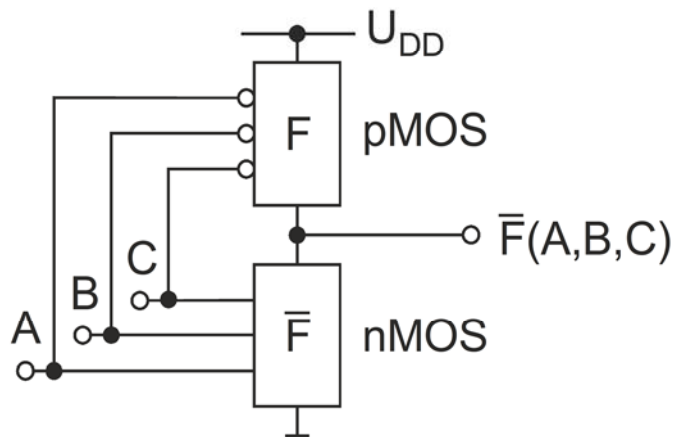


4. Statische Logik-Schaltungen mit CMOS

4.1. Grundlagen

Design

1. Logikfunktion
2. dc-Verhalten
3. Transienten



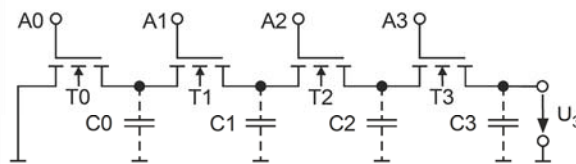
Logische Funktionen in CMOS Technologie lassen sich vereinfacht durch einen nMOS- und einen pMOS-Block darstellen.

Die Art der Verschaltung der Transistoren (Reihenschaltung – Parallelschaltung) ergibt dann die gewünschte logische Funktion.

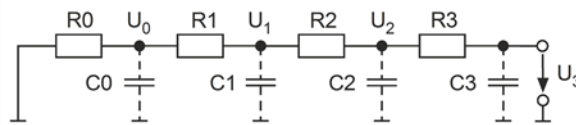
Als nächstes wollen wir uns die "kritischen" Pfade näher betrachten.

4.2. MOSFET-Reihenschaltungen

Entladung über n-MOSFET-Ketten



(a) MOSFET-Kette



(b) RC-Modell

$t = 0 \Rightarrow U_3 = U_{DD} \rightarrow \text{Entladung}$

$\Rightarrow A_0 = 1 = A_1 = A_2 = A_3 = \text{high!}$

$\Rightarrow \text{Über Drain-Source-Widerstand:}$

$$R_{DS,n} = \frac{1}{S_n} = \frac{1}{\beta_n \cdot (U_{DD} - U_m)}$$

Drift-Gleichungen:

$$-C_3 \cdot \frac{dU_3}{dt} = \frac{U_3 - U_2}{R_3} \rightarrow U_3(t) \approx U_{DD} \cdot e^{-t/\tau}$$

$$-C_2 \cdot \frac{dU_2}{dt} = \frac{U_2 - U_1}{R_2} - \frac{U_3 - U_2}{R_3} \quad \text{usw.}$$

$$t_D \cong \sum_{i=0}^{N-1} \tau_i \quad ; \quad \tau_i = \left(\sum_{j=0}^i R_j \right) \cdot C_i$$

$$\rightarrow \tau_0 = R_0 \cdot C_0 \quad R \sim \left(\frac{w}{L} \right)^{-1}$$

$$\tau_1 = (R_0 + R_1) \cdot C_1 \quad C \sim \left(\frac{w}{L} \right)$$

usw.

Da wir ja in CMOS-Schaltungen immer Kapazitäten umladen müssen sind die Reihenschaltungen der Transistoren die entscheidenden Elemente.

Zunächst wird die Reihenschaltung von nMOS-Transistoren analysiert. Dazu kann eine Schaltung nach obigem Bild skizziert werden.

Über die nMOS-Ketten wird die Ausgangskapazität des Gatters entladen. Ausgehend von einem High-Pegel am Ausgang (U_3) wird durch das Einschalten aller Transistoren (T0 ... T3) die in C_3 gespeicherte Ladung über die Drain-Source Widerstände der 4 Transistoren nach Masse abgeleitet. In den intrinsischen Kapazitäten der Transistoren ist ebenfalls Ladung gespeichert, was durch $C_0 \dots C_2$ angedeutet ist.

Der Drain-Source Widerstand der Transistoren ist wie angegeben definiert.

Damit lässt sich ein RC-Ersatzschaltbild erstellen und die Entladung des Ausgangs vereinfacht formelmäßig darstellen.

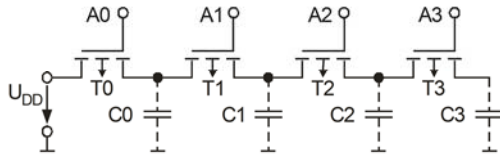
Wir können die Verzögerungszeit τ_D als Summe der einzelnen Entladezeitkonstanten τ_i definieren.

Die Zeitkonstanten τ_i können nach den angegebenen Formeln bestimmt werden.

Anmerkung: Nur C_3 ist auf U_{DD} aufgeladen. Die Kapazitäten C_0 , C_1 und C_2 können nur auf max. $U_{DD} - n \cdot U_{th}$ aufgeladen werden. Substrateffekte werden vernachlässigt.

4.2. MOSFET-Reihenschaltungen

Ladung über p-MOSFET-Ketten



(a) pMOS-Kette

$$A_0 = A_1 = A_2 = A_3 = 0 \Rightarrow \text{leitende pMOS!}$$

$$R_p = \frac{1}{\beta_p \cdot (U_{DD} - |U_{tp}|)}$$

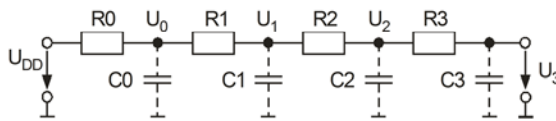
$$U_3(t) \cong U_{DD} \cdot e^{-t/t_D}$$

$$t_D \approx \sum_{i=0}^{N-1} \left(\sum_{j=0}^i R_j \right) \cdot C_i$$

$\Rightarrow$ Analog zu nMOS-Ketten!

$$\text{für } \left(\frac{w}{L} \right)_n = \left(\frac{w}{L} \right)_p \text{ folgt } R_p > R_n$$

$\rightarrow$ lange pMOS-Ketten sind ungünstig!



(b) RC-Modell

Für das Aufladen der Ausgangskapazität über pMOS-Ketten gelten die gleichen Ansätze, die wir für die nMOS-Ketten angenommen haben.

