
Logic Families

Note: Slides are (still) in German.

Übersicht

- CMOS
- Buffer
- Pass Gate Logik

- P-Last / Pseudo-NMOS
- Dynamische Logik

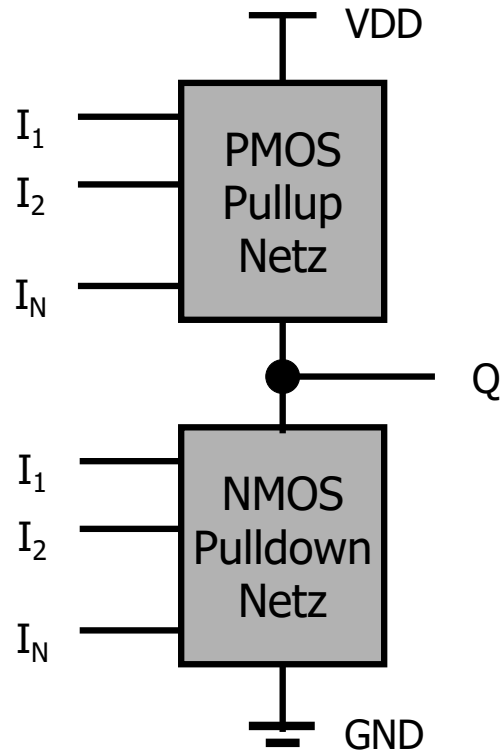
- DCVS
- Differentielle Logik CML

- ECL, TTL

CMOS

CMOS = Complementary MOS

- Die Einsen am Ausgang werden durch ein 'pullup' Netzwerk von Transistoren erzeugt, die Nullen durch ein 'pulldown' Netzwerk.
- Pullup und Pulldown sind DUALE Netzwerke, sie enthalten je N PMOS / NMOS Transistoren.
- **Ein CMOS Gatter mit N Eingängen hat 2N Transistoren**
- **CMOS ist sehr robust und daher die meistbenutzte Logikfamilie**



Vorteile von CMOS:

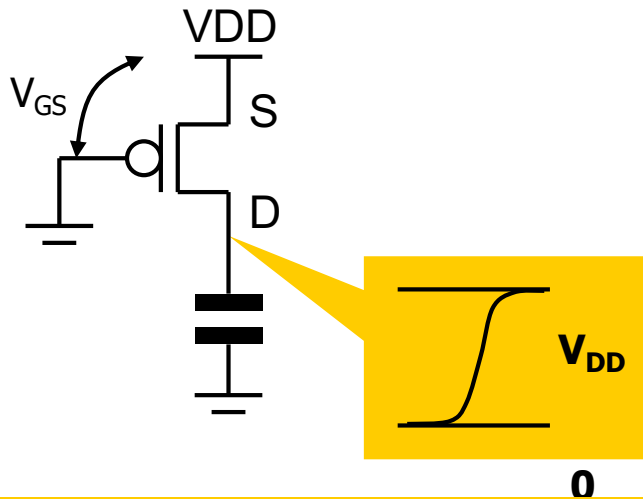
- volle Pegel (unabhängig von Versorgungsspannung, Parametervariationen und Transistorgrößen)
- daher i.d.R. hoher Störabstand
- kein DC Stromverbrauch
- Robust (Gatter funktionieren (logisch) immer, egal, wie die Transistoren dimensioniert sind)
- Migration zu neuen Technologien einfach

Nachteile:

- nicht die schnellste Logikfamilie
- für einige Schaltungen ungünstig (z.B. MUX)
- Layouts oft nicht sehr kompakt
- hohe Strom-Transienten während des Schaltens (Querstrom!)

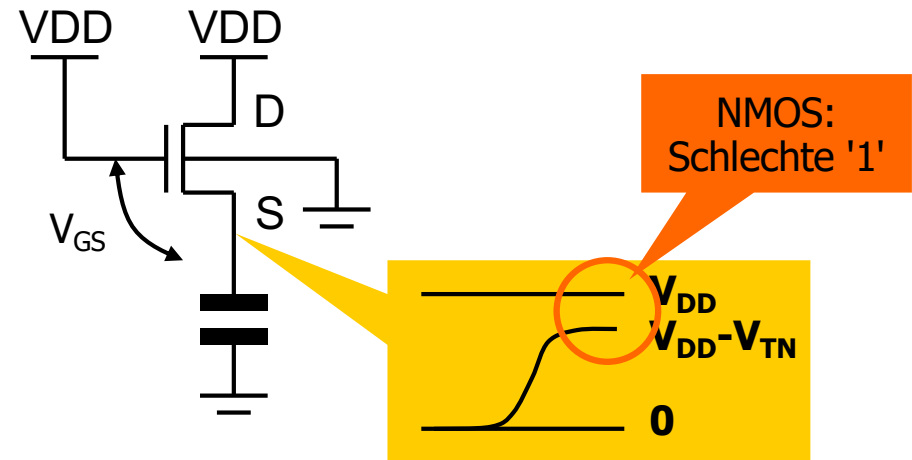
Warum PMOS für pullup ?

- PMOS als Pullup ist ok:



V_{GS} ist unabhängig von der Ausgangsspannung
⇒ der PMOS bleibt abgeschaltet

- NMOS als Pullup geht nicht:



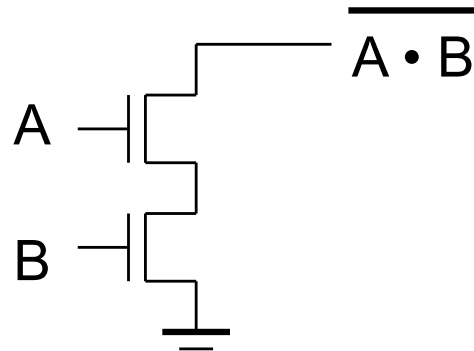
V_{GS} sinkt mit steigender Ausgangsspannung
⇒ der NMOS schaltet ab !

Die Ausgangsspannung steigt nur bis $V_{DD} - V_{TN}$.
Durch Substrateffekt sogar noch weniger !

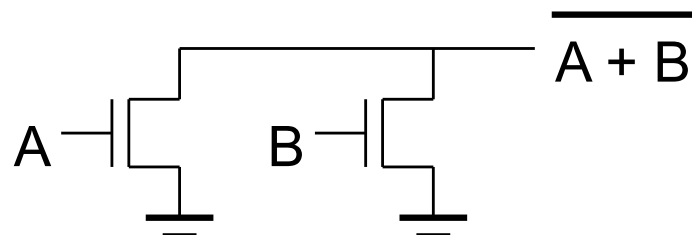
- Entsprechend ist ein PMOS als Pulldown ungeeignet.

Konstruktion des pulldown - Teils

- **Serielle** NMOS implementieren die **NAND** Funktion



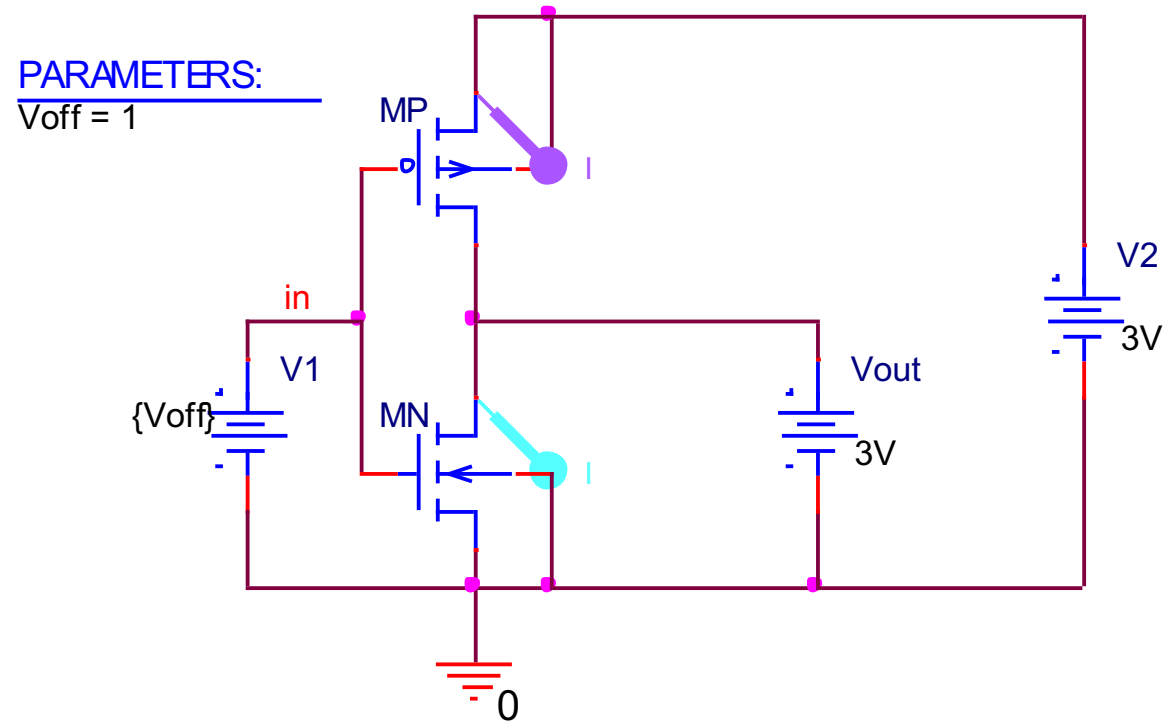
- **Parallelele** NMOS implementieren die **NOR** Funktion



- Damit lassen sich alle Funktionen der Form $Y = \text{NOT} (a + ab + cde + \dots)$ darstellen

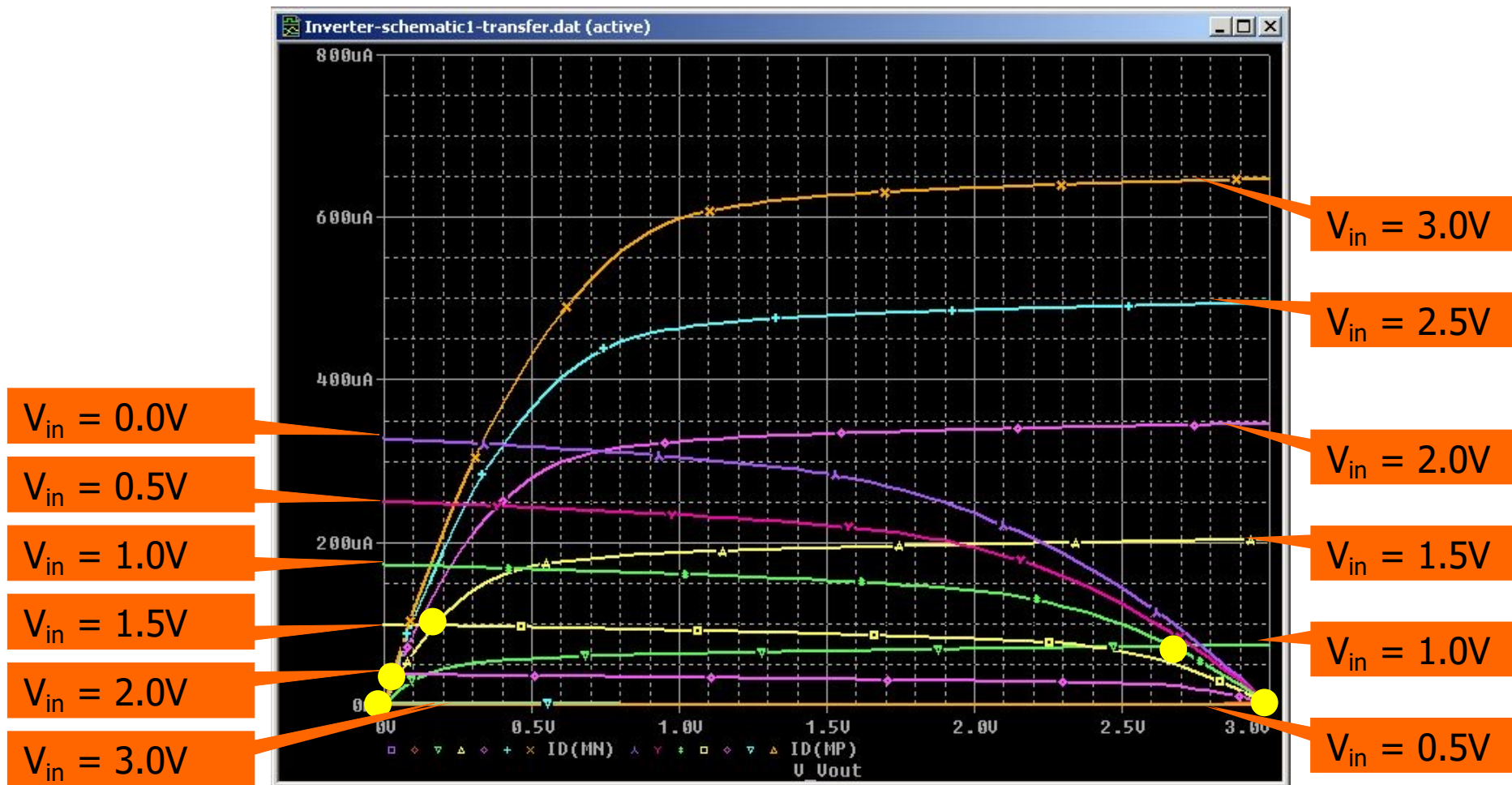
Der CMOS Inverter

- Schaltbild zur Simulation der nachfolgenden Seiten:



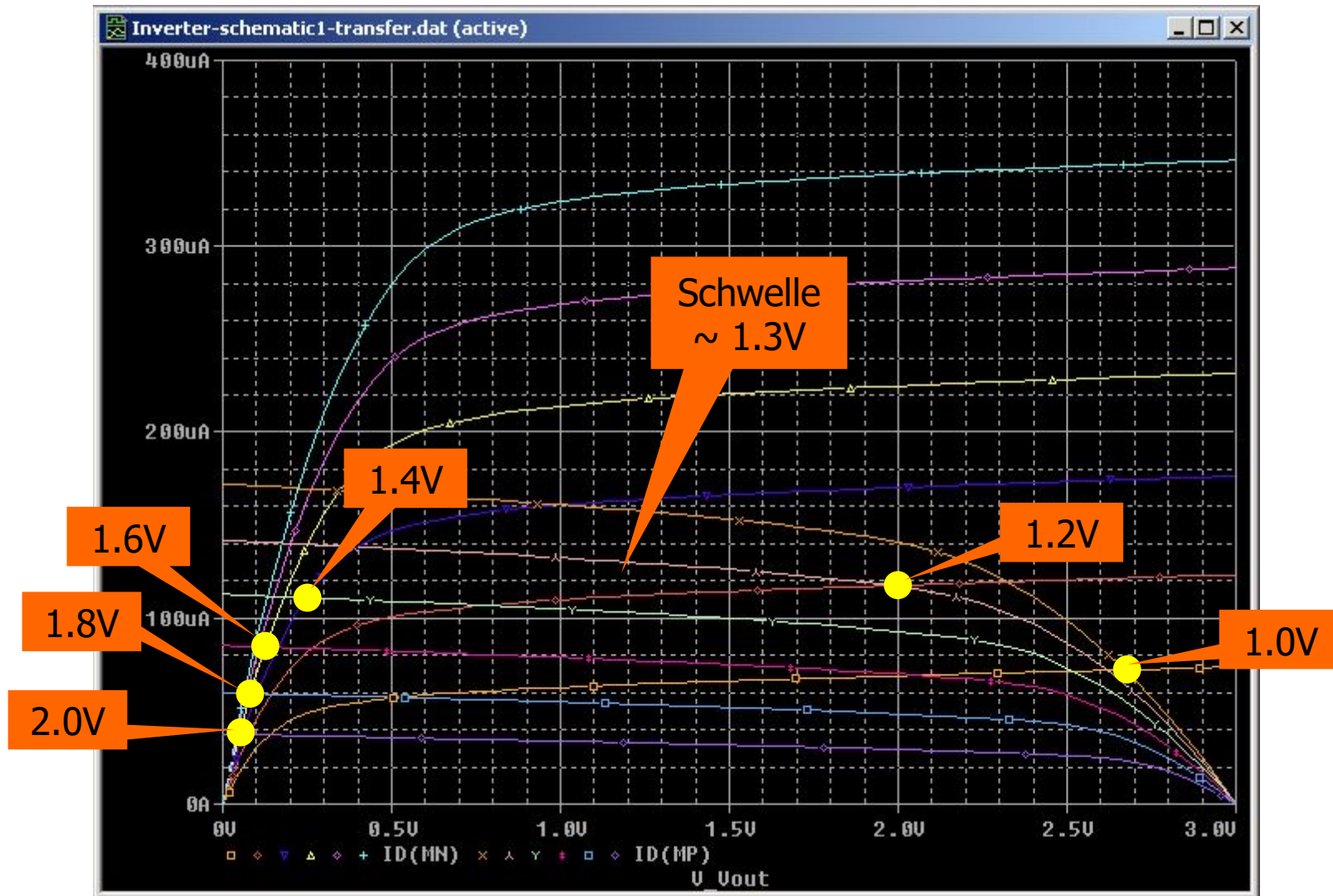
CMOS Inverter Transferkennlinie

- Entstehung der Transferkennlinie:
 - Für jede Eingangsspannung werden die Ausgangskennlinien des NMOS und des PMOS gezeichnet.
 - Der Ausgang nimmt die Spannung am Schnittpunkt an ($I_{D,N} = I_{D,P}$).
- Simulation mit $V_{DD}=3V$, $(W/L)_N = (W/L)_P = 0.8\mu\text{m}/0.25\mu\text{m}$, $V_{in} = 0..3V$ in Schritten von $0.5V$

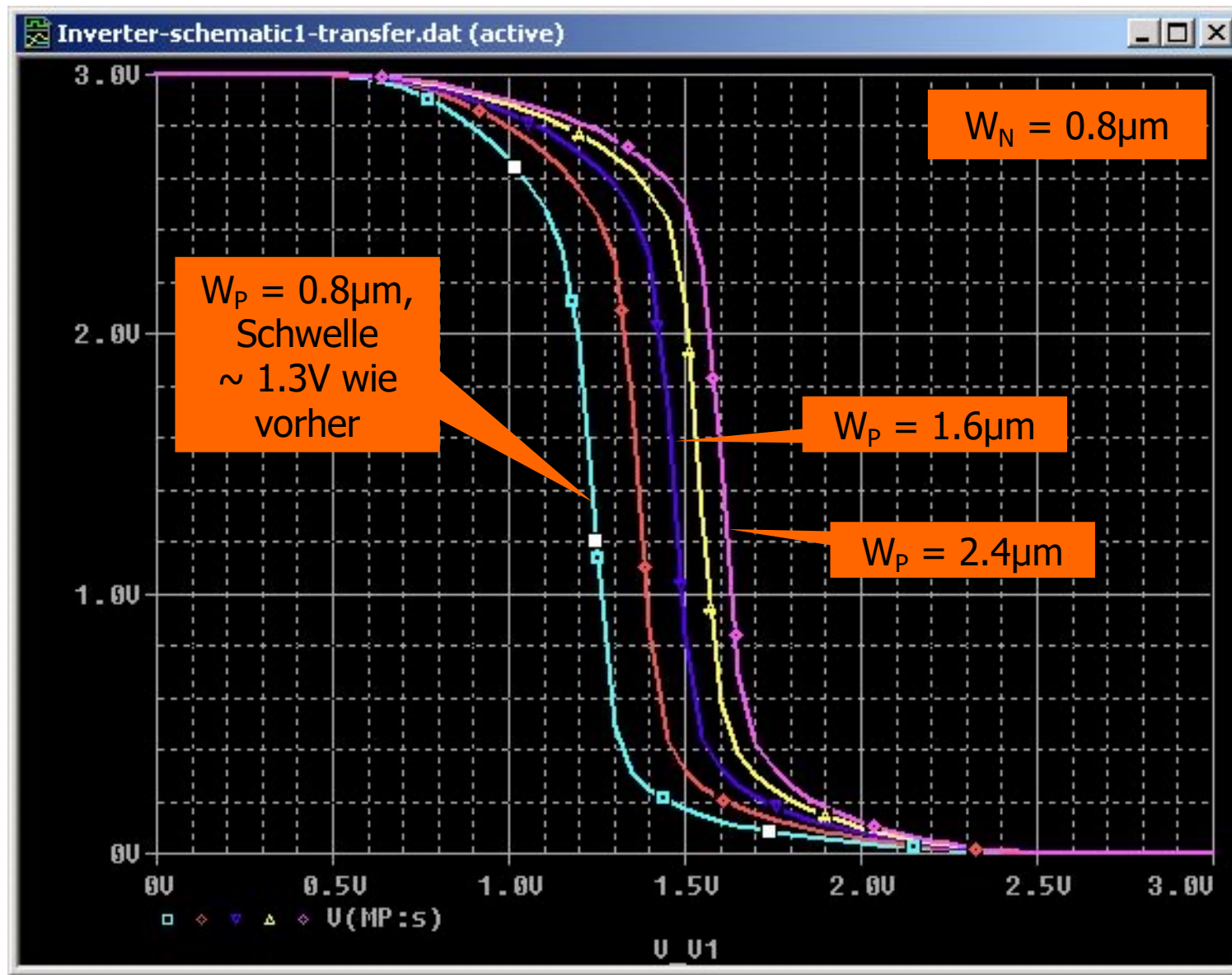


CMOS Inverter Transferkennlinie

- Details in der Schaltregion: Eingang **1.0V...2.0V** in Schritten von **0.2 V**



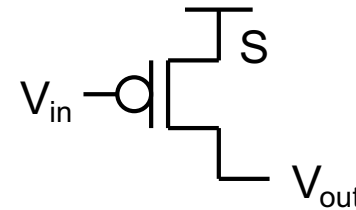
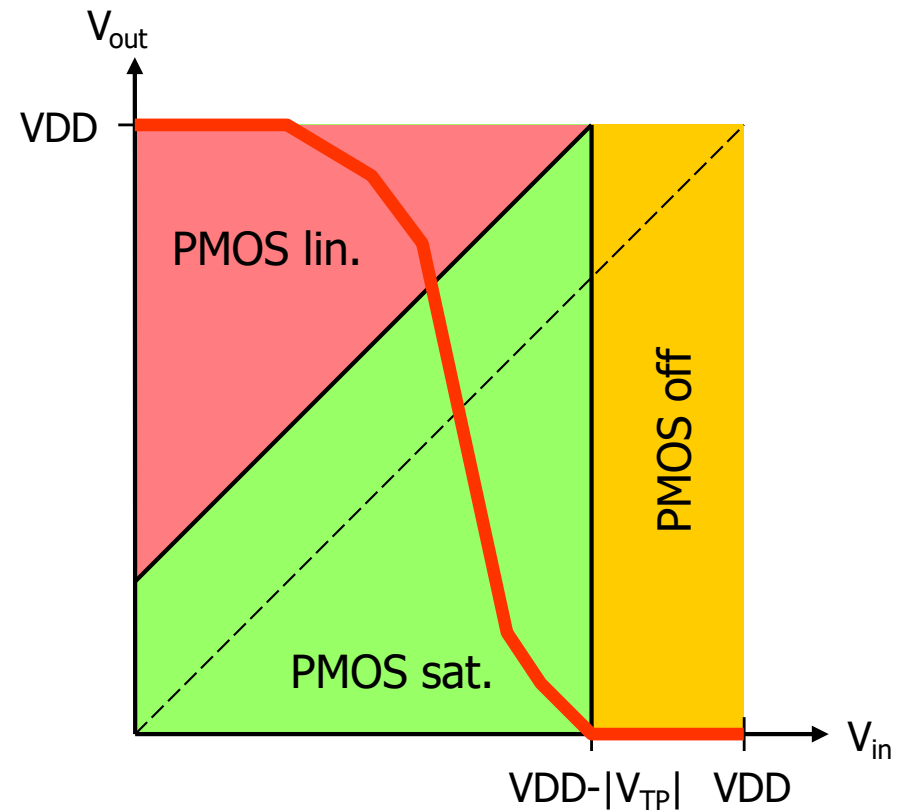
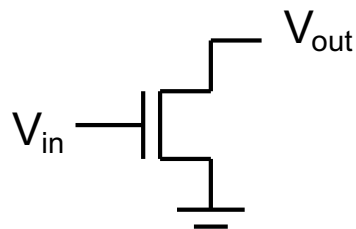
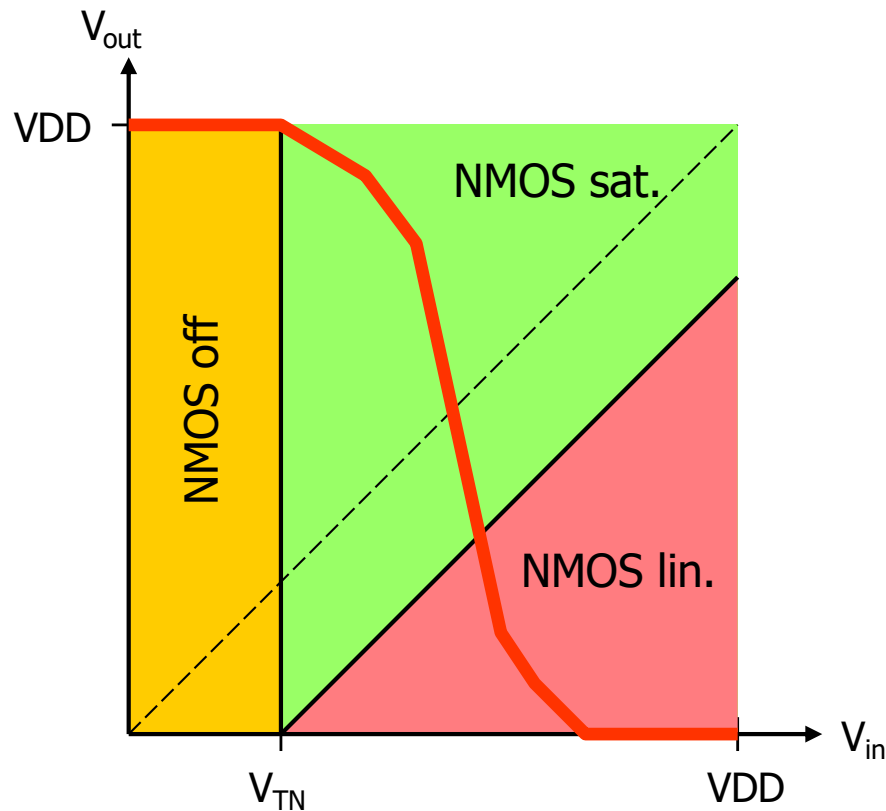
Test: Transfer Charakteristik (DC sweep)



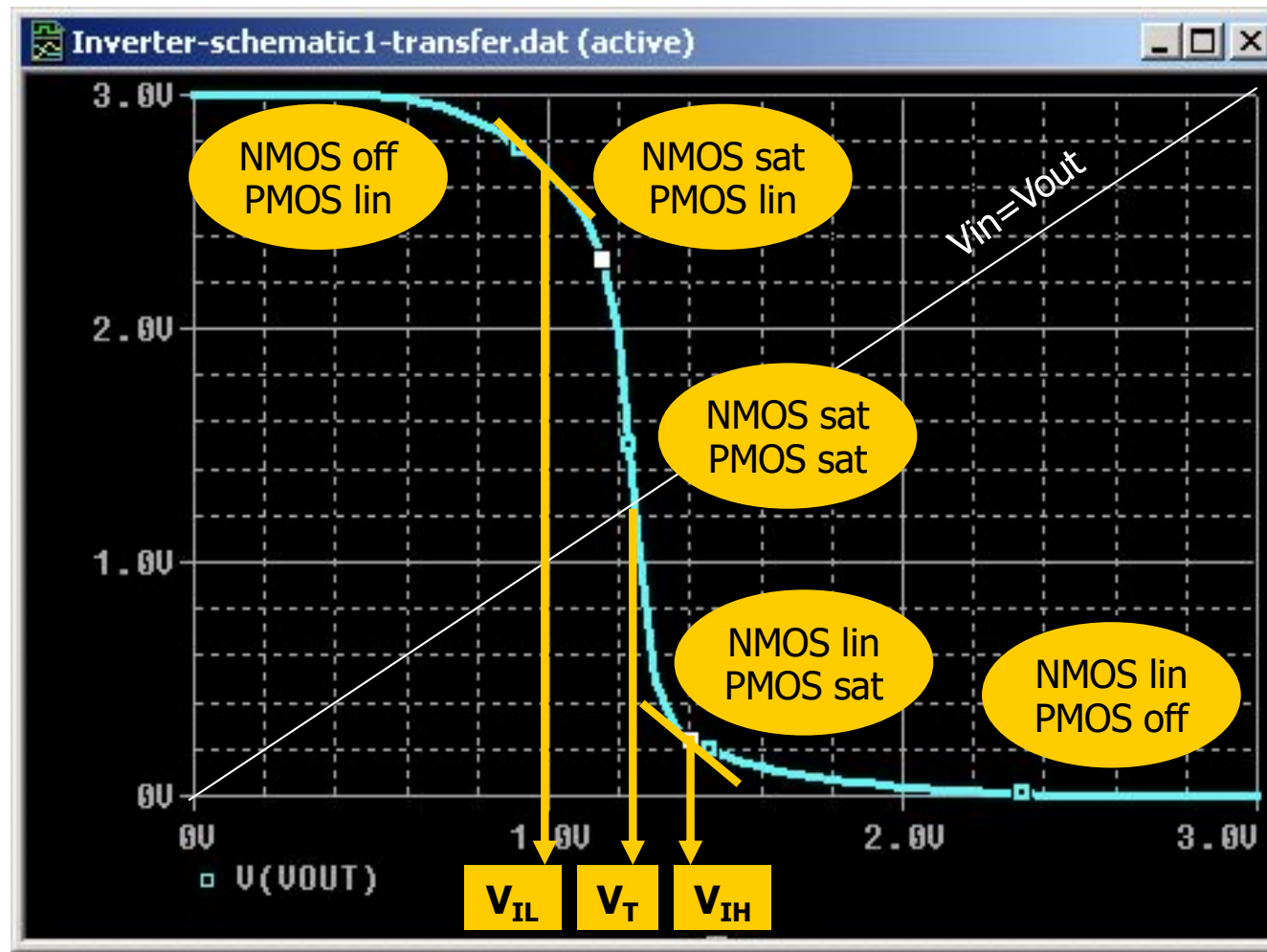
Bereiche in der Transferkennlinie

- Für den NMOS ist $V_{GS} = V_{in}$, $V_{DS} = V_{out}$.
- Daher muß in Sättigung $V_{out} > V_{in} - V_{TN}$ sein.

- Für den PMOS ist $V_{GS} = VDD - V_{in}$, $V_{DS} = VDD - V_{out}$.
- Das NMOS Diagramm ist daher um 180° gedreht.



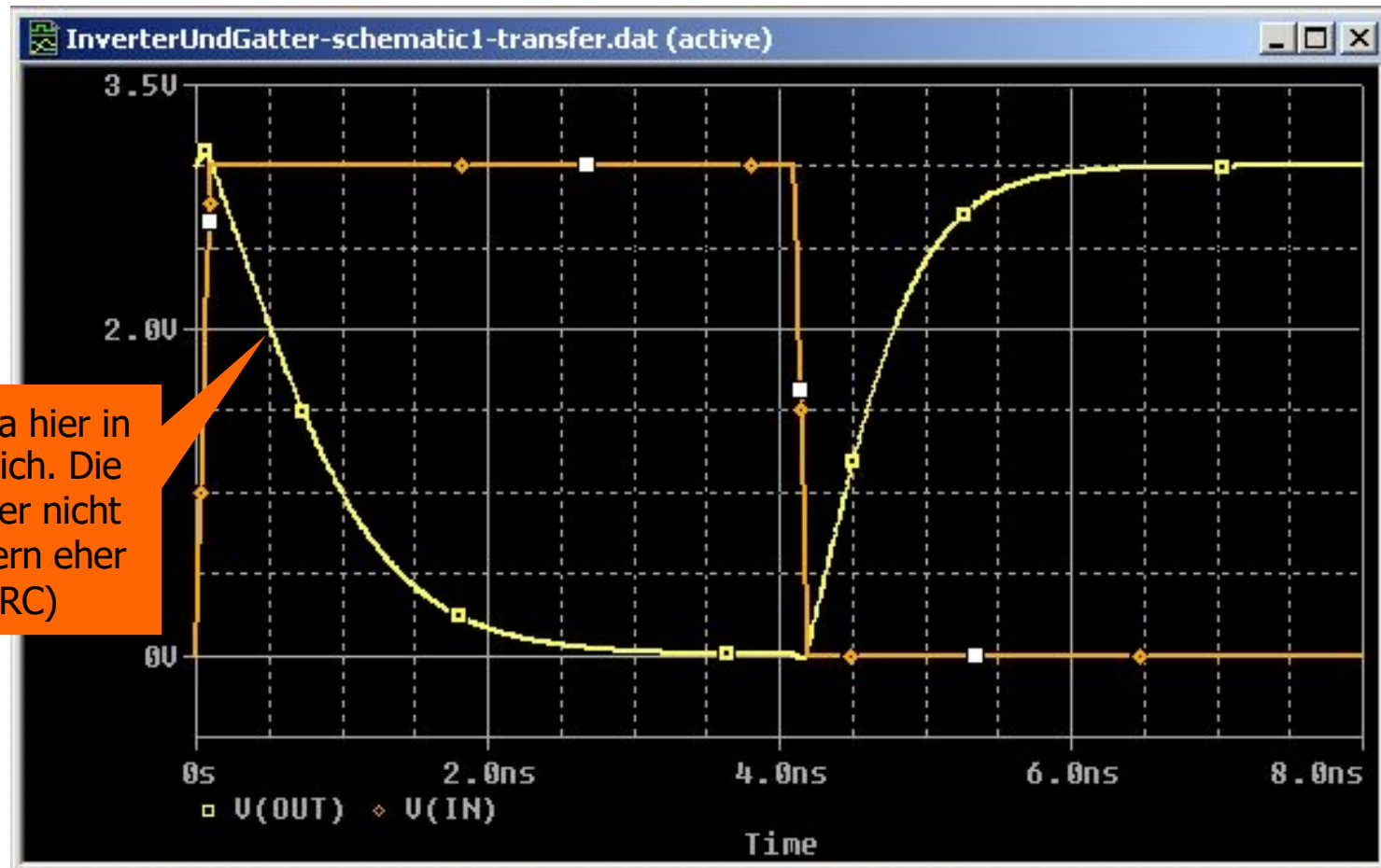
Bereiche in der Transferkennlinie



- Die Schwellen sind etwa da, wo Steigungen = -1 sind.
- Beachte: Dies gilt für einen 'statischen' DC-Sweep. Beim dynamischen Schalten sind die Transistoren schneller im linearen Bereich, weil die Gatespannung dann VDD bzw. 0 ist.

Transientenanalyse

- Simulation eines Inverters (Modelle 'nmos_simple' und 'pmos_simple' mit V_T je 1V) mit 100fF Last.
- $V_{dsat} = V_{GS} - V_T = 3V - 1V = 2V$



Berechnung der Schwelle

- Unter Vernachlässigung des Ausgangswiderstands kann man die **Schwelle** V_M abschätzen:
Es muß gelten:

$$I_{\text{NMOS}} = I_{\text{PMOS}}$$
$$K_N (W/L)_N (V_M - V_{\text{TN}})^2 = K_P (W/L)_P (V_{\text{DD}} - V_M - |V_{\text{TP}}|)^2 \quad (\text{beide MOS in Sättigung!})$$

$$V_M = \frac{r(V_{\text{DD}} - |V_{\text{TP}}|) + V_{\text{TN}}}{1 + r} \quad \text{mit} \quad r = \sqrt{\frac{\beta_P}{\beta_N}} \quad \text{mit} \quad \beta = K \frac{W}{L}$$

- Für $V_{\text{TN}} = |V_{\text{TP}}|$ wird die Forderung $V_M = V_{\text{DD}}/2$ durch $r=1$ erfüllt, d.h.

$$K_N (W/L)_N = K_P (W/L)_P$$

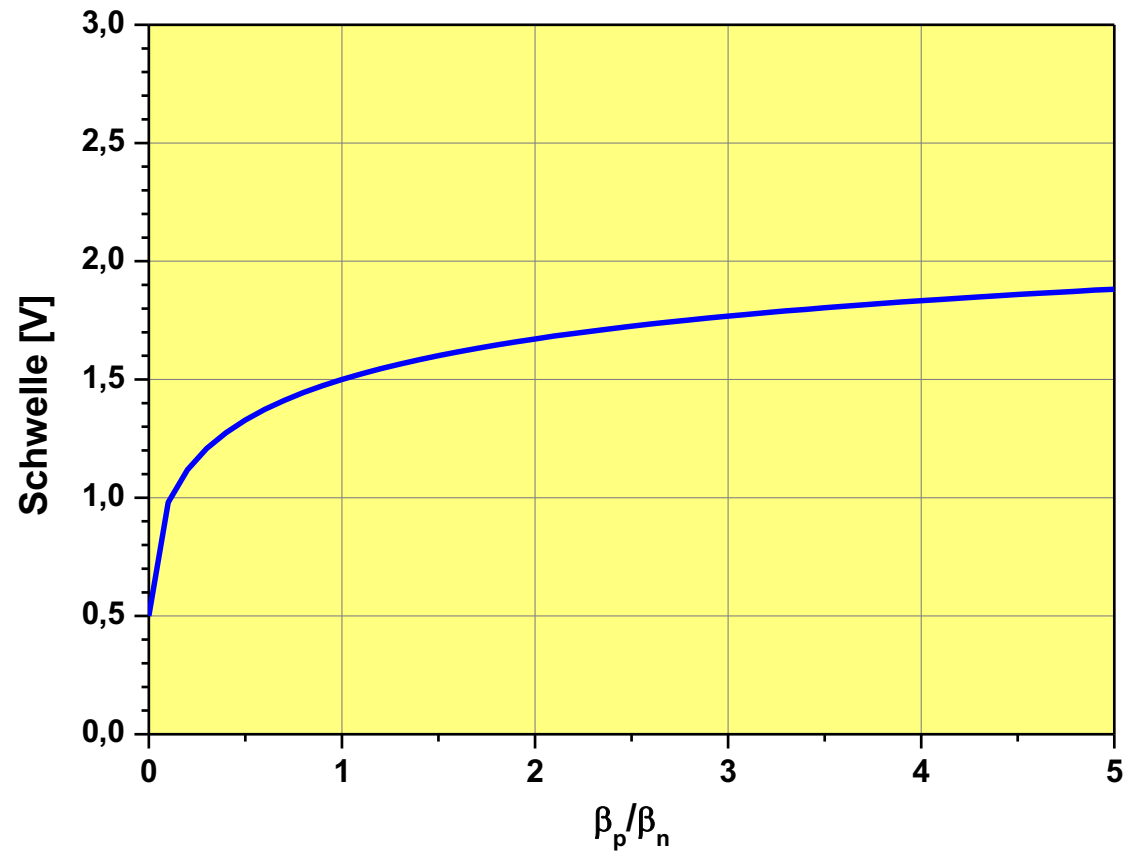
- Da normalerweise $K_N = 2 \dots 3 K_P$ folgt für eine Schwelle auf halber Versorgung:

$$(W/L)_P = 2 \dots 3 \times (W/L)_N$$

- In DSM hängt der Drainstrom nicht mehr quadratisch, sondern fast linear von der Gatespannung ab.
Am Ergebnis ändert das nichts!

Sensitivität der Schwelle

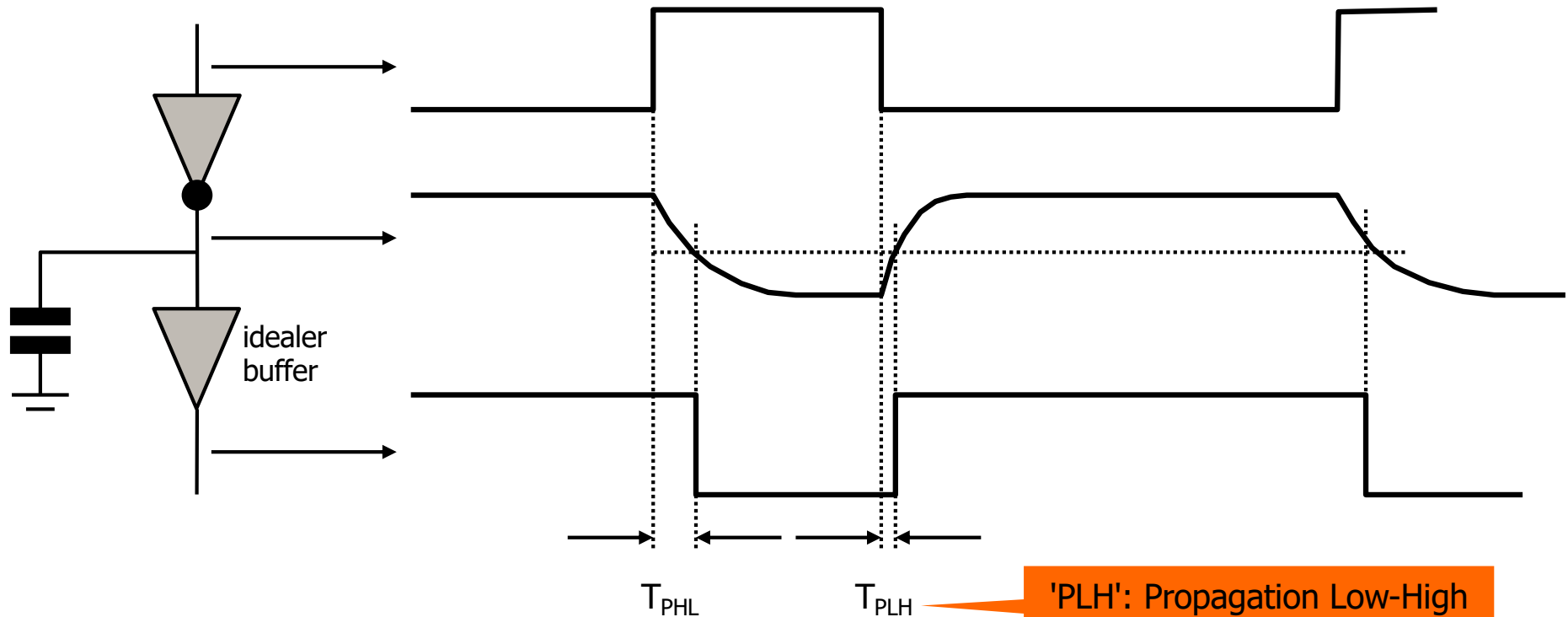
- Trägt man die (vereinfachende) Formel auf, so findet man ($V_{DD}=3.0V$, $V_{TN}=|V_{TP}| =0.5V$)



- Die Schwelle hängt recht unkritisch von k ab.
- Wichtigeres Kriterium für k sind Anstiegs- und Abfallzeiten.

