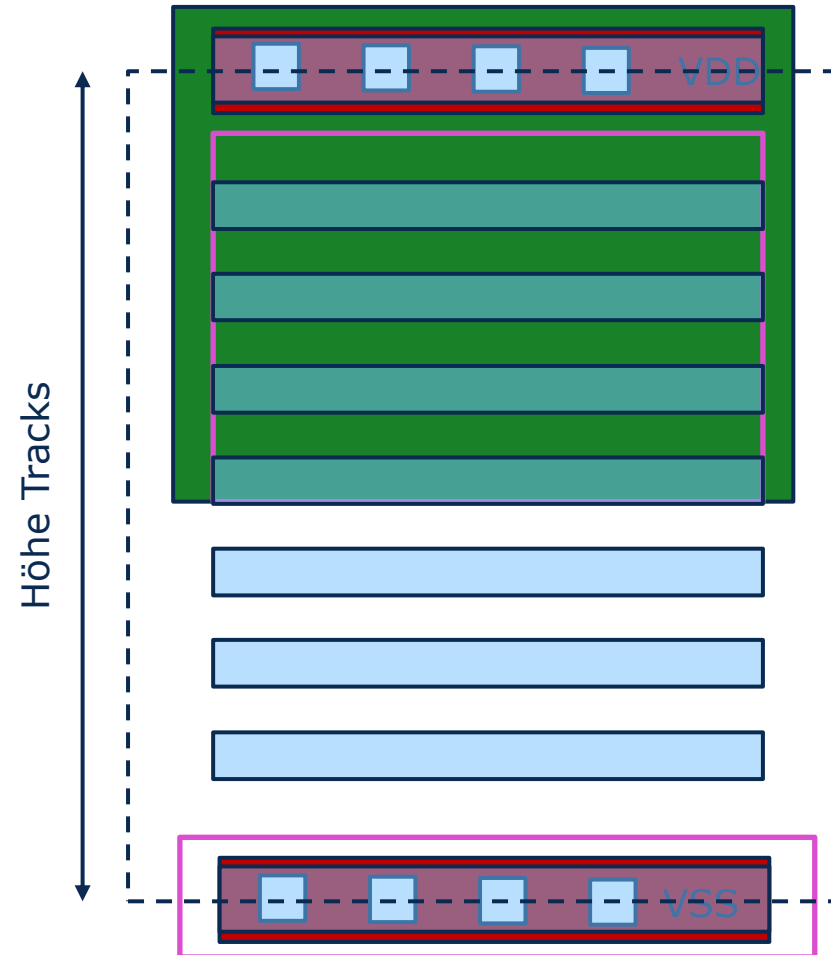


# Standardzellenbibliotheken

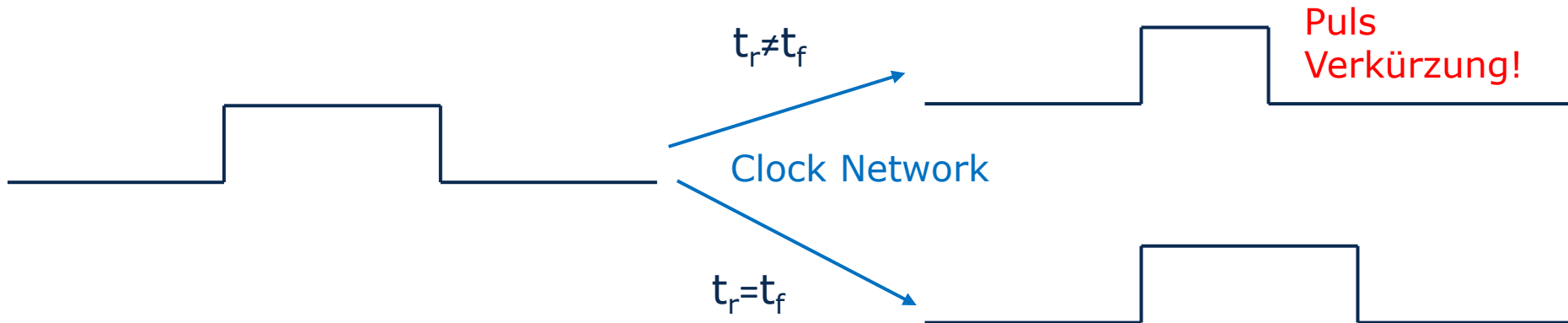
- Basis-IP für die Realisierung von digitalen Schaltungen in einer CMOS Technologie
- Grundlage für die automatisierte Synthese und Place& Route Implementierung
  
- Beinhalten:
  - Logikgatter Gatter in mehreren Treiberstärken
  - Clock Zellen in mehreren Treiberstärken
  - Sequentielle Zellen (FlipFlop, Latch, Clock Gate)
  - Kapazitive Filler Zellen
  - Tie1, Tie0
- Spezielle Power Management Blöcke
  - Power Switches
  - Level-Shifter
  - Isolation Cells
  - Retention Cells

- Layout Architektur der Standardzelle
  - Zellhöhe in Tracks (Anzahl Metal 1 Minimum Pitches), z.B.
    - 8T → Ultra-high density
    - 9T → High Density
    - 12T → High Performance
- Metal Layer für Power Rails, z.B. M1, M2
- Wannen- und Substratkontakte
  - In den Zellen
  - Nicht In den Zellen, spezielle Tap Zellen nötig
- Definition der Wannenhöhe
  - Maximale Fingerweite  $W_{f,max,n}$  NMOS
  - Maximale Fingerweite  $W_{f,max,p}$  PMOS
- Die Standardzellenarchitektur ist **Grundlage** für die Technologieentwicklung (10nm, 7nm, ...)