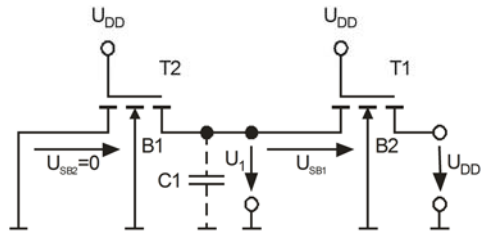
Wir erhalten damit formelmäßig auch die gleichen Ergebnisse.

Die Verzögerungszeit τ_D ist nicht in beiden Fällen gleich groß, auch wenn $\beta_n = \beta_p$ ist, da die intrinsischen Transistorkapazitäten aufgrund der in diesem Fall 3-fach größeren Fläche der pMOS Transistoren größer sind, als die der nMOS Transistoren.

Wird aber auch noch auf Chipfläche optimiert, d.h. die Weite der PMOS-Tr. ist weniger groß, verschlechtert sich der Aufladevorgang.

4.2. MOSFET-Reihenschaltungen

Substrat-Effekte



für pMOS

$$R_p = \frac{1}{\beta_p \cdot (U_{DD} - |U_{tp}|)}$$

mit Erhöhung von U_{SB}

Body-bias-Fehler

$$\gamma_n = \frac{1}{C_{ox}} \cdot \sqrt{2 \cdot e \cdot \epsilon_{Si} \cdot N_a}$$

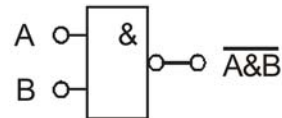
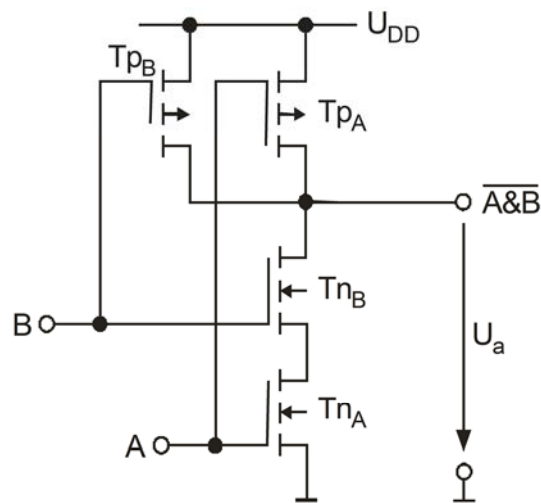
$$U_{SB1} = U_1 \rightarrow \text{Erhöhung } U_m$$

$$U_{m1} = U_{m0} + \gamma_n \cdot \left(\sqrt{2 \cdot |\phi_{fp}| + U_1} - \sqrt{2 \cdot |\phi_F|} \right)$$

$$\rightarrow \text{Erhöhung } R_{DS} \quad R_{DS1} = R_{n1} = \frac{1}{\beta_n \cdot (U_{DD} - U_{m1})} \quad U_{m1} > U_{m0}$$

Ziehen wir nun noch die Substrateffekte hinzu, die sich ergeben, wenn die Transistorketten in einer gemeinsamen Wanne integriert sind, müssen wir die Drain-Source Widerstände der Ketten entsprechend des gezeigten Beispiels korrigieren.

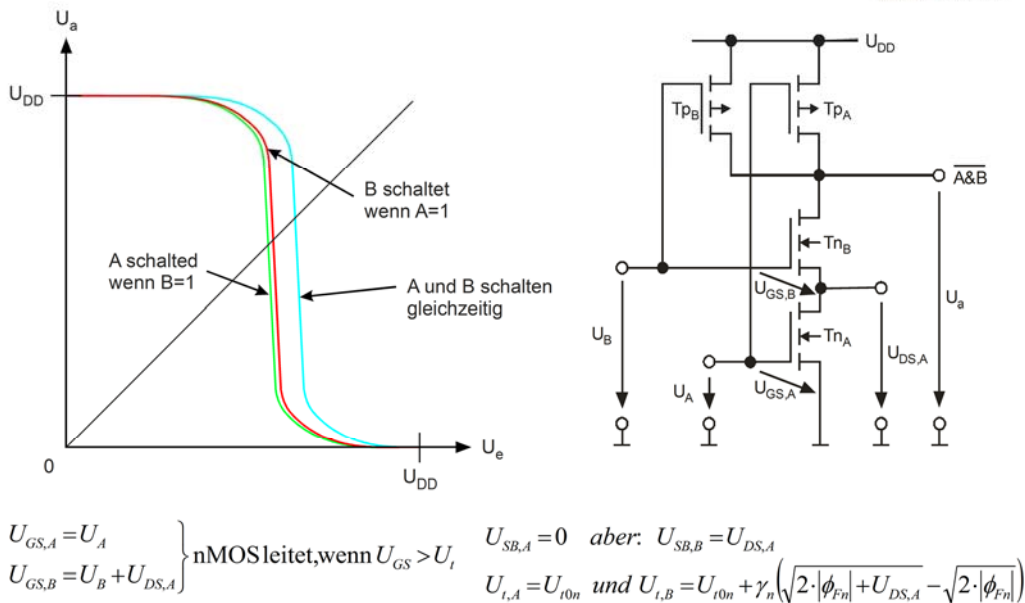
4.3. NAND-Gatter



A	B	$\overline{A \& B}$
0	0	1
1	0	1
0	1	1
1	1	0

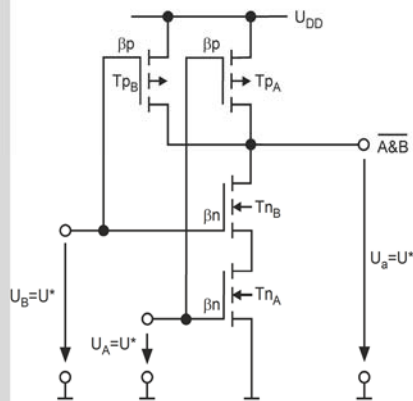
Nach den Betrachtungen der Umladezeitkonstanten wollen wir uns jetzt mit den statischen und dynamischen Eigenschaften einfacher logischer Gatter befassen. Ein NAND-Gatter mit zwei Eingängen besteht aus 2 in Reihe geschalteter nMOS- und 2 parallel geschalteter pMOS-Transistoren. Schaltung, logisches Symbol und Wahrheitstabelle.

4.3. NAND-Gatter: Statischer Betrieb



Im statischen Betrieb betrachten wir wieder die Übertragungsfunktion. Dabei stelle wir fest, dass im Gegensatz zum Inverter nun 3 unterschiedliche Übertragungskennlinien vorhanden sind. Die unterschiedlichen Schaltpunkte ergeben sich aus den 3 Varianten, welches Eingangssignal das Umschalten des Ausgangs von H -> L bewirkt. Aus den Maschengleichungen für die Eingangsspannungen an A bzw. B können die Spannungen bestimmt werden, bei denen die Transistoren A bzw. B beginnen zu leiten. Liegt ein gemeinsames Signal an A und B an, haben wir einen Inverter. Aus dem Inverterverhältnis β kann der zugehörige Schaltpunkt berechnet werden.

