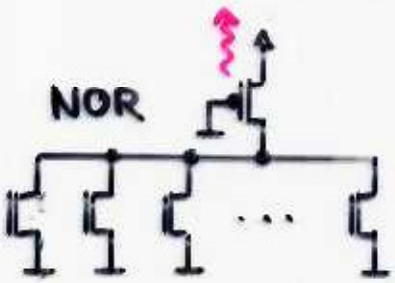
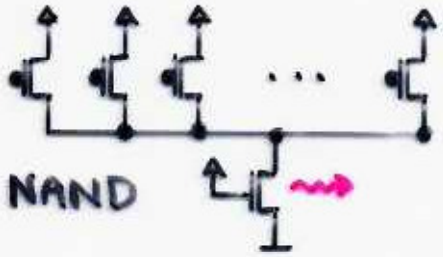
Statische Logik : Typen

① Verhältnislogik

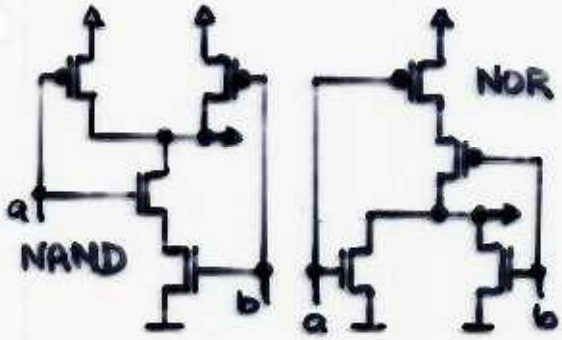
Stromverh. I_p/I_n; $P_{stat}\uparrow$; $A\downarrow$; $\tau\downarrow$; $C_e\downarrow$





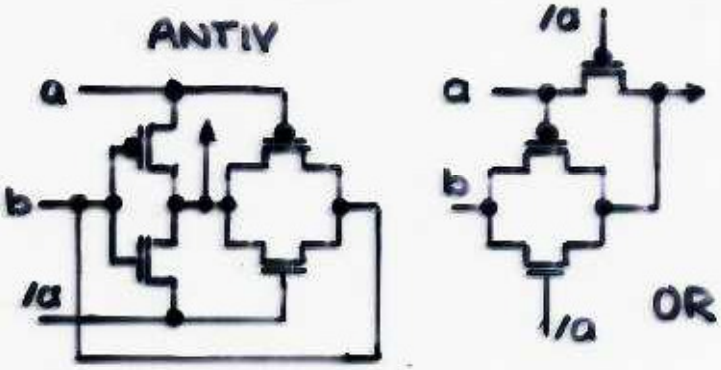
② Komplementärlogik

$C_e\uparrow$; $A\uparrow$; $\tau\uparrow$



③ Schalterlogik

$A\downarrow$; $C_e\downarrow$; τM

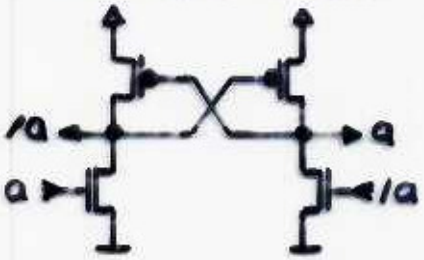


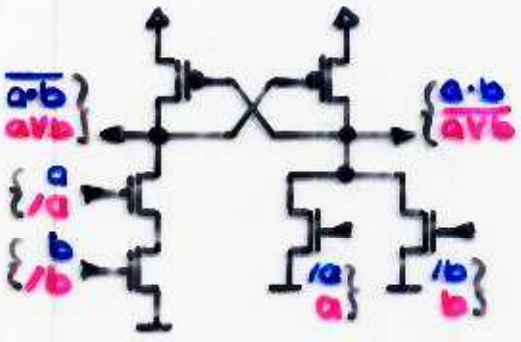
④ Spiegellogik

$C_e\downarrow$; $P_{dyn}\uparrow$; $A\uparrow$; $\tau\downarrow\downarrow$

(Cascode voltage switch logic: CVSL)

Treiber / Inverter:

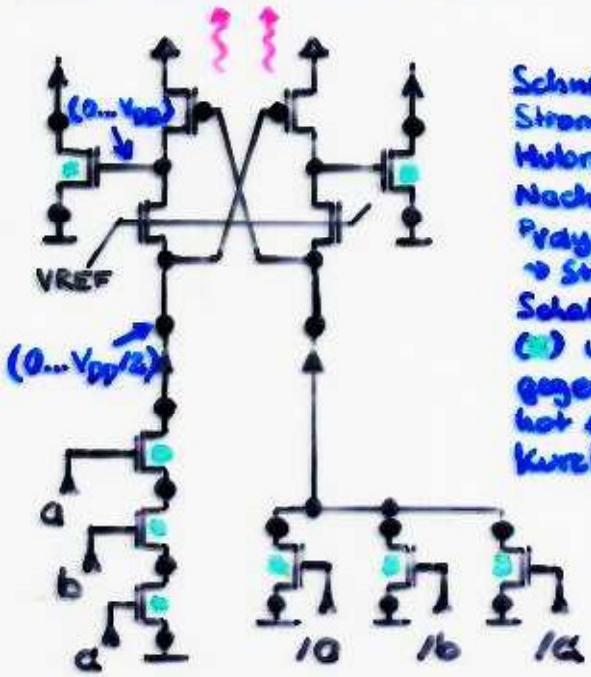




⑤ Differenzlogik

$C_e\downarrow$; $U_{Hub}\downarrow$; $P_{dyn}\downarrow\downarrow$; $\tau\downarrow\downarrow\downarrow$; $P_{stat}\uparrow$; $N_M\uparrow$

(Differential split-level logic: DSL)



Schnellste Logik: Stromtrigger & Hubreduzierung.
Nachteile:
$P_{dyn}\downarrow\downarrow$; $U_{Hub}\downarrow$
→ Störsicherh. ↓
Schalttransistoren (a) unempfindl. gegen V_{DS}-induz. hot electrons: Kurzkanal mögl.

