



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik



64-040 Modul InfB-RSB

Rechnerstrukturen und Betriebssysteme

[https://tams.informatik.uni-hamburg.de/
lectures/2024ws/vorlesung/rsb](https://tams.informatik.uni-hamburg.de/lectures/2024ws/vorlesung/rsb)

– Kapitel 16 –

Andreas Mäder



Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik

Technische Aspekte Multimodaler Systeme

Wintersemester 2024/2025



VLSI-Entwurf und -Technologie

Motivation

Halbleitertechnologie

Halbleiter

Herstellung von Halbleitermaterial

Planarprozess

CMOS-Schaltungen

Logische Gatter

Komplexgatter

Transmission-Gate

Tristate-Treiber

Latch und Flipflop

SRAM

CMOS-Herstellungsprozess

Programmierbare Logik

Entwurf Integrierter Schaltungen





Literatur





Das **Konzept des Digitalrechners** (von-Neumann Prinzip) ist völlig unabhängig von der Technologie:

- ▶ mechanische Rechenmaschinen
- ▶ pneumatische oder hydraulische Maschinen
- ▶ Relais, Vakuumröhren, diskrete Transistoren
- ▶ Integrierte Schaltungen: Bipolar, PMOS, NMOS, **CMOS**
- ▶ molekulare Schaltungen, Photonik
- ▶ Quantencomputer
- ▶ ...

Aber:

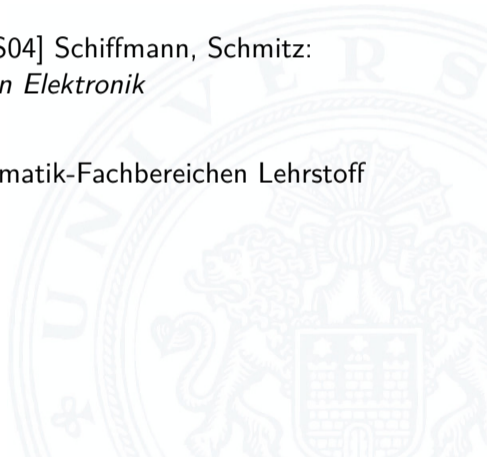
- ▶ nur hochintegrierte Halbleiterschaltungen („VLSI“) erlauben die Massenfertigung mit Milliarden von Komponenten
- ▶ **Halbleiterschaltungen** und **Planarprozess** sind essenzielle Basistechnologien



- ▶ Elektrotechnik wird in dieser Vorlesung nicht vertieft
- ▶ Schulwissen vorausgesetzt

- ▶ Einführung und Grundwissen zum Beispiel aus [SS04] Schiffmann, Schmitz:
Technische Informatik I – Grundlagen der digitalen Elektronik

- ▶ Elektrotechnik ist (weltweit) an den meisten Informatik-Fachbereichen Lehrstoff innerhalb der technischen Informatik
- ▶ Grundkenntnisse sollten aber bekannt sein ...





- ▶ Spannung U und Strom I als Grundgrößen
- ▶ elektrischer Stromkreis mit Quellen und Verbrauchern

- ▶ Spannungs- und Stromquellen
- ▶ Leitungen und Schalter
- ▶ Ohm'scher Widerstand
- ▶ Kondensator und Spule

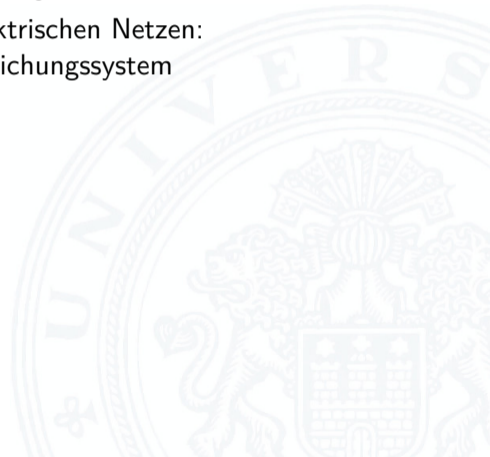
- ▶ Modellierung mit Kennlinien
- ▶ Zeitverhalten bei Änderungen von Spannung und Stromstärke





- ▶ Kirchhoff'sche Gesetze
 - ▶ **Knotenregel** die Summe aller Ströme in einem Knoten ist Null
 - ▶ **Maschenregel** die Summe aller Spannungen entlang einer Masche ist Null
- ▶ Berechnung von Spannungen und Strömen in elektrischen Netzen: Modellierung des Netzes ergibt (Differential-) Gleichungssystem

- ▶ passive Bauelemente
 - ▶ **Widerstand** $R = U/I$ Ohm'sches Gesetz
 - ▶ **Kondensator** $C = Q/U$ $I(t) = C \cdot dU/dt$
 - ▶ **Spule** $H = I \cdot n/l$ $U(t) = L \cdot dI/dt$



- ▶ Strom fließt nur, wenn ein geschlossener Stromkreis besteht
- ▶ Schalter als Elemente zum Öffnen und Schließen einer Verbindung

▶ **Serienschaltung**

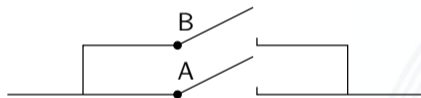
⇒ logisch and



Stromfluss, wenn alle Schalter geschlossen sind

▶ **Parallelschaltung**

⇒ logisch or



Stromfluss, wenn mindestens ein Schalter geschlossen ist

▶ **Wechselschaltung**

⇒ logisch xor



Stromfluss, wenn entweder Schalter A oder B geschlossen sind

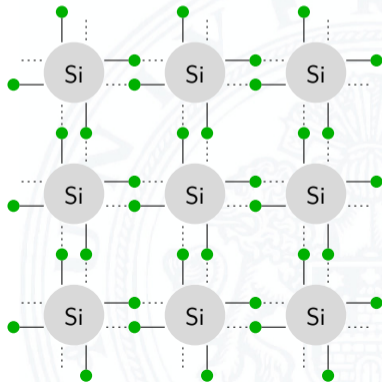


Halbleiter stehen zwischen *Leitern* (z.B.: Metalle) und *Isolatoren*.

- ▶ bei Raumtemperatur quasi nicht-leitend
- ▶ Leitfähigkeit steigt mit der Temperatur \Rightarrow Heißeleiter
- ▶ physikalische Erklärung über Bändermodell
siehe de.wikipedia.org/wiki/Halbleiter

Kristallstruktur aus 4-wertigen Atomen

- ▶ elementare Halbleiter: Ge, Si
- ▶ Verbindungshalbleiter: GaAs, InSb



Leitung im undotierten Kristall

- ▶ Paarentstehung: Elektronen lösen sich aus Gitterverband ein Paar aus Elektron und „Loch“ entsteht
- ▶ Rekombination: Elektronen und Löcher verbinden sich, als quasistatischer Prozess
- ▶ Eigenleitungsichte n_i : temperatur- und materialabhängig

Si : $1,2 \cdot 10^{10} \text{ cm}^{-3}$

Ge : $2,5 \cdot 10^{13} \text{ cm}^{-3}$

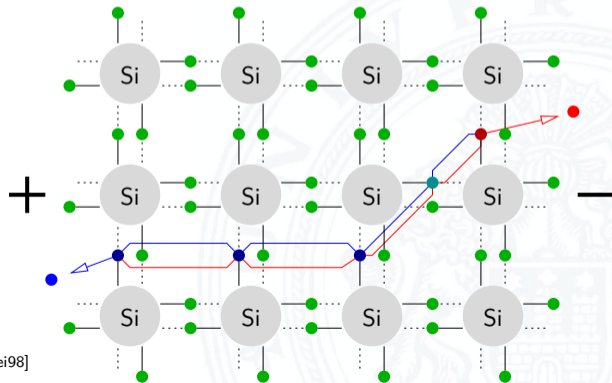
GaAs : $1,8 \cdot 10^6 \text{ cm}^{-3}$

bei $300^\circ \text{K} \approx 20^\circ \text{C}$

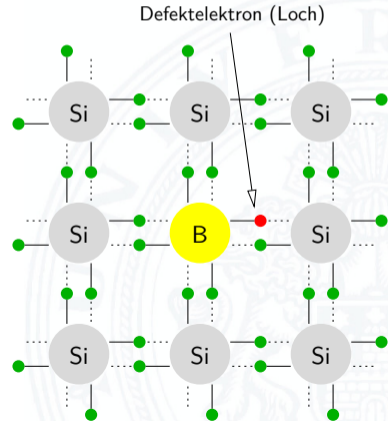
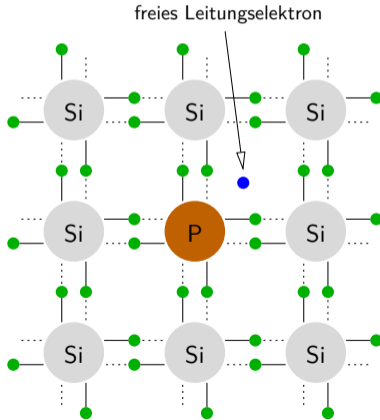
Atomdichte

Si : $5 \cdot 10^{22} \text{ cm}^{-3}$

- ▶ es gilt: $n_i^2 = n_n \cdot n_p$



- ▶ Ein kleiner Teil der vierwertigen Siliziumatome wird durch fünf- oder dreiwertige Atome ersetzt
- ▶ zusätzliche, frei bewegliche Ladungsträger



Dotierung mit Fremdatomen (cont.)

