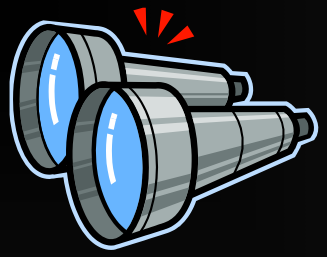
Aufbau logischer Gatter

Vom Feldeffekt-Transistor zum Supercomputer



Überblick

- Was ist CMOS ?
- Feldeffekt-Transistor & CMOS-Prozess
- kombinatorische Logikzellen
- sequenzielle Logikzellen
- weitere Logikfamilien

Der Feldeffekt-Transistor

... hat 3 Anschlüsse: Gate, Drain, Source

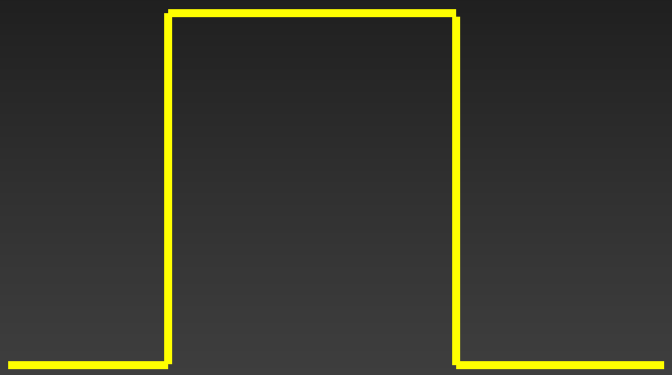
... funktioniert bei richtiger Auslegung
wie ein Schalter

... ist aber bei genauerer Betrachtung eigentlich ein analoges Bauelement

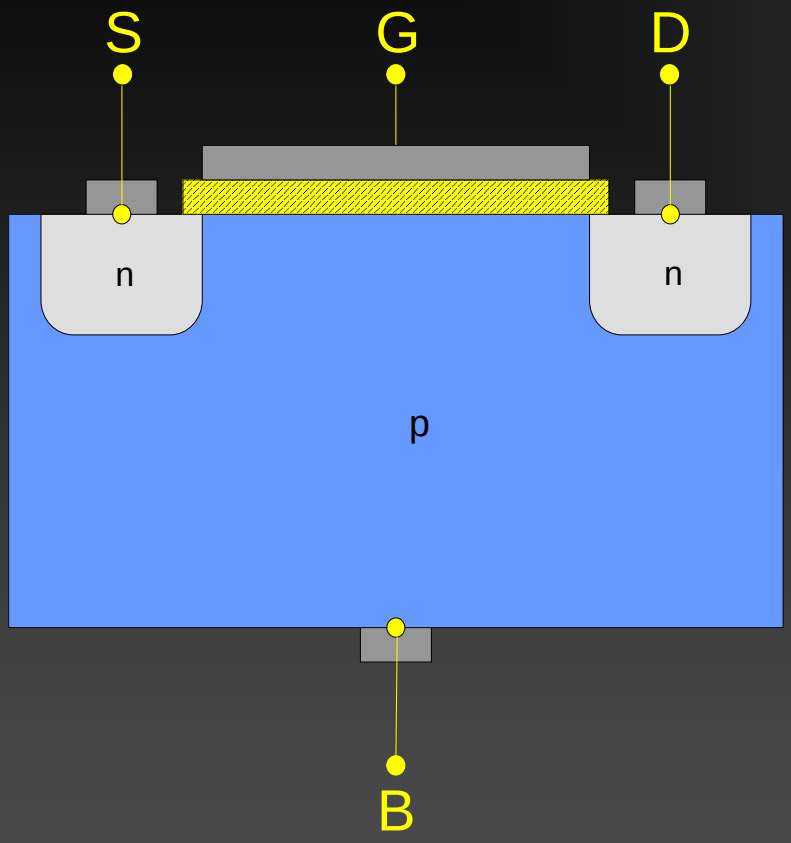
- analoge Zustandsübergänge (Schaltflanken)
- begrenzte Schaltzeiten
- Einschwingen und Überschwingen, etc.

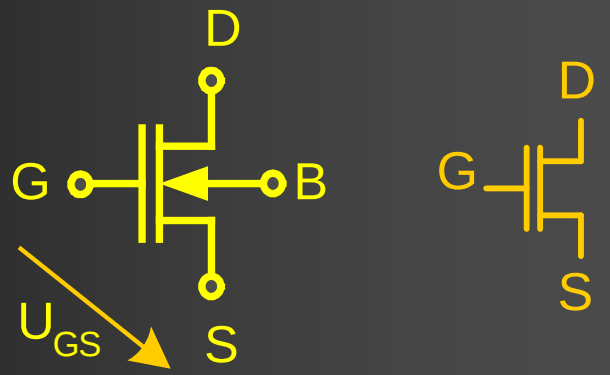
Schaltvorgang: Ideal & Realität

- Idealisierung
- Realität



n-Kanal Enhancement MOSFET

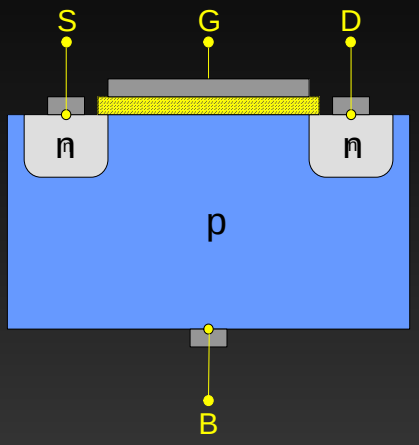




bei $U_{GS} = 0$
kein Stromfluß
=> "selbstsperrend".

bei $U_{GS} > U_{th} > 0$
Stromfluß von D nach S
(U_{th} ... Schwellspannung)

Was passiert im FET?



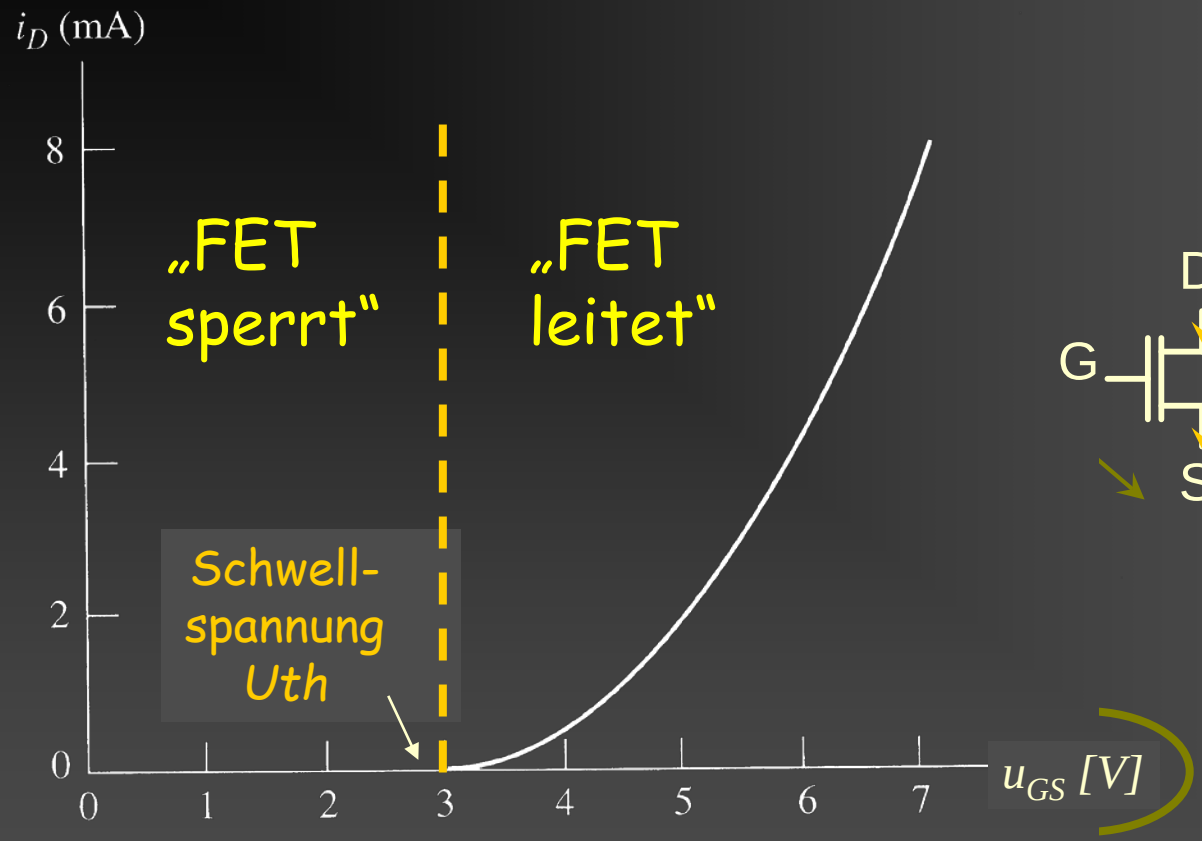
pn-Übergang: Gleichgewicht der Kräfte auf Elektron

- elektr. Kraft (zum Kern)
- Gitterkraft (zum Loch)

(thermodyn. Vorgänge, stark temperaturabh.!)

- U_{GS} bewirkt E-Feld (= zusätzl. elektr. Kraft auf Elektronen) und verschiebt dadurch Gleichgewicht.
- Bei $U_{GS} = U_{th}$ sind die Löcher im p-Si gefüllt; Elektronen können den Kanal zwischen D und S passieren.

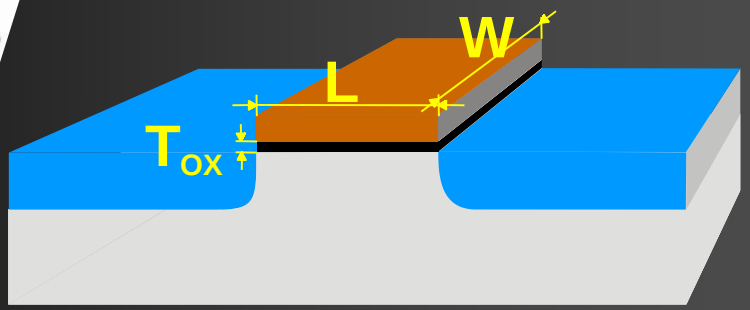
n-Kanal FET: Eingangskennlinie



Dimensionierung

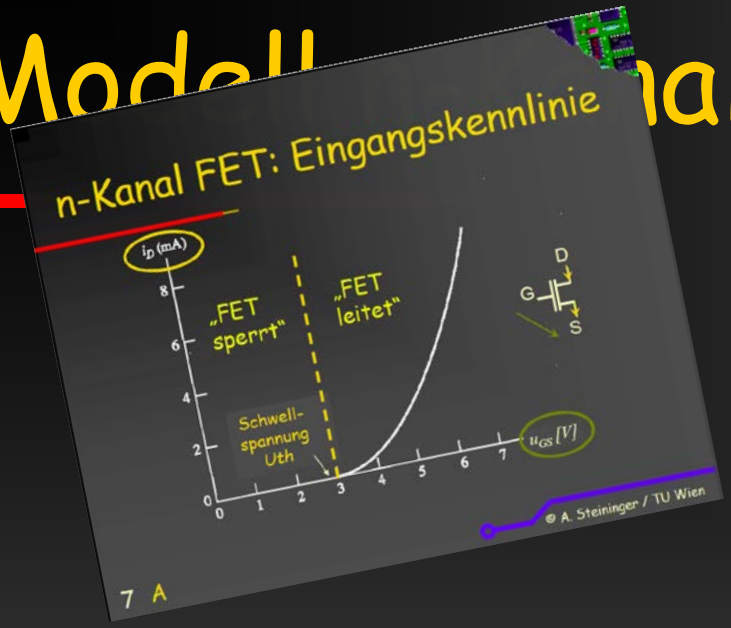
- Schwellspannung U_{th}
  - ✓ Grenzwert der Spannung zwischen „Schalter geschlossen" und „Schalter offen"
  - ➢ Einstellbar über Dotierung
- Ausgangsstrom I_{DSS}
  - ✓ Maximaler Strom, den der FET bei „Schalter geschlossen" führen kann
  - ➢ Einstellbar über Verhältnis von Kanallänge L zu Kanalbreite W: $I_{DSS} \propto W/L$ („Formfaktor")

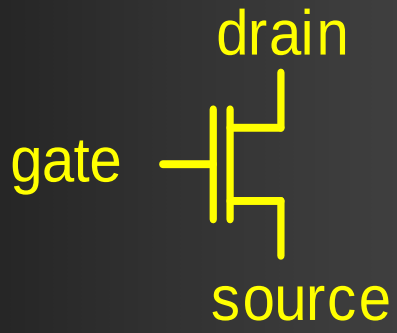
Formfaktor



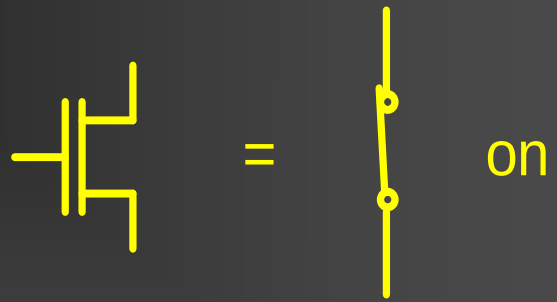
- Quotient aus Kanalbreite W und Kanallänge L
- bestimmt Sättigungsstrom des Schalters
- Erlaubt Einstellen der Treiberstärke
  - höhere Treiberstärke als X1 (X2, X4, X8) (X1 entspricht dem einfachen Inverter)
  - Angleich p-Kanal / n-Kanal (Mobilität d. Löcher schlechter => ca. Faktor 2)
  - Optimierung nach Performance / Fläche

Modell ...nal FET

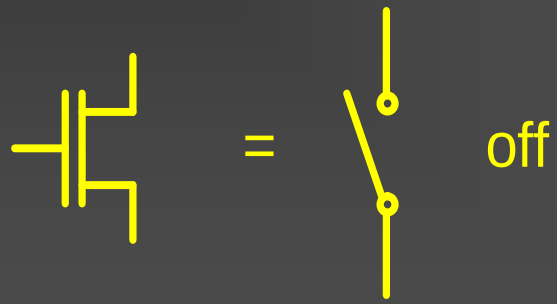




- bei logisch 1 ist der Schalter geschlossen



- bei logisch 0 ist der Schalter offen

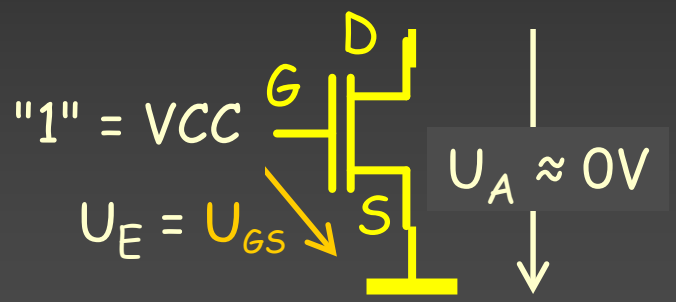


FET-Grundschaltung 1

„Sourceschaltung“

VCC

R



Gleichungssystem:

$U_A = VCC - R \cdot I_D$

$I_D = K\,[2(U_{GS}-U_{th})U_{DS}-U_{DS}^2]$

$U_{DS} = U_A$

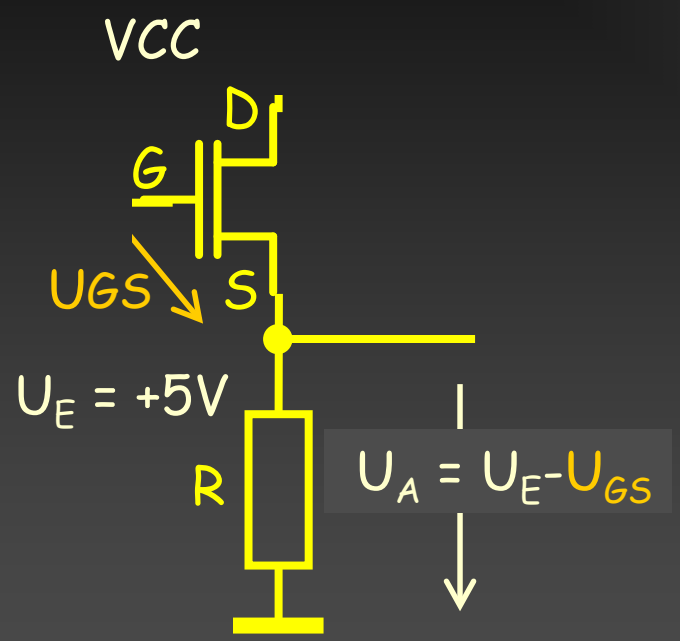
$U_{GS} = U_E$

Steuerspannung U_{GS} nur durch Eingangsspannung bestimmt

Spannung wird invertiert

FET-Grundschaltung 2

„Sourcefolger“



Gleichungssystem:

$U_A = R \cdot I_D$

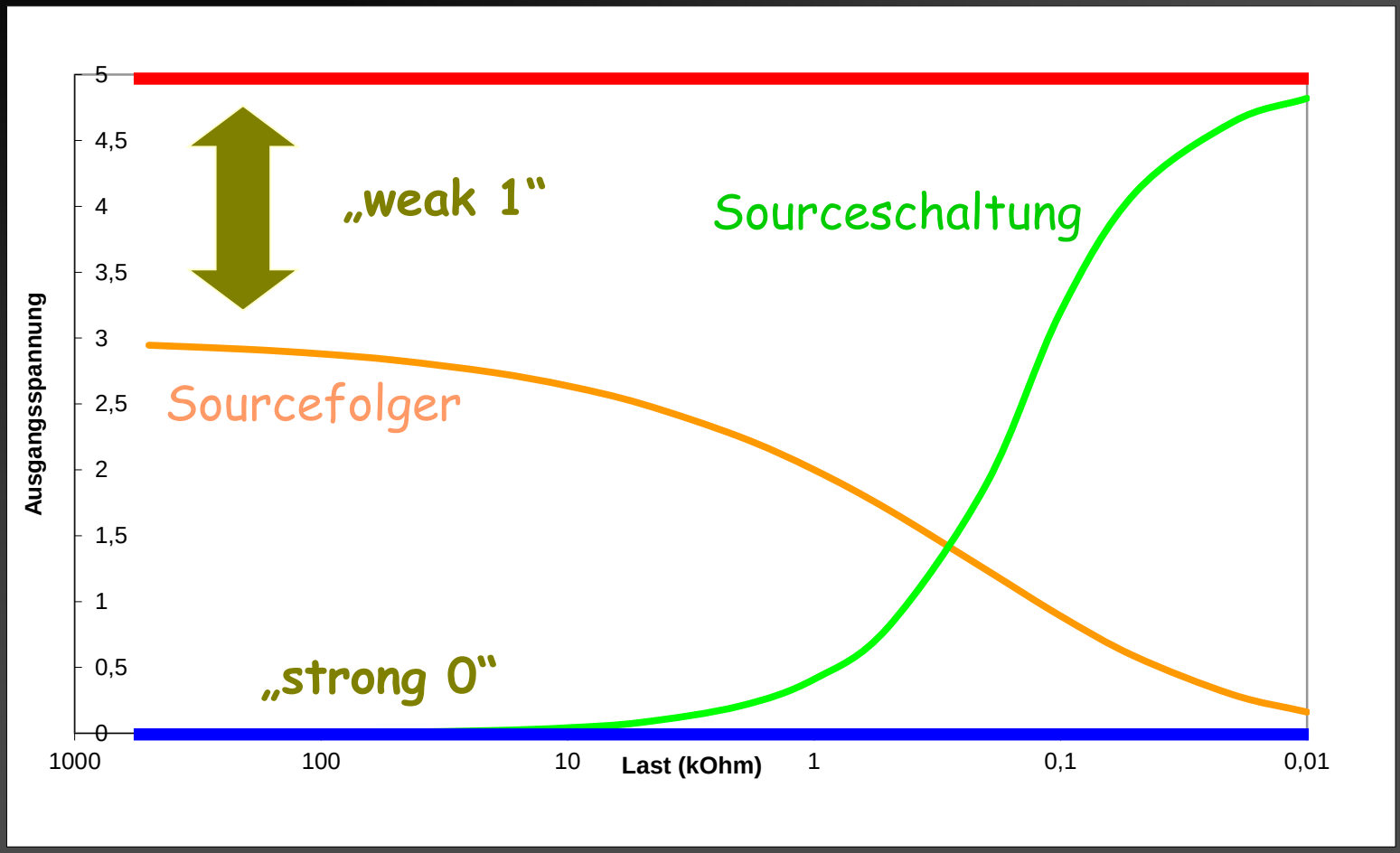
$I_D = K\,(U_{GS} - U_{th})^2$

$U_{GS} = U_E - U_A$

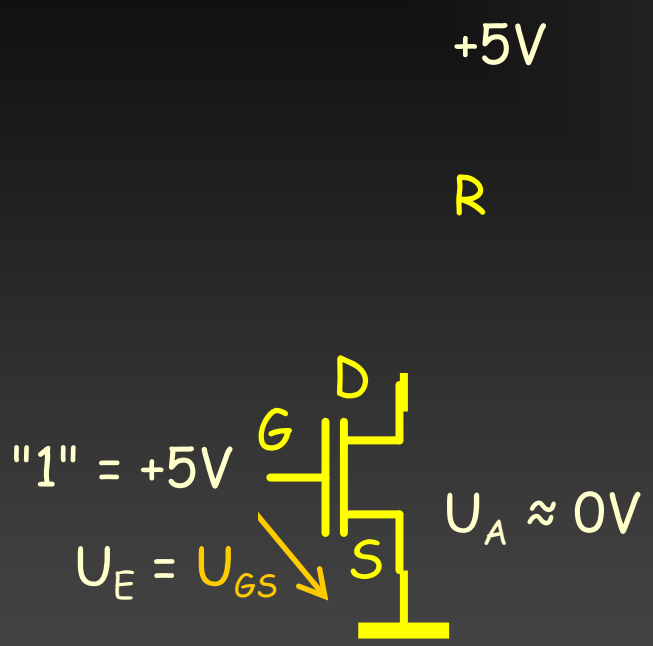
Ausgangsspannung U_A vermindert verfügbare Steuerspannung U_{GS} !

Ausgangsspannung U_A ist stets kleiner als Eingangsspannung

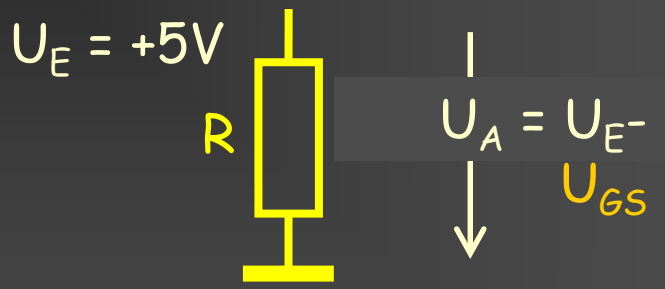
Vergleich der Schaltungen



Starke und schwache Pegel



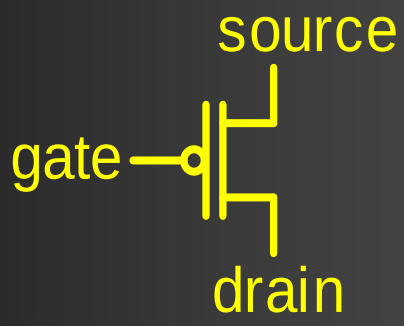
U_{GS} ist nur durch U_E bestimmt, unabh. von U_A



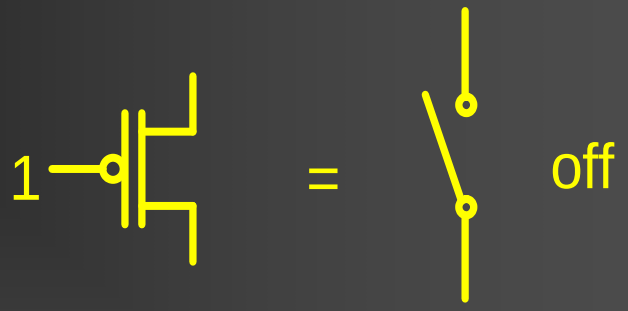
U_{GS} sinkt wenn U_A steigt => FET-Schalter öffnet! „Schalter" abh. v. Ausgang

Modell p-Kanal FET

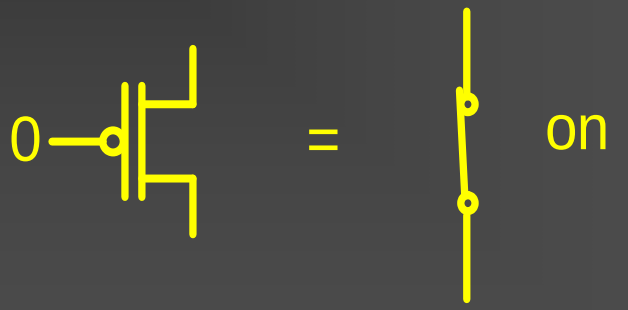
umgekehrt wie bei n-Kanal FET !



- bei logisch 1 ist der Schalter offen



- bei logisch 0 ist der Schalter geschlossen



Vorteil „komplementärer“ FETs

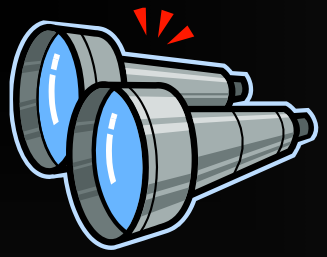
- n-Kanal FET kann
  - logisch „0“ aktiv treiben (strong 0),
  - logisch „1“ nur *sehr schwach* (weak 1)
- p-Kanal FET kann
  - logisch „1“ aktiv treiben (strong 1),
  - logisch „0“ nur *sehr schwach* (weak 0)
- Durch Kombination kann man *beide logischen Pegel aktiv* treiben

CMOS-Logik

- n-Kanal FET und p-Kanal-FET nennt man zueinander *„komplementär"*.
- es werden **MOS**FET-Transistoren verwendet (**M**etall/**O**xid/**S**emiconductor)

„**C**omplementary **MOS**FET" **CMOS**

- **CMOS treibt beide Logikpegel aktiv.**
- In CMOS lassen sich logische Funktionen besonders effizient implementieren.



Überblick

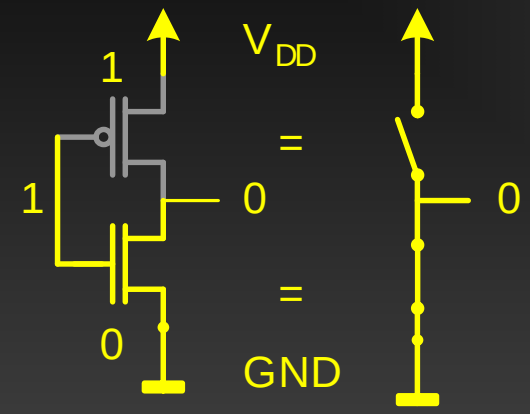
- Was ist CMOS ?
- Feldeffekt-Transistor & CMOS-Prozess
- **kombinatorische Logikzellen**
- sequenzielle Logikzellen
- weitere Logikfamilien

Der CMOS-Inverter: Funktion

„1" am Eingang:

p-FET offen

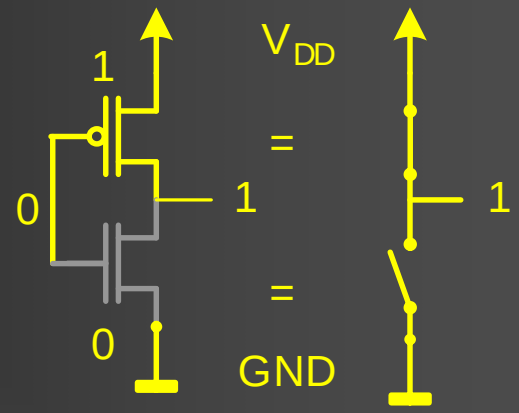
n-FET geschl.