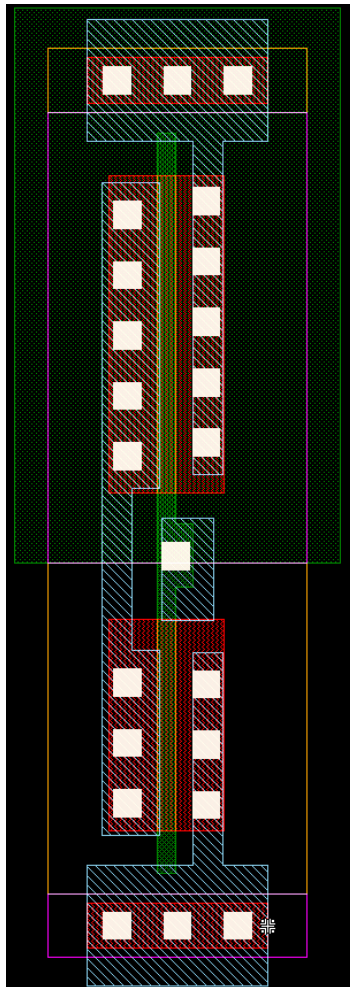
- Layout erlaubt maximale Fingerweite  $W_{f,max,n} + W_{f,max,p}$
- Gatelänge  $L=L_{min} \rightarrow$  minimaler  $R_n$  und  $R_p$ , kleine  $C_{in}$
- „Gate-Length Adder Option“ ( $L+\Delta L$ ) zur Reduktion von Leckströmen möglich für Low-Power Anwendungen
- Wie groß ist  $k=W_p/W_n$  zu wählen?
- Separate Optimierung für Datenpfad Zellen und Clock Zellen

- Ein Timing Pfad besteht aus einer Folge von rising- und falling-edge Delays
- → Sizing von **Datenpfad Gattern** für minimale Summe aus  $t_{d,r}$  und  $t_{d,f}$ 
  - $R'_p = q \cdot R'_n$  (relative Treiberstärke von NMOS und PMOS)
  - $W_p = k \cdot W_n$
  - $(t_{d,r} + t_{d,f}) \approx \ln(2) \cdot (R_p + R_n) \cdot (A \cdot C_{in,1})$
  - $(t_{d,r} + t_{d,f}) \approx \ln(2) \cdot \left( R'_p \cdot \frac{L}{W_p} + R'_n \cdot \frac{L}{W_n} \right) \cdot (A \cdot C'(W_n + W_p) \cdot L)$
  - $(t_{d,r} + t_{d,f}) \approx \ln(2) \cdot A \cdot L^2 \cdot C' \cdot R'_n \cdot \left( \frac{q}{k} + 1 \right) \cdot (1 + k)$
  - $\frac{d(t_{d,r} + t_{d,f})}{dk} = \ln(2) \cdot A \cdot L^2 \cdot C' \cdot R'_n \cdot \left( \frac{-q}{k^2} + 1 \right) = 0$
  - $k = \sqrt{q}$



- Sizing von **Clock Zellen** für gleiches  $t_{d,r}$  und  $t_{d,f}$ 
  - $t_{d,r} = t_{d,f}$
  - $\left(R'_p \cdot \frac{L}{W_p}\right) \cdot (A \cdot C'(W_n + W_p) \cdot L) = \left(R'_n \cdot \frac{L}{W_n}\right) \cdot (A \cdot C'(W_n + W_p) \cdot L)$
  - $\left(q \cdot R'_n \cdot \frac{L}{k \cdot W_n}\right) = \left(R'_n \cdot \frac{L}{W_n}\right)$
  - **$k = q$**

