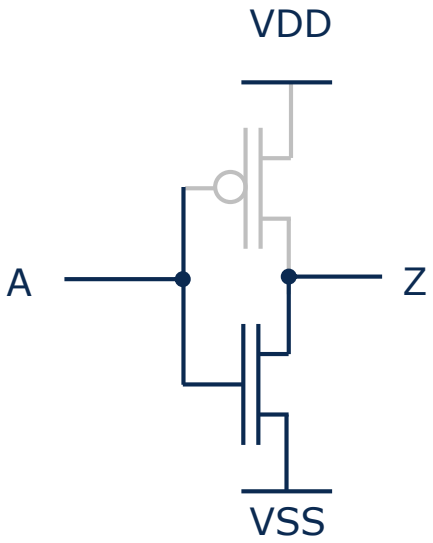


- RC Schaltungsmodell zur Berechnung der Verzögerungszeit

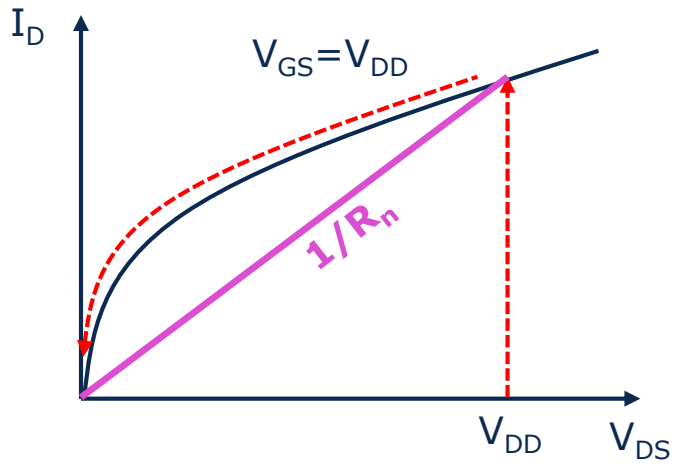


- Effektiver Schalterwiderstand von NMOS und PMOS
- Parasitäre Kapazitäten C_{in}, C_{out}, C_{load}
- Annahme: Umschalten der Eingänge **VOR** dem Umladen der Ausgänge → Vernachlässigung des Miller-Effekts

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



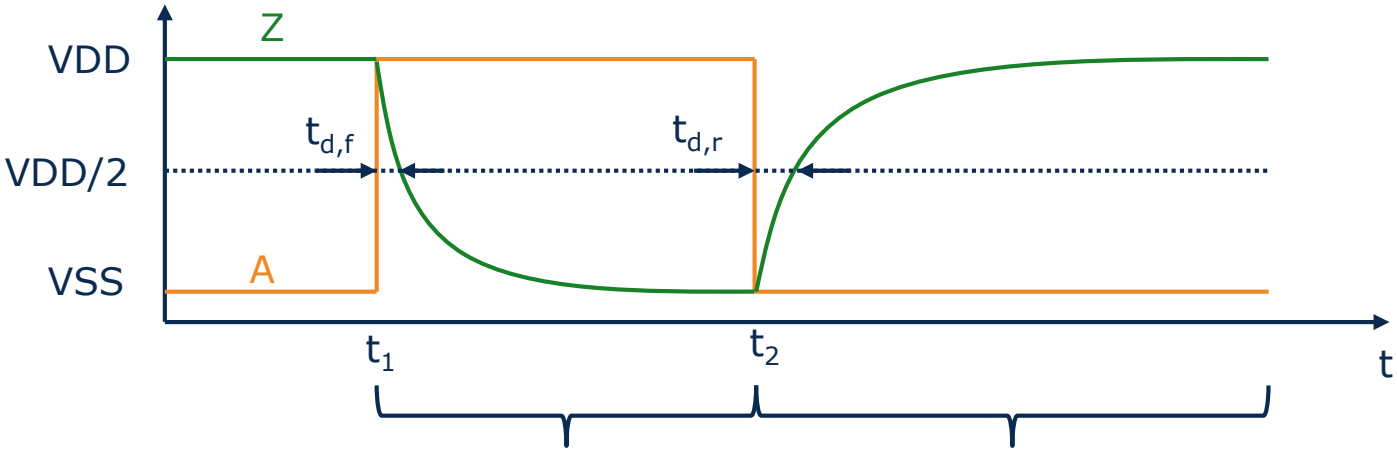
Beispiel: NMOS Pull-Down Z: 1→0



- Effektiver Schalterwiderstand
  - $R_n = \frac{V_{DD}}{\frac{KP_N}{2}\cdot\frac{W}{L}\cdot\left(V_{DD}-V_{th,N}\right)^2\cdot\left(1+\lambda_N\left(V_{th,N}\right)\right)} = R'_n \cdot \frac{L}{W}$

- Analoge Betrachtung für PMOS bei Z: 0→1

- Delay Berechnung: RC-System erster Ordnung



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

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

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

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

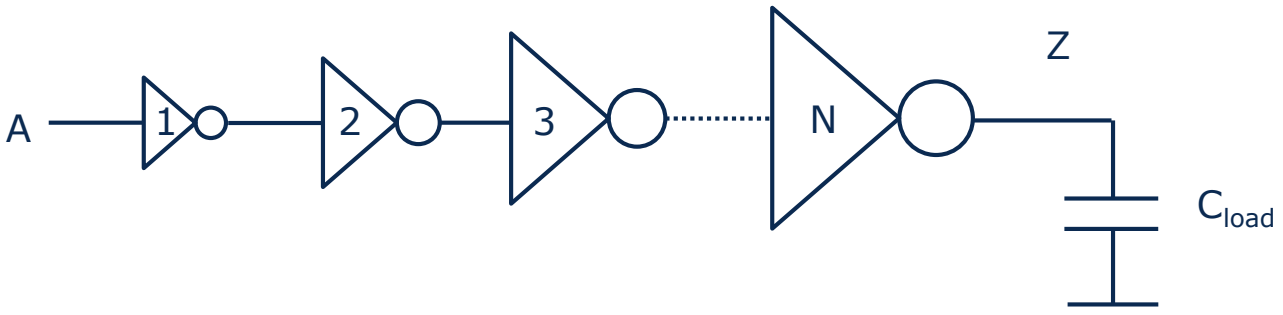
Schlussfolgerungen

- *Näherungsweise* Berechnung des Delays
- „Treiberstärke“: R↓ → C_{in}↑
  - Reduktion der Treiberstärke durch Reihenschaltungen in Pull-Up (z.B. NOR) oder Pull-Down (z.B. NAND) Pfaden
- Pull-Up und Pull-Down Pfade belasten den Input kapazitiv
  - Selektives Sizing von NMOS und PMOS beeinflusst das Delay!
- Ausgangskapazität meist durch C_{load} dominiert (Leitung + Eingangskapazität der Folgestufen)
- FanOut (FO):
  - **Verhältnis der Lastkapazität zur eigenen Eingangskapazität**

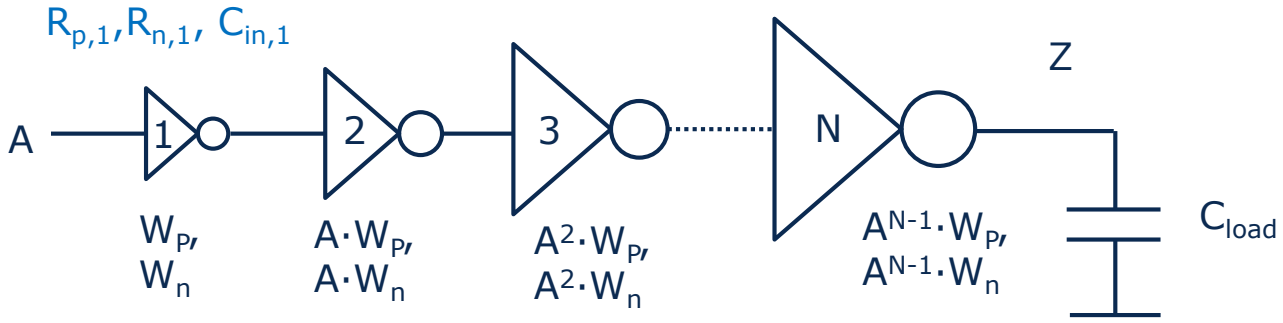
- Ausblick: Methode des Logical Efforts
  - Manuelles Optimieren von Logik bzgl. Delay durch Betrachtung des Verhältnisses aus Eingangskapazität zu Treiberstärke

Beispiel: Optimales Fan-Out

- Aufgabe:
  - Treiben einer großen Kapazität C_{load} mit einer Inverterkette
  - Welches ist das optimale Treiberstärken-Verhältnis zwischen den einzelnen Stufen für **minimales Delay** A→Z ?