- ▶ Donatoren, Elektronenspender: Phosphor, Arsen, Antimon
- ▶ Akzeptoren: Bor, Aluminium, Gallium, Indium

▶ Dotierungsdichten:

Stärke		Fremdatome [cm^{-3}]
schwach	n^-, p^-	$10^{15} \dots 10^{16}$
mittel	n, p	$10^{16} \dots 10^{19}$
stark	n^+, p^+	$10^{19} \dots$

- ▶ Beweglichkeit μ : materialspezifische Größe

$T = 300^\circ K$		Si	Ge	GaAs [$cm^2/(Vs)$]
Elektronen	μ_n	1500	3900	8500
Löcher	μ_p	450	1500	400

- ▶ Leitfähigkeit: ergibt sich aus Material, Beweglichkeit und Ladungsträgerdichte(n)
 $K = e(n_n\mu_n + n_p\mu_p)$



Dotierung mit Fremdatomen (cont.)

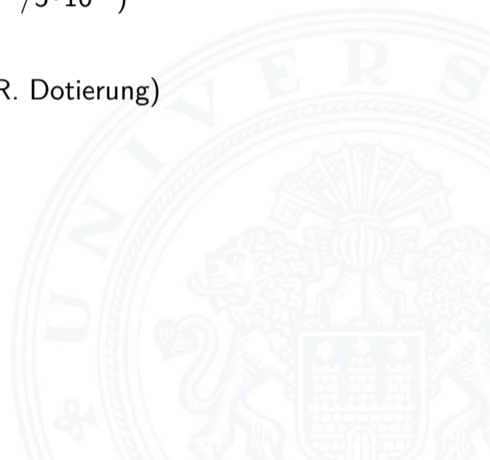
- ▶ selbst bei hoher Dotierung ist die Leitfähigkeit um Größenordnungen geringer als bei Metallen

Si 1 freier Ladungsträger pro 500 Atome ($10^{19}/5 \cdot 10^{22}$)

Met mindestens 1 Ladungsträger pro Atom

- ▶ Majoritätsträger: Ladungsträger in Überzahl (i.d.R. Dotierung)
Minoritätsträger: Ladungsträger in Unterzahl

$$n_i^2 = n_n \cdot n_p$$



Übersicht in: de.wikipedia.org/wiki/Silicium



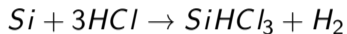
- ▶ Siliziumoxid (SiO_2): Sand, Kies ...
ca. 20% der Erdkruste
- ▶ Herstellung im Lichtbogenofen: Siliziumoxid + Koks
 $SiO_2 + 2C \rightarrow Si + 2CO$
- ▶ amorphe Struktur, polykristallin
- ▶ noch ca. 2% Verunreinigungen (Fe, Al ...)





Ziel: Fremdatome aus dem Silizium entfernen

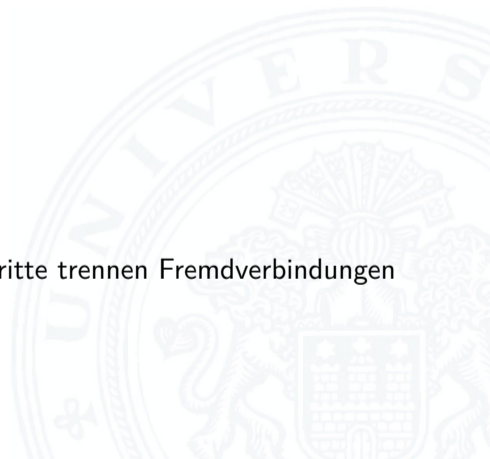
1. chemische Bindung des Siliziums



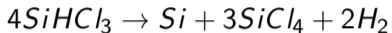
Reaktion mit Salzsäure erzeugt

$SiHCl_3$	Trichlorsilan
$SiCl_4$	Siliziumchlorid (10%)
SiH_2Cl_4	div. andere Chlorsilane/Silane
$FeCl_2, AlCl_3$	div. Metallchloride

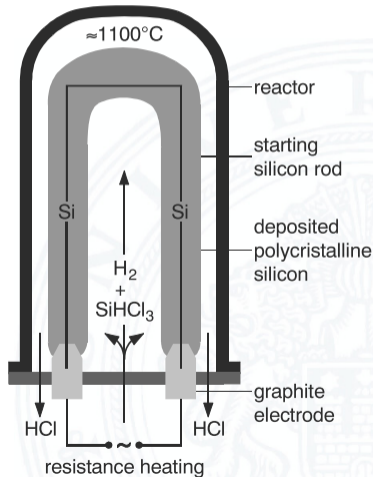
- ## 2. verschiedene Kondensations- und Destillationsschritte trennen Fremdverbindungen ab, hochreines Trichlorsilan entsteht
- < 1ppm Verunreinigungen



3. CVD (Chemical Vapour Deposition) zur Abscheidung des Trichlorsilans zu elementarem Silizium



- ⇒ polykristallines Silizium
< 0,1ppm Verunreinigungen



Weitere Ziele

- ▶ Einkristalline Struktur erzeugen
- ▶ Reinheit für Halbleiterherstellung erhöhen
<, \ll 1ppb
- ▶ ggf. Dotierung durch Fremdatome einbringen

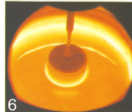
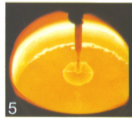
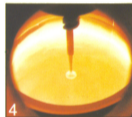
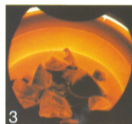
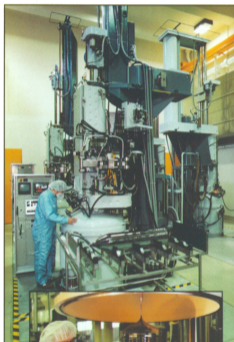
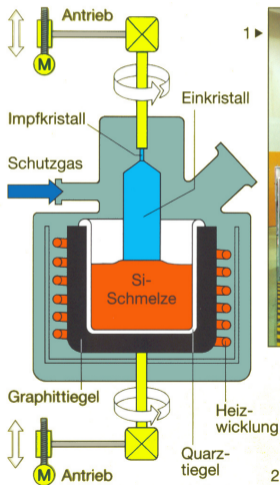
Es gibt dazu mehrere technische Verfahren, um

- ▶ polykristallines Silizium zu schmelzen
- ▶ monokristallin an Impfkristall anzulagern

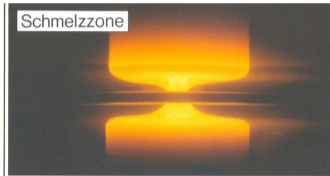
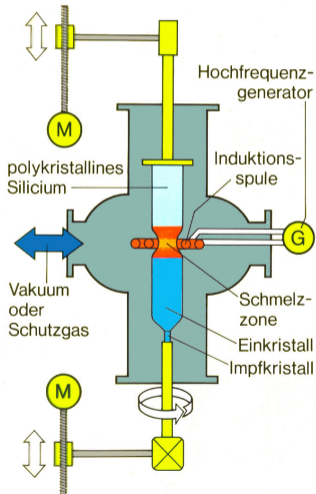
⇒ erzeugt Siliziumeinkristall: *Ingot*
 $\varnothing > 450$ mm, Höhe: > 2 m



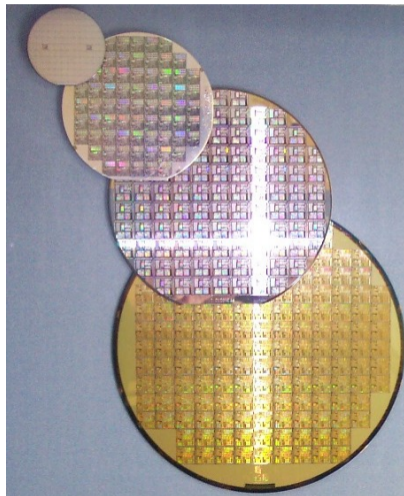
► Czochralski-Verfahren (Tiegelziehverfahren)



► Zonenschmelz- / Zonenziehverfahren

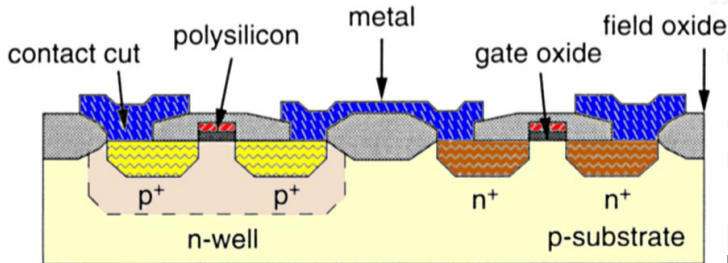


- ▶ weitere Bearbeitungsschritte: zersägen, schleifen, läppen, ätzen, polieren
- ▶ Durchmesser bis 45 cm
Dicke $< 1\text{mm}$
Rauhigkeit $\approx \text{nm}$
- ▶ Markierungen: Kerben, Lasercodes ...
früher „flats“



Technologien zur Erstellung von Halbleiterstrukturen

- ▶ Epitaxie: Aufwachsen von Schichten
- ▶ Oxidation von Siliziumoberflächen: SiO_2 als Isolator
- ▶ Strukturierung durch Lithografie
- ▶ Dotierung des Kristalls durch Ionenimplantation oder Diffusion
- ▶ Ätzprozesse: Abtragen von Schichten