• INVX2



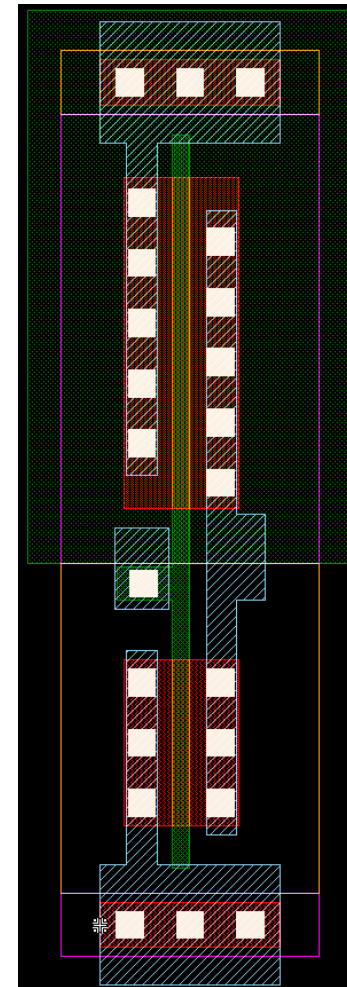
$$R'_p/R'_n = q \approx 2$$

$$W_p = 1,05 \mu\text{m}$$

$$W_n = 0,70 \mu\text{m}$$

$$k \approx 1.5$$

• CINVX2

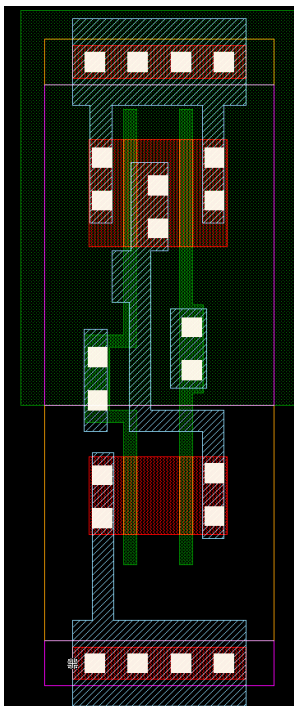


$$W_p = 1,1 \mu\text{m}$$

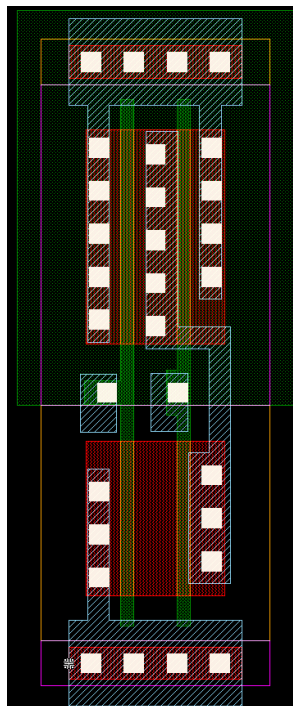
$$W_n = 0,55 \mu\text{m}$$

$$k \approx 2$$

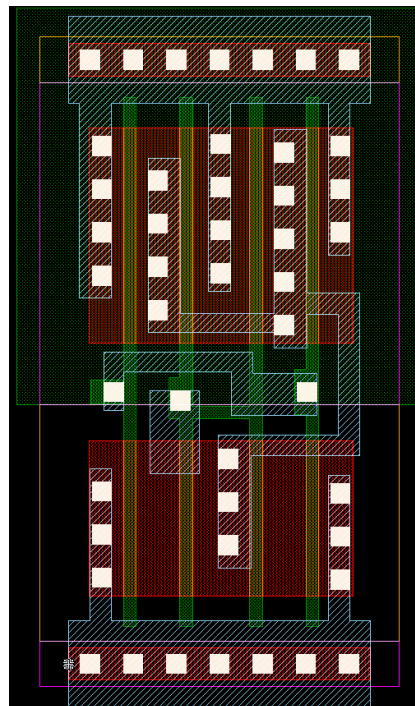
- Realisierung von Standardzellen gleicher Logikfunktion in mehreren Treiberstärken  $X$  durch Parallelschaltung von Transistorfingern ( $W \uparrow$ )