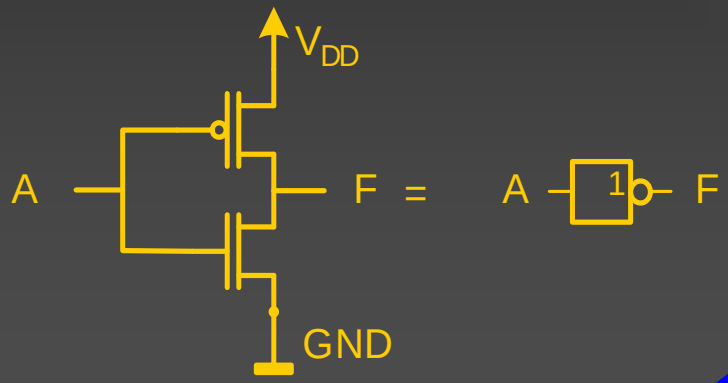


„0" am Eingang:

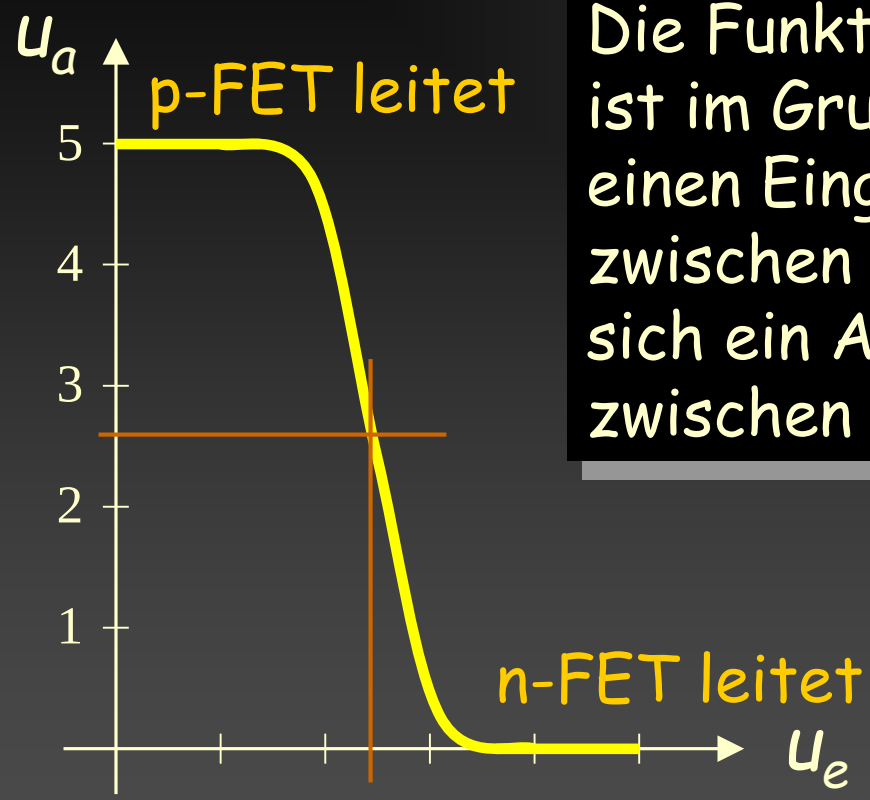
p-FET geschl.

n-FET offen



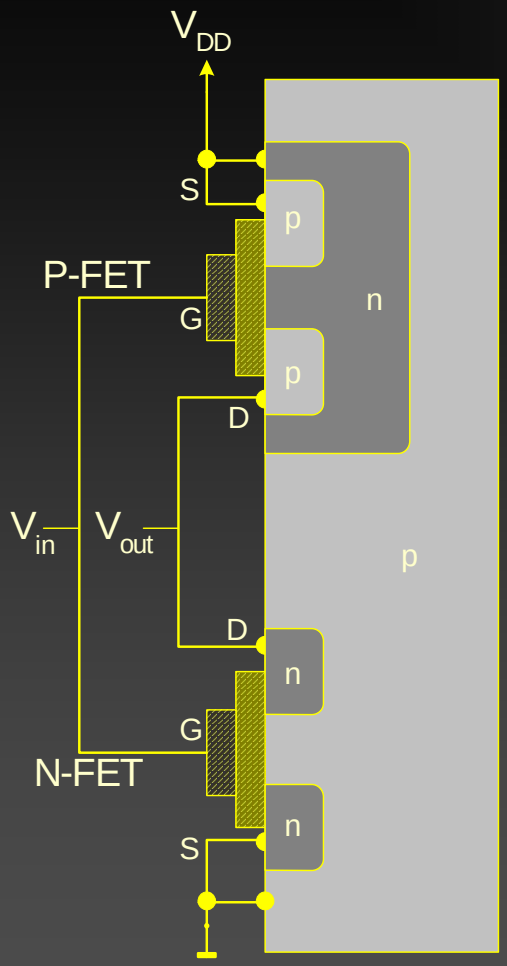


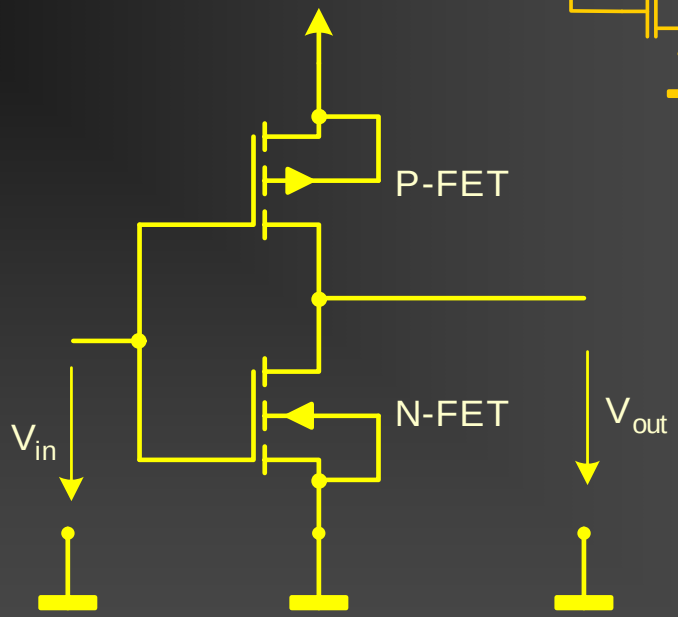
CMOS-Inverter: Kennlinie

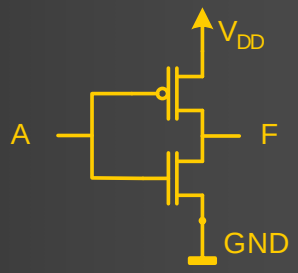


Die Funktion des Inverters ist im Grunde analog: Für einen Eingangspegel zwischen HI und LO kann sich ein Ausgangspegel zwischen LO und HI ergeben

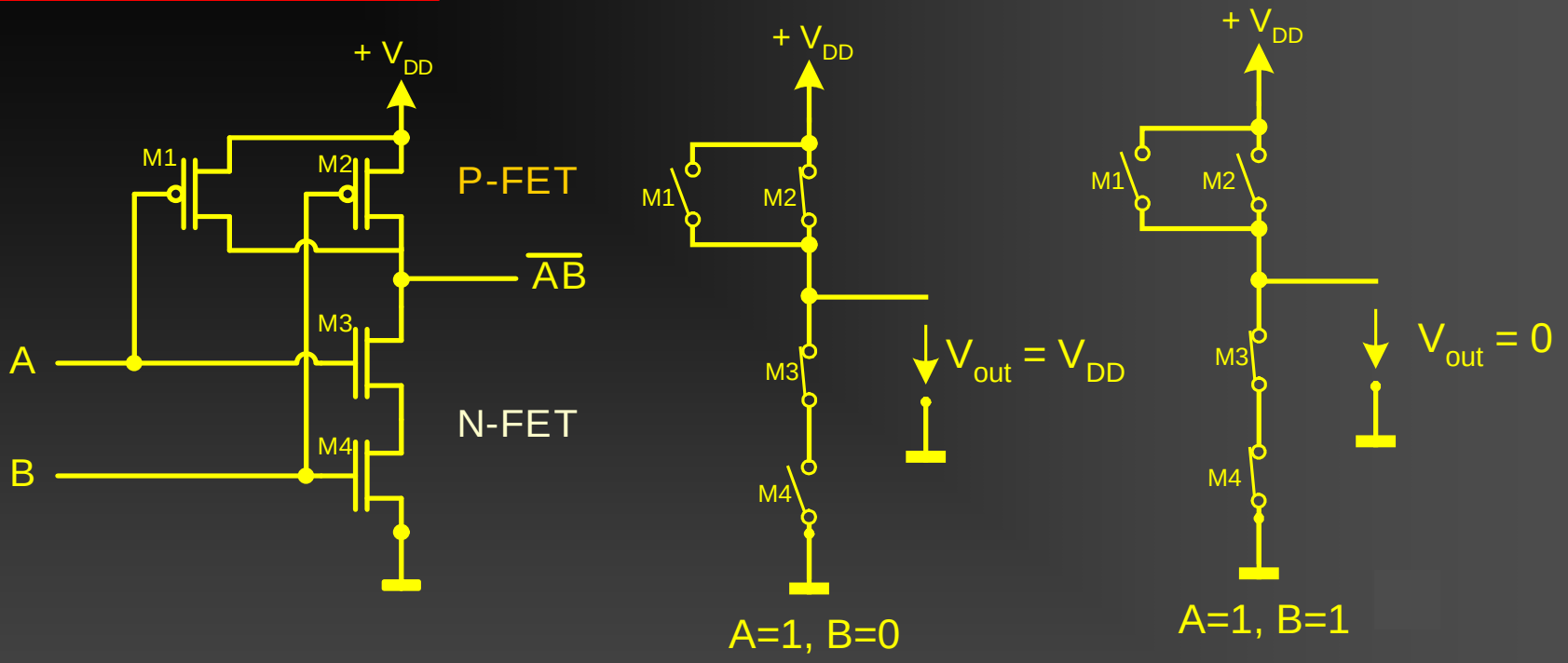
CMOS-Inverter: Technologie







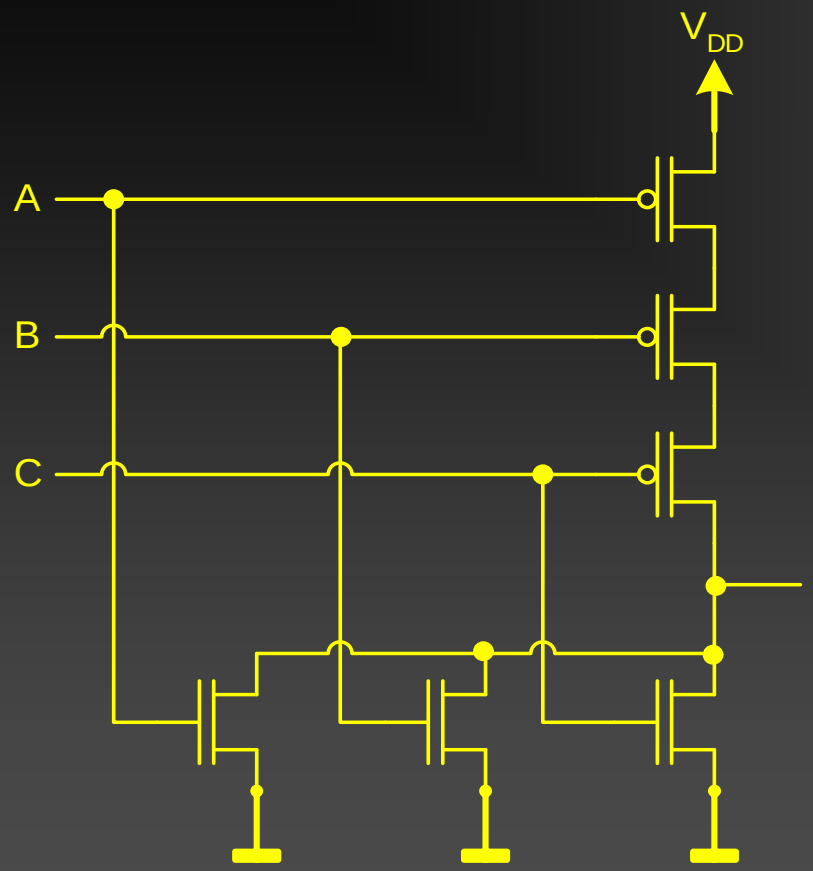
Aufbau eines CMOS-NAND2



p-FETs parallel ⇨ „0" an A oder B für Y = „1"

n-FETs in Serie ⇨ „1" an A und B für Y = „0"

Aufbau eines CMOS-NOR3



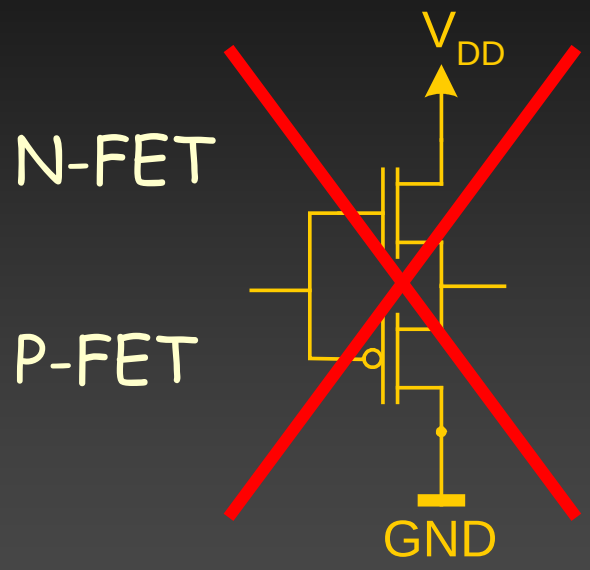
p-FETs in Serie
„0" an A, B und C
für Y = „1"

n-FETs parallel
„1" an A, B oder C
für Y = „0"

CMOS-Buffer

- Falsch

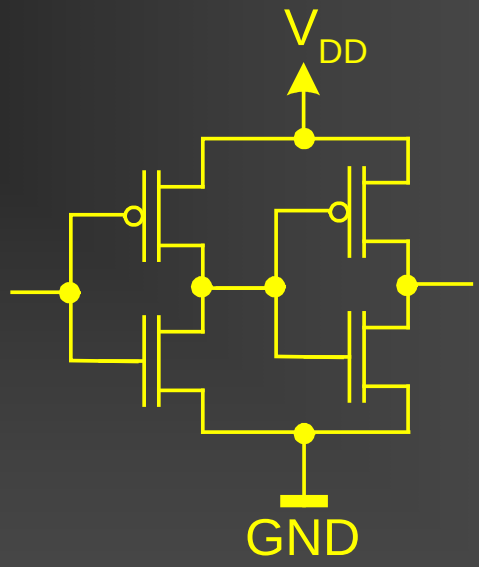
N & P-FET vertauscht



Nur schwache Pegel !

- Richtig

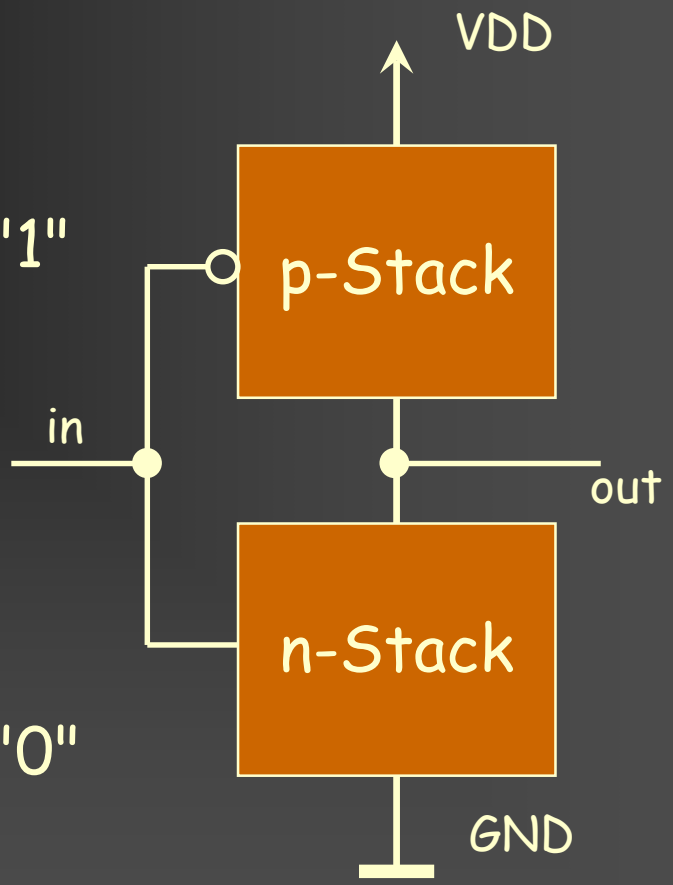
2 Inverter in Serie



2-stufige Schaltung !

