

# **Physikalische Grundlagen (phybasics)**

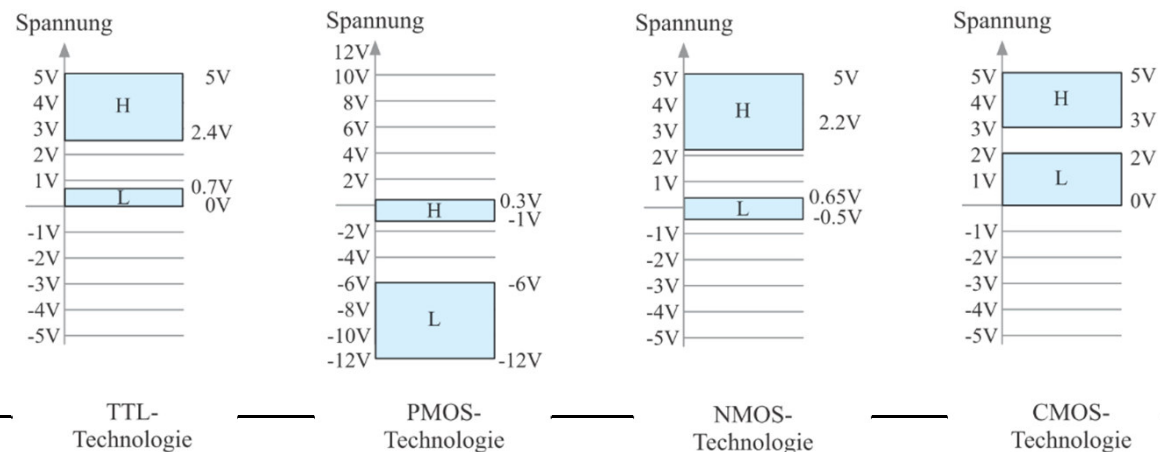
---

## **Physikalisches Verhalten logischer Gatter (phyverh/phybehav) - Einführung/Motivation**

- Bisher haben wir uns mit logischen Gattern, Ihre Anwendungen (kombinatorische Logik), und Ihre physikalische Realisierung (z.B. CMOS) sowie ihre tatsächliche industrielle Realisierung (Produktion) angesehen
- Was wir final noch betrachten müssen, um alle Zusammenhänge verstehen zu können ist das physikalische Verhalten der Gatterrealisierungen

## Phyverh - Positive und Negative Logik I/II

- Die Schaltkreisfamilien arbeiten physikalisch verschieden, aber gemeinsam ist, dass sie binär arbeiten und dass sie die Logikzustände anhand von Spannung unterscheiden
- Jedem Logikzustand (1 oder 0) wird dabei ein Pegelbereich zugewiesen (H(igh)-Pegel und L(ow)-Pegel)
  - Der H-Pegel ist dabei der Spannungsbereich der näher an  $+\infty$  liegt
  - der L-Pegel ist der Bereich, der näher an  $-\infty$  liegt



## Phyverh - Positive und Negative Logik II/II

- Typischerweise ordnet man dem L-Pegel den Wert 0 zu und dem H-Pegel den Wert 1
  - Dies nennt man dann positive Logik
- Es gibt aber auch die negative Logik
  - Hier ordnet man dem L-Pegel den Wert 1 und dem H-Pegel den Wert 0 zu