Beispiel: Optimales Fan-Out



- Sizing der Stufen um Faktor A
- Reduktion der effektiven Widerstände
  - $R_{n,i} = \frac{R_{n,1}}{A^{i-1}}$, $R_{p,i} = \frac{R_{p,1}}{A^{i-1}}$
- Erhöhung der Input Kapazität
  - $C_{in,i} = A^i \cdot C_{in,1}$
  - $C_{load} = A^N \cdot C_{in,1}$

Beispiel: Optimales Fan-Out

- Delay einer Stufe i:
  - $\left(t_{d,r} + t_{d,f}\right)_i = \ln(2) \cdot \left(\frac{R_{p,1}}{A^{i-1}} + \frac{R_{n,1}}{A^{i-1}}\right) \cdot \left(A^{i-1} \cdot C_{out,1} + A^i \cdot C_{in,1}\right)$
  - $\left(t_{d,r} + t_{d,f}\right)_i = \ln(2) \cdot \left(R_{p,1} + R_{n,1}\right) \cdot \left(C_{out,1} + A \cdot C_{in,1}\right)$
- Das Delay aller Stufen ist gleich
- Gesamtes Delay:
  - $\left(t_{d,r} + t_{d,f}\right)_{ges} = \ln(2) \cdot N \cdot \left(R_{p,1} + R_{n,1}\right) \cdot \left(C_{out,1} + A \cdot C_{in,1}\right)$
  - $\left(t_{d,r} + t_{d,f}\right)_{ges} = \ln(2) \cdot N \cdot \left(R_{p,1} + R_{n,1}\right) \cdot \left(C_{out,1} + \left(\frac{C_{load}}{C_{in,1}}\right)^{1/N} \cdot C_{in,1}\right)$

Beispiel: Optimales Fan-Out

- Minimales Delay:
- $$\frac{d(t_{d,r}+t_{d,f})_{ges}}{dN} = \ln(2) \cdot \left((R_{p,1} + R_{n,1}) \cdot C_{out,1} + (R_{p,1} + R_{n,1}) \cdot C_{in,1} \cdot \left(\left(\frac{C_{load}}{C_{in,1}}\right)^{1/N} + N \cdot \left(\frac{C_{load}}{C_{in,1}}\right)^{1/N} \cdot \frac{\ln\left(\frac{C_{load}}{C_{in,1}}\right)}{-N^2} \right) \right) = 0$$

- Vernachlässigung von $(R_{p,1} + R_{n,1}) \cdot C_{out,1}$ (intrinsisches Delay des ersten Inverters)
- $N_{opt} = ln \frac{C_{load}}{C_{in,1}}$
- $A_{opt} = \left(e^{N_{opt}}\right)^{1/N_{opt}} = e \approx 2{,}72$

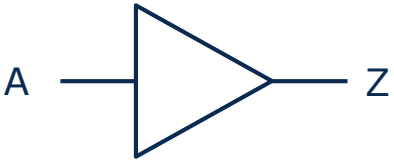
  - → Vergrößerung der Treiberstärke der einzelnen Stufen um ≈2,72 resultiert in minimalem Delay des Buffers

Beispiel: Optimales Fan-Out

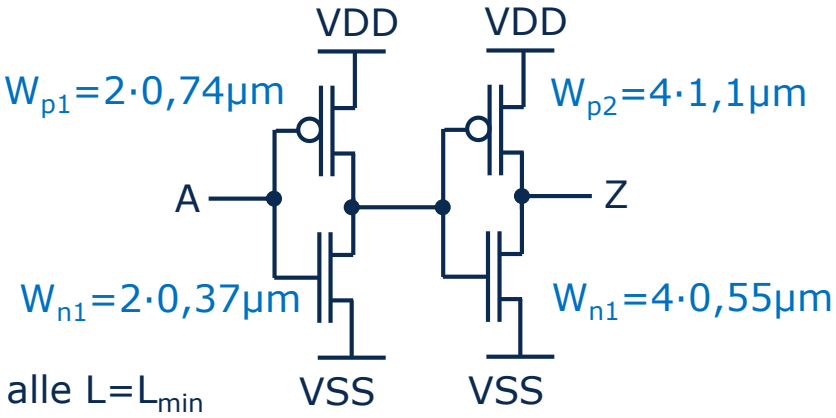
- Treiben einer Last mit Inverter-Kette:
  - erster Inverter: $R_{n,1}=R_{p,1}=4k\Omega$, $C_{in,1}=2{,}0fF$, $C_{out,1}=1{,}3fF$
  - Last: $C_{load}=1pF$
- → $N_{opt}\approx 6{,}21$ → **6 Stufen**
- $\left(t_{d,r}+t_{d,f}\right)_{opt}=\ln(2)\cdot 6\cdot\left(R_{p,1}+R_{n,1}\right)\cdot\left(C_{out,1}+e\cdot C_{in,1}\right)$ **≈226ps**

- Treiben dieser Last direkt mit erstem Inverter:
  - $\left(t_{d,r}+t_{d,f}\right)_{opt}=\ln(2)\cdot\left(R_{p,1}+R_{n,1}\right)\cdot\left(C_{out,1}+C_{load}\right)$ **≈5607ps**