Verzögerung = Durchlaufzeit

- Durch den verlangsamten Signalanstieg am Ausgang schalten nachfolgende Stufen etwas verzögert:



- Merke:
 - Durch Lastkapazitäten kommt es bei jedem Gatter zu Verzögerungen
 - Die Verzögerung steigt mit steigender kapazitiver Belastung des Ausgangs ('Fan-Out')
 - Fallende (T_{PHL}) und steigende (T_{PLH}) Flanke haben i.a. NICHT gleiche Verzögerung (sie werden durch unterschiedliche Transistoren gemacht!)
 - Die Verzögerungen und deren Lastabhängigkeiten müssen bei der Simulation genau berücksichtigt werden, denn sie limitieren letztlich die Geschwindigkeit einer Schaltung

Geschwindigkeit

- Hiermit ist meist Durchlaufzeit gemeint.
- Durchlaufzeit (t_{PLH} , t_{PHL}) entsteht durch die endliche Anstiegszeit des Ausgangssignals.
- Wird durch eine Last die Anstiegszeit langsamer, so wird effektiv die Durchlaufzeit länger.

Geschwindigkeit wird bestimmt durch:

- Die **Kapazität**, die umgeladen werden muß. Die wichtigsten sind
 - Drain-Gate Überlapp (Drain-Bulk und Drain-Gate sind klein, da MOS meist in Sättigung oder aus)
 - Drain-Dioden Sperrschichtkapazitäten
 - Leitungskapazitäten
 - Eingangskapazität der nächsten Stufe (Gate-Kapazitäten)
- Die **Transistorparameter**: K , die Schwelle und das W/L der Transistoren
- Die **Versorgungsspannung** (Höhere Versorgung \Rightarrow mehr Strom, aber auch höherer Hub)
- Die **Anstiegszeit des Eingangssignals**

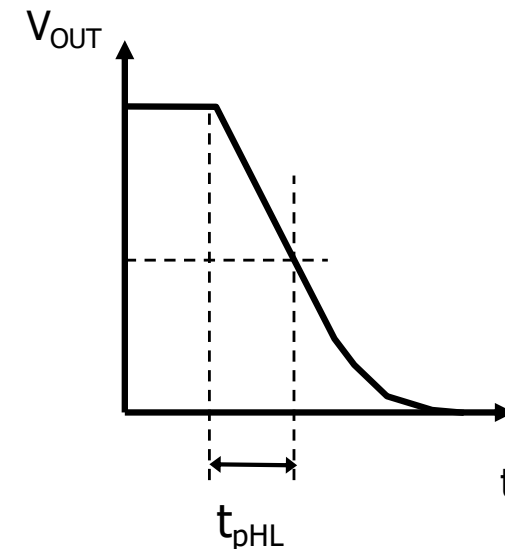
Faustregel:

- Wenn die **Lastkapazitäten dominieren** (Leitungen und C_{in} der nächsten Stufen, also z.B. bei hohem Fan-Out) braucht man **große Transistoren** ('Treiber', 'buffer')
- Ist die **Last klein**, so genügen **kleine Transistoren**

Merke: Minimale Gate Verzögerung für **0.25 μ m-0.35 μ m Technologien** ist etwa **50-100 ps**

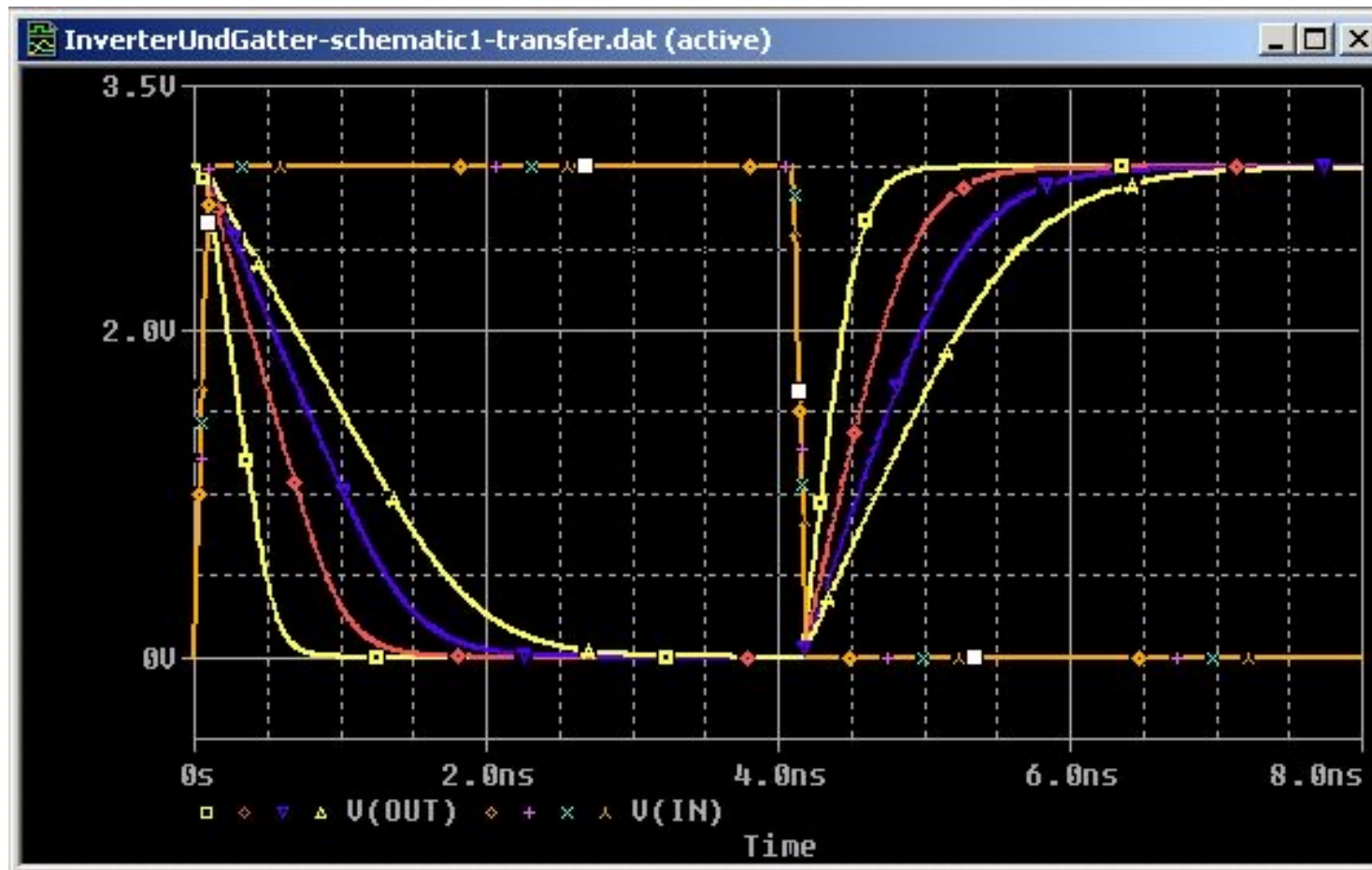
Geschwindigkeit

- Näherungsweise Berechnung:
 - Lastkapazität wird konstant angenommen (Real: Diodenkapazitäten sind spannungsabhängig)
 - Als Entladestrom wird der Sättigungsstrom genommen (Real: Strom nimmt ab, wenn MOS aus Sättigung kommt)
 - Quadratisches Modell wird angenommen (nicht mehr gültig in DSM. Transistoren liefern dort weniger Strom)
 - Ausgangswiderstand wird vernachlässigt (das ist hier ein kleiner Fehler)
- $I = (K/2)(W/L) (V_{DD}-V_{TN})^2$
- Linearer Abfall: $\Delta U / \Delta t = I / C_L$.
- $\Delta U = V_{DD} / 2$
- $t_{pHL} = C_L \times \Delta U / I = \frac{C_L V_{DD}}{K (W/L) (V_{DD}-V_{TN})^2}$
- Für $V_{TN} \ll V_{DD}$, also $V_{DD}-V_{TN} \sim V_{DD}$:
- $t_{pHL} \sim \frac{C_L}{K (W/L) V_{DD}}$
- Und entsprechend für die Anstiegszeit (mit dem PMOS)
- **Für gleiche Anstiegs- und Abfallzeiten muß $k_N = k_p$ sein**
- Die Erhöhung von K (via W bei minimalem L) erhöht auch die Diffusions- und Überlappkapazität. C/K wird dann konstant. Das setzt ein unteres Limit für die Verzögerung
- Erhöhung der Versorgungsspannung hilft, kostet aber viel Leistung (s.u.)



Transientensimulation

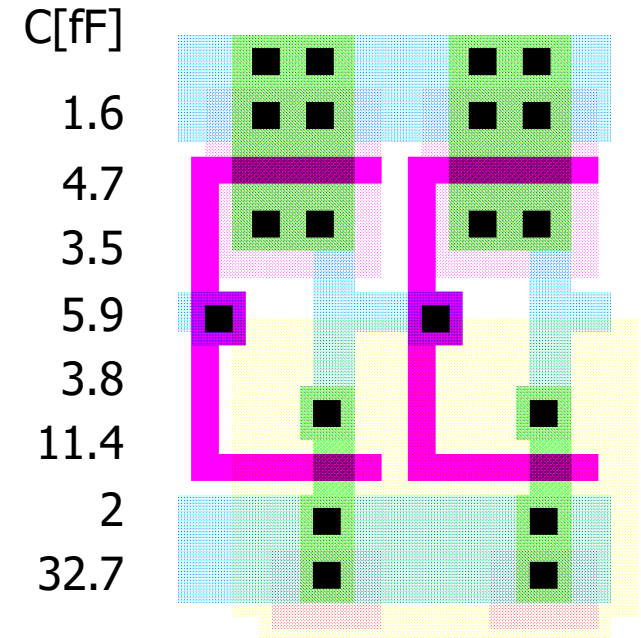
- PSPICE mit nmos25, pmos25.
- Externe Last von $C_L = 100\text{fF} \dots 400\text{ fF}$
- Man erkennt den linearen Anstieg der Verzögerung mit der Lastkapazität



Abschätzung der Kapazitäten

- Aus Rabey, 1.2µm Technologie, 5V Versorgung:

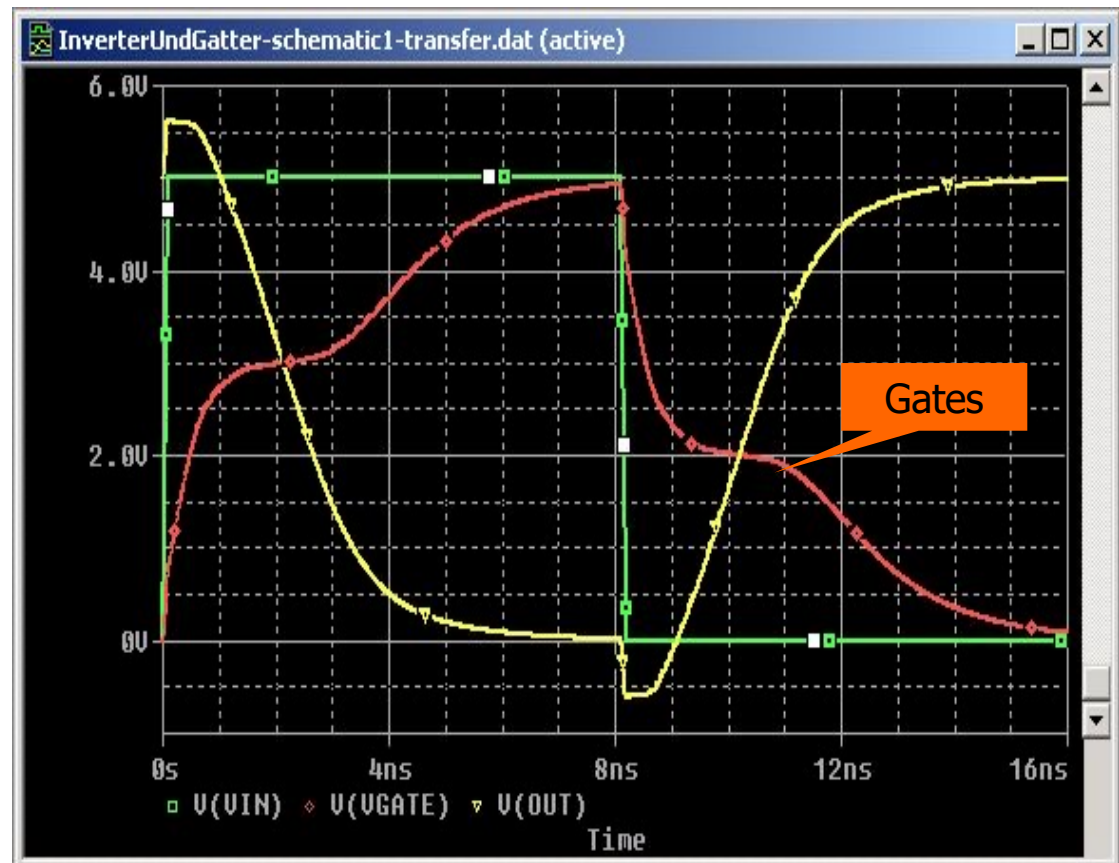
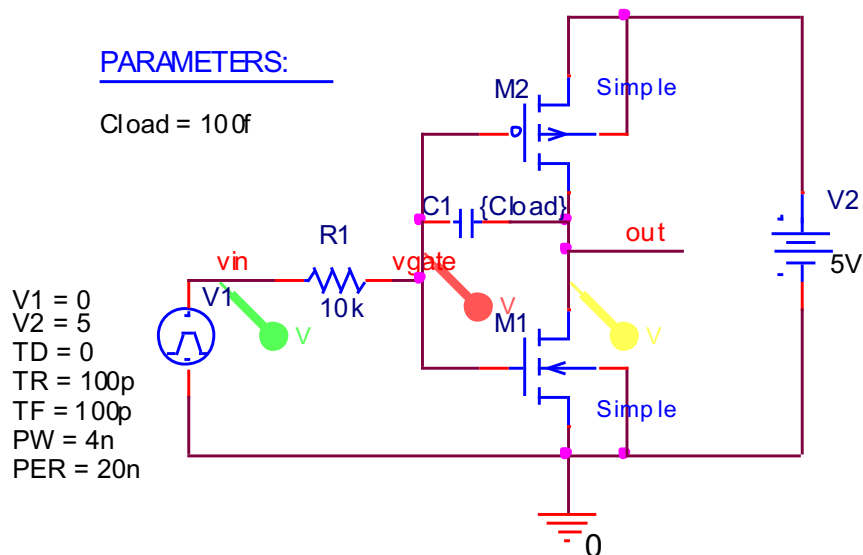
Capacitor	Expression	
C_{gd1}	$2 C_{GD0} W_n$	Überlapp
PMOS C_{gd2}	$2 C_{GD0} W_p$	
C_{db1}	$K_{eqn} (AD_n C_J + PD_n C_{JSW})$	Dioden
PMOS C_{db2}	$K_{eqp} (AD_p C_J + PD_p C_{JSW})$	
C_{g3}	$C_{ox} W_n L_n$	Gates
PMOS C_{g4}	$C_{ox} W_p L_p$	
C_w	From Extraction	Leitung
C_L	Σ	



- Der Miller-Effekt ist hier vernachlässigt (s. nächste Seite)
- Diese grobe Analyse zeigt:
 - Kapazitäten des Treibers und des angeschlossenen Inverters sind vergleichbar (bei Fanout = 1)
 - Der **PMOS trägt viel zur Kapazität bei** \Rightarrow ein **kleinerer PMOS** gibt eine kleinere mittlere Verzögerung (Mittelwert aus t_{pHL} und t_{pLH}). Man nimmt daher u.U. die leichte Reduktion im Störabstand und die Asymmetrie in t_r und t_f in Kauf
 - Bei kurzen Leitungen tragen diese wenig bei.
- In DSM werden die MOS-Kapazitäten kleiner, die der Leitungen sinken relativ wenig \Rightarrow Die korrekte Abschätzung (Extraktion) der Leitungskapazitäten wird wichtiger!

Miller Effekt

- Eine Kapazität zwischen Ausgang und Eingang (C_{GD}) trägt doppelt bei, da die beiden Kondensatorplatten ihre Potentiale in die entgegengesetzte Richtung ändern.
- Dem Gate muß man die Ladung $2 \times VDD \times C_{GD}$ zuführen
- In der folgenden Simulation sieht man, was während des Übergangs passiert (ein Serienwiderstand reduziert die Impedanz der Signalquelle, um den Effekt besser sichtbar zu machen):
- Der Miller-Effekt muß besonders bei analogen Schaltungen beachtet werden.



Leistungsverbrauch (1)

- Ist bei Hochleistungs-ICs eines der **dominierenden Probleme**
- Setzt sich aus mehreren Anteilen zusammen:

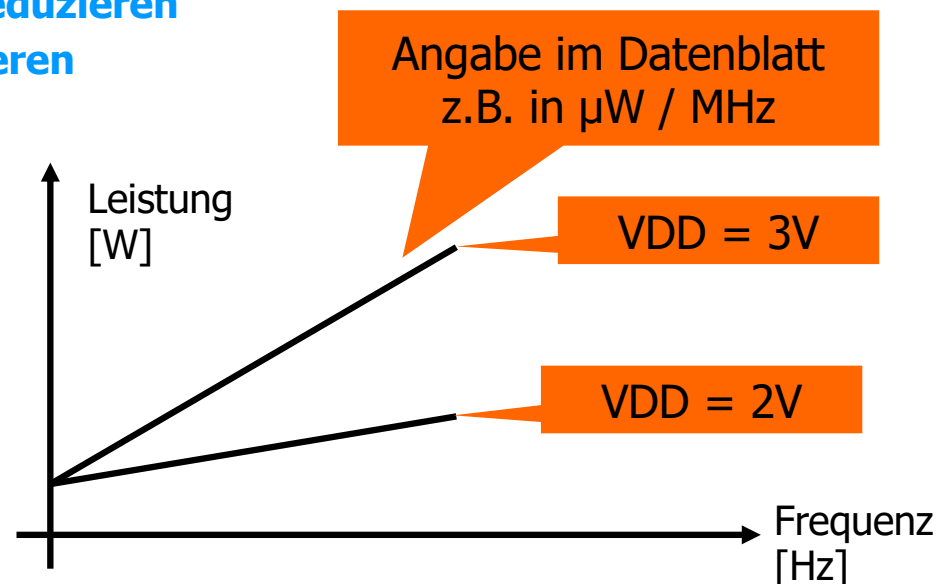
1. **Dynamischer Anteil** zum Auf- und Entladen von Last- und parasitären Kapazitäten

Die Ladung auf der Lastkapazität C_L ist $Q = VDD \times C_L$. Mittlerer Strom bei Frequenz f : $I = f \times Q$.
Die mittlere Leistung ist daher

$$P = I \times VDD = f \times VDD^2 \times C_L$$

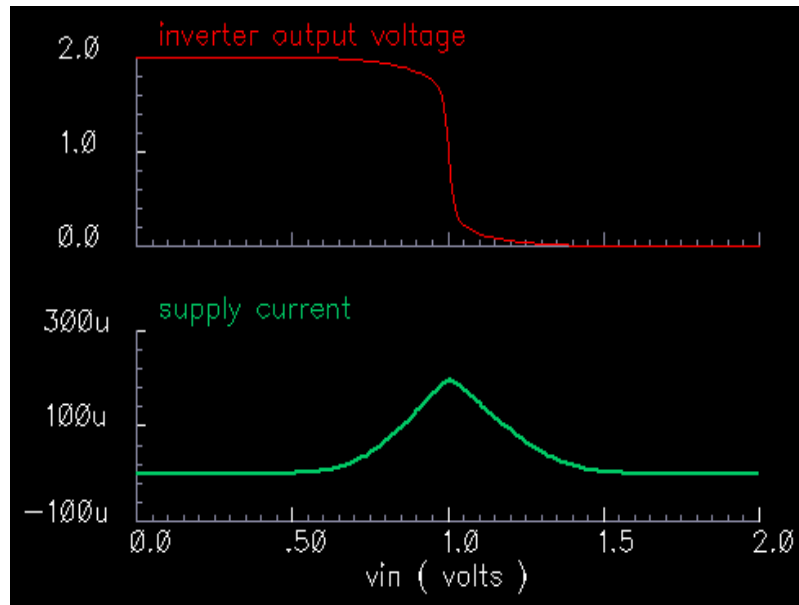
Während des Ladens wird $\frac{1}{2} \times C \times VDD^2$ im PMOS dissipiert, beim Entladen der Rest im NMOS.
Dieser Anteil hängt **nicht** von den Transistoreigenschaften ab.

- ⇒ **Lastkapazität reduzieren**
- ⇒ **Versorgungsspannung reduzieren**
- ⇒ **Betriebsfrequenz reduzieren**



Leistungsverbrauch (2)

2. **Dynamischer Kurzschlußstrom** zwischen VDD und Masse während des Schaltens



Man findet (mit $\beta = \beta_N = \beta_P$ und $V_T = V_{TN} = V_{TP}$)

$$P = (\beta/12)(V_{DD}-2V_T)^3(\tau/T)$$

mit $\tau =$ Anstiegszeit, $T =$ Periode

⇒ **Anstiegszeiten der Signale müssen kurz sein.** (Selbst für Signale mit niedriger Frequenz!)

2. Leckströme der Drain- und Sourcedioden. Hängt von deren **Größe** ab. $P = V_{DD} \times I_{leak}$

3. Leckstrom durch den Kanal (Subthreshold-Leakage, besonders bei niedrigen V_T in DSM Prozessen)

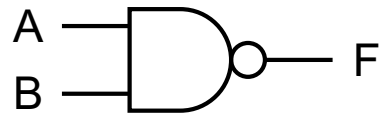
- Der Leistungsverbrauch eines Chips kann auch durch das Schaltungsdesign reduziert werden:
 - Weniger, optimierte Logik
 - Anzahl der Flanken (Pegelwechsel) reduzieren (Gray Code Zähler statt Binärzähler etc.)
 - Unbenutzte Teile abschalten

Wahl der Transistorgrößen

- $\beta = K \times W/L$ muß maximiert werden, daher $L = L_{\min}$
- Für die Wahl des Verhältnisses von β_p/β_n gibt es verschiedene Kriterien:
 - Maximaler Störabstand, also $V_T = V_{DD}/2 \quad \Rightarrow \beta_p = \beta_n$
 - gleiche Anstiegs- und Abfallzeiten $\Rightarrow \beta_p = \beta_n$
 - Minimale mittlere Verzögerung $\Rightarrow \beta_p < \beta_n$ (Kapazität durch PMOS kleiner halten!)
- Zusammen mit den Technologiekonstanten K_N und K_P legt man so W_P/W_N fest
- Der **Absolutwert** von W_N hängt von der zu treibenden Last ab.
Je größer die Last, desto größer sollten die W s sein.
Damit steigt aber auch die Eingangskapazität.
- Bei sehr großen Lasten sollte man daher lieber 'minimale' Inverter/Gatter benutzen und Buffer einfügen (s. später)

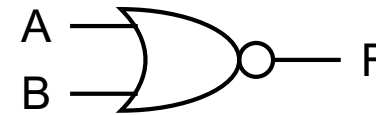
CMOS NAND / NOR

NAND2

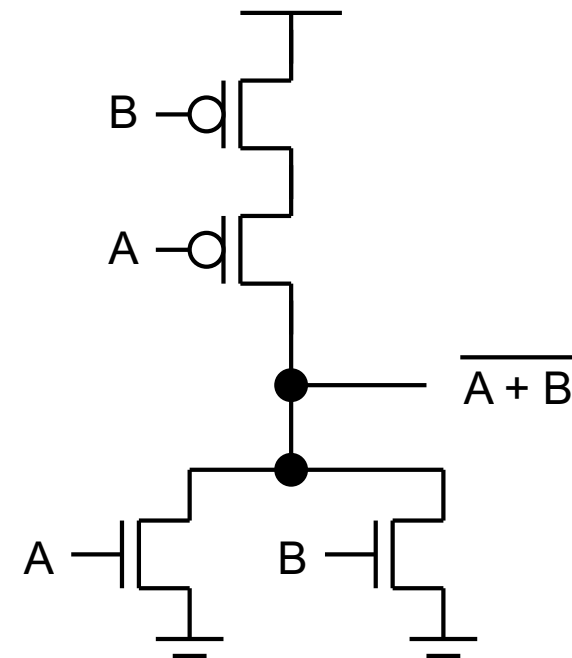
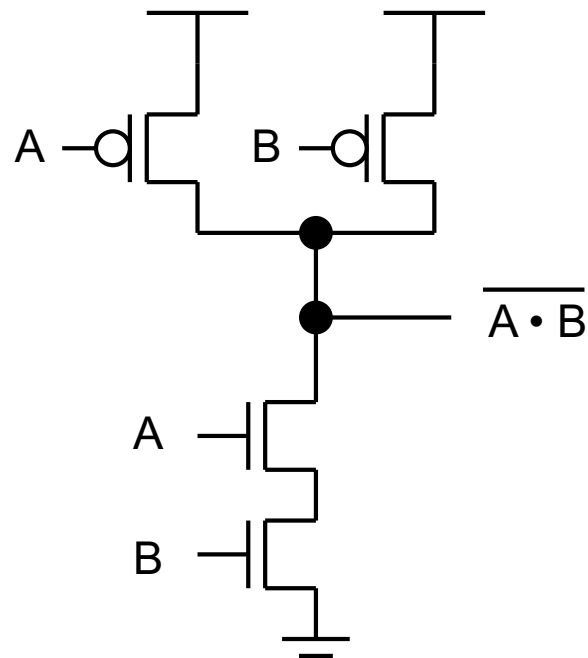


A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

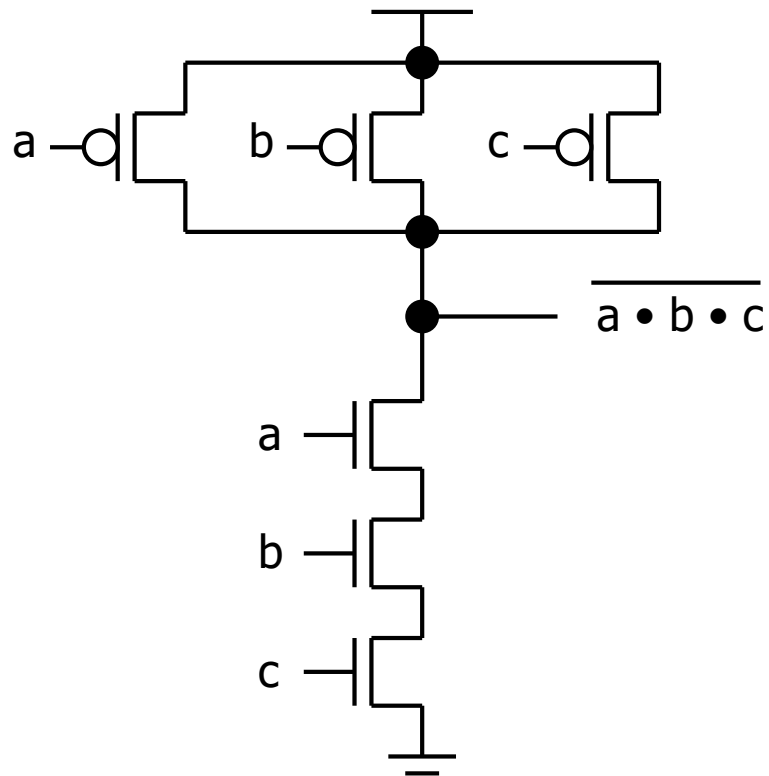
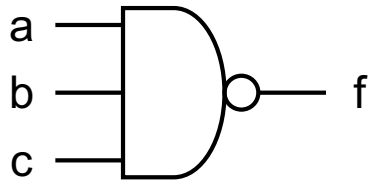
NOR2



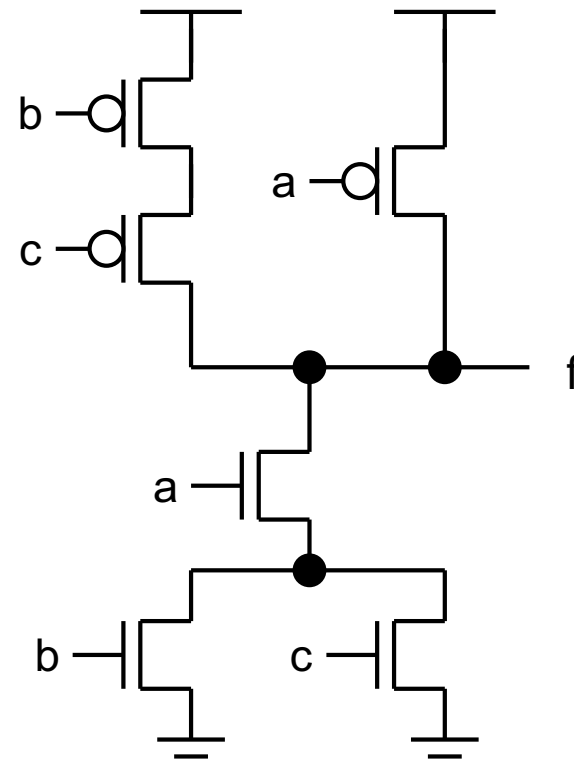
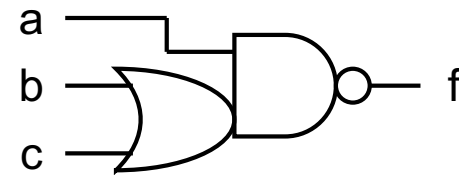
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



Larger Gates, Mixed Gates



NAND3

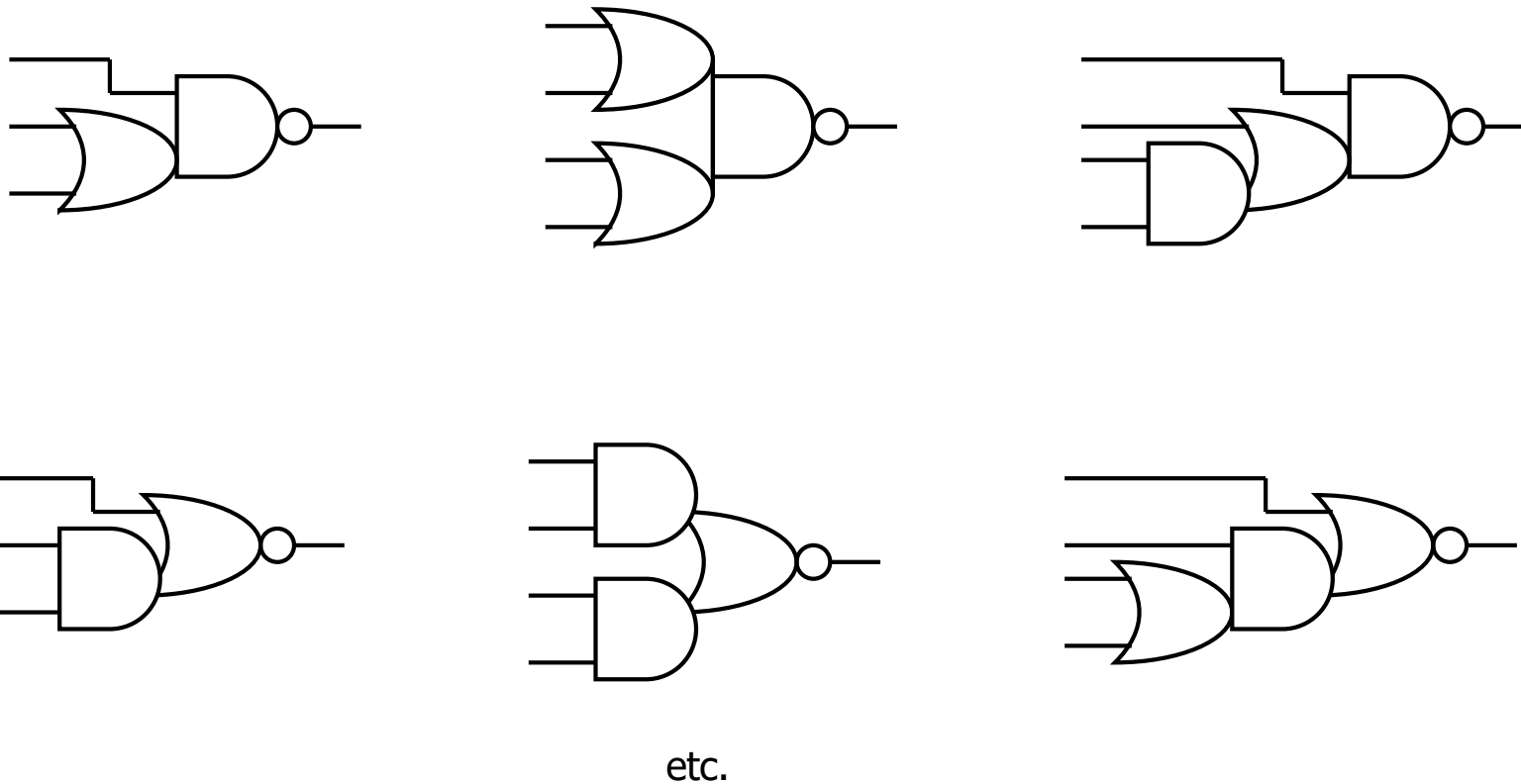


'OR2NAND1' = ON21 (in AMS)

- Merke: Mehr als ~ 5 Eingänge sind ungebräuchlich (zu viele MOS in Serie)

Weitere gemischte Gatter

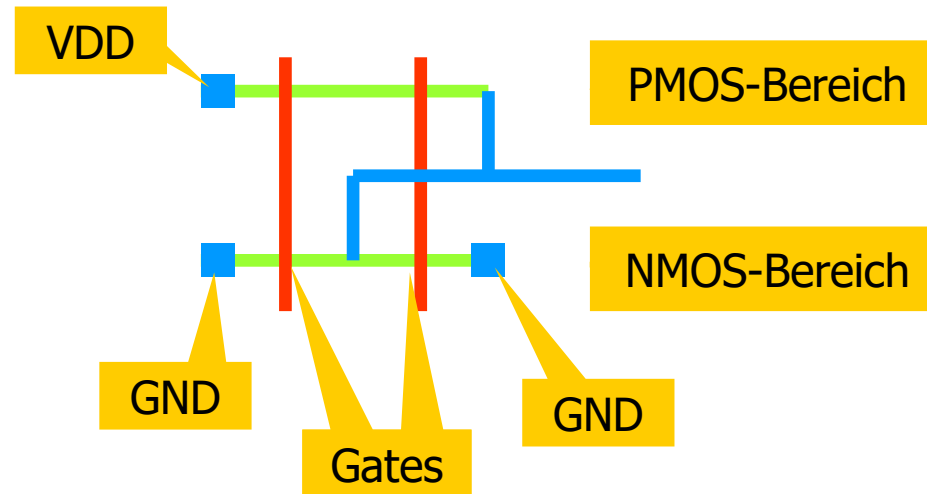
- Entsprechend kann man in CMOS direkt implementieren:



- Diese direkt implementierten Gatter benötigen bei N Eingängen $2N$ Transistoren ($N \times$ NMOS, $N \times$ PMOS)

Stick Diagramm

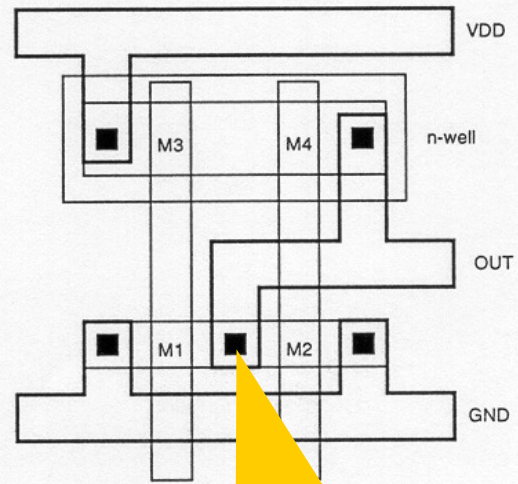
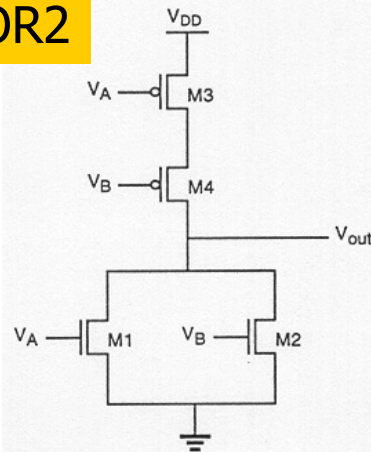
- Um die Topologie eines Layouts darzustellen zeichnet man 'Stick-Diagramme' mit vereinfachten FETs:



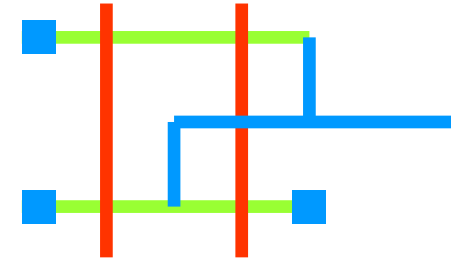
- Welches Gatter ist das ?