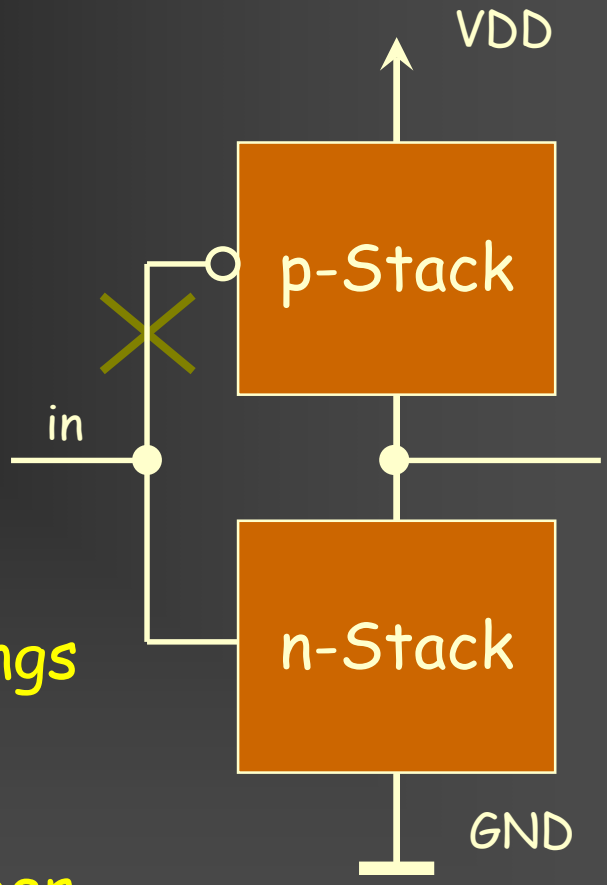
CMOS-Gatter: allg. Aufbau

- Der p-Stack
  - wird aus p-FETs gebildet
  - schaltet den Ausgang auf "1"
- Der n-Stack
  - wird aus n-FETs gebildet
  - schaltet den Ausgang auf "0"



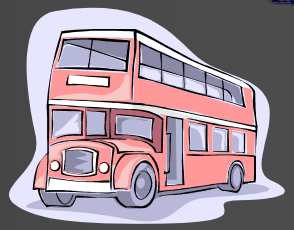
Tri-State-Ausgang

p-Stk	n-Stk	Ausgang
ein	aus	1
aus	ein	0
aus	aus	Tri-state
ein	ein	Kurzschluß



erlaubt Abschalten des Ausgangs über einen Steuereingang „output enable (OE)".
Vorteil: erlaubt mehrere Treiber an einem Bus

Tri-State Bus: Probleme



- Bus-Contention:

  auf einer Leitung ist zu einem Zeitpunkt mehr als ein Treiber aktiv => hohe Ströme, Pegel undefiniert

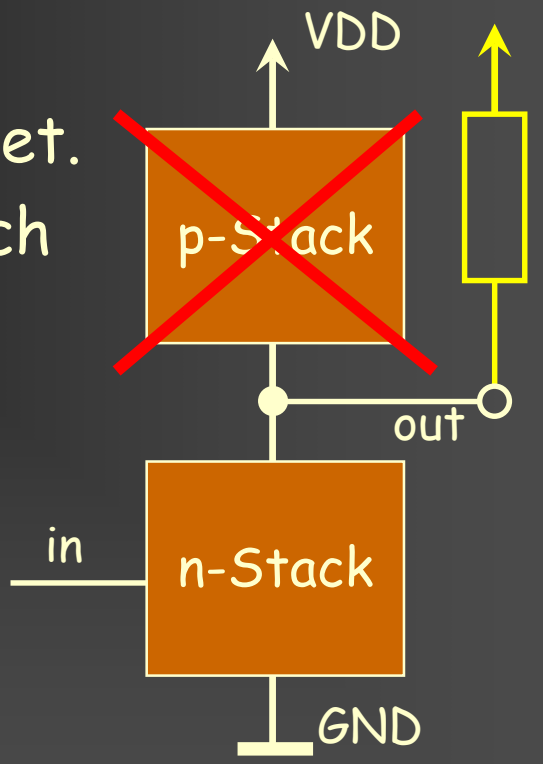
- Floating Bus:

  auf einer Leitung ist kein Treiber aktiv
  => Pegel undefiniert

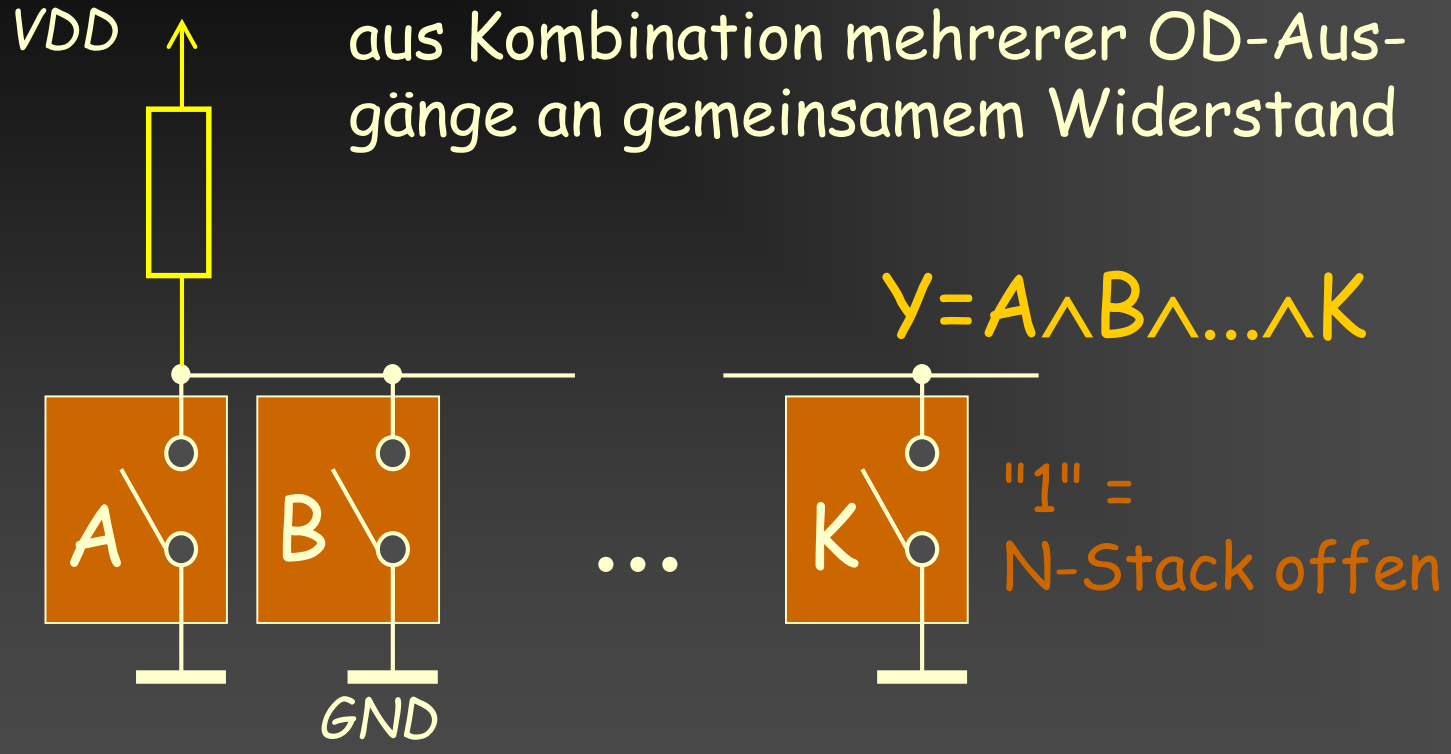
- Bus-Keeper (bus friendly Logic): FF aus antiparallelen Invertern hält den letzten Zustand, kann aber leicht „overruled" werden (schwache Treiberstärke)

Open-Drain Ausgang (OD)

- Der (aktive) p-Stack wird weggelassen. An seiner Stelle wird extern ein Widerstand verwendet.
- Ausgang "0" wird weiterhin durch den n-Stack erzwungen. Es sind auch größere Ströme zulässig.
- Ausgang "1" wird bei offenem n-Stack durch den Widerstand (schwach) hergestellt: Bei größeren Strömen bricht die Spannung ein.



Prinzip des „Wired AND"



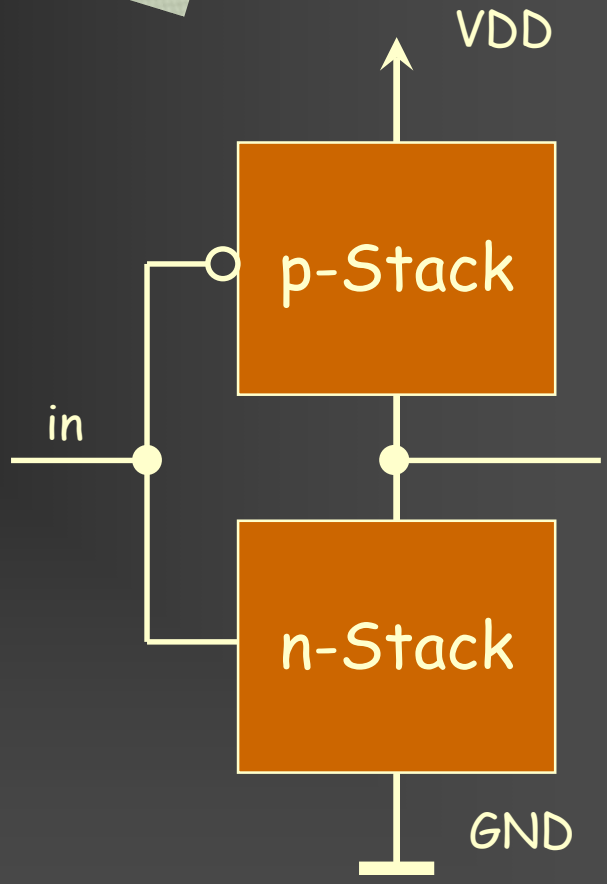
aus Kombination mehrerer OD-Ausgänge an gemeinsamem Widerstand

$Y = A \wedge B \wedge \ldots \wedge K$

"1" = N-Stack offen

Aktiver Ausgang

p-Stk	n-Stk	Ausgang
ein	aus	1
aus	ein	0
aus	aus	Tri-state
ein	ein	Kurzschluß



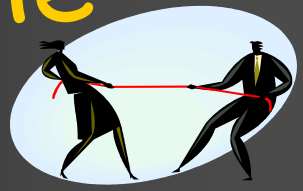
n-Stack: Aufbau

- Ein AND-Term wird durch Serienschaltung von FETs bzw. Blöcken realisiert, ein OR-Term durch Parallelschaltung.
- Durch geeignete Kombination lassen sich beliebige Boolsche Verknüpfungen realisieren, allerdings mit folgenden Einschränkungen:
  - Da der n-Stack genau dann durchschalten soll, wenn die Zielfunktion "0" ist, läßt sich nur eine AND/OR Verknüpfung mit Inversion am Schluß realisieren.
  - Da die n-FETs jeweils bei "1" am Eingang durchschalten, kann man also nicht mit invertierten Eingängen arbeiten.

p-Stack: Aufbau

- Ein AND-Term wird auch hier wieder durch Serienschaltung von FETs bzw. Blöcken realisiert, ein OR-Term durch Parallelschaltung.
- Für die Zielfunktion gelten folgende Einschränkungen:
  - Da der p-Stack genau dann durchschalten soll, wenn die Zielfunktion "1" ist, darf die Zielfunktion keine Inversion am Schluß haben.
  - Da die p-FETs jeweils bei "0" am Eingang durchschalten, kann man also nur mit invertierten Eingängen arbeiten.

Lösung der Widersprüche



- n-Stack
  - Inversion am Ende
  - nicht-invertierte Eingänge

De Morgan

$$\neg F(X1, X2, X3, \ldots Xn, \wedge, \vee)$$
$$=$$
$$F(\neg X1, \neg X2, \neg X3, \ldots \neg Xn, \vee, \wedge)$$

- p-Stack
  - keine Inversion am Ende
  - nur invertierte Eingänge

Entwurfsregeln im Überblick

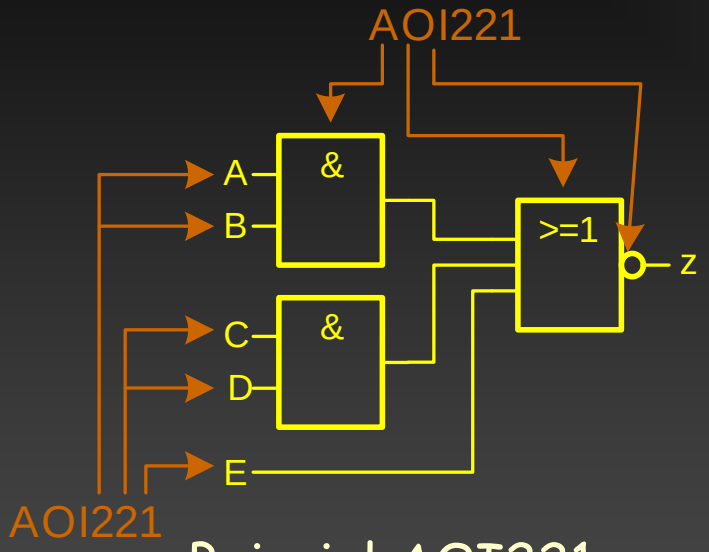
- Durch Kombination aus Serien- und Parallelschaltung lassen sich auch komplexere Funktionen als NAND und NOR realisieren:

  AND-OR-Invert bzw. OR-AND-Invert,

- In jedem Fall mit Inversion am Ausgang (wenn nötig extra Inverter nachschalten).
- In jedem Fall nicht invertierte Eingänge (wenn nötig extra Inverter vorschalten).
- In jedem Fall p-Stack oben und n-Stack unten.
- In jedem Fall p-Stack dual zu n-Stack.