	Pegel		
	x	y	z
0	L	L	L
1	L	H	L
2	H	L	L
3	H	H	H

	Positive Logik		
	x	y	z
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

	Negative Logik		
	x	y	z
0	1	1	1
1	1	0	1
2	0	1	1
3	0	0	0

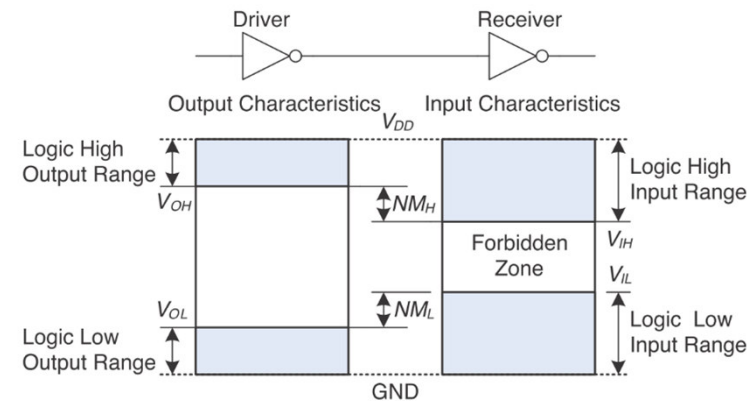
## Phyverh - Störsicherheit / Störabstand - Motivation und Prinzip I/II

- Die Beschränkung auf endliche Amplitudenstufen in einem digitalen System führt zu einer erhöhten Störsicherheit
- Gestörte digitale Signale können den ursprünglichen diskreten Amplitudenwerten eindeutig zugeordnet werden
- Beispiel
  - positive Logik (L=0, H=1)
  - Der Ausgang des Senders (driver) ist mit dem Eingang des Empfängers (receiver) verbunden



## Phyverh - Störsicherheit / Störabstand - Prinzip II/II

- Wenn der driver eine 1 also bei positiver Logik einen HIGH-Pegel schickt er eine Spannung zwischen  $V_{OH}$  und  $V_{DD}$
- Der Empfänger erkennt die 1, wenn die Spannung zwischen  $V_{DD}$  und  $V_{IH}$  liegt
- Erst wenn die Spannung in die forbidden zone fällt, ist das Verhalten unvorhersagbar
- D.h. Eine Übertragung funktioniert immer noch richtig, selbst wenn eine Störung vorhanden ist, die Ausgangsspannung um maximal  $NM_H$  verringert
- NM steht für noise margin (dt. Störabstand)



$$NM_L = V_{IL} - V_{OL}$$

$$NM_H = V_{OH} - V_{IH}$$

## **Phyverh - Störsicherheit / Störabstand - Anwendung**

- Dies ist ein riesiger Vorteil der Übertragung von digitalen Signalen im Vergleich zu analogen Signalen, wo jede auch noch so kleine Störung die Empfangsqualität beeinflusst
- Dies ist der Grund für den Siegeszug der Digitaltechnik
- Ohne diese Eigenschaft wäre die weltweite (störungsfreie) langstecken Kommunikation für Telefon und Internet, wie es es heute gibt, nicht so ohne weiteres umsetzbar

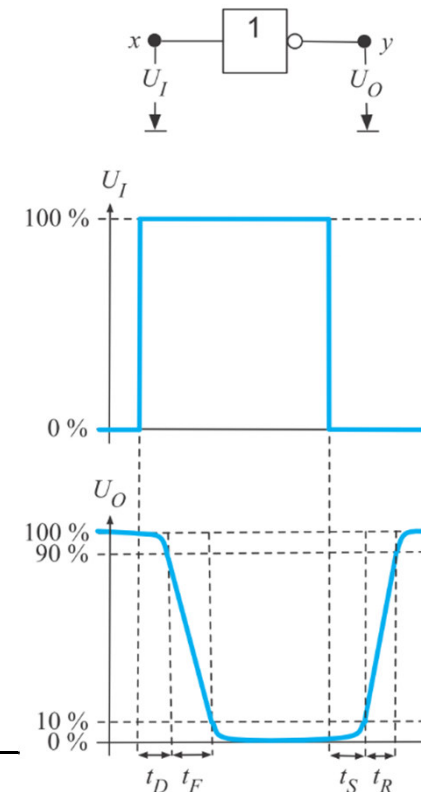
## **Phyverh - Schaltzeiten - Motivation**