Gatter Schaltung und Layout

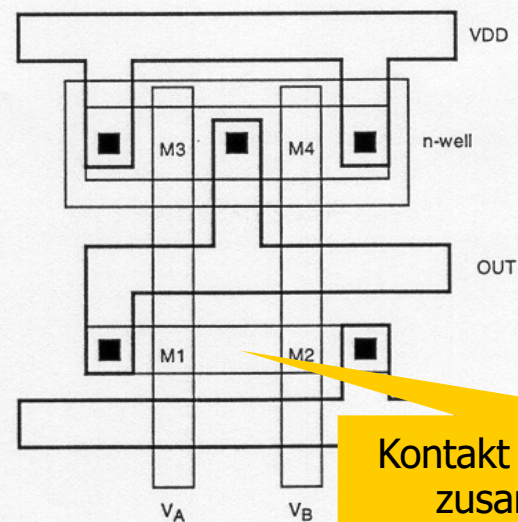
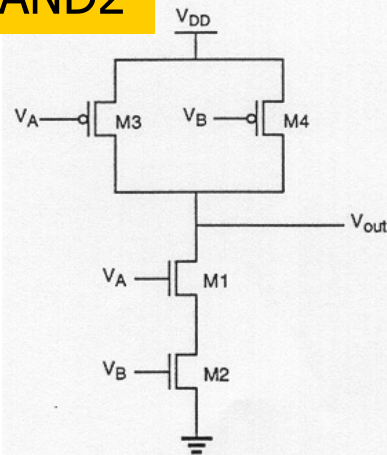
NOR2



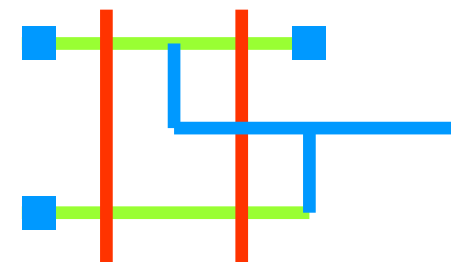
Drain gemeinsam genutzt



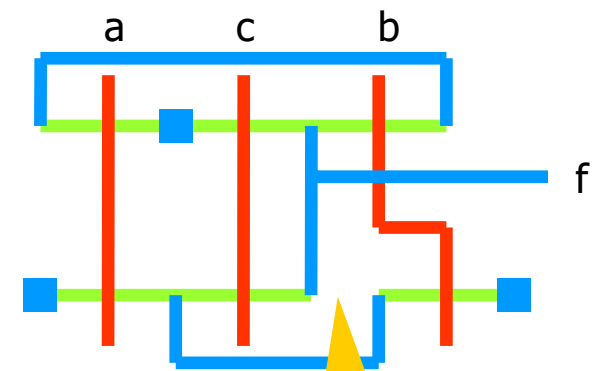
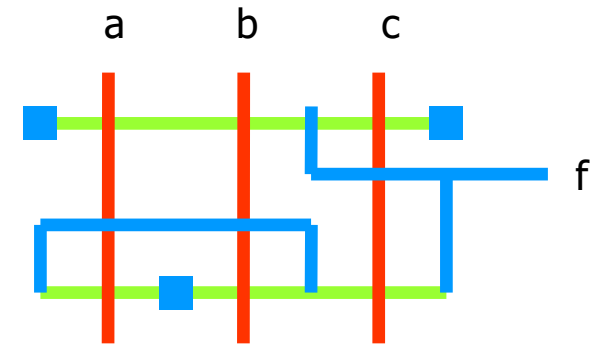
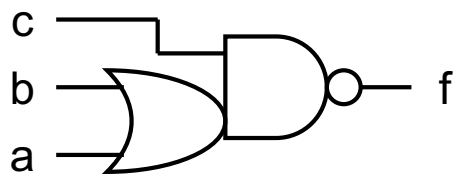
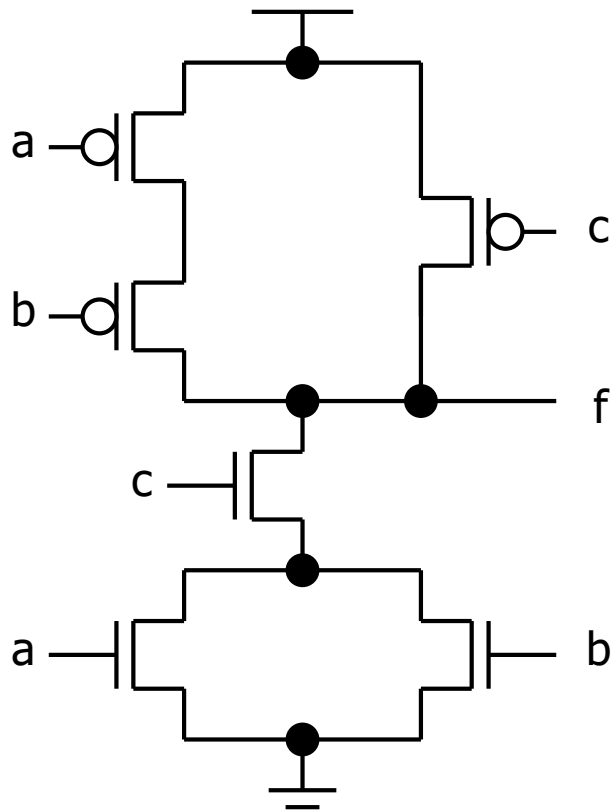
NAND2



Kontakt nicht nötig \Rightarrow Gates können zusammengeschoben werden



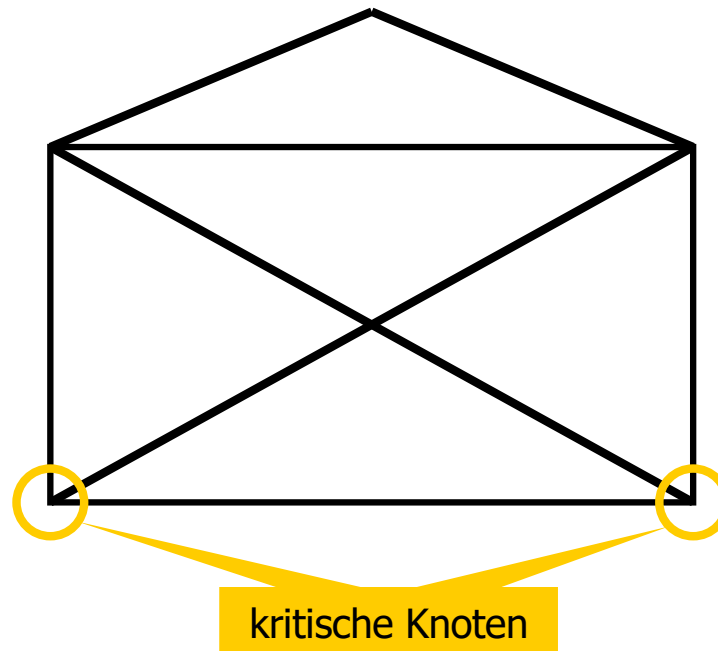
Zwei unterschiedliche Stick Layouts



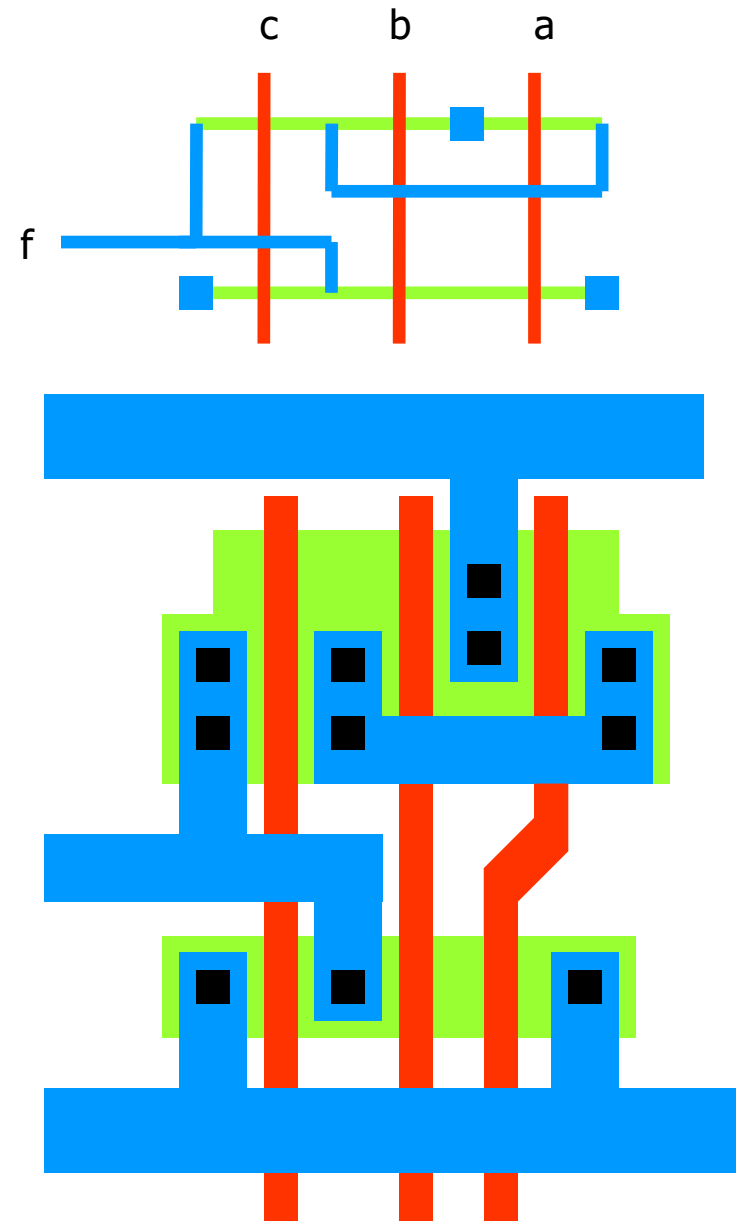
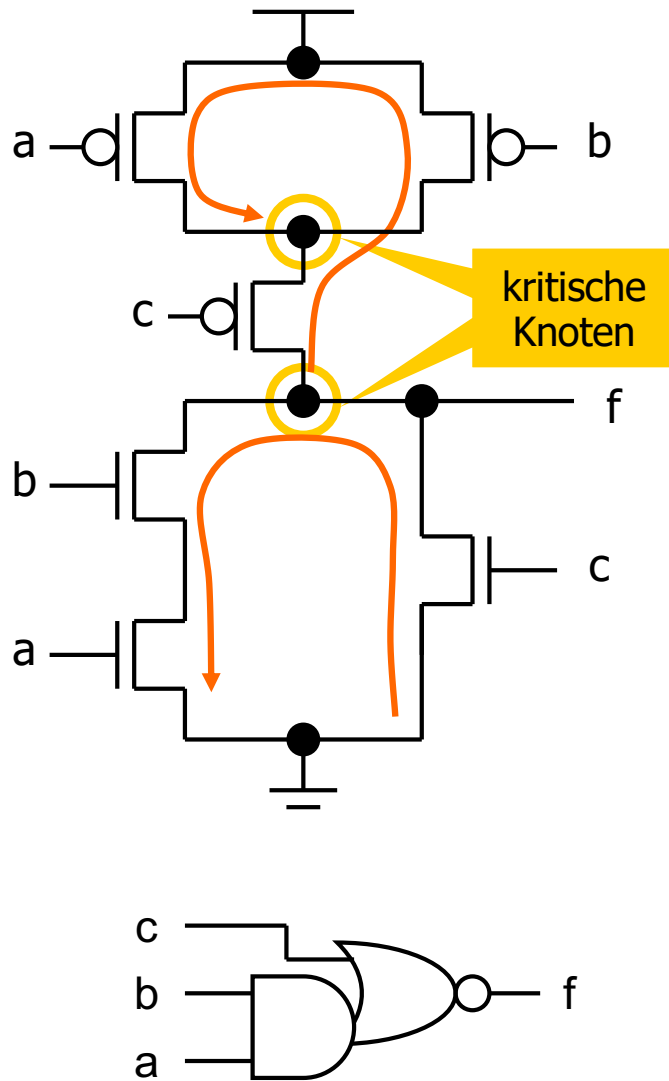
Diffusion hat Unterbrechung

Wie findet man eine 'gute' Transistorreihenfolge ?

- Eine 'gute' Transistorreihenfolge ergibt eine **ununterbrochene** Diffusion (kompaktes Layout)
- Man sucht dazu eine Transistorreihenfolge, so daß die (Euler-) Pfade durch die NMOS und PMOS alle MOS erreichen.
- Versorgungsspannung und Masse bilden jeweils einen Knoten
- Gibt es '**kritische** Knoten', an denen eine **ungerade** Anzahl Netze zusammenläuft, muß man dort anfangen/ aufhören
- Bei mehr als 2 Knoten mit ungerader Anzahl von Netzen gibt es keine Lösung. Die Diffusion muß dann unterbrochen werden.
- Bei mehreren Lösungen wählt man z.B. so aus, daß Signale 'innen' liegen, Versorgungen 'außen'

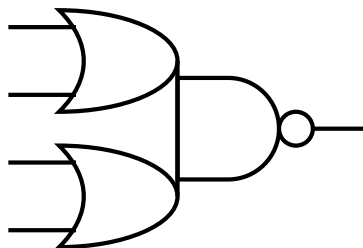
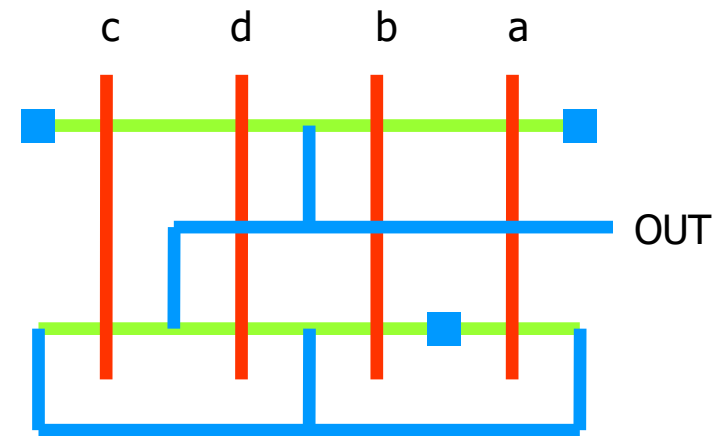
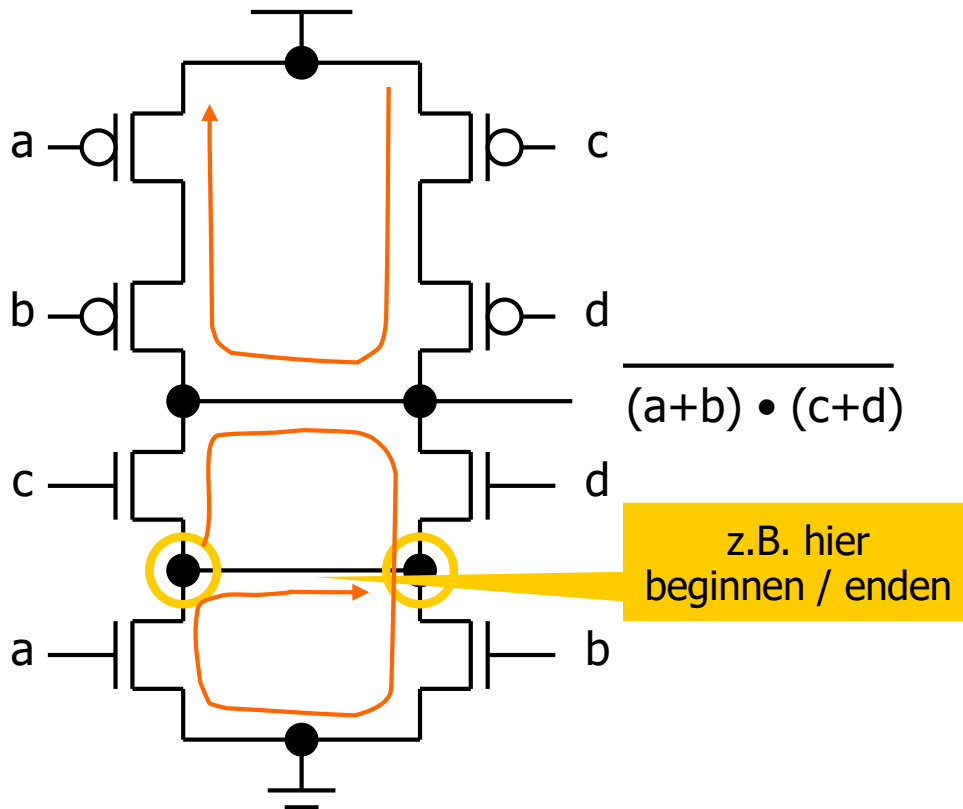


Beispiel 1

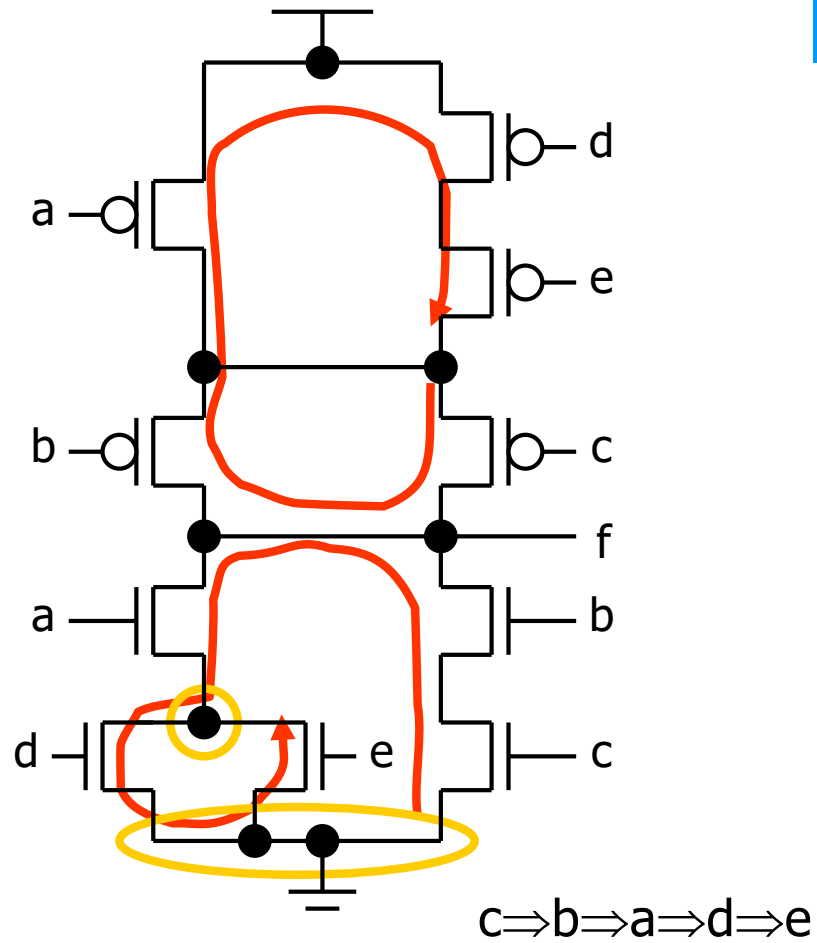


Beispiel 2

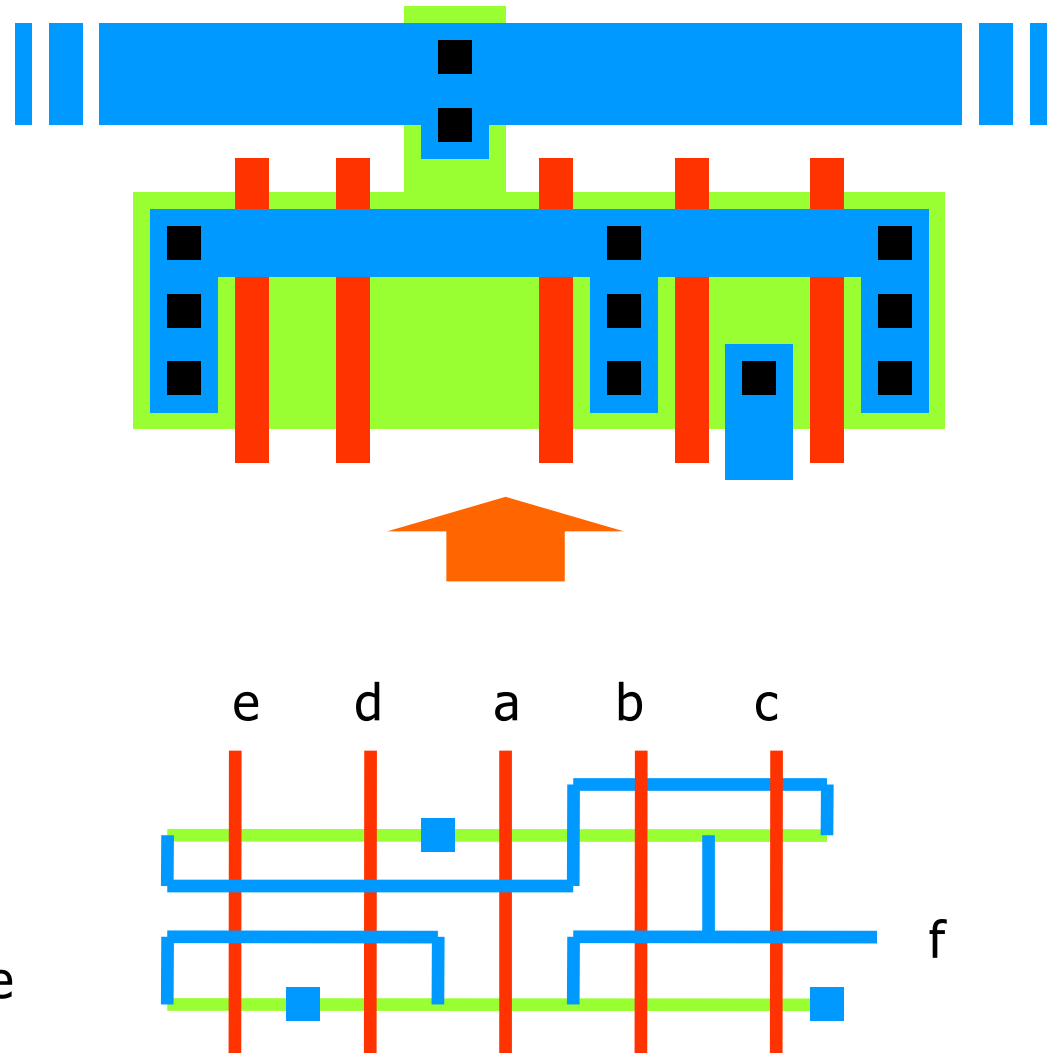
- Hier gibt es keinen kritischen Knoten (immer 2 oder 4 MOS), daher gibt es viele Möglichkeiten



Beispiel 3

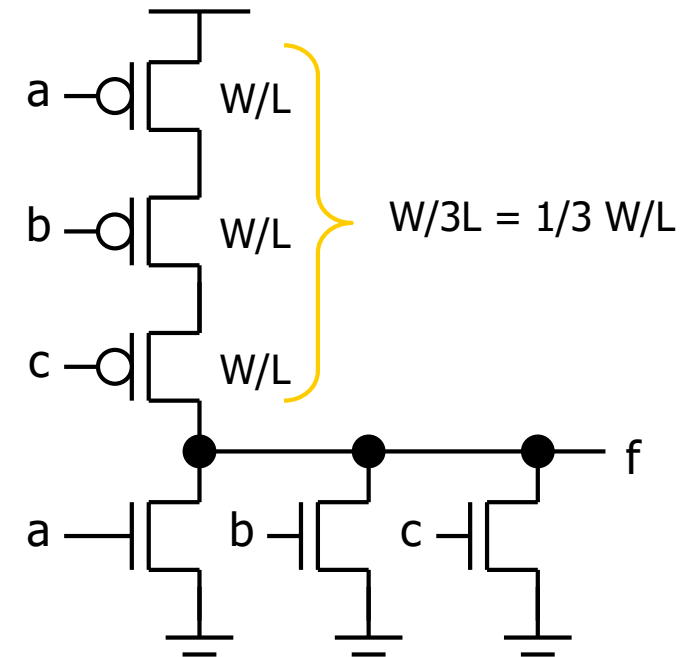


$$Y = \overline{a(d+e)+bc}$$



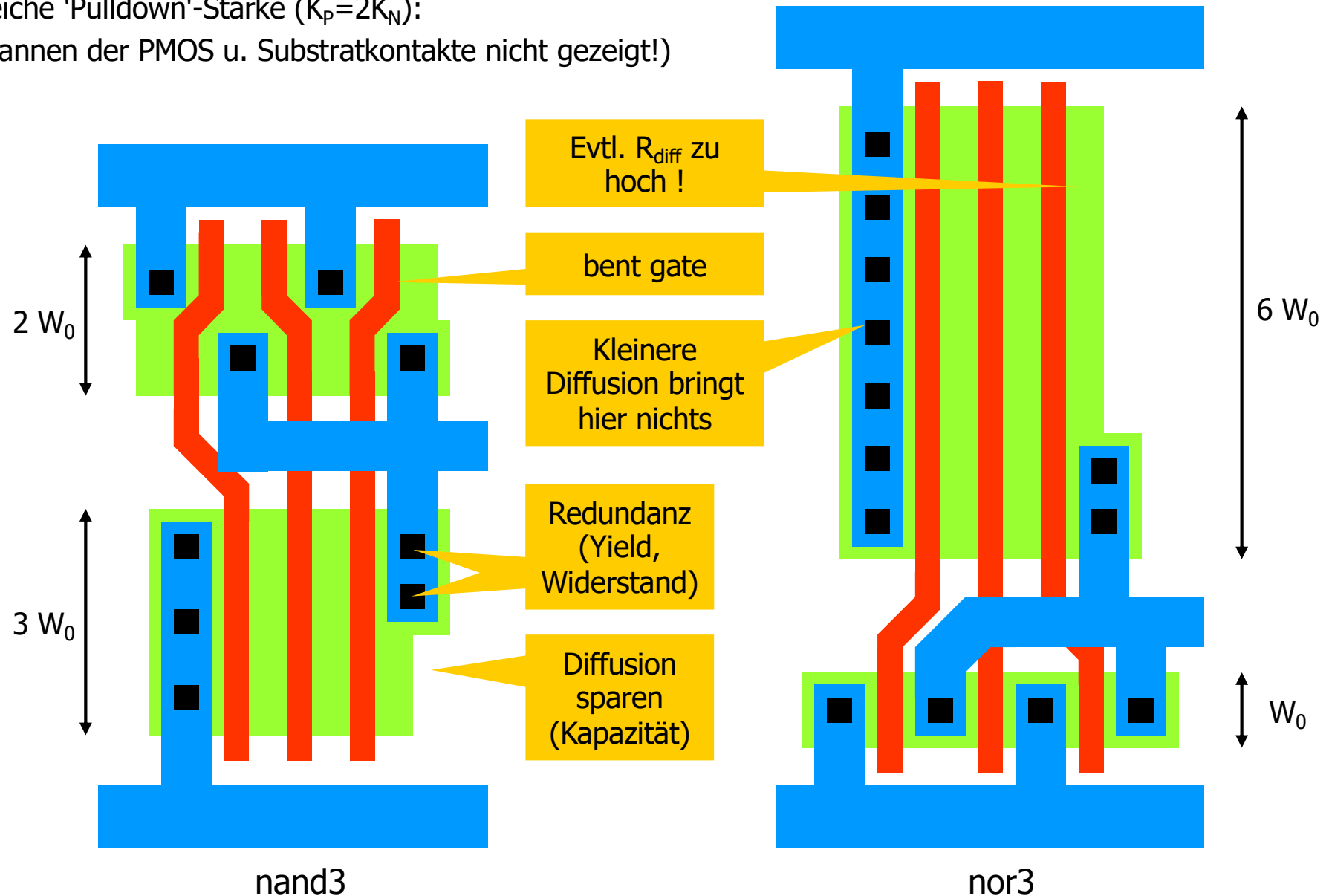
Transistorgrößen in MOS Gattern

- Betrachte z.B. NOR Funktion mit 3 Eingängen:
- Die Serienschaltung der PMOS hat bei 3 identischen W/L die effektive Größe $W/L_{\Sigma} = 1/3 W/L$.
- Die Anstiegszeit wird daher 3 x langsamer als im Inverter.
- Lösung (L ist bereits minimal):
W der PMOS wird 3x größer gemacht als im Inverter
(hier wären die PMOS also $(2-3) \times 3 = 8x$ größer als die NMOS...)
- Dies führt jedoch zu **sehr großen Eingangskapazitäten**, insbesondere bei PMOS und bei Gattern mit vielen 'gestapelten' Transistoren.
- Mehr als 3 gestapelte PMOS Transistoren werden daher kaum benutzt.
- **NAND Logik ist hier vorteilhaft**, weil die PMOS dort parallel sind und die Größe nicht erhöht werden muß.
(etwa 75% aller Logik benutzt NANDs' - laut 'design of high performance microprocessor circuits', IEEE Press)
- Ein NAND mit 3 Eingängen hat also etwa gleich große NMOS und PMOS Transistoren.



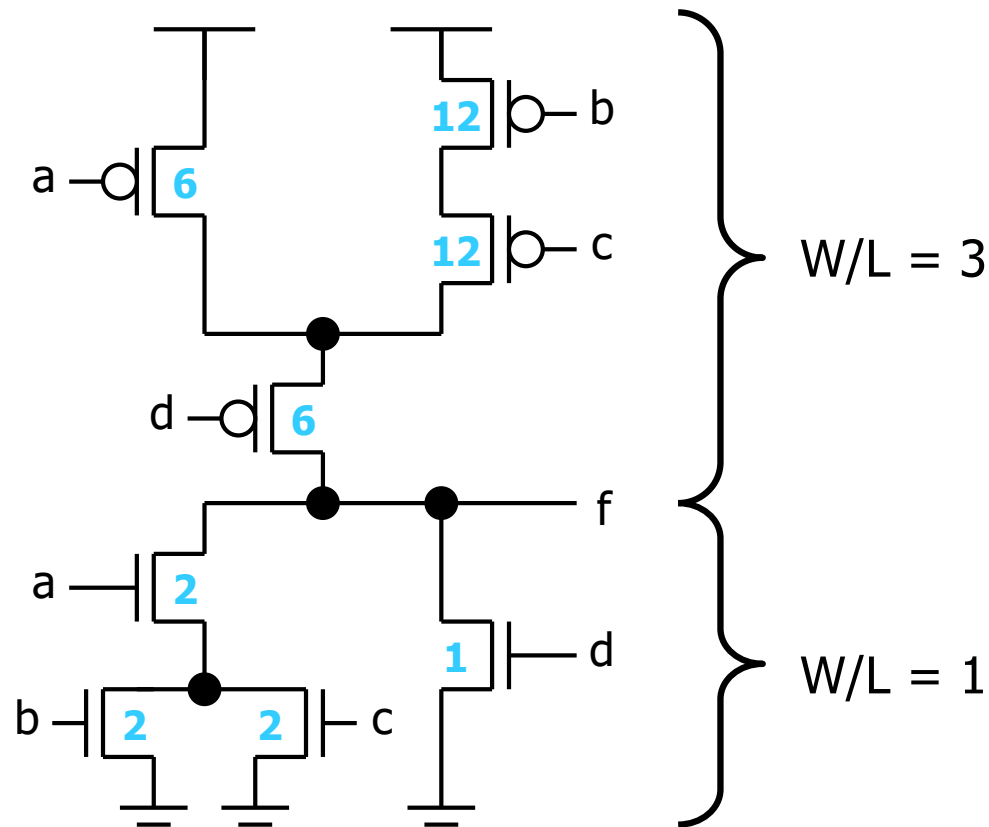
Layoutvergleich: nor3 vs. nand3

- Gleiche 'Pulldown'-Stärke ($K_p=2K_n$):
- (Wannen der PMOS u. Substratkontakte nicht gezeigt!)



Ideale Transistorgröße in komplexeren Gattern

- Beispiel für relatives W/L eines komplexen Gatters für symmetrisches Schalten bzw. Schwelle = $V_{DD}/2$ bei $K_N=3K_P$:



- Aber: In der Realität (Standardzellbibliotheken) wird dies oft nicht befolgt (gäbe sehr unschöne Layouts).
- **Wenn mehrere Eingänge gleichzeitig schalten, stimmt W/L nicht mehr!**

Kapazitäten der internen Knoten

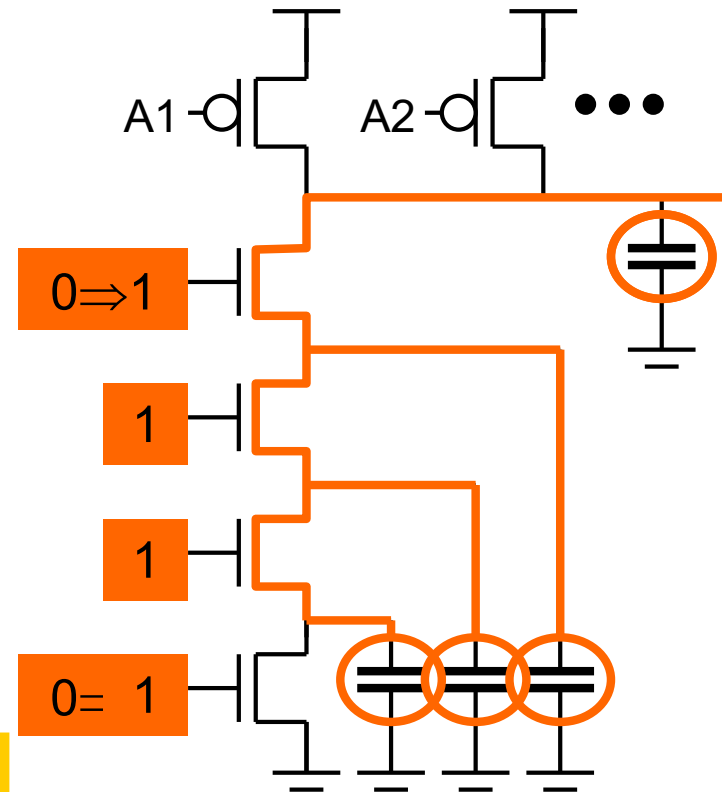
- Betrachte z.B. NAND Funktion mit 4 Eingängen:
- Die Knoten zwischen den NMOS Transistoren haben parasitäre Kapazitäten (Dioden)
- Die unteren Transistoren müssen daher höhere Kapazitäten umladen als die oberen: Die Verzögerung ist länger
(Dieser Fall gilt, wenn die Lastkapazität klein oder vergleichbar zu den internen Kapazitäten ist.)

Lösungen:

- **Zeitkritische Signale nahe am Ausgang anschließen**
- **Transistoren 'unten' breiter machen**, um den Durchgangswiderstand zu erniedrigen und die RC-Zeit zu reduzieren
(Diese 'tapered layouts' werden jedoch kaum mehr benutzt, da die Layouts sehr unsystematisch werden!)



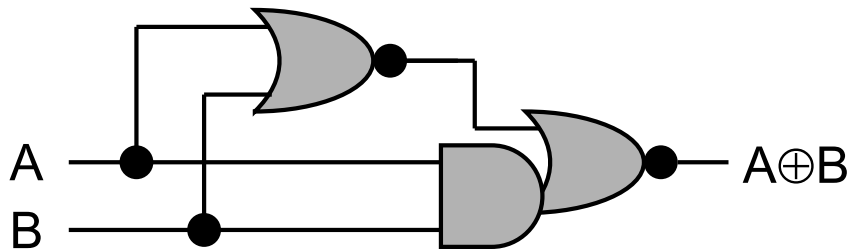
Hier muss der Poly-Active Abstand eingehalten werden, daher ist der Poly-Pitch auf dem Gate nicht minimal und die Diffusions-Kapazitäten sind unnötig hoch. Tapering bringt also nicht viel!



XOR

- XOR / XNOR sind wichtige Funktionen (Vergleicher, Zähler, Parität)

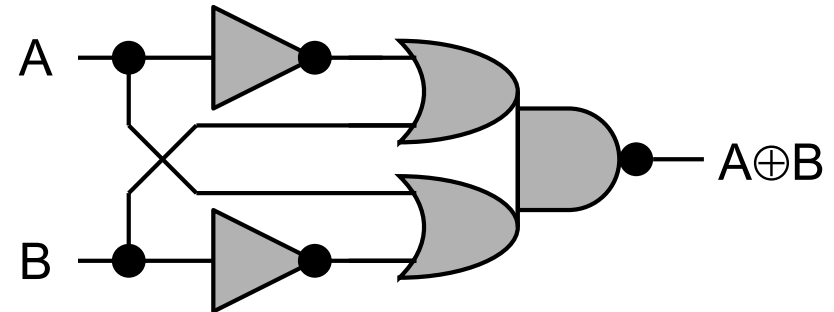
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



4 MOS

6 MOS

10 Transistoren

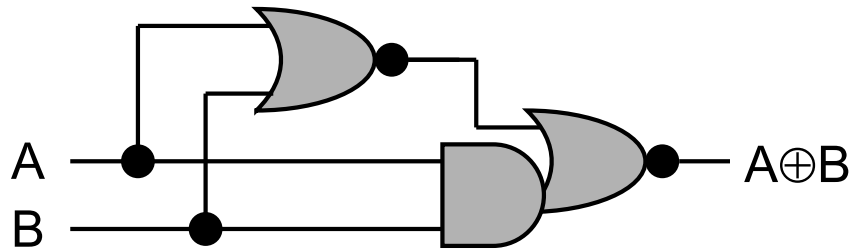


je 2 MOS

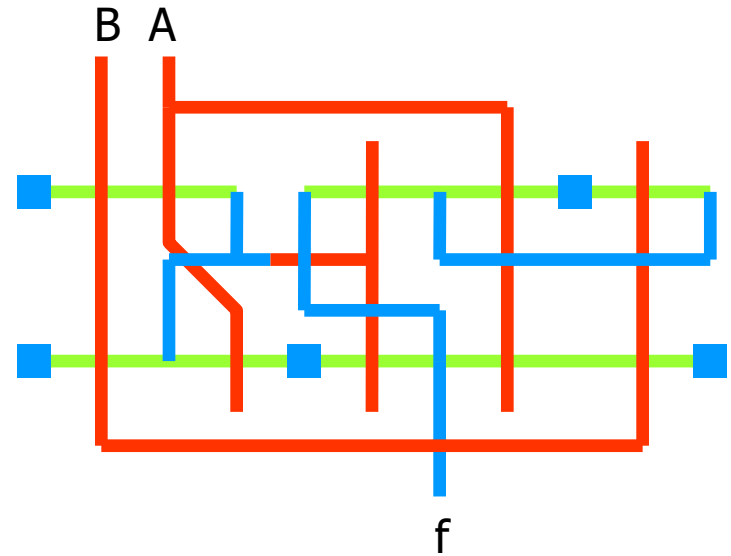
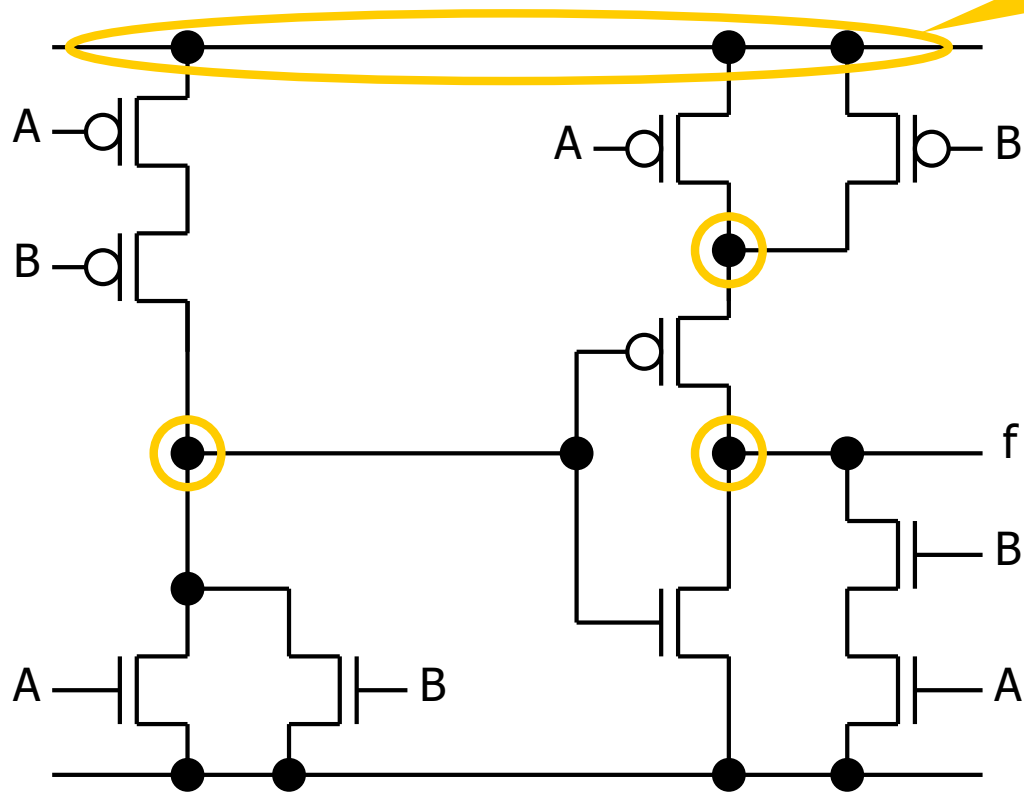
8 MOS

12 Transistoren

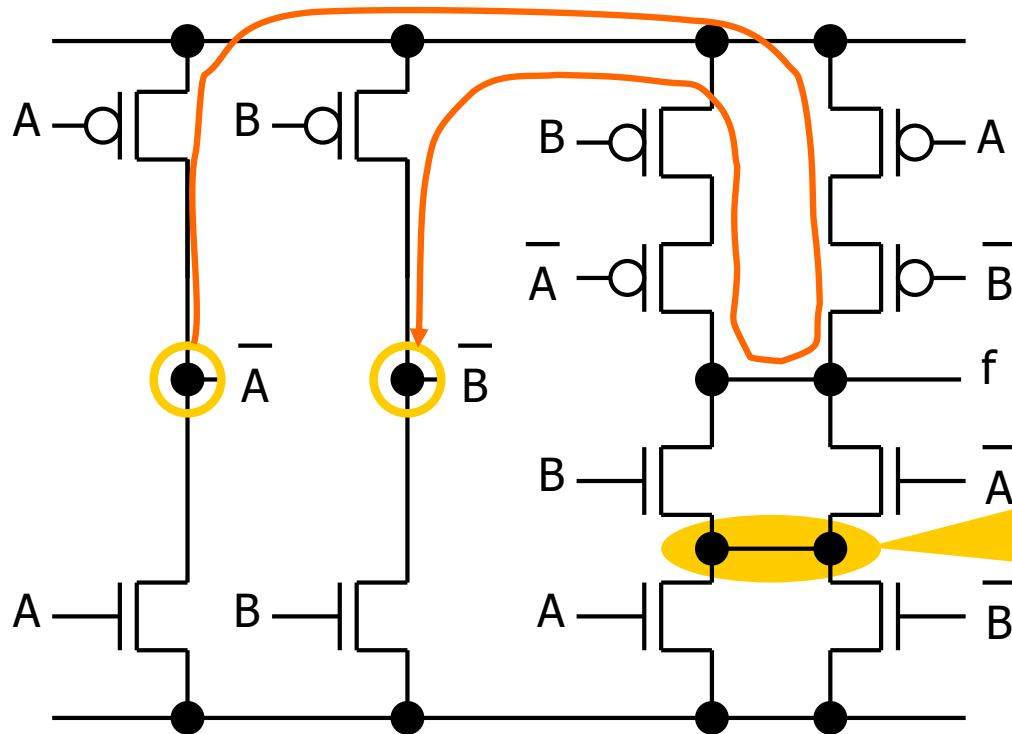
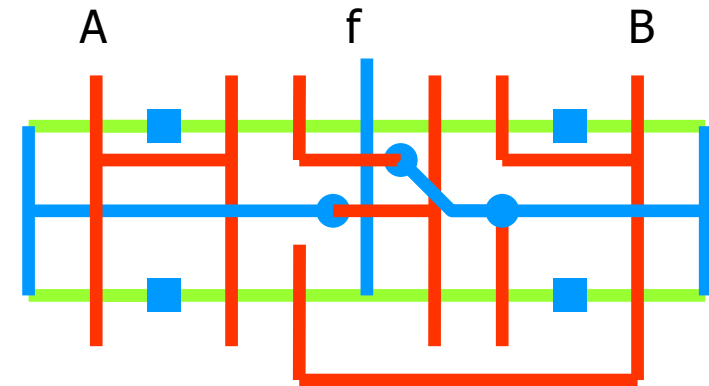
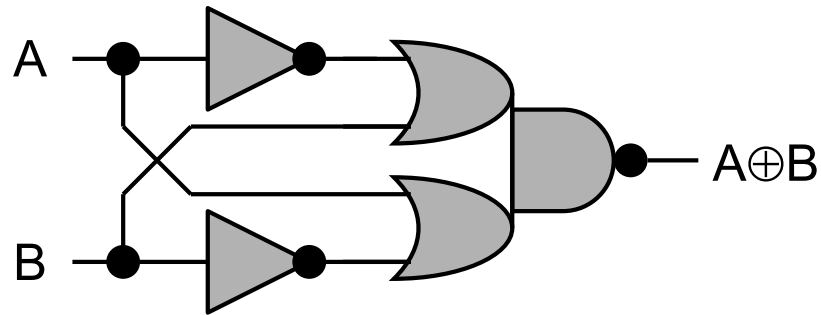
CMOS XOR Layout: 10 MOS



4 kritische Punkte.
Keine durchgehende Diffusion
bei PMOS möglich !



CMOS XOR Layout: 12 MOS



Diese Verbindung kann man weglassen, ohne die Funktion zu beeinträchtigen. (Weil A, !A und B, !B immer komplementär sind!) Das hintere Gatter ist dann kein 'echtes' CMOS mehr!

BUFFER

Treiben großer Lasten

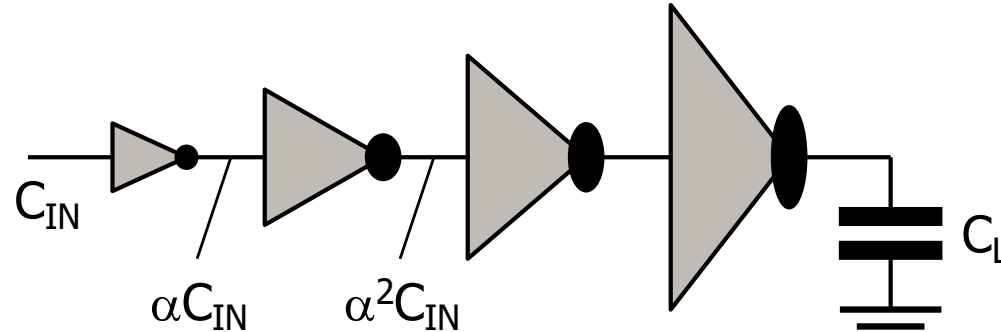
- Aufgabe: Treibe eine gegebene, große Last mit minimaler Verzögerung !
- Beispiele sind Netze mit hohem Fanout, Pads (externe Lasten haben leicht 10pF!), Taktleitungen
- Aus $t_p \sim \frac{C_L}{K (W/L) VDD}$ sieht man, daß man **K erhöhen** muß, um das hohe C_L zu tolerieren.
- Die Eingangskapazität eines großen Inverters ist aber hoch, so daß dieser wiederum schwer zu treiben ist. Das Problem ist also nur verschoben.



- Man denke auch an die hohen **Querströme**, die wegen der langsamen Anstiegszeit des Eingangssignals im großen Inverter rechts fließen !!