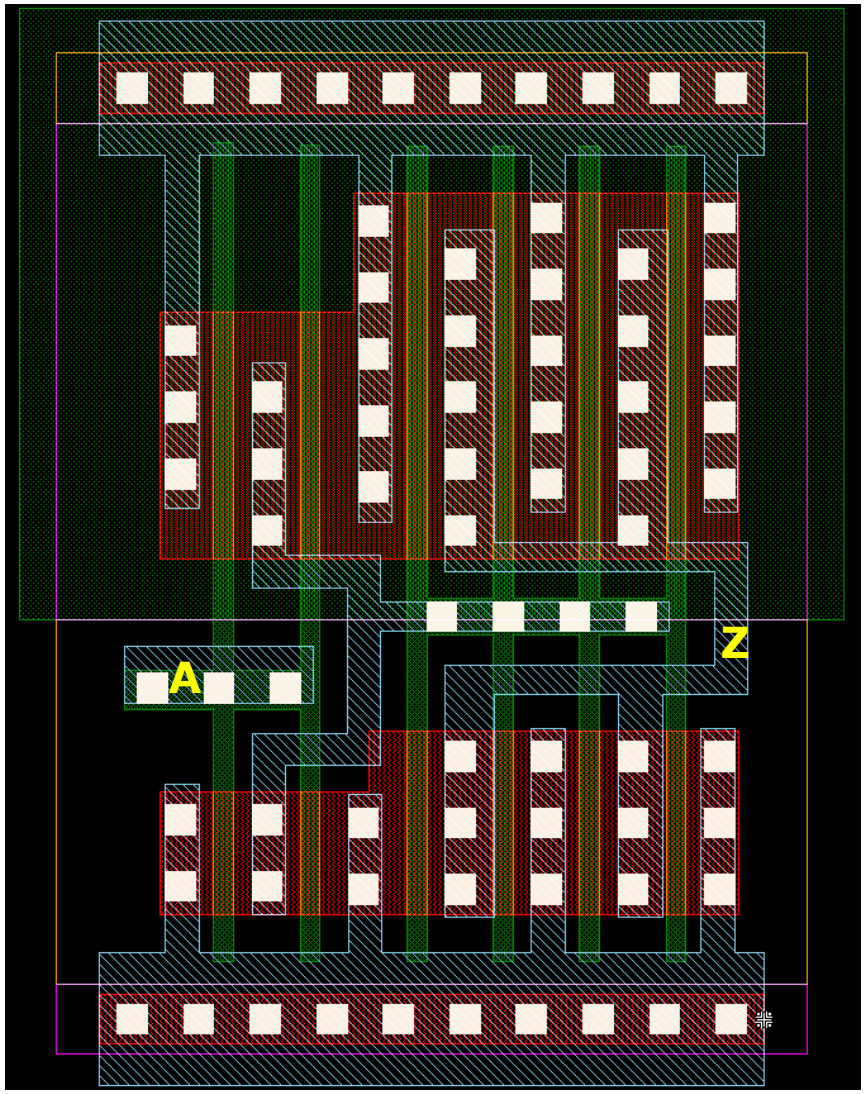
Beispiel: Clock Buffer Standardzelle



CBUFX8
Standardzelle

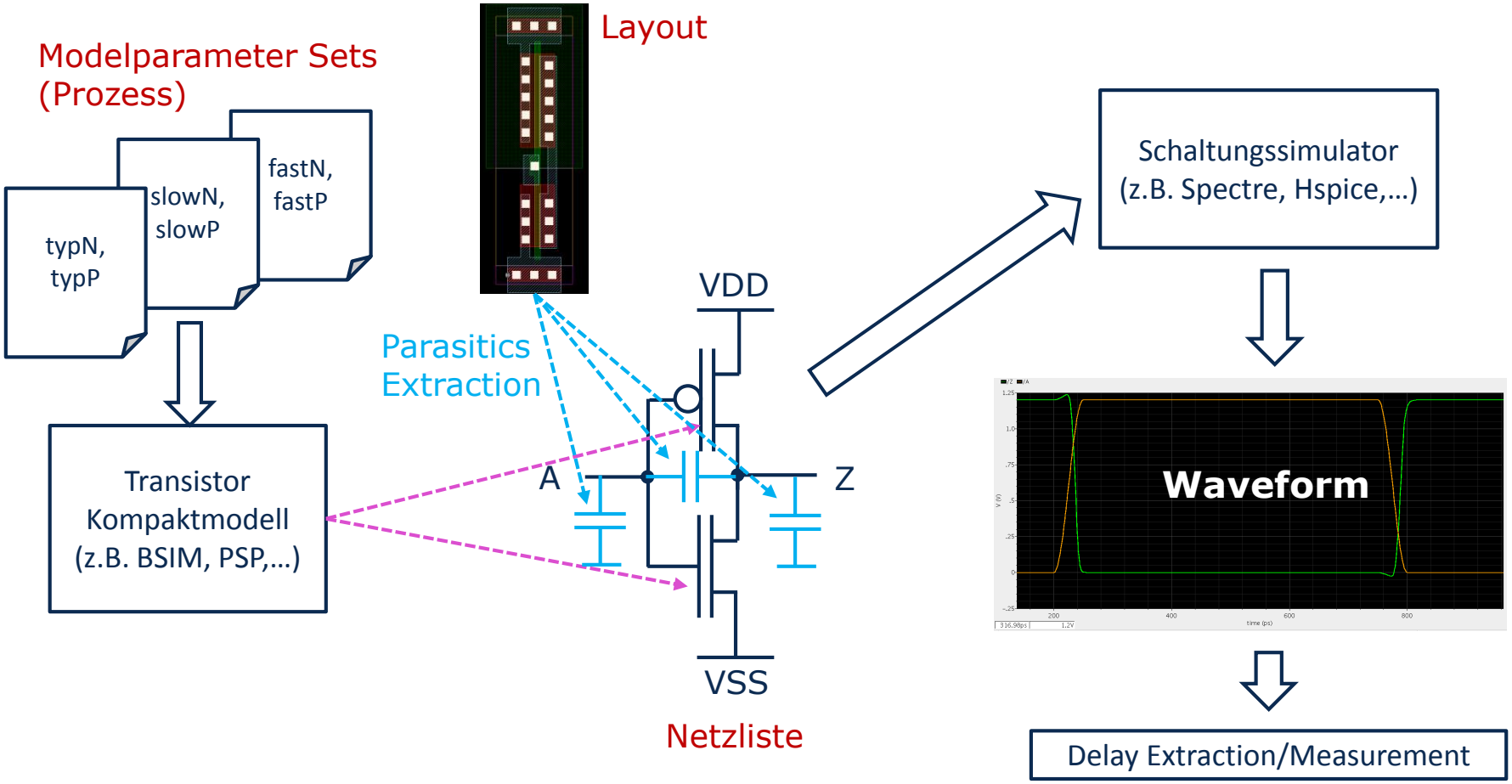


→ FanOut von A≈3

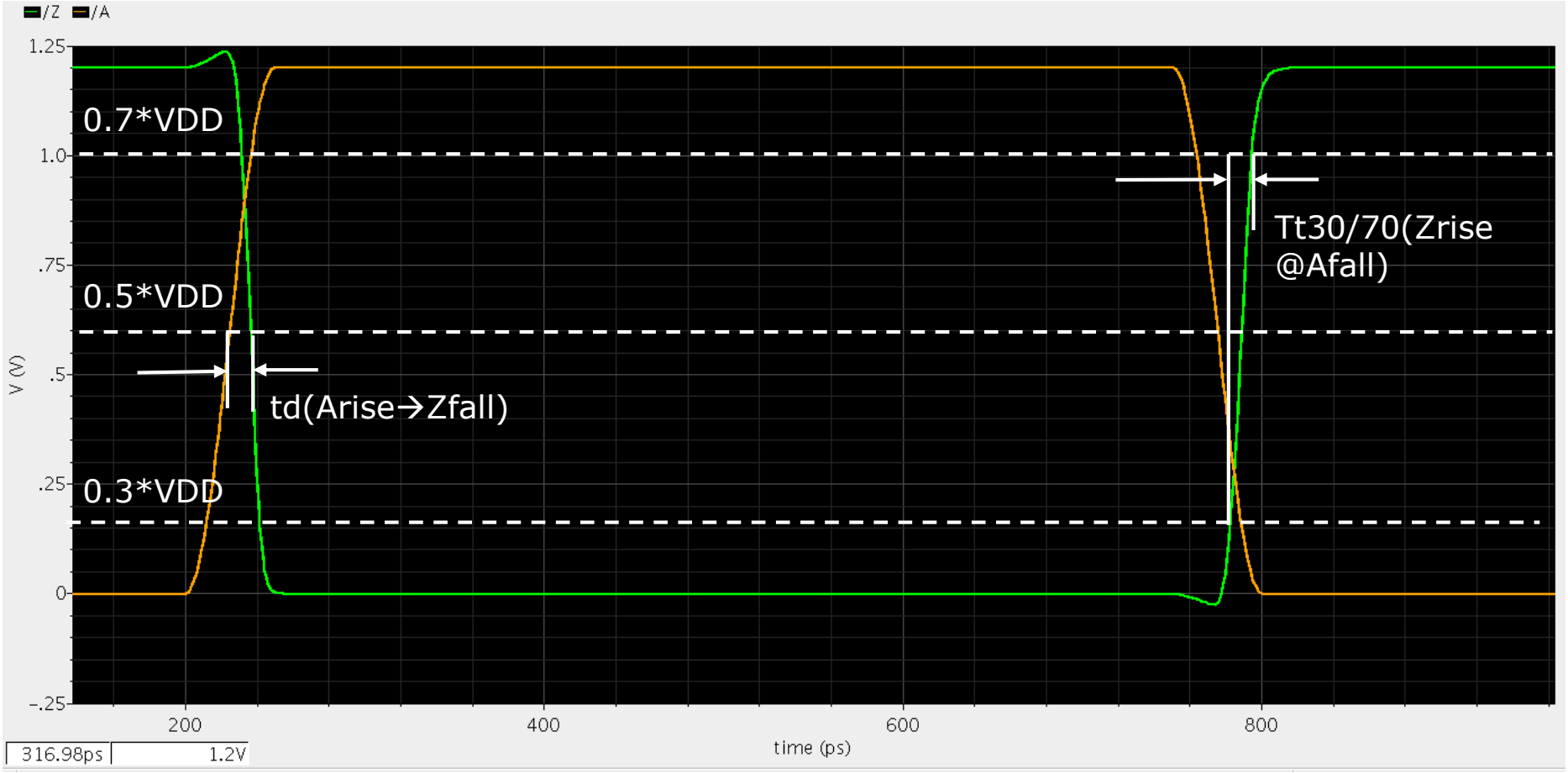


Timing Bestimmung durch Schaltungssimulation

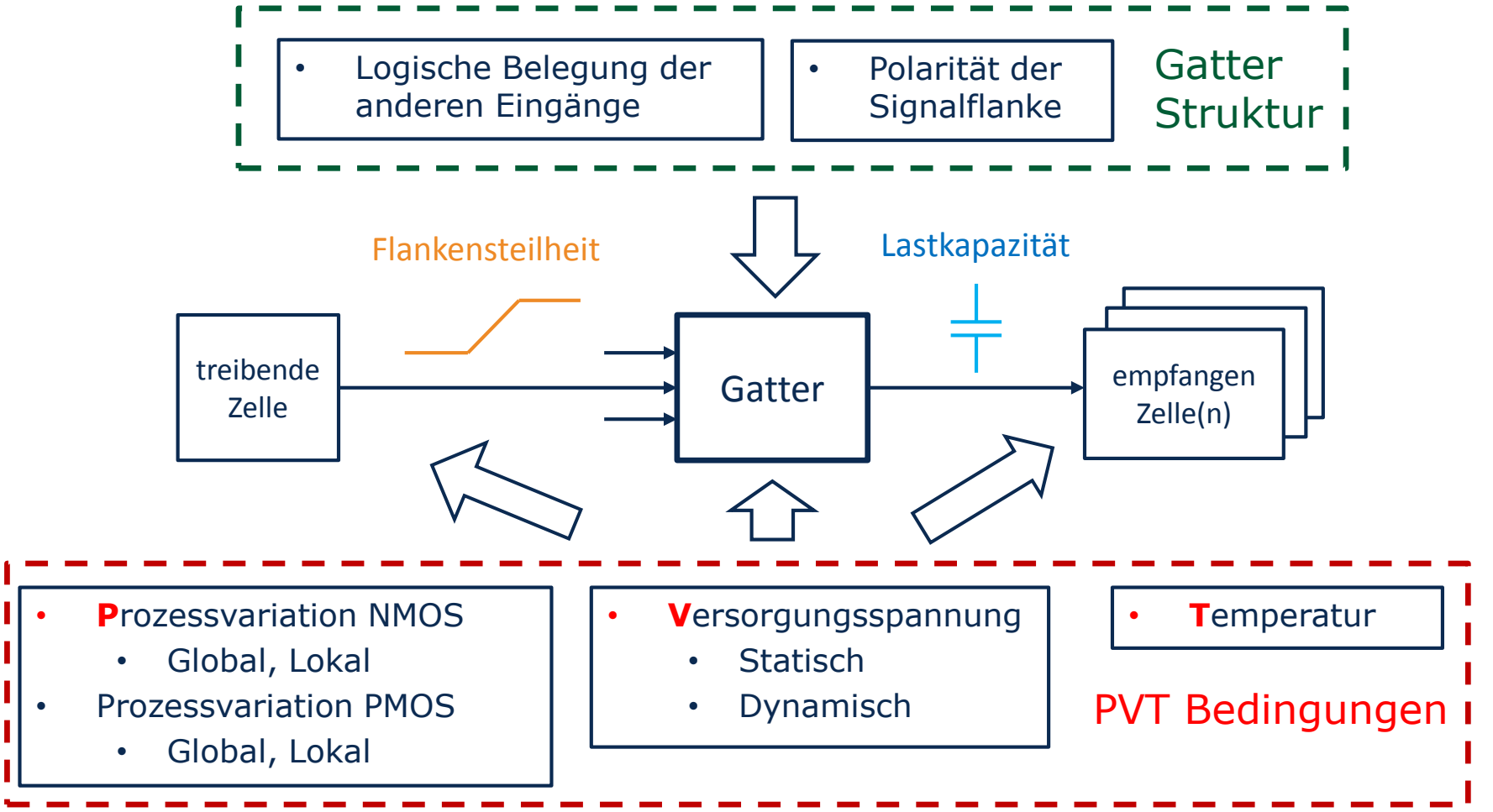
- Bestimmung des Timings durch Schaltungssimulation
- Nutzung von Kompaktmodellen für die Transistoren (z.B. BSIM, PSP)
- Extraktion von parasitären Layout-Kapazitäten und -Widerständen