- Bisher haben wir hauptsächlich betrachtet, ob eine Schaltung (überhaupt) funktioniert
- In der Praxis ist es aber auch wichtig zu wissen wie schnell eine Schaltung funktioniert
- Die Ausbreitungsgeschwindigkeit in typischen Leitern beträgt zwar in etwa die 0,6-fache Lichtgeschwindigkeit ist aber nicht unendlich



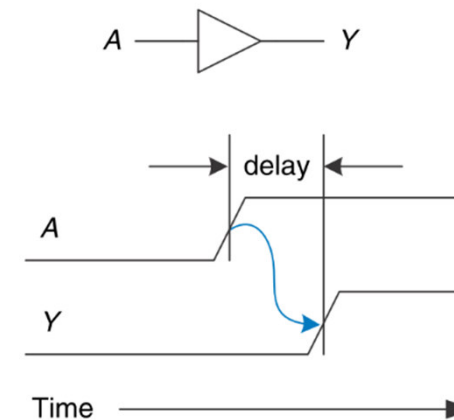
## Phyverh - Schaltzeiten - Wannenkurve (B)

- Aus dem Anlegen einer idealen Rechteckkurve am Eingang wird eine Wannenkurve am Ausgang
- Die nicht idealen Eigenschaften der Logikzelle verbreitert oder verkürzt die Signalflanke ( $T_f$  und  $T_r$ )
- Zusätzlich zur Verformung wird das Signal zeitlich verschoben ( $T_d$  bzw.  $T_{pd}$  und  $T_s$ )
- $T_d$  bzw.  $T_{pd}$  = delay time;  $T_s$  = storage time;  $T_f$  = fall time,  $T_r$  = rise time



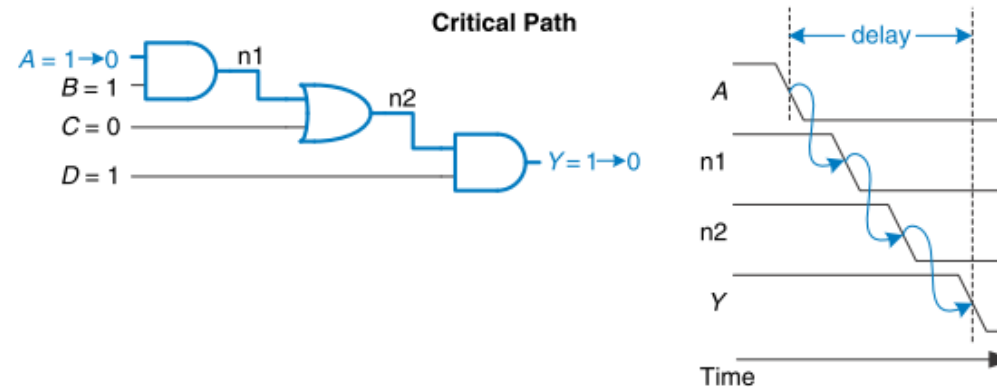
## Phyverh - Schaltzeiten - Propagation delay

- Der wichtigste Parameter ist aber das delay (auch propagation delay  $T_{pd}$  genannt)
- $T_{pd}$  ist die maximale Zeit von der Eingangsänderung bis der Ausgang sich ändert
- $T_{pd}$  wird in einem timing diagram von 50% der steigenden Flanke (rising edge) des Eingangssignals zu 50% der steigenden Flanke des Ausgangssignals bestimmt
- $T_{pd}$  zu berechnen ist nicht so trivial (Transistordetails!), aber wird typischerweise in Datenblättern der Hersteller angegeben und befindet sich üblicherweise im Bereich Pico- bis zig/hundert Nanosekunden



## Phyverh - Schaltzeiten - Critical path

- Tpd einer Schaltung wird entlang eines Pfades bestimmt und Gatter für Gatter aufsummiert
- Beispiel:

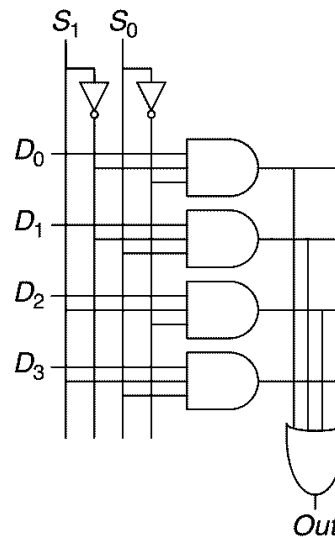


- Der kritische Pfad ist dabei der längste und damit langsamster Pfad durch eine Schaltung
- Der kritische Pfad begrenzt die Geschwindigkeit der Schaltung

## Phyverh - Schaltzeiten - Propagation delay - Übung

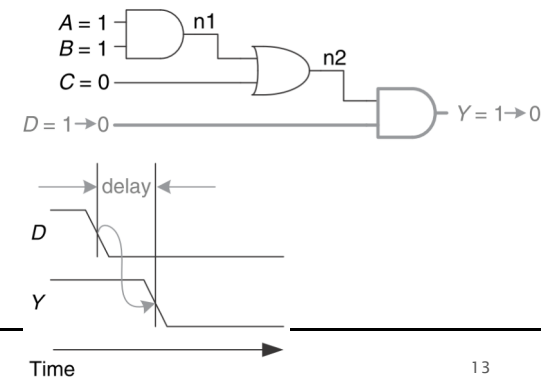
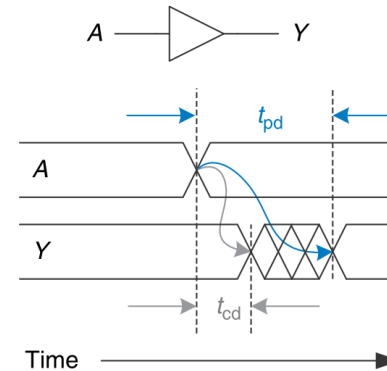
- Berechnen Sie  $T_{pd\_sy}$  (von S nach y) und  $T_{pd\_dy}$  (von D nach y) für folgende Schaltung (Multiplexer 4:1)
- Was ist der kritische Pfad? Kennzeichnen Sie ihn.

Gate	$t_{pd}$ (ps)
NOT	30
2-input AND	60
3-input AND	80
4-input OR	90



## Phyverh - Schaltzeiten - Contamination delay (B)

- Ein weiterer wichtiger Parameter ist aber das contamination delay  $T_{cd}$
- $T_{cd}$  ist die minimale Zeit von der Eingangsänderung bis der Ausgang sich ändert
- $T_{cd}$  einer Schaltung wird entlang eines Pfades bestimmt und Gatter für Gatter aufsummiert
- Der kürzeste Pfad (shortest path) ist dabei der kürzeste und damit schnellste Pfad durch eine Schaltung