4.3. NAND-Gatter: Statischer Betrieb



$$U_{in} = U^* = U_a$$

$$U_A = U_B = U^* = U_{BA} \rightarrow B \text{ gesättigt}, A \text{ linear}$$

$$1) I_D = \frac{\beta_n'}{2} \cdot (U^* - U_{t0n} - U_{DS,A})^2$$

$$= \frac{\beta_n'}{2} \cdot [2 \cdot (U^* - U_{t0n}) \cdot U_{DS,A} - U_{DS,A}^2]$$

somit

$$U^* = U_{in} + 2 \cdot \sqrt{\frac{I_D}{\beta_n}}$$

2) Beide pMos leiten (Sättigung)

$$I_D = 2 \cdot \left(\frac{\beta_p'}{2} \right) \cdot (U_{DD} - |U_{t0p}|)^2$$

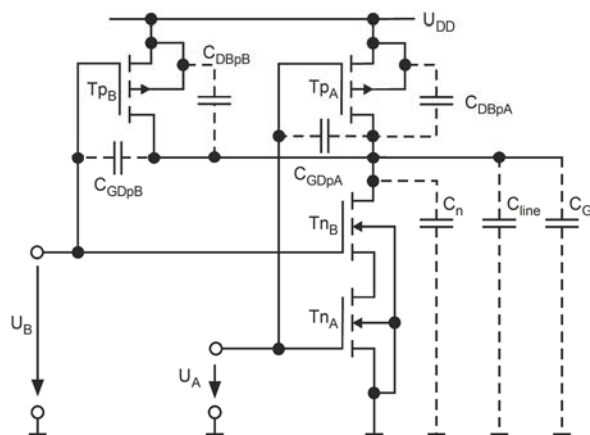
$$\text{linear: } I_D = \frac{\beta}{2} \cdot [2 \cdot (U_{GS} - U_t) \cdot U_{DS} - U_{DS}^2]$$

$$\text{Sättigung: } I_D = \frac{\beta}{2} \cdot (U_{GS} - U_t)^2$$

$$U^* = \frac{U_{t0n} + 2 \cdot \sqrt{\frac{\beta_n}{\beta_p}} \cdot (U_{DD} - |U_{t0p}|)}{\left(1 + \sqrt{\frac{\beta_n}{\beta_p}} \right)} \quad \text{wie Inverter aber Faktor 2!}$$

Die Berechnungen für diesen Schaltzustand sind hier ausgeführt.

4.3. NAND-Gatter: Schaltverhalten



$$C_a = (C_{GDpA} + C_{GDpB}) + K(\Delta U) \cdot [C_{DBpA} + C_{DBpB}] + \sum C_n + C_{line} + C_G$$

$$\text{mit } K(\Delta U) = \frac{2 \cdot U_{diff}}{U_{OH}} \cdot \left[\left(1 + \frac{U_{DD}}{U_{diff}} \right)^{\frac{1}{2}} - 1 \right]$$

Als nächstes wollen wir das Schaltverhalten eines NAND-Gatters untersuchen. Dazu müssen zuerst alle Kapazitäten, die auf den Ausgang wirken, erfasst werden. Dies sind zum einen die intrinsischen Kapazitäten der Transistoren selbst und die am Ausgang der Schaltung wirksamen Lastkapazitäten der Verbindungsleitung zum nächsten Gatter und die Eingangskapazität des angesteuerten Gatters selbst. Wir erhalten damit die angegebene Formel für C_a .

4.3. NAND-Gatter: LH-Schaltzeit (Laden)

$$U_a(t=0) = 0 \quad ; \quad (A = B = 1)$$

Laden durch pMOS

$$t_{LH} = \tau_p \cdot \left[\frac{2(|U_{tp}| - U_0)}{(U_{DD} - |U_{tp}|)} + \ln \left(\frac{2 \cdot (U_{DD} - |U_{tp}|)}{U_0} - 1 \right) \right]$$

$$U_0 = 0,1 \cdot U_{DD} \quad \rightarrow 10\% U_{DD}$$

$$\tau_p = \frac{C_a}{\left(\sum_j \beta_{pj} \right) \cdot (U_{DD} - |U_{tp}|)} \quad \text{für } \left(\sum_j \beta_j \right) \text{ gilt:}$$

$$A = 0 \rightarrow \beta_{p1} \quad (\text{Eingang A bewirkt Umschalten})$$

$$B = 0 \rightarrow \beta_{p2} \quad (\text{Eingang B bewirkt Umschalten})$$

$$A = B = 0 \quad (\text{beide Eingänge bewirken Umschalten})$$

$$\left(\sum_j \beta_j \right) = (\beta_{p1} + \beta_{p2})$$

Zuerst wird der Aufladevorgang am Ausgang (t_{LH}) betrachtet.

Die Anstiegs- und Abfallzeiten sind bekanntlich durch die Ausgangspegel 10%-90% definiert.

Wenn wir die Gleichungen, die wir beim Inverter abgeleitet haben leicht modifiziert anwenden, erhalten wir die hier angeführten Ergebnisse. Wir erkennen, dass in der Formel für die Zeitkonstante τ_p im Nenner die Größe β_p durch eine Summe ersetzt wurde.

Abhängig vom für das Umschalten wirksamen Eingang setzen wir dann die entsprechenden Werte ein.

Man erkennt, dass im Falle des "Inverterbetriebs" die Zeitkonstante aufgrund der Parallelschaltung der Transistoren nur noch halb so groß ist.

4.3. NAND-Gatter: HL-Schaltzeit (Entladen)

$$U_a(t=0) = U_{DD}$$

Delay :

$$t_D \cong R_{nA} \cdot C_{nA} + (R_{nA} + R_{nB}) \cdot C_a$$

$$t_{HL} = \tau_n \cdot \left[\frac{2 \cdot (U_m - U_0)}{U_{DD} - U_m} + \ln \left(\frac{2 \cdot (U_{DD} - U_m)}{U_0} - 1 \right) \right]$$

$$\tau_n \cong \frac{C_a}{\beta_{n,eff} \cdot (U_{DD} - U_m)}$$

$$\beta_{n,eff} = \beta_n' \cdot \left[\frac{W}{(L_{nA} + L_{nB})} \right] \quad \text{wie Einzel-MOSFET mit } L_{nA} + L_{nB}$$

Beim Umschaltvorgang von H nach L am Ausgang können wir die gleichen Überlegungen anstellen.