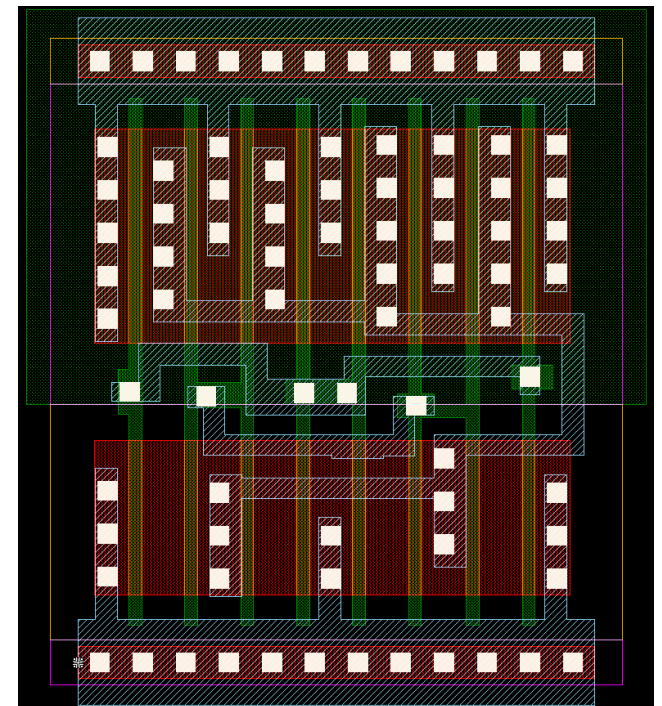
NAND2X1



NAND2X2



NAND2X4



NAND2X8



- HPSNLIB: Standardzellenbibliothek für 28nm CMOS
- Namenskonventionen:
  - Library: `<libname>_<tech>_<height>_<vt>`
    - Beispiel: `hpsnlib_g28_9t_RVT`
    - 4 verschiedene VT-Optionen, mit je 3 Gate-Length Adder Optionen → 12 Libraries
  - Zellen: `H_<height>_<vt>_<function>X<strenght>`
    - Beispiele: `H_9T_RVT_NAND3X2`
    - ca. 240 Zellen
- Zusätzliche Spezial-Libraries für:
  - Power Management (Level-Shifter, Power Switches)
  - Multi-Bit FlipFlops und Multi-Height Zellen

- Eine Standardzellenbibliothek sollte folgende Zellen enthalten:
  - Kombinatorische Logik invertiert und nicht invertiert
  - Kombinatorische Logik mit mehreren Eingängen
  - Verschiedene Treiberstärken der Logikzellen
  - Mehr Treiberstärken für Inverter und Buffer
  - Komplexe Logikzellen

## Beispiele:

```
H_9T_RVT_NAND2X2  
H_9T_RVT_AND2X2  
H_9T_RVT_NOR2X2  
H_9T_RVT_OR2X2
```

```
H_9T_RVT_NAND3X2  
H_9T_RVT_AND3X2
```

```
H_9T_RVT_AND2X1  
H_9T_RVT_AND2X2  
H_9T_RVT_AND2X4
```

```
H_9T_RVT_BUFX1  
H_9T_RVT_BUFX2  
H_9T_RVT_BUFX4  
H_9T_RVT_BUFX6  
H_9T_RVT_BUFX8  
H_9T_RVT_BUFX10  
H_9T_RVT_BUFX12  
H_9T_RVT_BUFX16  
H_9T_RVT_BUFX20
```

```
H_9T_RVT_AOI13X2  
H_9T_RVT_OAI222X1  
H_9T_RVT_MUX4X2  
H_9T_RVT_HAX2  
H_9T_RVT_FAX2  
H_9T_RVT_XOR2X2
```

- Clock Zellen und Clock Gates
- FlipFlops und Latches mit Posedge/Negedge Set/Reset Q/QN Varianten
- Scan-Varianten für FlipFlops (Design for Test)
- Tie1 und Tie0 Zellen
- Layout Filler Zellen mit und ohne Stütz-Kapazitäten

```
H_9T_RVT_CBUF2X2  
H_9T_RVT_CIN2X2  
H_9T_RVT_CDLY1  
H_9T_RVT_CNAND2X2  
H_9T_RVT_CGATEPX2
```

```
H_9T_RVT_DFNTQX2  
H_9T_RVT_DFPTQX2
```

```
H_9T_RVT_TIEH  
H_9T_RVT_TIEL
```

```
H_9T_RVT_FILL1  
H_9T_RVT_FILL2  
H_9T_RVT_FILLCAP4  
H_9T_RVT_FILLCAP8
```

```
H_9T_RVT_DFNQX2  
H_9T_RVT_DFPQX2  
H_9T_RVT_DFPQNX2  
H_9T_RVT_DFPRQX2  
H_9T_RVT_DFPSQX2
```

## • Vorteile großer Libraries:

- Mehr Optimierungspotential
- → Geringere Chipfläche
- → Geringere Verlustleistung
- → Höhere Taktfrequenz

## • Nachteile großer Libraries:

- Längere Toollaufzeiten
- Längere Laufzeit der Charakterisierung

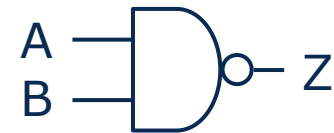
- Modellierung der Library im Liberty Format (.lib)
- <https://www.OpensourceLiberty.org>
- Ein .lib File beinhaltet Informationen zu den der Standardzellen, u.a.:
  - Interface (Pin Liste)
  - Funktion (Formel oder Wahrheitstabelle)
  - Eingangs- und Ausgangskapazitäten
  - Timing Arcs und Constraints
  - Power
  - Chipfläche