[WE93]



Links

- ▶ www.halbleiter.org
- ▶ eesemi.com
- ▶ resource.renesas.com/lib/eng/fab
- ▶ en.wikipedia.org/wiki/Semiconductor_device_fabrication
- ▶ de.wikipedia.org/wiki/Halbleitertechnik

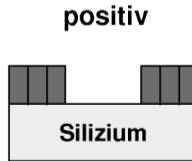
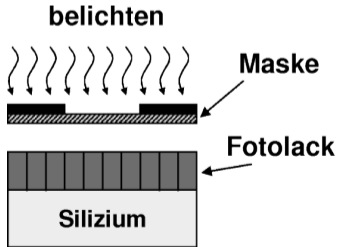
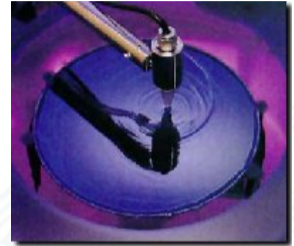
Schritte im Planarprozess

- ▶ n-Kanal-FET:
www.halbleiter.org/grundlagen/aufbau-eines-feldeffekttransistors
- CMOS Inverter, Abschnitt *16.4 CMOS-Herstellungsprozess*

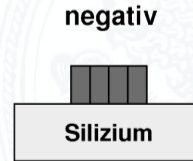
Übertragung von Strukturen durch einen Belichtungsprozess

1. Lack Auftragen (Aufschleudern)

- ▶ Positivlacke: hohe Auflösung, hohe Reproduzierbarkeit
- ▶ Negativlacke: robust, thermisch stabil



entwickeln





2. „Belichten“

- ▶ Maskenverfahren (UV-Lichtquelle): 1:1 Belichtung, Step-Verfahren
- ▶ Struktur direkt schreiben: Elektronen- / Ionenstrahl
- ▶ andere Verfahren: Röntgenstrahl- / Interferenzlithografie
- ▶ mechanischer Druck: Nanoimprint Lithografie (NIL)
- ▶ **EUV-Lithografie**: Technik der aktuellen Chip-Generationen

3. Entwickeln, Härten, Lack entfernen

- ▶ je nach Lack verschiedene chemische Reaktionsschritte
- ▶ Härtung durch Temperatur

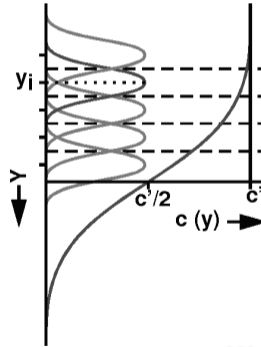
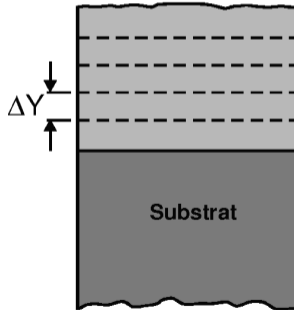
... weitere Schritte des **Planarprozess**

Problem: Strukturen \ll Lichtwellenlänge

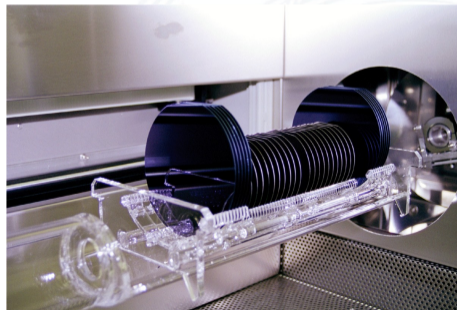
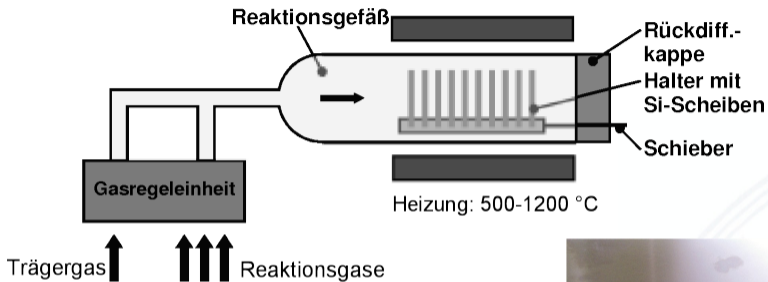


Fremdatome in den Siliziumkristall einbringen

- ▶ Diffusion
 - ▶ Diffusionsofen
 - ▶ gaußförmiges Dotierungsprofil, Konzentration der Dotieratome nimmt ab

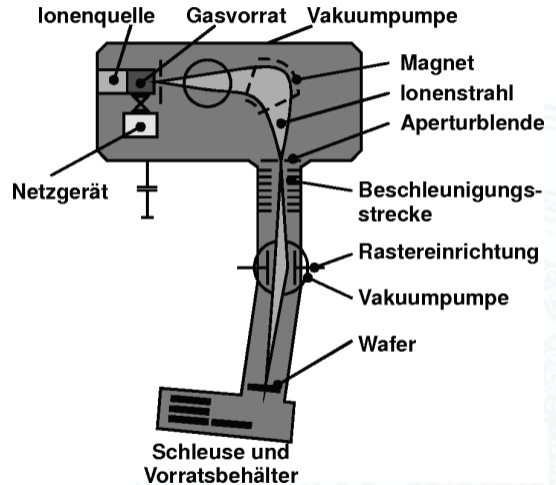


Dotierung (cont.)



▶ Ionenimplantation

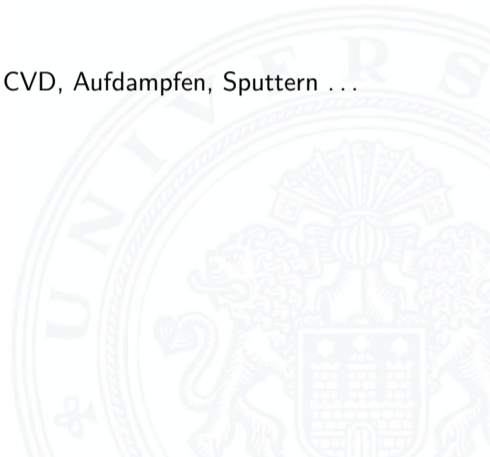
- ▶ „Beschuss“ mit Ionen
- ▶ Beschleunigung der Ionen im elektrischen Feld
- ▶ über die Energie der Ionen kann die Eindringtiefe sehr genau eingestellt werden
- ▶ „Temperung“ notwendig: Erhitzen des Einkristalls zur Neuorganisation des Kristallgitters



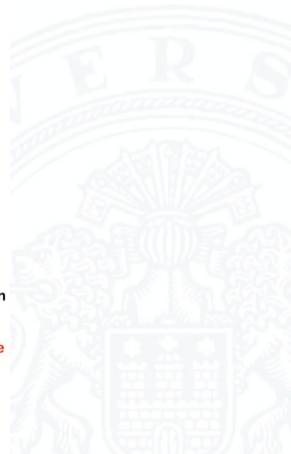
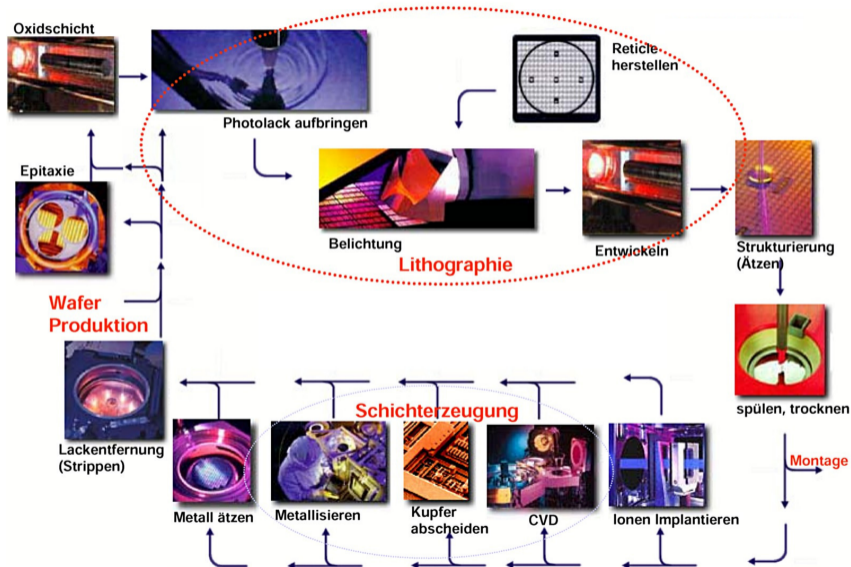


- ▶ zentraler Ablauf bei der Herstellung von Mikroelektronik
- ▶ ermöglicht die gleichzeitige Fertigung aller Komponenten auf dem Wafer

- ▶ Schritte
 1. Vorbereiten / Beschichten des Wafers: Oxidation, CVD, Aufdampfen, Sputtern ...
 2. Strukturieren durch Lithografie
 3. Übertragen der Strukturen durch Ätzprozesse
 4. Modifikation des Materials: Dotierung, Oxidation
 5. Vorbereitung für die nächsten Prozessschritte ...
 - ▶ vielfach Wiederholen
 - ▶ insgesamt hunderte Einzelschritte



Planarprozess (cont.)