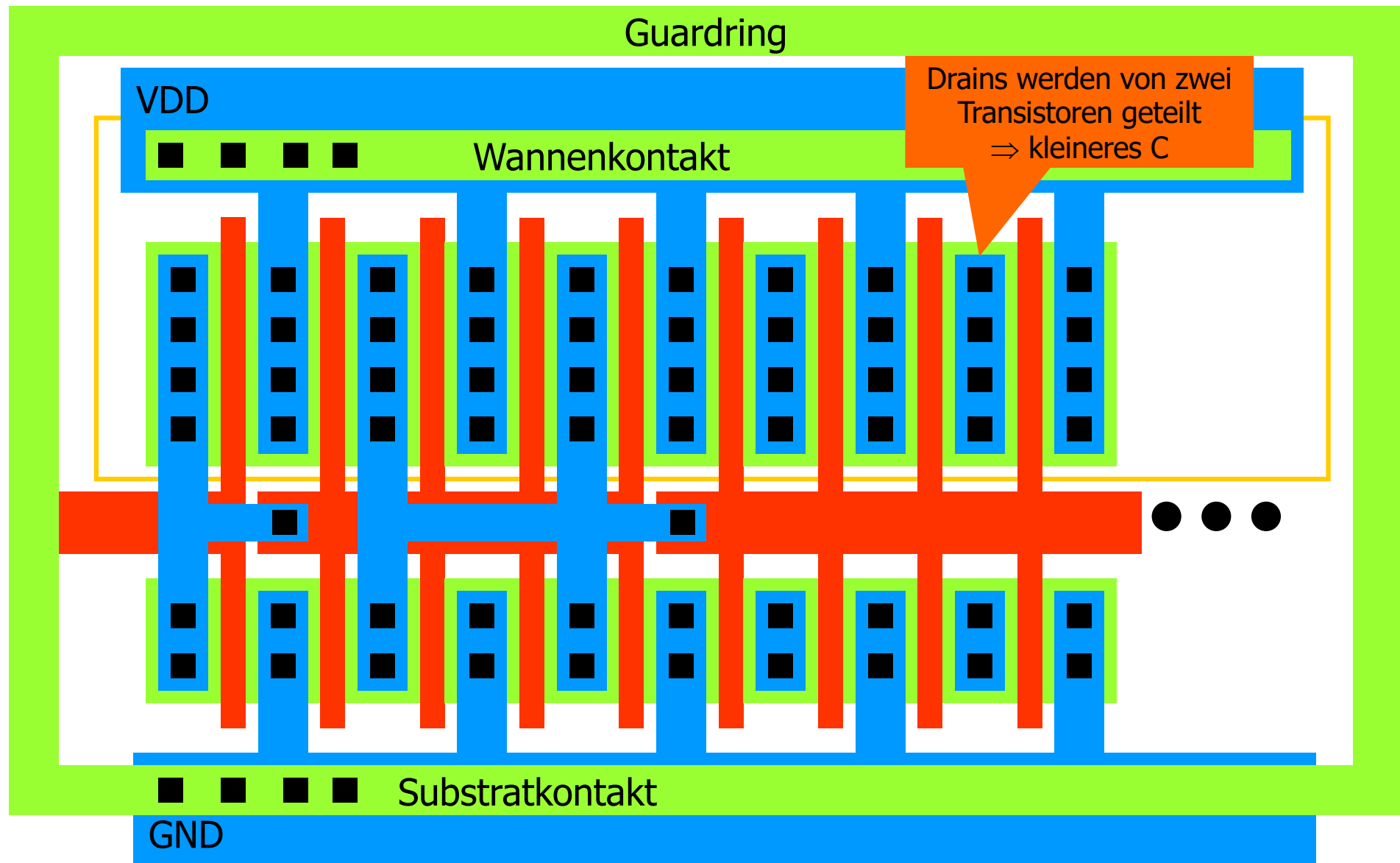
Inverterkette als Buffer

- **Lösung:** Kette aus zunehmend größer werdenden Invertern. Gesucht: Vergrößerung α pro Stufe



- Die Kette bestehe aus **n Invertern**, die um jeweils einen **Faktor α** größer werden
- Die Verzögerung **eines** Inverters sei $t_p = \alpha t_{p0}$
(t_{p0} ist die Verzögerung beim Treiben eines gleichgroßen Inv., das Treiben des α -fach größeren dauert länger)
- Die **Gesamtverzögerung** ist $T = n t_p = n \alpha t_{p0}$
- Es gilt $C_L = \alpha^n C_{IN}$ oder $n = \ln(C_L/C_{IN}) / \ln \alpha$
- Also **$T(\alpha) = n \alpha t_{p0} = (\alpha / \ln \alpha) \ln(C_L/C_{IN}) t_{p0}$**
- **Hier ist nur α unbekannt.** $\partial T(\alpha)/\partial \alpha = 0$ liefert **$\alpha = e = 2.718\dots$** und $T_{\min} = e \ln(C_L/C_{IN}) t_{p0}$
- **Jeder Inverter ist also etwa 3x so groß wie der vorhergehende**
- Man braucht **$\ln(C_L/C_{IN})$ Stufen**
- Aus verschiedenen Gründen (z.B. Minimierung des Querstroms oder der Fläche) sind etwas größere Verhältnisse (10) oft besser. Die Erhöhung der Verzögerung ist minimal (wenige %, s. Buch v. Veendrick).

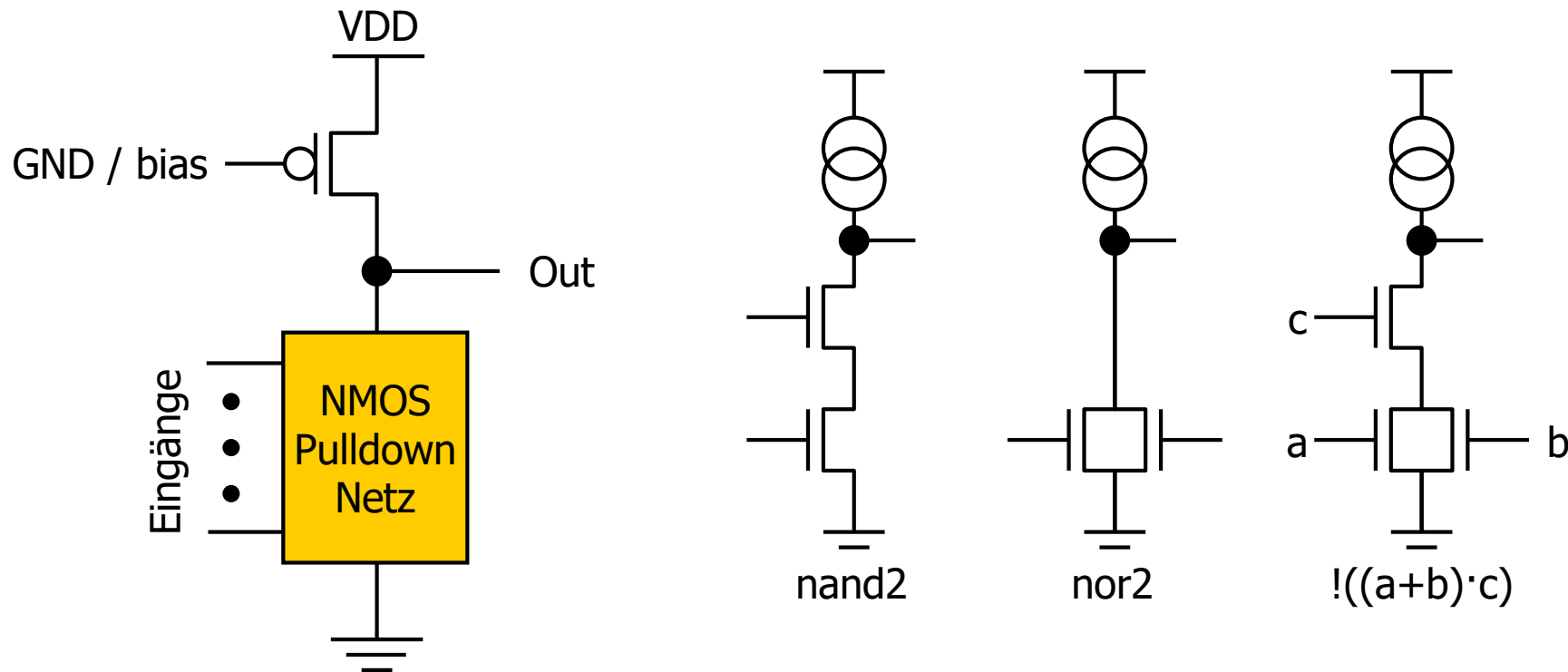
Typisches Buffer Layout



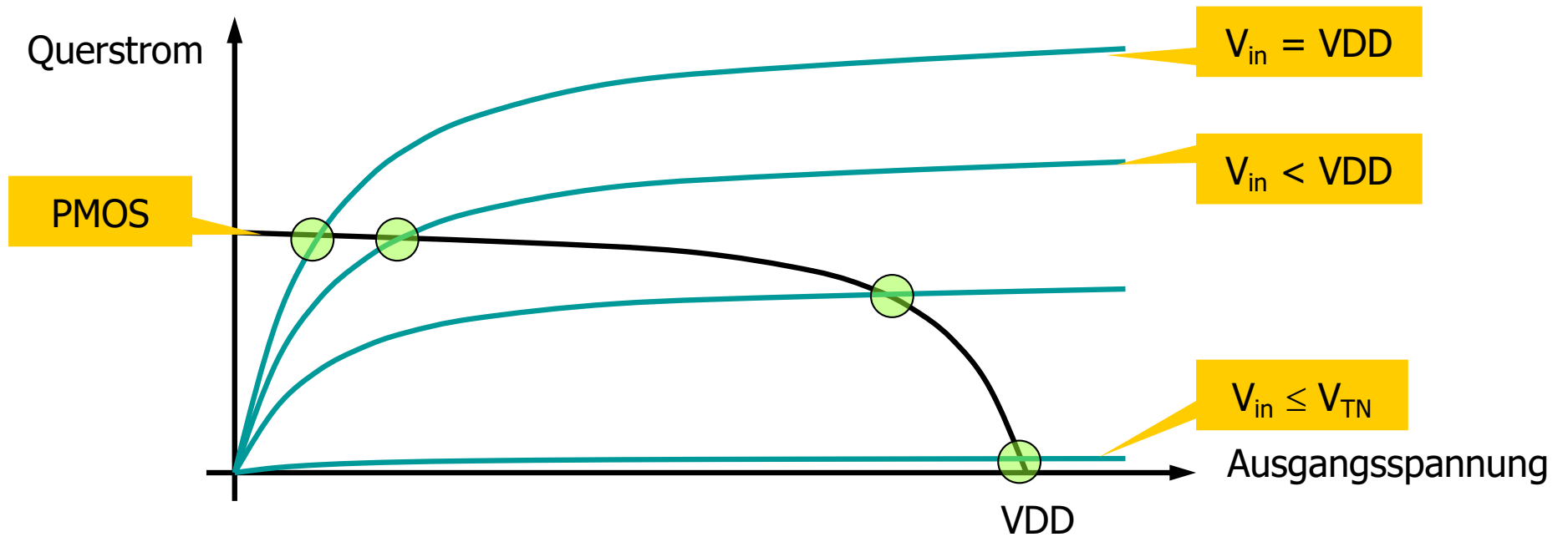
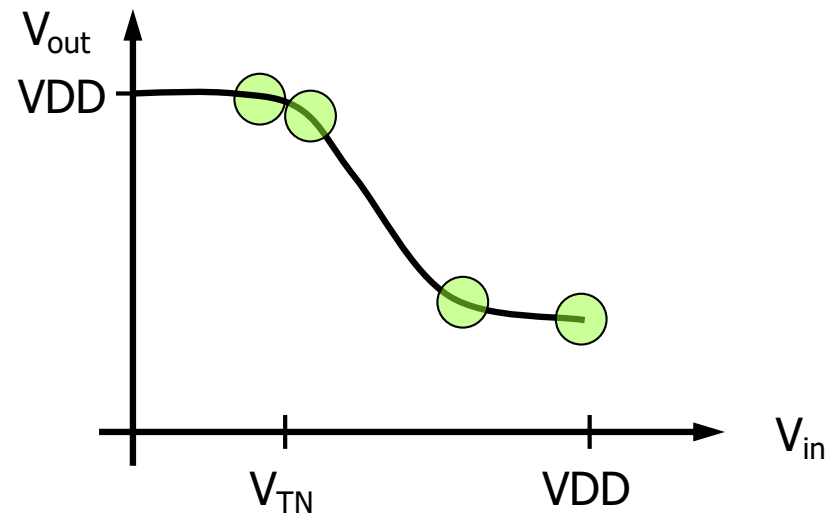
NMOS LOGIK

P-Last Logik ('pseudo – NMOS')

- Die Logische Funktion wird durch ein **NMOS Netzwerk** erzeugt, eine **P-Last** zieht nach VDD
 - Das aktive **NMOS** Netzwerk ist wegen $K_N > K_P$ vorteilhaft (Mobilität der Elektronen!)
- Der PMOS wirkt (fast) als Stromquelle
 - Besser ist es, die Gatespannung aus einem Referenzstrom zu erzeugen. Man benötigt dann eine **Biasspannung**
 - Bei $V_{Gate} = GND$ hängt der Strom stark von VDD ab – schlecht, aber häufig benutzt (kein Bias benötigt)

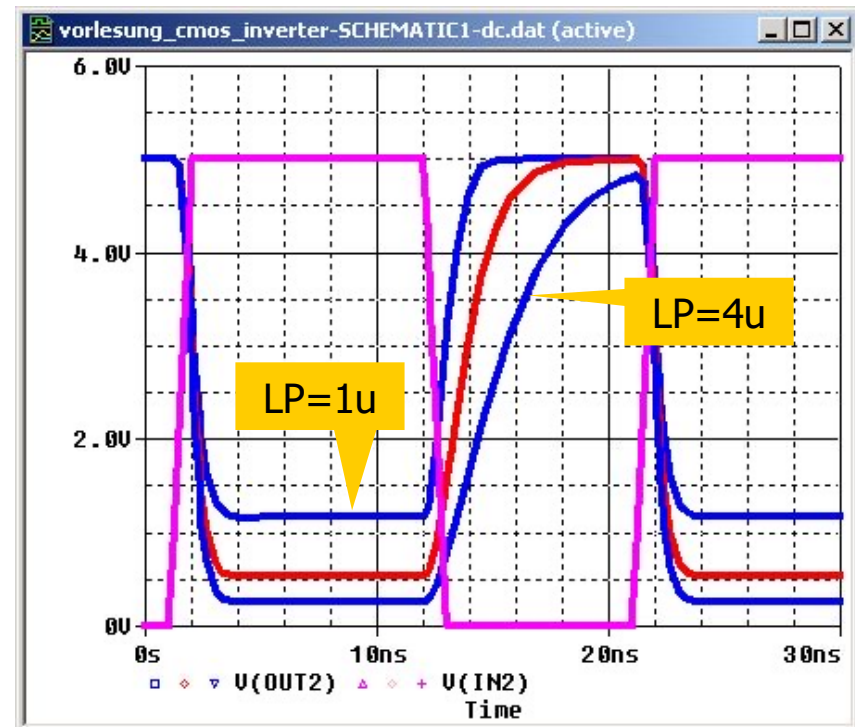
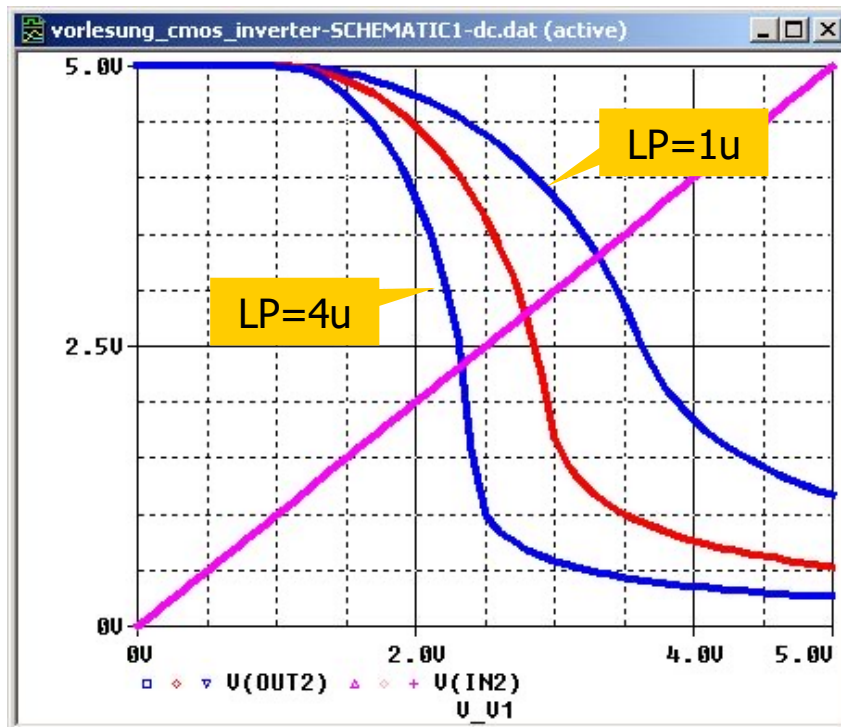
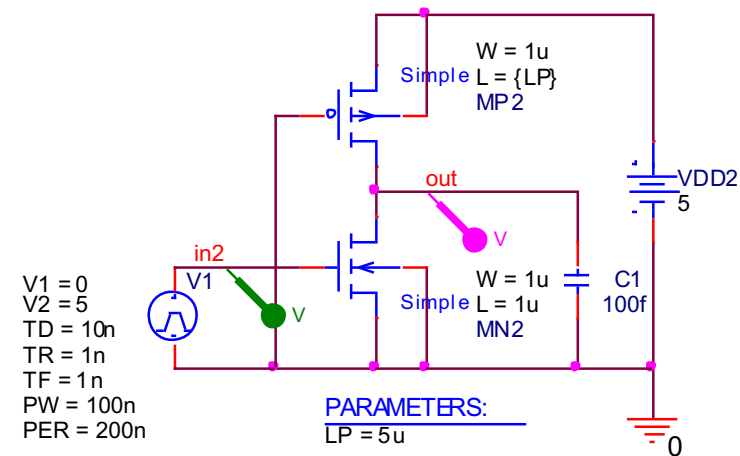


Übertragungsverhalten



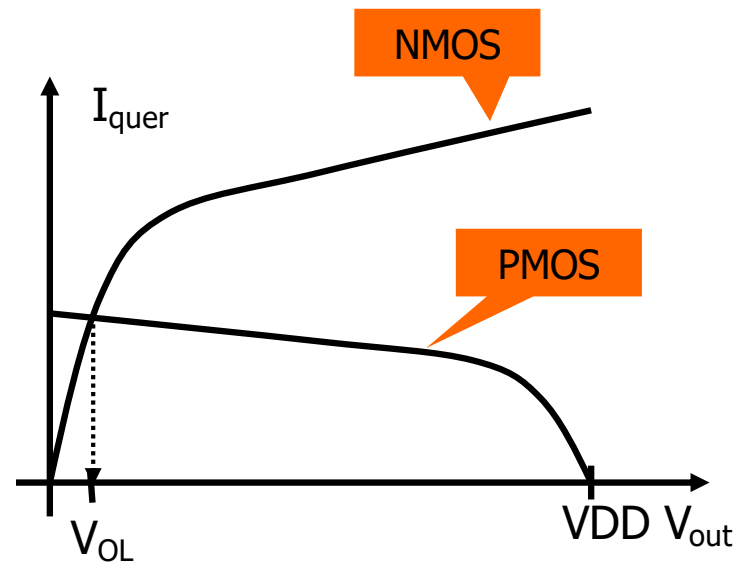
SPICE Simulation

- Hier: 'Simple' – Modelle mit verschiedenen langen PMOS
- Langer PMOS: + niedriges Low-Level
- langsame Anstiegszeit
- Kurzer PMOS: - hohes Low-Level
+ schnelle Anstiegszeit
- hoher Querstrom



Berechnung V_{OL}

- Berechnung der Low-Ausgangsspannung aus der Ausgangskennlinie von NMOS und Last



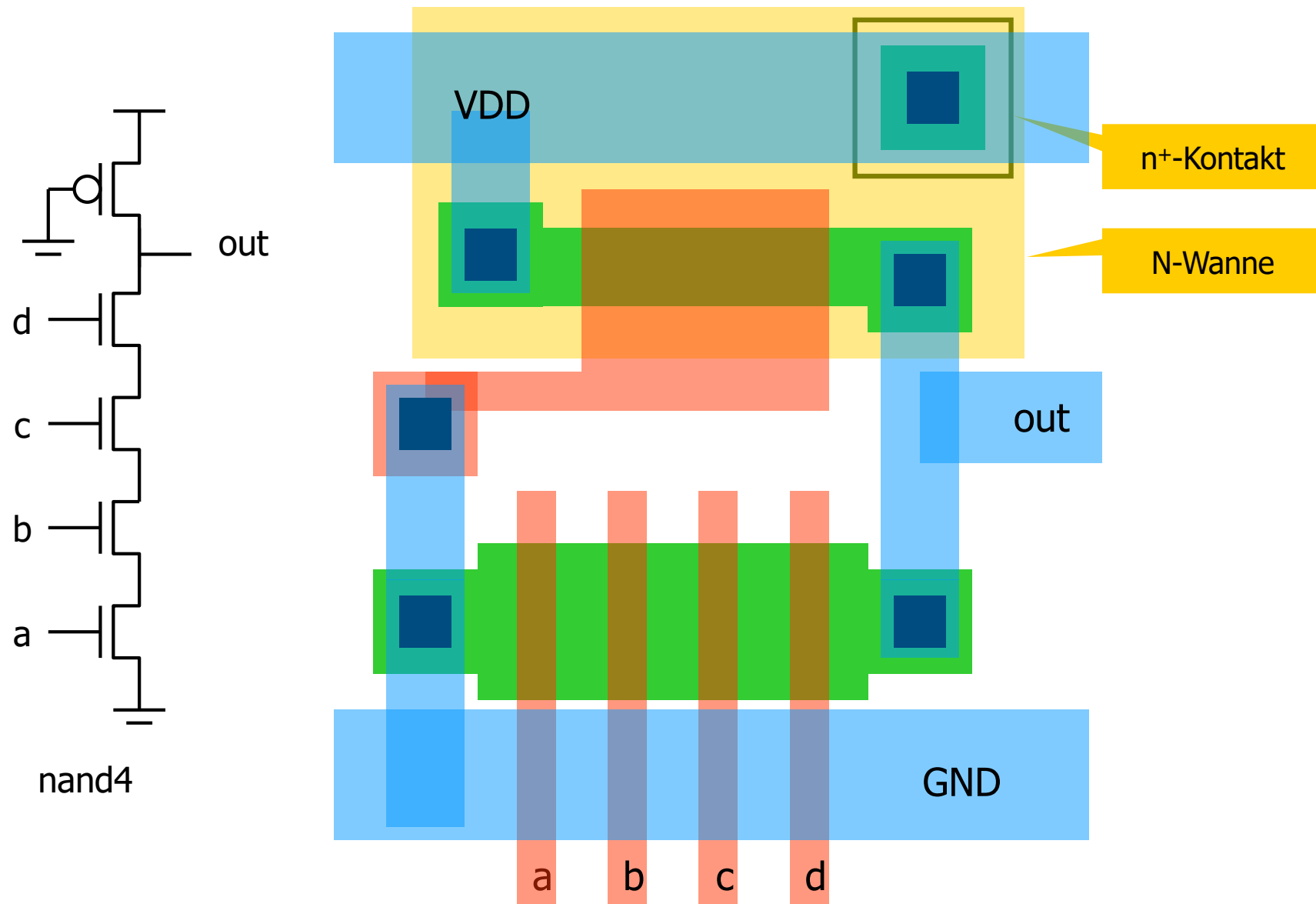
- NMOS ist im linearen Bereich, PMOS in Sättigung
- Mit $V_T = V_{\text{TN}} = |V_{\text{TP}}|$:

$$I_{\text{NMOS}} = k_N [(V_{\text{DD}} - V_T) V_{\text{OL}} - \frac{1}{2} V_{\text{OL}}^2] = \frac{1}{2} k_P (V_{\text{DD}} - V_T)^2 = I_{\text{PMOS}}$$

$$V_{\text{OL}} = (V_{\text{DD}} - V_T) [1 - \sqrt{1 - k_P/k_N}]$$

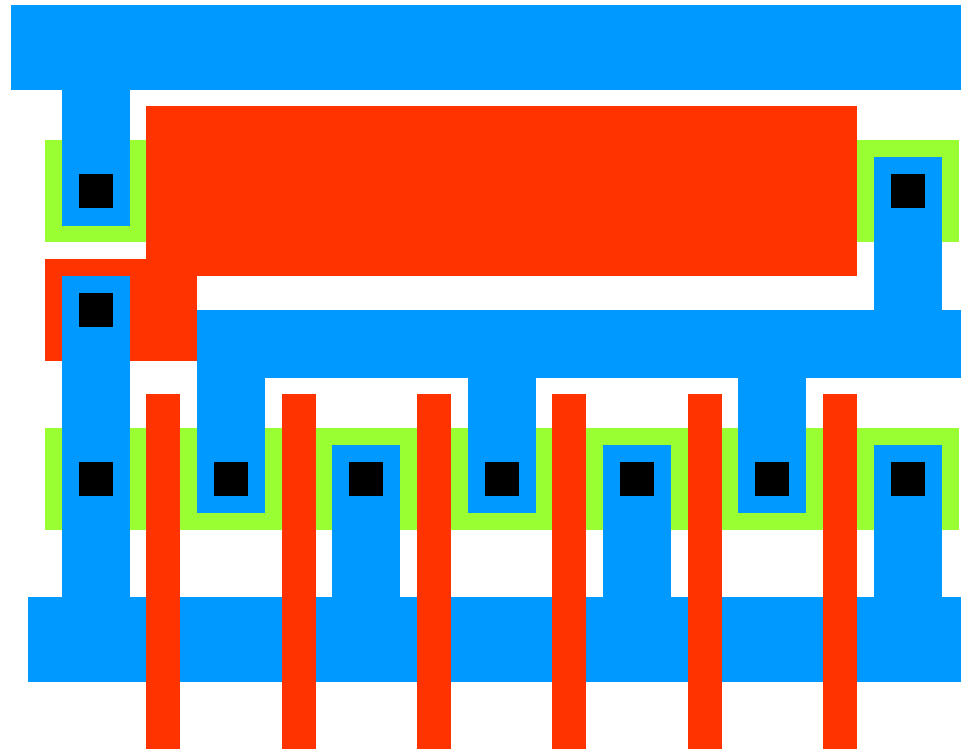
- k_p muß viel kleiner als k_n sein !!!
- Das low-level hängt stark von V_{DD} ab – schlecht!
- **V_{OL} muss natürlich viel kleiner sein als die (NMOS) Schwelle des nächsten Gatters**

Pseudo-NMOS NAND4 Gatter



P-Last NOR6

- Hier mit sehr langem PMOS. Sehr kompaktes Layout.



Pseudo-NMOS Logik

Vorteile

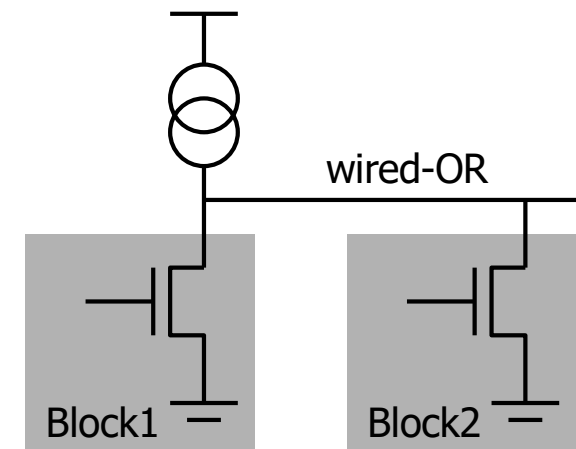
- Erfordert nur N+1 Transistoren (N NMOS, 1 PMOS)
- Topologie sehr einfach
- Kleine Eingangskapazität
- Gut geeignet z.B. für Decoder

Nachteile

- Statischer Stromverbrauch, wenn Ausgang auf Null ist
- Abfall- und Anstiegszeiten u.U. stark asymmetrisch
- Kleiner unterer Störabstand: $V_{IL} \sim V_{TNr}$, $V_{OL} > 0$
- Dimensionierung des PMOS ist kritisch ('**ratioed logic**):
 - Er darf nicht zu groß sein, damit die NMOS den Ausgang noch gegen 'Null' (i.e. unter V_{IL}) ziehen können
 - Er darf nicht zu klein sein, weil t_{pLH} dann zu groß wird

Anwendungen

- **Open-Drain Ausgang.** Mehrere Ausgänge können verbunden werden. Mit einer Last ergibt sich eine ODER-Verknüpfung '**Wired-OR**' (eigentlich NOR oder NAND...)
- Die Last kann an eine andere Spannung angeschlossen werden als VDD:
 - **Level-Shifter**
 - **Gunning Transceiver Logik** (GTL) mit niedriger Abschlußspannung



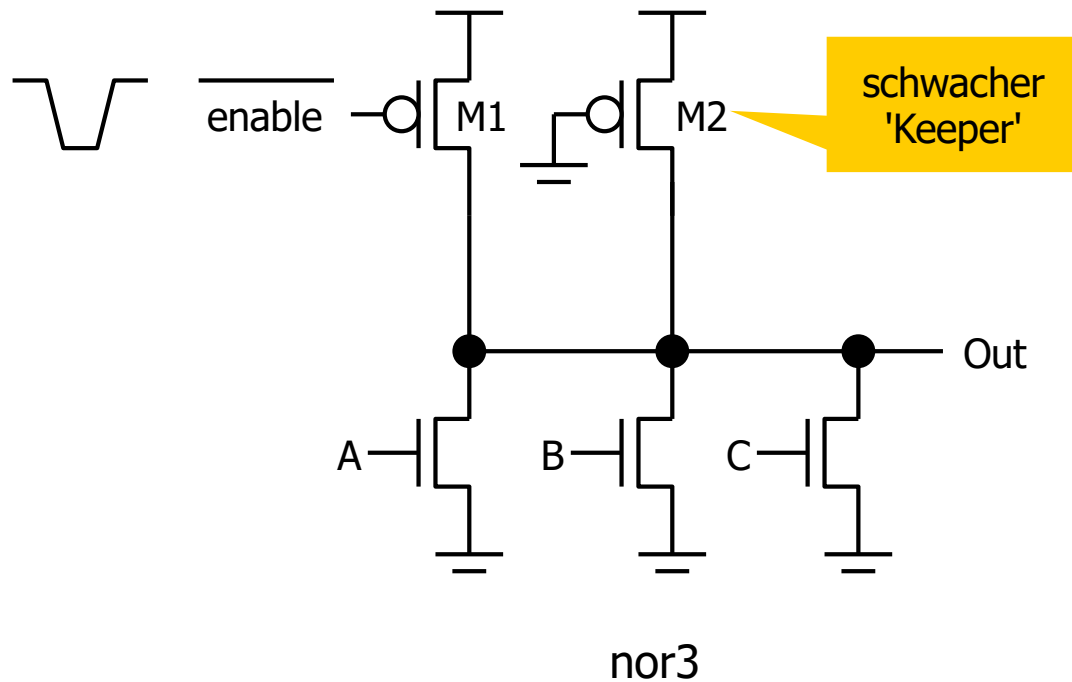
Exercise

- Create a NMOS inverter with a (long) PMOS load with a bias voltage from a current mirror
- Create a chain of two such inverters and transport a rectangle signal
- Vary the current in the PMOS in a wider range

- Observe the various effects on the output signal

Pseudo NMOS mit Adaptiver Last

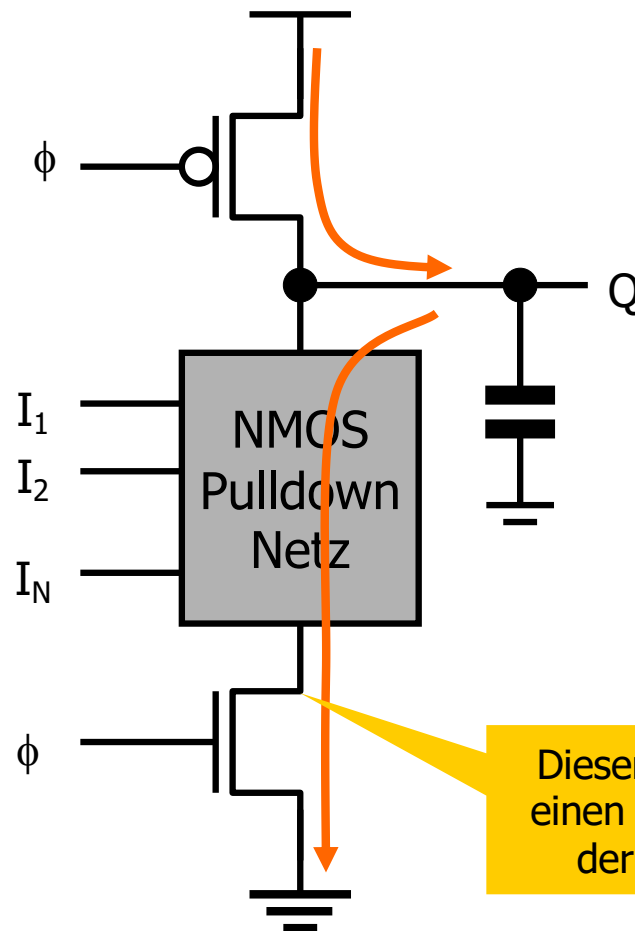
- M1 wird aktiv eingeschaltet, wenn die Logik benötigt wird
- Im ‚stand-by‘ - Modus sorgt ein schwacher M2 (kleines W/L) für statische High-Levels
- Reduziert z.B. den DC Stromverbrauch



DYNAMISCHE LOGIK

Dynamische Logik mit 'Precharge'

2 Phasen:



Precharge ($\phi=0$):

Ausgang wird auf 1 vorgeladen

Evaluation ($\phi=1$):

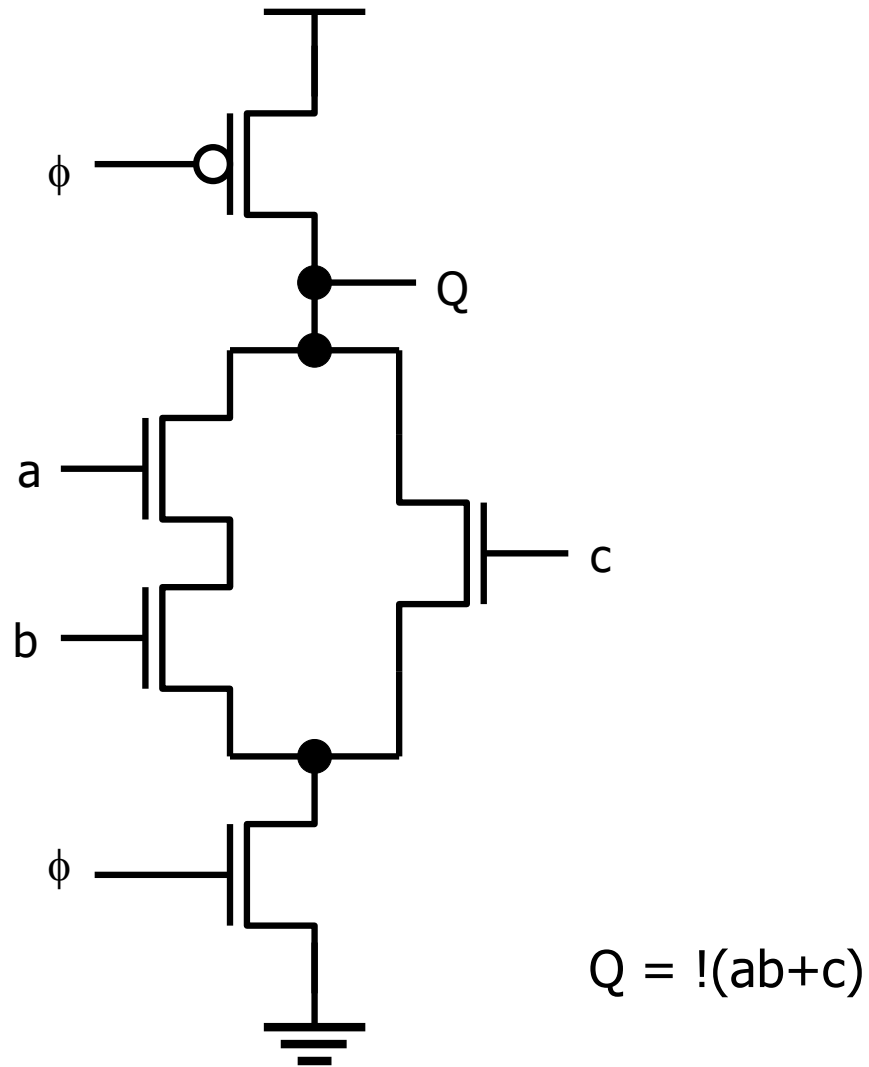
Ausgang geht auf 0, falls pulldown aktiv.

Nur EIN Übergang nach 0 ist möglich.

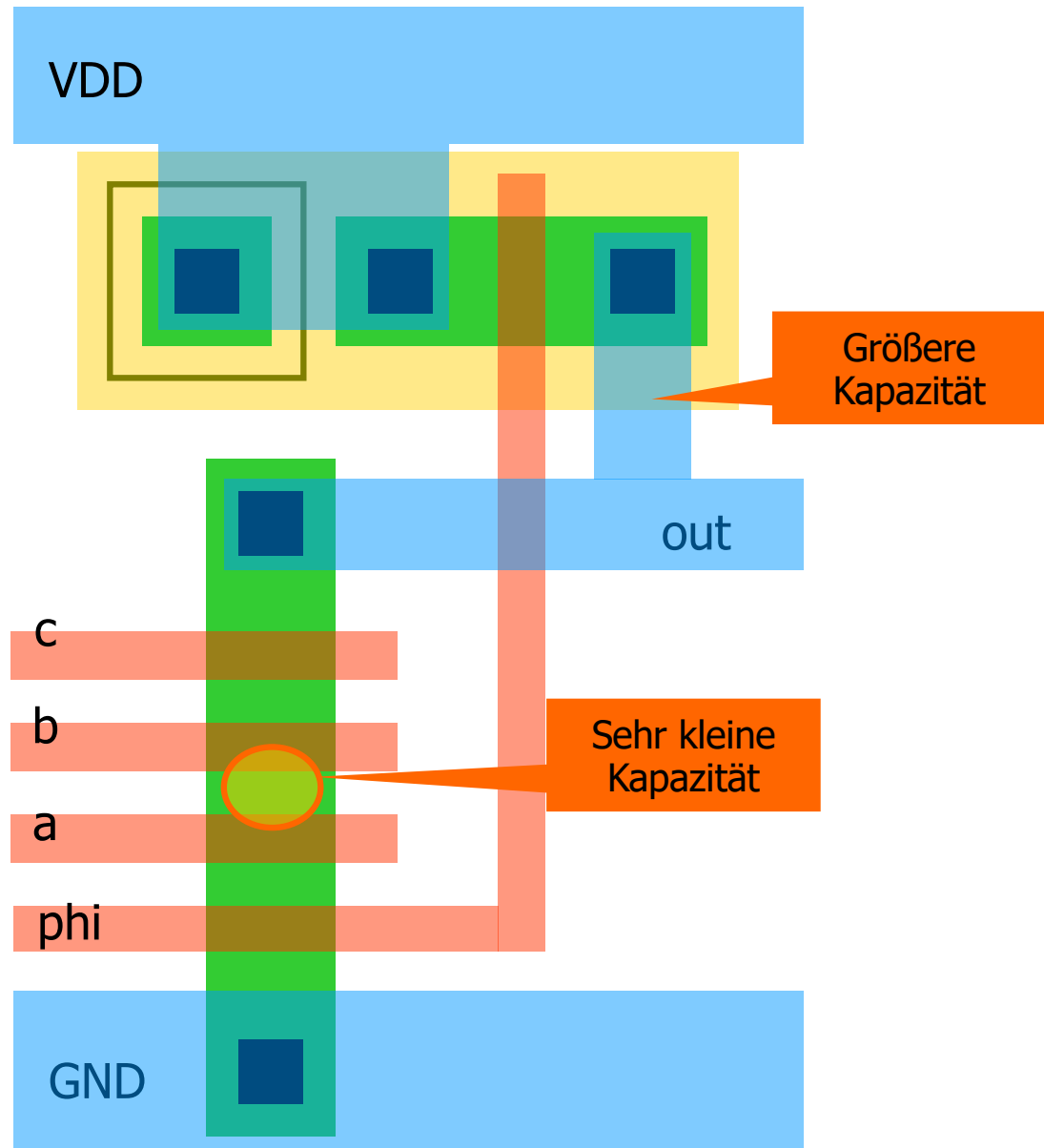
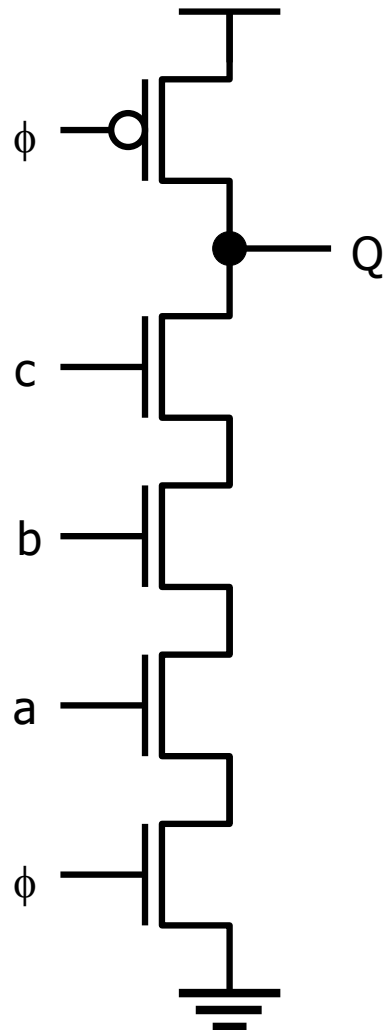
Dieser NMOS unterbindet einen Querstrom während der Precharge-Phase

Ähnlich: Komplementäre Schaltung mit PMOS pullup-Netz.