Wir gehen wieder von den Gleichungen des Inverters aus und ersetzen wieder bei der Zeitkonstante das β_n durch $\beta_{n,eff}$.

In diesem Fall sind die beiden nMOS-Transistoren immer in Reihe geschaltet. Damit wird $\beta_{n,eff}$ durch die angeführte Gl. beschrieben.

Abhängig davon, welcher Transistor das Umschalten auslöst, muss die korrekte Schwellspannung in die Gleichung für t_{HL} eingesetzt werden.

4.3. NAND-Gatter: Design

1. Timing

β_n, β_p – Anschließend erst dc

$LH \rightarrow pMOS \left(\frac{W}{L} \right)_{p1} \text{ und } \left(\frac{W}{L} \right)_{p2}$

wenn ein MOSFET leitet :

$$\left(\frac{W}{L} \right)_p = \frac{C_a}{\beta_n \cdot \tau_p (U_{DD} - |U_{tp}|)}$$

$HL \rightarrow nMOS \rightarrow \text{identisch } \left(\frac{W}{L} \right)_n$

$$\left(\frac{W}{L} \right)_n = \frac{2 \cdot C_a}{\beta_n \cdot \tau_n (U_{DD} - U_{in})}$$

2. DC – S^* Schaltpunkt $\rightarrow \beta_n / \beta_p$ Bestimmung

Aus den bisherigen Analysen des NAND-Gatters ergeben sich nun für das Schaltungsdesign folgende Schritte:

1. Auswahl der Schaltzeiten über die Bestimmung der W/L-Verhältnisse der Transistoren

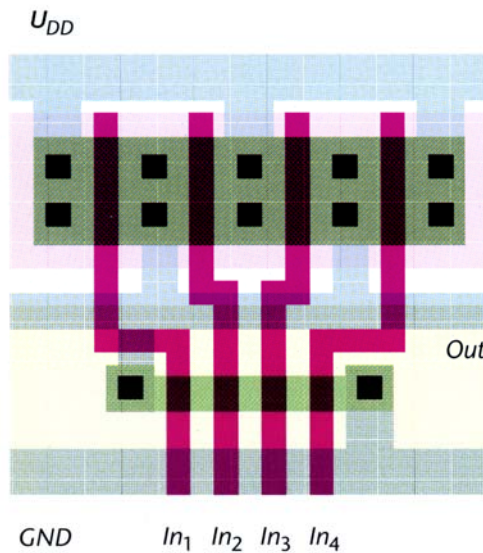
Dazu werden die Formeln für die Zeitkonstanten entsprechend umgeformt

2. Aus den ermittelten Werten können nun die Schaltpunkte bestimmt werden.

Gegebenenfalls müssen zur Optimierung der Zelle Iterationsschritte durchgeführt werden.

4.3. NAND-Gatter: N-fach NAND-Gatter

4-fach-NAND



-Erweiterung des 2 fache NAND durch N –
nMOSFETs + pMOSFETs

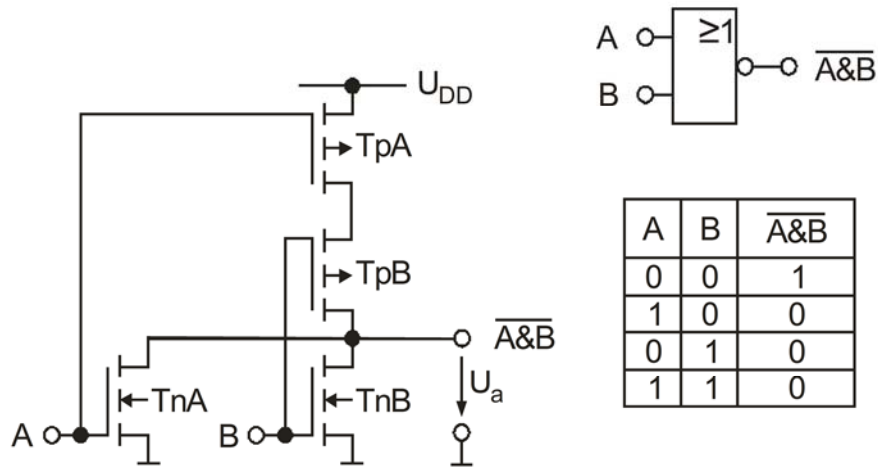
$$U_{th}^* = \frac{U_{i0n} + N \cdot \sqrt{\frac{\beta_n}{\beta_p}} \cdot (U_{DD} - |U_{i0p}|)}{1 + N \cdot \sqrt{\frac{\beta_n}{\beta_p}}}$$

1) $C_a^N = N \cdot C_a$ langsamer C_{GDP} und C_{DBP}

2) Entladung durch N – nMOSFETs
realistisch $N = 4, \dots, 5$

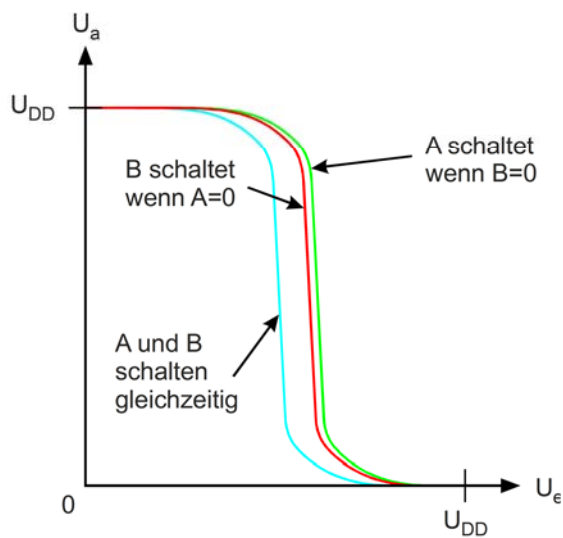
Ist dies geschehen, kann das Layout der Zelle in Angriff genommen werden.

4.4. NOR-Gatter



Für das NOR-Gatter erfolgt nun eine Analyse in der gleichen Weise wie zuvor beim NAND-Gatter.

4.4. NOR-Gatter: Statischer Betrieb



$$U^* = \frac{U_{i0n} + \frac{1}{2} \cdot \sqrt{\frac{\beta_n}{\beta_p}} \cdot (U_{DD} - |U_{i0p}|)}{1 + \frac{1}{2} \cdot \sqrt{\frac{\beta_n}{\beta_p}}}$$

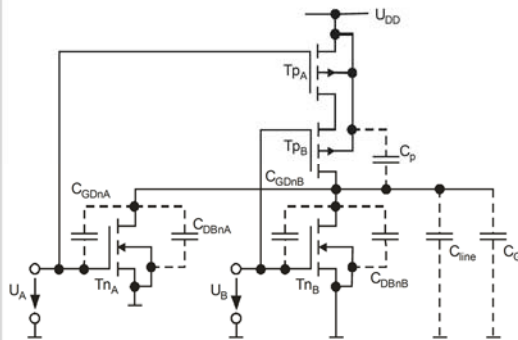
Auch hier sind die Übertragungskennlinien abhängig davon, welcher der beiden Eingänge den Umschaltvorgang auslöst.