- ▶ MOS: Metal Oxide Semiconductor

FET : Feldeffekttransistor

- ▶ de.wikipedia.org/wiki/Feldeffekttransistor
- ▶ de.wikipedia.org/wiki/MOSFET

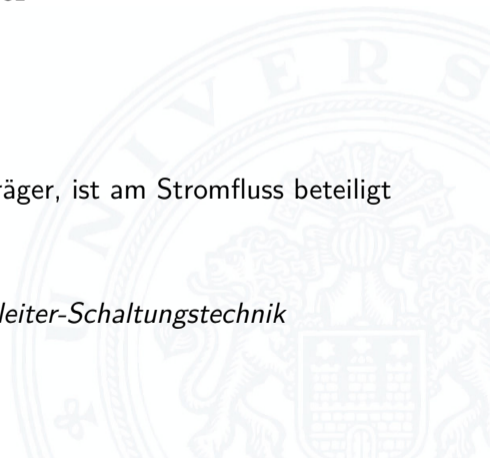
Literatur: [TSG19, WE93]

- ▶ unipolarer Transistor

nur eine Art von Ladungsträgern, die Majoritätsträger, ist am Stromfluss beteiligt

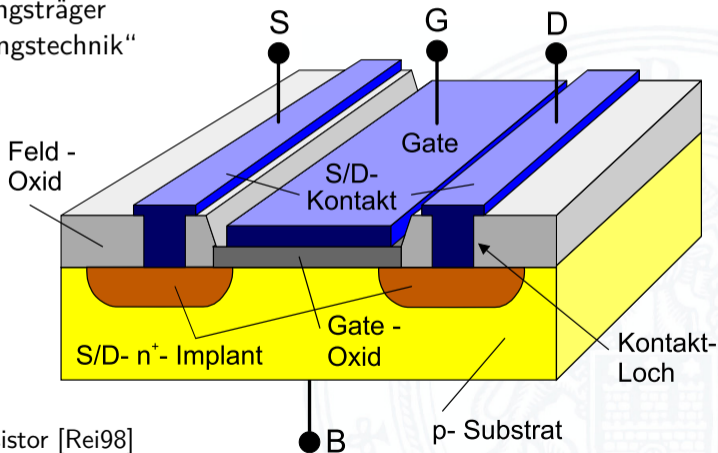
- ▷ im Gegensatz zu Bipolartransistoren

siehe z.B.: [TSG19] Tietze, Schenk, Gamm: *Halbleiter-Schaltungstechnik*



MOS-Transistor (cont.)

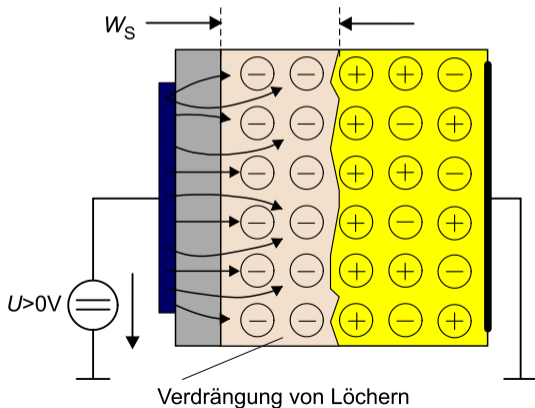
- **Source** Quelle der Ladungsträger
- Gate** steuert den Stromfluss
- Drain** Senke der Ladungsträger
- Bulk** siehe „Herstellungstechnik“



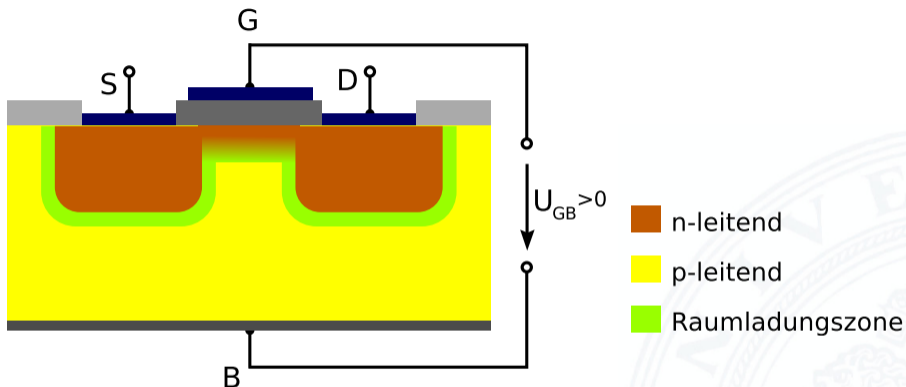
NMOS Transistor [Rei98]

► Funktionsweise

- Ladung des Gates erzeugt ein elektrisches Feld
- durch Inversion werden Ladungsträger unterhalb des Gates verdrängt
- ein leitender Kanal zwischen Source und Drain entsteht



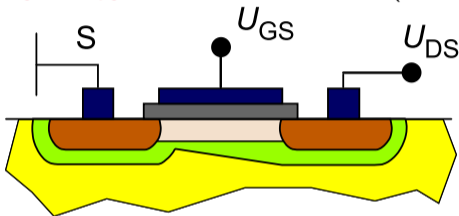
MOS-Kondensator [Rei98]



- ▶ Raumladungszone: neutral, keine freien Ladungsträger
- ▶ Schwellspannung U_P : abhängig von der Dotierungsdichte, den Parametern des MOS-Kondensators (Dicke und Material der Gate-Isolationsschicht) ...
 U_P möglichst klein: 0,3...0,8V früher: deutlich mehr

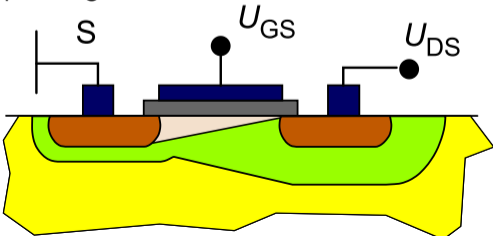
MOS-Transistor (cont.)

- ▶ $U_{DS} \ll U_{GS} - U_P$ normaler Betrieb (Triodenbereich)



[Rei98]

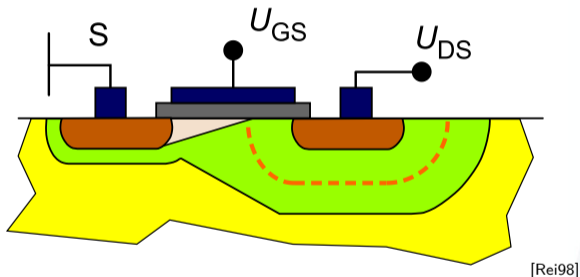
- ▶ $U_{DS} = U_{GS} - U_P$ Kanalabschnürung
Spannungsabfall zwischen S und D durch den Kanalwiderstand



[Rei98]

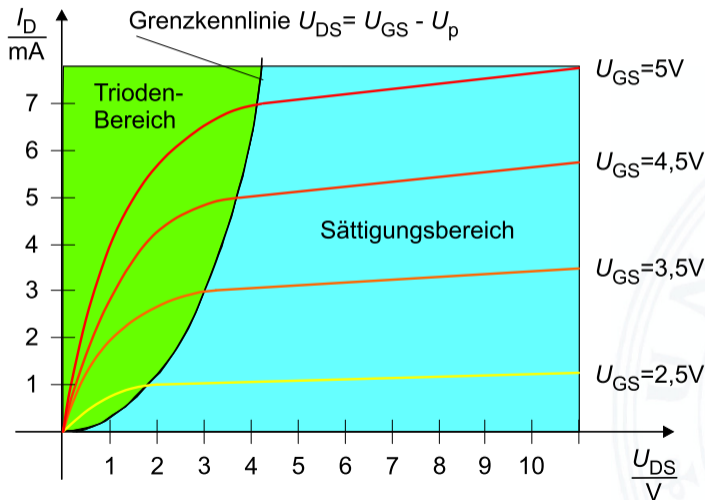
- ▶ $U_{DS} > U_{GS} - U_P$ Kanalverkürzung (Sättigungsbereich)

Der Kanal wird weiter verkürzt, die Spannung U_{DS} bewirkt ein virtuell größeres Drain durch Inversion. I_D wächst nur noch minimal.