Beispiel Dynamische Logik



Layout eines dynamischen NAND3 Gatters



Dynamische Logik

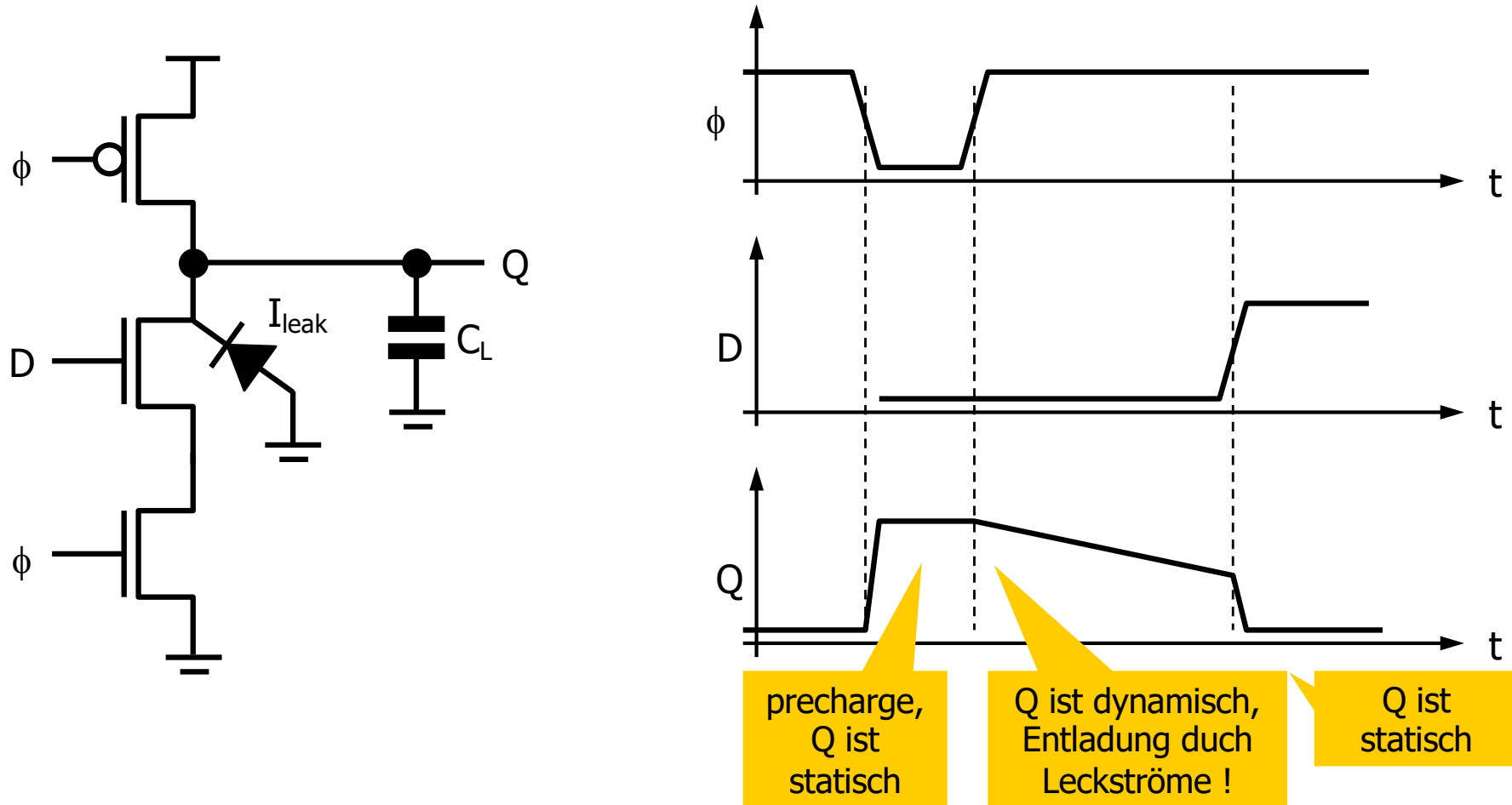
Vorteile:

- Nur N+2 Transistoren
- Sehr einfache Layouts
- Ratio-Less: Die Größe der Transistoren ist (relativ) unkritisch
- Kein statischer Stromverbrauch
- Schnelle Evaluation, da nur NMOS Kapazitäten
- Full swing Logik

Nachteile:

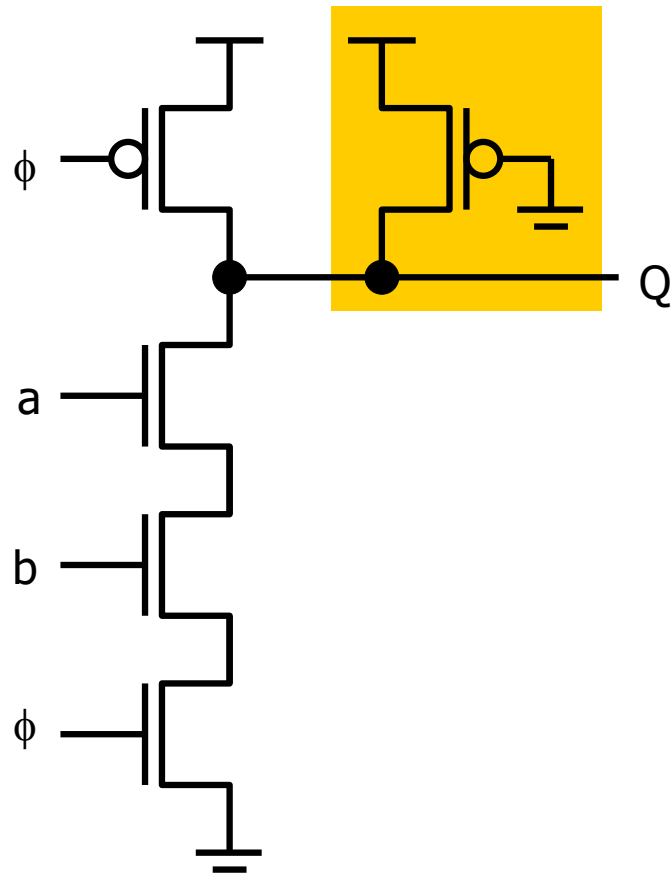
- Kleiner unterer Störabstand ($\sim V_{TN}$)
- Benötigt Taktsignal
- Nicht kaskadierbar....
- Probleme durch Leckströme...
- Probleme durch Ladungsumverteilung...
- Probleme durch Ladungsinjektion...

Problem: Leckströme

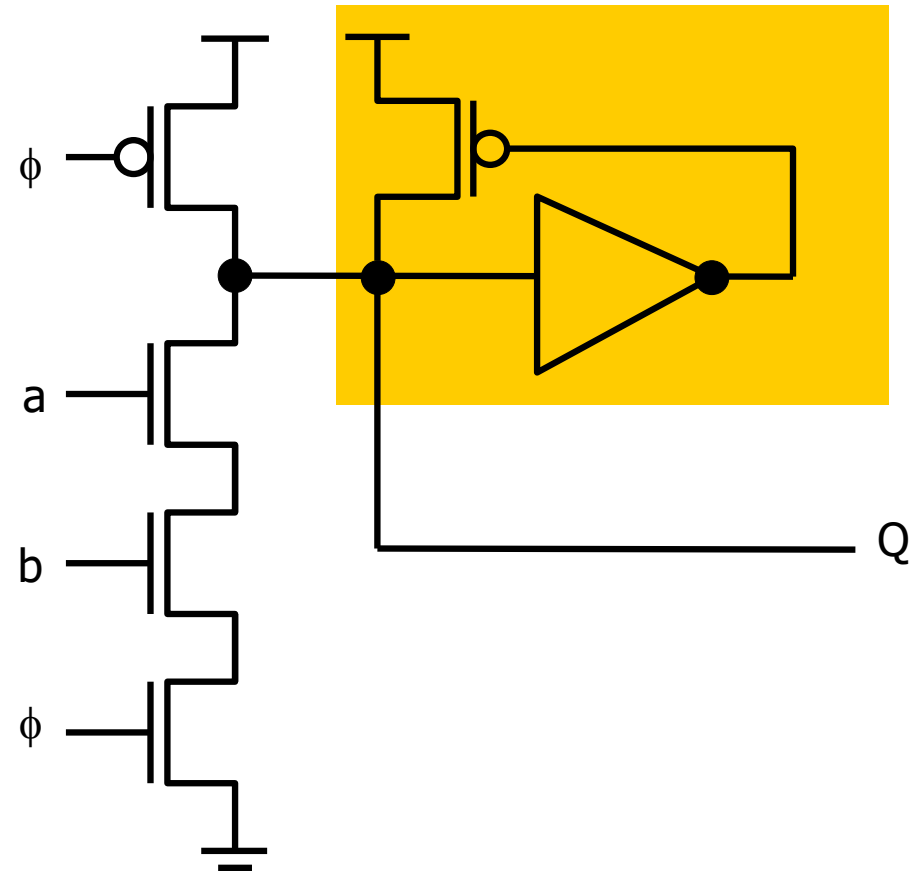


- Während der Evaluationsphase ist der Ausgang z.T. dynamisch.
- Die Gesamtkapazität C_L kann dann durch Leckströme I_{leak} mit $\Delta U/\Delta T = I_{leak}/C_L$ entladen werden.
- Hauptanteile an I_{leak} kommen von den Draindioden der MOS und Subthreshold-Strömen

Maßnahmen gegen Leckströme



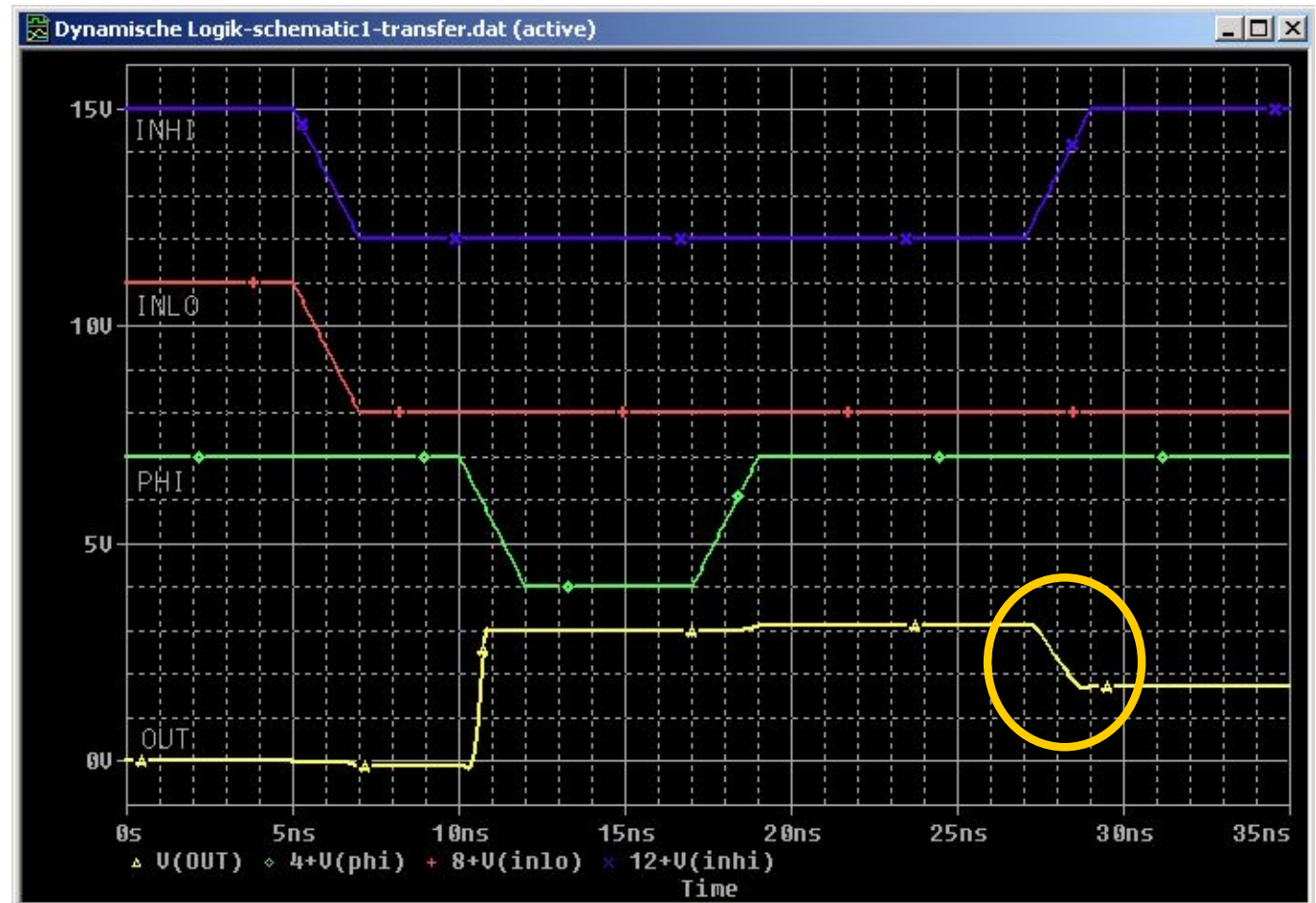
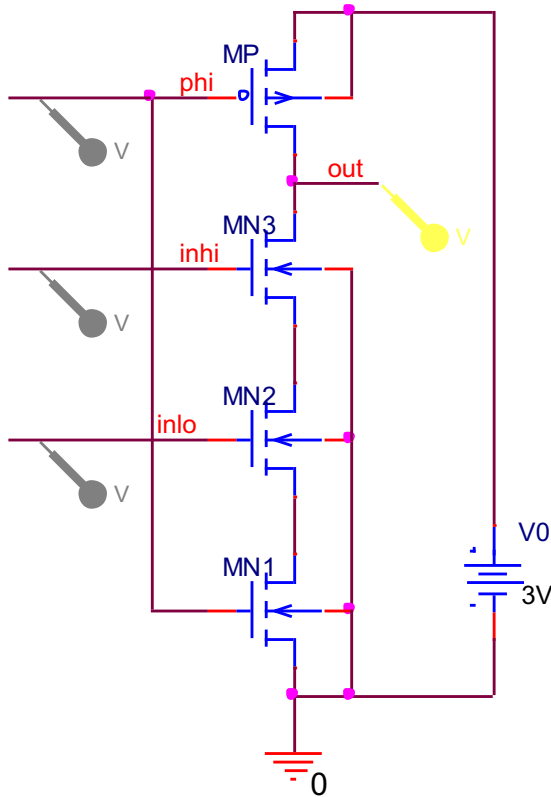
- Schwacher PMOS hält Ausgang auf VDD
- Statischer Stromverbrauch während Evaluation
- Ähnlich clocked pseudo-NMOS



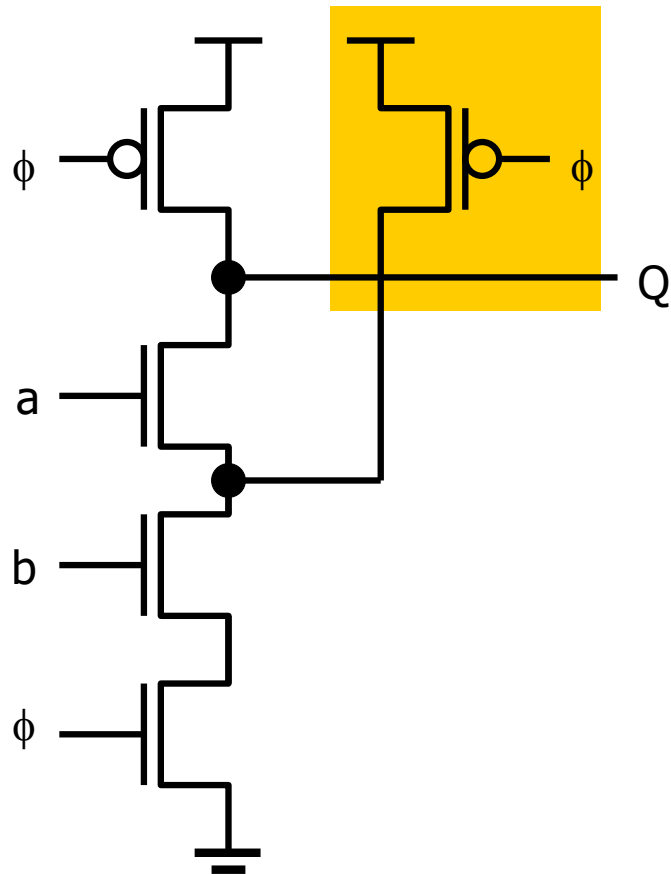
- Wie ‚Level Restore‘ bei Pass Gates
- Solange Q Null ist wird der dynamische Knoten mit einem PMOS nach 1 gezogen
- Wird der dynamische Knoten in der Evaluation nach 0 entladen, so ‚lässt der PMOS los‘

Problem: 'Charge redistribution'

- Die Ausgangsspannung kann unerwünscht einbrechen, wenn ein entladener parasitärer Kondensator (hier zwischen MN2 und MN3) mit dem Ausgang verbunden wird.
- Dies kann passieren, obwohl der Ausgang auf high bleiben sollte (Beispiel NAND, Eingang 'inlo' ist 0)

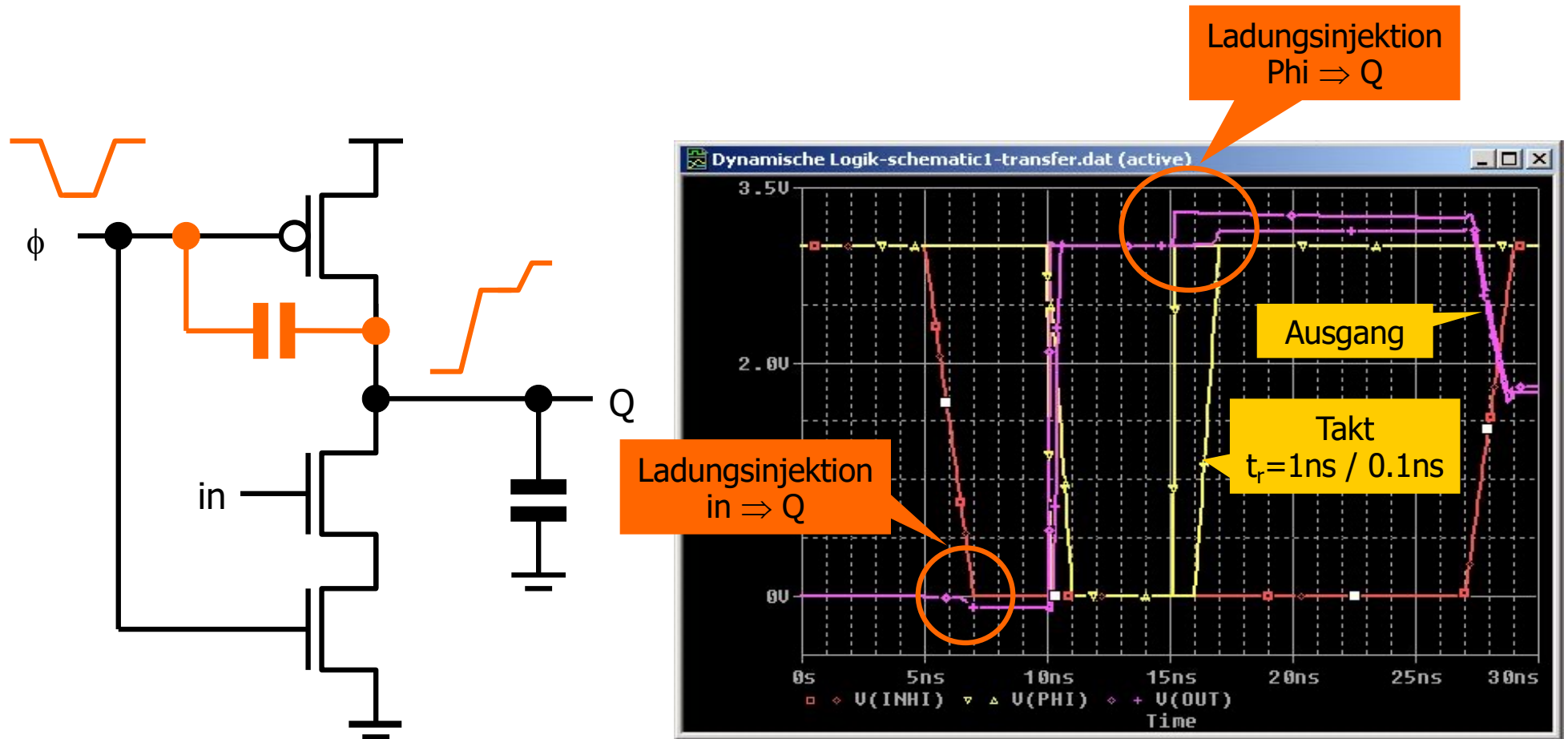


Gegenmaßnahme: Vorladen von Zwischenknoten



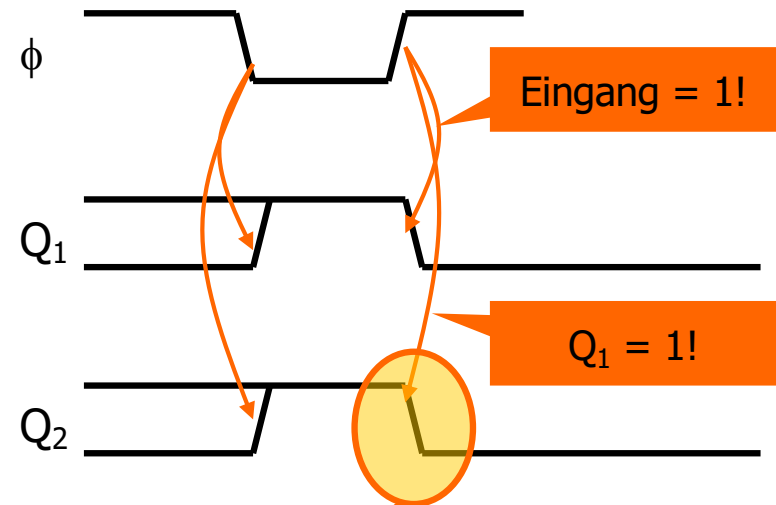
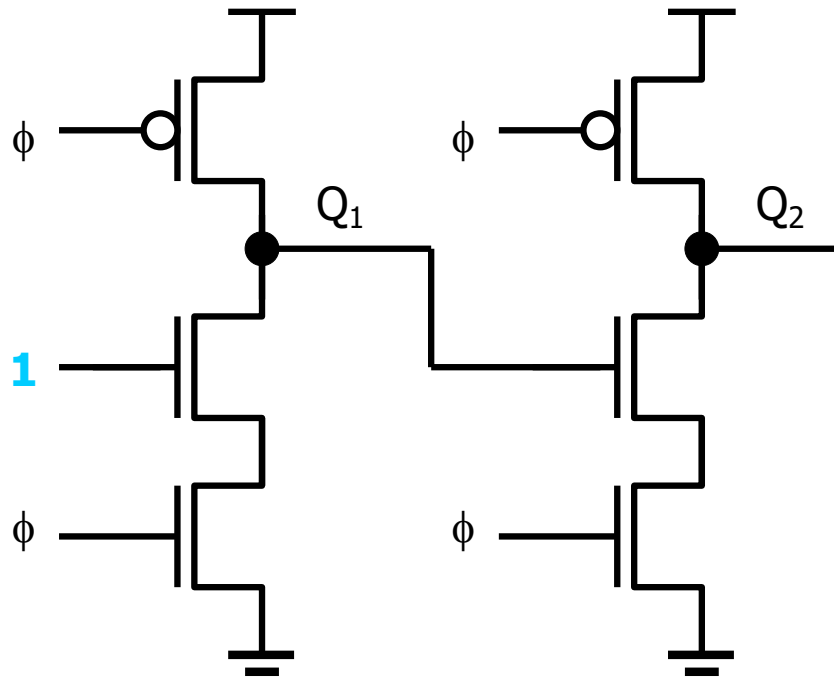
- Alle Zwischenknoten (oder zumindest kritische mit viel Kapazität) werden vorgeladen.
- Nachteil: totale Kapazität steigt, mehr Transistoren, höhere Last an Taktleitung
- Alternativ: Zwischenkapazitäten sehr klein halten (Diffusion minimal, keine Kontakte)

Problem: 'Clock Feedthrough'

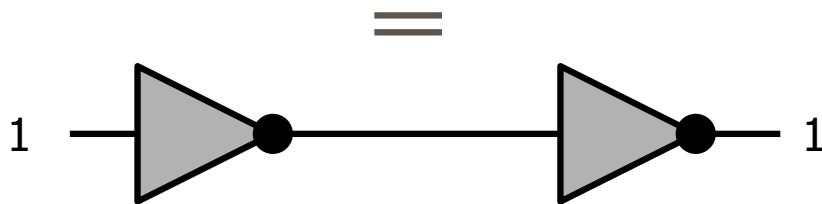


- Das Taktsignal kann durch die Gate-Drain Kapazität des PMOS auf den Ausgang durchkoppeln
- Die Stärke des Effekts hängt von der Anstiegszeit des Taktsignals und den Kapazitätsverhältnissen ab.
- Der Effekt kann positiv ausgenutzt werden, da er den Störabstand erhöht. Etwas charge redistribution stört dann nicht mehr...

Problem: Kaskadieren von N-Typ Precharge Gates



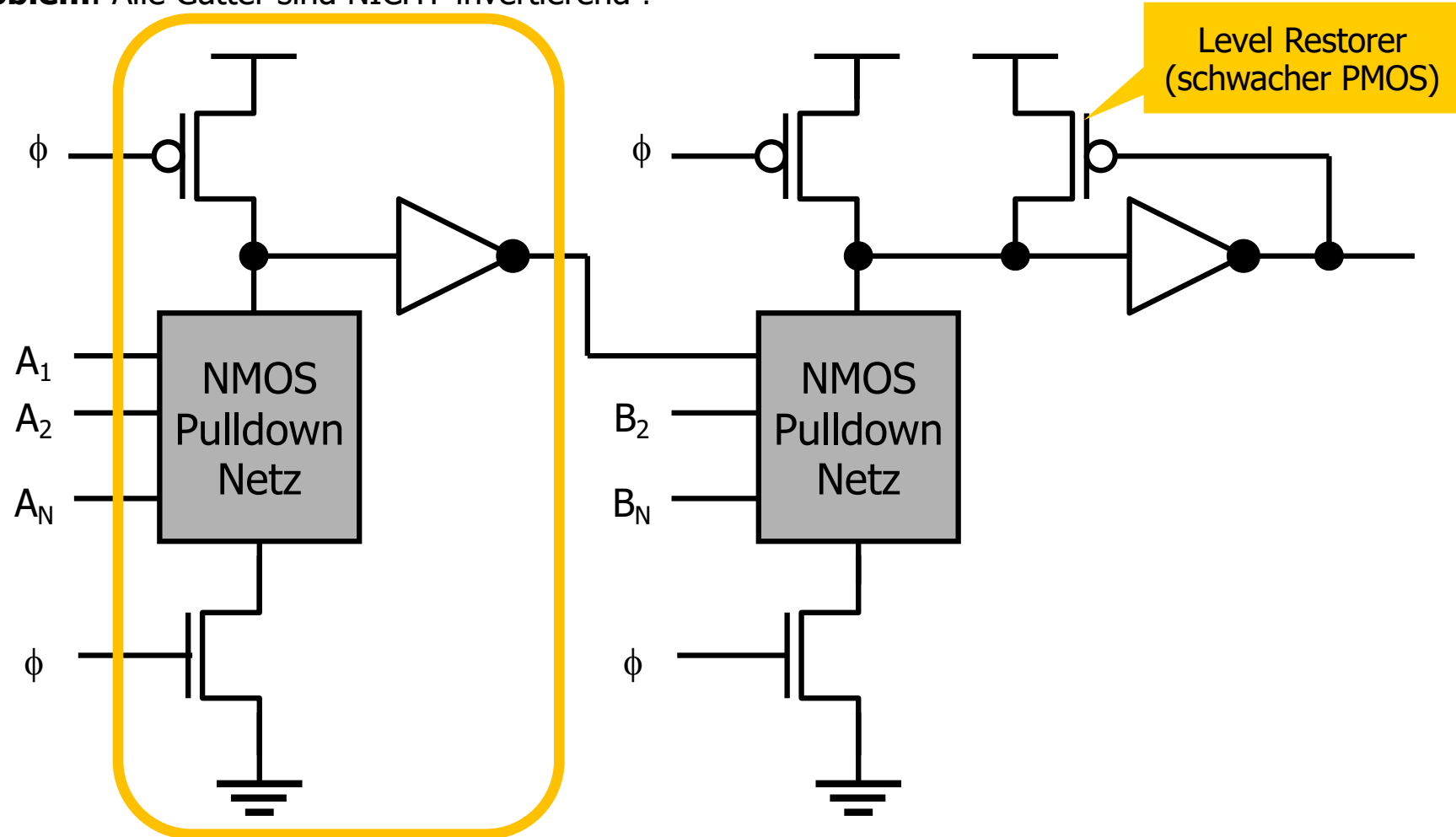
Q_2 müßte jetzt wieder auf 1 springen (Eingang des Gatters = 0), das geht in der dynamischen Logik aber nicht!



- Direktes Kaskadieren von N-Typ precharge Gattern funktioniert **NICHT**, weil die Einsen in den vorderen Gattern (nach dem Precharge) die hinteren Gatter umschalten. Diese können dann nicht mehr den korrekten Wert annehmen, da pro precharge Phase nur EIN 1-0 Übergang möglich ist.
- Lösungen: Inverter einfügen, N- und P- Stufen abwechseln

Lösung1: Domino Logik

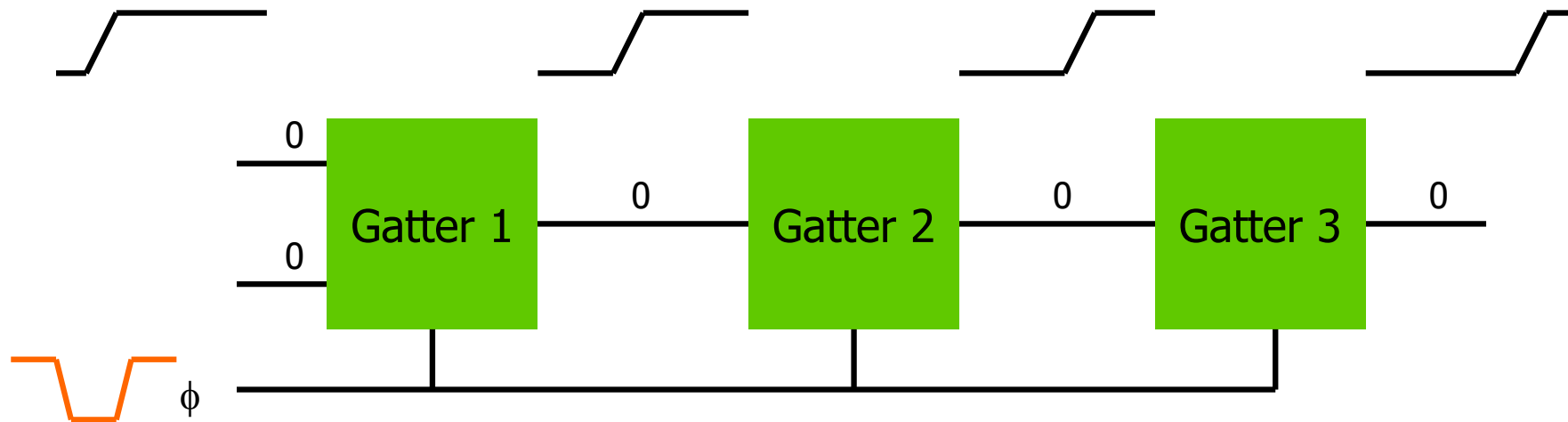
- Füge einen **Inverter** hinter jede N-Stufe
- Die Ausgänge sind dann in der Precharge-Phase NULL und es gibt kein Problem
- **Problem:** Alle Gatter sind NICHT-invertierend !



Beobachtung: Den unteren NMOS in den hinteren Stufen kann man weglassen (bei $\Phi=0$ sind alle $B_i=0$)!

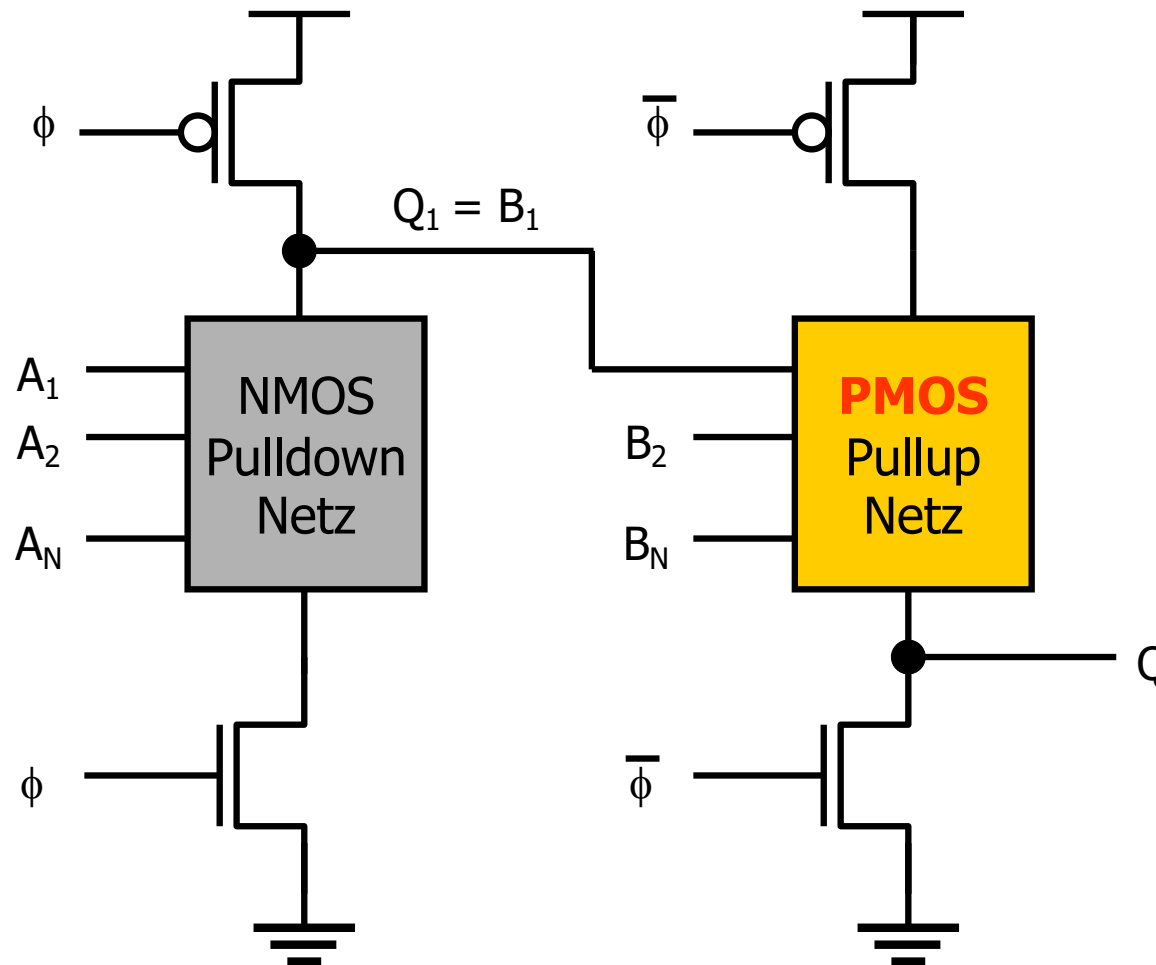
Domino Effekt

- Das Signal propagiert von links nach rechts durch und wirft eine Zelle nach der anderen um ('Domino')

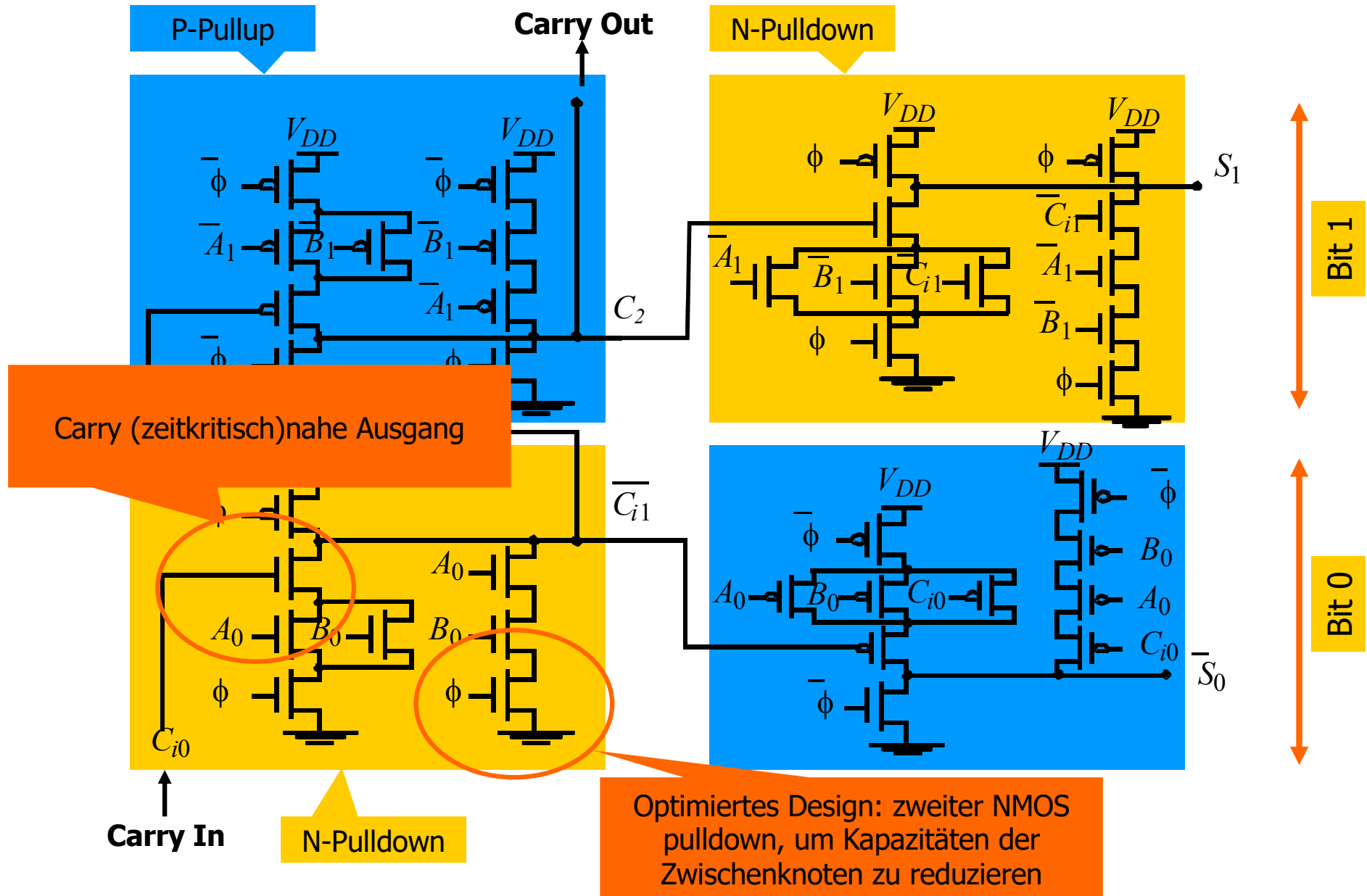


Lösung2: NP-CMOS

- Abwechselnd **N-Stufen** und **P-Stufen** benutzen
- Precharge der P-Stufen benutzt den inversen Takt.
- Nachteil: PMOS-Teile sind langsamer

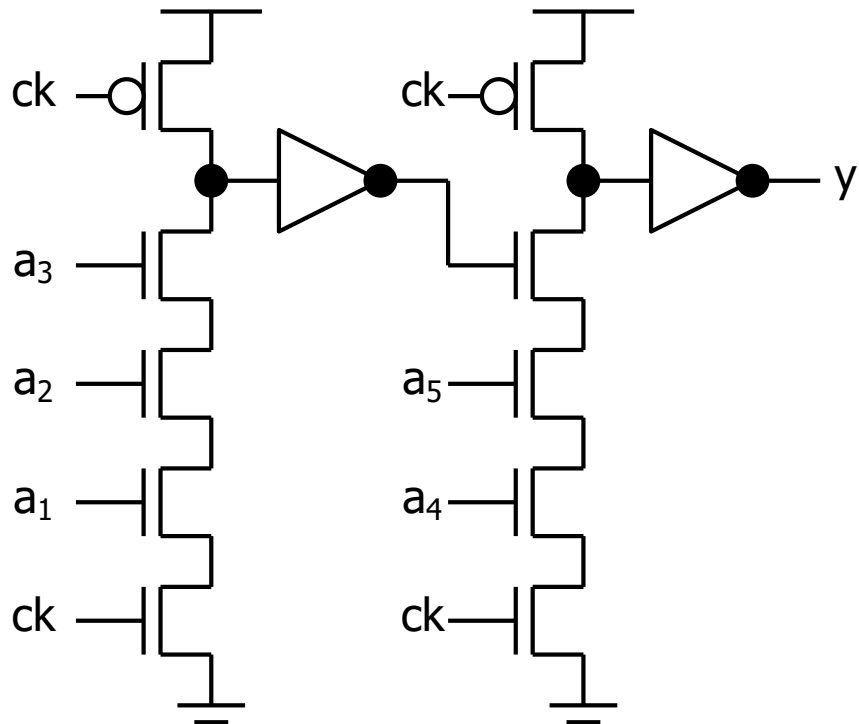


Beispiel: NP CMOS Volladdierer (2 Bit)



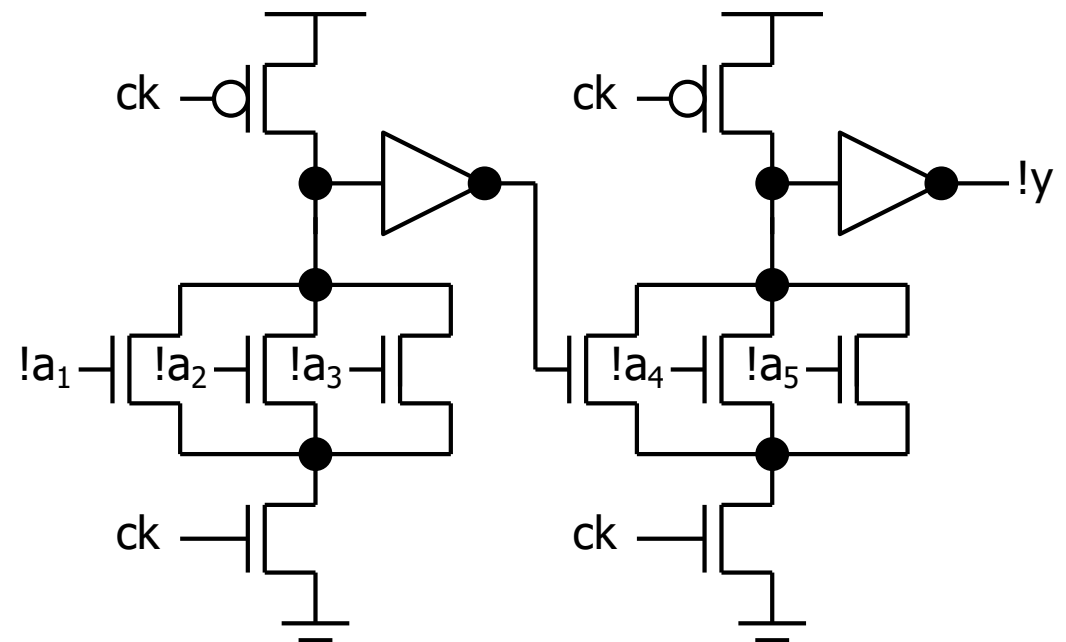
Beispiel: breites UND Gatter (Adressen-Decoder)

- Betrachte UND Gatter mit 5 Eingängen. $y = a_1 a_2 a_3 a_4 a_5 = !(a_1 + a_2 + a_3 + a_4 + a_5)$



y **springt** nur auf **1**, wenn alle **a_i** **high** sind

Implementierung als Domino UND

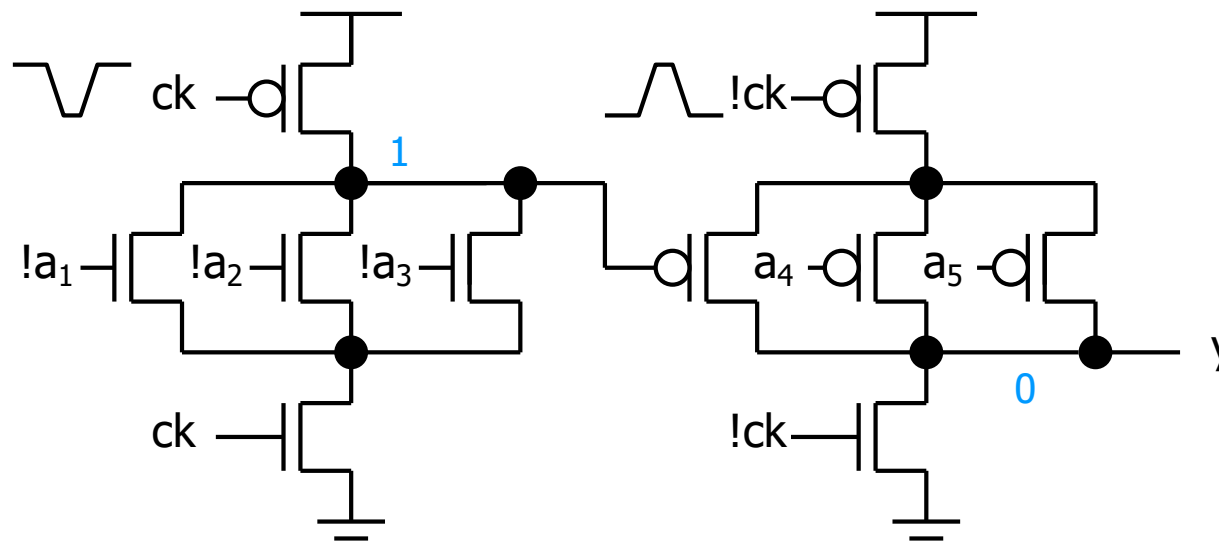


!y **bleibt** nur auf **0**, wenn alle **!a_i** **low** sind

Implementierung als Domino ODER
Weniger NMOS in Serie: BESSER

Beispiel: breites Gatter (Adressen-Decoder)

- Zum Vergleich: Implementierung als NP ODER. Die Logik im zweiten Gatter wird umgedreht



y **bleibt** nur auf **0**, wenn **!a₁-!a₃ low** sind **a₄-a₅ high** sind.