Für den Schaltpunkt ergibt sich die gezeigte Formel.

Vergleichen wir diese mit der des NAND-Gatters erkennen wir, dass anstelle des Faktors 2 nun ein Faktor $\frac{1}{2}$ in der Gleichung auftritt.

Diese Änderung wird durch die hier "vertauschte Reihen- bzw. Parallelschaltung der n- und p-Kanal Transistoren verursacht.

4.4. NOR-Gatter: Schaltverhalten



Entladung :

$$t_{HL} = \tau_n \cdot \left[\frac{2 \cdot (U_m - U_0)}{U_{DD} - U_m} + \ln \left[\frac{2 \cdot (U_{DD} - U_m)}{U_0} - 1 \right] \right]$$

$$\tau_n = \frac{C_a}{\left(\sum_j \beta_{nj} \right) \cdot (U_{DD} - U_m)}$$

Aufladung :

$$t_D \cong R_{pA} \cdot C_{pA} + (R_{pA} + R_{pB}) \cdot C_a$$

$$t_{LH} = \tau_p \cdot \left[\frac{2 \cdot (U_{ip} - U_0)}{(U_{DD} - |U_{ip}|)} + \ln \left(\frac{2 \cdot (U_{DD} - |U_{ip}|)}{U_0} - 1 \right) \right]$$

$$C_a = [C_{GDnA} + C_{GDnB}] + K(\Delta U) \cdot [C_{DBnA} + C_{DBnB}] + \sum C_p + C_{line} + C_G$$

$$\tau_p \cong \frac{C_a}{\beta_{peff} \cdot (U_{DD} - |U_{ip}|)}$$

$$L_{p,eff} = L_{pA} + L_{pB}$$

Für die Analyse des Schaltverhaltens wenden wir auch hier die Grundlagen vom Inverter an.

Wir erkennen, dass analog zu den bisherigen Betrachtungen auch hier die Gleichungen zur Bestimmung der der Schaltzeiten "vertauscht" sind.

4.4. NOR-Gatter: Design and n-fach-Gatter

1. Schaltverhalten $\rightarrow$ Aspektverhältnis

$$\text{High-Low: } \left(\frac{w}{L}\right)_n = \frac{C_a}{\beta_n' \cdot \tau_p \cdot (U_{DD} - U_{in})}$$

$$L \rightarrow H: \left(\frac{w}{L}\right)_p = \frac{2 \cdot C_a}{\beta_n' \cdot \tau_p (U_{DD} - |U_{tp}|)}$$

2. Setzen von U^* $\rightarrow$ Optimierung von $\left(\frac{\beta_n}{\beta_p}\right)$

n -fach NOR:

$$U^* = \frac{U_{i0n} + \left(\frac{1}{N}\right) \cdot \sqrt{\frac{\beta_n}{\beta_p}} \cdot (U_{DD} - |U_{i0p}|)}{1 + \left(\frac{1}{N}\right) \cdot \sqrt{\frac{\beta_n}{\beta_p}}} \quad N_{\max} = 4, \dots, 5$$

Auch für das Design dieser Zelle gelten die gleichen Vorgehensweisen wie beim NAND-Gatter.

4.5. Vergleich zwischen NAND- und NOR-Gatter

NAND und NOR = CMOS- Design

- Bei gleichen N und gleicher Fläche sind NAND schneller!

Why?

Serienschaltung der Transistoren begrenzt die Umschaltzeit!

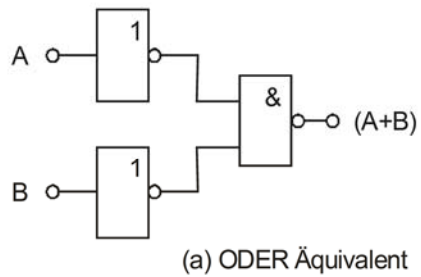
→ *NAND* – $H \rightarrow L$ über n – MOSFET – Kette

→ *NOR* → $L \rightarrow H$ über p – MOSFET – Kette

da $R_n < R_p$

$$\tau_{D,NAND} < \tau_{D,NOR}$$

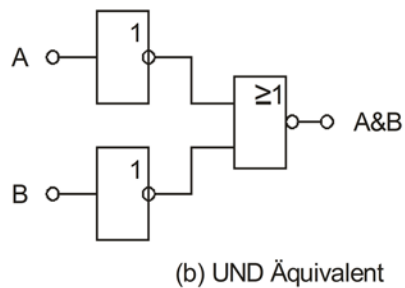
4.6. OR- und AND-Gatter



DeMorgan- Regeln:

$$F_1 = \overline{\overline{A} \& \overline{B}} = A + B \quad OR$$

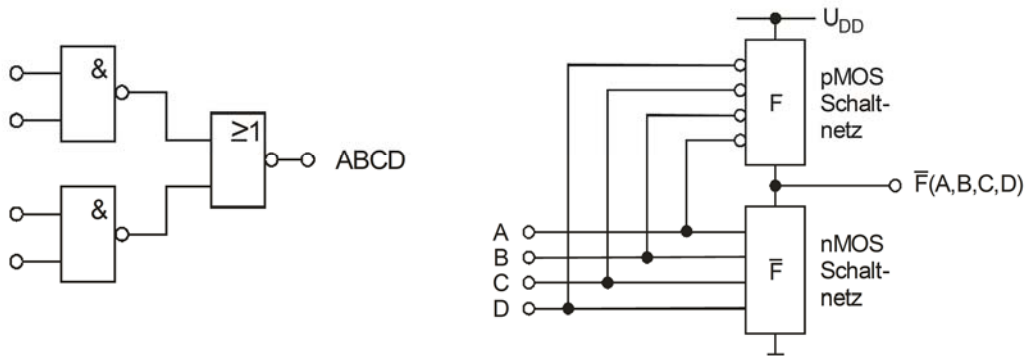
$$F_2 = \overline{\overline{A} + \overline{B}} = A \& B \quad AND$$



Um OR bzw. AND-Gatter aufzubauen gibt es 2 Möglichkeiten:

1. Das Nachschalten eines Inverters an das NOR bzw. NAND-Gatter,
2. das Vorschalten von Inverters und ein "Vertauschen" der Logikfunktion, wie dies auf dieser Folie gezeigt wird.