- ⇒ kurze Kanäle aktueller Submikronprozesse können allein durch hohe Spannungen U_{DS} leitend werden (Durchgreifbetrieb)
- ⇒ einer der Gründe für sinkende Versorgungsspannungen

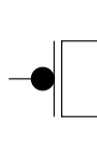
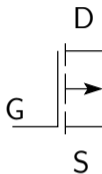
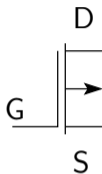
- Kennlinienfeld: N-Kanal Transistor, selbstsperrend



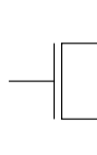
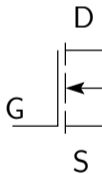
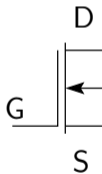
[Rei98]

MOS-Transistor: Schaltsymbole

P-Kanal



N-Kanal



selbstleitend
Verarmungs-

selbstsperrend
Anreicherungstyp



Complementary Metal-Oxide Semiconductor

die derzeit dominierende Technologie für alle hochintegrierten Schaltungen

- ▶ Schaltungsprinzip nutzt n-Kanal und p-Kanal Transistoren
 - ▶ alle elementaren Gatter verfügbar
 - ▶ effiziente Realisierung von *Komplexgattern*
 - ▶ *Transmission-Gate* als elektrischer Schalter
 - ▶ effiziente Realisierung von Flipflops und Speichern
-
- + sehr hohe Integrationsdichte möglich, gut skalierbar
 - + sehr schnelle Schaltgeschwindigkeit der Gatter
 - + sehr geringer Stromverbrauch pro Gatter möglich
 - + Integration von digitalen und analogen Komponenten



- ▶ Schaltungsprinzip
- ▶ Inverter und nicht-invertierender Verstärker
- ▶ NAND, NAND3, NOR (und AND, OR)
- ▶ XOR

- ▶ Komplexgatter
- ▶ Transmission-Gate

- ▶ Beispiele für Flipflops
- ▶ SRAM



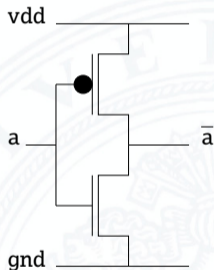
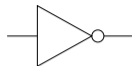


- ▶ Transistoren werden als Schalter betrachtet
- ▶ zwei zueinander **komplementäre** Zweige der Schaltung
 - ▶ n-Kanal Transistoren zwischen Masse und Ausgang y 1 on
 - ▶ p-Kanal $-$ $-$ V_{dd} und Ausgang y 0 on
 - ▶ p-Kanal Zweig komplementär („dualer Graph“) zu n-Kanal Zweig: jede Reihenschaltung von Elementen wird durch eine Parallelschaltung ersetzt (und umgekehrt)
- ▶ immer ein direkt leitender Pfad von entweder V_{dd} („1“) oder Masse / Gnd („0“) zum Ausgang
- ▶ niemals ein direkt leitender Pfad von V_{dd} nach Masse
- ▶ kein statischer Stromverbrauch im Gatter



Funktionsweise

- ▶ selbstsperrende p- und n-Kanal Transistoren
- ▶ komplementär beschaltet
- ▶ Ausgang: Pfad über p-Transistoren zu V_{dd}
–"– n-Transistoren zu G_{nd}
- ▶ genau *einer* der Pfade leitet



- ▶ Eingang p-Tran. n-Tran. Ausgang

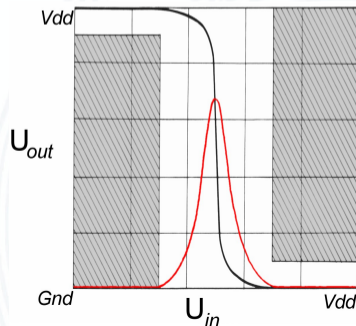
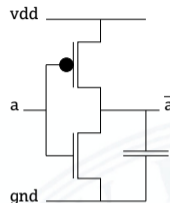
$a = 0 \rightarrow$ leitet / sperrt \rightarrow über T_P mit V_{dd} verbunden = 1

$a = 1 \rightarrow$ sperrt / leitet \rightarrow über T_N mit G_{nd} verbunden = 0

Leistungsaufnahme

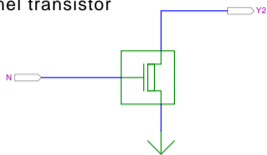
1. $U_{in} = 0$, bzw. V_{dd} : Sperrstrom, nur μA
 \Rightarrow niedrige statische Leistungsaufnahme
2. Querstrom beim Umschalten:
kurzfristig leiten beide Transistoren
 \Rightarrow Forderung nach steilen Flanken
3. Kapazitive Last: Fanout-Gates
Energie auf Gate(s): $W = \frac{1}{2} C_T V_{dd}^2$
Verlustleistung_(0/1/0): $P = C_T V_{dd}^2 \cdot f$

\Rightarrow Transfercharakteristik

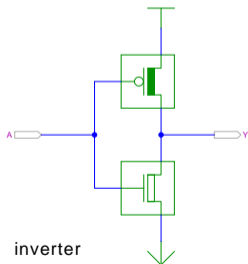
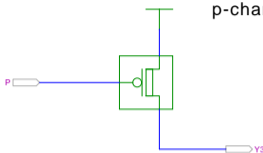


Hades: n- und p-Kanal Transistor, Inverter, Verstärker

n-channel transistor

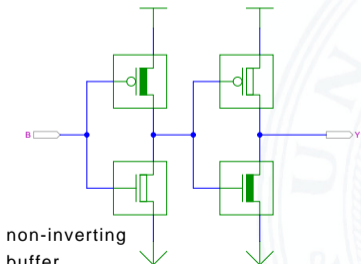


p-channel transistor



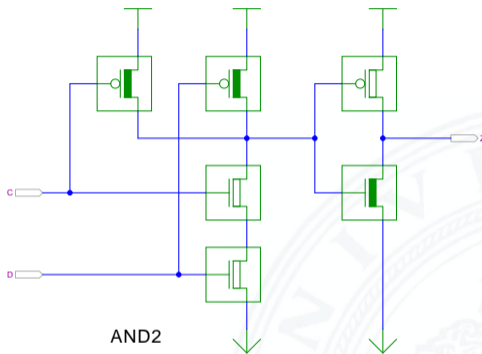
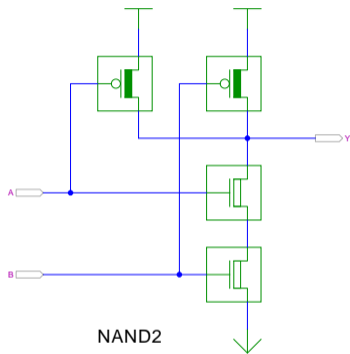
inverter

$$Y = \overline{A}$$



non-inverting
buffer

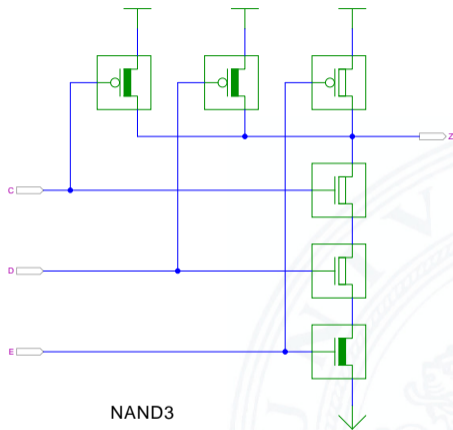
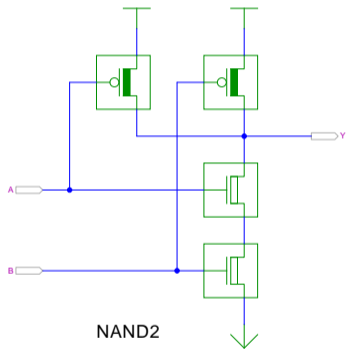
$$Y = \overline{\overline{A}} = A$$



[HenHA] Hades Demo: 05-switched/40-cmos/nand

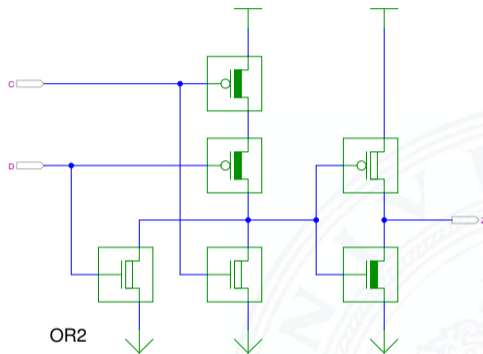
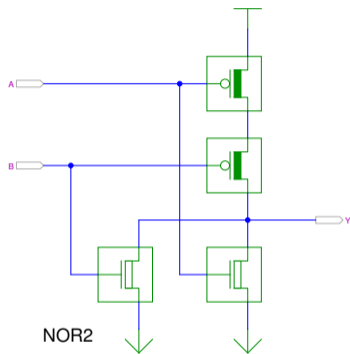
- ▶ NAND: n-Transistoren in Reihe, p-Transistoren parallel
- ▶ AND: Kaskade aus NAND und Inverter

NAND- und AND-Gatter (cont.)



