
Vorlesung T2

4. Digitale Schaltungstechnik: NMOS- und CMOS-Gatter

Norman Hendrich

Universität Hamburg, Fachbereich Informatik

Vogt-Kölln-Str. 30, D 22527 Hamburg

hendrich@informatik.uni-hamburg.de

`tams-www.informatik.uni-hamburg.de/lehre/`

Übersicht

- Boole'sche Algebra
- MOS-Transistor als Schalter

Gatter in NMOS-Technologie

- Inverter
- Schalt- und Last-Transistor
- NOR- und NAND-Gatter

Gatter in CMOS-Technologie

- Inverter, NOR, NAND
- Komplexgatter
- "transmission gates"
- Flipflop, SRAM- und DRAM-Zellen

Literaturhinweise

Vorlesungs-Skript T2

<http://tams-www.informatik.uni-hamburg.de/lehre/ss2003/>

Alle üblichen Lehrbücher zur technischen Informatik

Interaktive CMOS-Gatter Demonstration (Java Applets):

<http://tams-www.informatik.uni-hamburg.de/applets/cmos/>

Herstellungsschritte (Masken) für einen CMOS-Inverter:

<http://tech-www.informatik.uni-hamburg.de/lehre/icPrak/cmosInv.pdf>

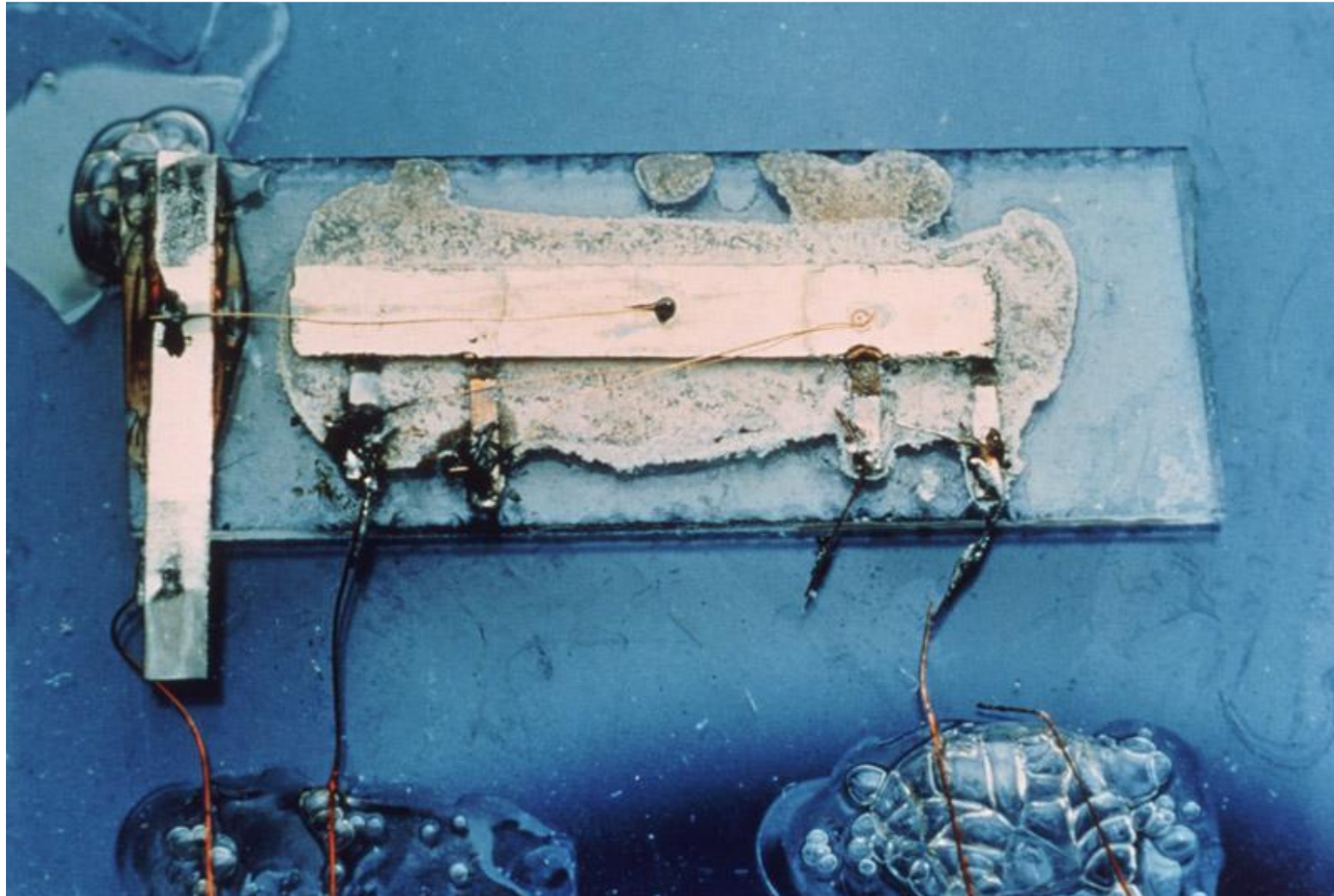
Einige "Klassiker" zur Vertiefung:

Mead & Conway, "Introduction to VLSI systems", Addison-Wesley 1980

Weste & Eshragian, "Principles of CMOS VLSI design", Addison-Wesley, 1993

International Technology Roadmap for Semiconductors, public.itrs.net

Erstes IC



(Jack Kilby, Texas Instruments 1958, Nobelpreis 2000)

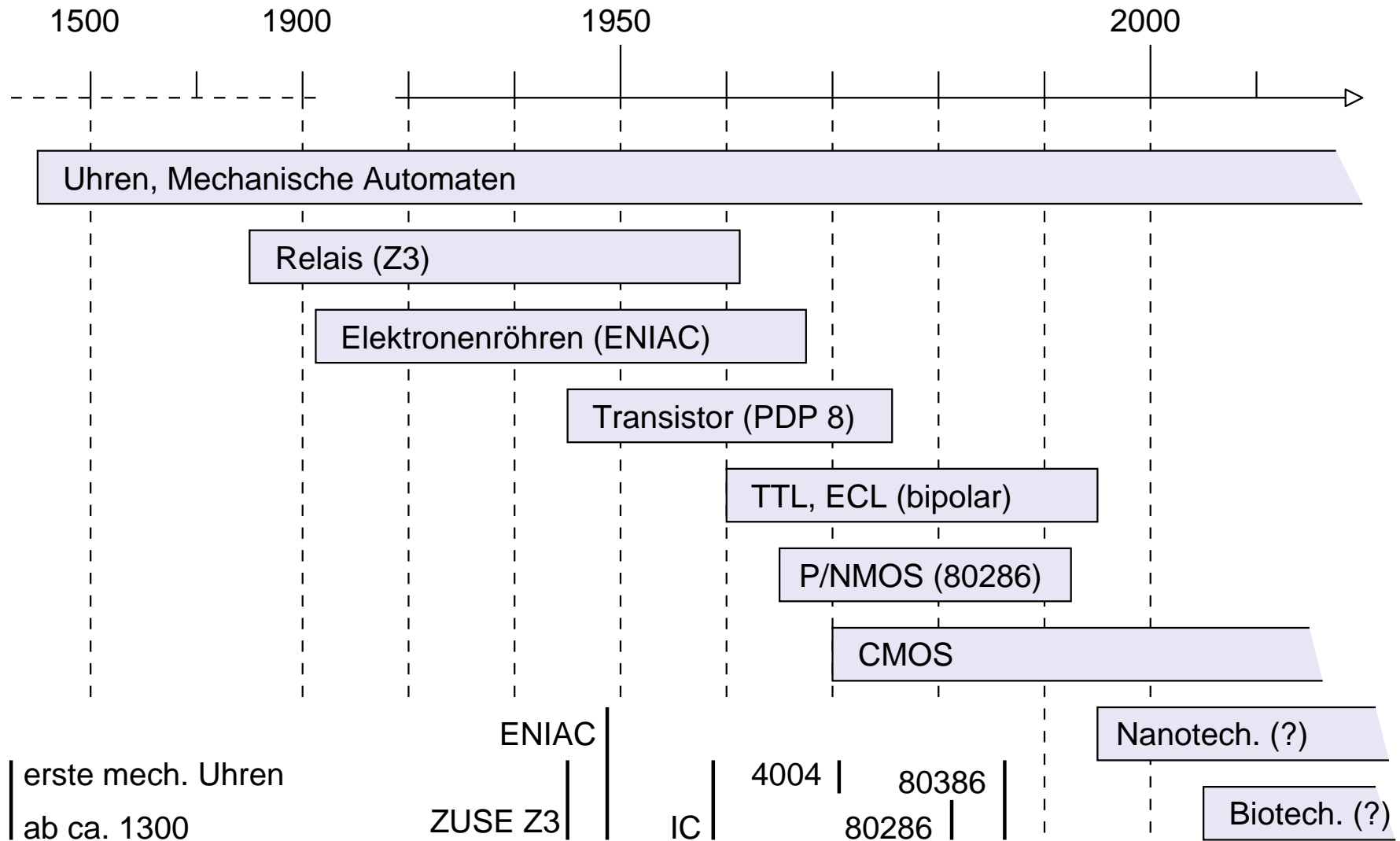
Boole'sche Algebra

- logische Operationen '+' und '*' auf dem GF(2)
- de-Morgan Regeln usw,
- einige wichtige Grundfunktionen:

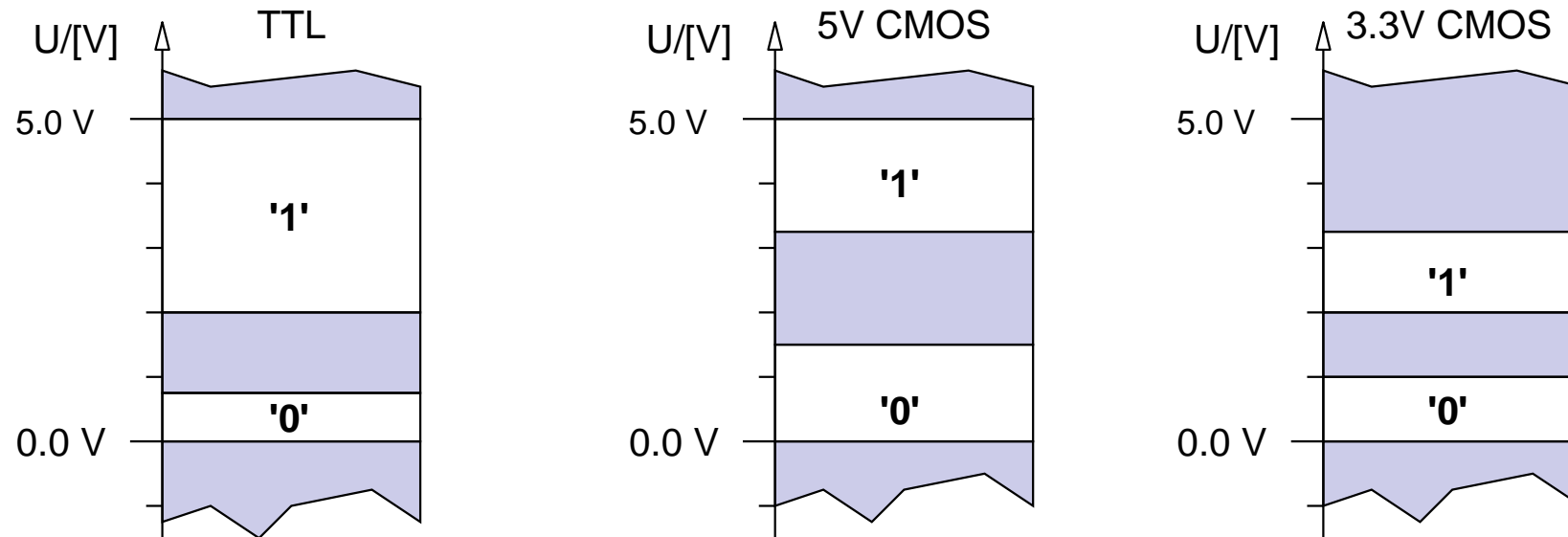
AND			NAND			OR			NOR		
A	B	Y	A	B	Y	A	B	Y	A	B	Y
0	0	0	0	0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	1	0	0
1	1	1	1	1	0	1	1	1	1	1	0

- Erinnerung: logische Vollständigkeit, z.B. { AND, NOT }
- diverse Möglichkeiten zur technischen Realisierung
- unter anderem mit MOS-Transistoren: Hochintegration

Zeitachse



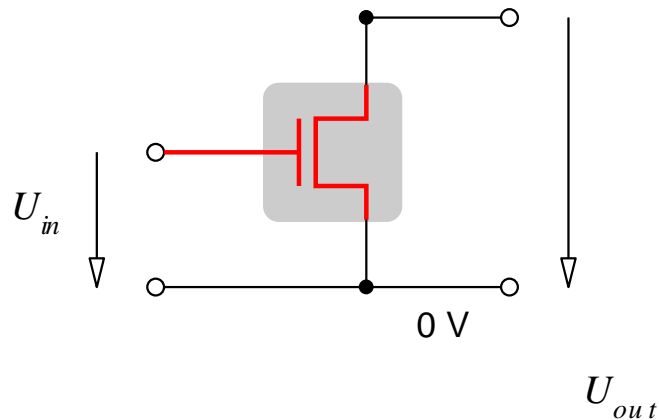
Repräsentation



Repräsentation := Zuordnung phys. Größen zu den logischen Werten:

- Spannungen (TTL, NMOS, CMOS-Technologien)
- Ströme (ECL-Technologie)
- Magnetisierung (Disketten, Festplatten)
- Beleuchtungsstärke (CD, DVD, Glasfaserkommunikation)
- usw.

MOS-Transistor:

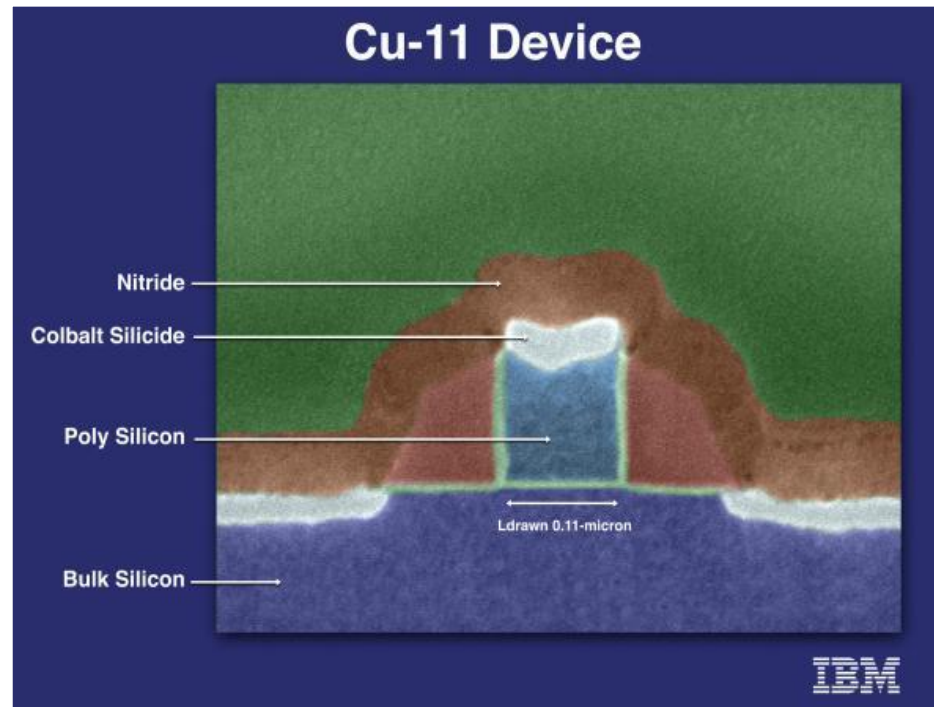


MOS-Transistor:

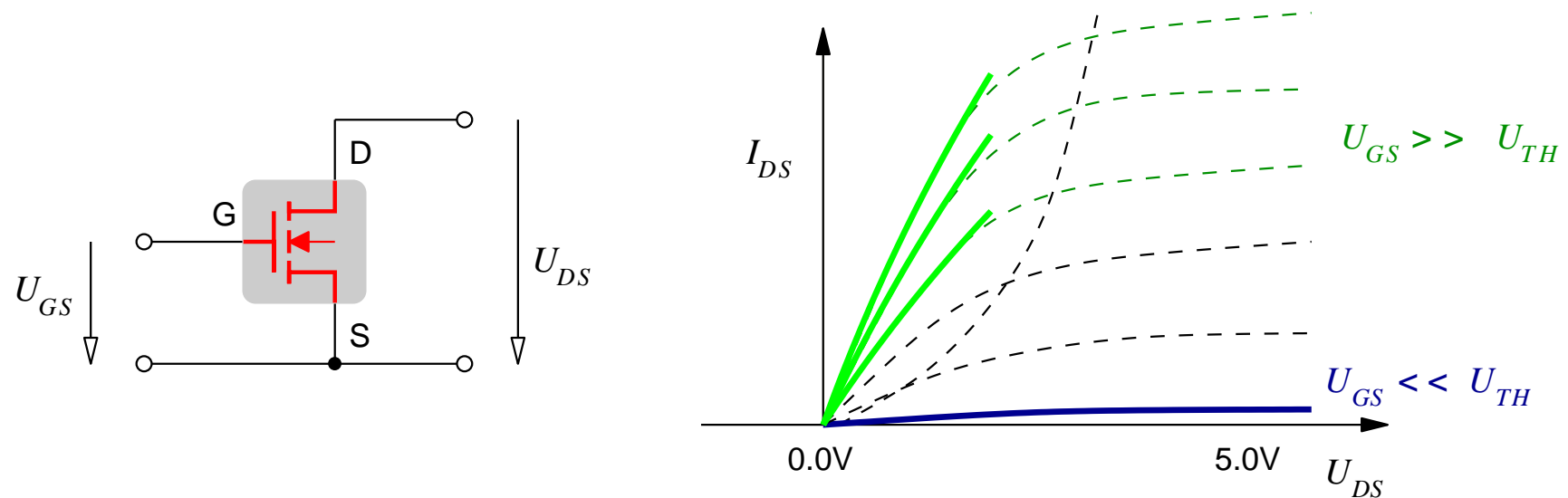
- einfaches Prinzip
- komplexe Details
- sehr genauere theoretische Modelle verfügbar