## PVT Corner

```
library(hpsnlib_g28_9t_RVT_wc_0d90V_125C) {  
  delay_model : table_lookup ;  
  library_features(report_delay_calculation, report_power_calculation);  
  time_unit : 1ns ;  
  voltage_unit : 1V ;  
  current_unit : 1uA ;  
  capacitive_load_unit(1, ff);  
  pulling_resistance_unit : 1kohm ;  
  leakage_power_unit : 1nW ;  
  input_threshold_pct_fall : 50 ;  
  input_threshold_pct_rise : 50 ;  
  output_threshold_pct_fall : 50 ;  
  output_threshold_pct_rise : 50 ;  
  slew_derate_from_library : 0.5 ;  
  slew_lower_threshold_pct_fall : 30 ;  
  slew_lower_threshold_pct_rise : 30 ;  
  slew_upper_threshold_pct_fall : 70 ;  
  slew_upper_threshold_pct_rise : 70 ;  
  ...  
}
```

```
...
cell(H_9T_RVT_NAND2X2) {
  area : 0.702 ; ← Fläche
  leakage_power() {
    related_pg_pin : "VDD" ;
    when : "!A&!B" ;
    value : " XXX " ;
  }
  leakage_power() {
    related_pg_pin : "VDD" ;
    when : "!A&B" ;
    value : " XXX " ;
  }
  leakage_power() {
    related_pg_pin : "VDD" ;
    when : "A&!B" ;
    value : " XXX " ;
  }
  leakage_power() {
    related_pg_pin : "VDD" ;
    when : "A&B" ;
    value : " XXX " ;
  }
}
```

**Leakage Power**



```

...
pin(A) {
  capacitance : 1.36194 ;
  direction : input ;
  max_transition : 1.2799999999999998 ;
  related_ground_pin : VSS ;
  related_power_pin : VDD ;
  related_bias_pin : "VDDNW" ;

  internal_power() {
    related_pg_pin : "VDD" ; Internal Power, wenn kein Toggle an Z
    when : "!B" ;

    fall_power(pwr_tin_9) {
      index_1("0.006, 0.018, 0.036, 0.068, 0.116, 0.21, \ ←  $t_{tt,A}$ 
              0.396, 0.712, 1.28");
      values(" XXX, XXX, XXX, XXX, XXX, XXX, XXX, XXX, XXX ");
    }

    rise_power(pwr_tin_9) {
      index_1("0.006, 0.018, 0.036, 0.068, 0.116, 0.21, \ ←  $t_{tt,A}$ 
              0.396, 0.712, 1.28");
      values("-XXX, -XXX, -XXX, -XXX, -XXX, -XXX, -XXX, -XXX, -XXX ");
    }
  }
}
...

```

```

...
pin(Z) {
  direction : output ;
  function : "(!A)|(!B)" ;
  max_capacitance : 43.199999999999996 ;
  output_voltage : default ;
  related_ground_pin : VSS ;
  related_power_pin : VDD ;
  internal_power() {
    related_pg_pin : "VDD" ;
    related_pin : "A" ;
    when : "B" ;
    fall_power(pwr_tin_oload_9x8) {
      index_1("0.006, 0.018, 0.036, 0.068, 0.116, 0.21, \
              0.396, 0.712, 1.28");
      index_2("0.3, 0.9, 1.8, 3.6, 7.2, 14.4, 28.8, 48");
      values(
        " XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        "- XXX, - XXX, - XXX, - XXX , - XXX, - XXX, - XXX, - XXX ", \
        " XXX,  XXX,  XXX,  XXX ,  XXX,  XXX,  XXX,  XXX ");
    }
    rise_power(pwr_tin_oload_9x8) {
      index_1("0.006, 0.018, 0.036, 0.068, 0.116, 0.21, \
              0.396, 0.712, 1.28");
      index_2("0.3, 0.9, 1.8, 3.6, 7.2, 14.4, 28.8, 48");
      values( ... );
    }
  }
}

```

$t_{tt,A}$

$C_{load,Z}$

**Internal Power,  
wenn Toggle an  
Ausgang Z**



```

...
timing() {
  related_pin : "A" ;
  timing_sense : negative_unate ;
  timing_type : combinational ;

  cell_fall(tmg_ntin_oload_9x8) {
    index_1("0.006, 0.018, 0.036, 0.068, 0.116, 0.21, \
0.396, 0.712, 1.28");
    index_2("0.3, 0.9, 1.8, 3.6, 7.2, 14.4, 28.8, 48");
    values("XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ", \
"XXX, XXX , XXX , XXX, XXX, XXX, XXX, XXX ");
  }
  cell_rise(tmg_ntin_oload_9x8) {
    ...
  }
  fall_transition(tmg_ntin_oload_9x8) {
    ...
  }
  rise_transition(tmg_ntin_oload_9x8) {
    ...
  }
}
...

```

- Definition von **Timing Arcs an Output Pins**,
- **Beispiele:**
  - `related_pin` : zugehöriger Input Pin
  - `timing_sense`:
    - `positive_unate` : rise → rise, fall → fall
    - `negative_unate`: rise → fall, fall → rise
    - `non_unate`: beliebig
  - `timing_type`:
    - `combinational`: Kombinatorische Logik
    - `rising edge`: Clk→Q bei posedge FlipFlop
    - `falling edge`: Clk→Q bei negedge FlipFlop
- Definition von **Constraints an Input Pins**
- **Beispiele:**
  - `setup_falling`, `hold_rising`, `min_pulse_width`

- Ablegen von Timing Werten in Lookup Tabellen (**LUT**) (1D, 2D):
- Beispiele:
  - $t_{d,A \rightarrow Z, rise} = \mathbf{LUT}(t_{tt,A, fall}, C_{load,Z})$
  - $t_{setup\_rising,D, fall} = \mathbf{LUT}(t_{tt,CLK, rise}, t_{tt,D, fall})$
- Table Lookup und Interpolation durch das Design Tool