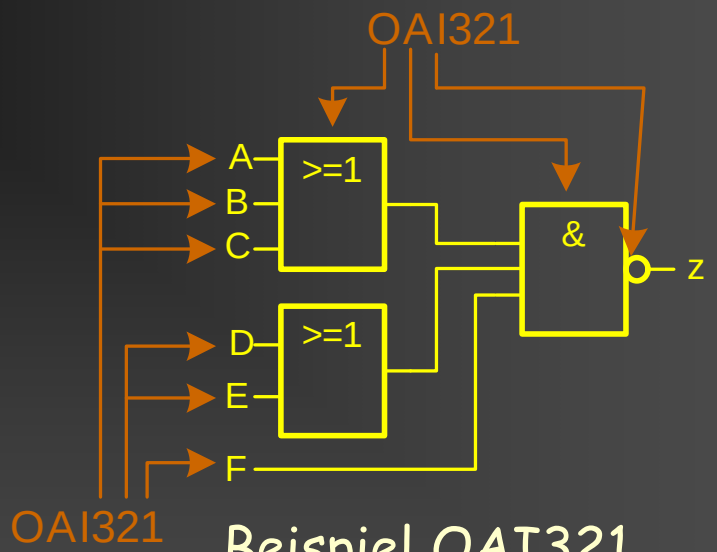
AOI und OAI: Terminologie

AND-OR-Invert



Beispiel AOI221

OR-AND-Invert



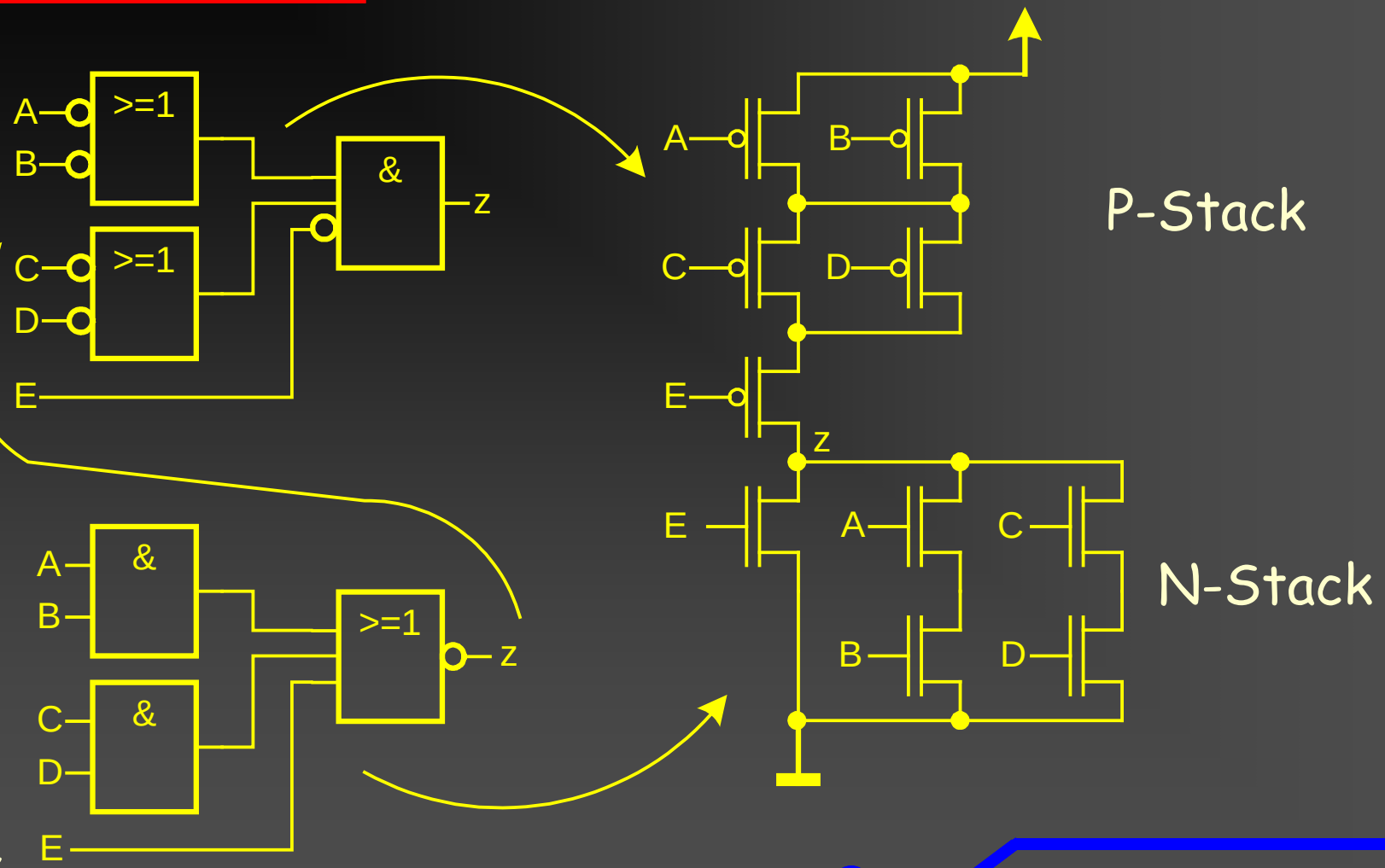
Beispiel OAI321

AOI- und OAI-Zellen sind sehr effizient durch Serien/Parallelschaltung von FETs realisierbar

Entwurfsregeln für AOI & OAI

1. Gleichung G entsprechend der Funktion aufstellen (Inversion am Schluß!)
   - ➡ n-Stack (strong „0“)
     Inversion am Ausgang erfolgt automatisch
2. Inversion zu Eingängen transformieren: (De Morgan) => Gleichung G*
   - ➡ p-Stack (strong „1“)
     Inversion an d. Eingängen erfolgt automatisch wegen p-Kanal-FET
3. In beiden Fällen gilt: AND = Serienschaltung
   OR = Parallelschaltung

Entwurfsbeispiel AOI221



Rechenbeispiel

- Alarmanlage:
  - 1 Innenkreis mit 1 Bewegungsmelder B, aktivierbar über Schalter S1
  - 1 Außenkreis mit 2 Türkontakten K1 und K2, aktivierbar über Schalter S2
  - Alle Schalter und Kontakte low-aktiv
  - Auslösung d. Sirene über Signal AL (high-aktiv)
- Gesucht: Realisierung als AOI oder OAI

Umformungen

$AL = (\neg S1 \wedge \neg B) \vee (\neg S2 \wedge (\neg K1 \vee \neg K2))$

AOI:

$AL = (\neg S1 \wedge \neg B) \vee (\neg S2 \wedge \neg K1) \vee (\neg S2 \wedge \neg K2)$

- nicht invertierend => Inverter am Ausg.
- invertierte Eingänge => Inverter an allen Eing.

OAI:

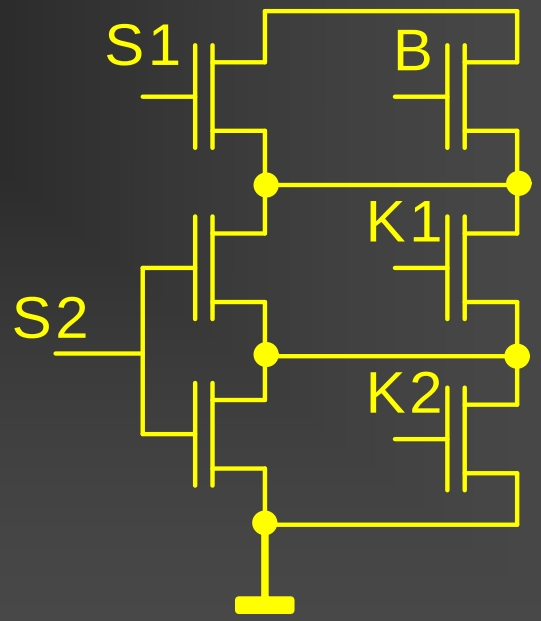
$\neg AL = (S1 \vee B) \wedge (S2 \vee K1) \wedge (S2 \vee K2)$

- Inversionen bereits an den richtigen Stellen
  => viel günstiger zu realisieren

Alarmanlage als OAI: n-Stack

$\neg AL = (S1 \vee B) \wedge (S2 \vee K1) \wedge (S2 \vee K2)$

- n-Stack:
  - S1 parallel B
  - S2 parallel K1
  - S2 parallel K2
  - alle Parallel-elemente in Serie

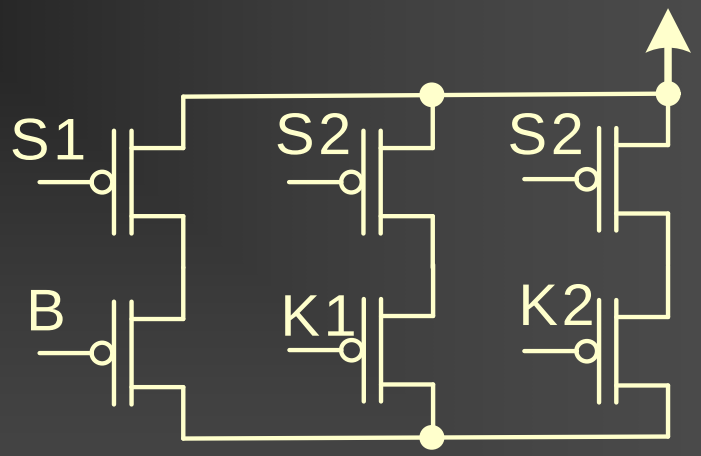


Alarmanlage als OAI: p-Stack

$$\neg AL = (S1 \vee B) \wedge (S2 \vee K1) \wedge (S2 \vee K2)$$

$$AL = (\neg S1 \wedge \neg B) \vee (\neg S2 \wedge \neg K1) \vee (\neg S2 \wedge \neg K2)$$

- p-Stack:
  - S1 in Serie mit B
  - S2 in Serie mit K1
  - S2 in Serie mit K2
  - Alle Serien-elemente parallel



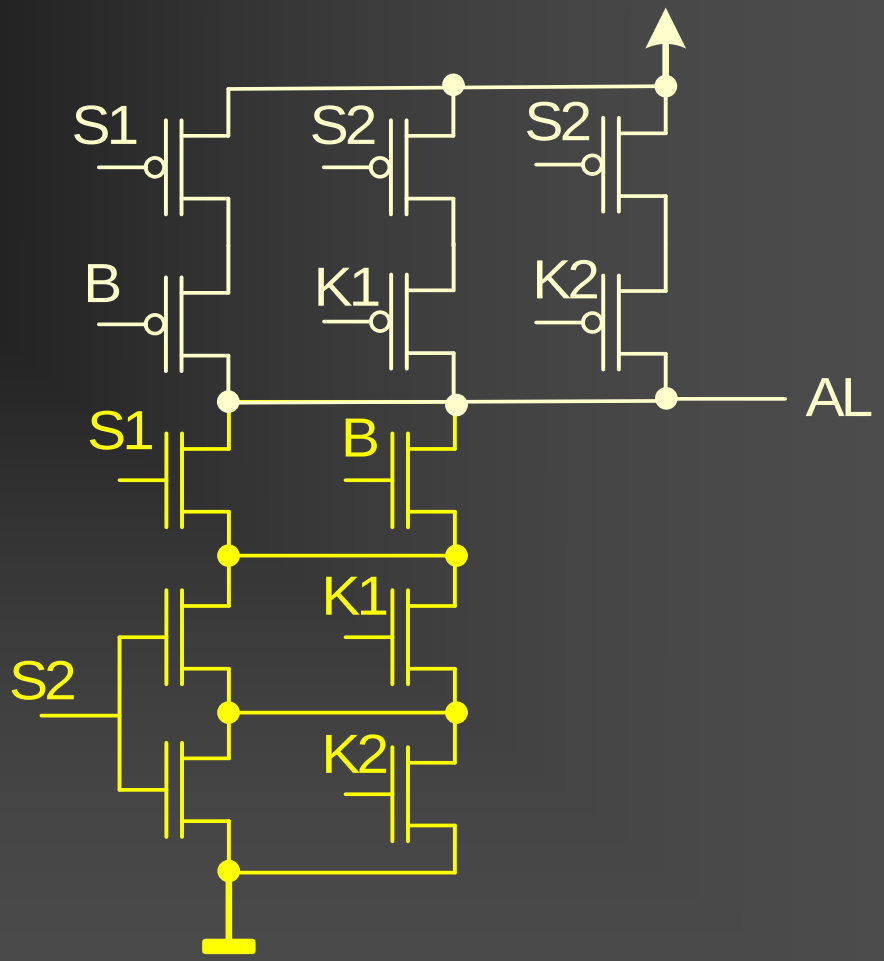
Alarmanlage als OAI: Lösung

- p-Stack:

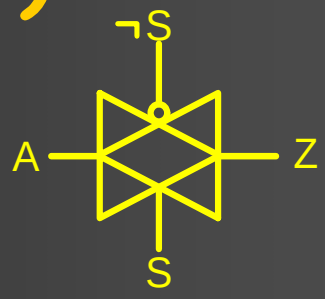
S1 + B, S2 + K1, S2 + K2, alle parallel

- n-Stack:

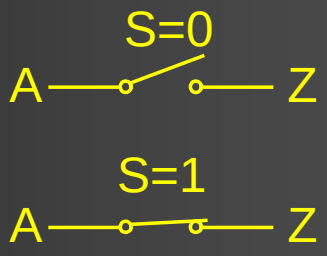
S1 par B, S2 par K1, S2 par K2, alle in Serie



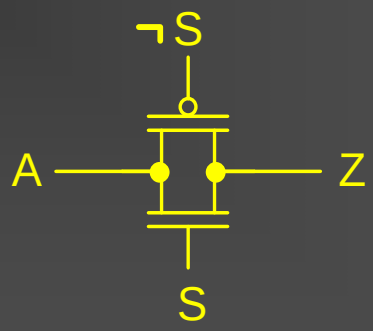
Transmission-Gate (TG)



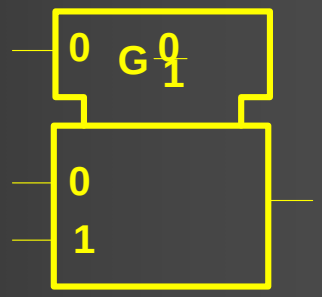
- **Funktion:**
  Schaltbare Verbindung zwischen zwei Leitungen („offen" = echte Trennung, keine Maskierung)



- **Realisierung:**
  n-Kanal FET und p-Kanal FET parallel
  (strong ´1´ und strong ´0´!)



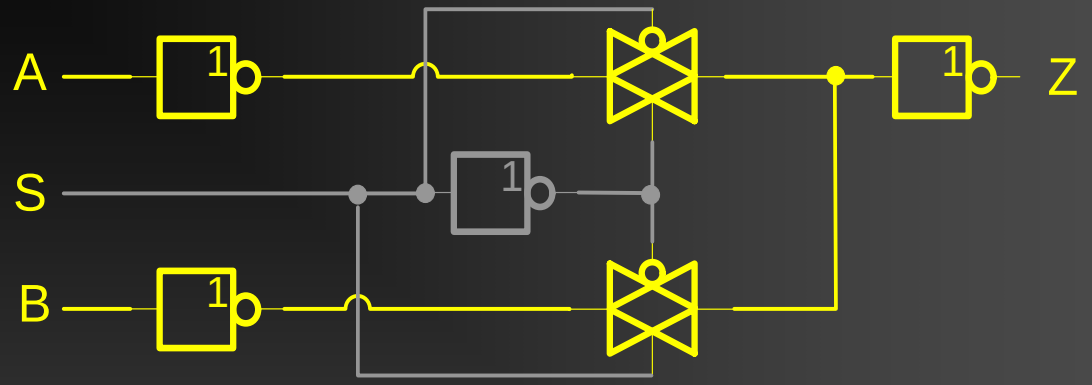
Multiplexer (Mux)



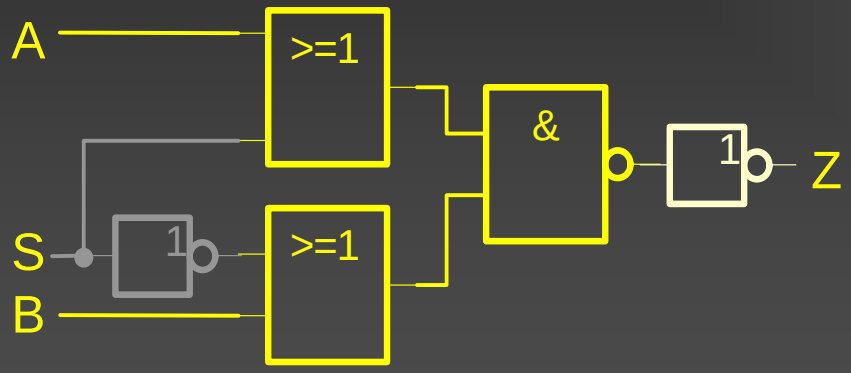
- Funktion:
  Von mehreren (*n*) Eingangssignalen wird über einen Steuereingang eines ausgewählt und an den Ausgang durchgeschaltet.
- Realisierung:
  - eines von *n* Transmission Gates wird aktiviert
  - Kombinatorische Verknüpfung:
    2:1 Mux als OAI22 + Inverter
    4:1 Mux als OAI3333 + Inverter