Digitaltechnik mit MOS-Transistoren

- einfaches "Schalter"-Modell reicht völlig aus



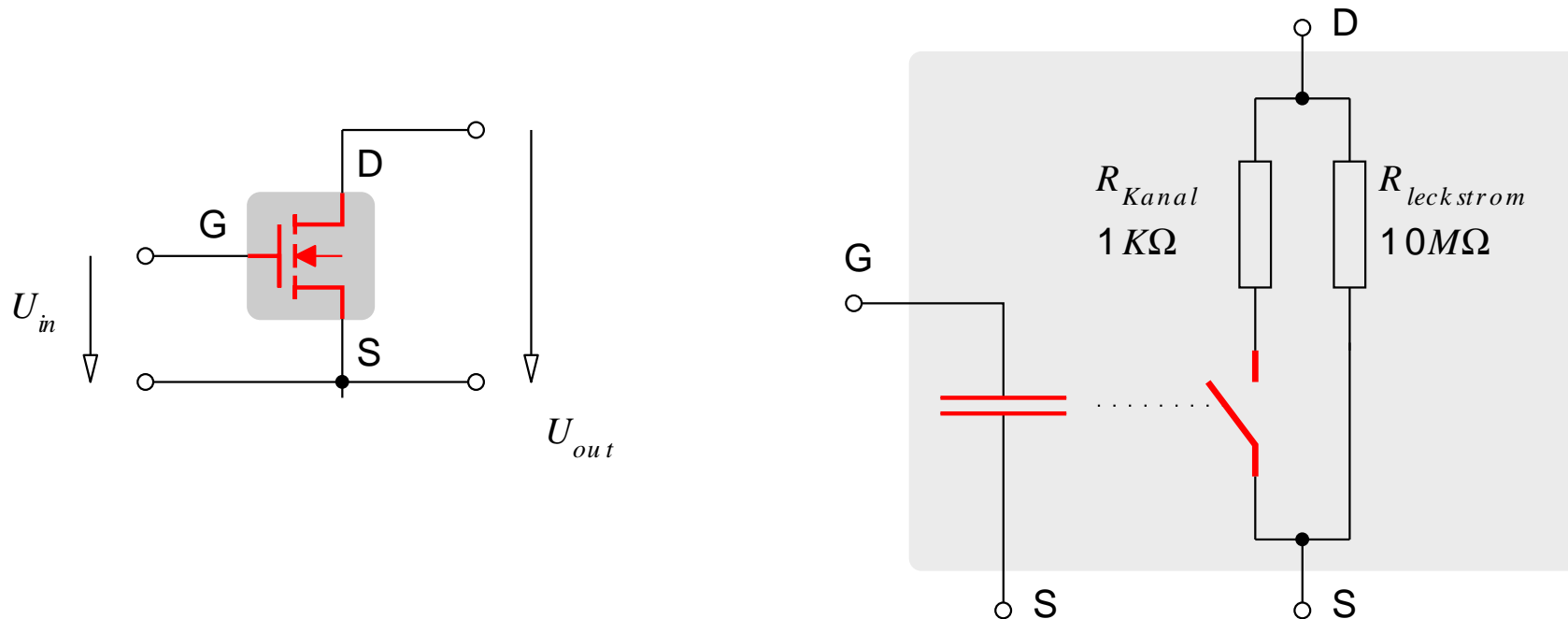
NMOS-Transistor als Schalter



sehr einfaches Verhalten wenn $U_{GS} \gg U_{TH}$ oder $U_{th} \ll U_{TH}$:

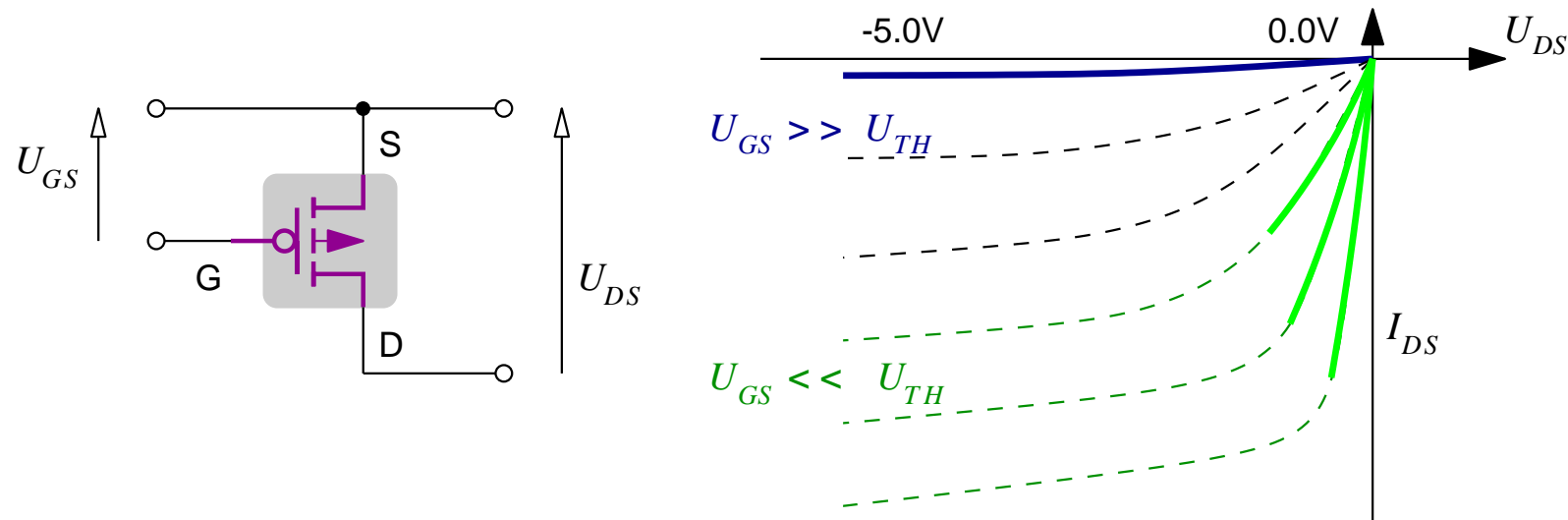
- $U_{GS} \gg U_{TH}$ "eingeschaltet", der Transistor leitet
Kanal- Widerstand typ. 10..1000 Ω
- $U_{GS} \ll U_{TH}$ "ausgeschaltet", der Transistor ist gesperrt
Leckströme, Widerstand > 10M Ω

NMOS: Ersatzschaltbild



- idealer Schalter für den leitenden / gesperrten Kanal
- Widerstände modellieren das nicht-ideale Verhalten
- idealer Kondensator zwischen Gate und Source
- Schwellspannung U_{TH} über Dotierung einstellbar
- PMOS- Transistor komplementär: eingeschaltet für $U_{GS} \ll U_{TH}$

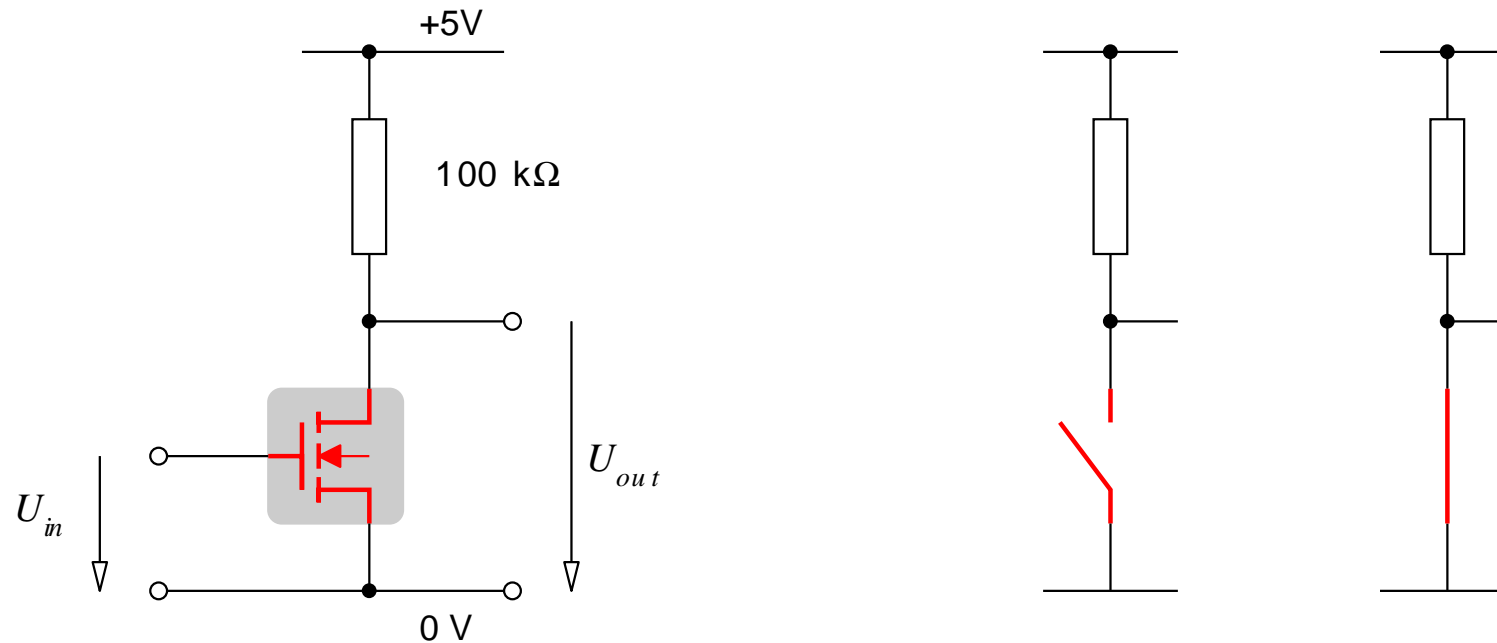
PMOS-Transistor als Schalter



PMOS-Transistor komplementär zum NMOS-Transistor

- Schwellspannung $U_{TH} < 0$, z.B. - 2.5 V
- $U_{GS} \ll U_{TH}$ "eingeschaltet", der Transistor leitet
Kanal- Widerstand typ. 10..1000 Ω
- $U_{GS} \gg U_{TH}$ "ausgeschaltet", der Transistor ist gesperrt
Leckströme, Widerstand $> 10M\Omega$

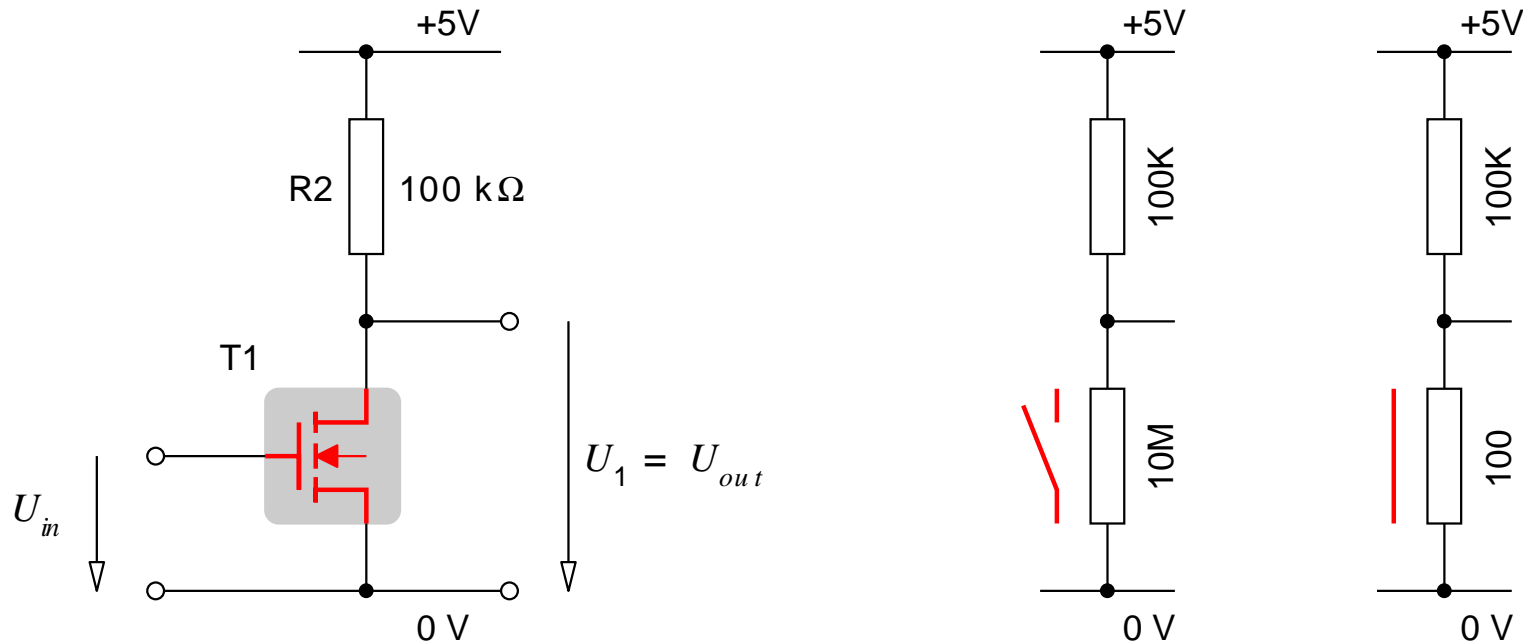
NMOS: Inverter



Inverter besteht aus zwei Komponenten:

- Schalt-Transistor (NMOS, selbstsperrend)
- Lastwiderstand (Aufbau und Dimensionierung s.u.)
- Warum ist das ein Inverter? Spannungsteiler ausrechnen:

NMOS: Spannungen am Inverter



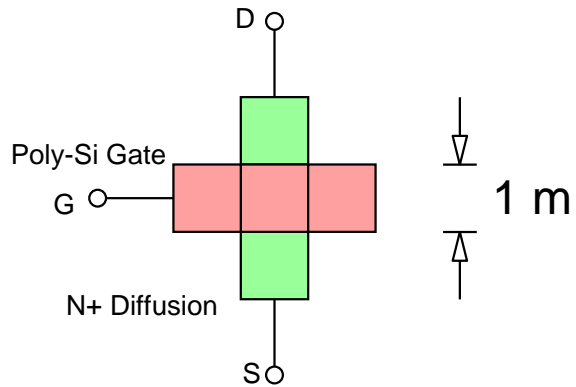
Spannungsteiler:

$$U_1 = U_{ges} * R_1 / (R_1 + R_2)$$

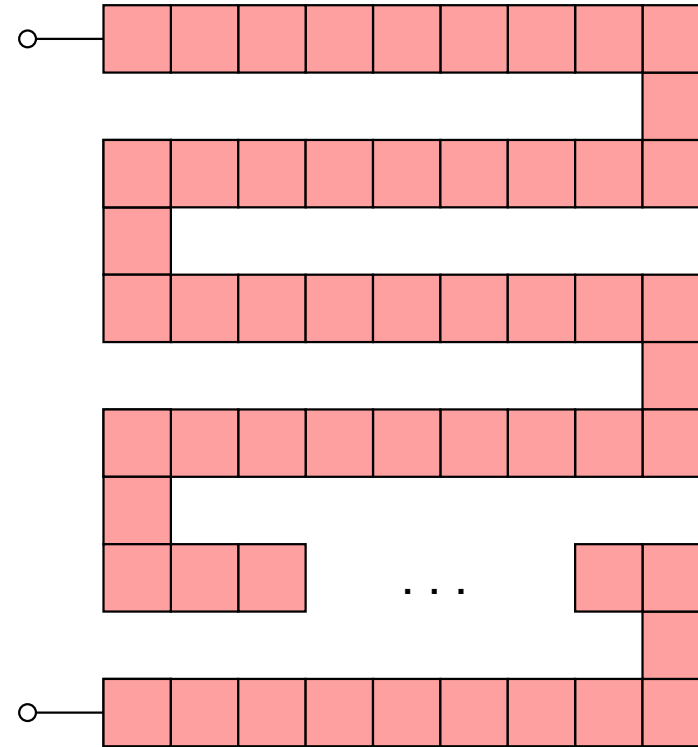
$U_{in} = 0V$ (logische '0')
 $\Rightarrow R_1 = 10M\Omega$ (T1 sperrt)
 $\Rightarrow U_1 = 5V * (10M / 10.1M)$
 $\Rightarrow U_1 = 4.95V$ (logische '1')

$U_{in} = 5V$ (logische '1')
 $\Rightarrow R_1 = 100\Omega$ (T1 leitet)
 $\Rightarrow U_1 = 5V * (100 / 100.010)$
 $\Rightarrow U_1 = 0.005V$ (logische '0')

NMOS: Lastwiderstand?!