	$C_{load}$ Vektor			
$t_{tt}$ Vektor				

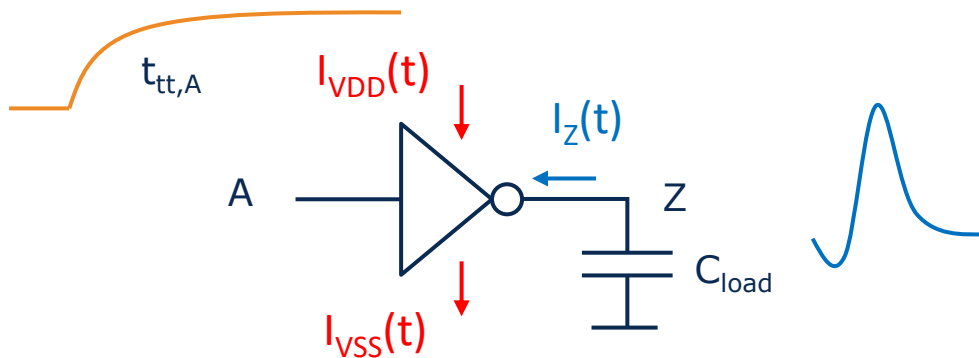
skalare Werte

## • Vorteile :

- Berücksichtigt nicht-lineare Delay Effekte
- Kompaktes Library Format
- Wird von vielen Tools unterstützt

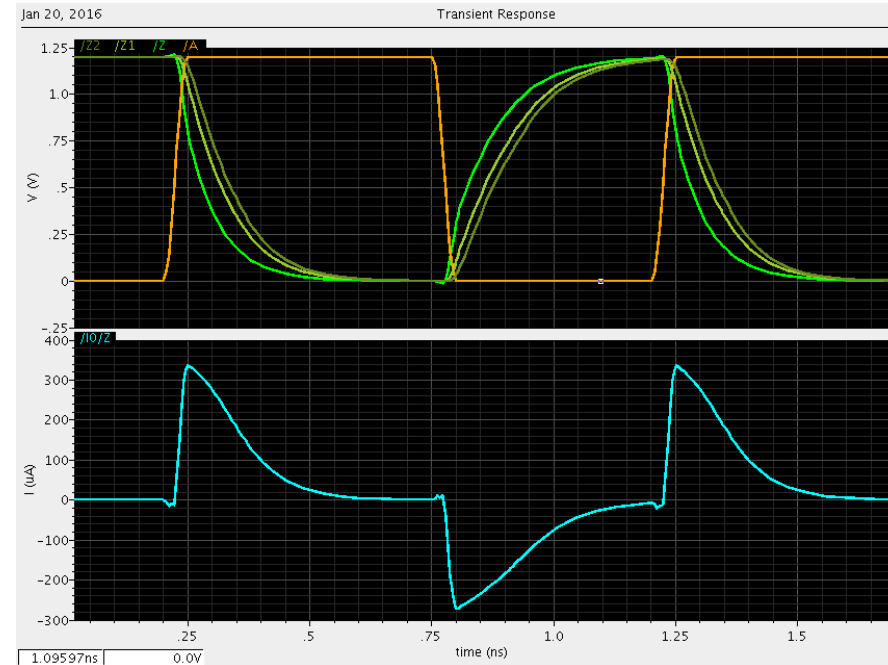
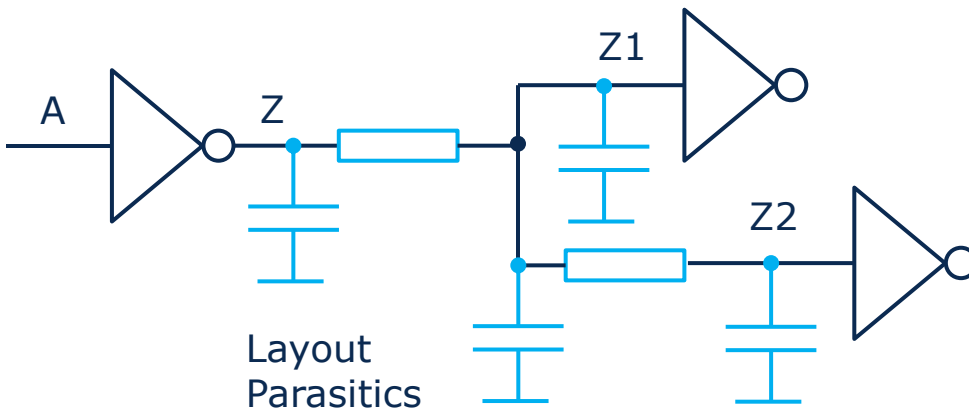
## • Nachteile :

- Ungenau bei Delay Beeinflussung durch parasitäre RC Effekte der Leitungen
- Crosstalk Analyse Schwierig