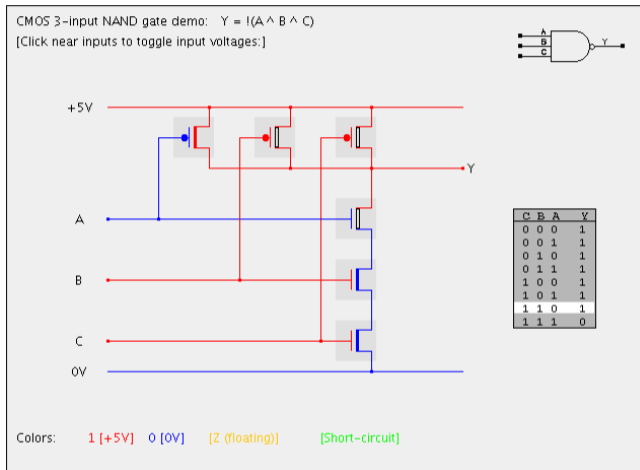
[HenHA] Hades Demo: 05-switched/40-cmos/nand3

- ▶ n-Transistoren in Reihe, p-Transistoren parallel
- ▶ normalerweise max. 4 Transistoren in Reihe (Spannungsabfall)



[HenHA] Hades Demo: 05-switched/40-cmos/nor

- ▶ Struktur komplementär zum NAND/AND
- ▶ n-Transistoren parallel, p-Transistoren in Reihe
- ▶ Reihenschaltung von p-Kanal Transistoren schlechter \Rightarrow oft langsamer als NAND



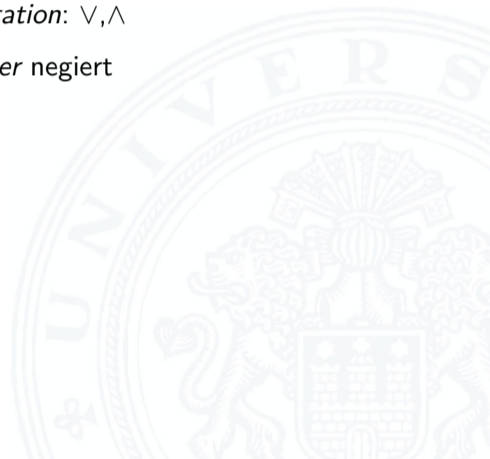
- ▶ Interaktive Demonstration der CMOS-Grundgatter [HenCM]
tams.informatik.uni-hamburg.de/applets/cmos



Gatterfunktionen

- ▶ Schaltungen: *negierte monotone boole'sche Funktionen*
- ▶ Beliebiger schaltalgebraischer Ausdruck *ohne Negation*: \vee, \wedge
- ▶ Negation des gesamten Ausdrucks: Ausgang *immer* negiert
- ▶ je Eingang: ein Paar p-/n-Kanal Transistoren
- ▶ Dualitätsprinzip: n- und p-Teil des Gatters

n-Teil	p-Teil	Logik, ohne Negation
seriell	\Leftrightarrow parallel	$\equiv \wedge$ / und
parallel	\Leftrightarrow seriell	$\equiv \vee$ / oder





▶ Konstruktion

1. n-Teil aus Ausdruck ableiten
beliebige Parallel- und Serienschaltung der n-Transistoren
 2. p-Teil dual dazu entwickeln
komplementäre Seriell- und Parallelschaltung der p-Transistoren
 - ▶ typischerweise max. 4 Transistoren in Reihe
- ▶ viele invertierende logische Funktionen effizient realisierbar
 - ▶ Schaltungslayout automatisch synthetisierbar
 - ▶ zwei gängige Varianten
 - ▶ AOI-Gatter („AND-OR-invert“)
 - ▶ OAI-Gatter („OR-AND-invert“)





Beispiel: $\overline{(a \wedge b \wedge c)} \vee d \vee (e \wedge f)$

AND-OR-INVERT Struktur

\Rightarrow AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte
(6+2)+(0)+(4+2)+6 Transistoren
- ▶ Komplexgatter mit 12 Transistoren





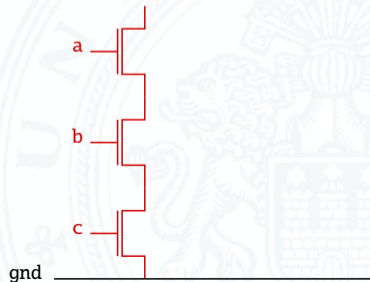
Beispiel: $\overline{(a \wedge b \wedge c) \vee d \vee (e \wedge f)}$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte $(6+2)+(0)+(4+2)+6$ Transistoren
- ▶ Komplexgatter mit 12 Transistoren





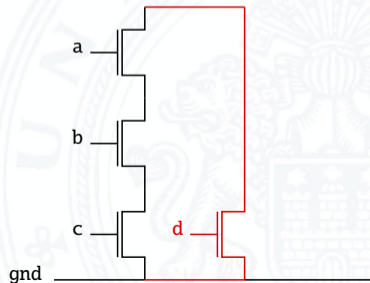
Beispiel: $\overline{(a \wedge b \wedge c) \vee d \vee (e \wedge f)}$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte
(6+2)+(0)+(4+2)+6 Transistoren
- ▶ Komplexgatter mit 12 Transistoren





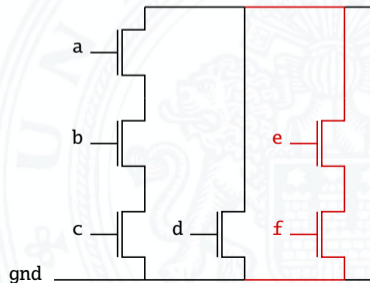
Beispiel: $\overline{(a \wedge b \wedge c)} \vee d \vee (e \wedge f)$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte
(6+2)+(0)+(4+2)+6 Transistoren
- ▶ Komplexgatter mit 12 Transistoren





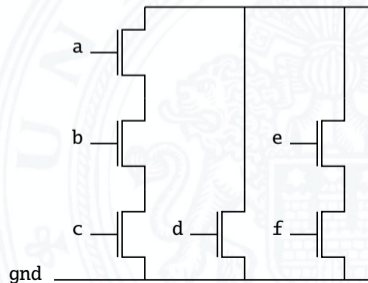
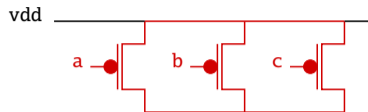
Beispiel: $\overline{(a \wedge b \wedge c)} \vee d \vee (e \wedge f)$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte $(6+2)+(0)+(4+2)+6$ Transistoren
- ▶ Komplexgatter mit 12 Transistoren

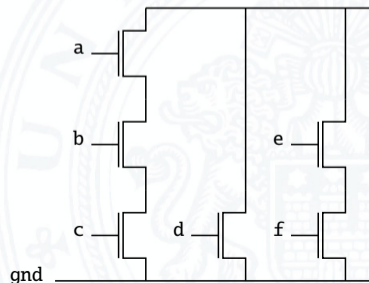
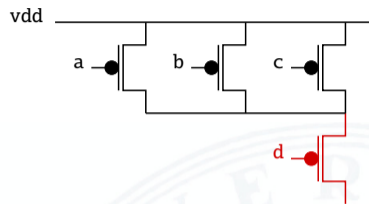


Beispiel: $\overline{(a \wedge b \wedge c) \vee d \vee (e \wedge f)}$

AND-OR-INVERT Struktur

\Rightarrow AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme
- ▶ direkte Realisierung hätte $(6+2)+(0)+(4+2)+6$ Transistoren
- ▶ Komplexgatter mit 12 Transistoren

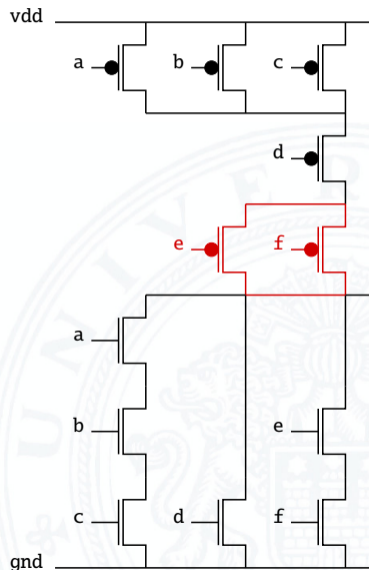


Beispiel: $\overline{(a \wedge b \wedge c)} \vee d \vee (e \wedge f)$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme
- ▶ direkte Realisierung hätte $(6+2)+(0)+(4+2)+6$ Transistoren
- ▶ Komplexgatter mit 12 Transistoren





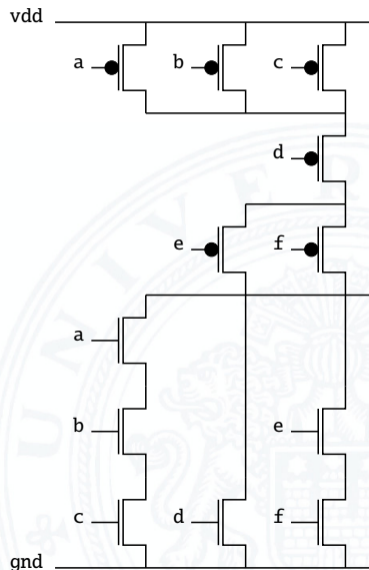
Beispiel: $\overline{(a \wedge b \wedge c) \vee d \vee (e \wedge f)}$

AND-OR-INVERT Struktur

⇒ AOI312-Gatter

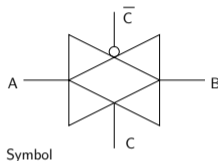
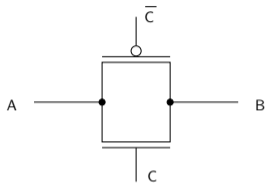
- ▶ AND-Verknüpfung: a, b, c
- ▶ Eingang: d
- ▶ AND-Verknüpfung: e, f
- ▶ NOR-Verknüpfung der drei Terme

- ▶ direkte Realisierung hätte $(6+2)+(0)+(4+2)+6$ Transistoren
- ▶ Komplexgatter mit 12 Transistoren



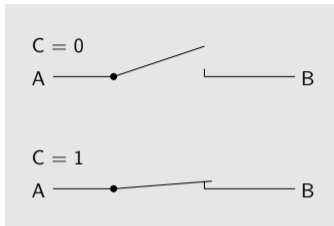
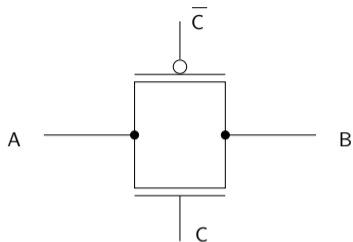


Transmissions-Gatter (*transmission gate, t-gate*)



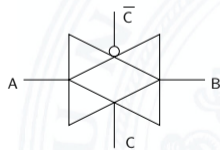
- ▶ Paar aus je einem n- und p-Kanal MOS-Transistor
 - ▶ symmetrische Anordnung
 - ▶ Ansteuerung der Gate-Elektroden mit invertierter Polarität
- ⇒ entweder beide Transistoren leiten oder beide sperren
- ▶ Funktion entspricht **elektrisch gesteuertem Schalter**
 - ▶ effiziente Realisierung vieler Schaltungen

Transmission-Gate (cont.)