Transistor-Layout
(1.0 m Prozess)



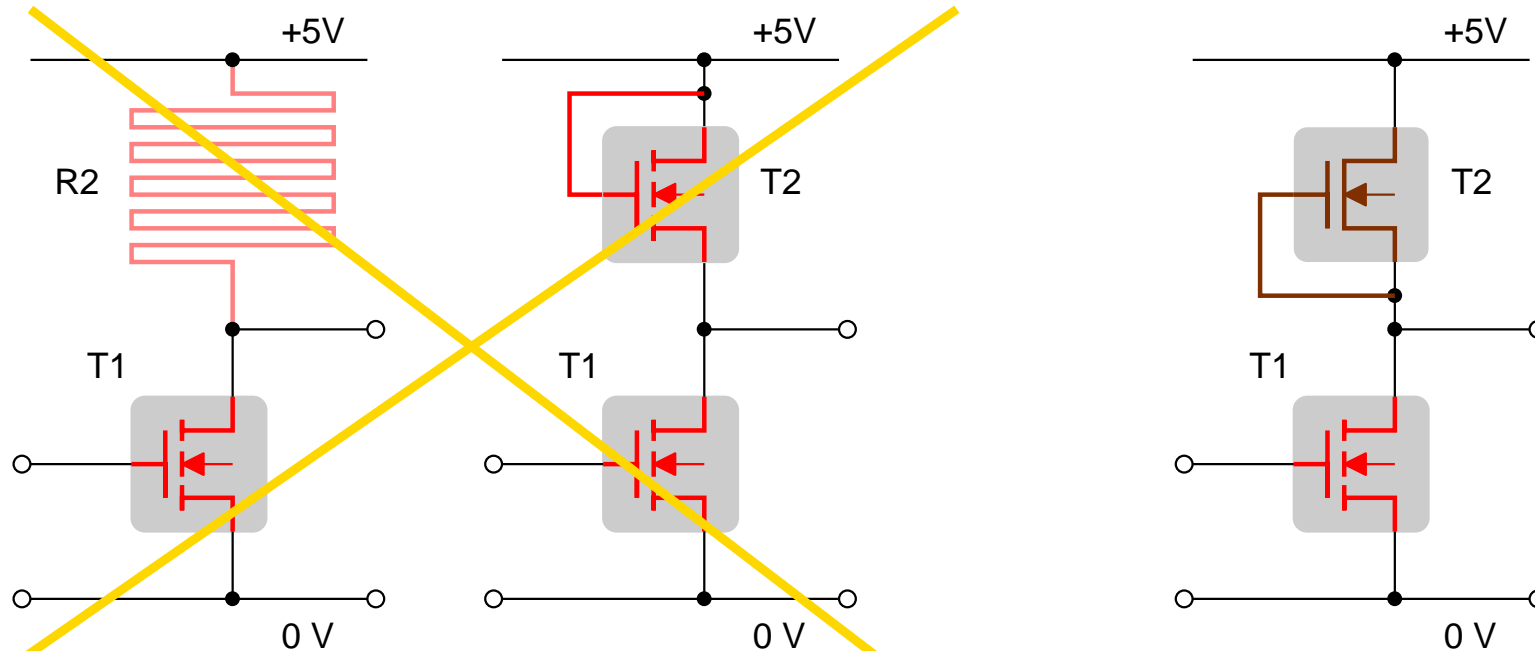
typ. Widerstandswerte:

Poly $40\Omega/$ Alu $0.04\Omega/$

⇒ $100\text{k}\Omega$ erfordert 2500 Polysilizium

⇒ viel grösser als der Transistor, Platzverschwendung

NMOS: Lasttransistor



Polysilizium-
Widerstand

(Platzbedarf zu hoch)

Last-Transistor
(selbstsperrend)

(zu langsam)

Last-Transistor
(selbstleitend)

(erfordert extra Maske)

- selbstleitender Last-Transistor, Gate mit Source verbunden
- Widerstandswert über Dotierung einstellbar
- 1 zusätzlicher Prozess-Schritt erforderlich

NMOS: *Layout des Inverters*



(a) Patterning SiO₂.



(d) Placing diffused region.



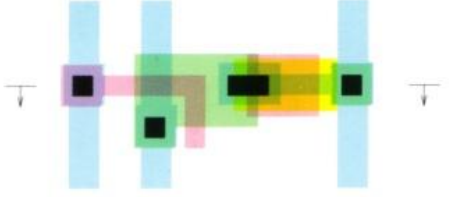
(b) Patterning ion implantation.



(e) Placing contact cuts.



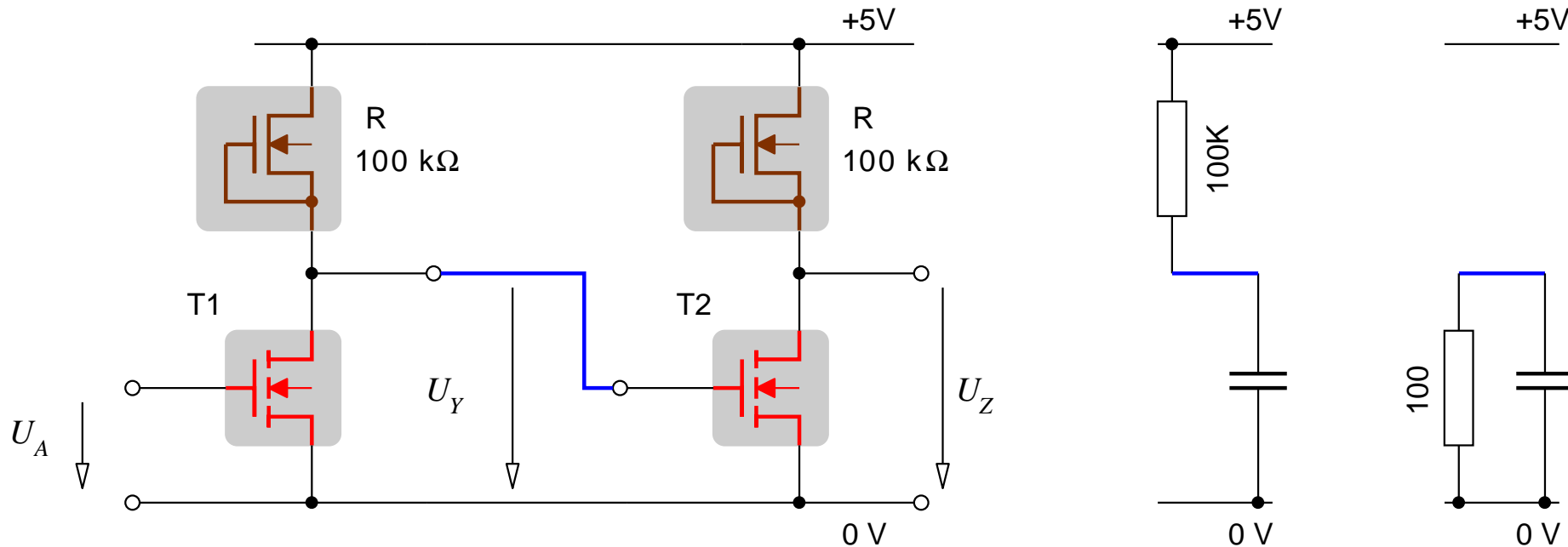
(c) Patterning polysilicon.



(f) Patterning the metal layer.

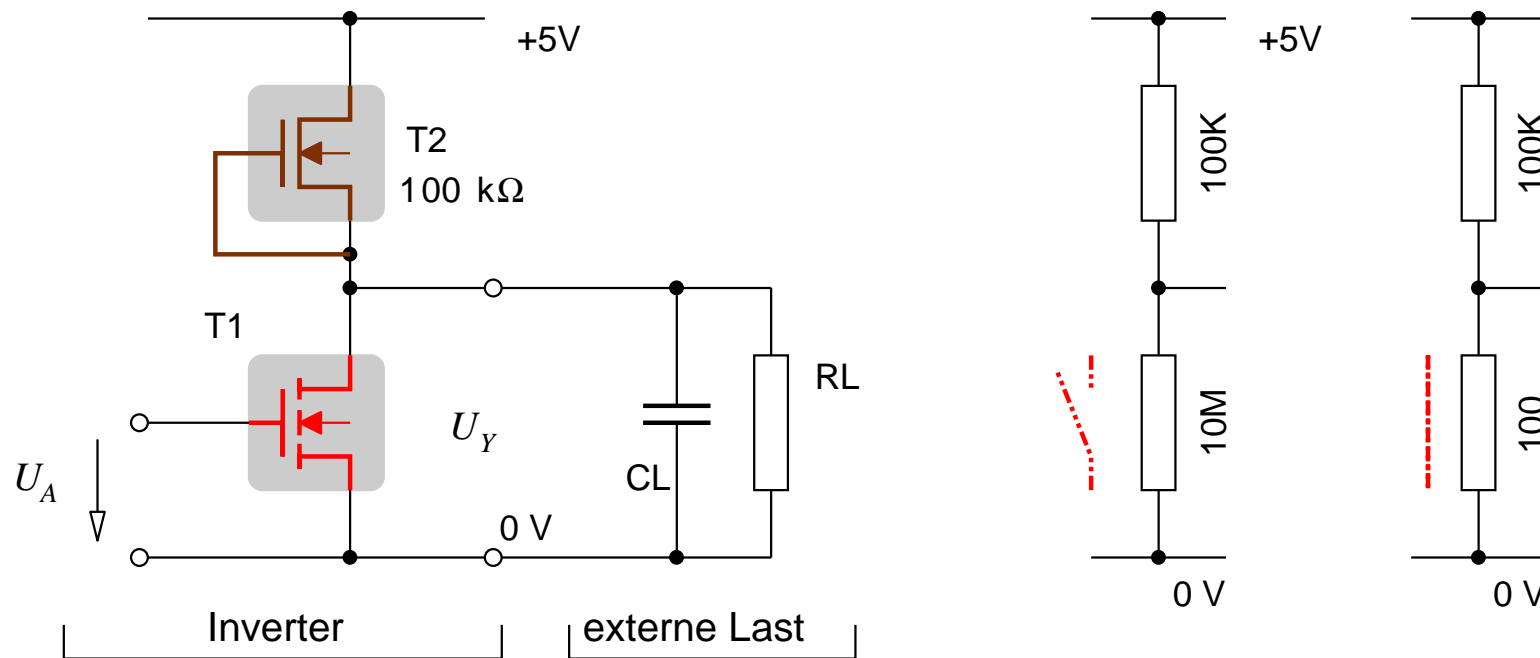
(Mead & Conway 1980)

NMOS: Kaskadierung



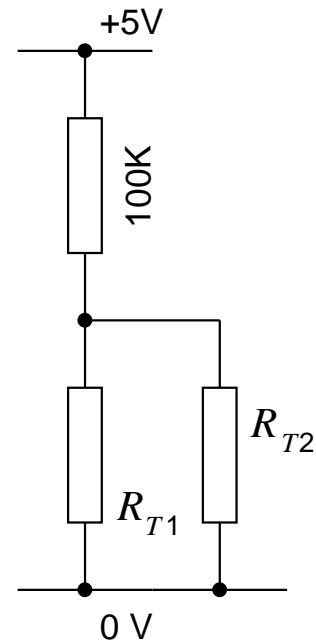
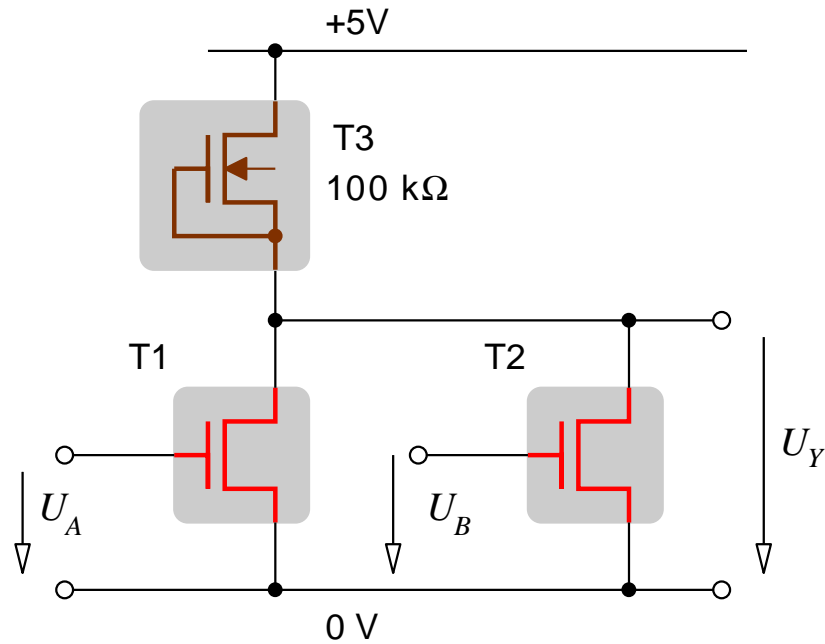
- Inverter liefert Ausgangsspannung 4.95V bzw. 0.005V
=> Hintereinanderschaltung ist direkt möglich
- Entladen (über T1) viel schneller als Aufladen (über R)
- Unterschiedliche Schaltgeschwindigkeiten 0->1 und 1->0

NMOS: Stromverbrauch (Inverter)



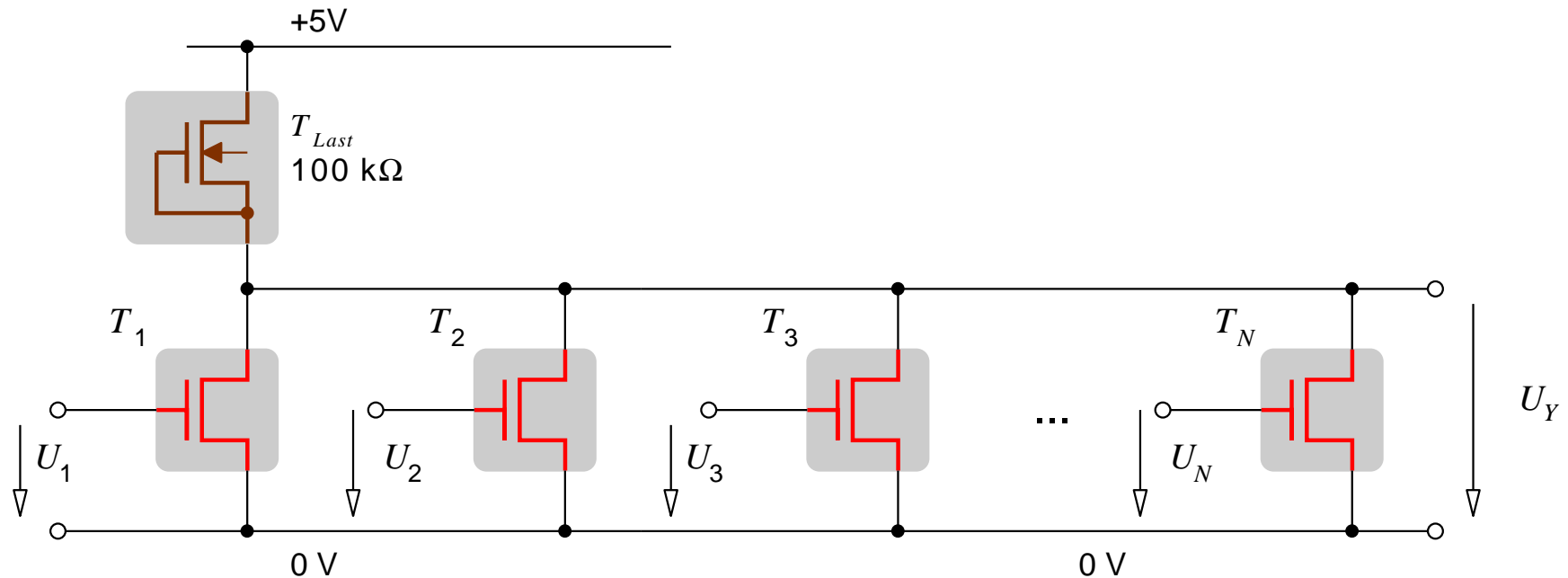
- dynamischer Anteil: Umladen der kapazitiven Last (s.u.)
 - statischer Anteil: über T2 und ohm'sche Last
 - statischer Anteil: über T2 und T1, wenn T1 leitend
- $U = 5V, R_{T2} = 100K\Omega \Rightarrow I = 50\mu A, P = 250\mu W$
- => limitierender Faktor für NMOS-Hochintegration (!)