4.7. Kombinatorische Logik



$$F = \overline{A \& B + D \& C} = A \& B \& C \& D$$

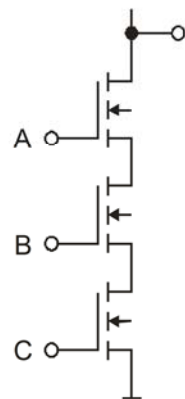
N – Inputs $\rightarrow 2N$ – MOSFETs

- Regeln – jeder Eingang ist mit Gate von pMOS und nMOS verbunden
– pMOS nMOS sind komplementär

Diese Art des Aufbaus von Logikschaltkreisen wird auch als kombinatorische Logik bezeichnet.

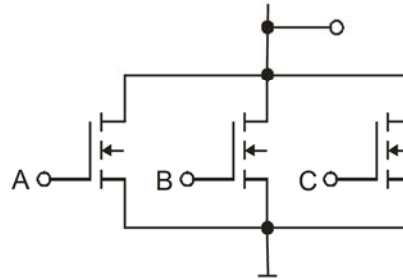
Für eine allgemeine Betrachtung geht man wieder davon aus, dass man Schaltnetze aus nMOS und pMOS verwendet und jeder Eingang der Schaltung immer ein komplementäres Transistorpaar ansteuert.

4.7. Kombinatorische Logik: Funktionsblöcke



$$F = A \& B \& C$$

(a) Regel 1:
Serienschaltung
von n-MOSFETs

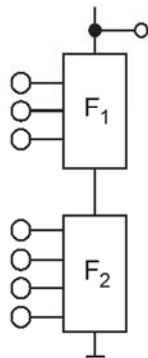


$$F = A + B + C$$

(b) Regel 2:
Parallelschaltung
von n-MOSFETs

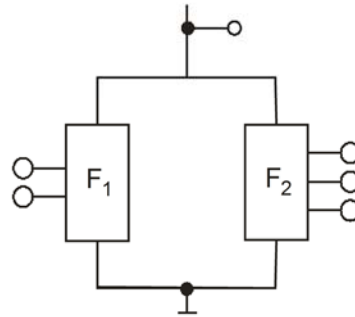
Zur Realisierung der Logikfunktion kann man die nMOS-Tr. Entweder in Reihe (-> NAND) oder parallel (-> NOR) schalten.

4.7. Kombinatorische Logik: Funktionsblöcke



$$F = F_1 \& F_2$$

(c) Regel 3:
Serienschaltung
von Logik-Blöcken

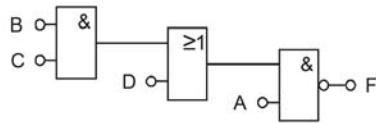


$$F = F_1 + F_2$$

(d) Regel 4:
Parallelschaltung
von Logik-Blöcken

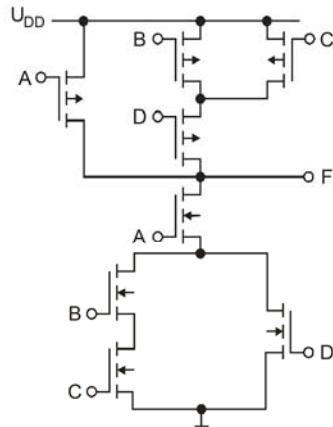
Die nächsten Regeln sagen, dass wir diese nMOS-Funktionsblöcke entweder in Reihe oder parallel schalten können.

4.7. Kombinatorische Logik: Beispiel (AOI)



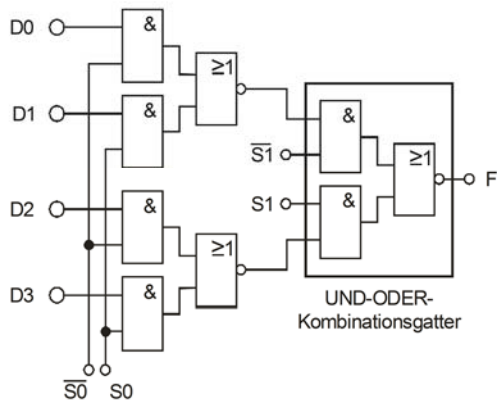
$$F = \overline{A \cdot B \cdot C + A \cdot D}$$

$$F = \overline{A \cdot (B \cdot C + D)}$$



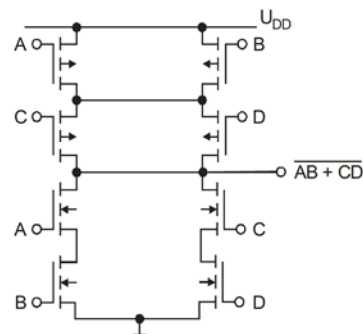
Daraus lässt sich dann eine so genannte AND-OR-Invert –Funktion zusammenbauen. Hier ist eine der möglichen Schaltungsvarianten gezeigt.

4.7. Kombinatorische Logik: Beispiel (MUX)



S0	S1	F
0	0	D0
1	0	D1
0	1	D2
1	1	D3

$$F = D_0 \cdot \overline{S_0} \cdot \overline{S_1} + D_1 \cdot S_0 \cdot \overline{S_1} + D_2 \cdot \overline{S_0} \cdot S_1 + D_3 \cdot S_0 \cdot S_1$$

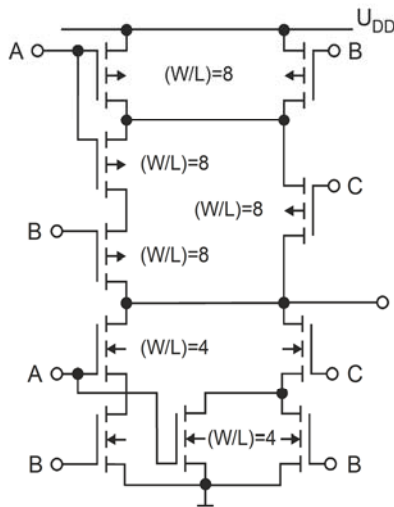


Ein weiteres Beispiel für kombinatorische Logik ist der Schaltungsaufbau eines digitalen Multiplexers.

Über 2 Auswahlleitungen wird einer von vier Eingängen ausgewählt und an den Ausgang weitergeleitet.

Die Realisierung des AND-OR Gatters ist rechts unten zu sehen.