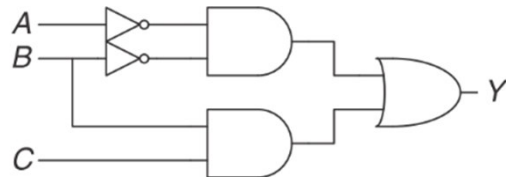


## **Phyverh - Glitches - Motivation**

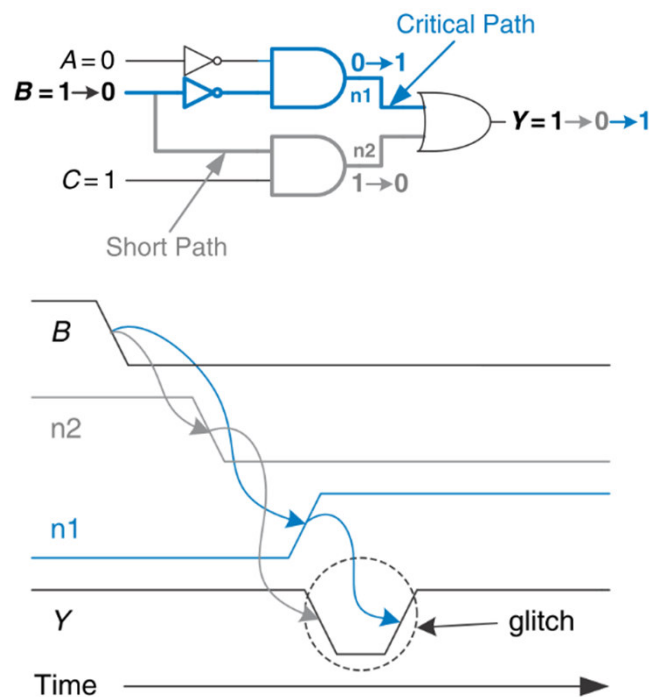
- Als glitch oder hazard (dt. Störimplus) bezeichnet man einen kurzzeitigen Wechsel Pegels eines Signals, dass mit der bool'schen Algebra nicht erklärbar ist
- So kann beispielsweise ein Signalwechsel am Eingang zu mehreren Signalwechseln am Ausgang führen
- Normalerweise verursachen glitches (in synchroner Logik) keine Probleme
- Trotzdem ist es aber es wichtig, zu realisieren, dass es sie gibt und sie in Timing-Diagrammen zu erkennen (Oszilloskop, Logikanalyser, Simulation)

## Phyverh - Glitches - Beispiel

- Beispiel  $A=0$ ,  $C=1$  und  $B$   $1 \rightarrow 0$



- Short Path  $n2 \rightarrow 0$ ;  $y \rightarrow 0$ ; critical Path  $n1 \rightarrow 1$ ;  $y \rightarrow 1$
- $n2$  fällt auf 0 bevor  $n1$  steigt;  $y$  fällt auf 0; schließlich steigt  $n1$ ; dann steigt auch  $y$  wieder auf 1
- Solange wir  $T_{pd}$  abwarten können bis wir das Ergebnis auszuwerten, sind gliches kein Problem (typ. In synchroner Logik)



## Phyverh - Lastfaktoren - Motivation

- Auf der abstrakten Ebene der bool'schen Algebra kann man beliebig viele Gatter an einen Ausgang hängen und damit beliebig Komplexe Ausdrücke bilden
- In der Praxis werden den Transistoren durch die Halbleiterphysik deutliche Grenzen gesetzt
- Die Grenzen werden durch die Spannung und den Strom gesetzt, den jedes Halbleiterelement braucht
- Beispiel:
  - Wird ein Ausgang eines Gatters mit zu vielen Eingängen anderer Gatter verbunden, so sinkt der Ausgangspegel unter die gültigen Bereiche (vgl. H- und L-Pegeldefinitionen) ab
  - Die Schaltvorgänge können nicht mehr richtig durchgeführt werden