NMOS: NOR-Gatter



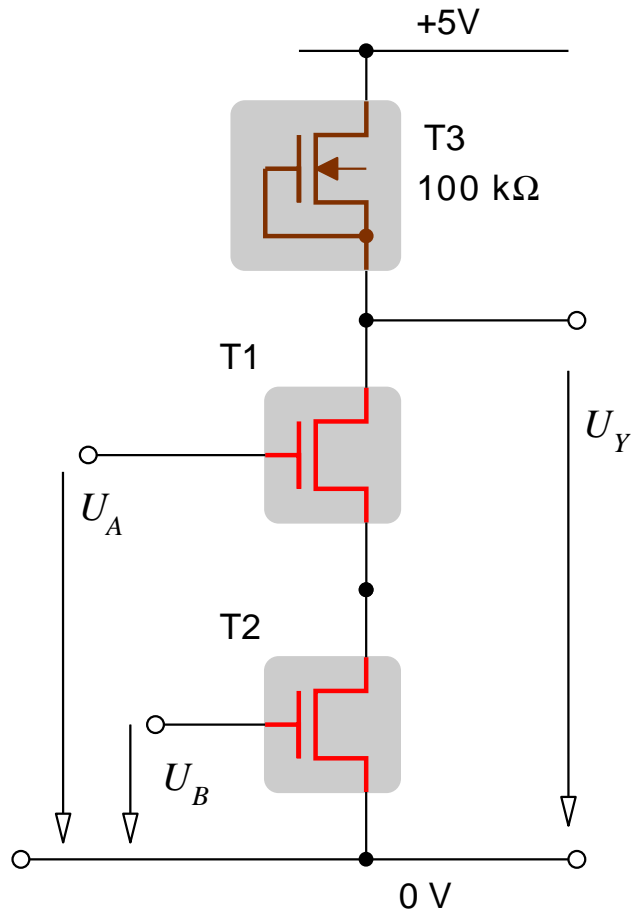
Funktion:					Interpretation:		
U_A	U_B	R_{T1}	R_{T2}	U_Y	A	B	Y
0V	0V	10MΩ	10MΩ	4.90V	0	0	1
0V	5V	10MΩ	100Ω	0.005V	0	1	0
5V	0V	100Ω	10MΩ	0.005V	1	0	0
5V	5V	100Ω	100Ω	0.0025V	1	1	0

NMOS: *n*-input NOR



- Parallelschaltung erlaubt NOR-Gatter mit vielen Eingängen
- sehr kompakt - Anwendung auch in ROMs und PLAs
- wenn mindestens ein Eingang '1' ist, ist der Ausgang auf '0'
- aber: Strom fließt, während mindestens ein Eingang '1' ist

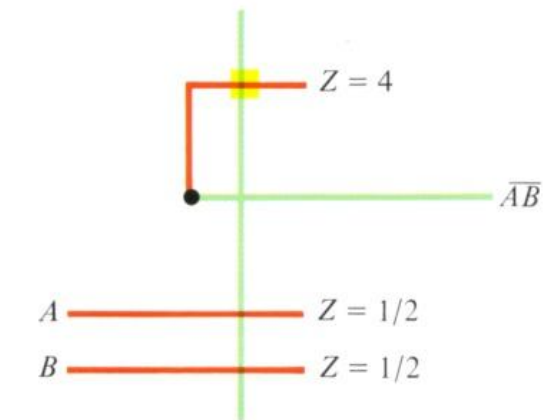
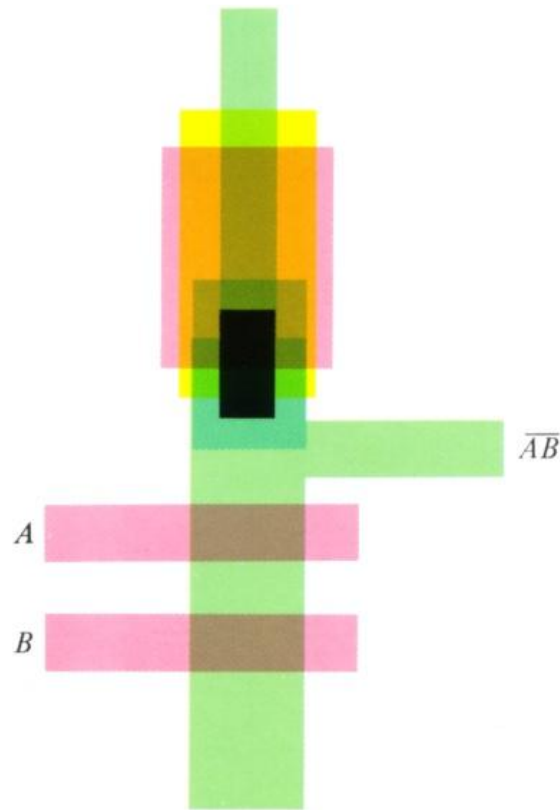
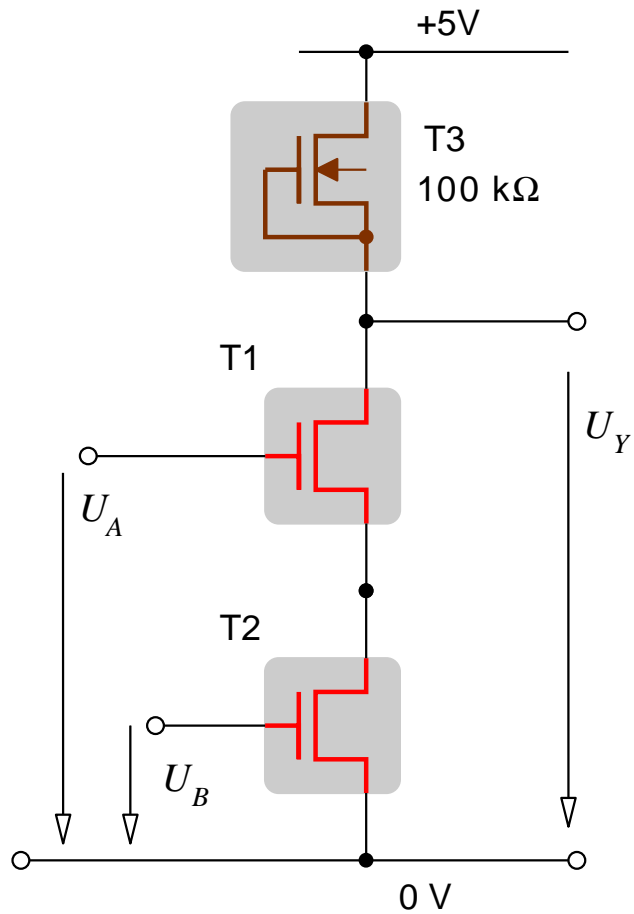
NMOS: NAND-Gatter



Reihenschaltung der Schalt-Transistoren:

Funktion:					A	B	Y
U_A	U_B	R_{T1}	R_{T2}	U_Y			
0V	0V	10M Ω	10M Ω	4.90V	0	0	1
0V	5V	10M Ω	100 Ω	4.90V	0	1	1
5V	0V	100 Ω	10M Ω	4.90V	1	0	1
5V	5V	100 Ω	100 Ω	0.005V	1	1	0

NMOS: NAND-Gatter Layout

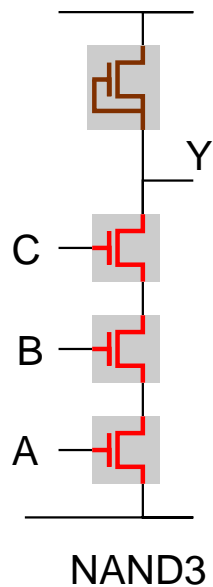
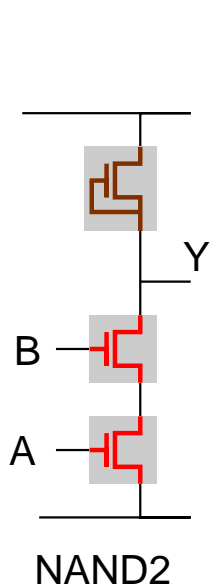
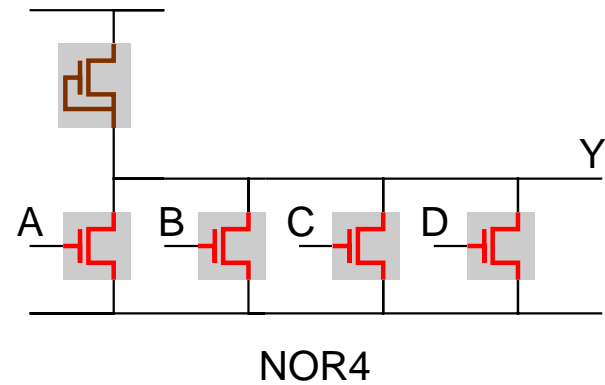
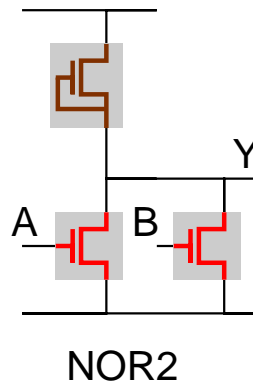
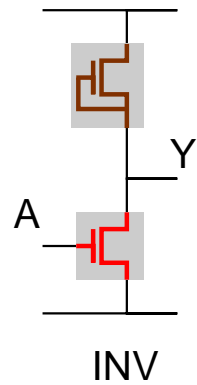
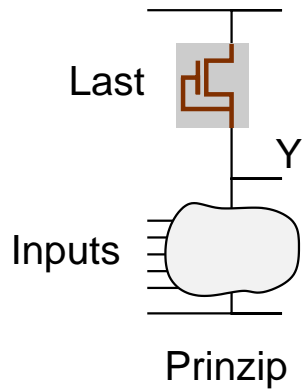


(a) NAND gate layout geometry.

(b) NAND gate topology (stick diagram).

Trick: Verwendung zweier Gates hintereinander
(spart Platz gegenüber zwei separaten Transistoren)

NMOS: Übersicht



NOR: Schalttransistoren parallel
NAND: Schalttransistoren in Reihe

Lasttransistor dient nur als Widerstand

kompliziertere Funktionen aus diesen Grundkomponenten aufbauen

CMOS-Technologie

"CMOS" := complementary metal-oxide semiconductor

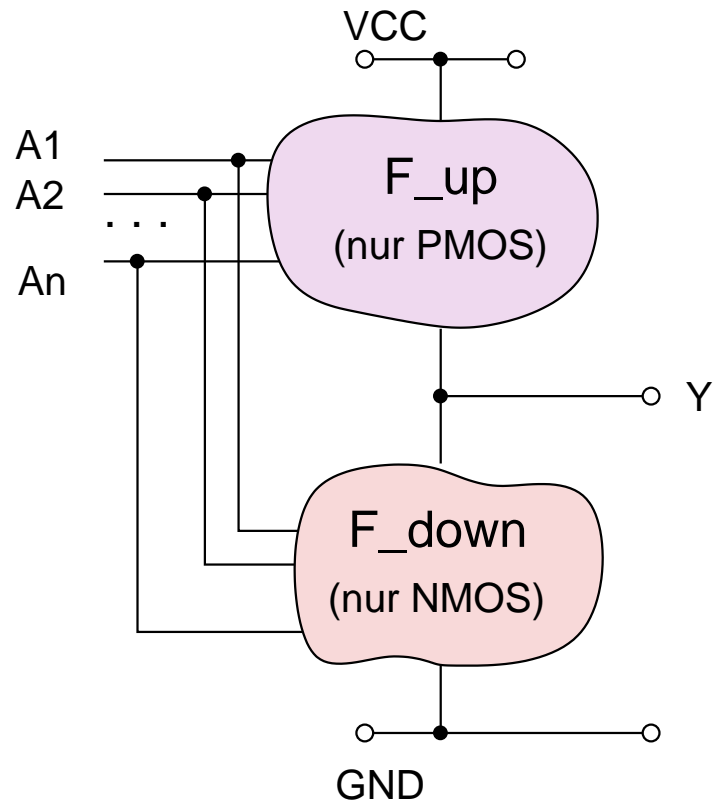
- verwendet NMOS- und PMOS-Transistoren
- aufwendiger Herstellungsprozess
- etwas höherer Platzbedarf als NMOS

komplementäre Schaltungstechnik ("static CMOS"):

- zueinander duale Funktionen für NMOS- und PMOS-Transistoren
- kein statischer Stromverbrauch
- hohe Störsicherheit
- symmetrisches Schaltverhalten

=> ideal für Hochintegration

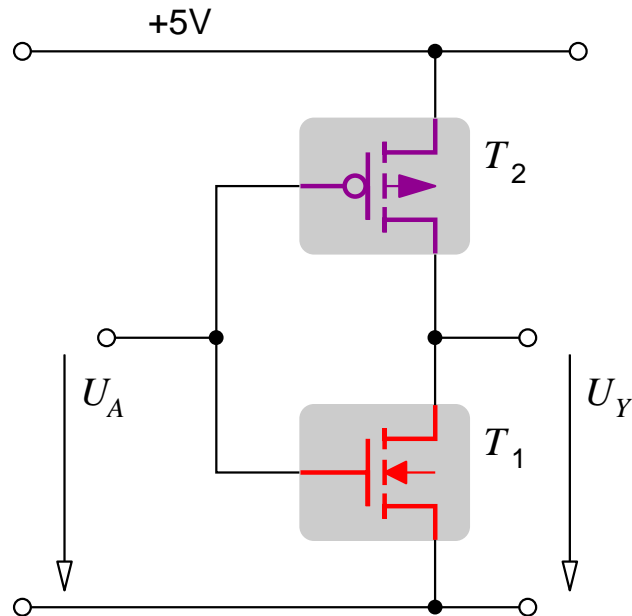
CMOS: Prinzip



"statische" CMOS-Gatter:

- zwei Teilfunktionen F_up/F_down
- F_up zwischen VCC und Y
nur PMOS-Transistoren
- F_down zwischen Y und GND
nur NMOS-Transistoren
- Eingänge A1..An gehen
an beide Teilfunktionen
- $F_{up} = \text{NOT}(F_{down})$

CMOS: Inverter



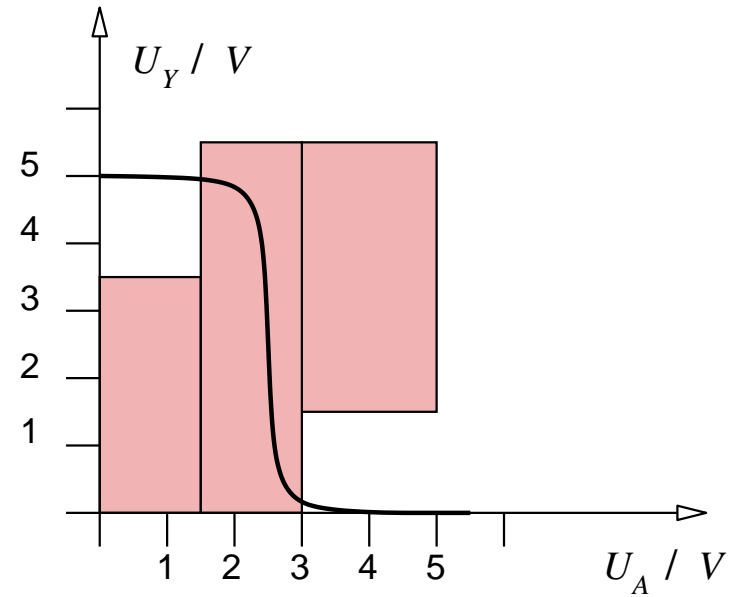
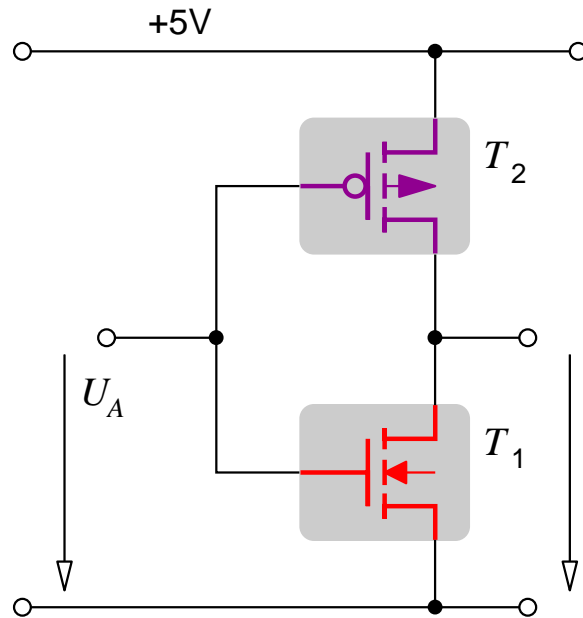
Funktion:				
U_A	T_1	T_2	A	Y
0V	S	L	0	1
5V	L	S	1	0

L = leitet, S = sperrt

- entweder T1 oder T2 leitet
- der jeweils andere Transistor ist dann gesperrt
- passende Dimensionierung erlaubt symmetrisches Verhalten
- d.h. T2 genauso schnell und stark wie T1

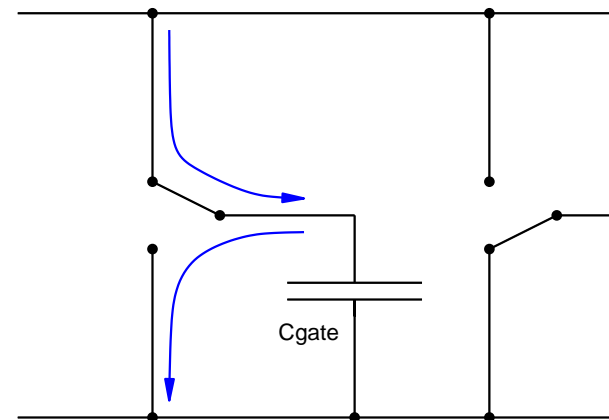
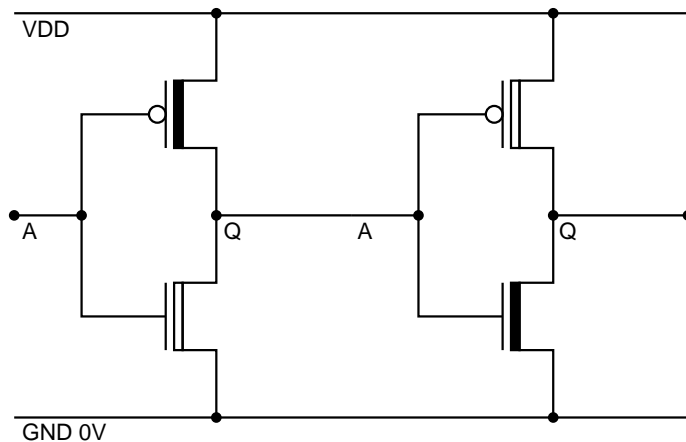
(typisch: T2 dreimal so gross wie T1, weil Löcher langsamer als Elektronen)

CMOS: Inverter-Kennlinie



- Kennlinie verläuft steiler als beim NMOS-Inverter
- fast ideales Verhalten
- vgl. Matlab-Simulation für NMOS

CMOS: Leistungsverbrauch



- (fast) kein statischer Stromverbrauch
- Kurzschluss-Strom beim Umschalten
- Umladen der Gate-Kapazität dominiert, also:

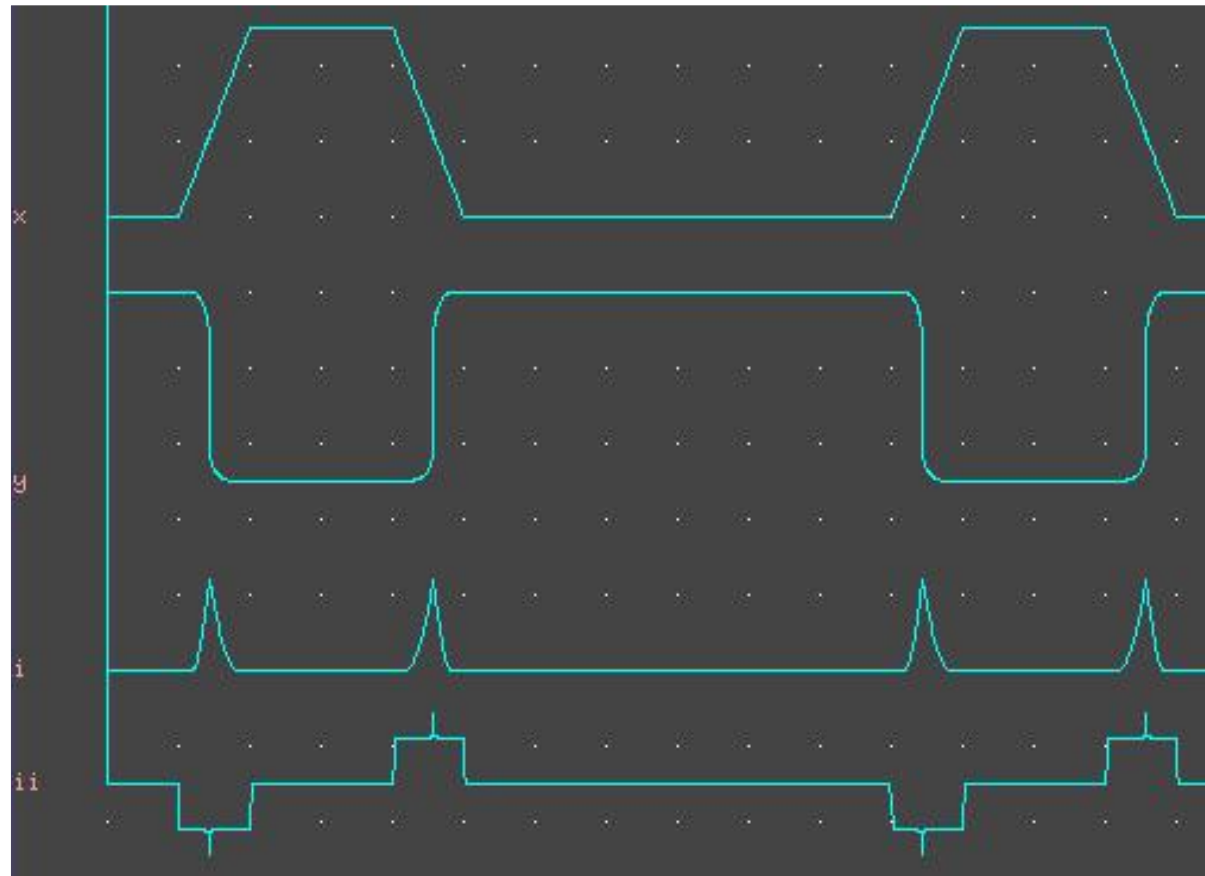
$$Q = C \cdot U$$

$$I = dQ/dt \sim f \cdot C_{\text{gate}} \cdot U$$

$$P = U \cdot I \sim f \cdot U^2$$

proportional zur Taktfrequenz

CMOS: Inverter-Stromverbrauch



Eingangsspannung

U_A

Ausgangsspannung

U_Y

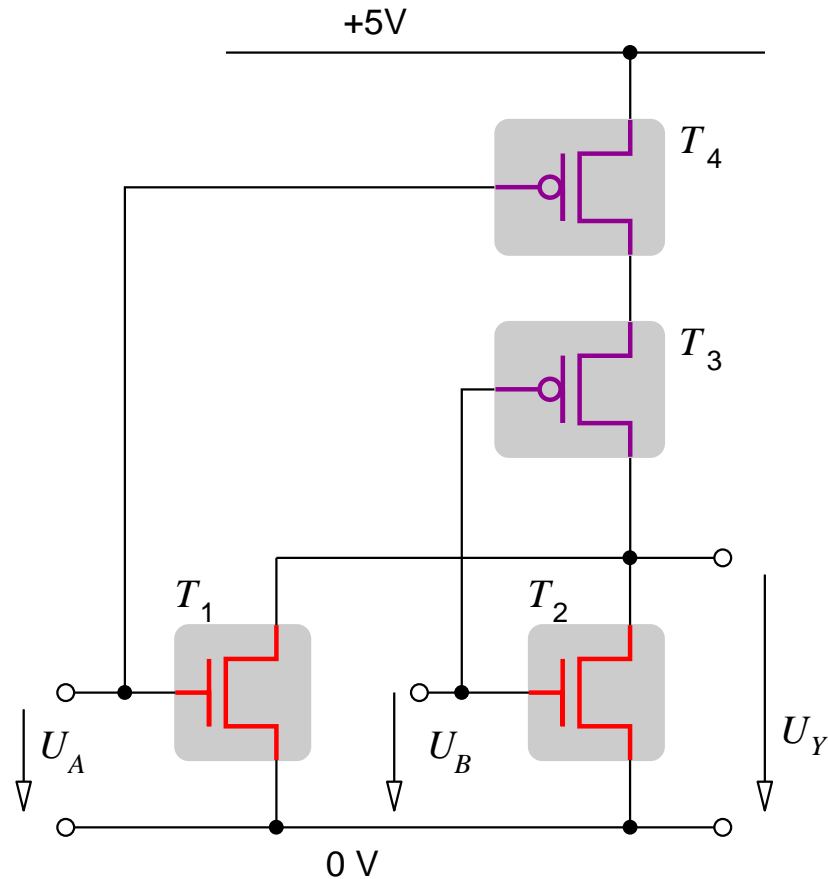
$I_{Kurzschluss}$

I_{gesamt}

(AnaLOG Simulation)

- relativer Anteil des Kurzschluss-Stroms hier recht gering
- abhängig von Dotierung und Dimensionierung der Transistoren

CMOS: NOR-Gatter



- interaktive Demo

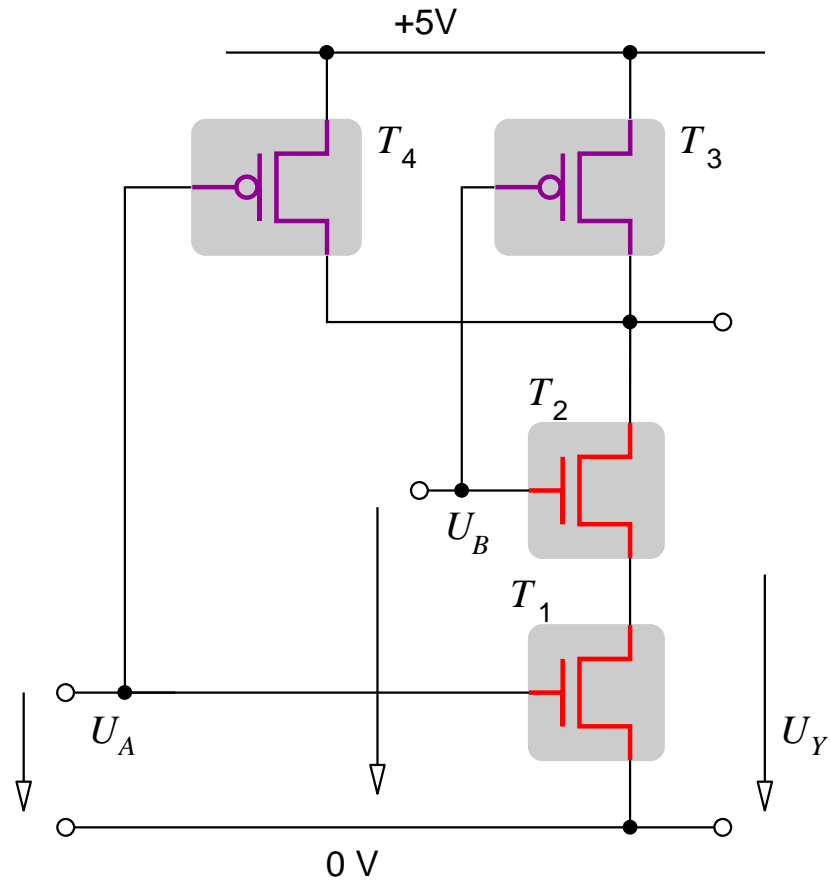
Funktion:						A	B	Y
U_A	U_B	T_1	T_2	T_3	T_4			
0V	0V	S	S	L	L	0	0	1
0V	5V	S	L	S	L	0	1	0
5V	0V	L	S	L	S	1	0	0
5V	5V	L	L	S	S	1	1	0

L = leitet, S = sperrt

N-Transistoren parallel
P-Transistoren in Reihe

- leitender Pfad von Y zu VCC/GND
- kein leitender Pfad von VCC zu GND
- kein statischer Stromverbrauch

CMOS: NAND-Gatter



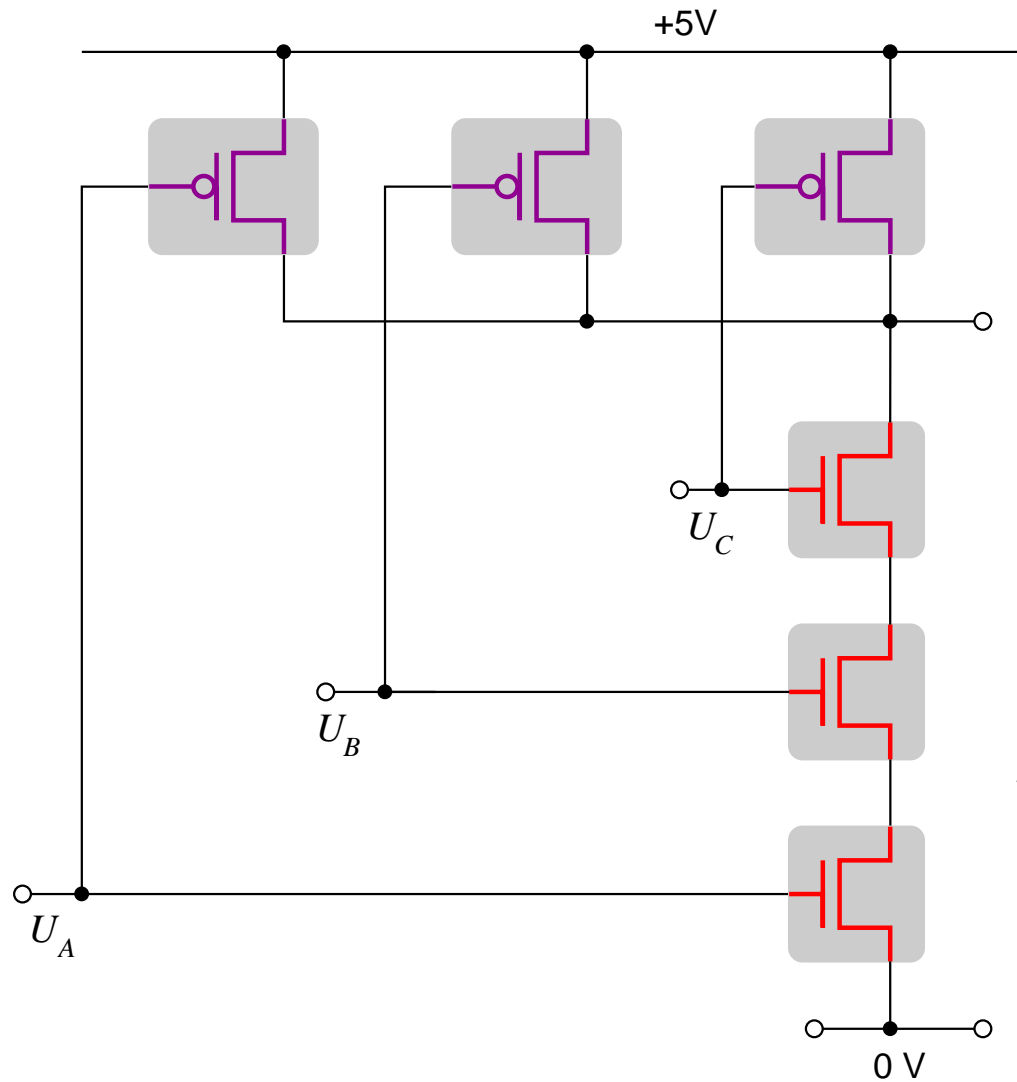
Funktion:									
U_A	U_B	T_1	T_2	T_3	T_4	A	B	Y	
0V	0V	S	S	L	L	0	0	1	
0V	5V	S	L	S	L	0	1	1	
5V	0V	L	S	L	S	1	0	1	
5V	5V	L	L	S	S	1	1	0	

L = leitet, S = sperrt

N-Transistoren in Reihe
P-Transistoren parallel

- interaktive Demo

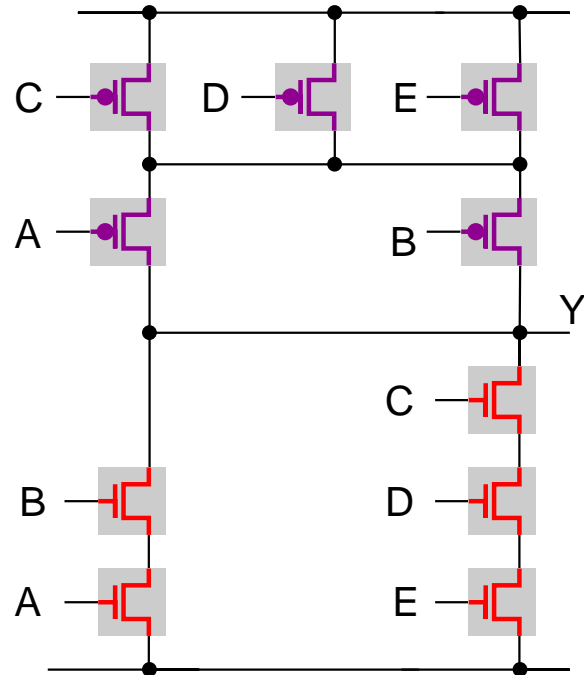
CMOS: 3-input NAND



N-Transistoren in Reihe
P-Transistoren parallel

- Demo

CMOS: Komplexgatter



für bestimmte Funktionen, z.B.:

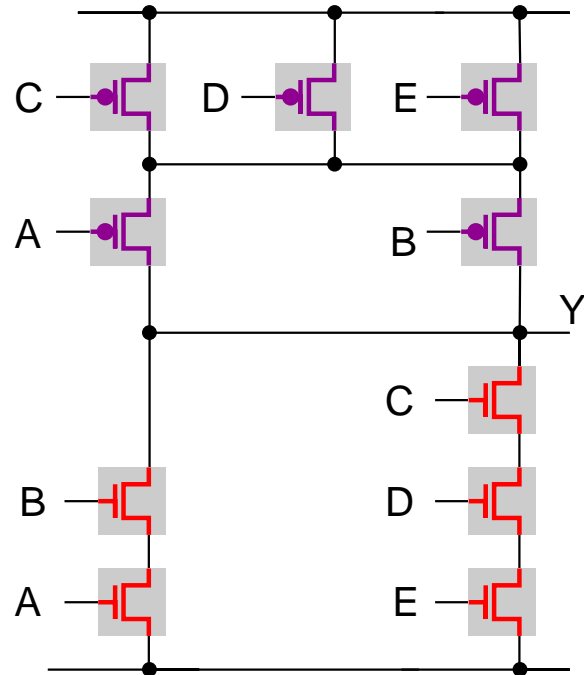
$$Y = ! (AB | CDE)$$

Konstruktion:

- UND-Terme als Reihenschaltung von N-Transistoren (F_down)
- ODER-Term als Parallelschaltung der UND-Terme
- duale Funktion für F_up (PMOS)

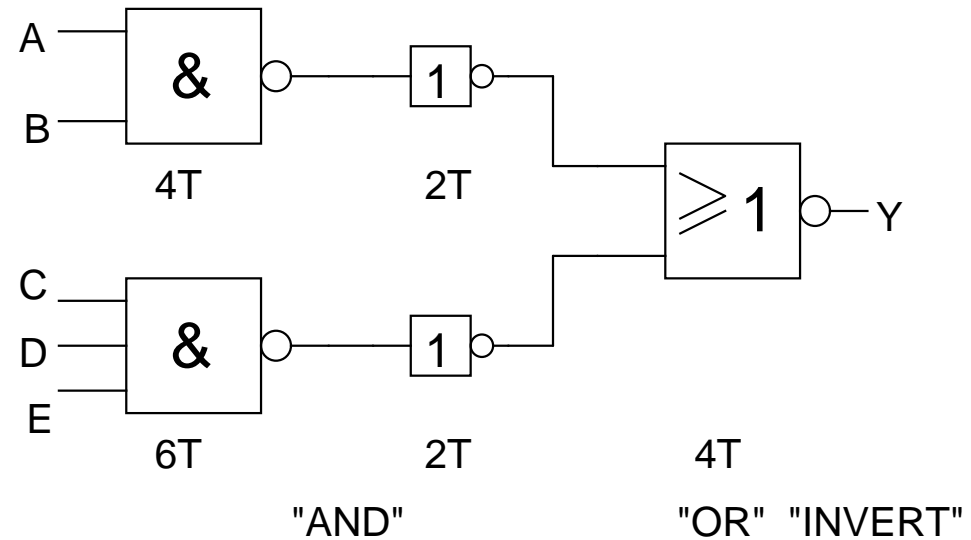
- Beispielfunktion heisst "AOI32": AND3 AND2 OR INVERT
- entsprechend auch OAI (or-and-invert)
- aus elektr. Gründen nicht mehr als max. 4 Transistoren in Reihe