4.7. Kombinatorische Logik: Design



$$F = A \& B + A \& C + B \& C = A \& B + (A + B)C$$

1) Berechne C_a ;

Design eines Inverters mit Ziel: Transient $\left(\frac{w}{L}\right)_{n,inv}$; $\left(\frac{w}{L}\right)_{p,inv}$

2) Konstruiere den nMOS-Block

Bestimme max. Zahl m der in Serie geschalteten nMOSFETs

$$\left(\frac{w}{L}\right)_n = m \cdot \left(\frac{w}{L}\right)_{n,inv}$$

3) Konstruiere den pMOS-Block

Bestimme max. Zahl der parallel geschalteten pMOSFETs

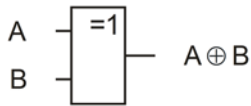
$$\left(\frac{w}{L}\right)_p = m \cdot \left(\frac{w}{L}\right)_{p,inv}$$

Auch für Logikzellen, die mit kombinatorischer Logik aufgebaut sind, müssen bestimmte Regeln beim Design beachtet werden.
Hier ein Beispiel

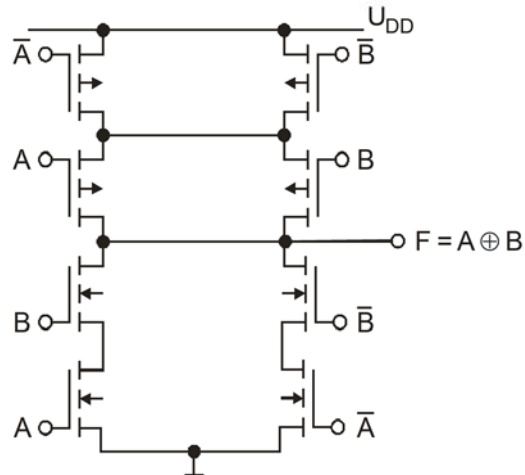
4.8. XOR (Antivalenz) -Gatter

XOR-Gatter

$$F = A \oplus B = A \& \bar{B} + \bar{A} \& B$$



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

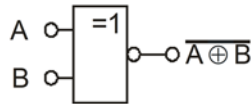


Mithilfe von AND-OR Gattern werden auch die Antivalenzschaltung (diese Folie) und die Äquivalenzschaltung (nächste Folie) aufgebaut.
Wie man aber erkennen kann, werden zusätzlich noch 2 Inverter benötigt, um die invertierten Signale zu generieren.

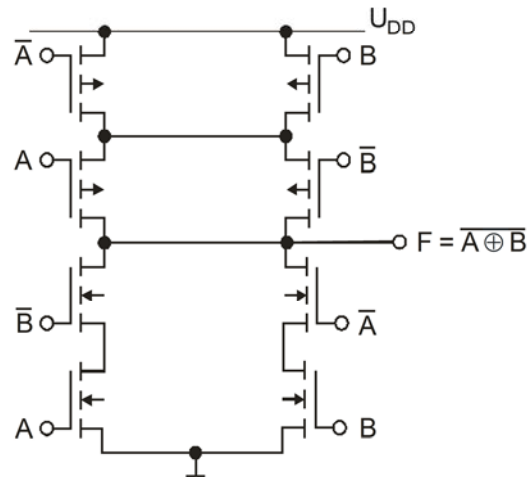
4.8. XNOR- und Äquivalenz-Gatter

Äquivalenz-Gatter

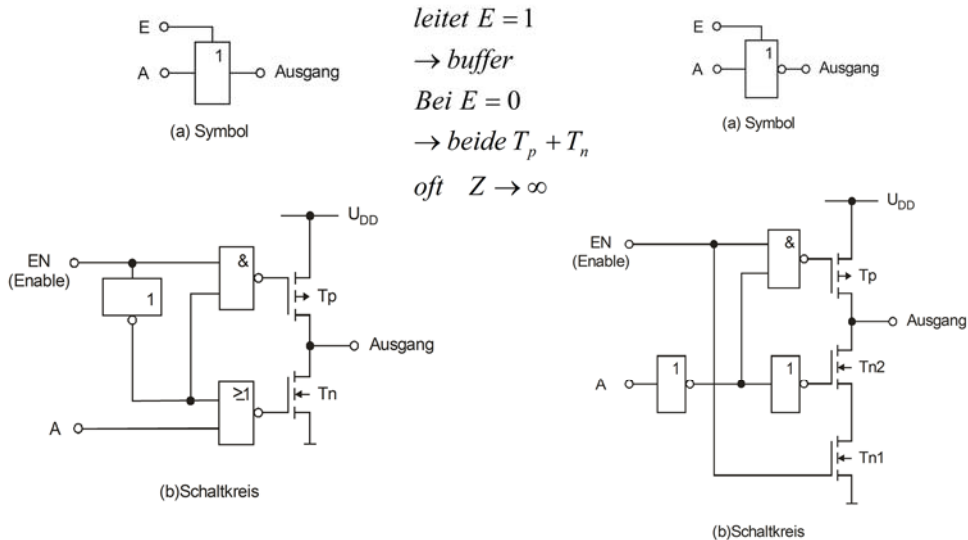
$$F = \overline{A \oplus B} = A \& B + \overline{A} \& \overline{B}$$



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



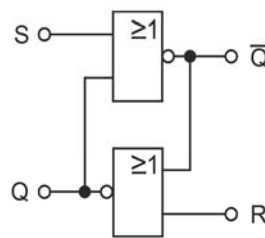
4.9. Tri-State Ausgang



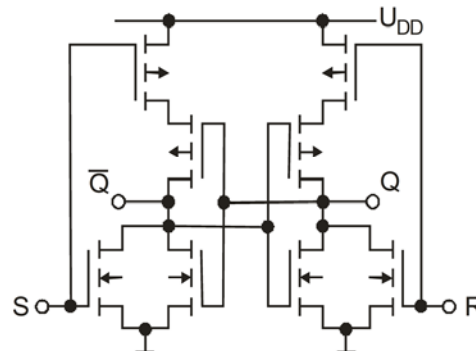
Tri-State Schaltungen werden oft als Ausgangsstufen für Bussysteme eingesetzt. Neben den beiden logischen Ausgangs-Zuständen H und L kann über den Enable-Eingang ein dritter Zustand: "hochohmig" eingestellt werden. Es gibt eine Reihe von unterschiedlichen Auslegungen für eine solche Stufe. Hier sind 2 davon gezeigt.

4.11. Flip-Flop mit Gattern

S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	0	0



SR-FF mit NOR-Gattern

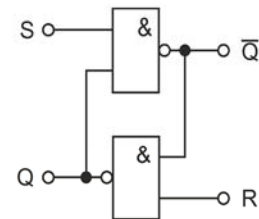


Weitere Beispiele für digitale Standardzellen sind verschiedene Flipflops aus Gattern. Hier ein RS-FF, das mit zwei NOR-Gattern aufgebaut ist. Die Problematik bei dieser Art von FF-Schaltungen ist, dass für bestimmte Eingangszustände (hier: 1,1) die Ausgangssignale nicht den logischen Deklarationen entsprechen. Deshalb darf diese Eingangskombination nicht angelegt werden!