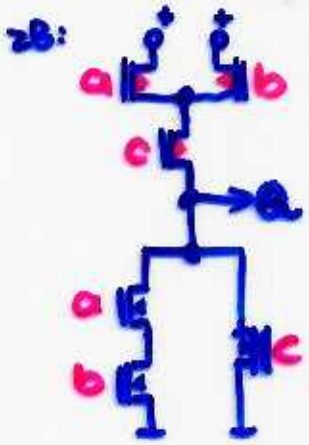
AND/OR/NAND/NOR

KOMLEMENTÄRLOGIK: (CMOS)

Standardform ↝ zid: Statisch Verlustleistungsfrei:

{Parallelschaltung / Serienschaltung} n-Teil ⟺ {Serienschaltung / Parallelschaltung} p-Teil

zB:



p-Teil → $(ab)\vee c = \overline{(\bar{a}\vee\bar{b})\bar{c}}$

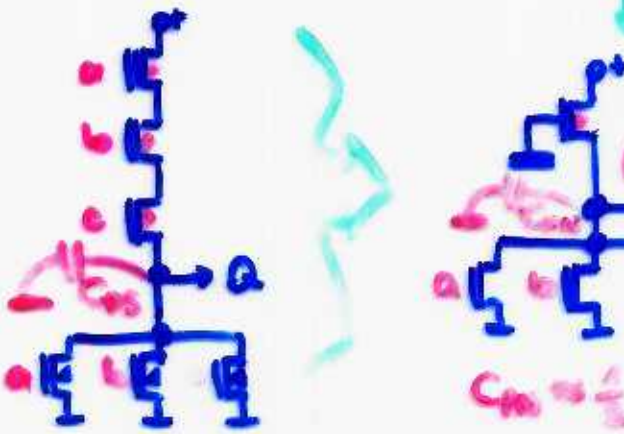
n-Teil → $\overline{(ab)\vee c} = (\bar{a}\vee\bar{b})\bar{c}$

Problem: großer Flächenbedarf für ...

Serien- oder Parallelschaltung?

↳ Laufzeitkette: ungünstig.

VERHÄLTNISLOGIK: (CMOS, NMOS)

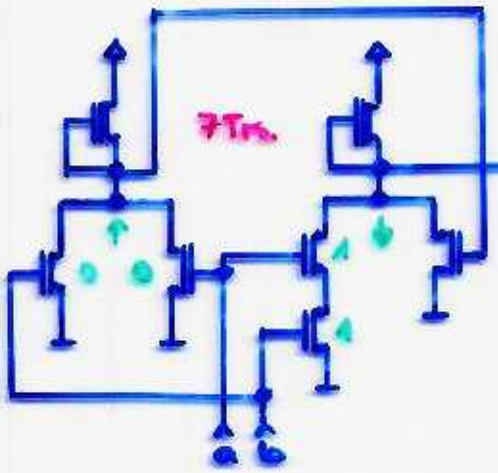


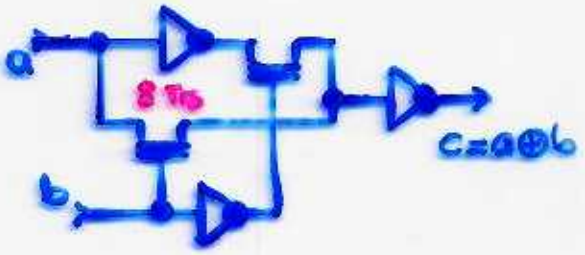
Geschwindigkeitsnachteile nur bei großen, ...

ANTIVALENZ/AEQUIVALENZ

NMOS:

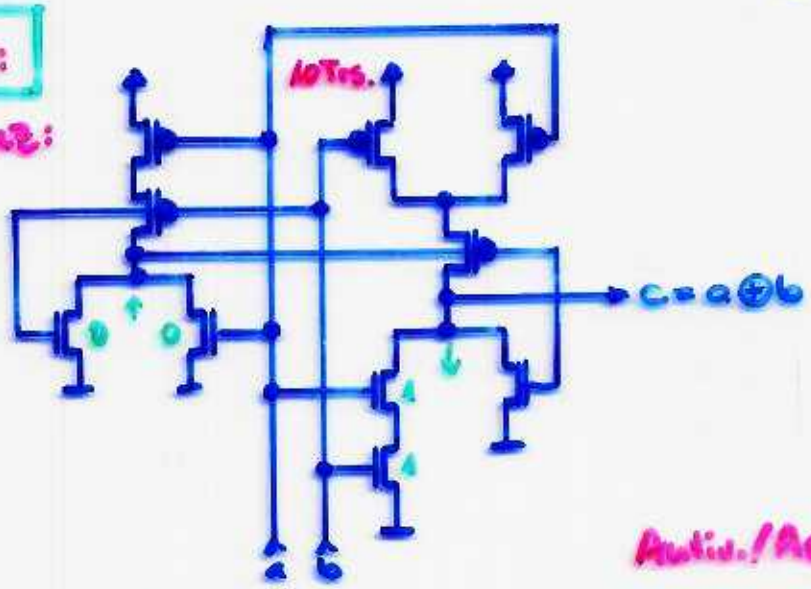
Antivalenz:





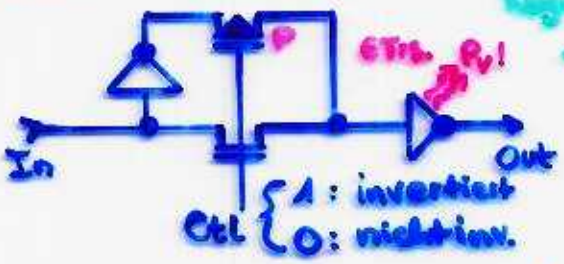
CMOS:

Antivalenz:

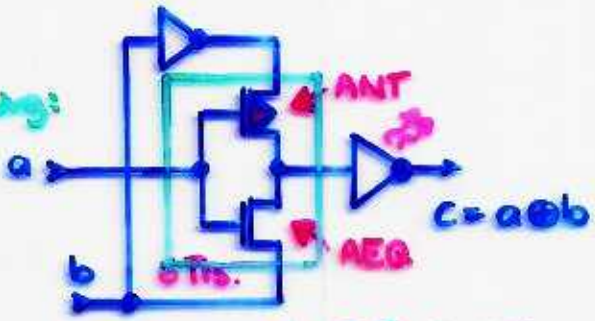


Antiv./Aequivalenz:

inv./ninv. Treiber:

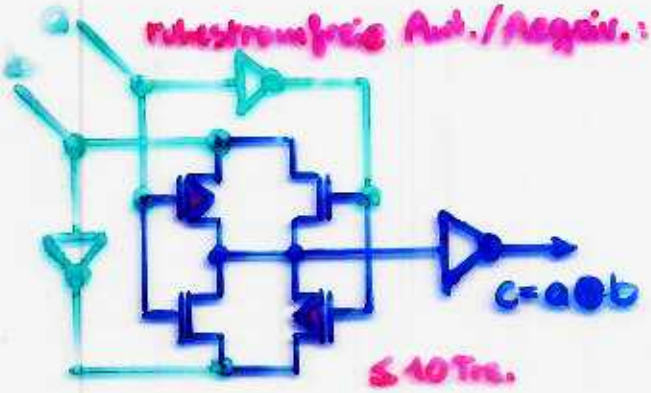


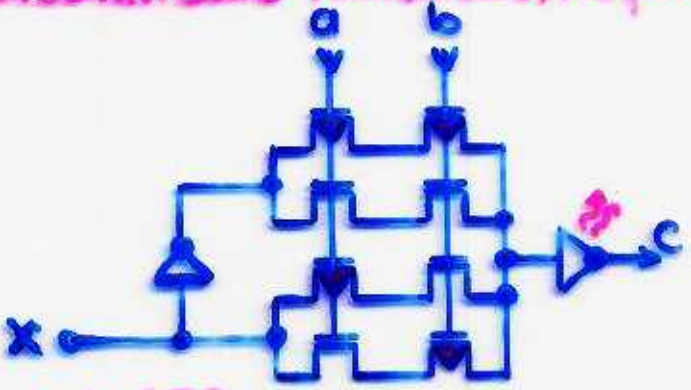
Realisierung:



6 Transistoren

Umschaltbare Antivalenz/Aequivalenz

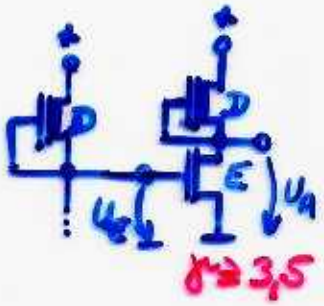




1: AEQ

zu Verhältnislogik:

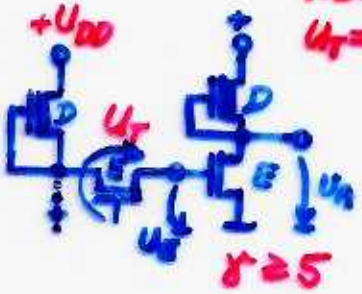
ED - Inv.



$\gamma \geq 3{,}5$

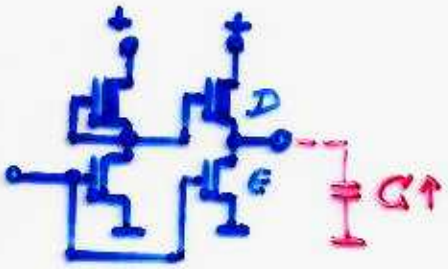
ED - Inv.