CMOS: Komplexgatter



10 Transistoren

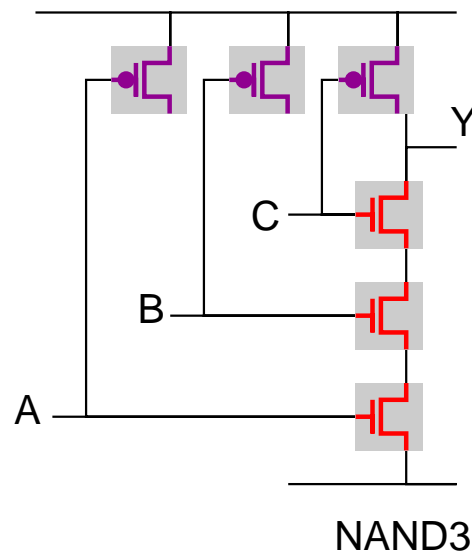
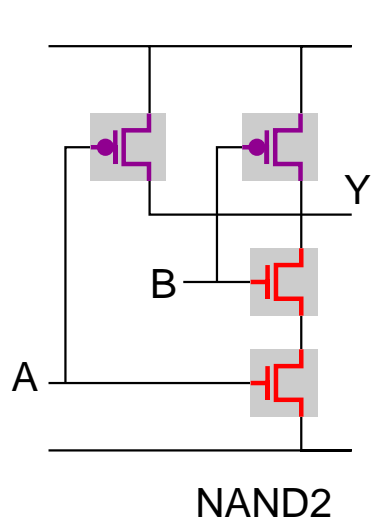
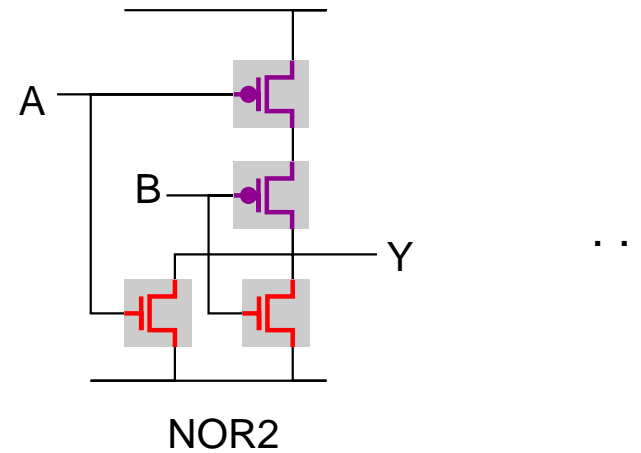
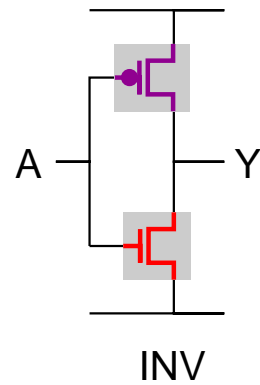
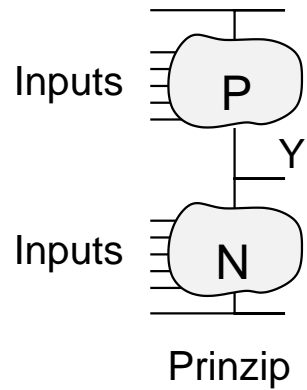
$$Y = ! (AB | CDE)$$



gesamt 18 Transistoren

- AOI-Komplexgatter vs. Aufbau mit einzelnen Gattern
- Komplexgatter kleiner und schneller
- werden beim Schaltungsentwurf oft benutzt

CMOS: Übersicht



elementare Gatter:

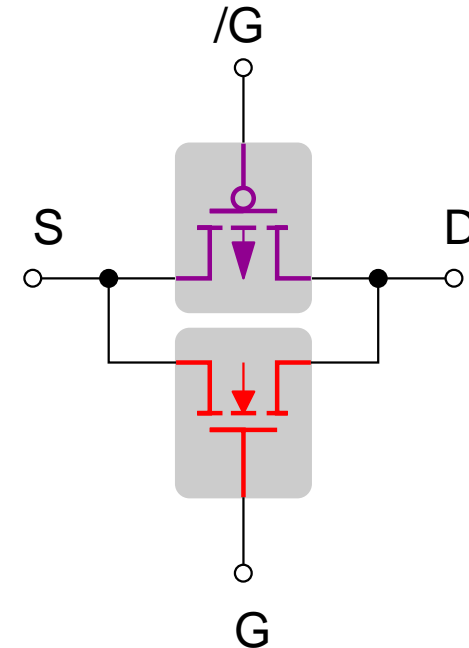
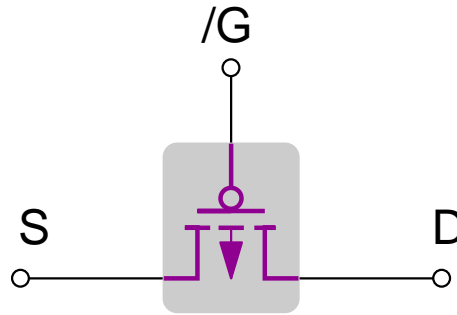
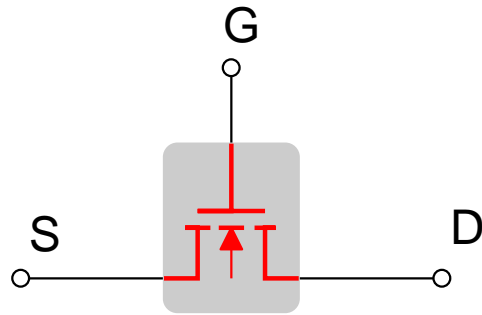
Inverter

NOR2, NOR3, (NOR4)

NAND2, NAND3, NAND4

AOI/OAI Komplexgatter

CMOS: *pass transistor, t-gate*

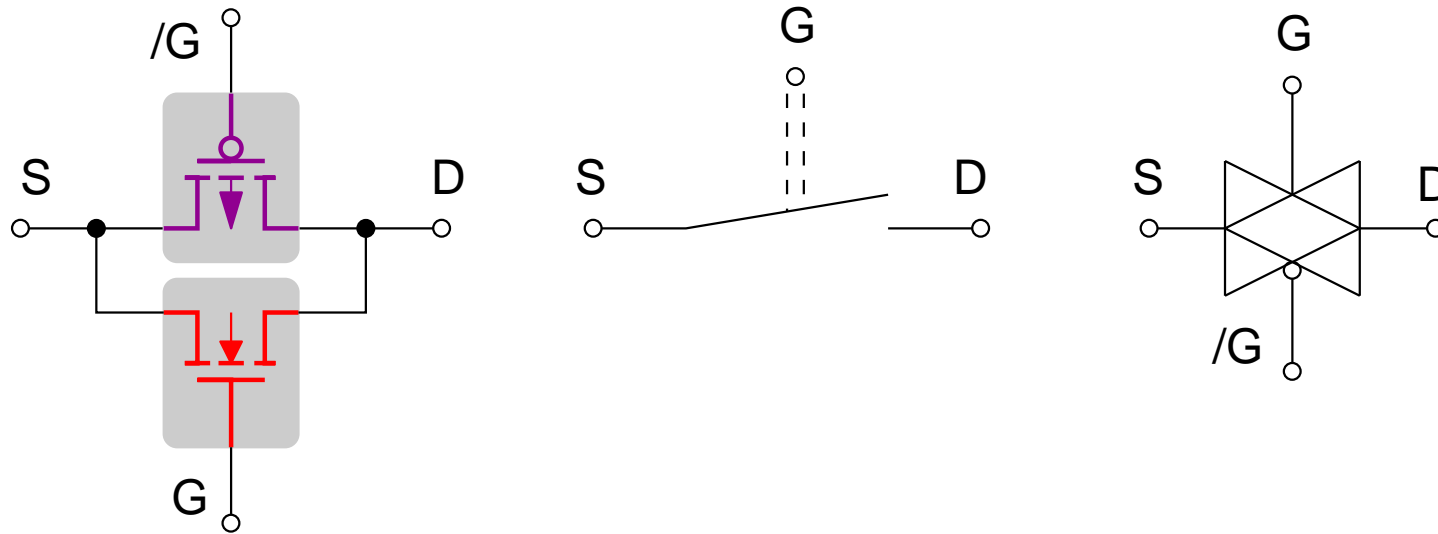


U_S	U_G	U_D
0V	0V	sperrt
0V	5V	0V (leitet)
5V	0V	sperrt
5V	5V	$5V - U_{TH}$

U_S	U_G	U_D
0V	0V	U_{TH}
0V	5V	sperrt
5V	0V	5V (leitet)
5V	5V	sperrt

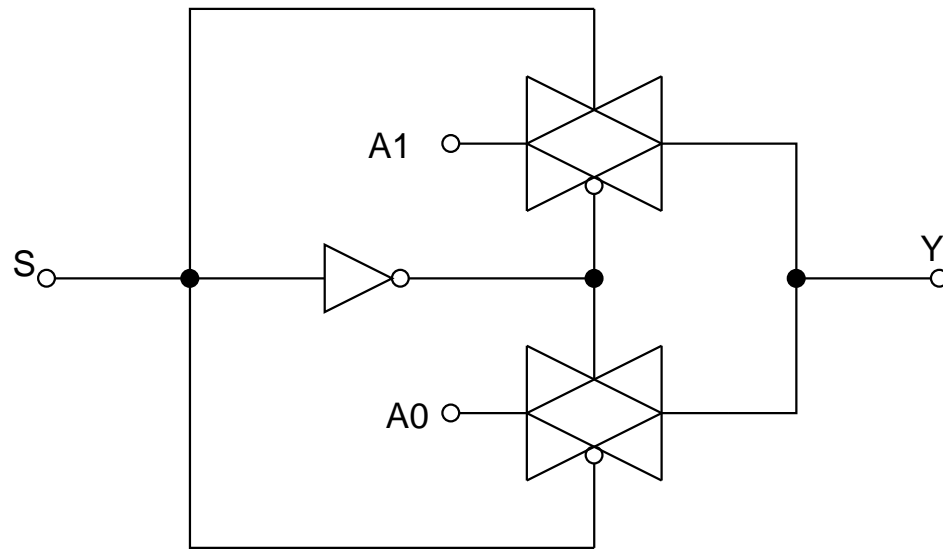
- NMOS Transistor leitet 0V gut, aber 5V nur schwach
 - PMOS Transistor leitet 5V gut, aber 0V nur schwach
- => Kombination beider Transistoren zum "transmission gate"
(Achtung: Ansteuerung mit inversen Pegeln G, /G)

CMOS: Funktion des t-gate



- $G=0, /G=1$: beide Transistoren sperren, Schalter offen
- $G=1, /G=0$: beide Transistoren leiten, Schalter geschlossen
- bei Bedarf zusätzlicher Inverter zum Erzeugen von $/G$ aus G
- erlaubt effiziente Realisierung bestimmter Funktionen

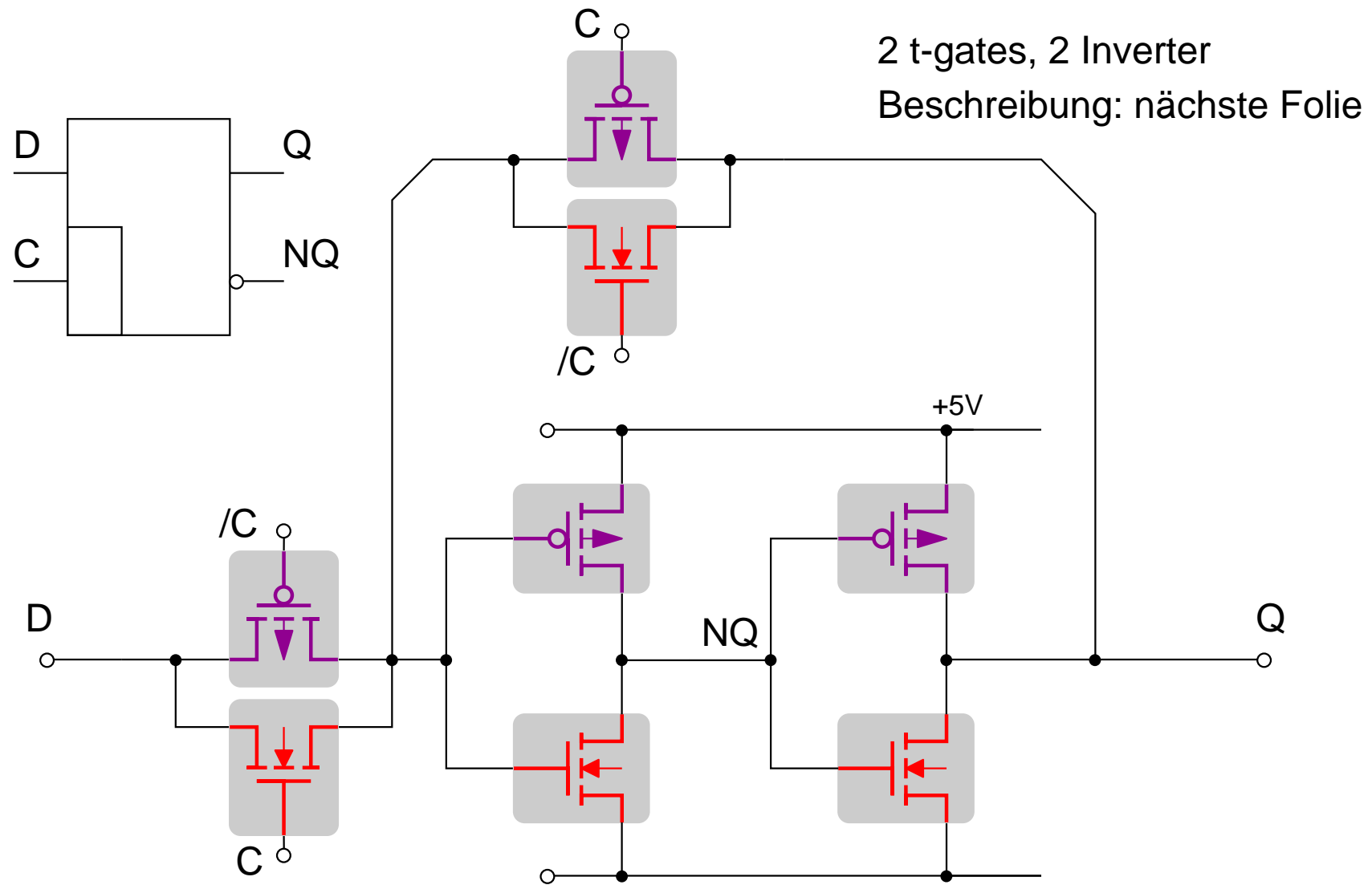
CMOS: Multiplexer mit t-gates



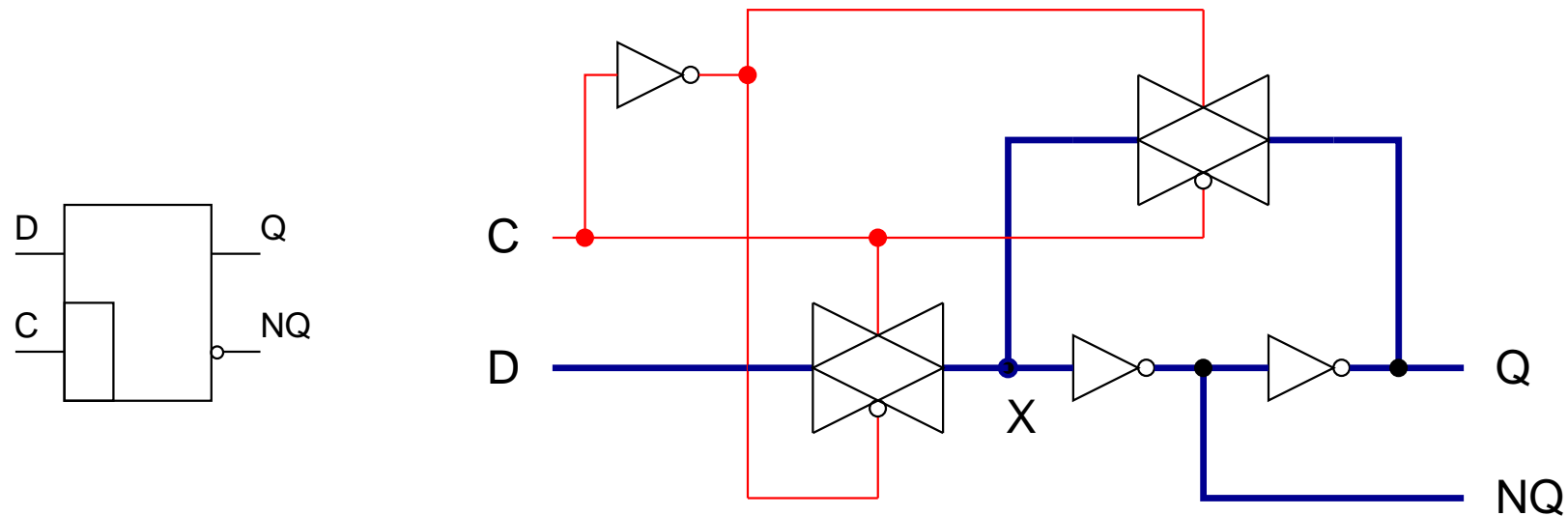
S	A1	A0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- t-gates erlauben effiziente Realisierung von Multiplexern
- Beispiel 2:1 Multiplexer: zwei t-gates, ein Inverter (6 Transistoren)
- aber: die Eingangssignale (A1,A0) werden nicht verstärkt
- solche Schaltungen können nicht beliebig verkettet werden