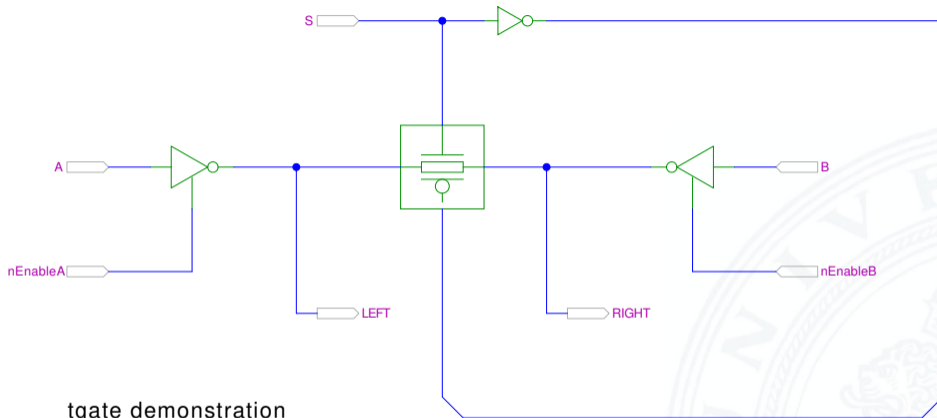
elektrisch gesteuerter Schalter:

- ▶ $C = 0$: keine Verbindung von A nach B
- ▶ $C = 1$: leitende Verbindung von A nach B
- ▶ symmetrisch in beide Richtungen





Transmission-Gate: Demo



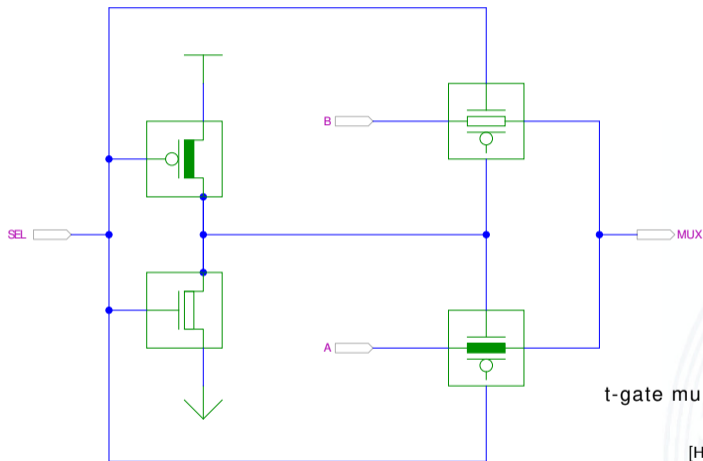
tgate demonstration

[HenHA] Hades Demo: 05-switched/40-cmos/tgate

- ▶ Werte A und B anlegen, Treiber mit enable-Signalen aktivieren
- ▶ Gatter mit S ein- oder ausschalten



T-Gate Multiplexer

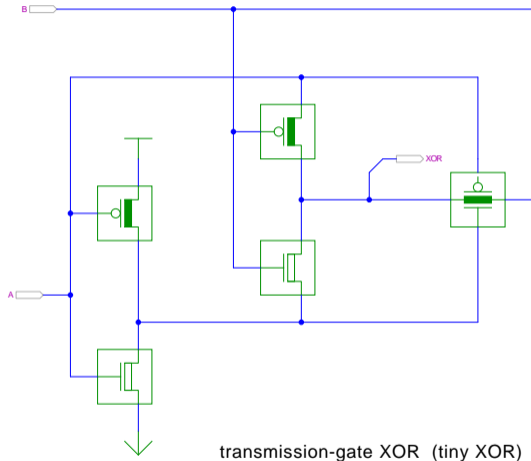


t-gate multiplexer

[HenHA] Hades Demo: 05-switched/40-cmos/mux-tgate

- ▶ kompakte Realisierung (4 bzw. 6 Transistoren)
- ▶ Eingänge A und B werden nicht verstärkt \Rightarrow nur begrenzt kaskadierbar

T-Gate XOR-Gatter



[HenHA] Hades Demo: 05-switched/40-cmos/xor-tgate

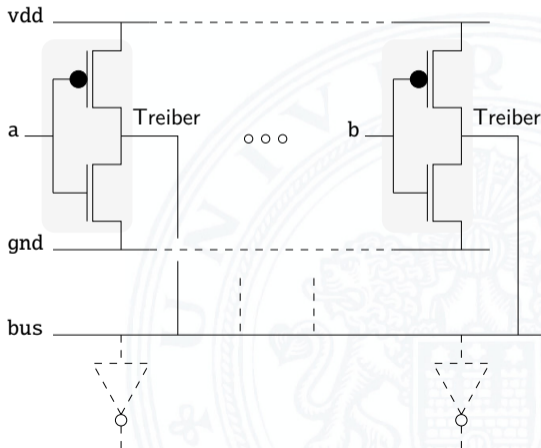
- ▶ kompakte Realisierung des XOR (nur 6 Transistoren)
- ▶ Eingang *B* nicht verstärkt \Rightarrow nur begrenzt kaskadierbar



Bussysteme

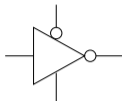
- ▶ Quellen: „Bustreiber“ (Werte: 0, 1)
- ▶ Senken : Gattereingänge
- ▶ Probleme
 - ▶ *Kurzschluss*
 - ▶ *offene Eingänge*

⇒ Tristate



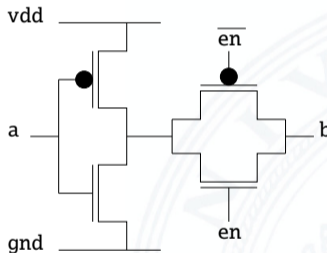


Beispiel: Tristate-Inverter



Funktionsweise

- ▶ Ausgang elektrisch trennen
z.B. mit Transmission-Gate
- ▶ 3-Pegel: 0, 1, Z *hochohmig*



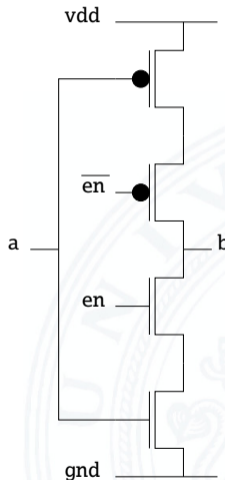
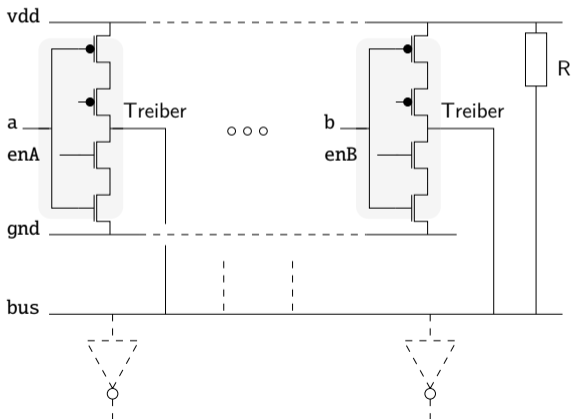
- | Enable | Verbindung | Ausgang |
|----------|---------------|--------------------------|
| $en = 0$ | → getrennt | → $bus = Z$ hochohmig |
| $en = 1$ | → geschlossen | → $bus = \bar{a}$ $f(a)$ |



Tristate-Treiber (cont.)