Beispiel: simulierter Signalverlauf mit Timing Messungen

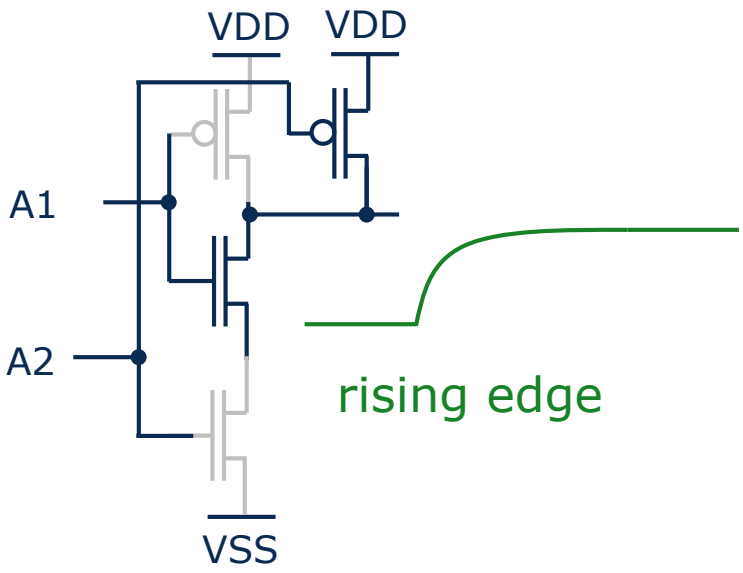


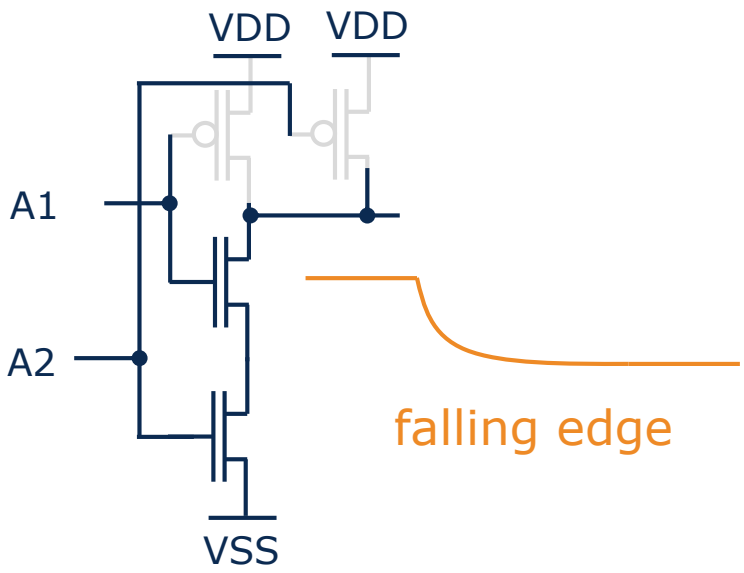
Abhängigkeit des Timings



Abhängigkeit des Timings: Polarität der Signalflanken

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

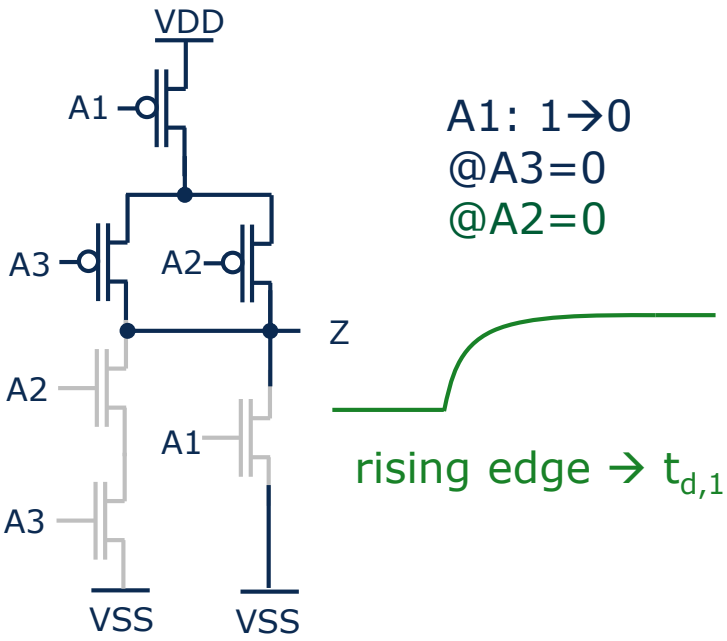




- **Berücksichtigung im Design Flow:**
  - Modellierung individueller Delays in pro Signalflanke