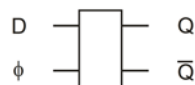
4.11. Flip-Flop mit Gattern

SR-FF mit NAND-Gattern

S	R	Q	$\bar{Q}$
0	0	1	1
1	0	1	0
0	1	0	1
1	1	Q	$\bar{Q}$

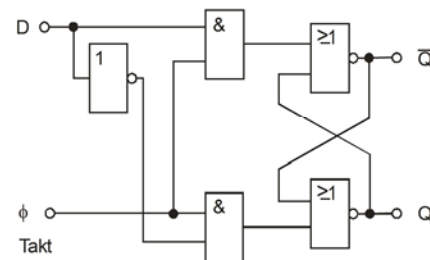


D-FF mit NAND-Gattern



(a) Symbol

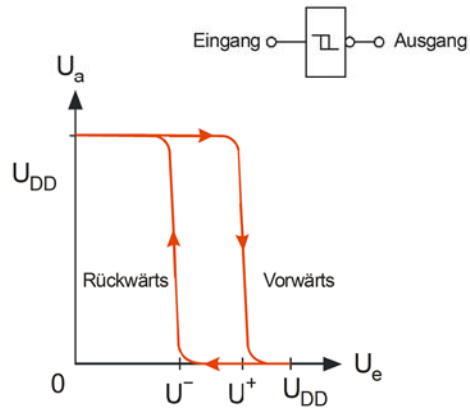
$\phi = 1 \rightarrow \text{Data latched in FF}$



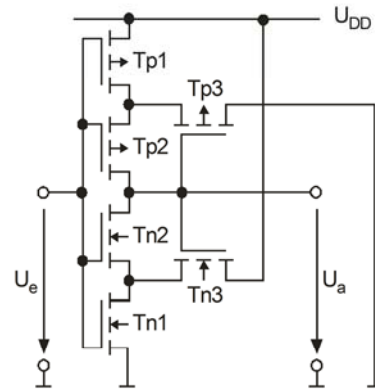
(b) Schaltkreis

Ein RS-FF, das mit NAND-Gattern aufgebaut ist, unterliegt den gleichen Einschränkungen wie das auf der vorangegangenen Folie.
Erst zusätzliche Logikschaltungen an den Eingängen verhindern diese Restriktionen.
Das Einfachste diese Flipflops ist das hier gezeigte D-FF.

4.12. Schmitt-Trigger



$$U_H = U^- - U^+$$



Müssen Signale "konditioniert" werden, bieten sich Schaltungen, wie der Schmitt-Trigger an.

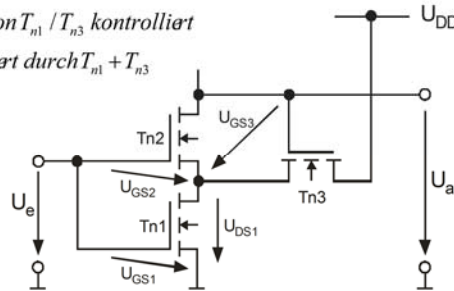
Ein Schmitt-Trigger hat eine Hysterese, d.h. die Schaltpunkte sind nicht wie bei den einfachen Gattern sowohl beim Übergang von H->L wie auch beim Übergang von L->H an der gleichen Stelle der Übertragungsfunktion.

Der Vorteil des Schmitt-Triggers ist es, das langsame oder mit Störungen behaftete Signale in einwandfreie logische Pegel mit kurzen Anstiegs- und Abfallzeiten umgewandelt werden.

4.12. Schmitt-Trigger: n-MOS-Analyse

$U_{DS1} \rightarrow$ wird von T_{n1} / T_{n3} kontrolliert

U^+ ist kontrolliert durch $T_{n1} + T_{n3}$



$U_e = 0 \text{ V}$ – steigt an!

$$U_{GS1} = U_e$$

$$U_{GS2} = U_e - U_{DS1}$$

$$U_{GS3} = U_a - U_{DS1}$$

T_{n1} schaltet bei $U_{GS1} = U_{m1}$

T_{n2} bei $U_e = U_{m2} + U_{DS1} \equiv U^+$

$$\text{pMOS - Hälfte} \quad U^- = \frac{\sqrt{\frac{\beta_{p1}}{\beta_{p3}} \cdot (U_{DD} - U_p)}}{1 + \sqrt{\frac{\beta_{p1}}{\beta_{p3}}}}$$

$$U_{DS1} = U^+ - U_m \quad T_{n2} \rightarrow \text{on}$$

$\rightarrow T_{n1} \rightarrow \text{Sättigung}$

nMOS - Hälfte

$$I_1 = \frac{\beta_{n1}}{2} \cdot (U^+ - U_m)^2 \quad U^+ = \frac{U_{DD} + \sqrt{\frac{\beta_{n1}}{\beta_{n3}}} \cdot U_m}{1 + \sqrt{\frac{\beta_{n1}}{\beta_{n3}}}}$$

$$I_3 = \frac{\beta_{n3}}{2} \cdot (U_{DD} - U^+)^2 \quad \text{d.h.} \quad \frac{\beta_{n1}}{\beta_{n3}} = \left(\frac{w/L_1}{w/L_3} \right)^2$$

Zur Einstellung der Schaltpegel und der Hysterese muss eine Analyse des Schaltungsdesigns durchgeführt werden.

Dazu werden die beiden Hälften der Schaltung getrennt voneinander betrachtet.

4.12. Schmitt-Trigger: Gesamtdesign

Symmetrischer Schmitt-Trigger

$$\left. \begin{aligned} U^+ &= \frac{1}{2} \cdot U_{DD} + \Delta U \\ U^- &= \frac{1}{2} \cdot U_{DD} - \Delta U \end{aligned} \right\} U_H = 2 \cdot (\Delta U)$$

$$\text{mit } \beta_{rel} = \frac{\beta_{n1}}{\beta_{n3}} = \frac{\beta_{p1}}{\beta_{p3}} \quad \text{and} \quad U_{m0} = |U_{tp0}| = U_t$$

$$\Delta U = \frac{U_{DD} \cdot (1 - \sqrt{\beta_{rel}}) + 2 \cdot \sqrt{\beta_{rel}} \cdot U_t}{2 \cdot (1 + \sqrt{\beta_{rel}})}$$

$$\rightarrow \sqrt{\beta_{rel}} = \frac{U_{DD} - 2 \cdot \Delta U}{U_{DD} + 2 \cdot \Delta U - 2 \cdot U_t}$$

Für ein symmetrisches Design lässt sich die Hysterese mit den gezeigten Formeln gut abschätzen.