Multiplexer-Realisierungen

- TG
  (3GE)



- OAI
  (3GE)



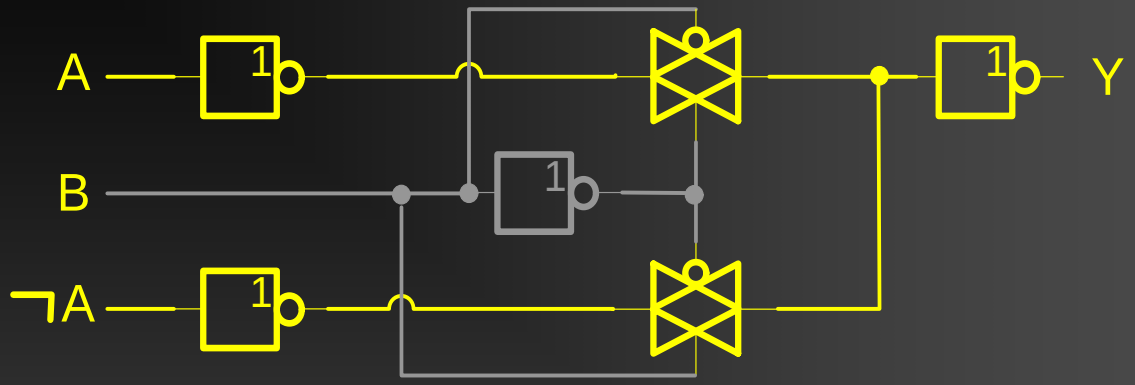
Exklusiv-ODER (XOR)

- Funktion:
  logische Verknüpfung $Y = (A \wedge \neg B) \vee (\neg A \wedge B)$

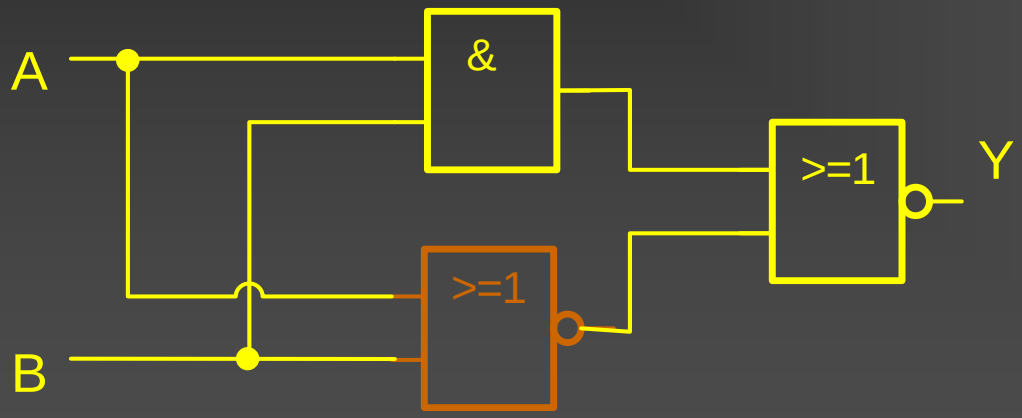
- Realisierung:
  - Multiplexer:
    B am Select-Eingang wählt zwischen A und $\neg A$
  - Kombinatorische Verknüpfung: AOI21 + NOR
    $\neg Y = (A \wedge B) \vee (\neg(A \vee B))$

XOR-Realisierungen

- TG
  (3GE)



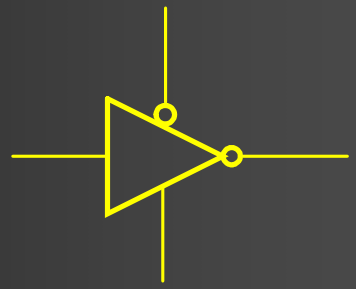
- AOI
  (2.5GE)



Getakteter Inverter

- Funktion:

  Wie Transmission Gate, aber
  - Signal wird invertiert
  - Takt als Steuersignal (S)

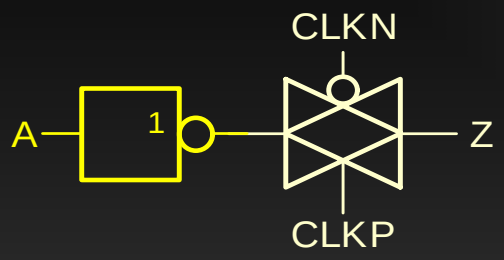


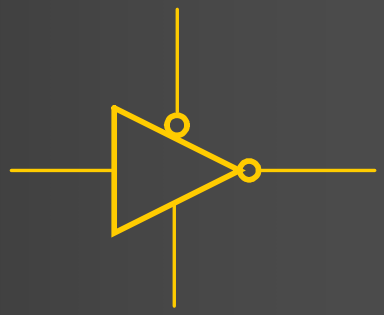
- Realisierung:

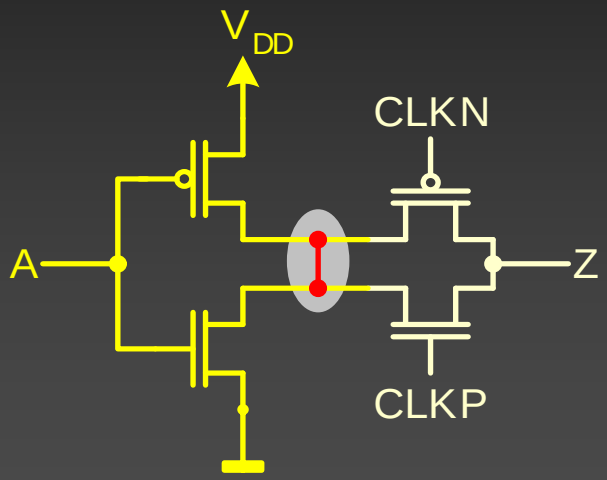
  Serienschaltung Inverter + Transmission Gate
  Dabei läßt sich eine Verbindung einsparen (siehe nächste Folie)

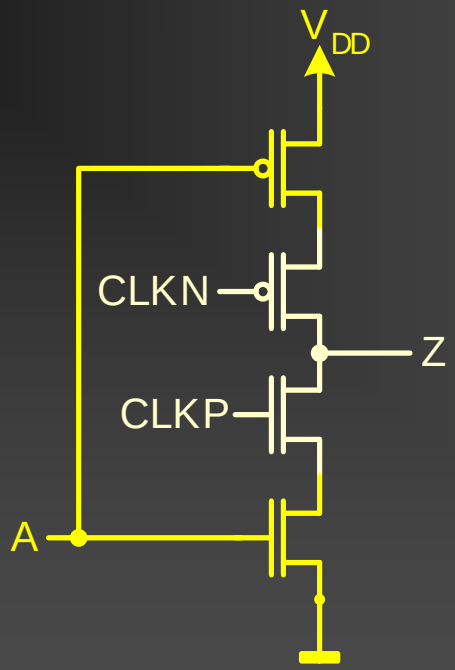
- Anwendung: bei Latch und Flip-Flop

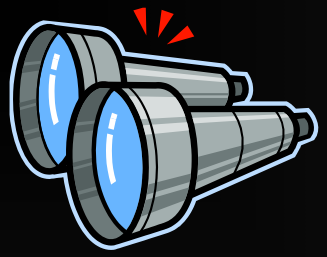
Getakteter Inv.: Realisierung







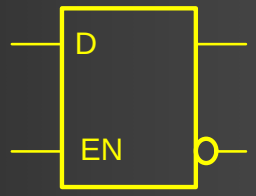




Überblick

- Was ist CMOS ?
- Feldeffekt-Transistor & CMOS-Prozess
- kombinatorische Logikzellen
- **sequenzielle Logikzellen**
- weitere Logikfamilien

Latch



- **Funktion:** *(positive enable)*

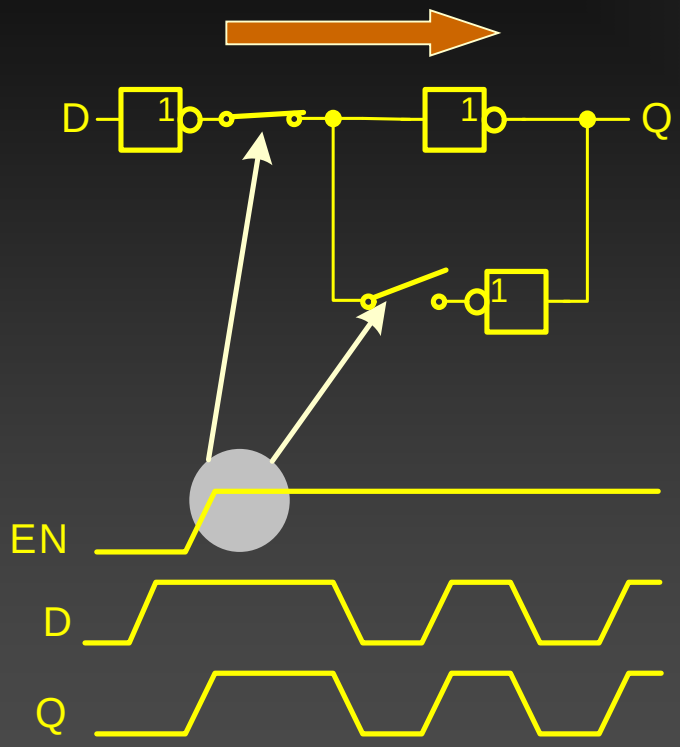
  Eingänge D (Data) und EN (Enable), Ausgang Q
  - Transparent: D wird direkt auf Q abgebildet
  - Hold: letzter Zustand von Q wird eingefroren

- **Realisierung:**

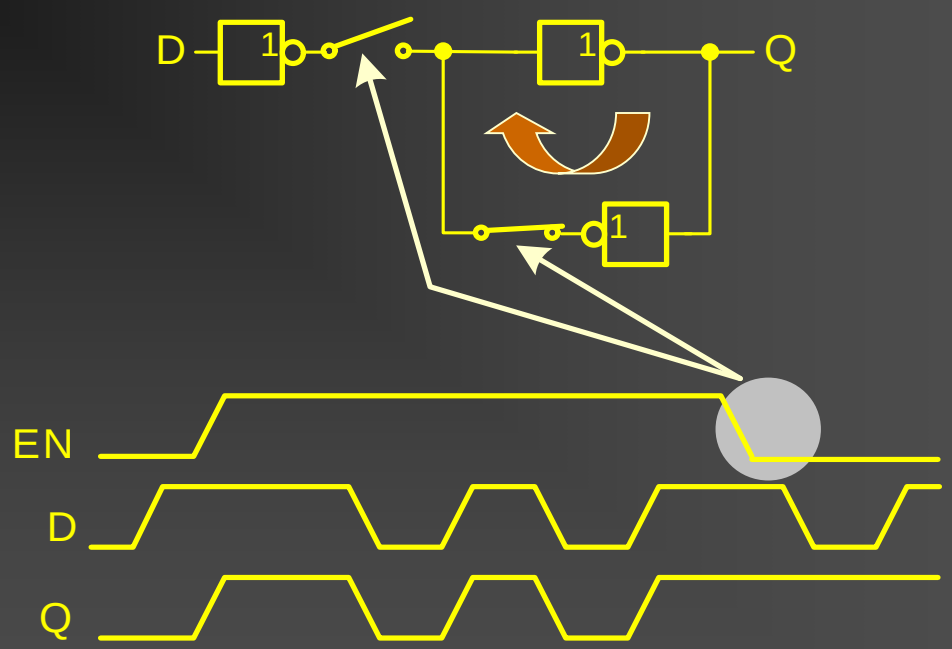
  Ausgang Q wird entweder von D angesteuert (transparent) oder von sich selbst (Rückkopplung). Umschaltung mittels Multiplexer aus zwei TGs

Funktionsmodell eines Latch

transparent



hold



Wiederholung

Grenzen der Geschwindigkeit

- Wellenausbreitung
  - Information kann sich niemals schneller als mit Lichtgeschwindigkeit ausbreiten. (ca. 20cm/ns)
- Ladevorgänge
  - Das Laden von Kapazitäten mit begrenztem Strom beansprucht Zeit. ($\tau = RC$)
- Bewegung der Ladungsträger
  - Bewegung/Diffusion von Ladungsträgern im Halbleiter erfolgt nur mit begrenzter Geschwindigkeit. (Sättigungswert bei Si typ. 0,1 mm/ns)

Setup- und Hold-Time

- Eine Änderung am Eingang muss die Rückkopplungsschleife vollständig durchlaufen

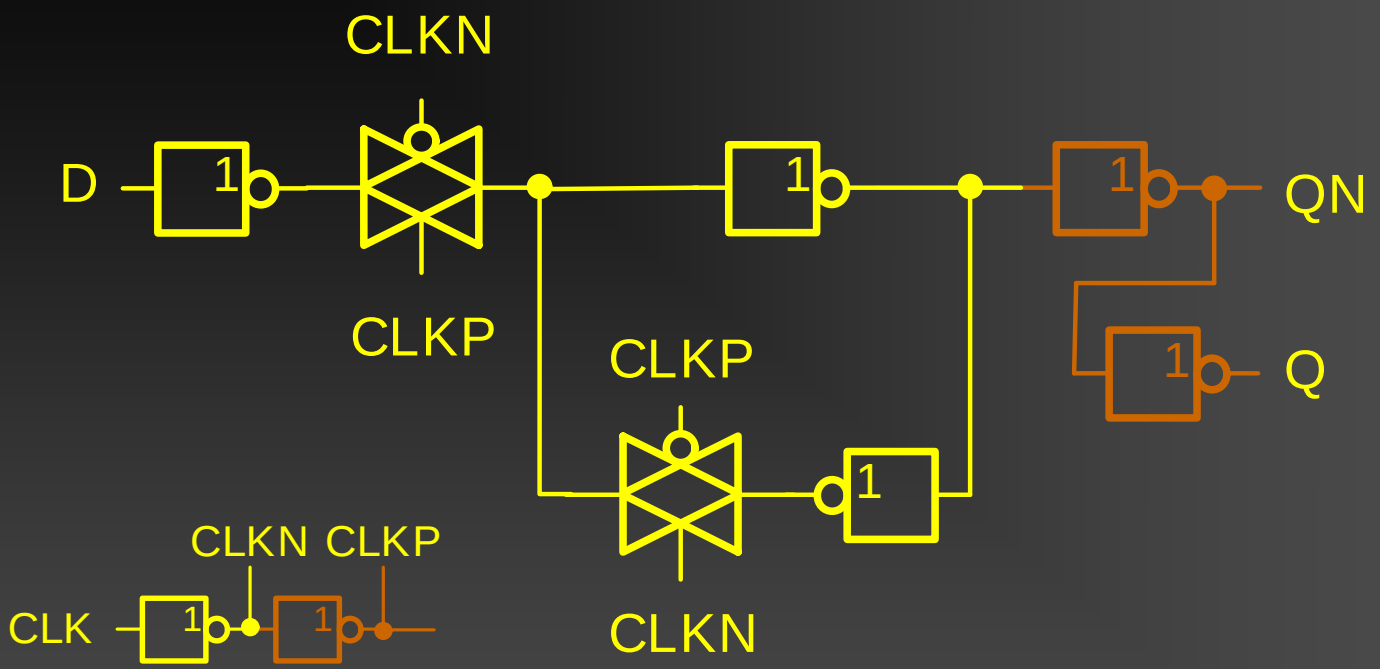
  UND

- Die TGs müssen umgeschalten werden

Diese Vorgänge brauchen **Zeit:**
„Decision Window" (= SetupTime + Hold-Time)

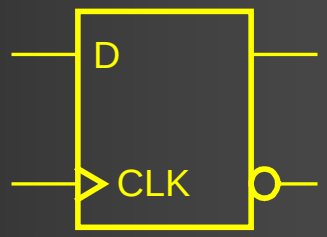
Innerhalb dieses „Decision Window" dürfen keine Flanken am Eingang auftreten (Metastabilität!)