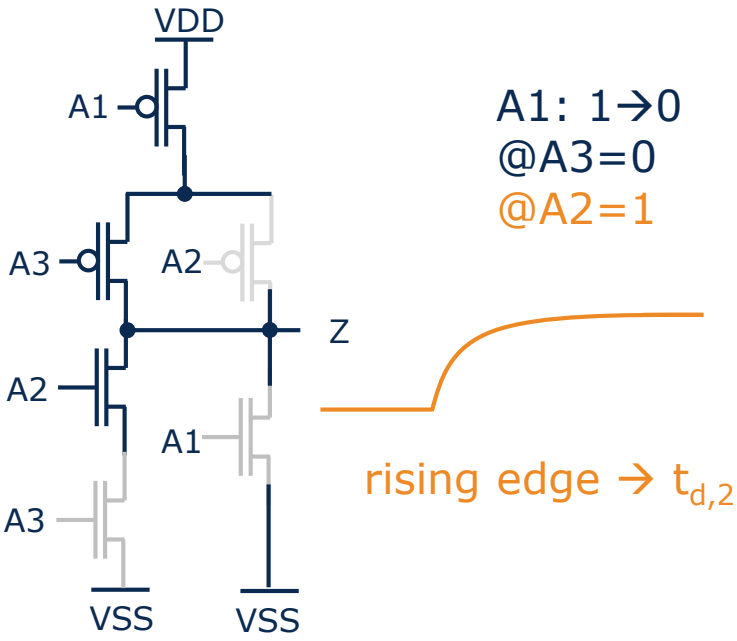
Abhängigkeit des Timings: Signalbelegung anderer Inputs

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



A1: 1→0
@A3=0
@A2=0

rising edge → $t_{d,1}$



A1: 1→0
@A3=0
@A2=1

rising edge → $t_{d,2}$

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

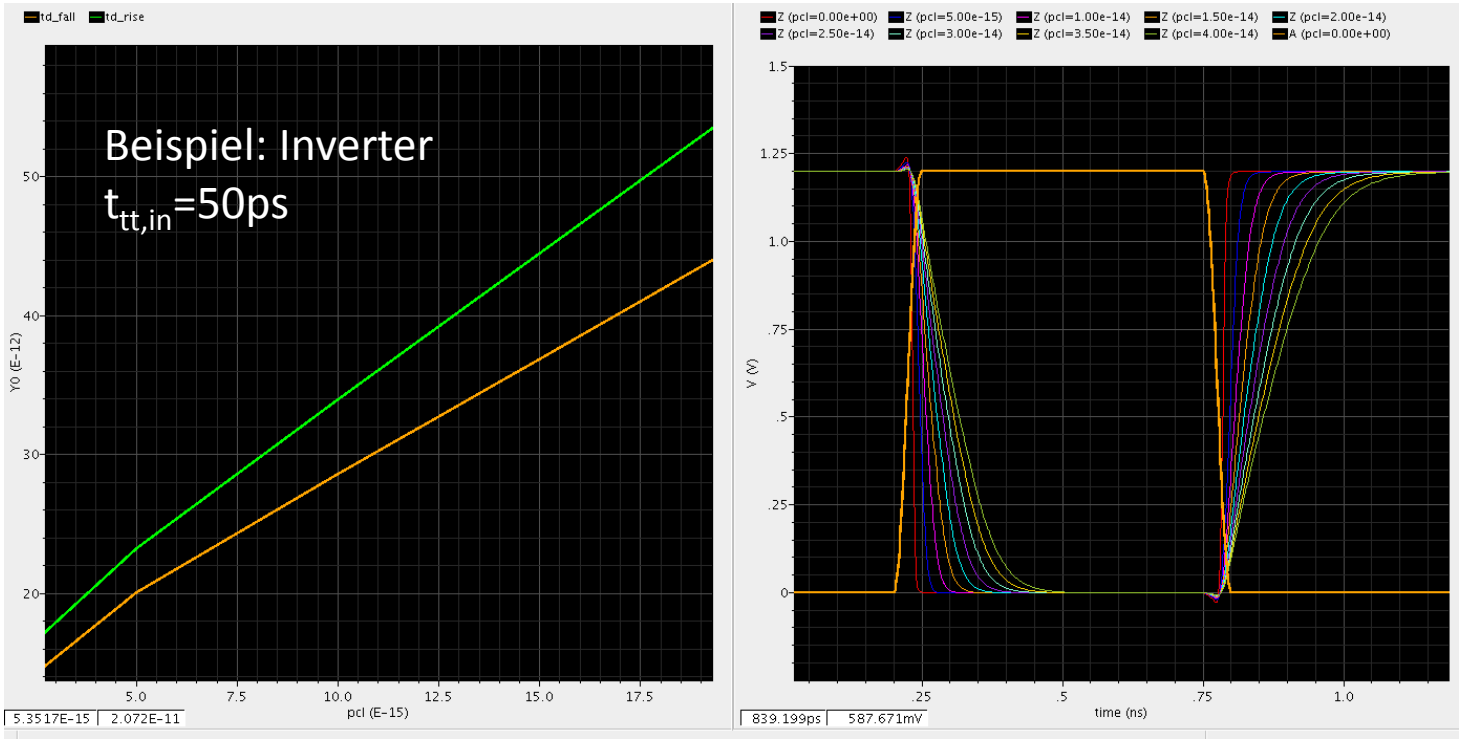
Abhängigkeit des Timing: Lastkapazität

- Direkte Abhängigkeit des Delays von der Lastkapazität

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

(analog für $t_{d,r}$)