CMOS: D-Latch mit t-gates



CMOS: D-Latch

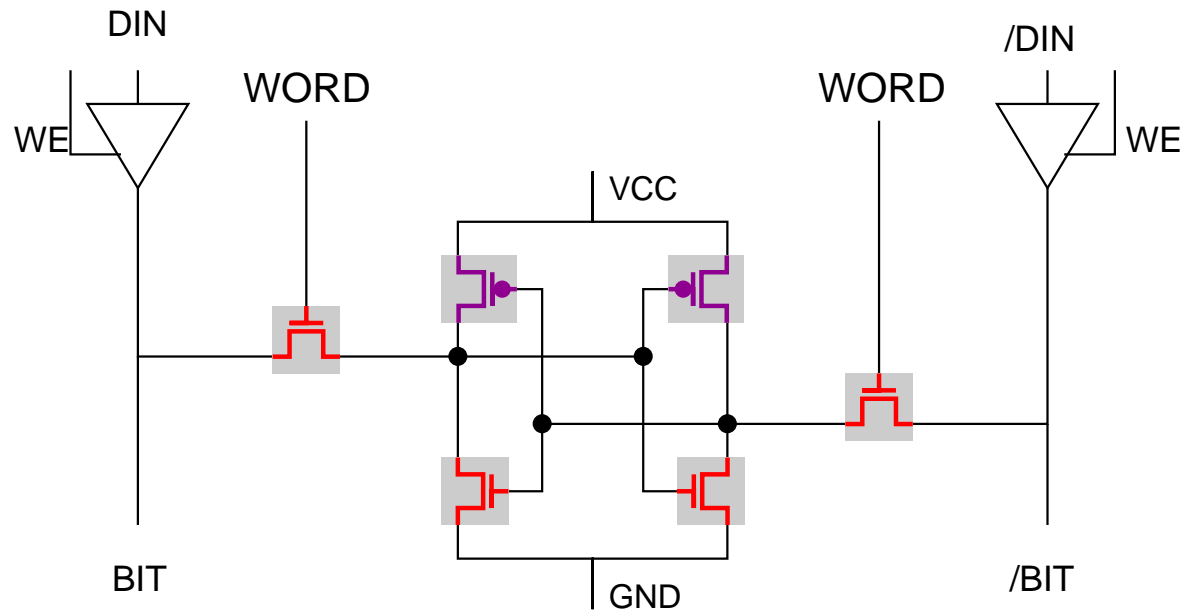


C=1: erstes t-gate leitet, direkter Pfad von D über X und NQ nach Q

C=0: erstes t-gate offen, aber Rückkopplung von Q nach X, der Wert von X wird daher gespeichert

- t-gate Latch: 8 Transistoren
2 Transistoren zur Erzeugung von \overline{C}
- statisches Latch: 4 NAND2-Gatter, 16 Transistoren

CMOS: 6T-SRAM Zelle



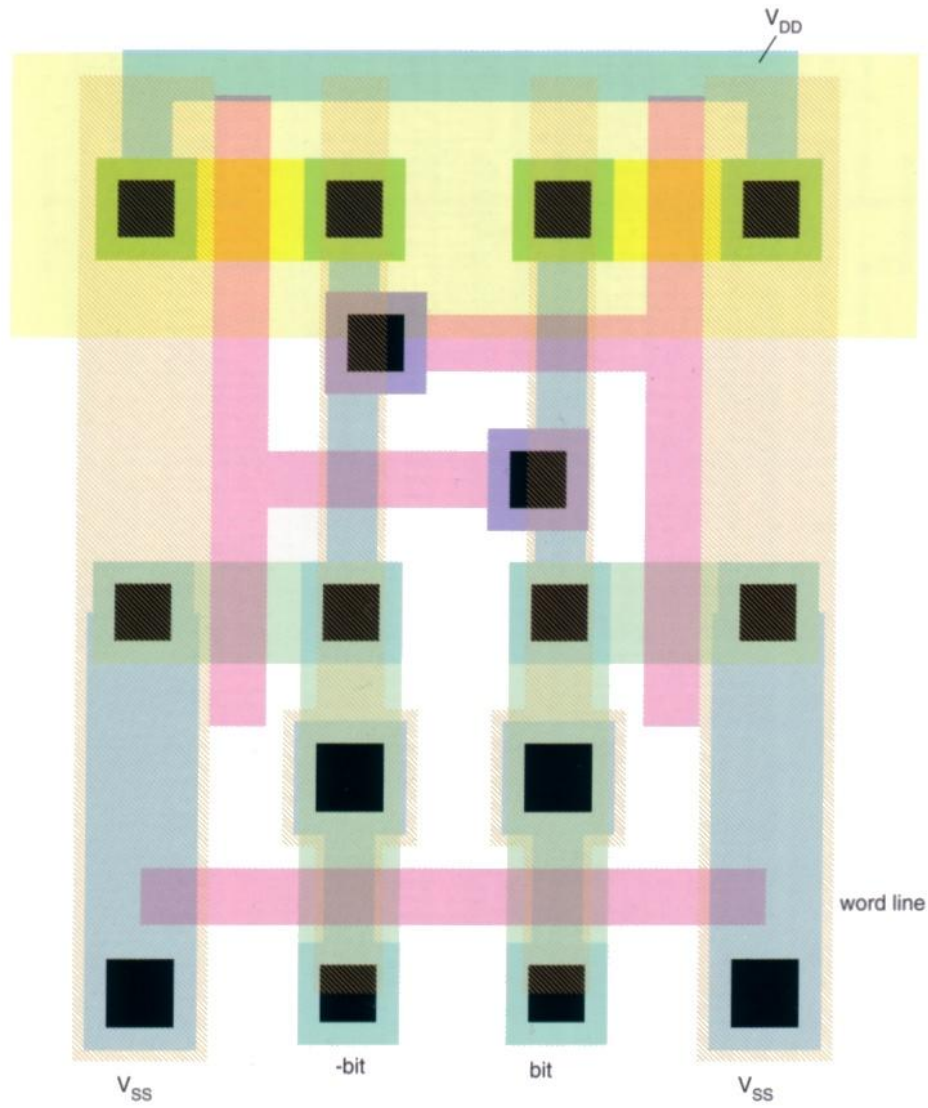
- zwei hintereinandergeschaltete Inverter zum Speichern
- einfache NMOS pass-Transistoren statt t-gates

$W = 0$: pass-Transistoren offen, Zelle speichert

$W = 1, WE=0$: Bitline-Treiber passiv: Daten auslesen

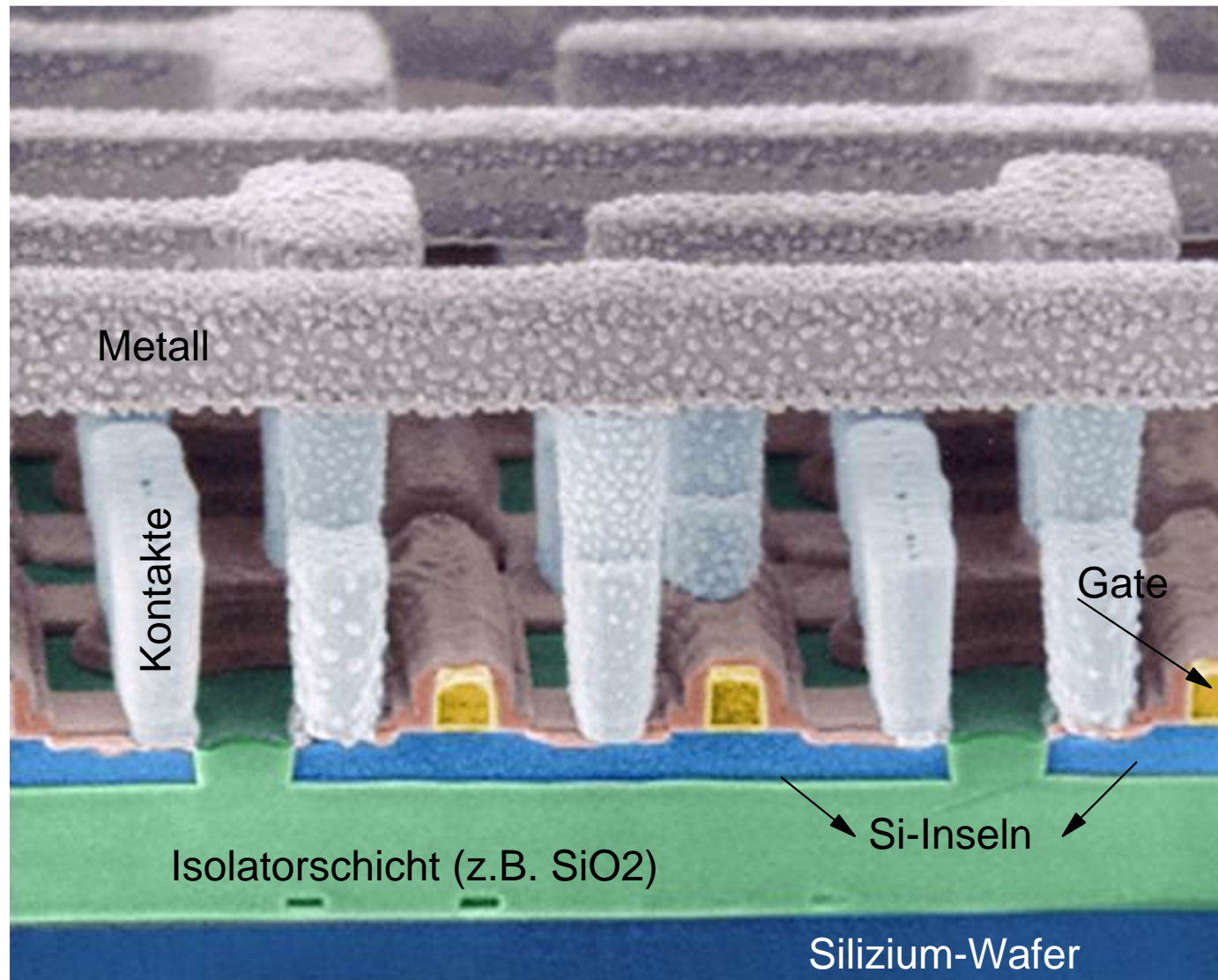
$W = 1, WE=1$: Bitline-Treiber aktiv: neuen Datenwert schreiben

6T SRAM: *Layout*



(6T SRAM layout aus Weste/Eshragian, 1983)

CMOS: SOI "silicon on insulator"



(IBM 2001)

Zusammenfassung

- Boole'sche Algebra
- MOS-Transistor als Schalter

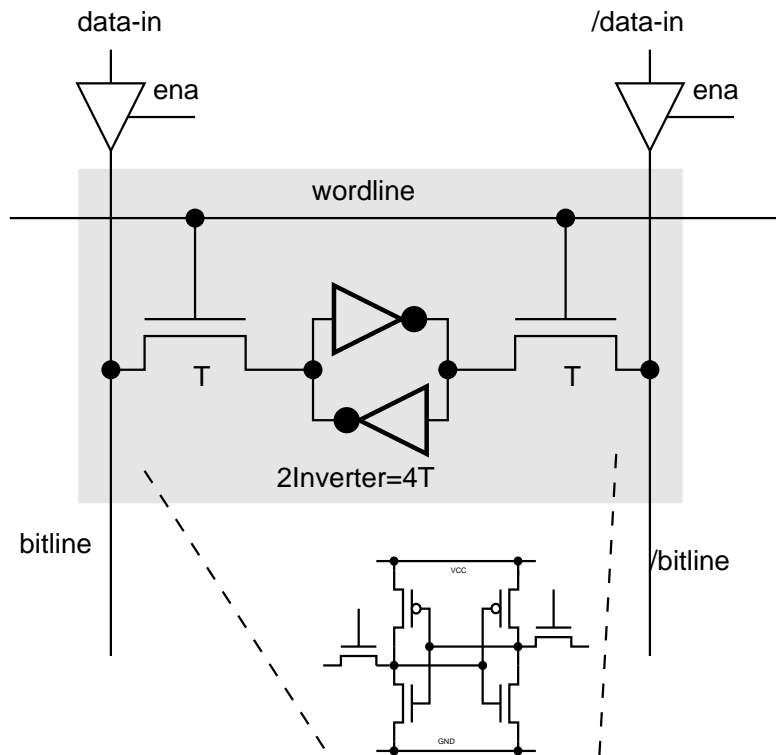
Gatter in NMOS-Technologie

- Inverter
- Schalt- und Last-Transistor
- NOR- und NAND-Gatter

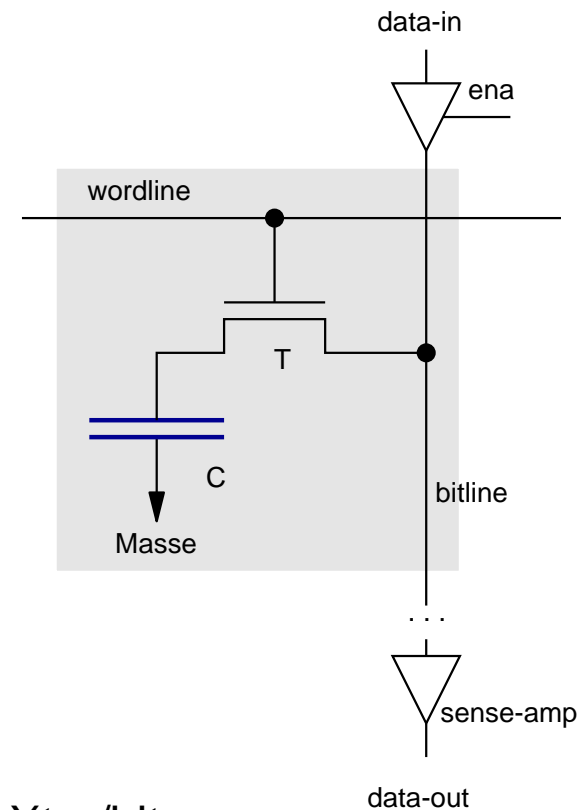
Gatter in CMOS-Technologie

- Inverter, NOR, NAND
- Komplexgatter
- "transmission gates"
- Flipflop, SRAM- und DRAM-Zellen

DRAM vs. SRAM

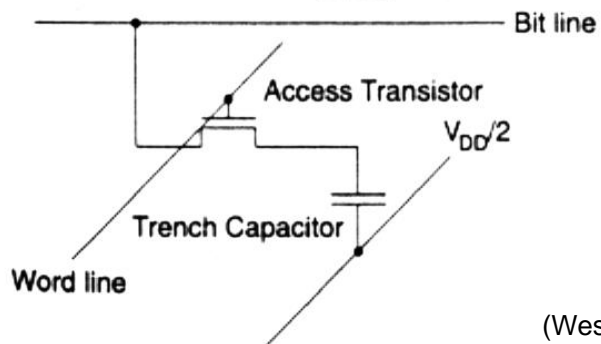
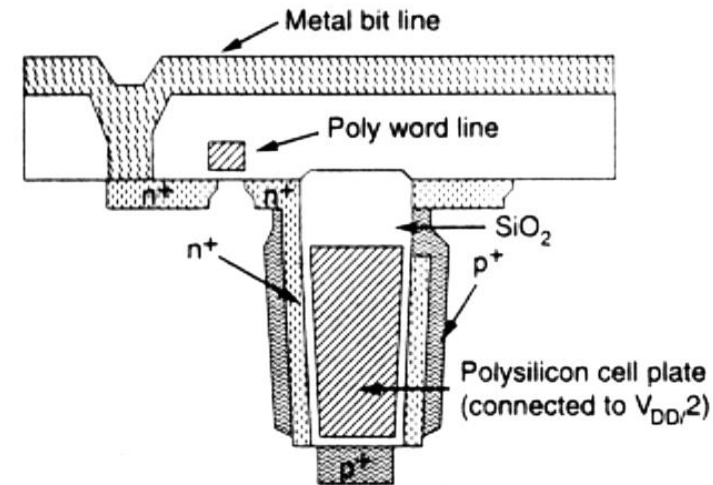


- 6 Xtors/bit
- statisch (kein refresh)
- schnell
- 10 .. 50X DRAM-Fläche



- 1 Xtor/bit
- $C=10\text{fF}$: ~200.000 Elektronen
- langsam (charge-sharing)
- minimale Fläche

DRAM: Trench-Kondensator



(Weste Eshragian)

- Bauform Trench: Platten vertikal am Rand eines Grabens
- Bauform Stack: mehrere horizontale Schichten

DRAM: Stack / Trench-Kondensator

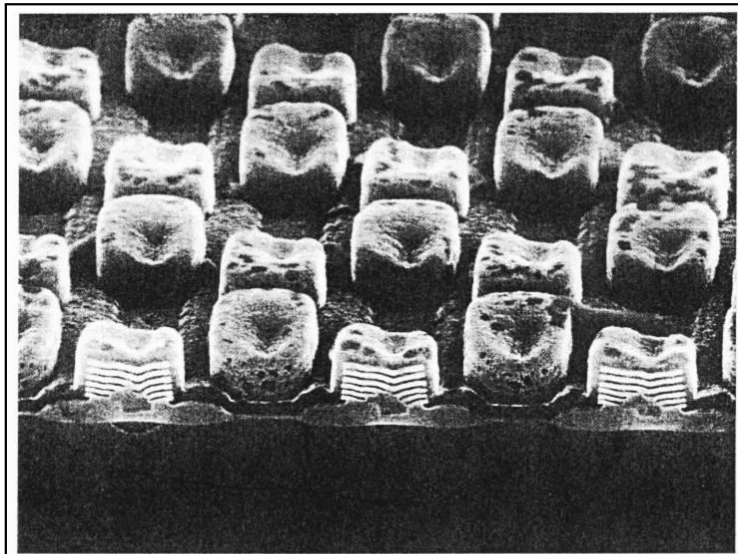
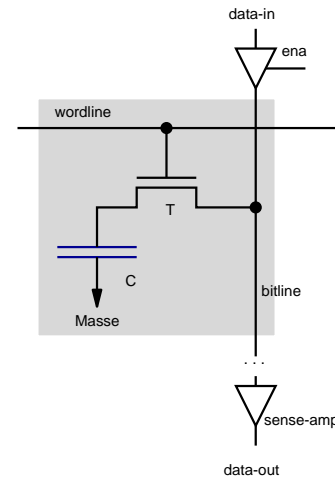