- Sehr kompakt. Kann sehr einfach auf sehr viele Eingänge erweitert werden.

Exercise

- Implement such an NP cascaded OR gate with 3 + 3 inputs.
- Simulate it

Zusammenfassung dynamische Logik

Konzepte:

- Domino oder NP Logik
- Level Restorer zur Reduktion von Leckströmen – 'pseudo-statisch'
- Evtl. Precharge von internen Knoten!

Vorteile:

- Sehr kurze Durchlaufzeit in der Evaluationsphase (bis zu 2x schneller)
- Full swing
- Nicht 'ratioed'
- Sehr einfache und kompakte Layouts

Nachteile:

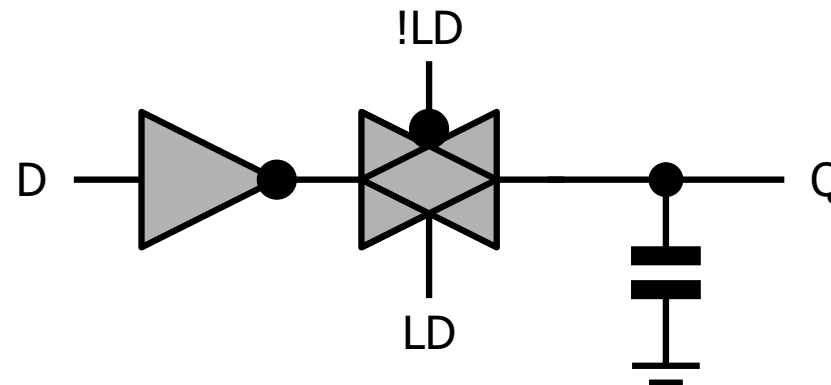
- Nur nicht-invertierende Funktionen
- Benötigt Takt
- (Speicherzeit begrenzt)
- Störabstand gering
- Vorsicht vor Ladungsumverteilung (charge-redistribution)
- Höherer Stromverbrauch durch Takt (bis zu 2x höher).
Hängt auch von der Polarität der Implementierung ab.

FLIPFLOPS IN DYNAMISCHER LOGIK

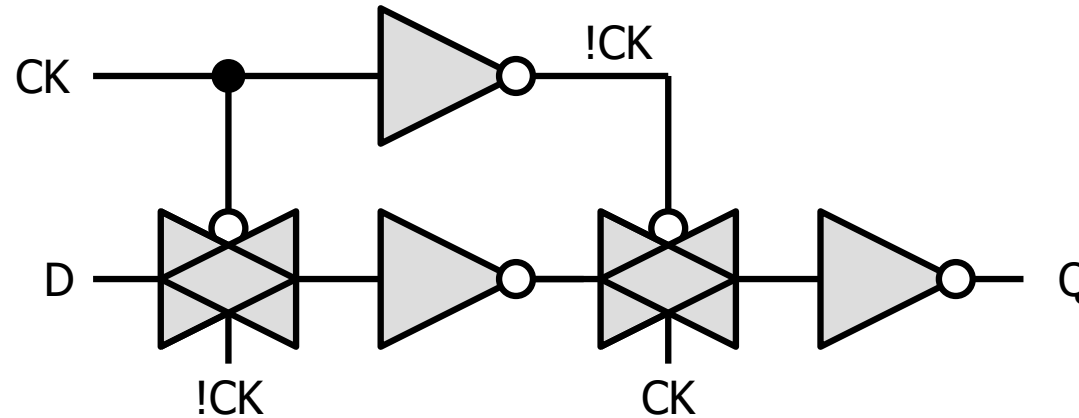
Dynamische Logik

- In dynamischer Logik werden Knoten für kurze Zeit nicht getrieben.
- Sie behalten ihren Spannungszustand durch die Kapazität des Knotens.
- Die Speicherzeit ist beschränkt durch Leckströme (Dioden, Subthreshold Leitung der MOS)

- Beispiel: Dynamisches Latch:



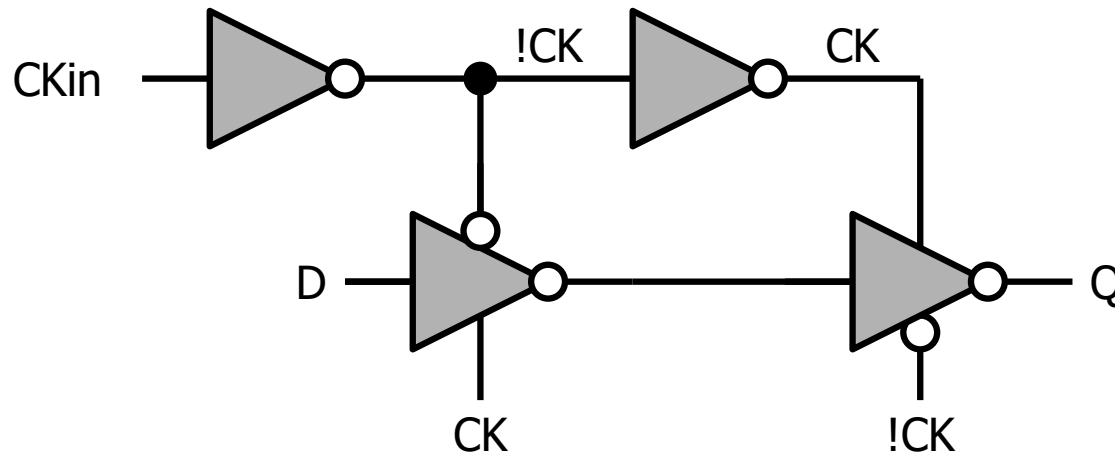
Dynamisches Ein-Phasen-FF



- Vorsicht: CK und !CK müssen genau komplementär sein, sonst ist das FF für kurze Zeit transparent. Beste Lösung: interne Erzeugung von !CK.
- Vorsicht: Klappt nicht bei langsamer Anstiegszeit von CK. Lösung: CK puffern (zweiter Inverter).

Ein-Phasen FF: C²MOS

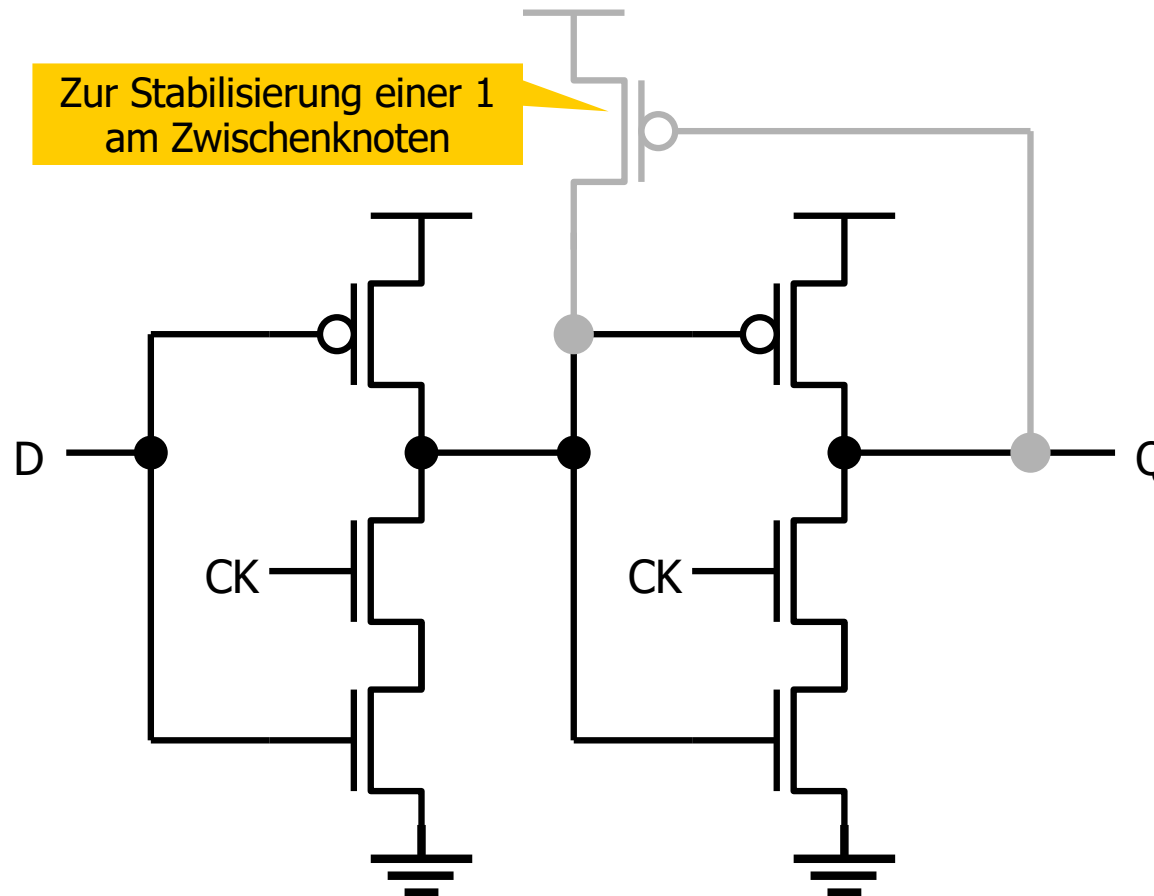
- Wie vorher, aber mit Gated Invertern.
- Layout manchmal angenehmer



- CK und !CK müssen eigentlich genau komplementär sein, sonst wird das LATCH evtl. transparent
- Bei genauem Nachdenken nicht so kritisch

'Echtes' Ein-Phasen dynamisches LATCH

- Vermeidet das Problem des Überlappens von CK und !CK in FFs
- Funktion: - Bei CK=1 (transparente Phase) hat man zwei in Serie geschaltete Inverter.
- Bei CK=0 behält der Ausgang Q seinen Zustand dynamisch bei:
Eine 1 bei Q kann nicht zu 0 werden, weil der hintere CK - NMOS sperrt.
Eine 0 bei Q kann nicht durch den PMOS zu einer 1 werden, weil der Zwischenknoten nicht 0 werden kann.



- Gibt es ähnlich mit invertiertem Ausgang und/oder invertierter Clock (PMOS in die Mitte)

Exercise

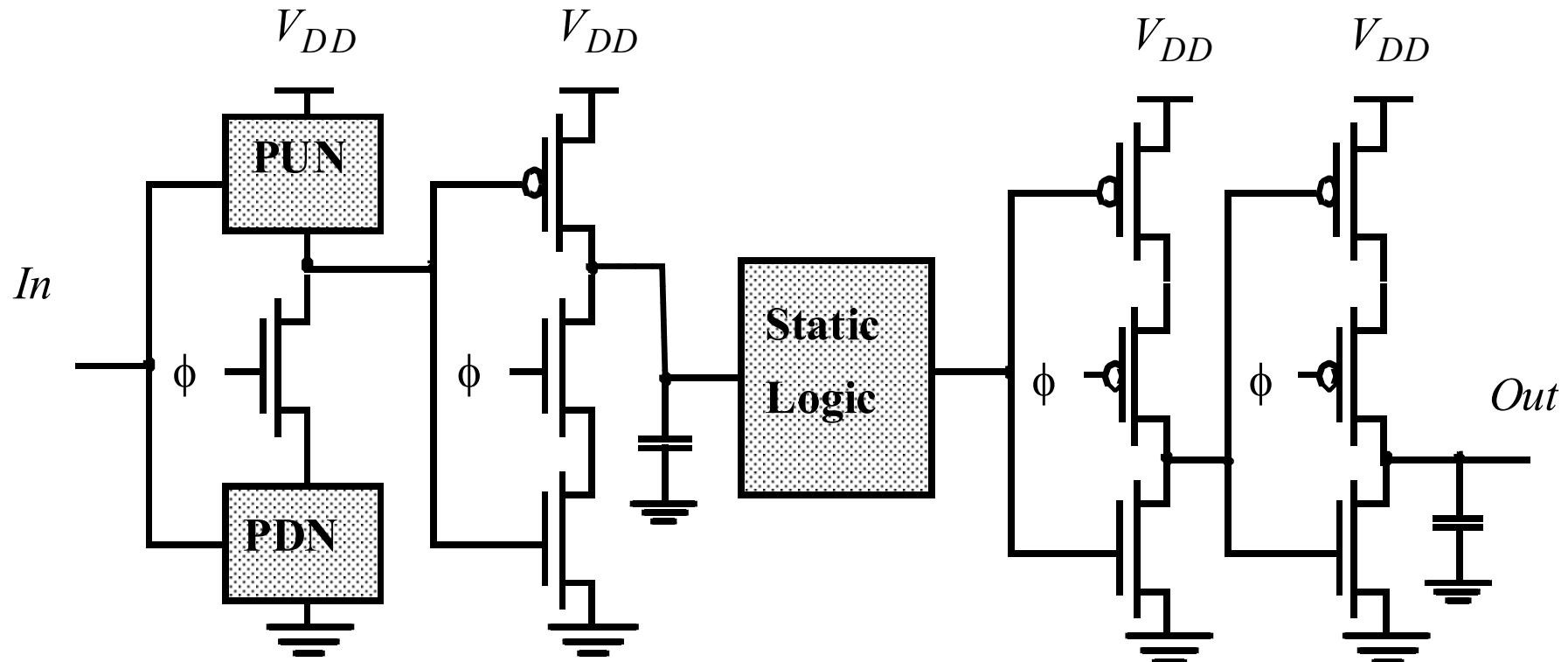
- Implement and simulate the true single phase latch of the previous page.
- Verify that it is tracking for $CK = 1$ and opaque in the 'hold' phase

- Observe what happens to the intermediate node when D is changing in the hold phase

- Implement the same functionality with PMOS switch transistors so that it is transparent for $CK = 0$. You must rearrange the MOS!

TSPC - True Single Phase Clock Logic

- Ein Flipflop kann aus zwei solchen Latches aufgebaut werden (N-Typ und P-Typ)
 - Logik kann in die PU/PD Netze eingebaut werden
 - Logik kann (natürlich) auch zwischen die Stufen eingebaut werden

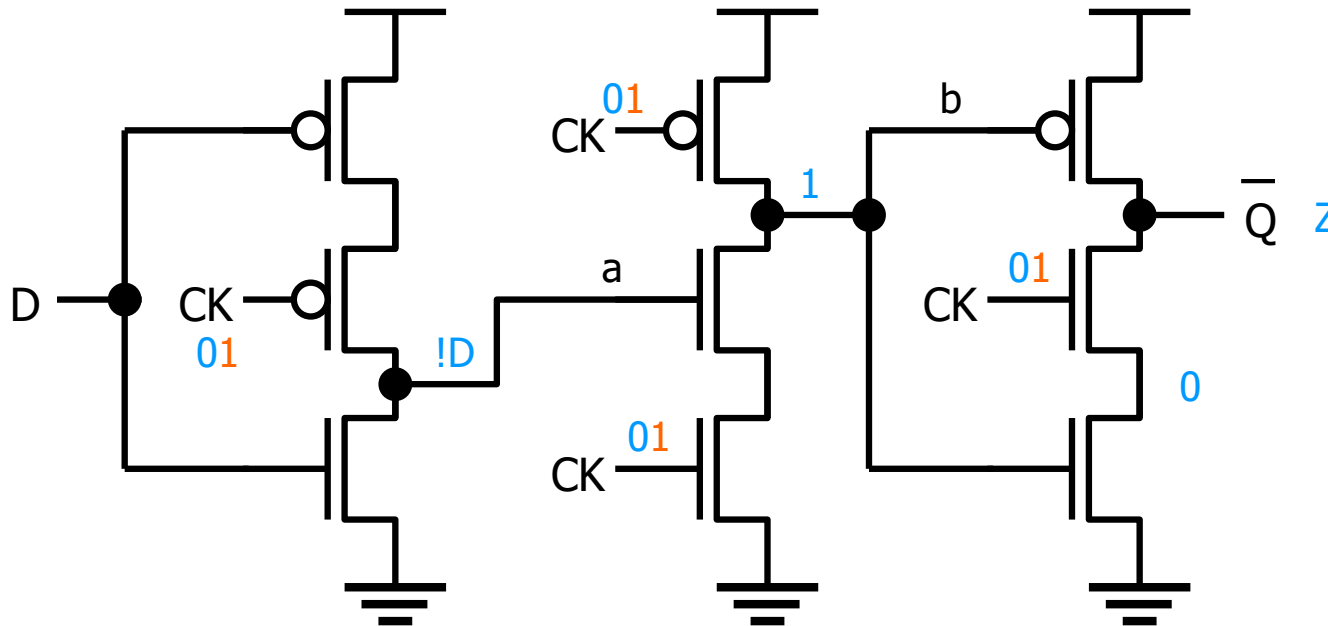


**Including logic into
the latch**

**Inserting logic between
latches**

True Single Phase Clocking: Svensson - FF

- Durch geschicktes Hintereinanderschalten von single-Phase Latches bekommt man ein sehr schnelles und kompaktes Single-Phase-**FF** (hier auf positive Flanke getriggert) (Yuan, Svensson, 1989)



- Funktion:
 - bei **CK=0** ist der Ausgang hochohmig. Am Punkt a liegt !D an. Punkt b ist auf 1.
 - bei **CK=1** und **D=0**: a ist 1. b springt auf 0. Ausgang geht auf 1 (**=!D**).
Falls D jetzt noch auf 1 schaltet, geht a auf 0. b wird tristate 0, der Ausgang bleibt dabei unverändert!
 - bei **CK=1** und **D=1**: a ist 0, der mittlere Teil blockiert. b bleibt auf 1, Ausgang geht auf 0 (**=!D**).
Falls D jetzt noch auf 0 schaltet, wird a tristate.
- Operation sehr trickreich
- Dynamisch! Vorsicht vor Ladungsumverteilung und Ladungsinjektion. Sehr riskantes Design
- In die erste Stufe kann zusätzlich Logik eingebaut werden

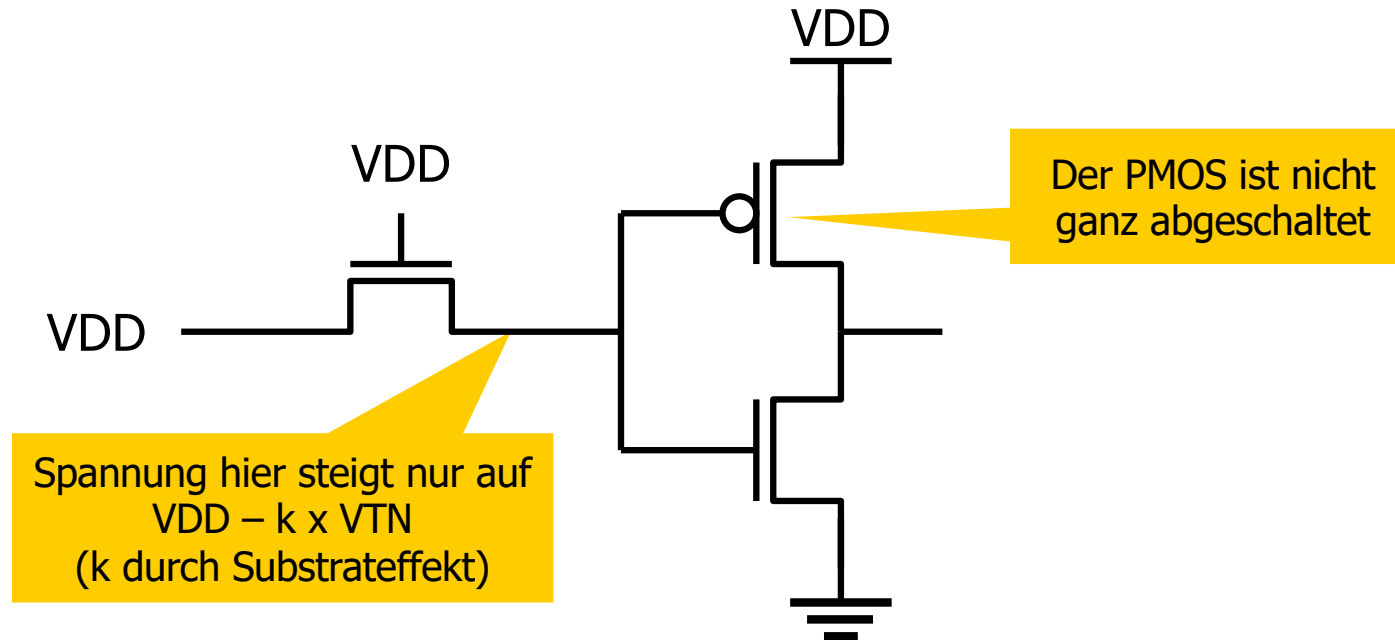
Exercise

- Simulate the Svensson Flipflop
- Check many (all) typical cases (D changes after or before edge)

PASS GATE LOGIK

Pass-Transistor Logik: Nur NMOS Schalter ?

- IDEE: 1/0 werden seriell durch MOS (Drain-Source) ‚durchgereicht‘
- Gleiches Problem wie bei CMOS:
NMOS Transistoren machen ‚gute‘ Nullen, aber ‚schlechte‘ Einsen.
- Genauer:

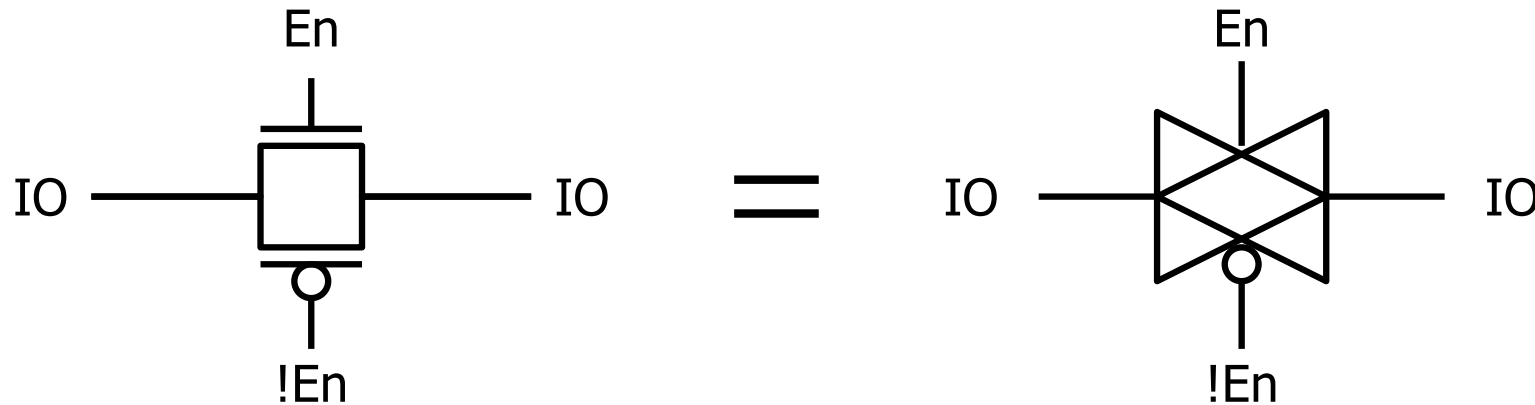


- Problem: Statischer Stromverbrauch, Verlust an Störabstand
- Daher: NMOS und PMOS parallel schalten

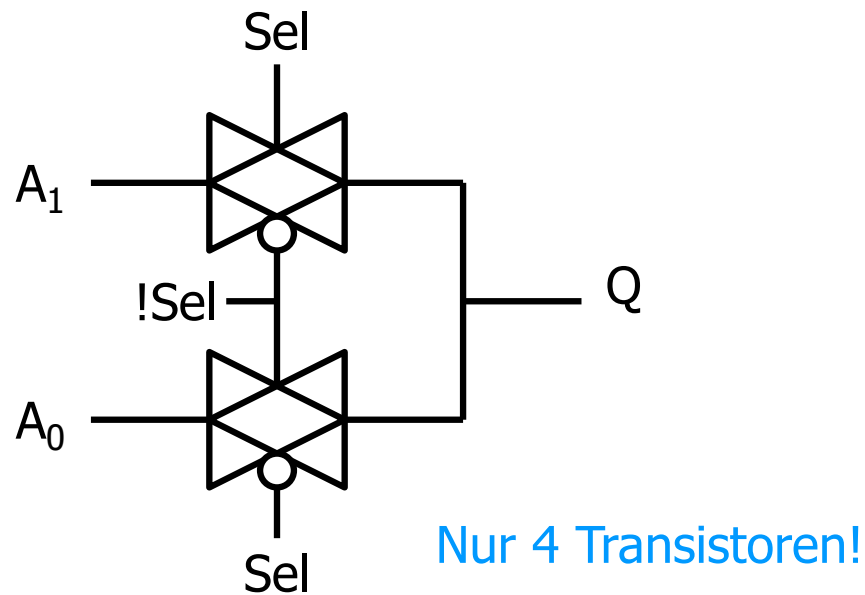
- Aber: Wenn man weiß, was man tut, ist u. U. auch nur ein NMOS ok...

Pass Gate Logik

- Elementarer Block: **Transmission Gate** aus PMOS und NMOS. Symbol wie ein Absperrhahn.



- Beispiel Multiplexer



Einschub: Widerstand eines MOS

- Vom Widerstand kann man nur im **linearen Bereich** sprechen, d.h. wenn

$$0 < \mathbf{V_{DS}} \ll V_{GS} - V_T$$

- In diesem Bereich gilt:

$$I_D = K W/L [(V_{GS} - V_T) V_{DS} - 1/2 V_{DS}^2] = K W/L [V_{GS} - V_T - 1/2 V_{DS}] V_{DS}$$

- In der [] - Klammer kann man $1/2 V_{DS}$ gegenüber $V_{GS} - V_T$ vernachlässigen, also gilt näherungsweise

$$I_D \sim K W/L (V_{GS} - V_T) V_{DS}$$

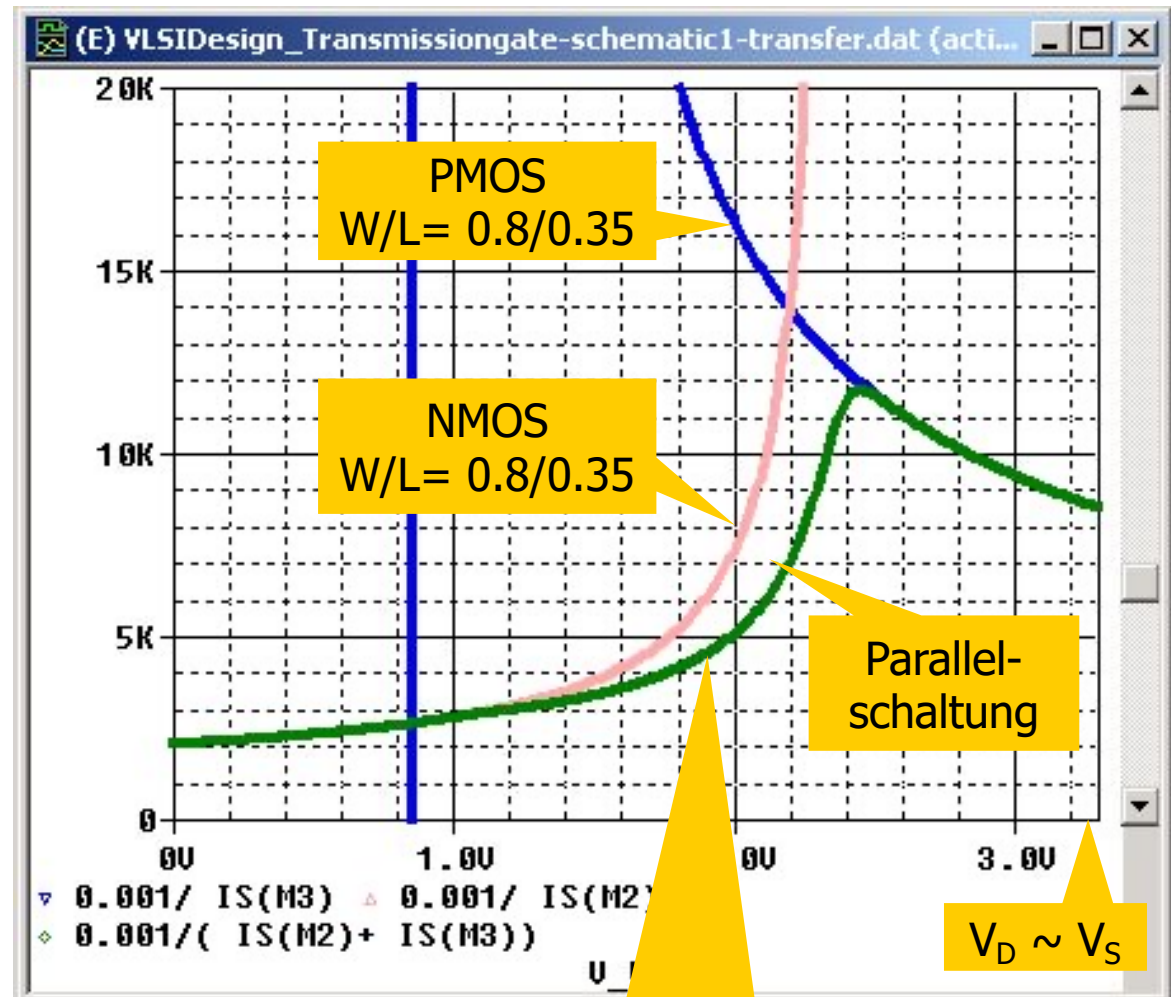
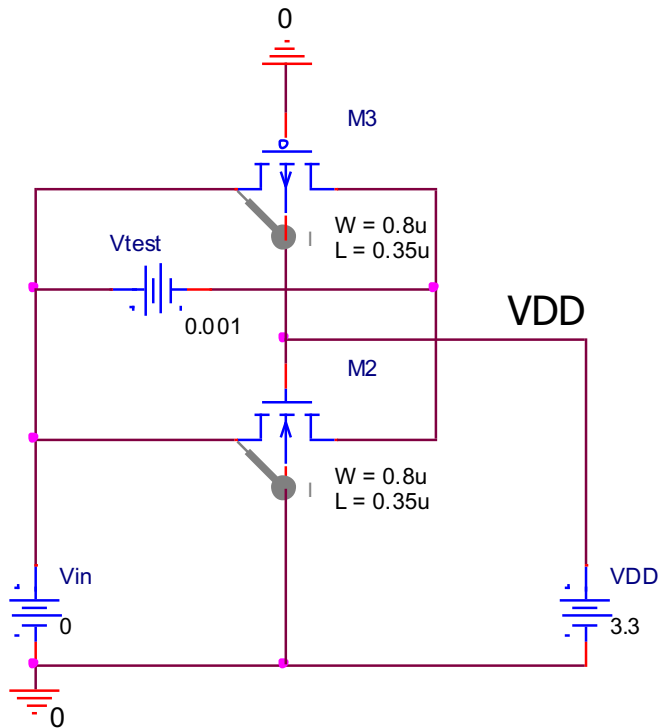
- Also ist der Leitwert des Kanals

$$g_{DS} = I_D / V_{DS} \sim K W/L (V_{GS} - V_T), \quad \text{der Kehrwert ist der Widerstand}$$

- Einsetzen von Werten einer 0.35 μm Technologie ergibt (für einen minimalen NMOS bei 3.3V)

$$g_{DS} \sim 175 \mu\text{A}/\text{V}^2 \times 0.8\mu\text{m}/0.35\mu\text{m} \times (3.3\text{V} - 0.5\text{V}) \sim \mathbf{1\text{mA}/\text{V}, \text{d.h. } 1\text{k}\Omega}$$

Simulation: Widerstand Transmission Gate

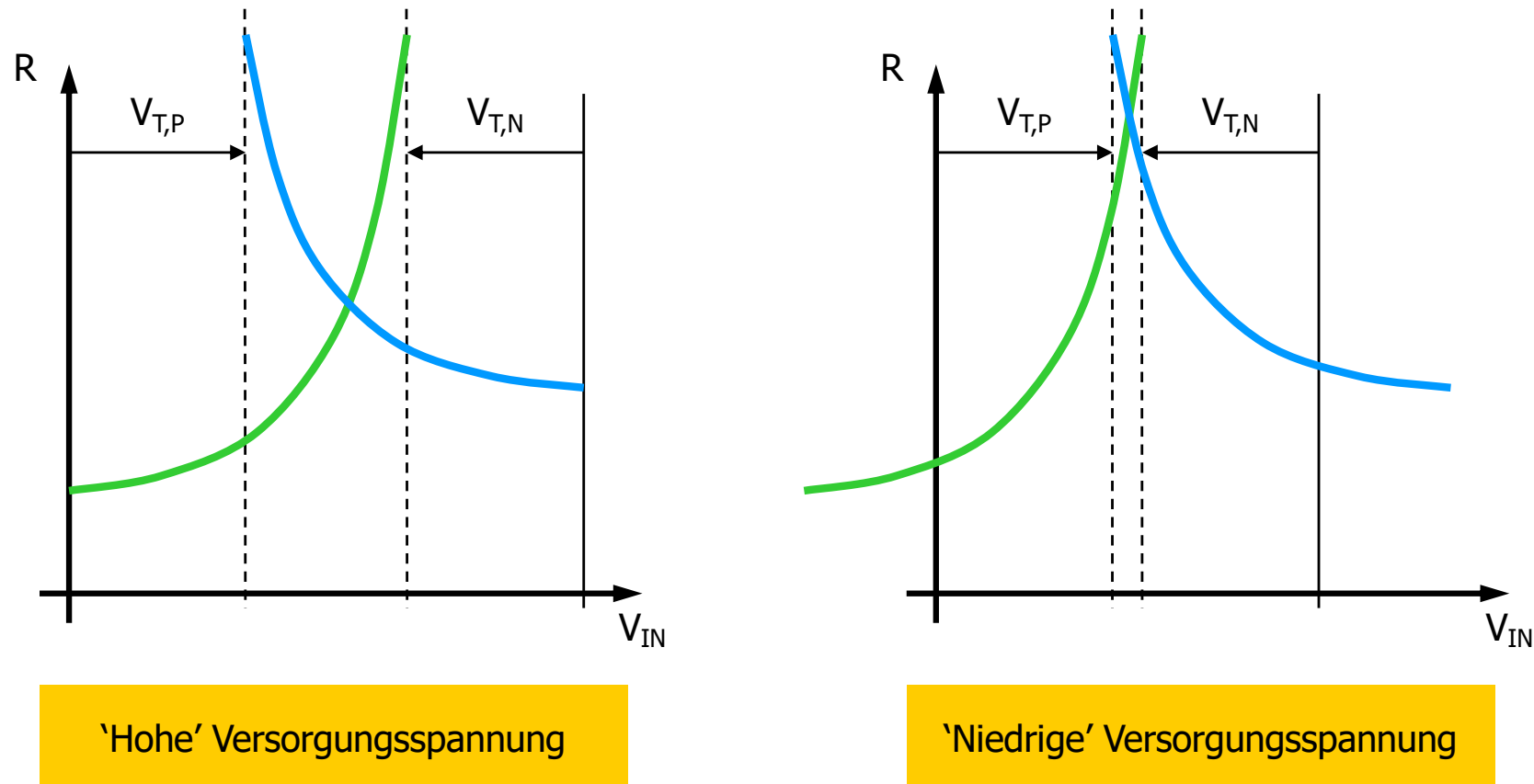


- Auch hier muss der PMOS breiter als der NMOS gemacht werden
- Typ. RC : $3\text{k}\Omega \times 3\text{fF} = 10\text{ps}$, aber $(3 \times 3\text{k}\Omega) \times (3 \times 3\text{fF}) = 100\text{ps}$!

Durchlasswiderstand hat die Größenordnung $1\text{k}\Omega$

Problem bei niedrigen Versorgungsspannungen

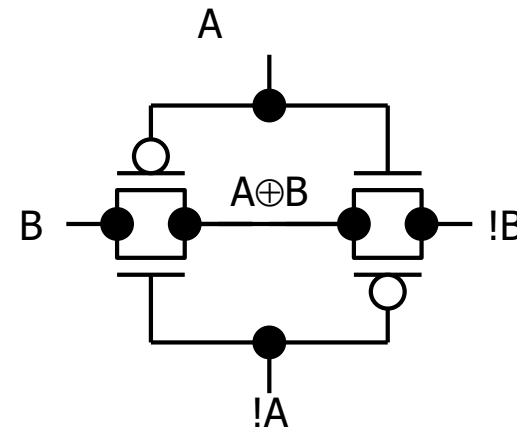
- Bei modernen Prozessen mit sehr kleinen Gatelängen sinken die Versorgungsspannungen, die Schwellenspannungen bleiben aber etwa gleich (wegen Subthreshold Leakage!)
- Dies führt zu hohen Widerständen von Transmission-Gates im mittleren Spannungsbereich:



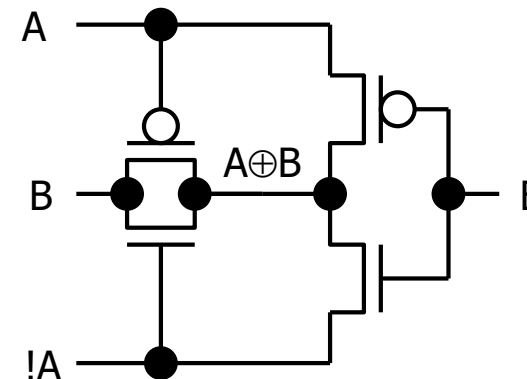
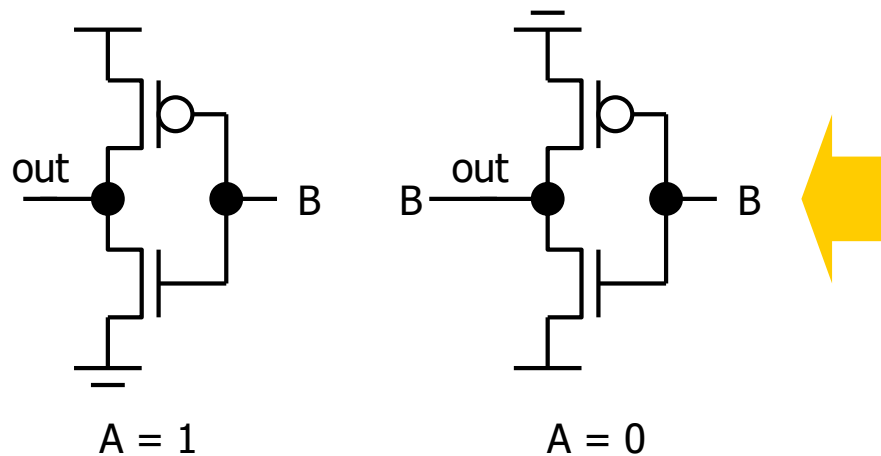
- Wenn $V_{DD} < k_1 V_{T,N} + k_2 |V_{T,P}|$ wird, funktionieren die Schalter nicht mehr! (k_1, k_2 für Substrateffekt)

XOR mit Pass Gate Logik

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



Möglichkeit 1:
MUX
4 MOS + 2 Inverter



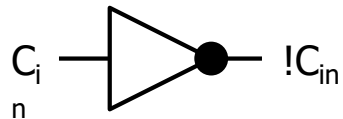
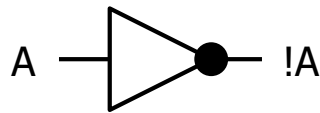
Möglichkeit 2:
Raffiniert!
4 MOS + 1 Inverter

Vorsicht: Bei der Abschätzung der Geschwindigkeit muss man die MOS in den Schaltungen berücksichtigen, die die Signale A und B erzeugen!