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

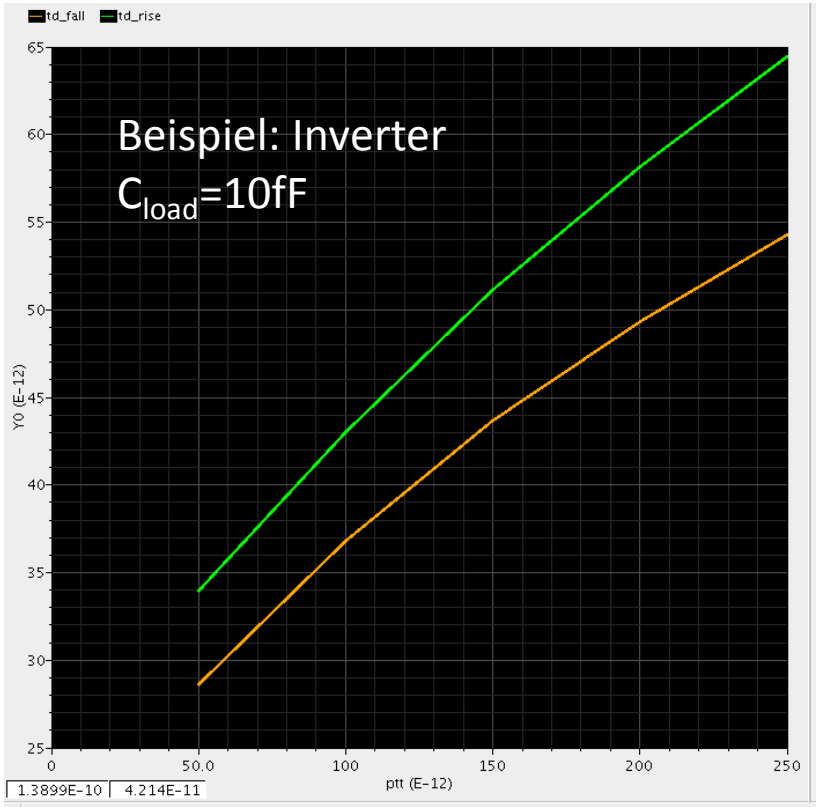


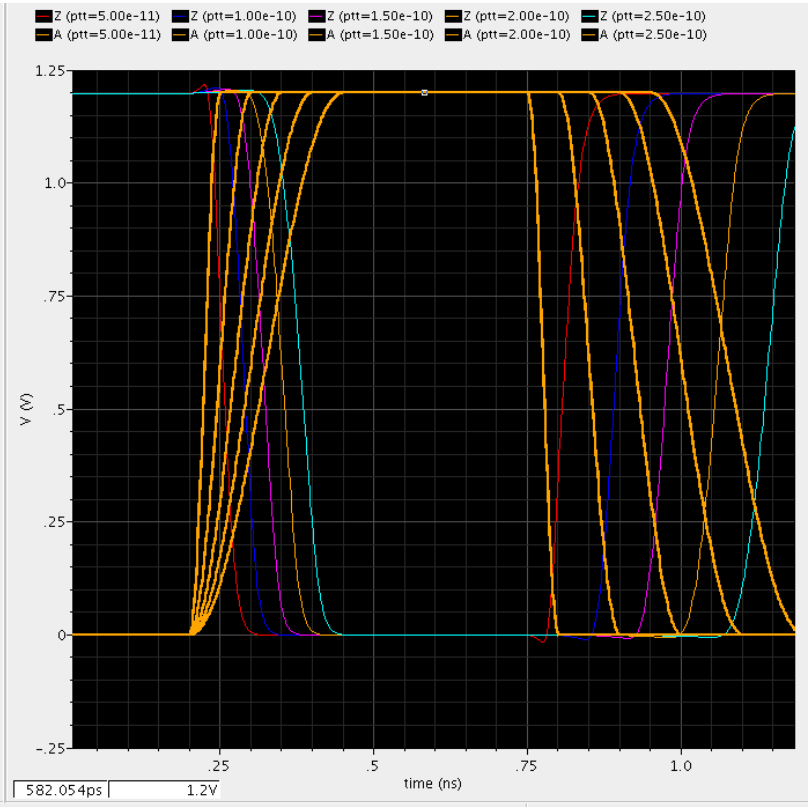
- **Berücksichtigung im Design Flow:**
  - Modellierung des Delays $t_d = f(C_{load})$ und Output Transition Time $t_{tt,out} = g(C_{load})$, z.B. als Lookup-Tabelle

Abhängigkeit des Timing: Eingangstransition

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

$t_{tt,in} \uparrow \rightarrow t_d \uparrow$





- **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

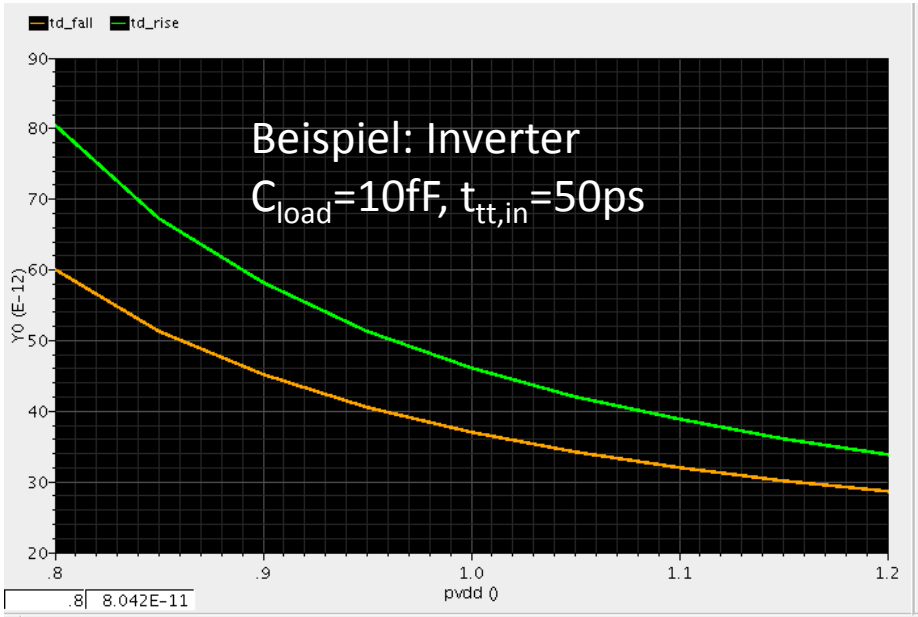
Abhängigkeit des Timing: Versorgungsspannung

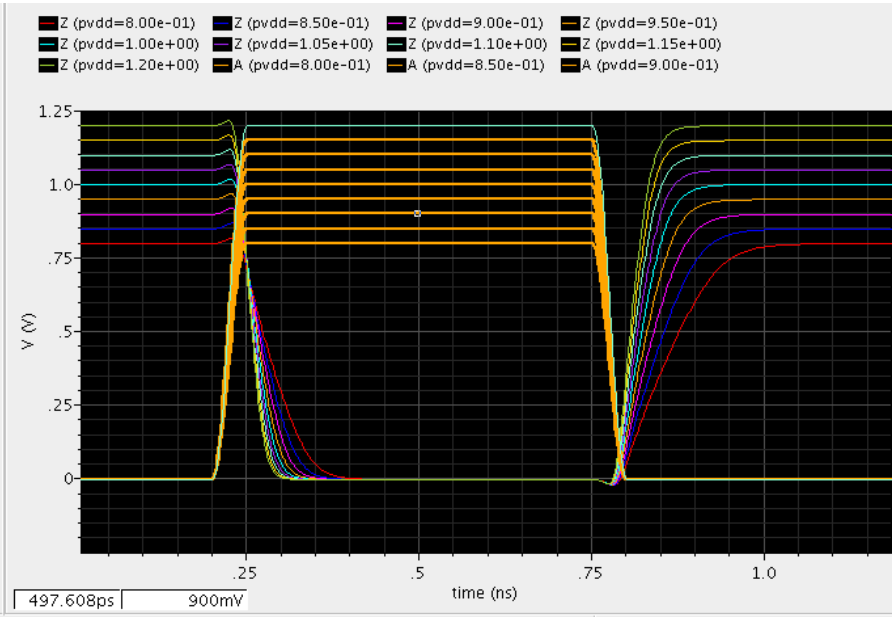
- Abhängigkeit des Delays von der statischen VDD

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

(analog für $t_{d,r}$)

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





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

Abhängigkeit des Timing: Temperatur

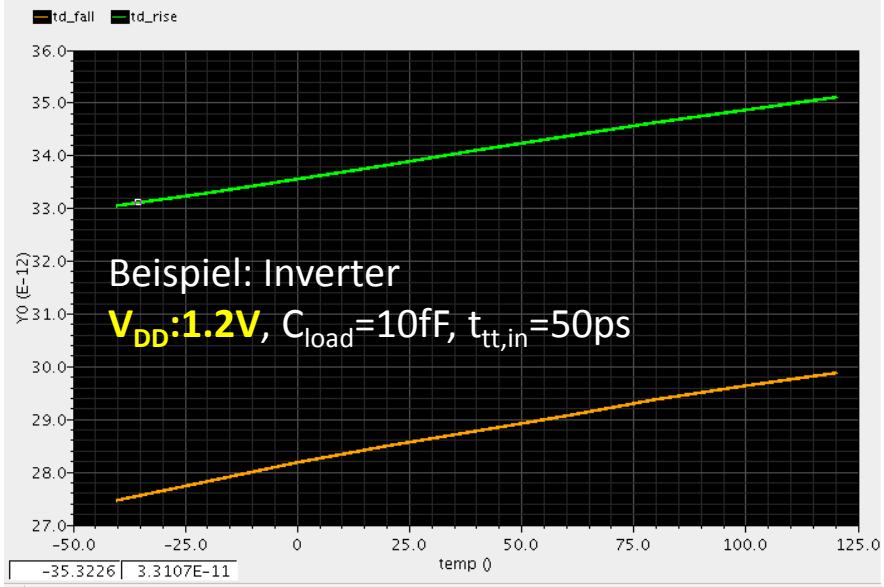
- Temperatur beeinflusst maßgeblich die Schwellspannung V_{th} und die Leitfähigkeit des Kanals (KP, λ)