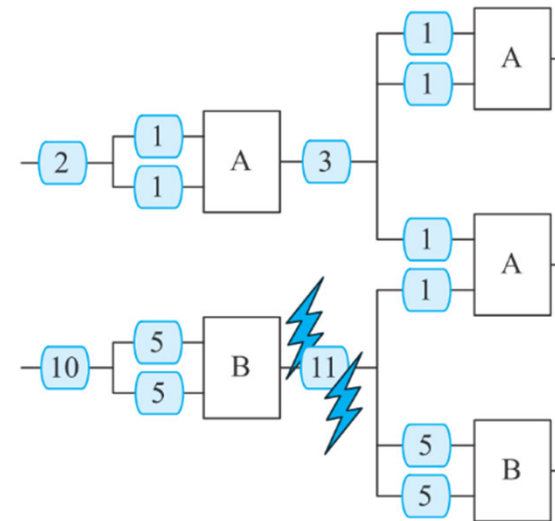


## Phyverh - Lastfaktoren - Prinzip

- Wie sich nun die Gatter im Zusammenhang mit anderen Gattern verhalten beschreiben die Lastfaktoren
  - Eingangslastfaktor (Fan-In)
    - Der Fan-In gibt an wie stark ein Gatter den Ausgang des vorhergehenden Gatters belastet
    - Normale elementare Gatter haben einen Fan-In von 1
  - Ausgangslastfaktor (Fan-Out)
    - Der Fan-Out eines Gatters gibt an, wie viele nachgeschaltete Gatter maximal gespeist werden können (bis die vorgegebenen Pegelwerte verletzt werden)
    - Wenn alle Gatter einen Fan-In von 1 haben gibt der Fan-Out die Anzahl der anschließbaren Gatter an

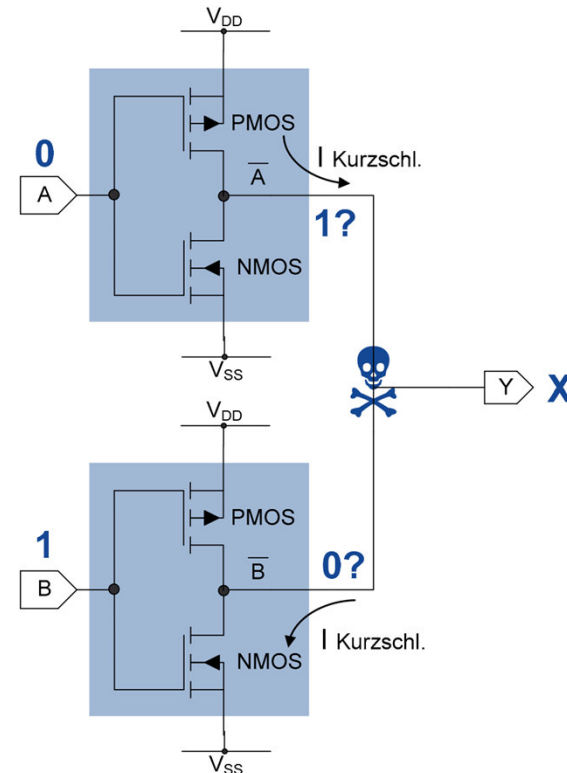
## Phyverh - Lastfaktoren - Beispiel

- Beispiel:
  - Gatter Typ A: Fan-In 1; Fan-Out-10
  - Gatter Typ B: Fan-In 5; Fan-Out-10
- Bei dem unteren linken B-Gatter tritt eine Überlast auf -> Ersetzen durch Gatter mit höherem Fan-Out oder Einfügen von zusätzlichen Zwischengliedern (buffer)



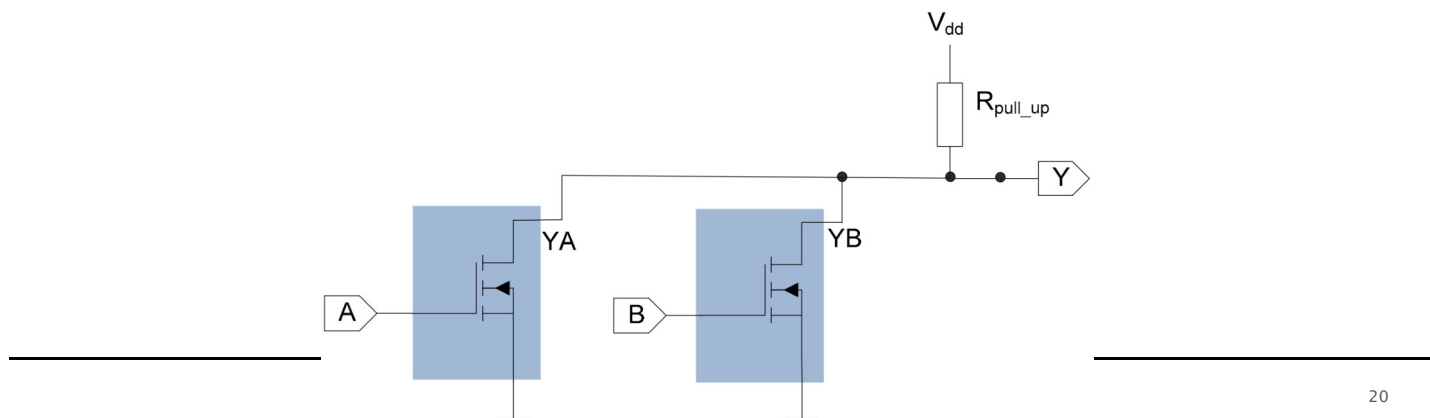
## Phyverh - Ausgabeschaltungen - Motivation keine Ausgänge zusammenschalten