nSGT:
$U_T = 92\,U_{DD}$

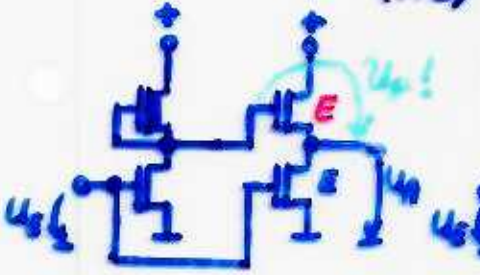


$\gamma \geq 5$

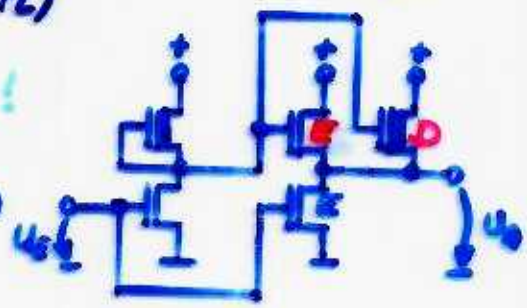
inv. Superbuffer



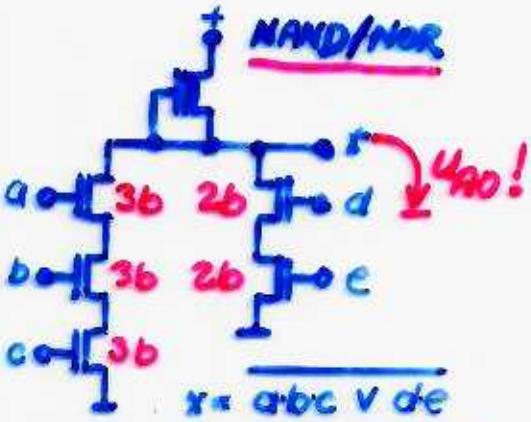
EE - Superbuffer (TTL)



EED - Superbuffer



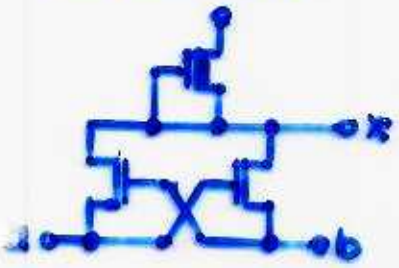
NAND/NOR



$x = \overline{abc \vee de}$

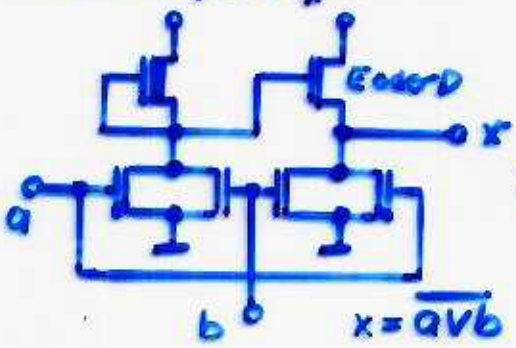
$(x = \bar{a} \vee \bar{b} \vee \bar{c} \cdot \bar{d} \vee \bar{e})$

Aequivalenz



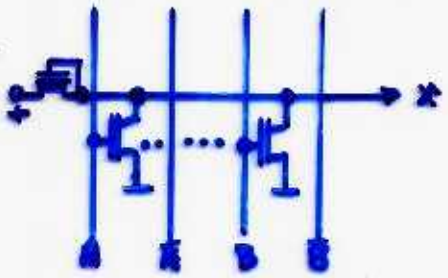
$x = ab \vee \bar{a}\bar{b}$

NOR - Superbuffer

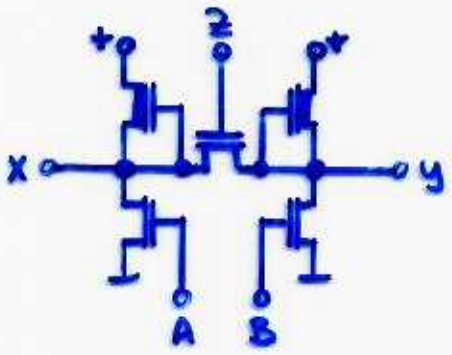


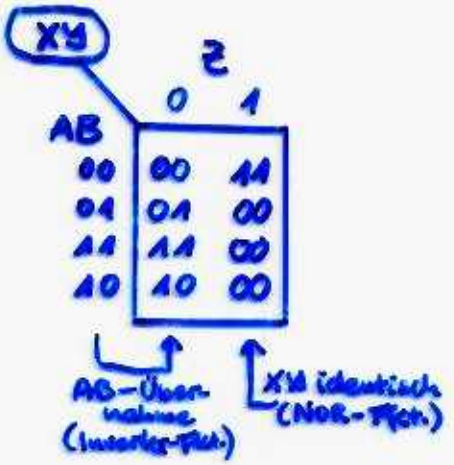
$x = \overline{a \vee b}$

DECODER (NOR)

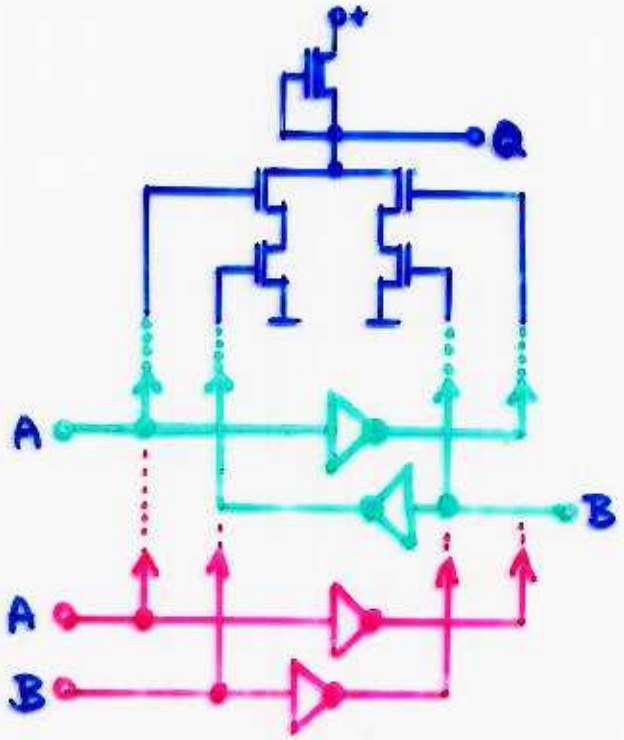


Schalt-NOR:





Äquivalenz (statisch)
Antivalenz (statisch)



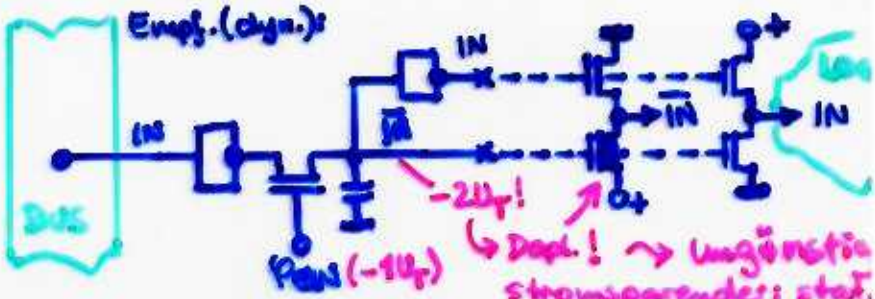
ÄQUIVALENZ:

Q	B=0	B=1
A=0	1	0
A=1	0	1

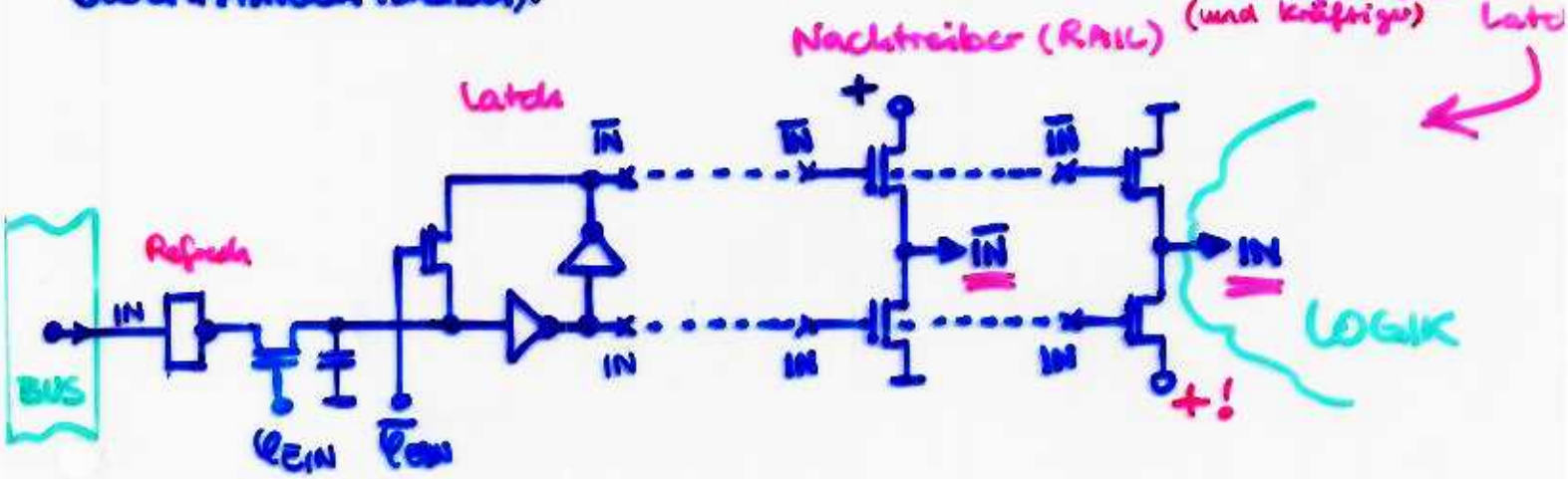
ANTIVALENZ:

Q	B=0	B=1
A=0	0	1
A=1	1	0

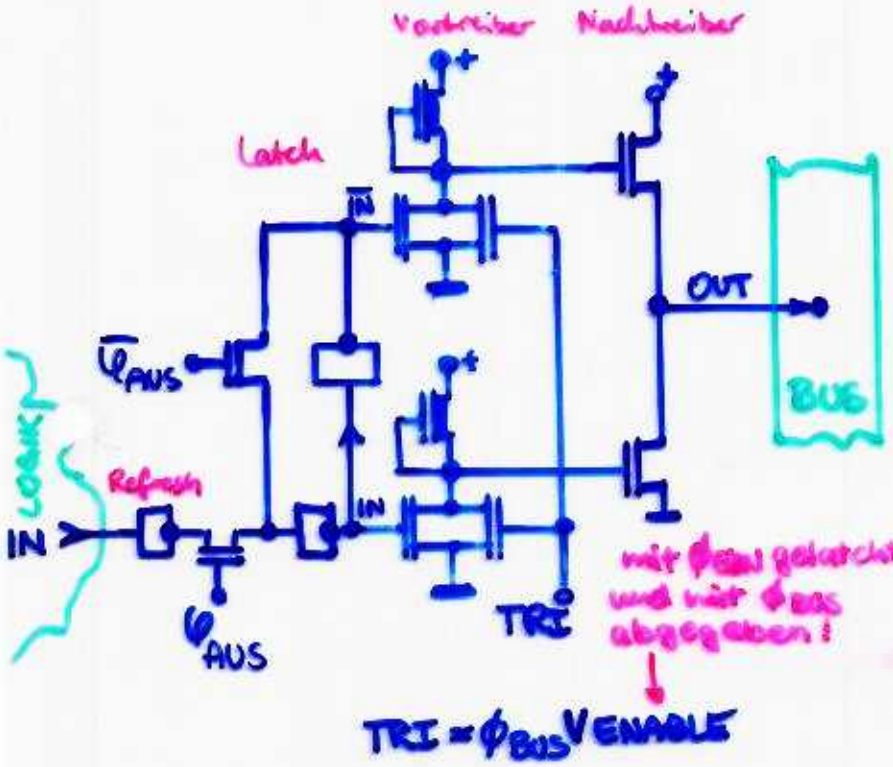
SCHNITTSTELLEN:

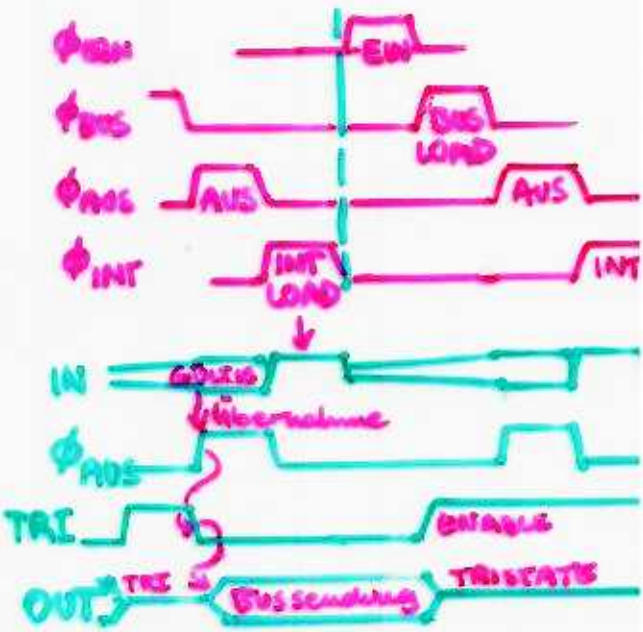


BUSEMPFÄNGER (statisch):



BUSSENDER (TRISTATE-BUS)(statisch):





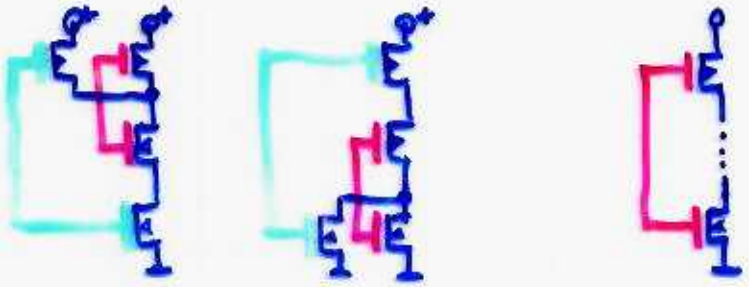
TRI darf erst nach φBUS off öffnen
Sonst geht Müll auf dem Bus!

3D-CMOS

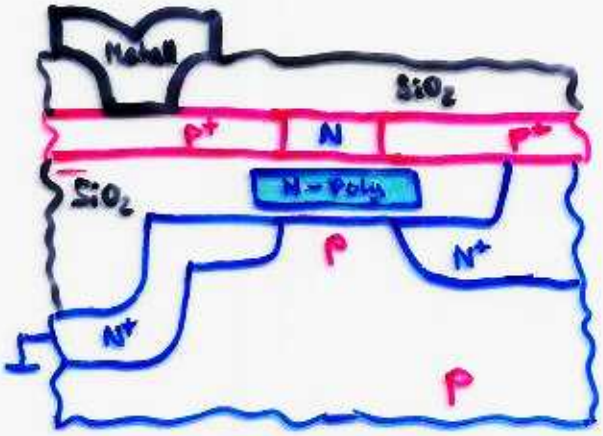
IEEE - SC 19 No 1 Febr. 84 S.37-39

B. Haeffinger, S.T. Liu, B. Vajdic „A Three-Dimensional CMOS Design Methodology"

CMOS-typisch: gemeinsamer Gateanschluß zweier Transistoren (p und n)



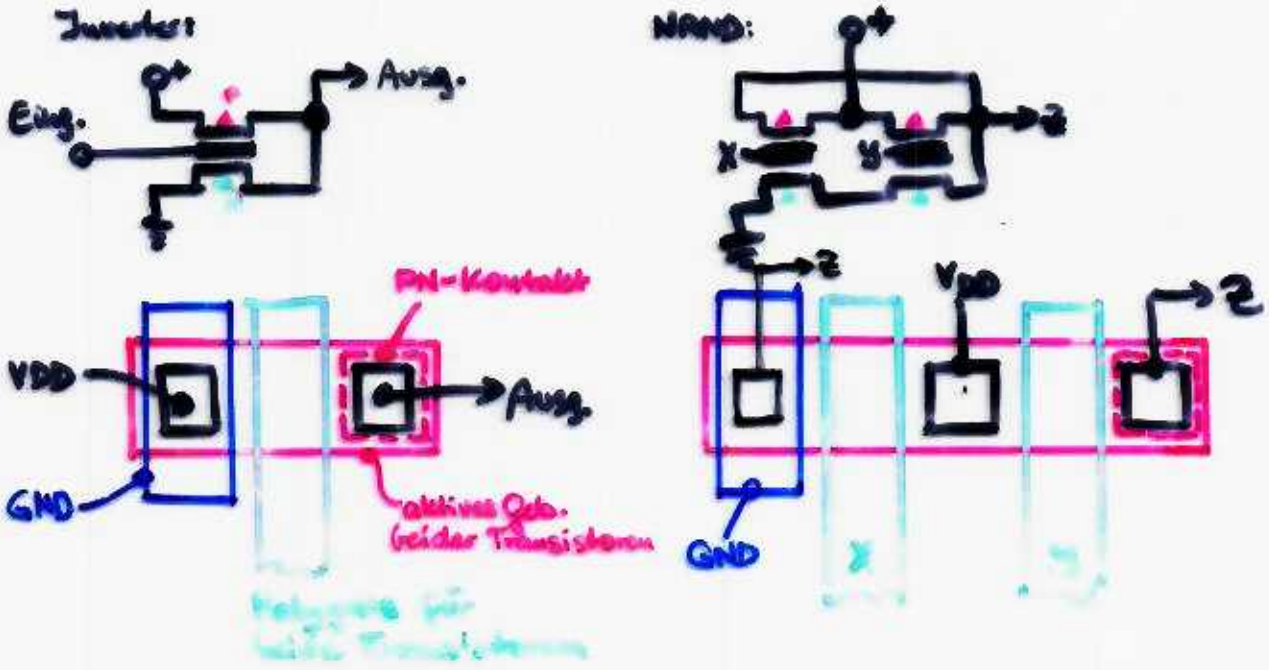
Idee: n- und p-Kanal-Gate identisch: beide Gates und beide Transistoren (akt. Gebiet) mit selb. identischer Maske strukturiert.