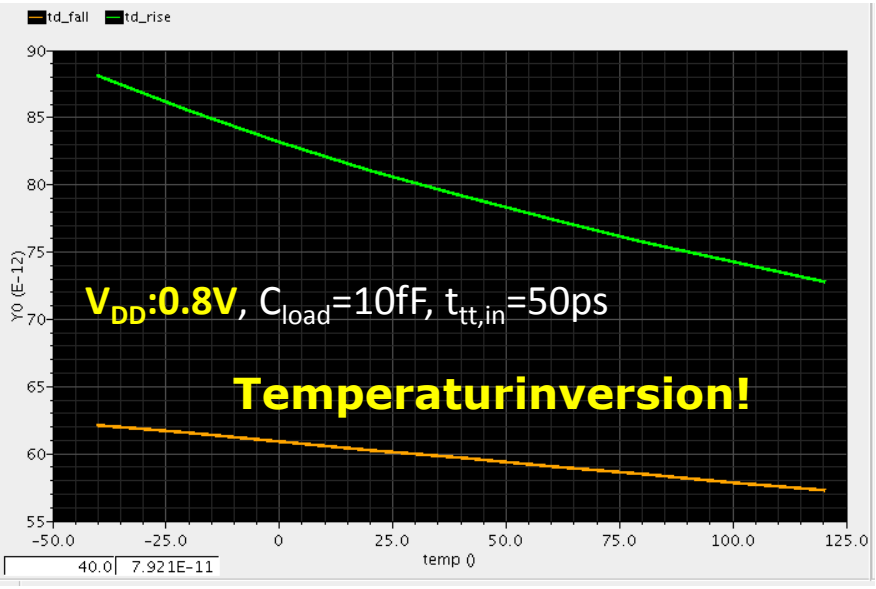
$$t_{d,f} = \ln(2) \cdot \frac{V_{DD}}{\frac{KP_{N/P}}{2} \cdot \frac{W}{L} \cdot (V_{DD} - V_{th,N})^2 \cdot (1 + \lambda_N(V_{th,N}))} \cdot (C_{out} + C_{load})$$

(analog für $t_{d,r}$)

T ↑ → V_{th} ↓

T ↑ → KP ↓





- **Berücksichtigung im Design Flow:**
  - Bestimmung des Timings durch Schaltungssimulation im Temperaturbereich des Zielsystems (z.B. -40°C bis 125°C) → **PVT Corner**

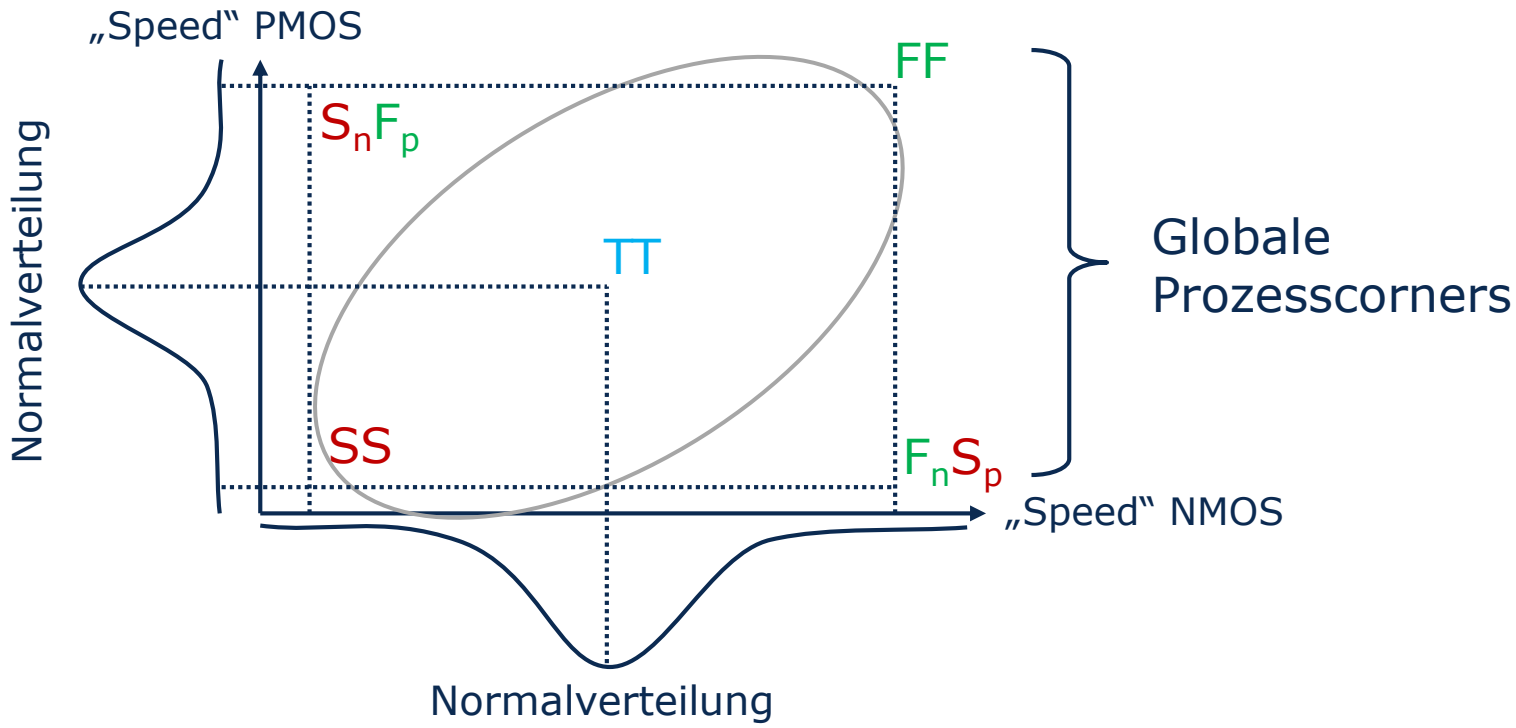
Prozessvariationen in CMOS Technologie

- Der CMOS Fertigungsprozess unterliegt **zufälligen Schwankungen**
- → Die Eigenschaften von NMOS und PMOS Transistoren unterliegen Schwankungen (z.B. W, L, V_{th}, μ, 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**

Prozessvariationen in CMOS Technologie

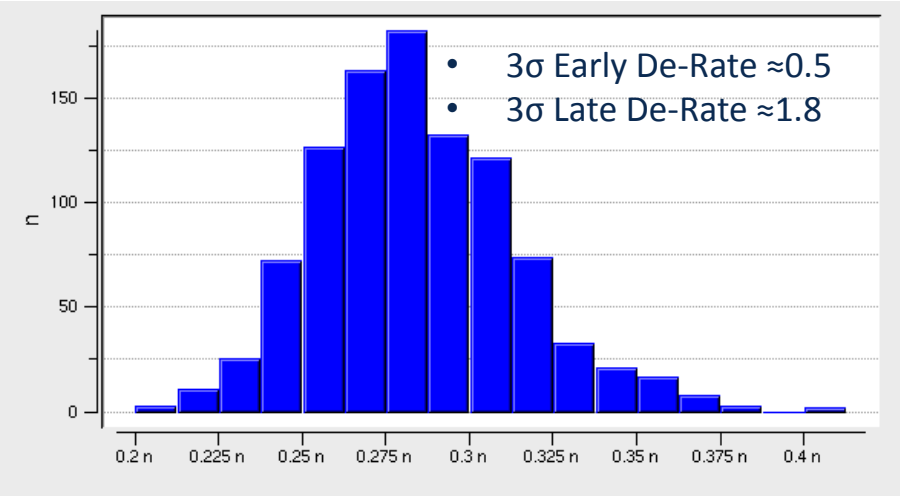
- Unabhängige, globale Variation der NMOS und PMOS Parameter
- Definition von Prozess Corners an den +-3σ Grenzen der Parameterverteilungen