- Als Letztes haben wir betrachtet wieviel Eingänge an einen Ausgang angeschlossen werden können
- Was wir noch nicht diskutiert haben ist, dass Ausgänge von Standardgattern nicht zusammengeschlossen werden dürfen
- Der obere Inverter (InvA) mit  $A=0$  und der unter Inverter (InvB) mit  $B=1$  angesteuert
- Der PMOS des InvA und der NMOS des InvB schaltet
- Es kommt zum Kurzschluss über PMOS des InvA und NMOS des InvB
- Das zerstört die Gatterausgänge und führt zu einem undefinierten Pegel X



## Phyverh - Ausgabeschaltungen - Open-Drain / Open-Collector Ausgang - Motivation und Prinzip

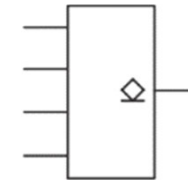
- Wie wir gesehen haben, dürfen Standardausgänge nicht zusammengeschlossen werden
- Es gibt aber Ausgangstypen wie den Open-Drain (bei CMOS) oder Open-Collector (Bei TTL), die zusammengeschlossen werden dürfen
- Beim Open Drain, wird der PMOS durch einen Widerstand ersetzt und der Drain-Anschluss des NMOS nach außen geführt
- Dieser Widerstand (auch Pull-up Widerstand genannt) stellt die Verbindung zu  $V_{dd}$  her und stellt sicher, dass der Ausgangspegel auf H-Pegel gebracht wird, falls keiner der NMOS-Transistoren durchgeschaltet ist



---

## Phyverh - Ausgabeschaltungen - Open-Drain / Open-Collector Ausgang - Anwendung

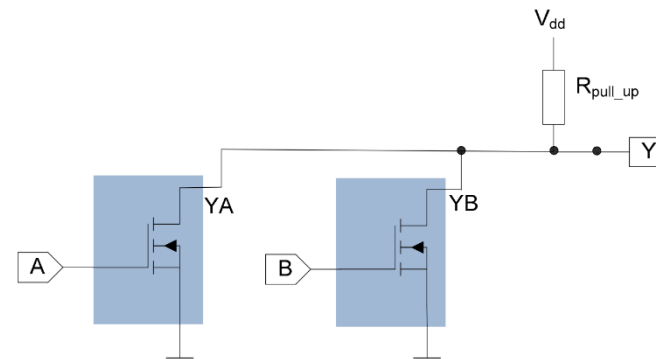
- Open-Drain Ausgänge werden durch ein Rautensymbol mit Balken gekennzeichnet (Der Unterstrich/Balken unten steht dabei für die L-Dominanz)



- Anwendung Open-Drain Schaltung:
  - Anschluss von einer Vielzahl von Sensoren an einen Mikrocontroller
  - Die Sensoren werden alle am Ausgang mit einem Pull-up-Widerstand verknüpft und an einem Port des Mikrocontrollers angeschlossen