Effekte: z.B. Volladder:

Bulk-CMOS:	13 × 15 = 195	λ^2
SOI:	11 × 12 = 132	λ^2
3D-CMOS:	8 × 7 = 56	λ^2

→ Kleiner als NMOS
→ Verlustleistungsfrei (statisch)
→ minimal denkbar: 5+1 Masken
↳ fehlerarm entwerfbar,
↳ CAD-freundlich



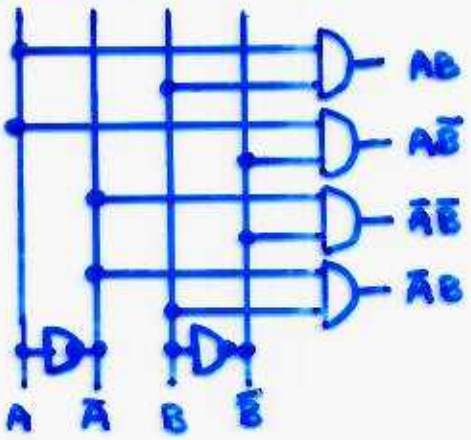
— : Metall-P^+-Kontakt
— : aktives Transistorgebiet
— : vergrabener Bulkanschluß

3D

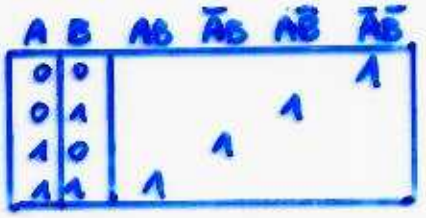
Programmierbare Logik:

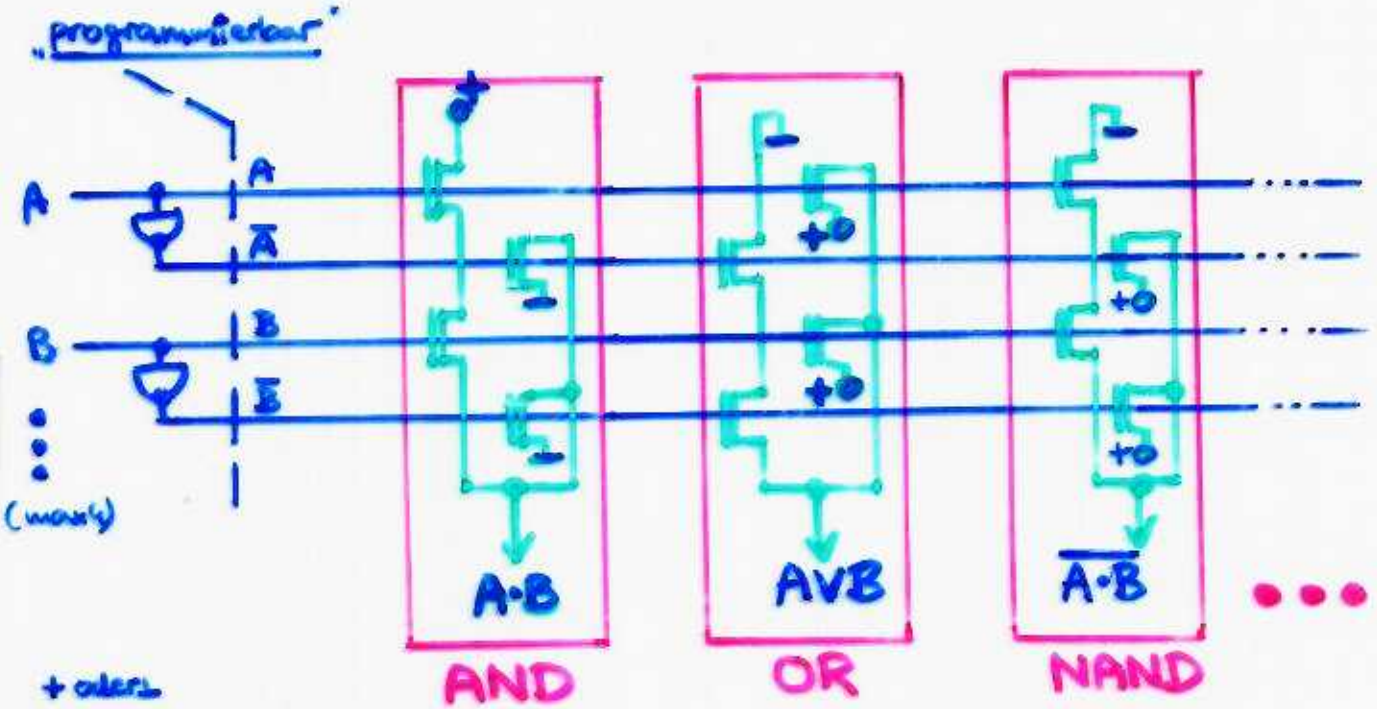
DECODER

klassisch (zb. 2bit):