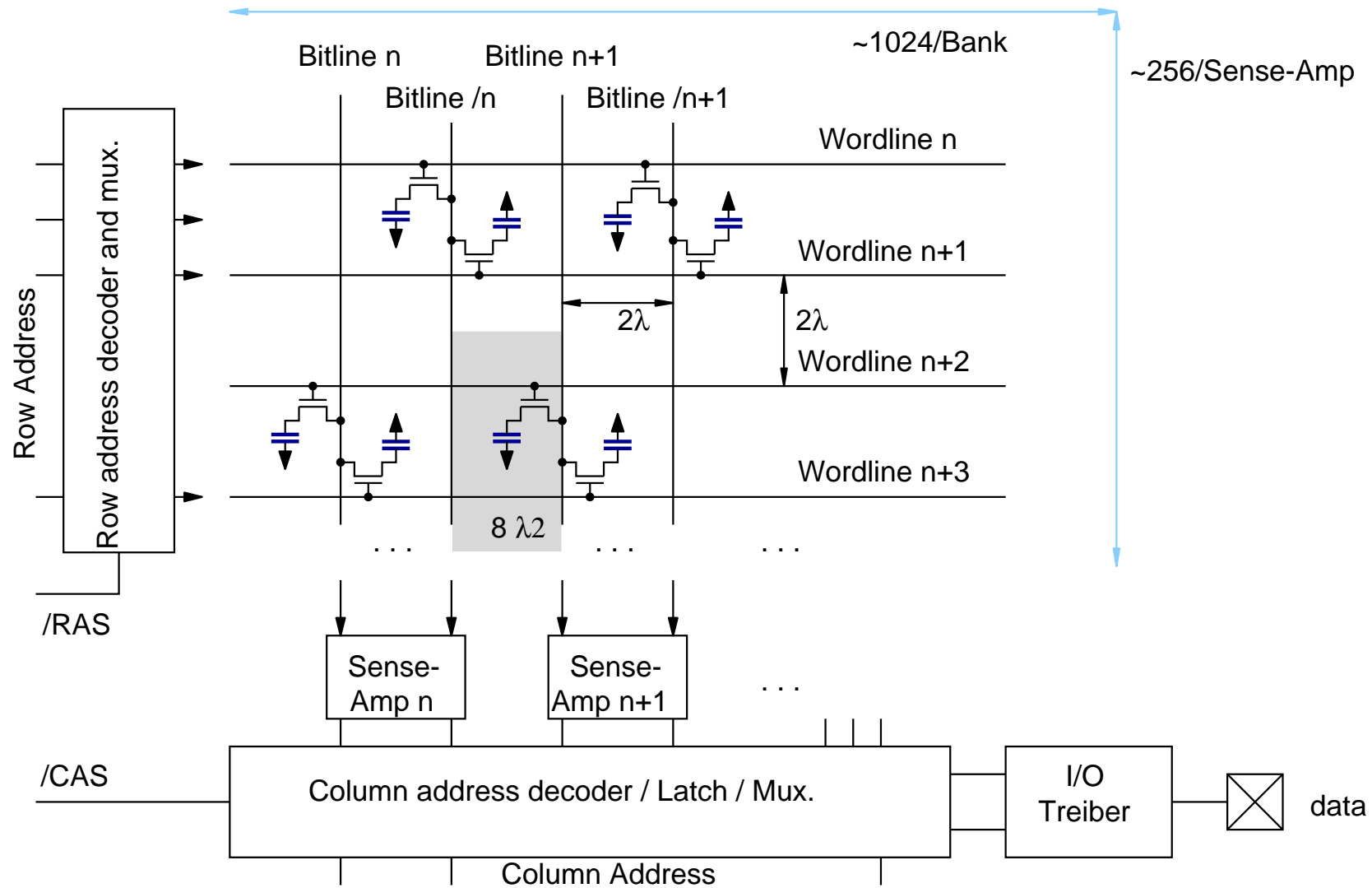
Abb. 7: Prototyp von Speicherzellen (Stapelkondensatoren) für zukünftige Speicherchips wie den Ein-Gigabit-Chip. Da für DRAM-Chips eine minimale Speicherkapazität von 25 fF notwendig ist, bringt es erhebliche Platzvorteile, die Kondensatorelemente vertikal übereinander zu stapeln. Die Dicke der Schichten beträgt etwa 50 nm. (Foto: Siemens)



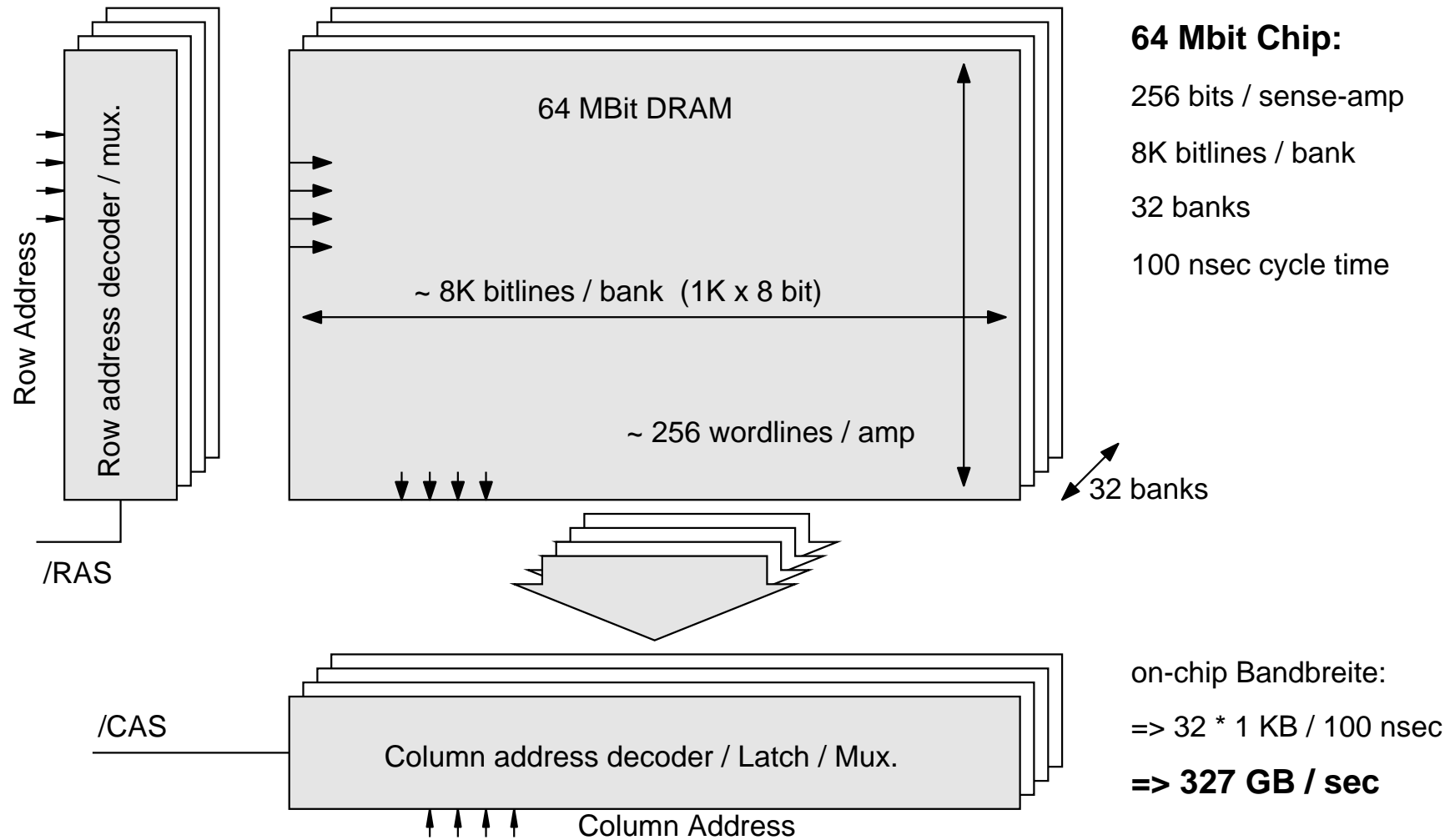
- "stacked capacitors"
(Siemens 1Gb DRAM Prototyp 96)
- $C=10\text{fF}$: ~ 200.000 Elektronen

- "trench capacitors"
(IBM CMOS-6X embedded DRAM)

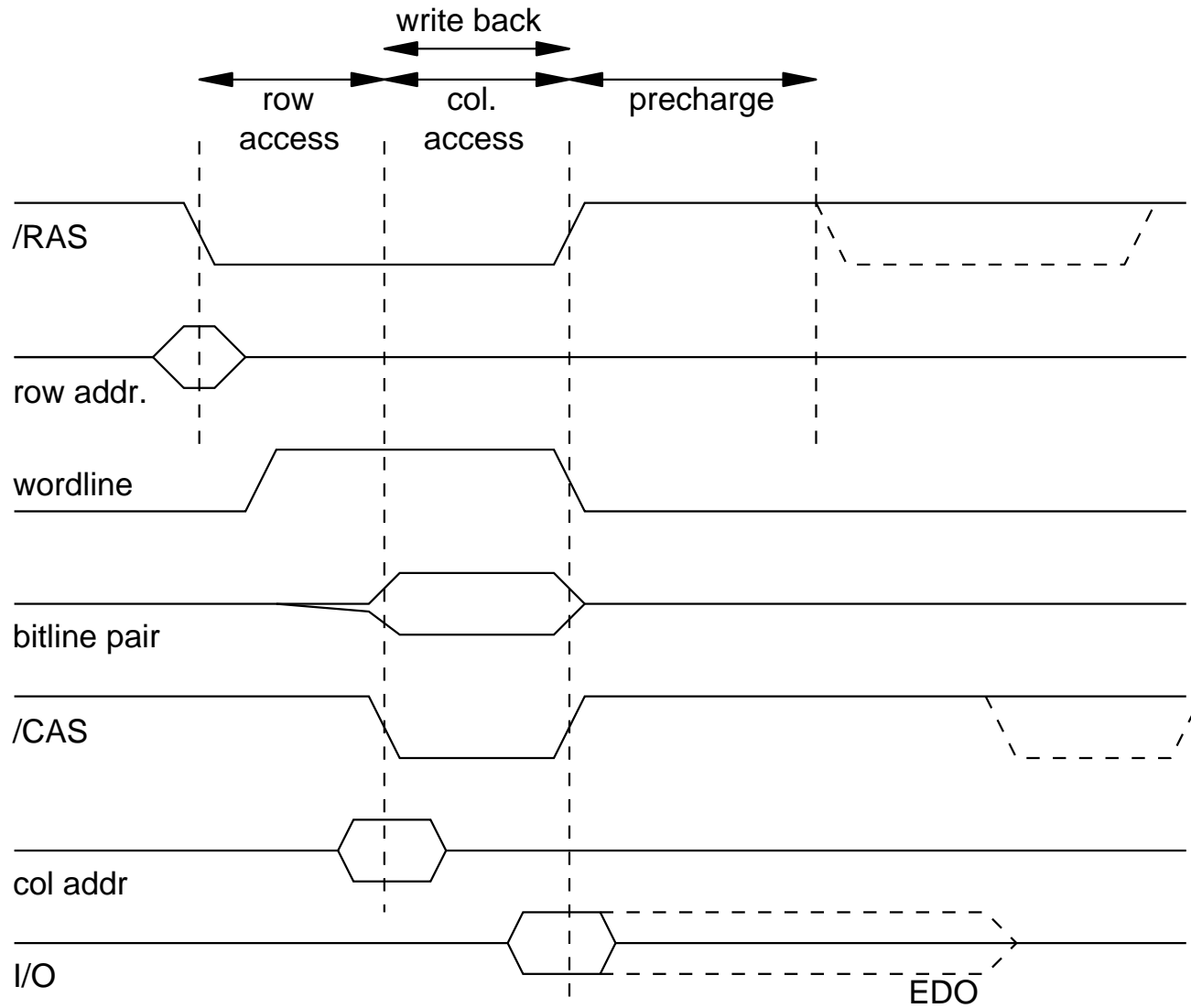
DRAM: Layout



DRAM: Organisation / Bandbreite



DRAM: Ansteuerung (asynchron)



DRAM: Funktion

Read:

- $/\text{RAS} = 0$: Auswahl der Wordline, Aktivierung der Bitlines
Auslesen und Auswertung der selektierten Zellen
- $/\text{CAS} = 0$: Auswahl der Bitline, Ausgabe der Daten
Zurückschreiben der gelesenen Daten (!)
- $/\text{RAS} = 1$: Precharge der Bitlines

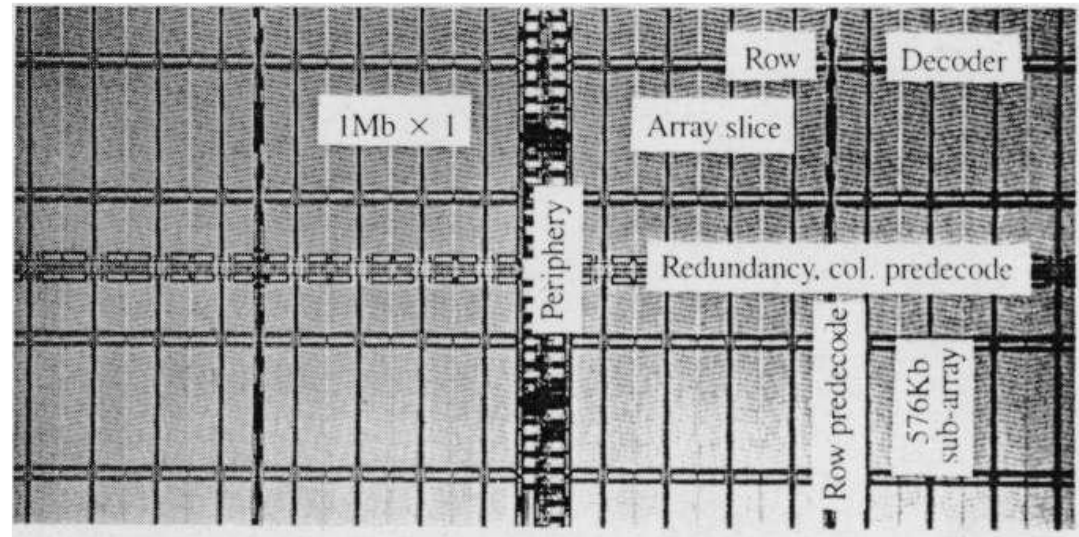
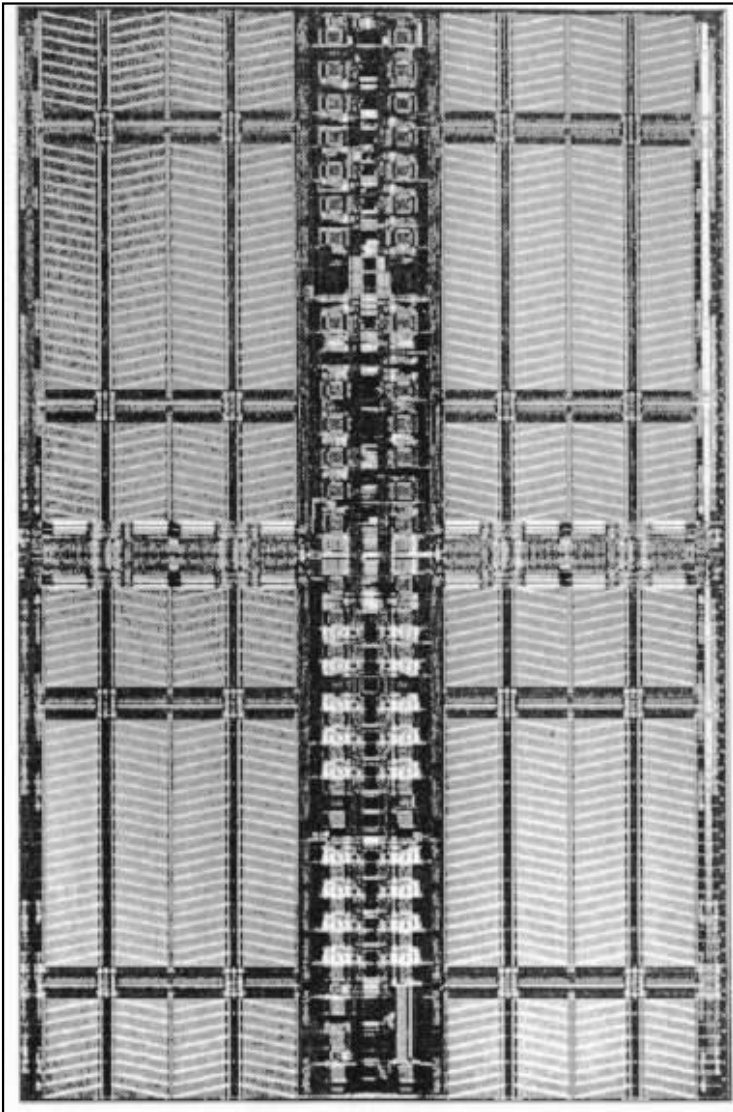
Write:

- $/\text{CAS} = 0$: Zurückschreiben der gelesenen + neuer Daten

SDRAM: zusätzliche Register, diverse Burst-Modi

Refresh: alle 16 .. 32 ms notwendig

DRAM: Floorplan (IBM 4Mbit)



- Größenvergleich zwischen I/O, Col/Row-Decoder, Array
- Konfiguration nach Marktlage links: 4 Mbit, oben: 16 Mbit
- Redundanz für besseren Yield: links: 4.0/4.5 Mbit Kapazität/brutto

(IBM JR&D 1995)

DRAM: Trend und Dilemma

- Preisverfall: 16Mb: 50\$ @ 1/96 -> 10\$ @ 12/96 -> 4\$ @ 12/97
- Anzahl DRAMs / Computer sinkt:
 - Kapazität steigt mit 50% - 60% / Jahr
 - Software benötigt 33% / Jahr
 - Mindestanzahl gegeben durch Busbreite vs. DRAM-Breite (4bit)
- überhaupt ein Markt für große DRAMs? (256Mb, 1Gb, ...)

# Chips	'86	'89	'92	'96	'99	'02
	1Mb	4Mb	16Mb	64Mb	256Mb	1Gb
4 MB	32	8			60% / Jahr	→
8 MB		16	4			
16 MB			8	2		
32 MB			16	4	1	
64 MB				8	2	
128 MB	33% / Jahr				4	1
256 MB	↓				8	2