Realisierung eines Latch



Aufwand: 7 Inv. + 2 TGs = 18 Trans = 4.5 GE

D-Flip-Flop



- Funktion:

  Eingänge D (Data) und CLK (Clock), Ausgang Q

  Der Zustand von D wird jeweils mit der **aktiven Flanke** auf Q übernommen und eingefroren.

- Realisierung:

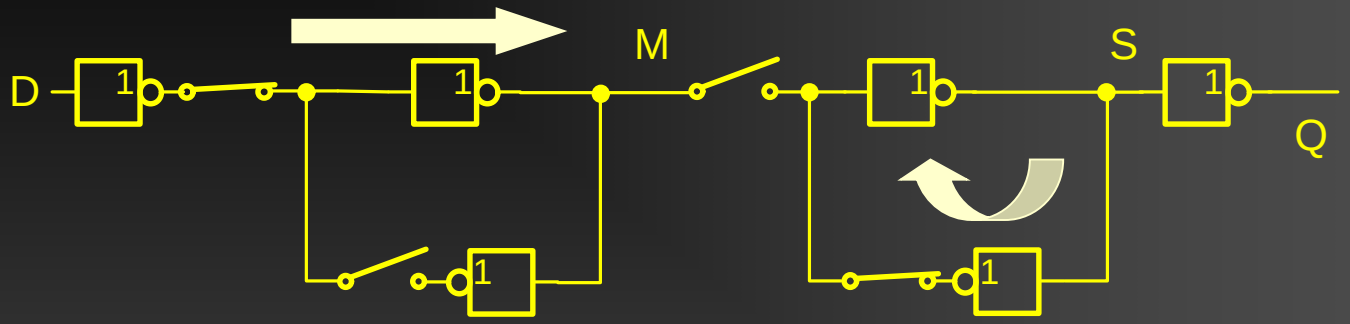
  zwei Latches in Master/Slave-Schaltung

Funktionsmodell eines Flip-Flop

CLK = 0

load master

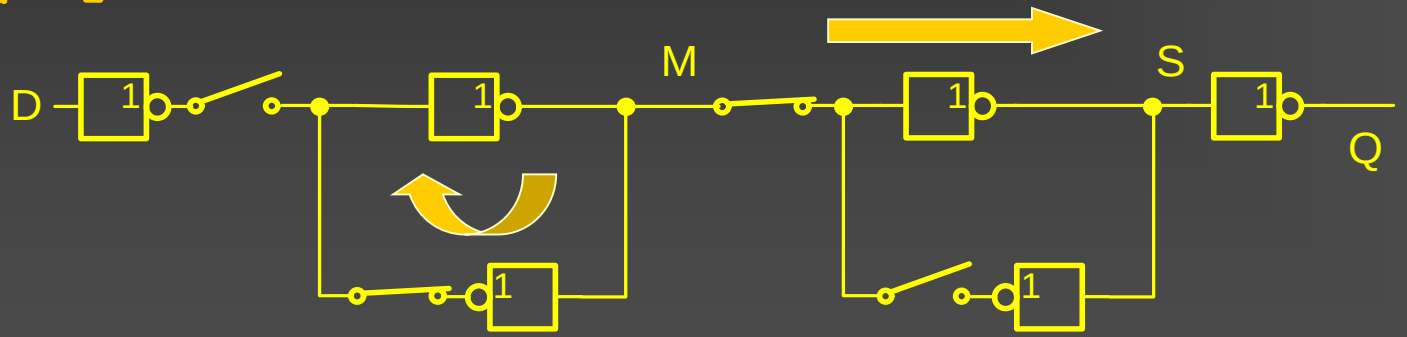
hold slave



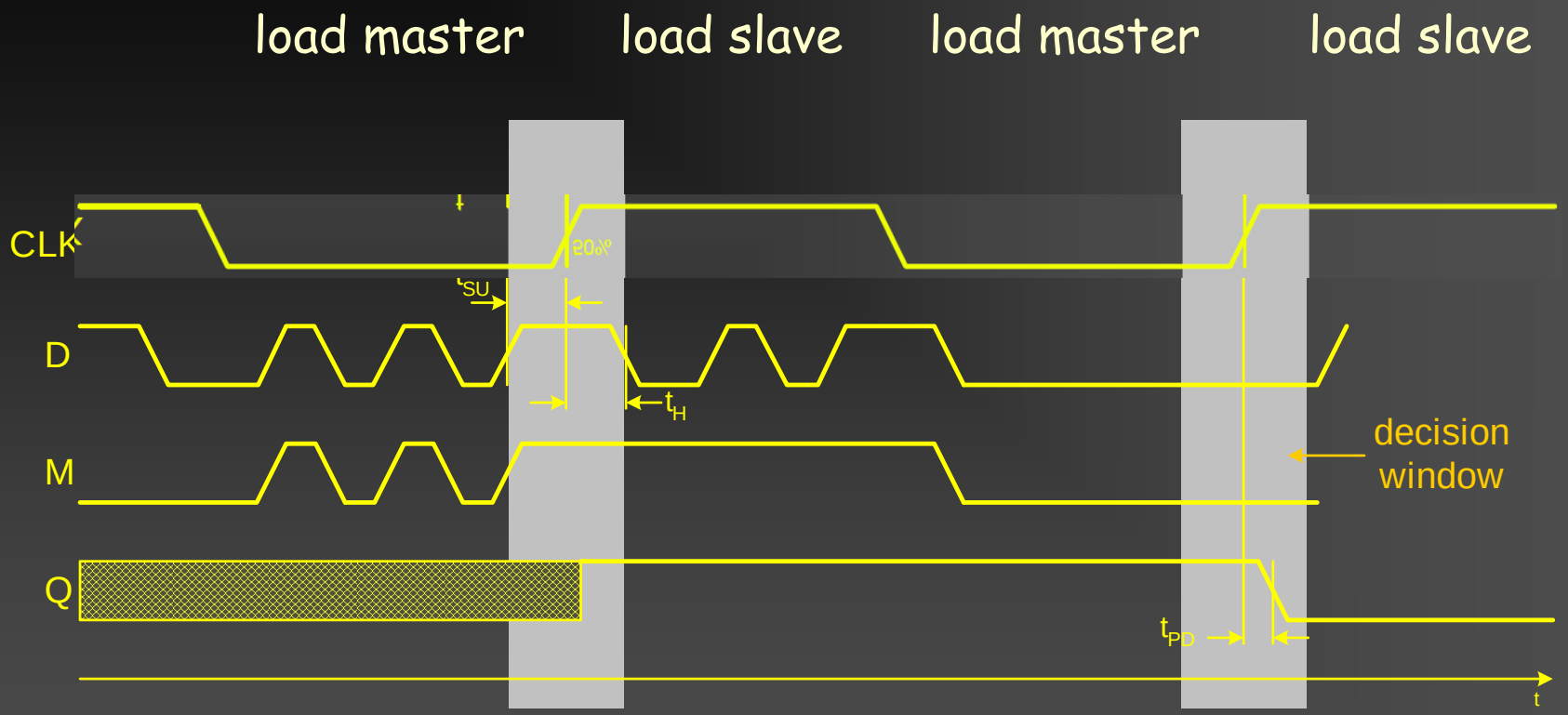
CLK = 1

hold master

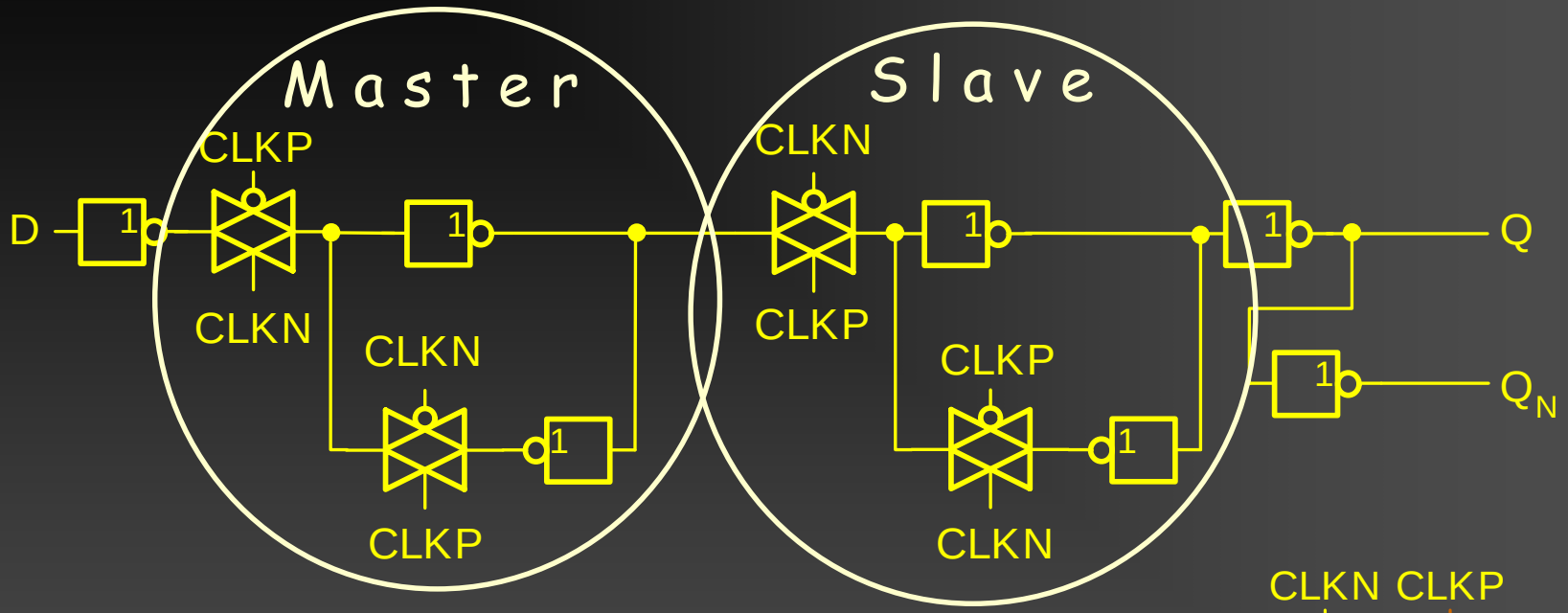
load slave

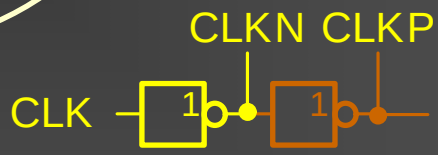


Decision Window beim Flip-Flop



Realisierung eines Flip-Flop





- Je 1 Latch für Master und Slave
- Taktversorgung gemeinsam
- Eingangs- und Ausgangsbuffer gemeinsam

Flip-Flop: Schaltungsaufwand

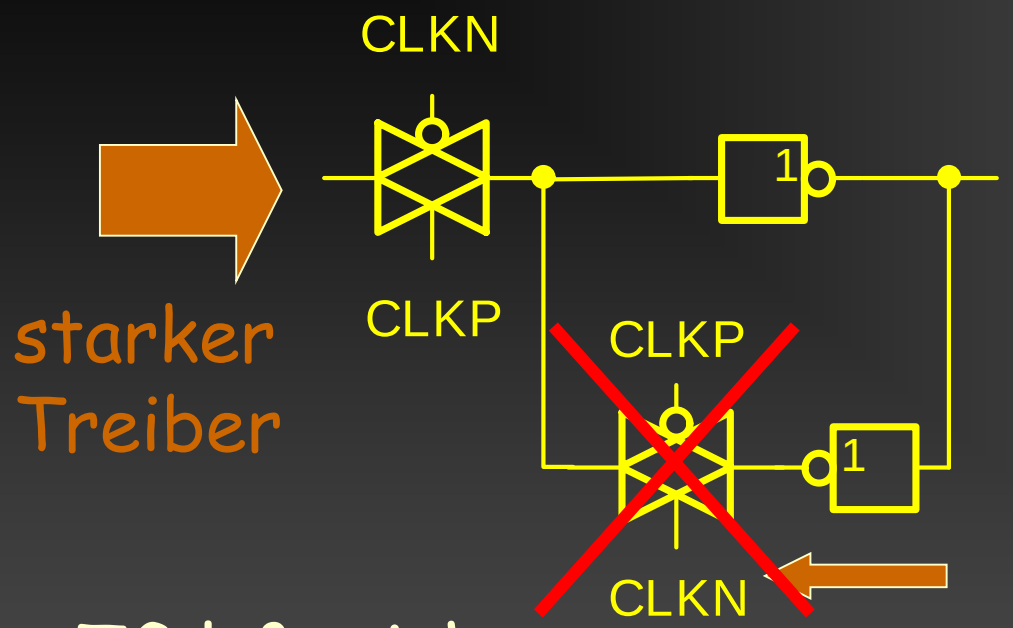
2 Latches entsprechen 36 Trans. = 9 GE, aber durch folgende Einsparungen

- Taktversorgung nur einmal (2 Inv.)
- Bufferung am Ausgang nur einmal (2 Inv.)
- Buffer am D-Eingang des Slave entfällt (1 Inv.)