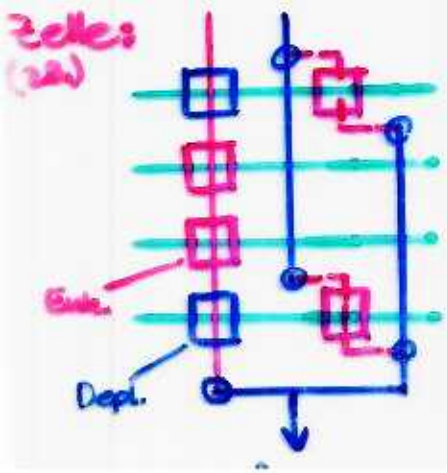


Nachteil: • hoher P_V
• schlecht „erweiterbar"





+ oder-

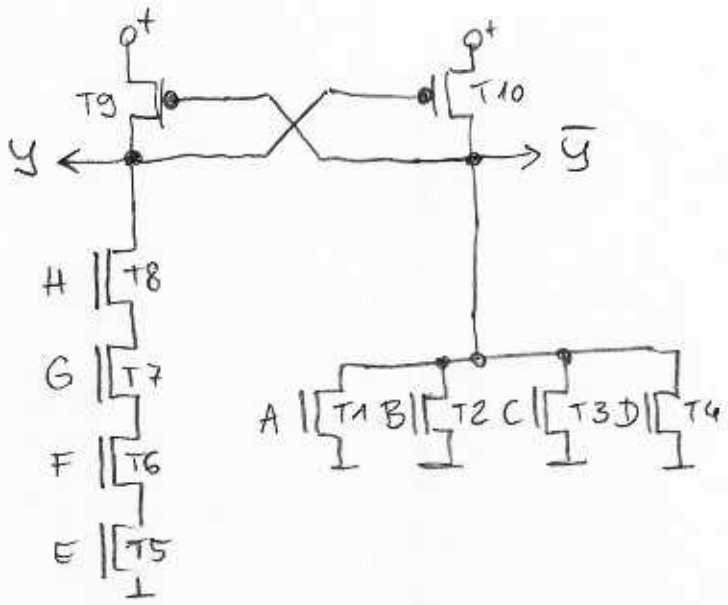


Vorteile:

- P_V ↓
- „erweiterbar"
- bei Verwendung RAIL-codierter Terme (A und $\bar{A}$, ...) Verlustleistungsfreiheit (statisches P_V) und Vermeidung „wilder" Logik !

GH 10.6.92

IAND4, IOR4 (Spiegellogik-Gatter)

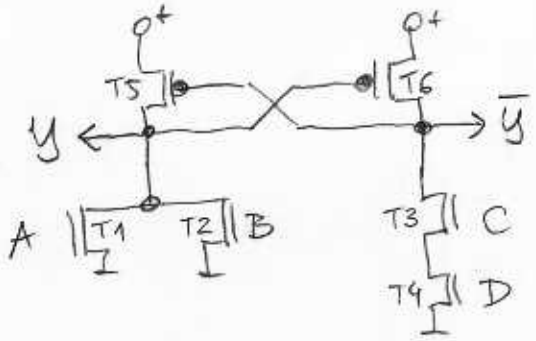


A	B	C	D	E	F	G	H	y	$\bar{y}$
a	b	c	d	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	OR	NOR
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	a	b	c	d	NAND	AND

1 Layout mit 2 Pinouts für AND/NAND bzw. OR/NOR zu bezeichnen: 2 Zeilen

GH 10.6.92

IAND2, IOR2 (Spiegellogik-Gatter)



A	B	C	D	y	$\bar{y}$
a	b	$\bar{a}$	$\bar{b}$	NOR	OR
$\bar{a}$	$\bar{b}$	a	b	AND	NAND

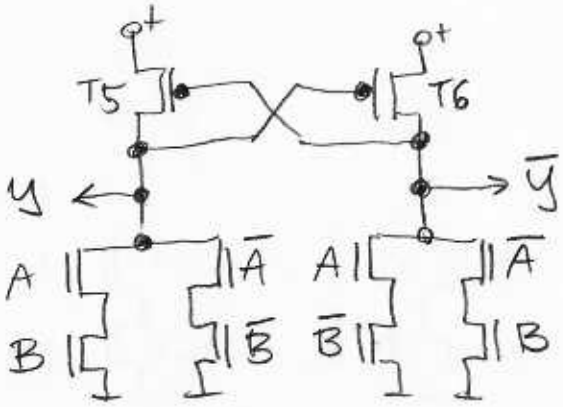
1 Layout mit 2 Pinouts für AND/NAND bzw. OR/NOR zu bereitstellen: 2 Zellen!

T1, T2	(B/L)	= 2/1
T3, T4	(B/L)	= 3/1
T5, T6	(B/L)	= 6/1

GH 10.6.92

IANT2

(Spiegellogik-Gatter)



A	B	$\bar{A}$	$\bar{B}$	y	$\bar{y}$
a	b	$\bar{a}$	$\bar{b}$	ANT	AEQ

1 Layout mit 1 Pinout für ANTIVALENZ-Flat.