	$C_{load}$ Vektor			
$t_{tt}$ Vektor				

- Modellierung des Gatterausgangs als **Stromquelle**
- Ablegen von Stromkurven  $I_Z(t)$  Lookup Tabellen (**LUT**) (1D, 2D)
- Table Lookup und Interpolation durch das Design Tool
- Berechnung der Signalverläufe auf den Leitungen und am Empfänger durch Netzwerksimulation
- Zusätzlich Speichern der Stromkurven  $I_{VDD}(t)$  und  $I_{VSS}(t)$  → akkurate Power- und IR-Drop-Analyse

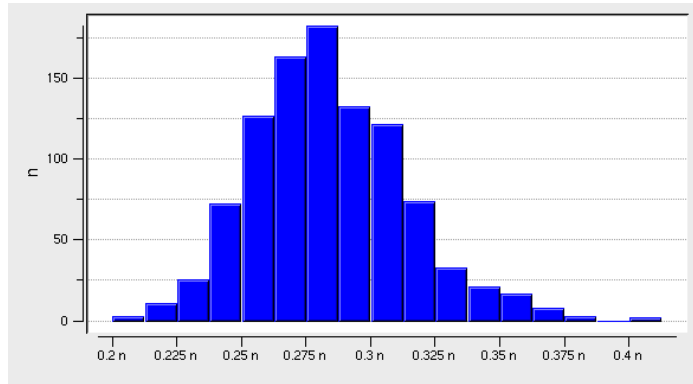


## • Vorteile :

- Akkurate Berechnung der Timings in parasitären RC Netzwerken
- Genaue Berücksichtigung von Crosstalk möglich
- Akkurate zeitliche Modellierung der Verlustleistung → IR Drop

## • Nachteile :

- Aufwändigere Charakterisierung
- Große Library Files
- Längere Toollaufzeiten
- Nicht von allen Tools unterstützt



- Modellierung von Timing Variabilität
- Separate Berücksichtigung von Early/Late Variabilität
- Möglichkeiten:
  - OCV (On-chip variation)
  - AOCV (Advanced on-chip Variation)
  - Liberty Variation Format (LVF) erlaubt die Integration in die .lib Modelle

```

ocv_sigma_cell_rise(tmg_ntin_oload_7x6) {
    sigma_type : "early" ;
    index_1("0.072, 0.288, 0.72, 1.44,
    index_2("0.001, 0.004, 0.016, 0.04
    values("0.103387, 0.118936, 0.1872
            "0.10373, 0.119775, 0.19111
            "0.105332, 0.120836, 0.1882
            "0.108998, 0.12192, 0.19456
            "0.113178, 0.142619, 0.1917
            "0.176103, 0.172004, 0.2493
            "0.297117, 0.301633, 0.3341
    }

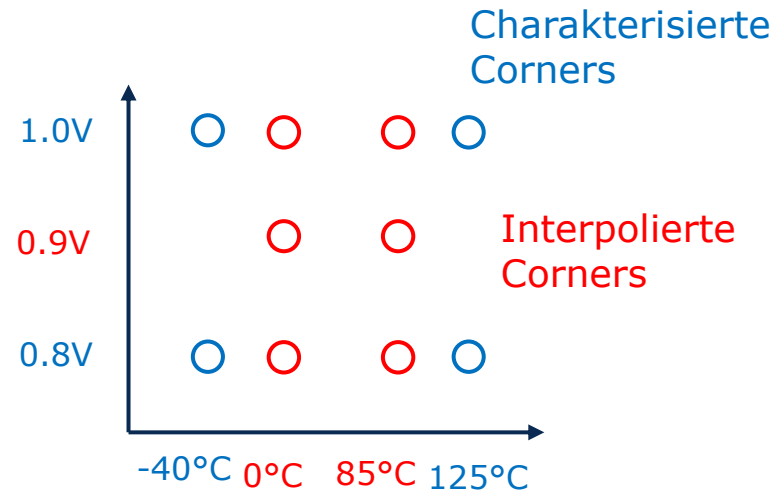
ocv_sigma_cell_rise(tmg_ntin_oload_7x6) {
    sigma_type : "late" ;
    index_1("0.072, 0.288, 0.72, 1.44,
    index_2("0.001, 0.004, 0.016, 0.04
    values("0.133355, 0.151417, 0.2344
            "0.133635, 0.152405, 0.2408
            "0.136092, 0.15377, 0.23600
            "0.138176, 0.156242, 0.2436
            "0.142256, 0.171527, 0.2365
            "0.195079, 0.188664, 0.2894
            "0.302766, 0.309019, 0.3573
    }
    
```