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

Abhängigkeit des Timing: Prozess (lokal)

- 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 $0.5 \cdot t_d$ … $1.8 \cdot t_d$ aufweisen

- **Berücksichtigung im Design Flow:**
  - Extraction der Delay Variabilität durch Monte-Carlo Simulationen und De-Rating in der Timing Analyse (z.B. AOCV Methode)

Zwischenfazit

- 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:**
  - **P**rozess (NMOS,PMOS), Backend (extraction)
    - z.B. tc (TT, typical backend RC), wc (SS, worst RC), bc (FF, best RC)
  - **V**ersorgungsspannung
    - Spannungsbereich laut Chip Spezifikation
    - z.B. 1.0V±10% → (0.9V; 1.0V; 1.1V)
  - **T**emperatur
    - Temperaturbereich lauf Chip Spezifikation
    - z.B. -40°C … 125°C → (-40°C; 25°C; 125°C)

- **Charakterisierung** der Standardzellen in den Corners
  - Simulation der Delays in Abhängigkeit von Struktur, Transition Zeit, Lastkapazität) → siehe Abschnitt Standardzellbibliotheken

Beispiel: Corners

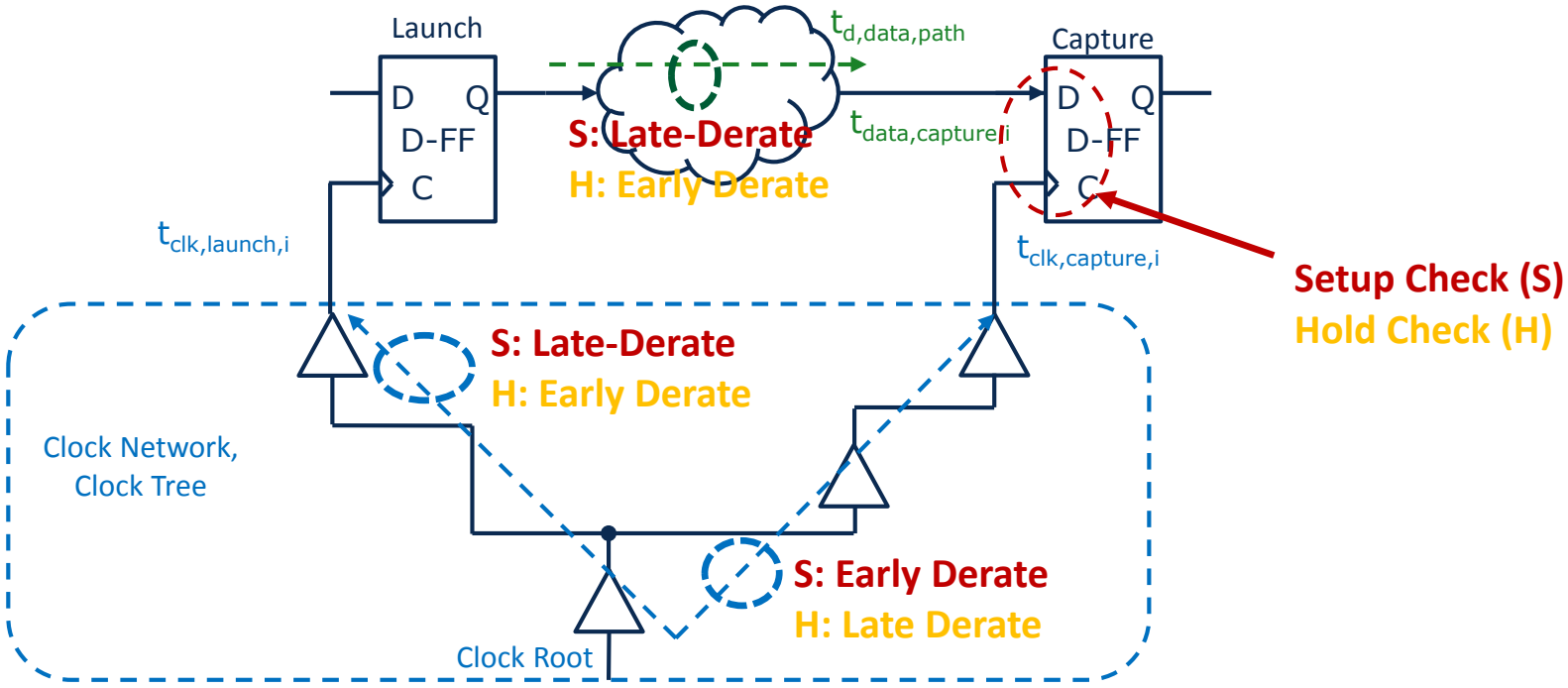
- 28nm CMOS Standardzellenbibliothek HPSNLIB
- Prozess und Backend Parasitics:
  - wc (SS und worst RC), tc (TT und typical RC), bc (FF und best RC)
- VDD Corners:
  - 1.0V ± 10%, 1.1V ± 5% 0.8V ± 5%
- Temperatur:
  - -40°C bis 125°C

- Standardzellen charakterisiert in **15 Corners**
- 3 typical, 6 worst, 6 best

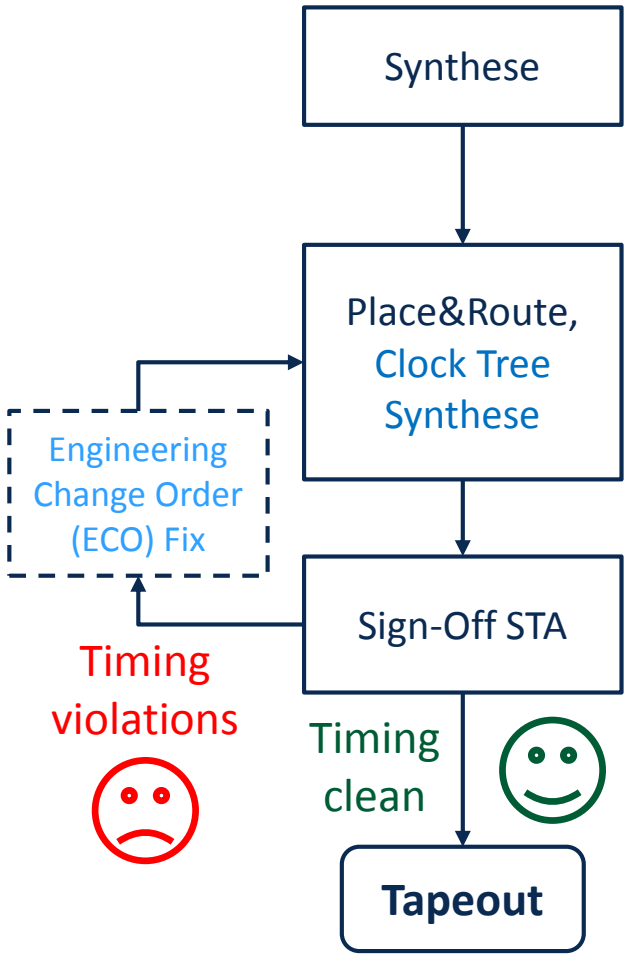
```
hpsnlib_g28_9t_RVT_bc_0d84V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_0d84V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d10V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d10V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d155V_125C.nldm.lib
hpsnlib_g28_9t_RVT_bc_1d155V_m40C.nldm.lib
hpsnlib_g28_9t_RVT_tc_0d80V_25C.nldm.lib
hpsnlib_g28_9t_RVT_tc_1d00V_25C.nldm.lib
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
```

Timing Analyse unter Variabilität des Timings

- 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



Timing Sign-off Strategie



	Checks	Corners	Beispiel
Synthese	Setup	worst timing	wc_0d90v_125c
Place&Route, Clock Tree Synthese	Setup + Hold	worst timings best timings	wc_0d90v_125c bc_1d10v_m40c
Sign-Off STA	Setup + Hold	worst timings best timings	wc_0d90v_125C bc_1d10v_m40C wc_0d90v_m40c bc_1d10v_125C

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