## Phyverh - Ausgabeschaltungen - Open-Drain / Open-Collector Ausgang - Wired AND

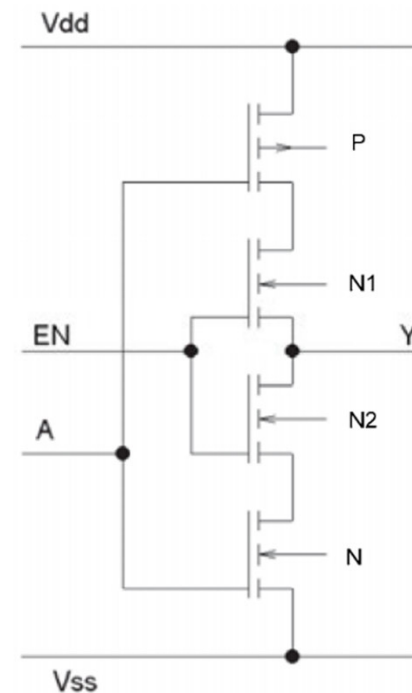
- Nun nehmen wir an, dass die Gatter vor dem Zusammenschließen die Pegel YA und YB hatten
- Nun spielen wir die 4 Kombinationsmöglichkeiten der Pegel durch
- Am Ausgang ergibt sich immer ein L-Pegel, außer beide Transistoren sind nicht durchgeschaltet und haben H-Pegel
- Das ist genau die Wertetabelle der AND-Funktion
- Deshalb wird diese Art der Verknüpfung auch wired-AND genannt, da sich die Verdrahtung eine weitere AND-Funktion ergibt



A	YA	B	YB	Y
H	L	H	L	L
H	L	L	H	L
L	H	H	L	L
L	H	L	H	H

## Phyverh - Ausgabeschaltungen - Tri-State Ausgang - Motivation und Prinzip I/II

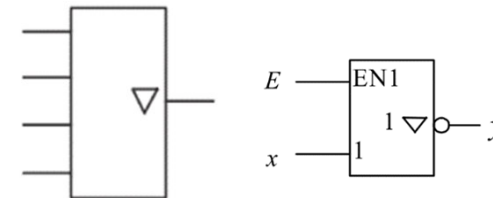
- Ein weitere Ausgangstyp, der die Zusammenschaltung von Ausgängen erlaubt ist der Tri-State Ausgang
- Die Treiberlogik innerhalb eines Gatters wird abgeschaltet
- Dies ermöglicht bidirektionale Busverbindungen zwischen Komponenten, die abwechselnd (time-multiplex) als Sender oder Empfänger betrieben werden
- Damit überhaupt hier in diesem CMOS-Inverter-Beispiel Strom zum Ausgang Y gelangen kann, muss N1 und N2 geschaltet sein
- Dies ist der Fall, wenn an EN ein H-Pegel anliegt



## Phyverh - Ausgabeschaltungen - Tri-State Ausgang - Prinzip II/II

- Wenn EN aber einen L-Pegel hat, so ist die Verbindung zu Vdd bzw. Vss hochohmig
- Man bezeichnet den Ausgang als „floatend“ da sich das Potential (Spannung) am offenen Ausgang durch nicht ideale Leckströme frei zwischen L und H einstellen kann
- Die Schaltung kann also neben H und L noch eine dritten Zustand der mit „Z“ (hochohmig, high ImpedanZ) annehmen
- Tri-State Ausgänge sind im Schaltsymbol durch ein auf der Spitze stehendes Dreieck gekennzeichnet

EN	A	Y
L	L	Z
L	H	Z
H	L	H
H	H	L





## Phyverh - Ausgabeschaltungen - Tri-State Ausgang - Anwendung Bussystem

- Mehrere Tri-State-Ausgänge dürfen in einer Busschaltung zusammengeschlossen werden, wenn sichergestellt ist, das zu jedem beliebigen Zeitpunkt nur einer der Enable-Eingänge aktiv ist
- Wenn EN\_1 -> OUT\_1 an Bus und IN\_1; Bus -> IN\_2
- Wenn EN\_2 -> OUT\_2 and Buss und IN\_2; Bus -> IN\_1