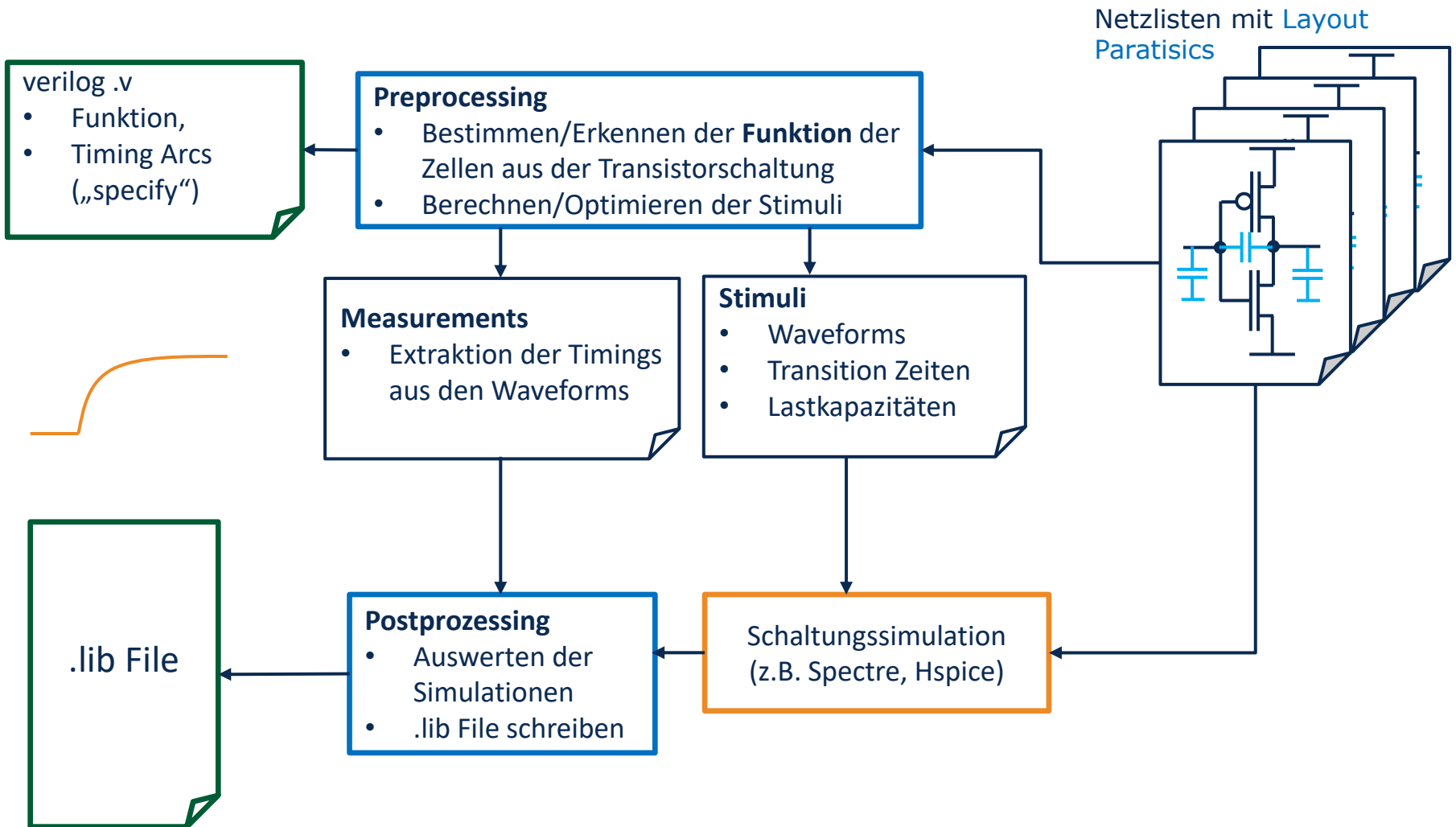
- Eine PVT Corner pro .lib File
- Charakterisierung der Library für ein bestimmtes Set an Corners
- Designs mit verschiedenen Versorgungsspannungen oder DVFS/AVFS benötigen spezifische Corners
- Lib-Scaling über Spannung und Temperatur
- Interpolation der .lib File Daten

```

voltage_map(VDDNW, 0.9);
voltage_map(VDD, 0.9);
voltage_map(VSS, 0);

operating_conditions(wc_0d90V_125C) {
    temperature : 125 ;
    voltage : 0.9 ;
}
    
```







View	File	Inhalt	verwendet für
Datasheet	.pdf	„Lesbare“ Beschreibung der Zellen, Funktion, Timing und Power (grob)	Library Auswahl
Liberty File	.lib	Verhalten, Timing (qualitativ und quantitativ), Power, Fläche	Synthese, Place&Route, STA, Power Analyse, EM-IR Analyse
DFT View		Fehlermodelle für DFT	Synthese DFT Insertion, Testpattern Generierung
Layout Abstract	.lef	Layout Pins, Metal Blockages, Shape	Place&Route
Verhaltensmodell	.v	Verhalten, Timing (qualitativ)	Gate-Level Simulation
Layout	.gds	Komplettes Layout	Chip DRC, LVS, <b>Tape-out</b>
LVS Netlist	.cdl	Transistorschaltung	LVS
Extracted Netlist	.cdl	Transistorschaltung und Layout Parasitics	(Re-)Charakterisierung