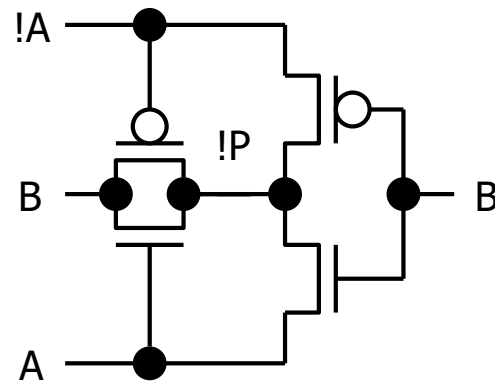
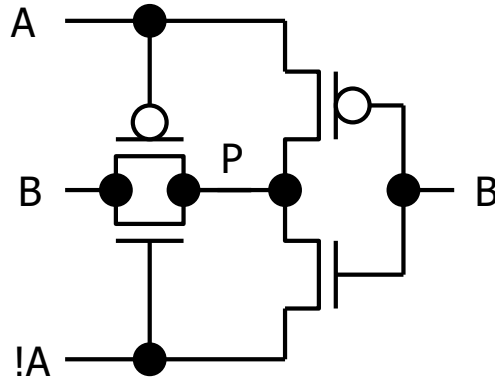
Volladdierer mit Transmission Gates

C_{in}	A	B	C_{out}	S	P
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	0

orange: $C_{out} = C_{in}$ ($P=1$)
 weiss: $C_{out} = A$ (oder B)

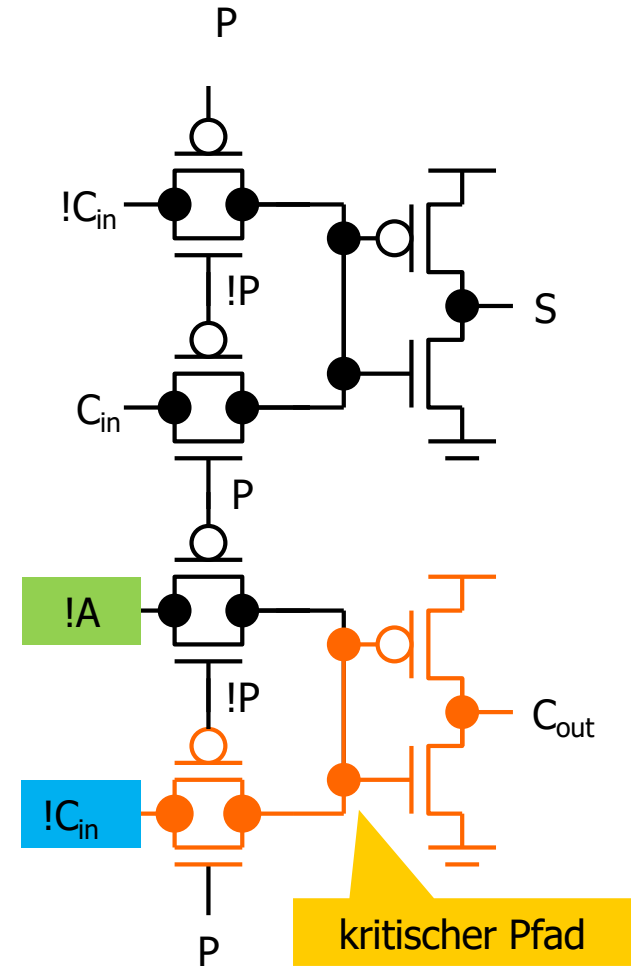


Vorbereitung



XORs: $P=A \oplus B$

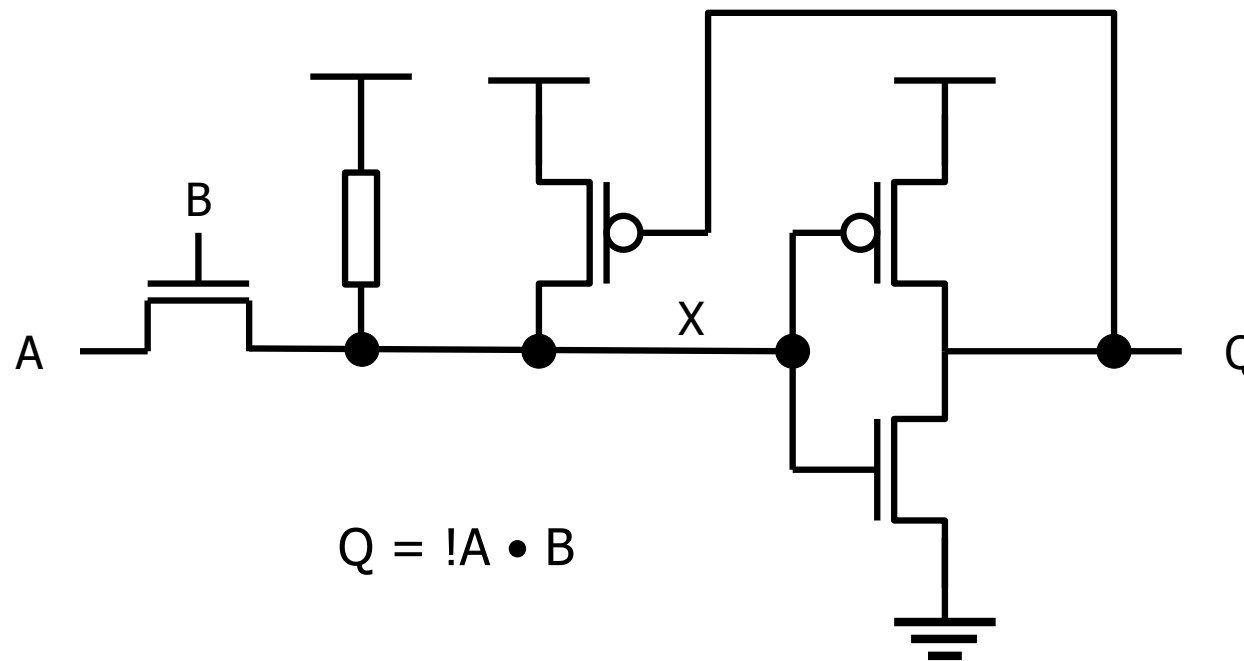
(Ein Inverter für !P wäre langsamer)
 ($P = \text{'propagate'}$)



MUXe: $S=P \oplus C_{in}$, $C_{out}=..$

'NMOS –Only' Transmission Gate Logik: ‚Level restore‘

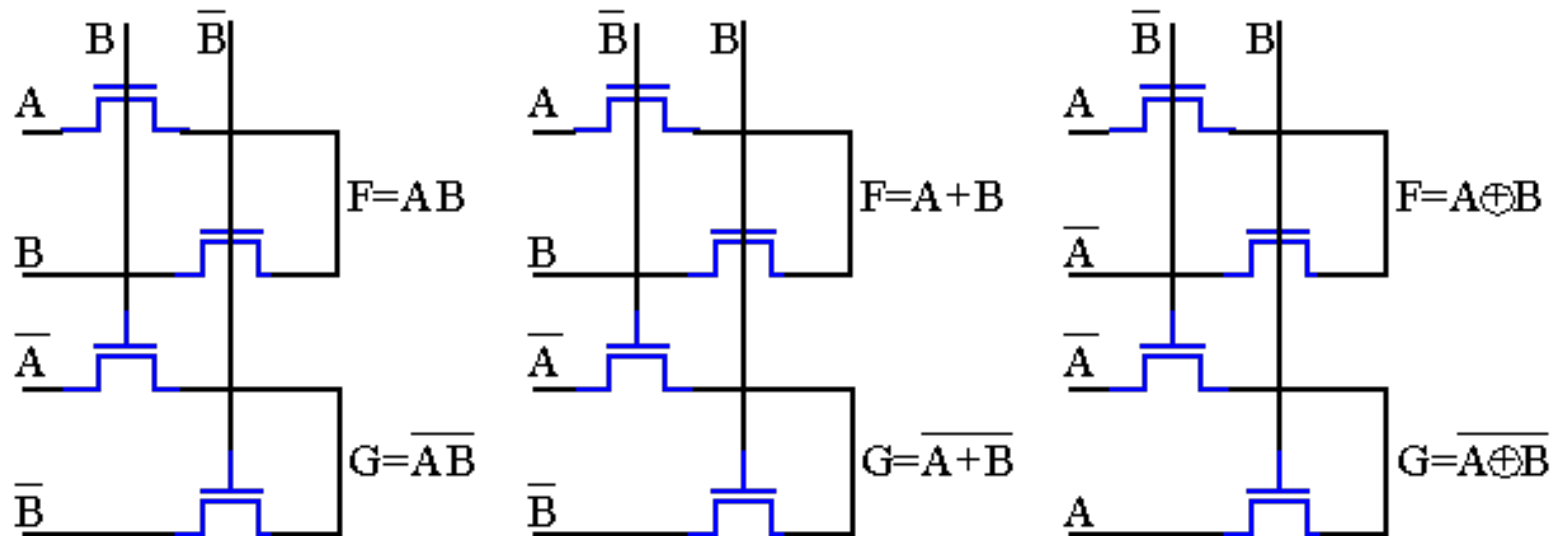
- ‚NMOS-Only‘ geht nicht direkt wegen des schlechten High-Pegels, statischen Stromverbrauchs etc. (s.o.)
- Trick: Nutze einen PMOS **Level-Restorer** um aus einer ‚schlechten 1‘ ein VDD Niveau zu machen.
⇒ ‚full swing‘ wird wiederhergestellt:



- Vorsicht: der PMOS muss richtig dimensioniert werden, damit auch ‚schwache‘ Nullen bei A (z.B. aus seriellen NMOS Transistoren) den Knoten X noch nach Masse ziehen können

Complementary Pass Logic (CPL)

- Idee: Liefere mit jedem Signal auch das Inverse Signal.
- Dann ist immer eine Null dabei, die man 'gut' mit NMOS verarbeiten kann.



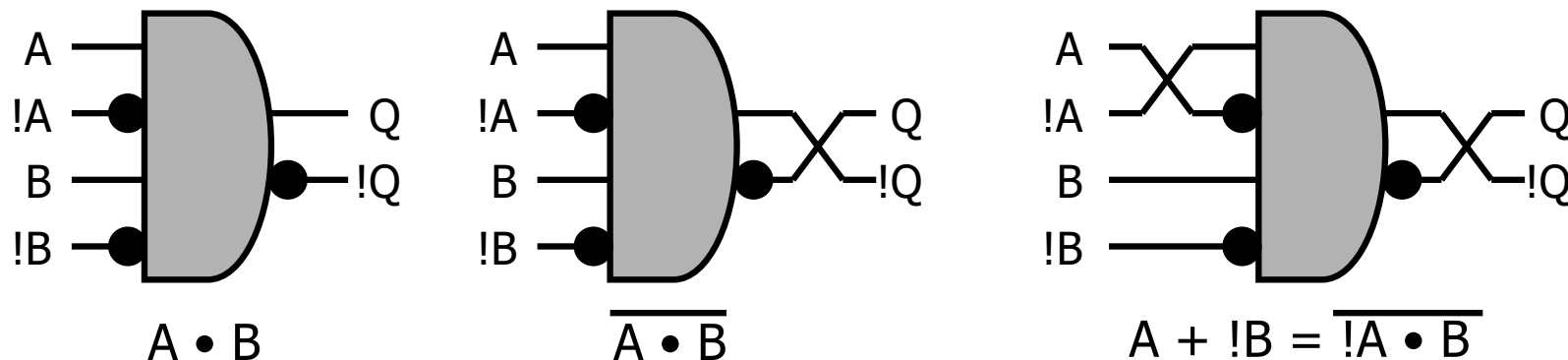
DIFFERENTIELLE CMOS LOGIK

Differenzielle Logik

- Einige Logikfamilien erzeugen einen Ausgang Q und das Inverse $!Q$.
- Sie benötigen neben den Eingängen I_1, \dots, I_N meist auch die Inversen $!I_1, \dots, !I_N$

Vorteile:

- Es werden viel weniger Logikfunktionen benötigt, da durch Vertauschen von Signalen die Negation erreicht wird. **Für Fan-In = 2 werden nur 2 Gatter benötigt.** Frage: Wieviele für Fan-In = 3?



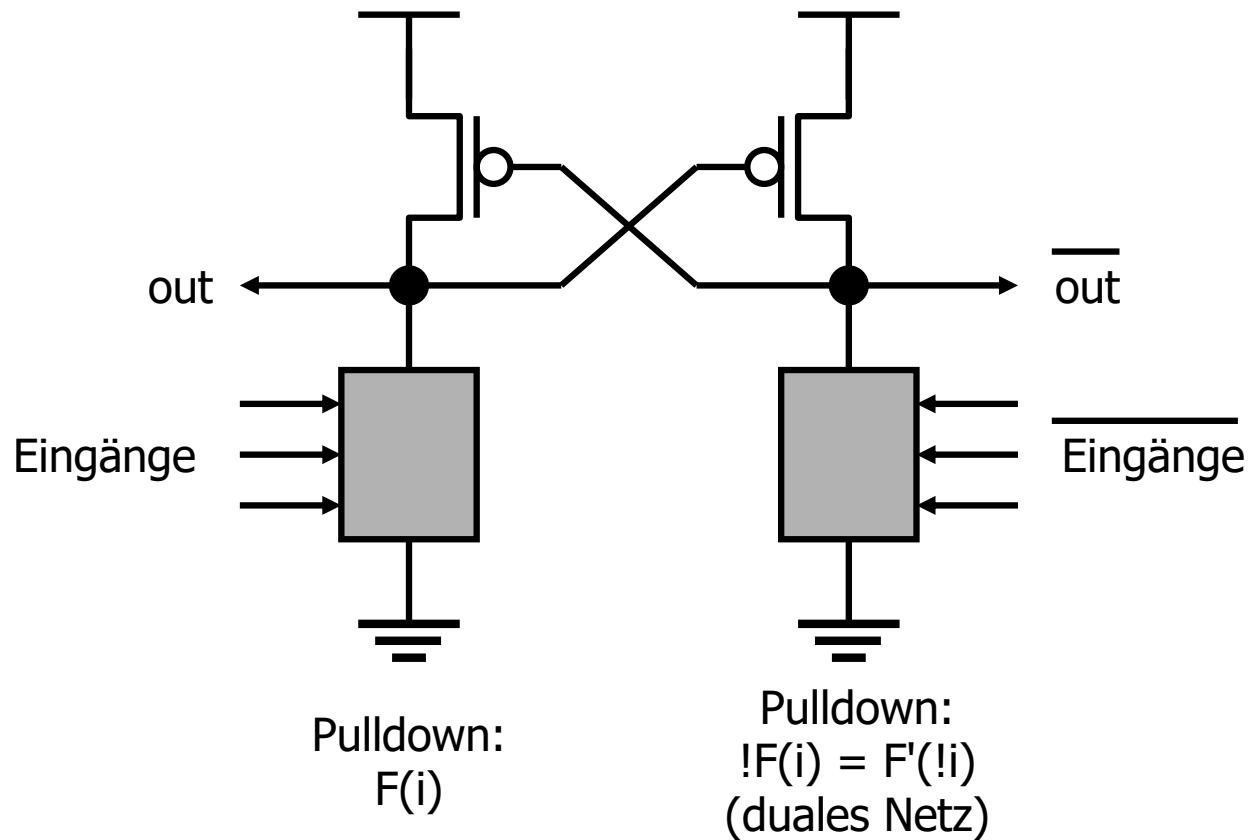
- Auch Funktionen mit Zustandsspeicherung sind sehr kompakt
- Anstiegs- und Abfallzeit sind gleich.
- Man kann mit kleinerem Signalhub arbeiten

Nachteile:

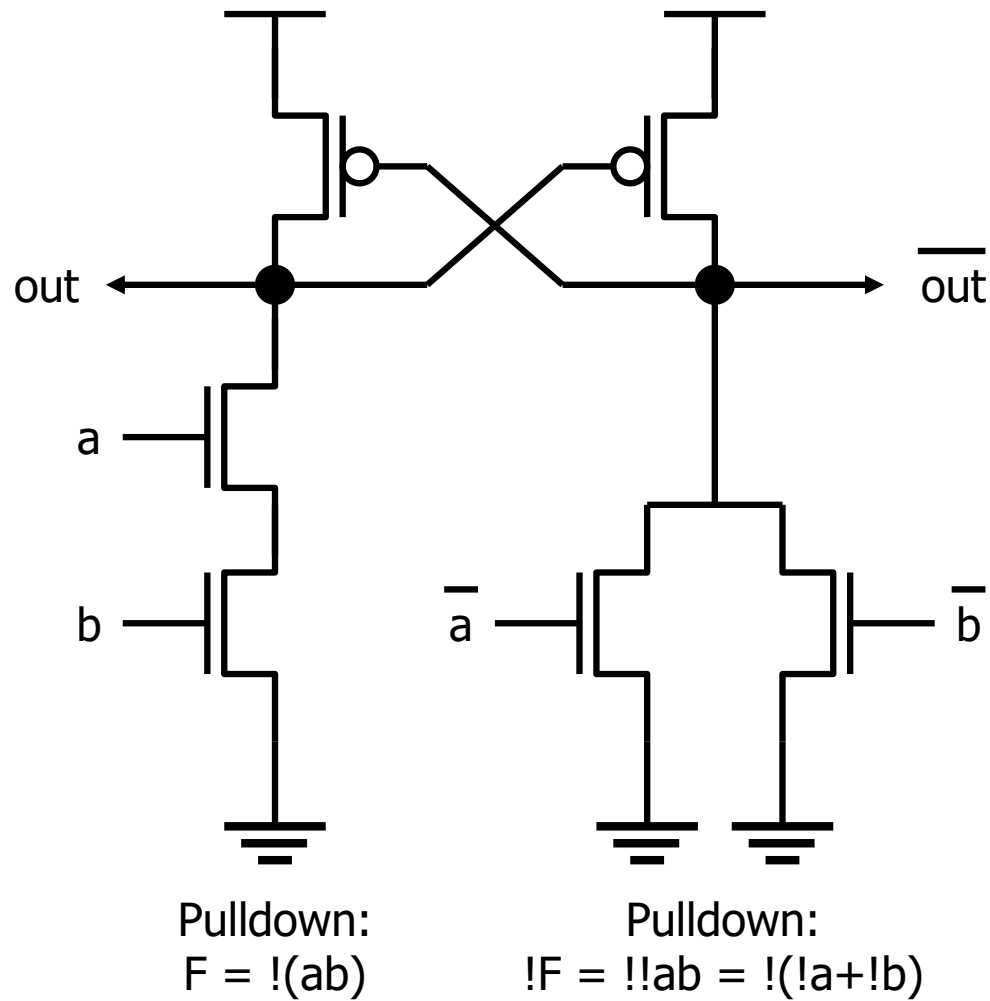
- Es müssen doppelt so viele Signale verlegt werden
- Die Leistungsaufnahme bei 'full swing' Signalen wird verdoppelt
- Es werden mindestens $2N+2$ Transistoren für ein Gatter mit einem FanIn von N benötigt

DCVS Logik

- DCVSL = differential (dual) cascode voltage switch logic
- Ein 'Latch' aus zwei PMOS Transistoren wird von zwei komplementären **NMOS** – pulldown Netzen umgeschaltet
- Durch positives Feedback ist das Umschalten sehr schnell, auch wenn die Eingangssignale langsam sind

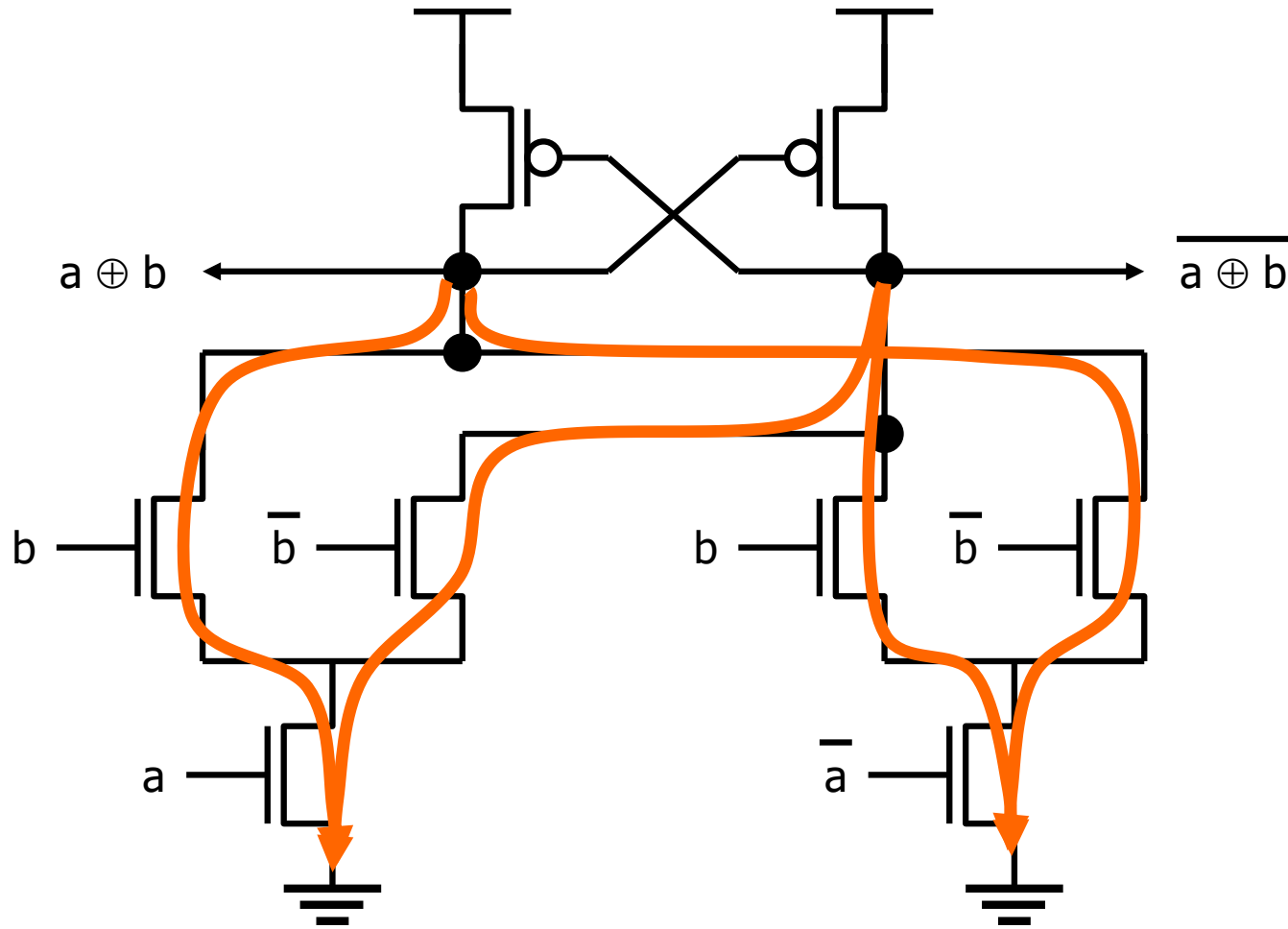


DCVS NAND



- Dieses Gatter kann für jede Funktion von zwei Variablen verwendet werden, in der genau **einer** der vier möglichen Ausgangszustände ausgewählt wird.

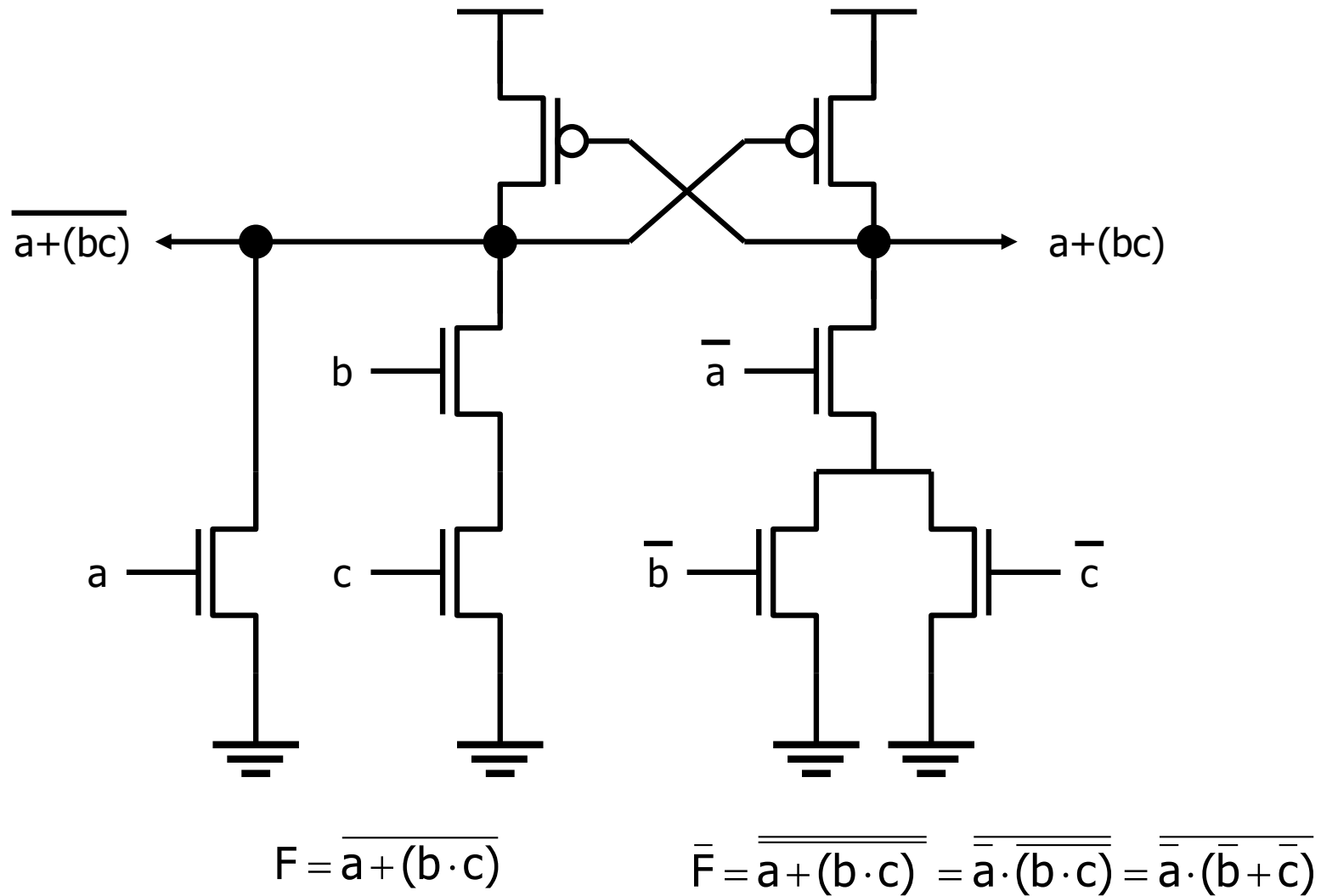
DCVS XOR



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

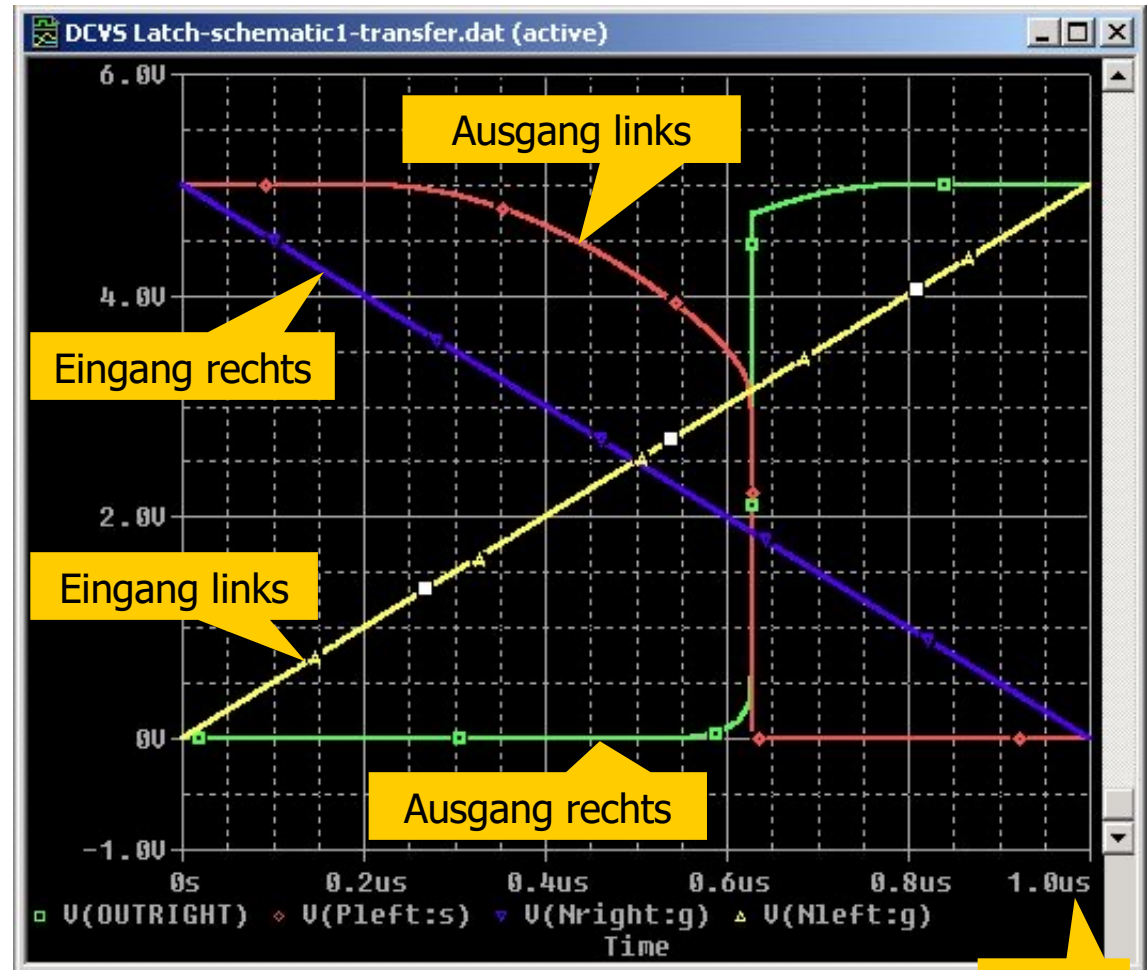
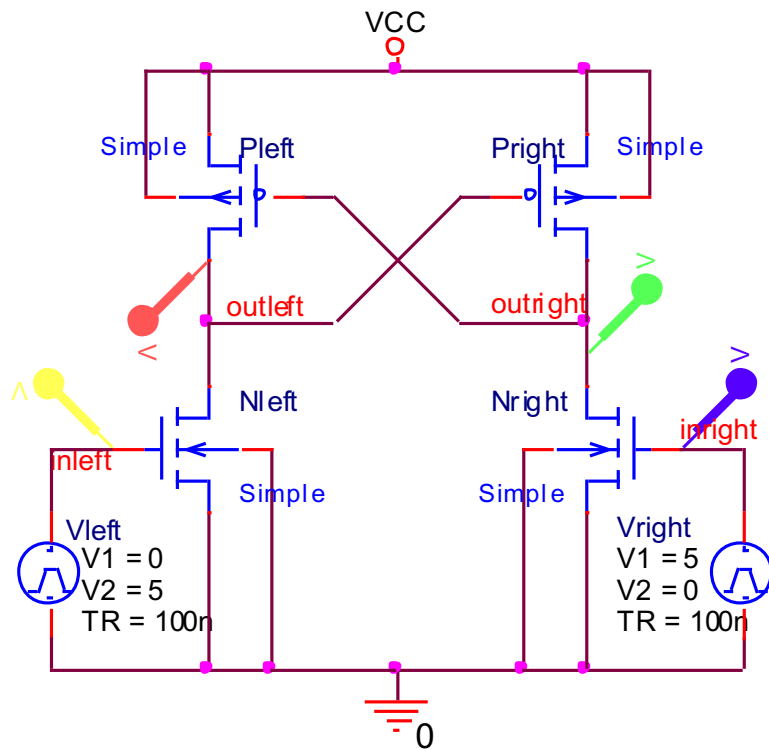
- Dieses Gatter kann für jede Funktion von zwei Variablen verwendet werden, in der genau **zwei** der vier möglichen Ausgangszustände ausgewählt werden.

DCVS Beispiel mit 3 Eingängen



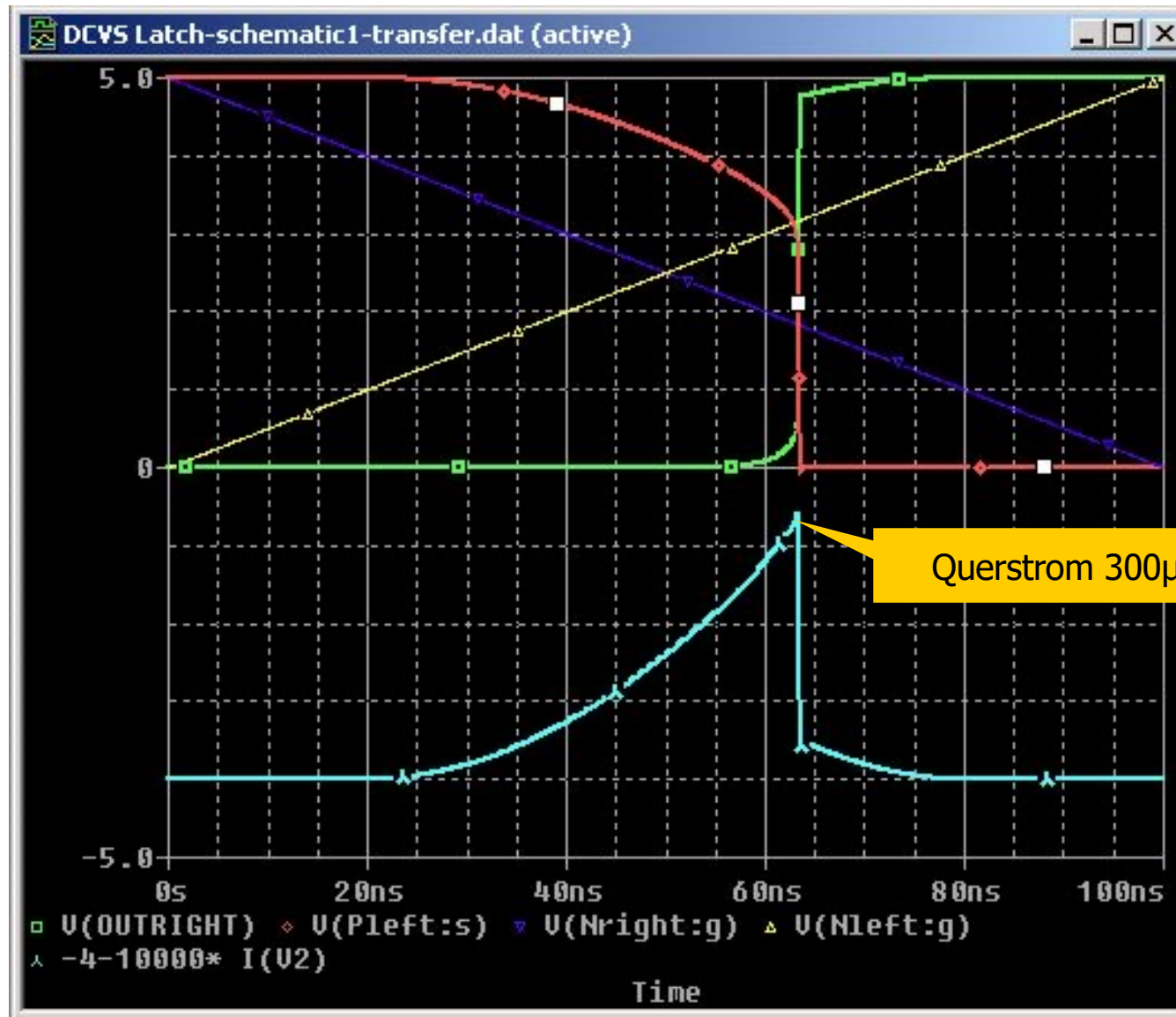
DCVS Inverter

- Simulation mit PSPICE mit sehr langsamen Eingangssignalen (Anstiegs/Abfallzeit = $1\mu\text{s}$)
- Hier $(W/L)_P = (W/L)_N$



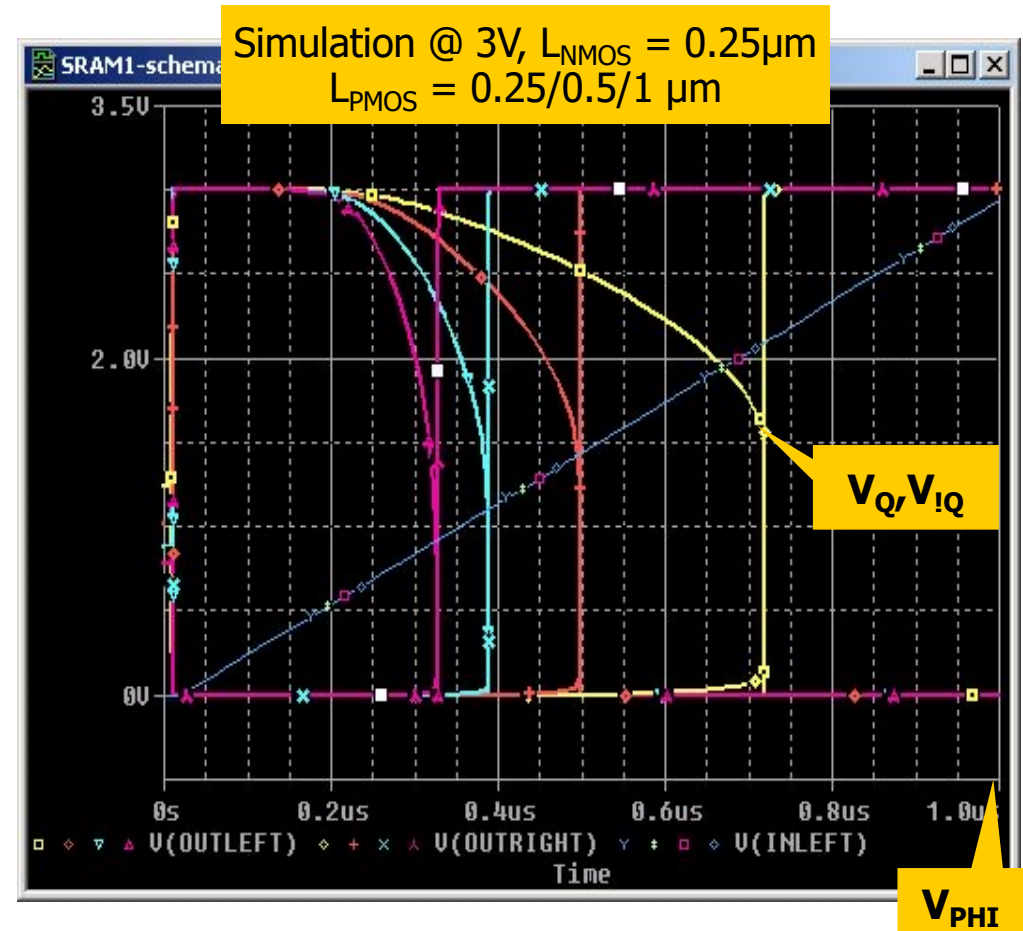
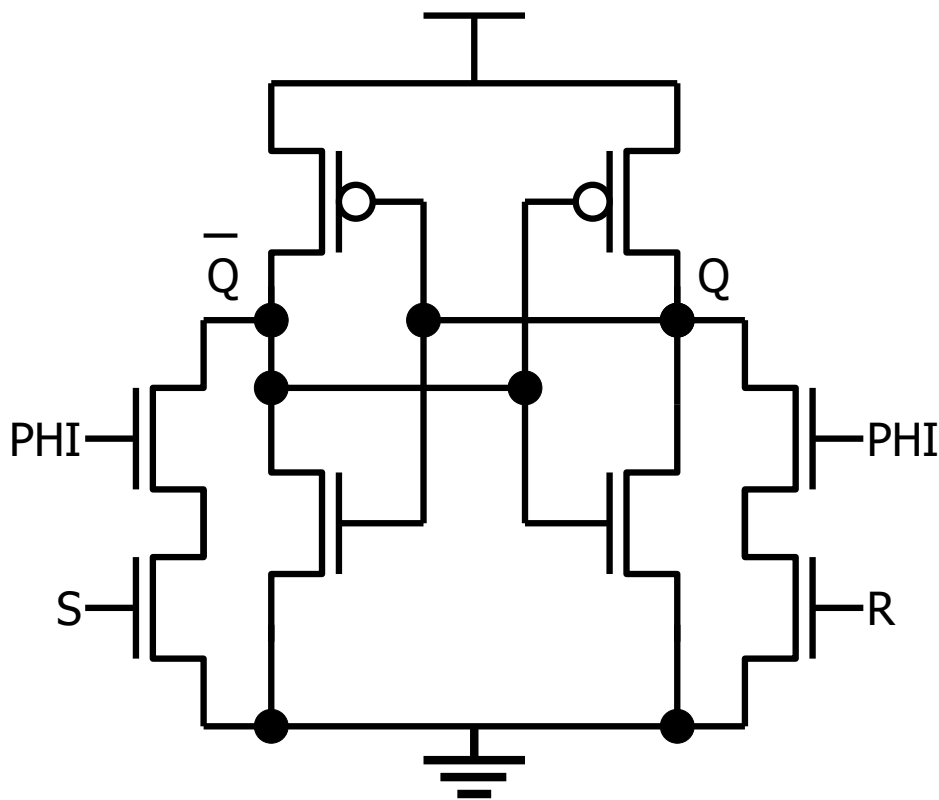
- Ausgang ist sehr schnell durch das positive Feedback

Detail: Querstrom beim Umschalten



Ähnlich JK-FF: Getaktetes SR – FlipFlop

- Sehr einfache Schaltung möglich:
- Kreuzgekoppelte Inverter, die umgeworfen werden (ähnlich statische Speicherzelle)
- Achtung: 'ratioed' Design: PMOS Transistoren dürfen nicht zu stark sein, sonst 'flippert' die Zelle nicht



DCVS Logik

Vorteile:

- Statische Logik
- kein DC-Stromverbrauch
- Kleine Eingangskapazitäten
- Schnelles Umschalten durch positive Rückkopplung
- Volle CMOS Levels
- Weniger Gatter benötigt durch Verfügbarkeit der inversen Signale und durch komplexe Logikfunktionen

Nachteile:

- Mehr Transistoren $2N+2$ vs. $2N$ (CMOS)
- 'Ratioed Logik': PMOS darf nicht zu groß sein
- Hoher Querstrom beim Umschalten. NMOS-Netz treibt 'gegen' den PMOS bis er umschaltet
- Höherer Routing-Aufwand durch komplementäre Signale
- Höherer dynamischer Leistungsverbrauch
(Nicht so klar: Doppelt so viele Signale, aber kleinere Eingangskapazitäten)

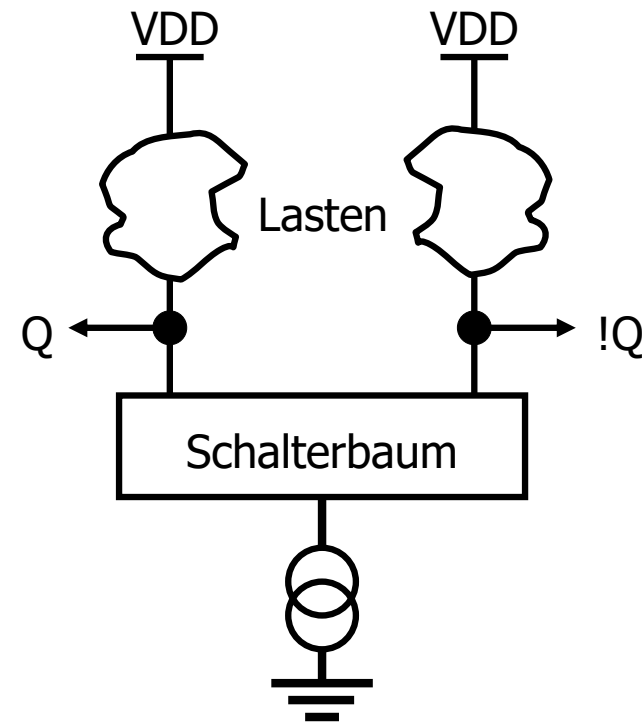
Ausblick:

- Es gibt auch differentielle CMOS Logikfamilien, die einen kleinen Signalhub haben
- Viele Vorteile, aber schwierigeres Design
- Sehr ähnlich wie ECL, aber mit MOS Transistoren

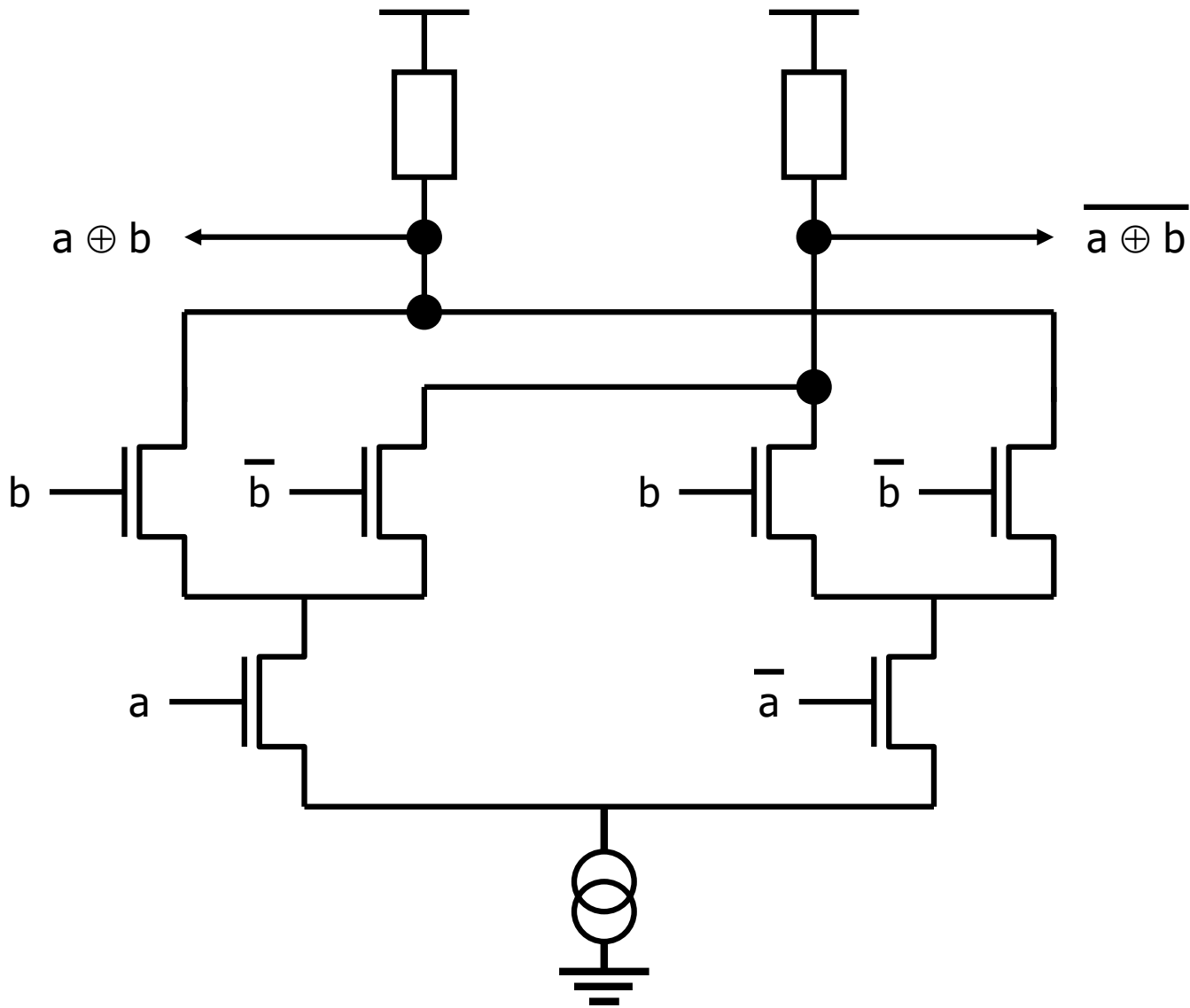
CURRENT MODE LOGIC (CML)

CML

- CML = 'Current Mode Logik'
- Gleiches Konzept wie Differential ECL aber ohne Emitter-Folger
- Der Strom wird im Schalterbaum nach links oder rechts gelenkt
- Durch das Fehlen des Buffers können große Lastkapazitäten schlecht getrieben werden.
- Problem ist der Last-'Widerstand':
 - Ausgangslevels müssen zum Eingang passen
 - Levels sollten nicht stark vom Strom abhängen
 - Es gibt verschiedene Lösungen...
- Schalterbaum ist wie bei DCVS
- Umschaltspannungen sind (in starker Inversion) höher als bei ECL, in schwacher Inversion vergleichbar.
- Einer der Vorteile:
Konstantstromoperation - keine Spikes auf Versorgung

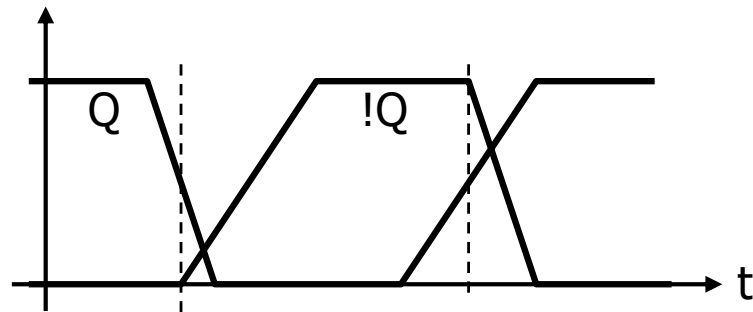
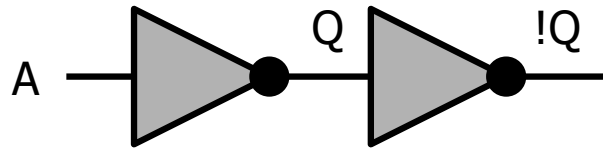


CML XOR

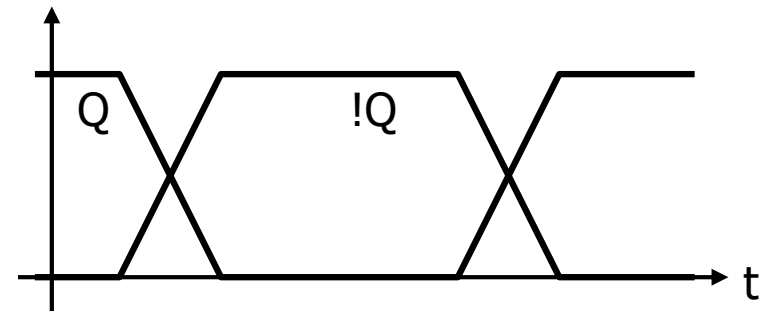
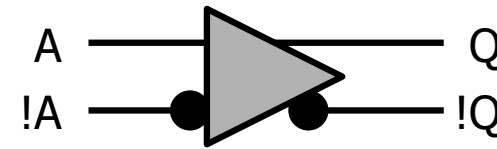


Vorteil 'echt' differentieller Logik

- Wegen der Symmetrie sind die Verzögerungszeiten für beide 'Flanken' gleich
- Die Signale sind immer echt komplementär

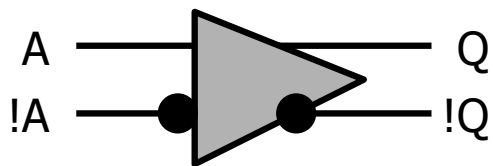
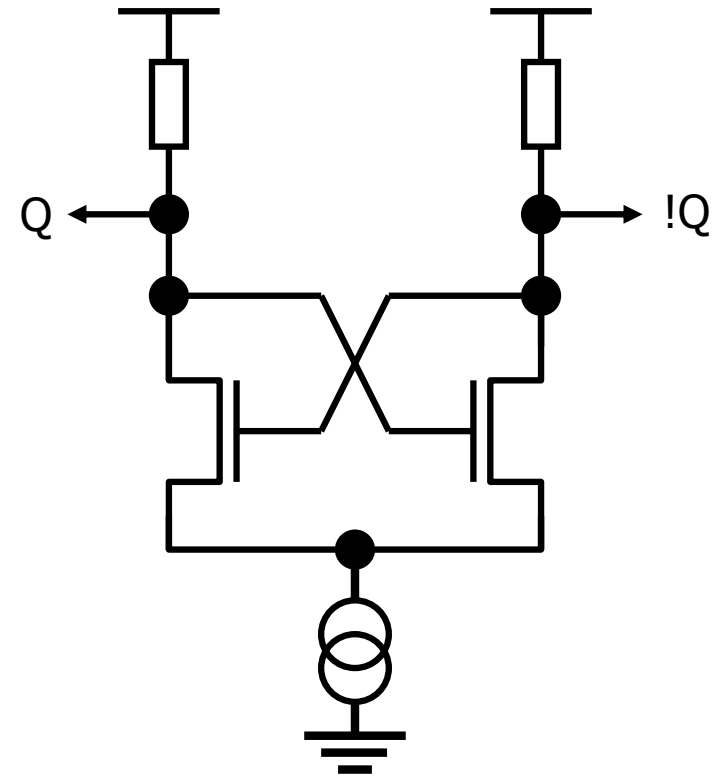
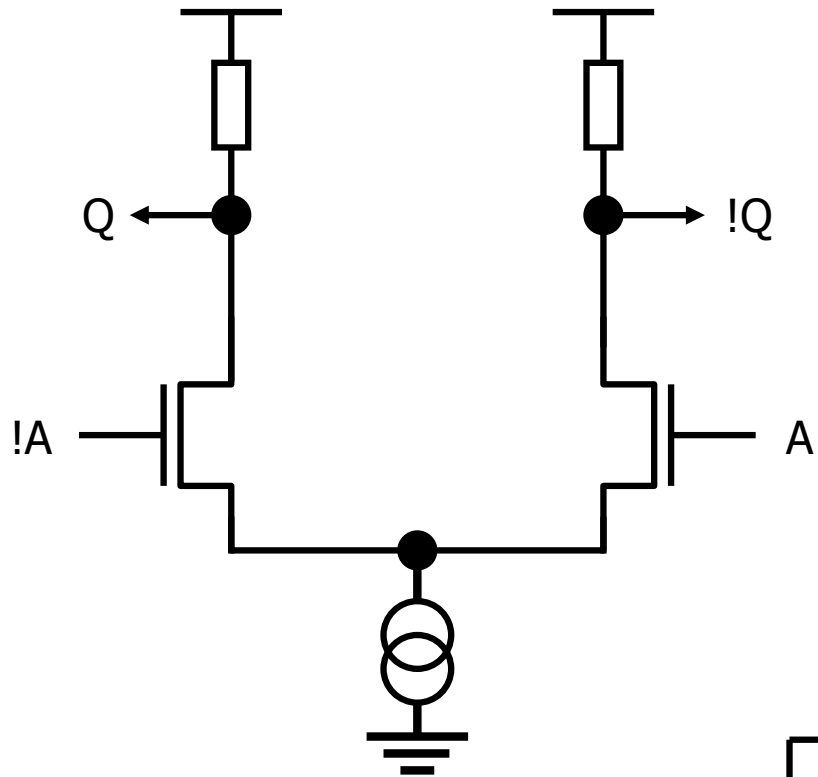


Single-ended

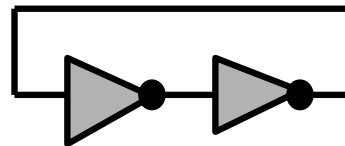


Differentiell

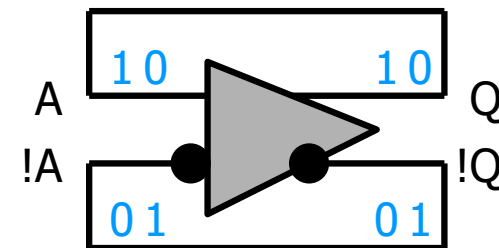
CML LATCH



Inverter / Buffer

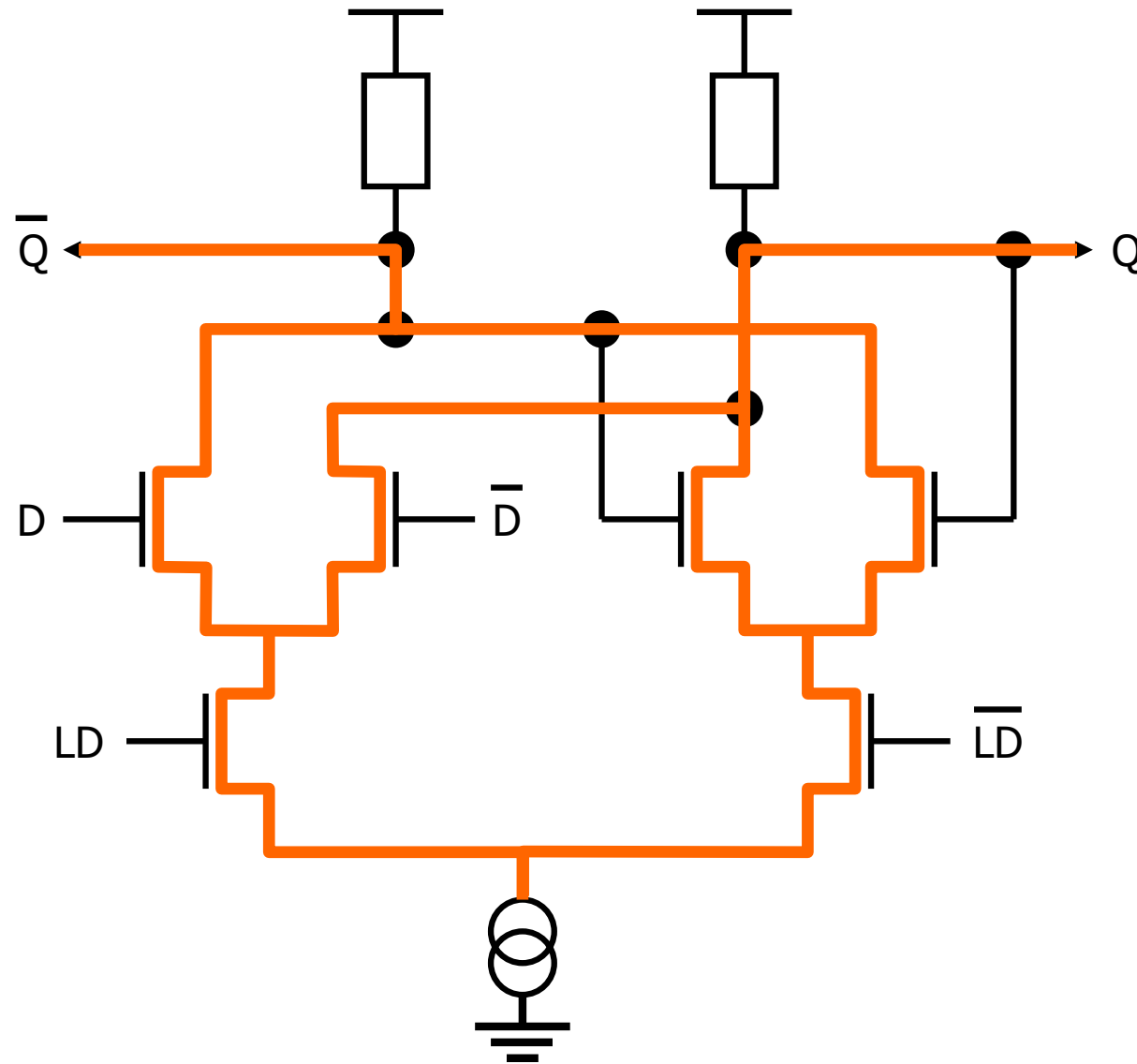


Rückkopplung
(wie bei CMOS)



Latch (Speicherzelle)

D – Latch in Differenzieller Logik



LD = 0

Exercise

- Design a CML inverter
- Chose an output voltage swing ΔV
- Chose the resistor and current to get that swing
- Can the output properly drive another CML inverter ?
 - Is the swing large enough to switch the next stage?
 - Are the high and low levels 'suited'?

- Make a ring oscillator with, e.g., 5 stages. How fast is it?
- Compare to CMOS !

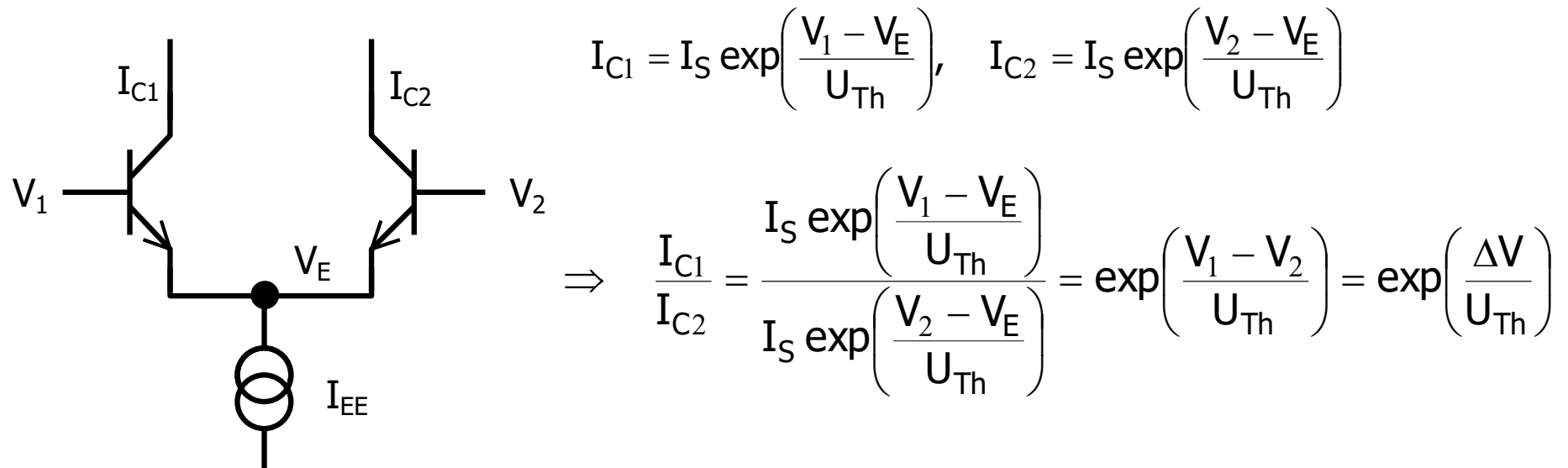
- Vary the bias current and see how the speed changes. Explain.

- Can you find a solution to limit the swing for larger currents ?

ECL

Zentrales Element von ECL: das differentielle Paar

- Der Strom im bipolaren Transistor ist $I_C \sim I_S \exp(U_{BE}/U_{Th})$.
Dabei ist der Sättigungsstrom I_S eine Bauteilkonstante, und $U_{Th} = kT/q \sim 26\text{mV}@300\text{K}$ die Temperaturspannung.



$$\frac{I_{EE}}{I_{C1}} = \frac{I_{C1} + I_{C2}}{I_{C1}} = 1 + \frac{I_{C2}}{I_{C1}} = 1 + e^{-x} \quad \text{mit} \quad x = \frac{\Delta V}{U_{Th}}$$

$$\frac{I_{C1}}{I_{EE}} = \frac{e^x}{1 + e^x} = \alpha \quad \text{mit} \quad x = \frac{\Delta U}{U_{Th}}$$

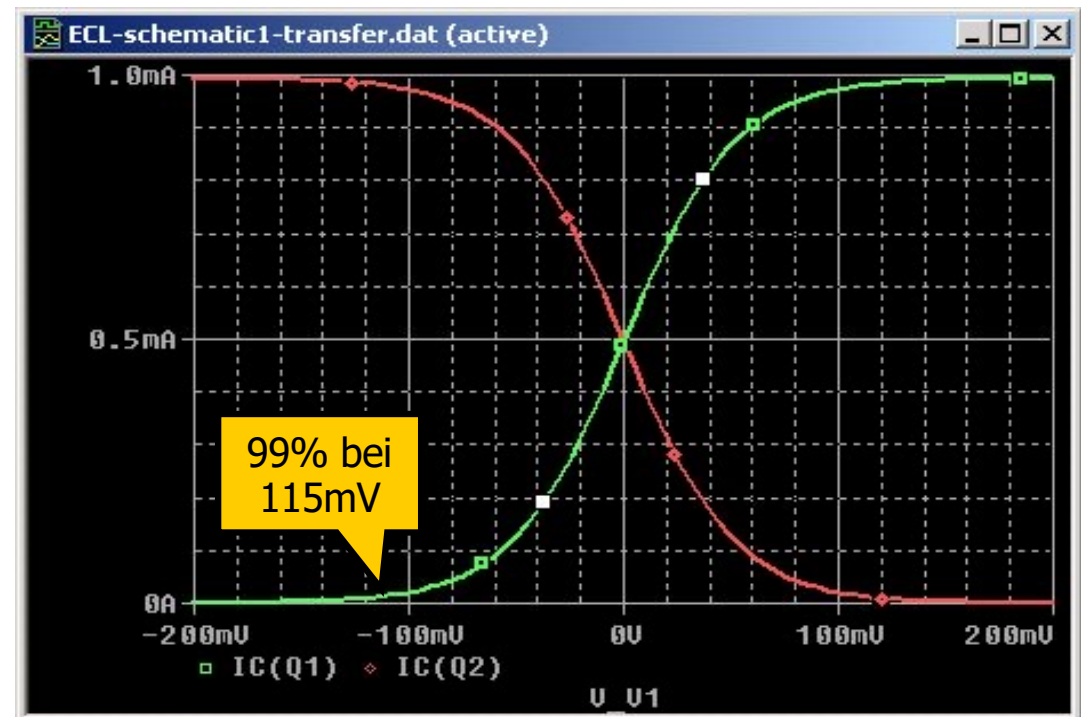
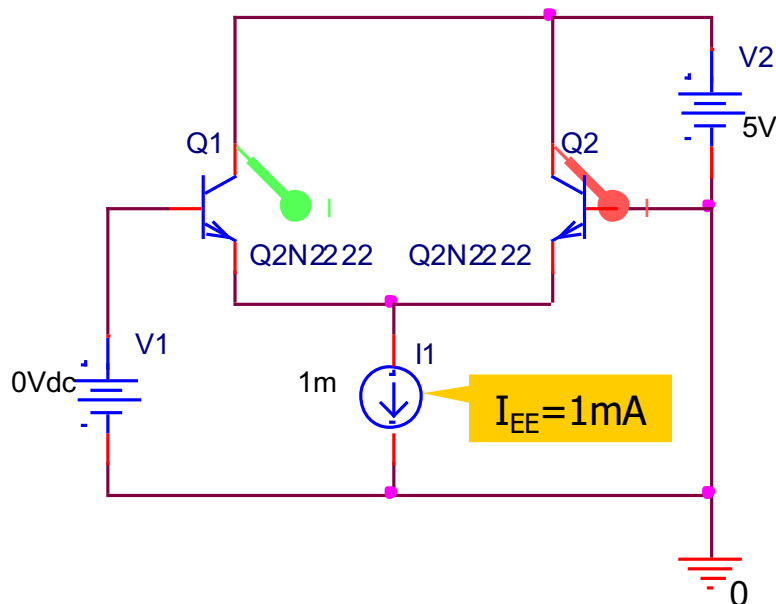
Zentrales Element von ECL: das differentielle Paar

- Strom im bipolaren Transistor $I_C \sim I_S \exp(U_{BE}/U_{TH})$. Rechnung s. Tafel.

Ergebnis:
$$\frac{I_{C1}}{I_{EE}} = \frac{e^x}{1 + e^x} = \alpha \quad \text{mit} \quad x = \frac{\Delta U}{U_{Th}} \Rightarrow V_{IL,IH} = V_{ref} \pm U_{Th} \cdot \ln \frac{\alpha}{1 - \alpha}$$

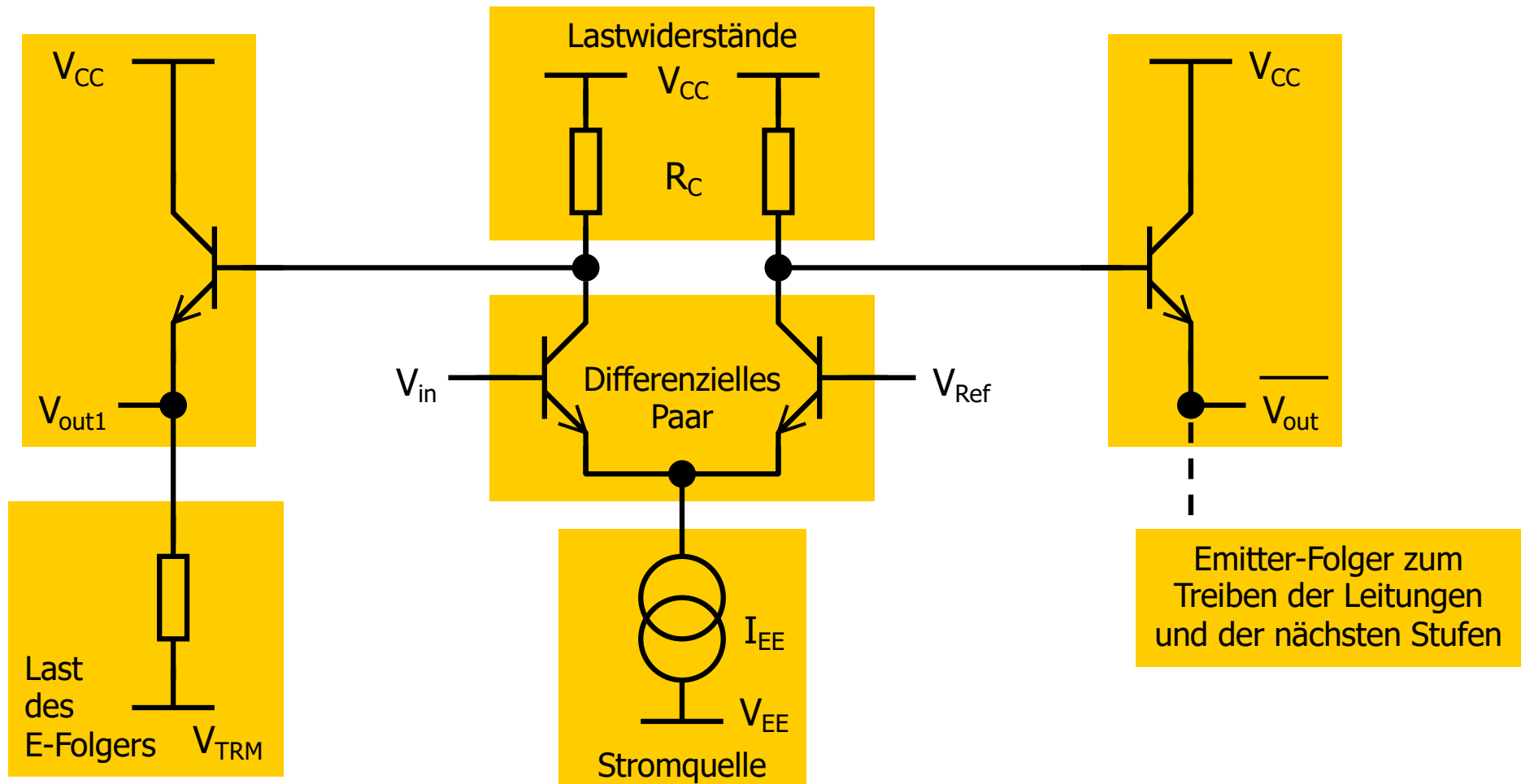
- Mit $\alpha = 0.01$ ergibt das nur 115 mV (unabhängig von I_{EE} !)
- Kleiner Signalhub & die bipolaren Transistoren kommen nie in Sättigung \Rightarrow schnell.

- SPICE Simulation:

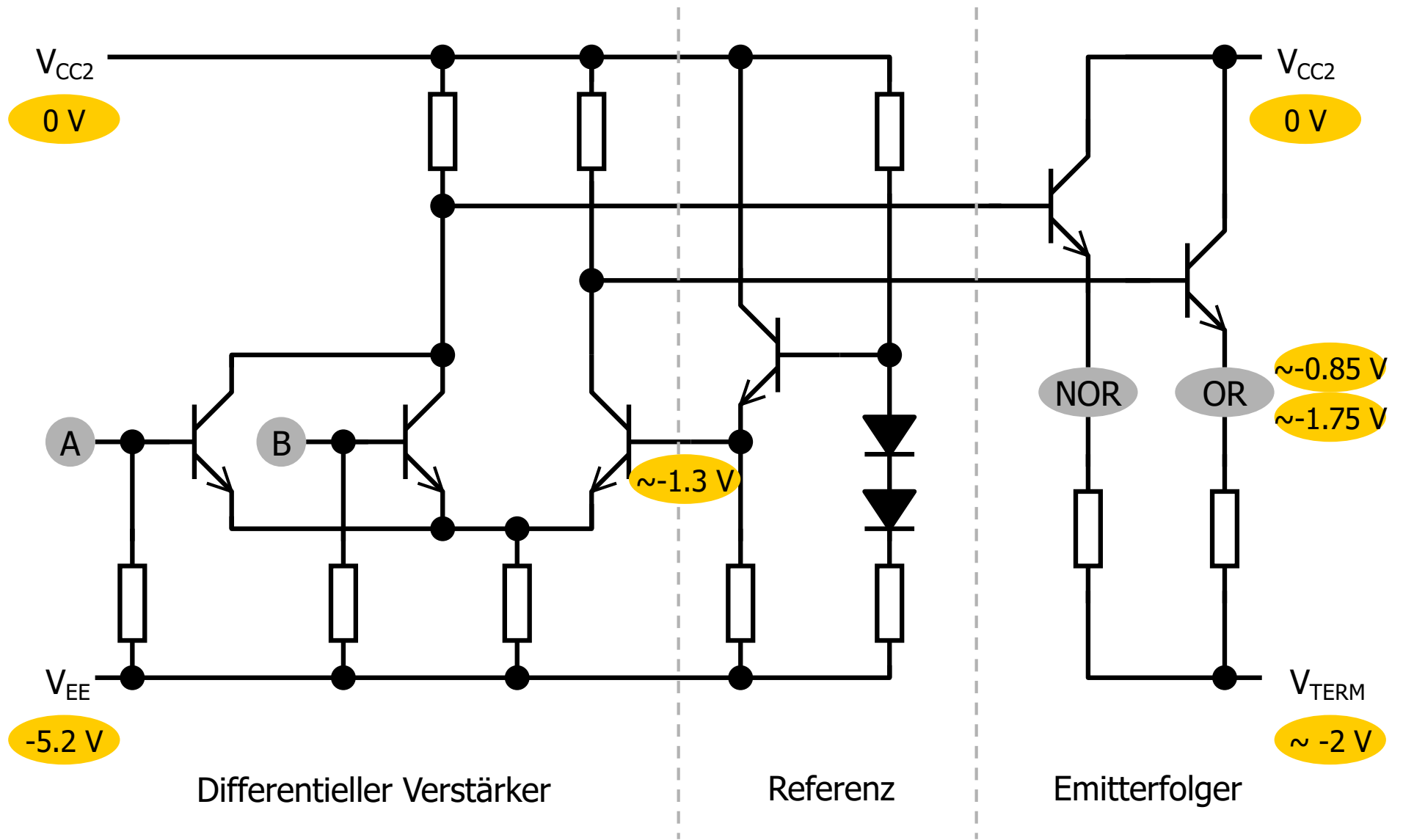


ECL

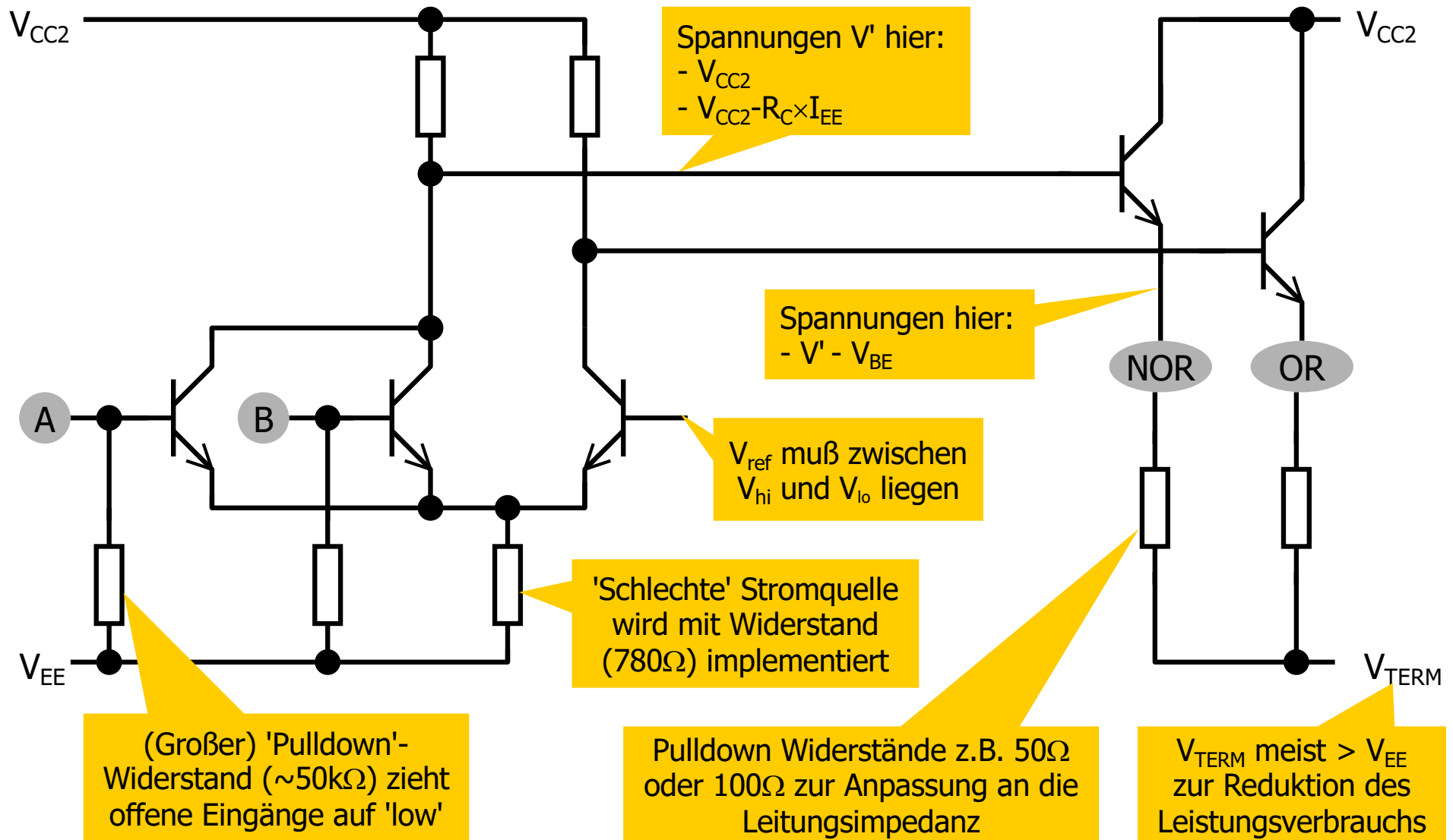
- ECL = 'Emitter Coupled Logik' - sehr schnelle Logik mit **bipolaren** Transistoren
- Ein differentielles Paar lenkt I_{EE} in einen der beiden Lastwiderstände.
- Meist sind Ausgang und Komplement verfügbar.
- Signalhub: $\Delta U = I_{EE} \times R_C \ll V_{CC}$: '**low swing**' (bei ECL: ca. 800mV)



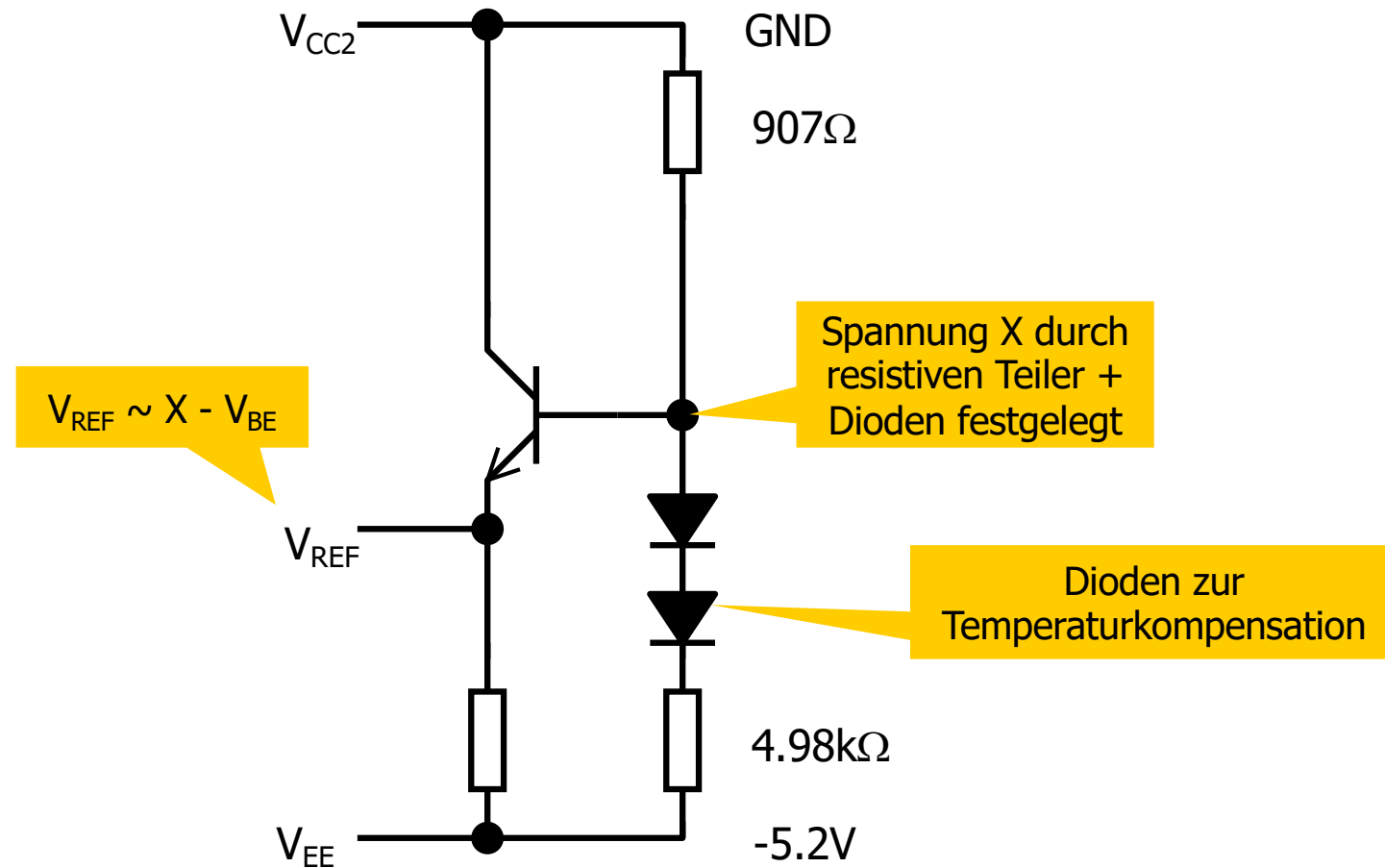
OR/NOR Gate mit mehr Details



ECL: Differenzverstärker und Emitterfolger

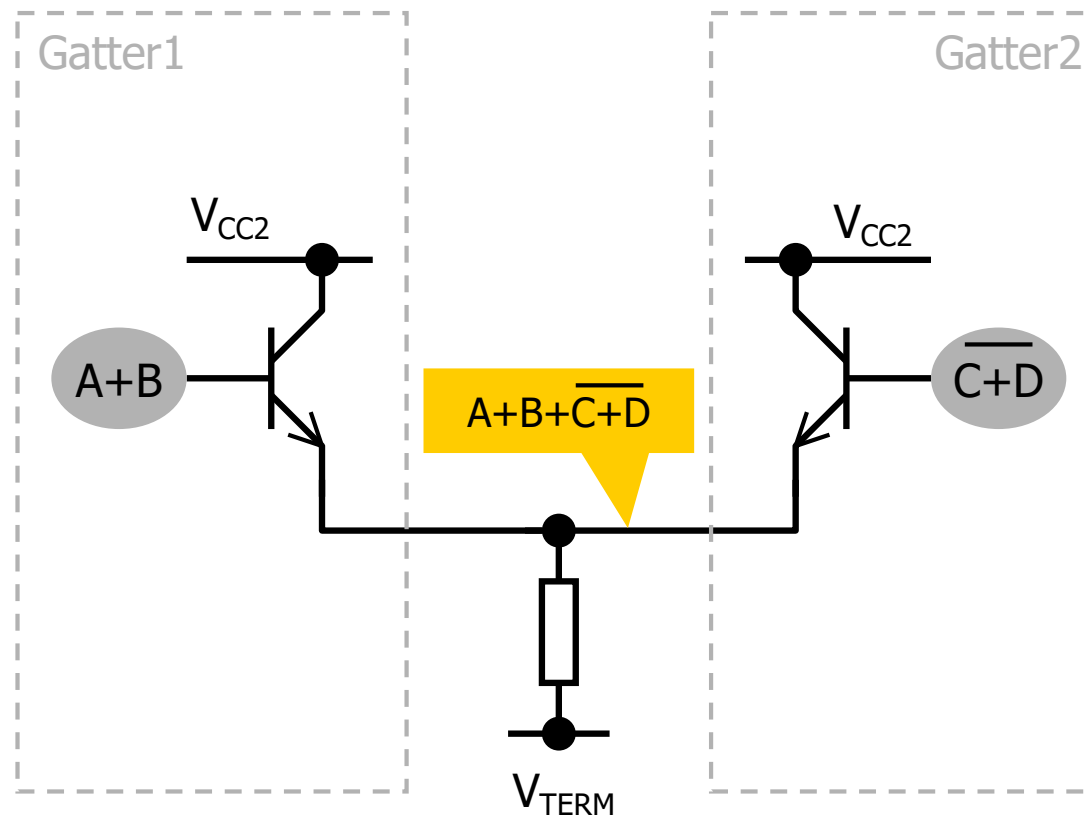


Erzeugung der Referenzspannung



Wired-OR in ECL

- Mehrere Ausgänge können direkt verbunden werden. Sie bekommen nur einen Pulldown-Widerstand.
- Diese 'Wired-OR' kostet keinerlei Ressourcen und spart Leistung



ECL Gatter 1967

- Bipolare Transistoren (und damit ECL) gab es lange vor CMOS !