ergibt sich ein Aufwand von

9 Inv. + 4 TGs = 26 Trans. = 6.5 GE

Weiteres Einsparpotential

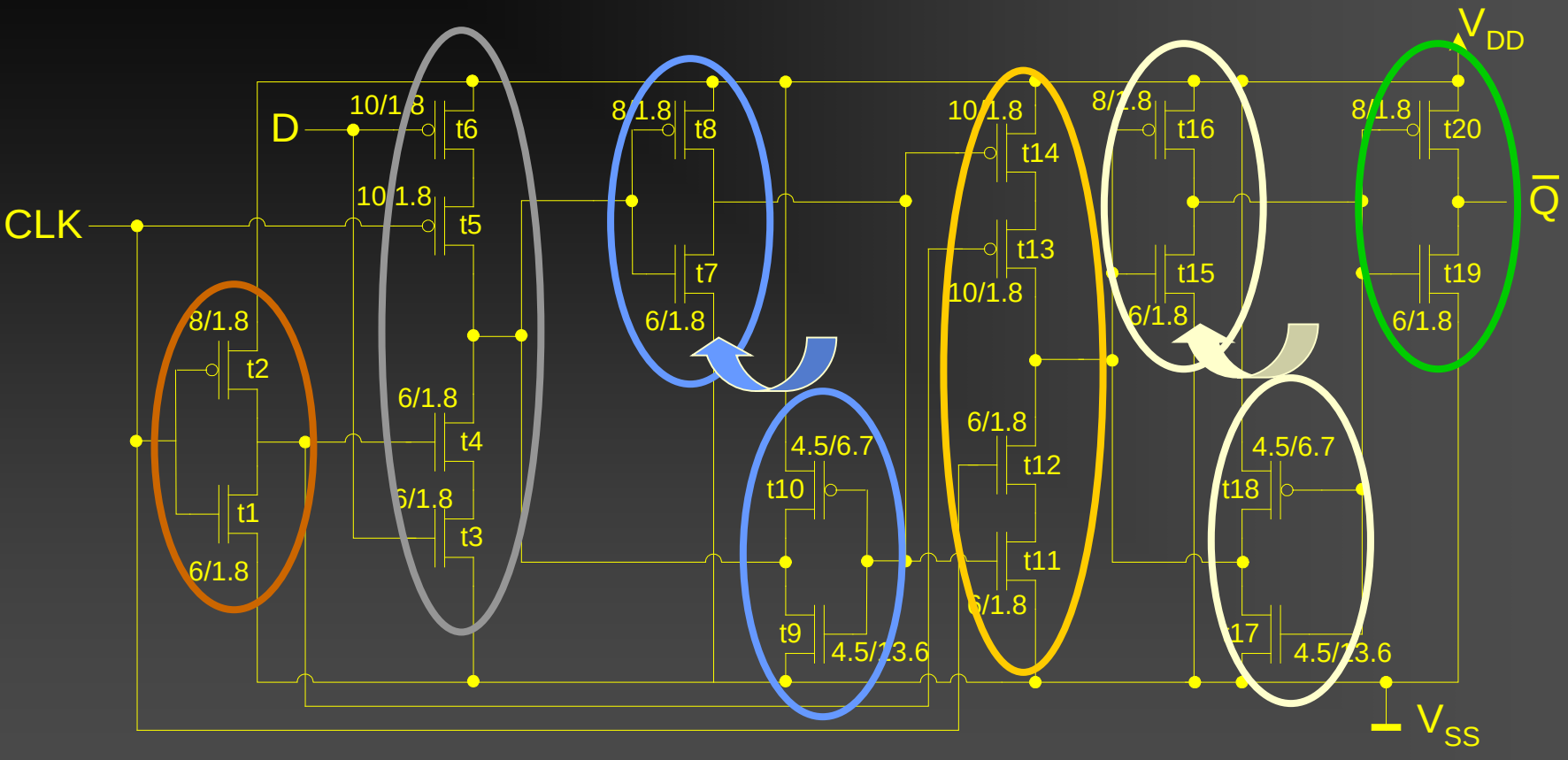


Treiberstärke einstellbar über W/L ☺

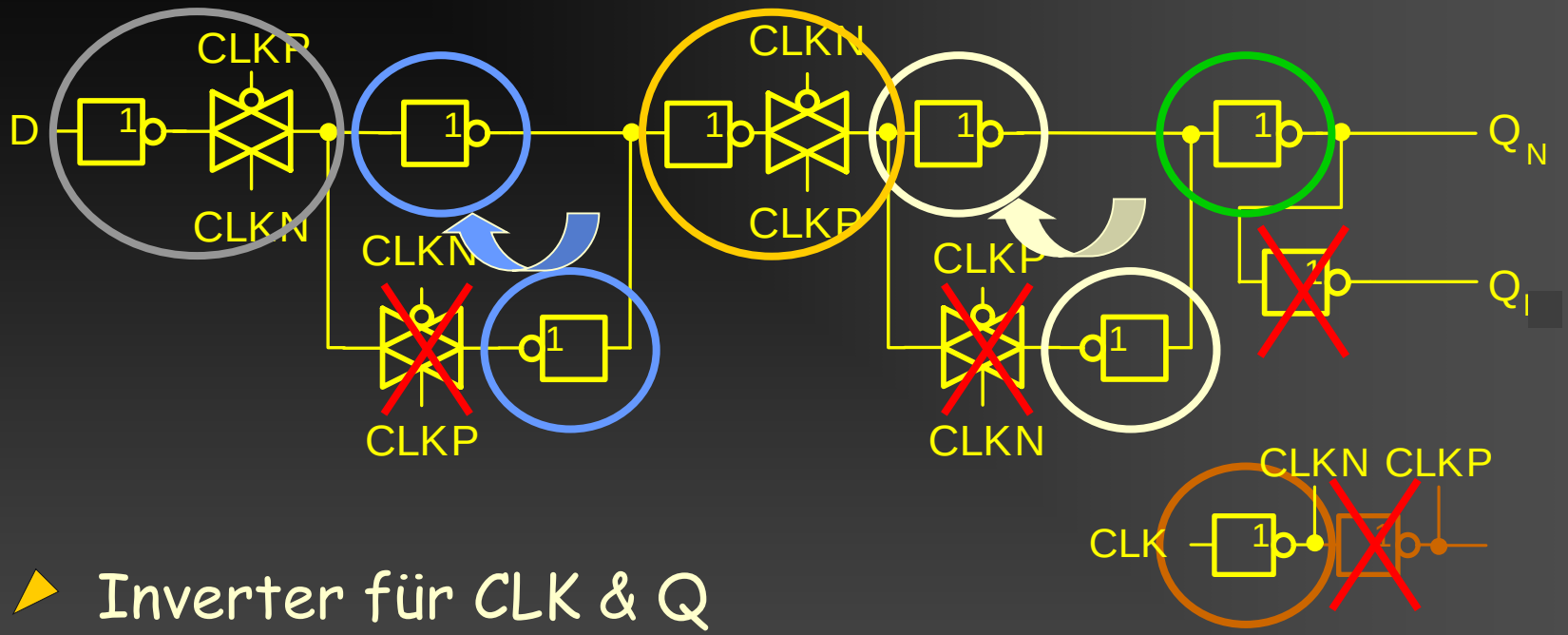
TG läßt sich einsparen

schwacher Treiber

Flip-Flop: Implementierung



Realisierung eines Flip-Flop



- Inverter für CLK & Q
- 2 Speicherschleifen: TGs eingespart (Treiberstärke)
- TGs jeweils am Eingang
- Buffer eingespart (definierte Verhältnisse)

Setup/Hold bei anderen FFs ?

- Bei allen Typen von Flip-Flops und Latches gibt es die Setup/Hold-Problematik
  (wenn auch in unterschiedlicher Ausprägung)
- Beim SR-Latch kann z.B. kann es zu Metastabilität kommen durch
  - einen zu kurzer Puls an S bzw. R, oder
  - die "gleichzeitige" (= zu rasch aufeinander-folgende) Deaktivierung von S und R
- Es gibt kein Patentrezept gegen Metastabilität.

Register

Ein Register ist ein Array von Flip-Flops.

Ein 16-bit Register ist also

- ein Array aus 16 D-Flip-Flops
- mit gemeinsamem Takt
- mit gemeinsamem Clear, Enable, etc.
- Ein- und Ausgänge sind typischerweise zu „Bussen" zusammengefasst (Daten, Adressen)

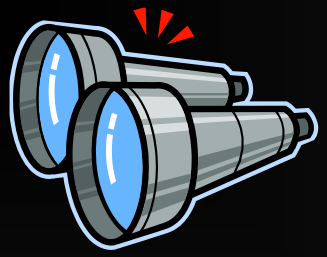
Realisierung eines Speichers

- Flip-Flops: ca. 20 Transistoren/Bit
- SRAM (siehe später): 6 Transistoren/Bit
- DRAM (siehe später): 1 Transistor/Bit (+1 Kondensator)

Realisierung größerer Speicher…

- mittels Flip-Flops ist extrem ineffizient.
- unbedingt mittels RAM-Blöcken aus der Library realisieren.

!



Überblick

- Was ist CMOS ?
- Feldeffekt-Transistor & CMOS-Prozess
- kombinatorische Logikzellen
- sequentielle Logikzellen
- **weitere Logikfamilien**

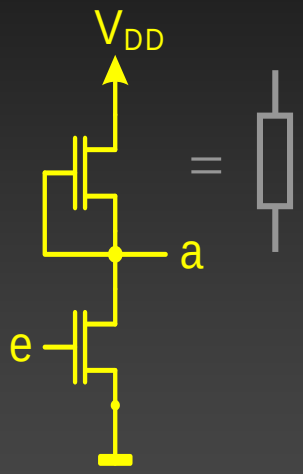
Weitere MOS-Logikfamilien

NMOS

nur n-Kanal FETs

Nachteil:

„weak" 1

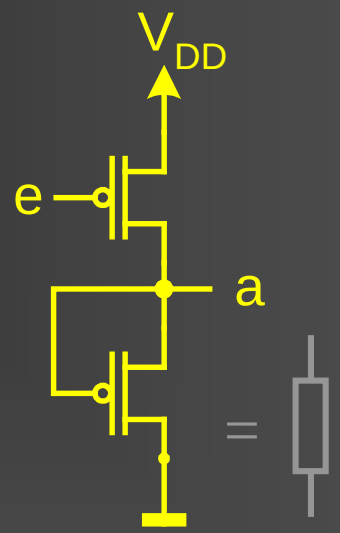


PMOS

nur p-Kanal-FETs

Nachteil:

„weak" 0

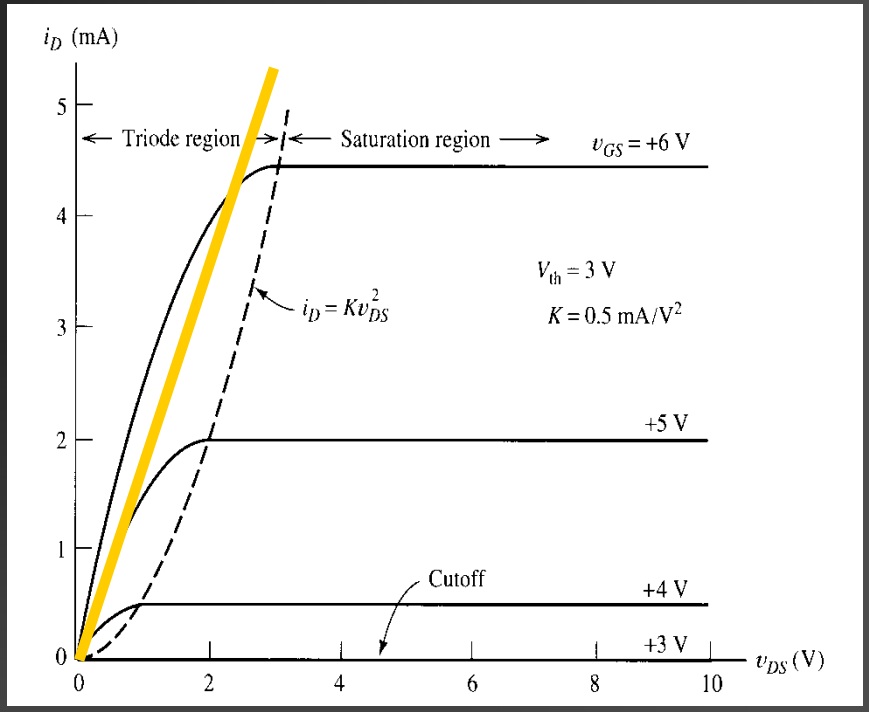


Vorteil: Fertigung billiger (weniger Masken)

Nachteil: statischer Stromverbrauch

Der FET als Widerstand

- Integration von Widerständen ist schwierig (Kohle, Metallfilm?)
- Bei geeigneter Auslegung verhält sich FET in guter Näherung wie Widerstand
- Widerstand einstellbar über Formfaktor W/L



➡ Realisierung v. Widerständen dch FETs

Bipolare Logikfamilien: TTL

TTL (Transistor-Transistor-Logic)

- Prinzip: Logische Verknüpfungen über Dioden-Netzwerke bzw. Transistor-Schalter
- sehr ähnlich wie bei CMOS, aber mit Bipolar-Transistoren statt FETs
- verbraucht im Ruhezustand mehr Energie
- weitgehend kompatibel zu CMOS
- weitgehend von CMOS abgelöst
- legendäre 74xx-Serie (Sylvania 1963)

Bipolare Logikfamilien: ECL

ECL (Emitter-Coupled Logic)

- Prinzip: Umschalten von Strompfaden in Differenzverstärkern (mit Bipolartransistoren)
- Wenig Spannungshub, keine Sättigung, daher extrem schnell
- Extrem hoher Leistungsverbrauch
- Weit verbreitete 10K und 100K-Familien
- Nicht kompatibel zu CMOS und TTL
- Anwendung: Glasfaser-Interface, ATM

Bi-CMOS Logik

... BIpolar und CMOS gemischt

- Schaltung hauptsächlich in CMOS realisiert, aber
- Bipolar-Transistoren sind für höhere Ströme geeignet, daher für die Ausgangsstufen (Treiber) verwendet, z.T für direkte Ansteuerung von kleineren Motoren o.ä.
- Mischen der Technologien macht Fertigung komplizierter und daher teurer

Zusammenfassung (1)

- Grundelement der digitalen Logik ist der Enhancement-FET, wobei bei CMOS der n-Kanal-Typ und der p-Kanal komplementär zum Einsatz kommen.
- Die wichtigsten Parameter des FET sind Schwellspannung und Ausgangsstrom (bzw. Formfaktor)
- Im Idealfall verhält sich ein FET wie ein Schalter: der n-Kanal-FET schließt bei 1 am Steuereingang, der p-Kanal-FET bei 0.
- Die Idealisierung als Schalter funktioniert nur unter geeigneten Randbedingungen. Bei genauerer Betrachtung (im Zeit oder Amplitudenbereich) verhält sich der FET wie ein analoges Bauelement.

Zusammenfassung (2)

- Der Inverter ist die Grundstruktur aller Logik-funktionen. Er lässt sich technologisch einfach implementieren.
- Ersetzt man die beiden Einzeltransistoren durch einen sog. n-Stack bzw. p-Stack, so lassen sich bei geeigneter Abstimmung allgemeine logische Funktionen wie AOI und OAI implementieren, sowie als Sonderfälle auch NAND und NOR.
- Nicht invertierende Funktionen können in CMOS nicht einstufig realisiert werden.
- Weitere typische Elemente sind Transmission Gate, Multiplexer und getakteter Inverter.

Zusammenfassung (3)

- Mittels getakteter Inverter kann ein Latch realisiert werden, durch Master/Slave Kombination zweier Latches ein Flip-Flop.
- Aufgrund der Einschwingzeit der Datenpfade (und insbesondere der Speicherschleife) darf innerhalb des „Decision-Window" (Summe aus Setup- und Hold-Time) keine Änderung der Daten erfolgen, sonst kann Metastabilität auftreten.
- Die Realisierung eines ganzen Speichers mittels Flip-Flop oder Latch ist sehr ungünstig, effizienter sind hier SRAM oder DRAM.

Zusammenfassung (4)

- Neben dem komplementären Ausgang gibt es den Tri-State Ausgang sowie den Open Drain Ausgang.
- Die CMOS-Technologie ist derzeit am weitesten verbreitet, in besonderen Anwendungen findet man jedoch auch bipolare Logikfamilien wie TTL oder ECL, oder auch Bi-CMOS (für hohe Treiberleistung).