Tristate-Bussystem

- ▶ *pull-up/-down* Widerstand R (offene Eingänge)
- ▶ nur **genau ein** Treiber gleichzeitig aktiv





1. statisch

- ▶ Speicherung: Rückkopplung von (statischen) Gattern, siehe: „Schaltwerke – Flipflops“
- + taktunabhängig
- + sicher

2. quasi-statisch

- ▶ Speicherung: Rückkopplung von Gattern
- ▶ Transmission-Gates als Multiplexer
- + taktunabhängig
- + kleiner

3. dynamisch

- ▶ Speicherung: Gate-Kapazitäten
- ▶ verschiedene Taktschemata/Schaltungsvarianten
- muss getaktet werden
- schwieriger zu Entwerfen (wegen Taktschema)
- + Integration in Datenpfade (arithmetische Pipelines)
- + sehr hohe Taktfrequenzen
- + sehr klein





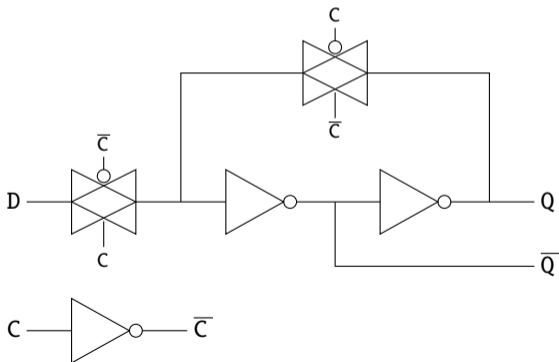
D-Latch: quasi-statisch

► Transmission-Gates als Schalter

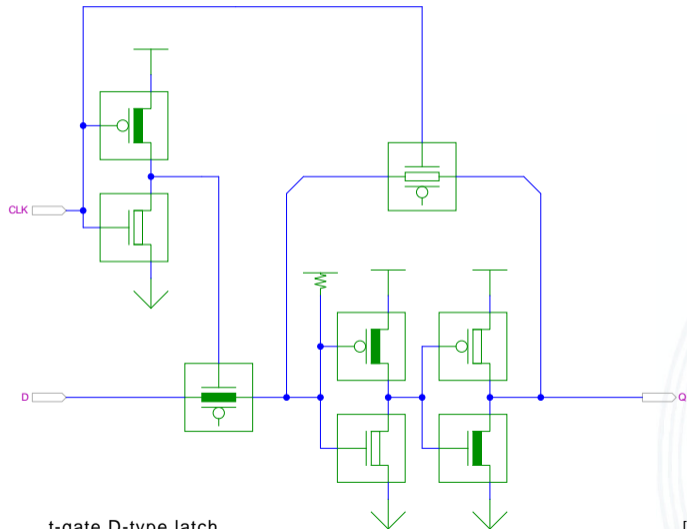
$C = 1$ Transparent: Eingang über die Inverter zum Ausgang

$C = 0$ Speicherung: Rückkopplungspfad aktiv

Latch, high-aktiv



D-Latch: quasi-statisch (cont.)

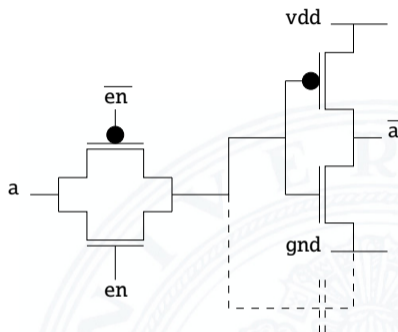


t-gate D-type latch

[HenHA] Hades Demo: 05-switched/40-cmos/latch

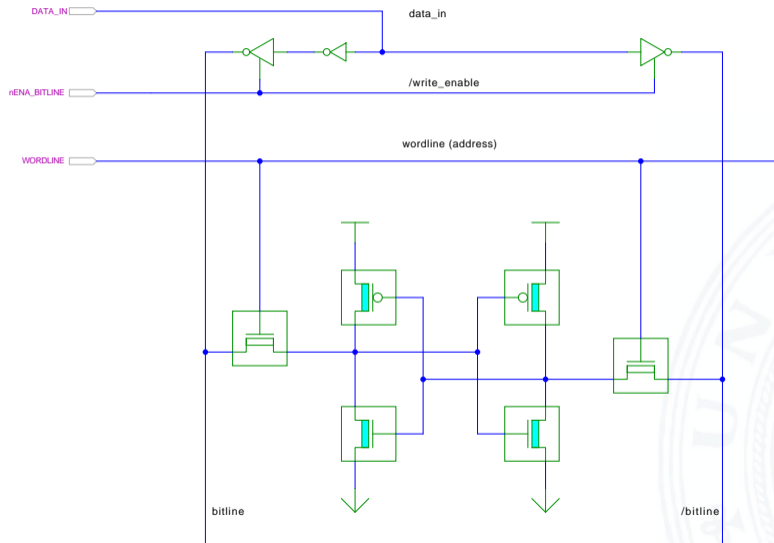
Schaltungsprinzip

- ▶ Speicherung auf Gate-Kapazitäten

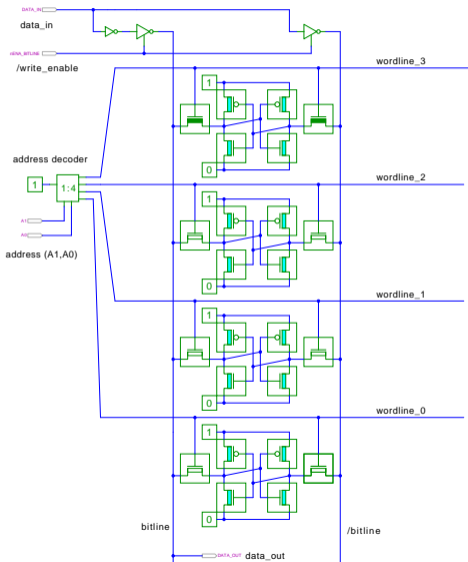


- ▶ Verbindung mit Logikgattern möglich \Rightarrow arithmetische Pipelines
- ▶ viele unterschiedliche Taktschemata/Funktionsweisen
 - ▶ **C²MOS** Clocked CMOS
 - ▶ **No Race** 2-Phasen Takt
 - ▶ **True Single Phase Clock**
 - ▶ ...

SRAM: Sechs-Transistor Speicherstelle („6T“)



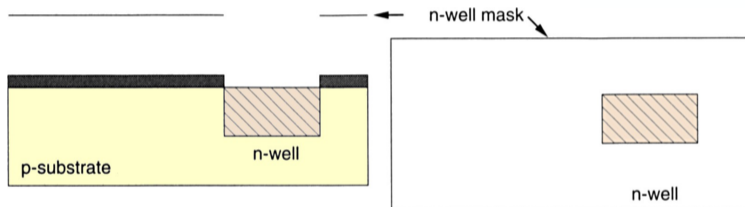
Prinzip des SRAM



Ein n-Wannen Prozesses

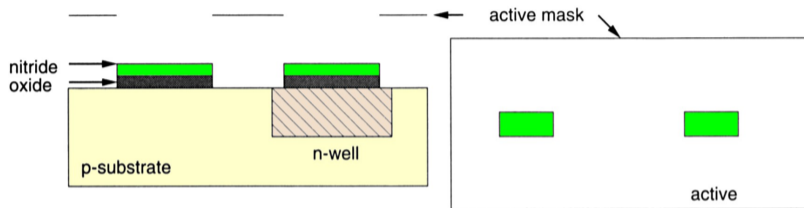
Weste, Eshragian: *Principles of CMOS VLSI Design* [WE93]

1. Ausgangsmaterial: p-dotiertes Substrat
2. n-Wanne



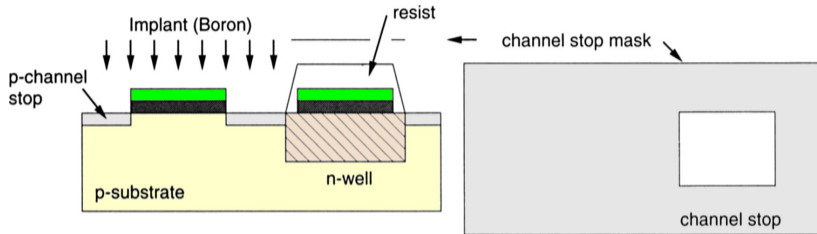
- ▶ Dotierung für p-Kanal Transistoren
- ▶ Herstellung: Ionenimplantation oder Diffusion

3. „aktive“ Fläche / Dünnoxid



- ▶ Spätere Gates und p^+ -/ n^+ -Gebiete
- ▶ Herstellung: Epitaxie SiO_2 und Abdeckung mit Si_3N_4

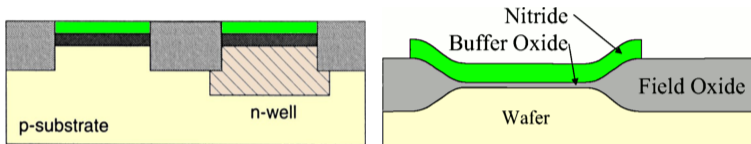
4. p-Kanalstopp



- ▶ Begrenzt n-Kanal Transistoren
 - ▶ p-Wannen Maske, bzw. \neg n-Wanne
 - ▶ Maskiert durch Resist und Si_3N_4
 - ▶ Substratbereiche in denen keine n-Transistoren sind
 - ▶ Herstellung: p^+ -Implant (Bor)
- ▶ n-Kanalstopp aktueller Prozesse: analog dazu

5. Resist entfernen

6. Feldoxid aufwachsen – SiO_2



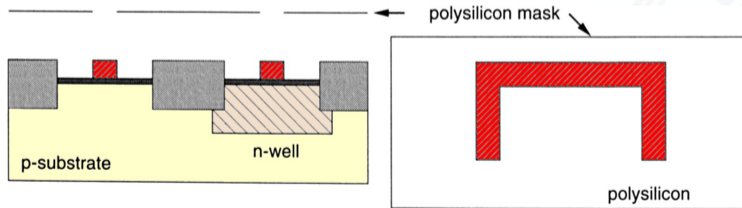
- ▶ LOCOS: **L**ocal **O**xidation of **S**ilicon
- ▶ Maskiert durch Si_3N_4
- ▶ Wächst auch lateral unter Si_3N_4/SiO_2 (aktive) Bereiche („bird's beak“)
- ▶ Der aktive Bereich wird kleiner als vorher maskiert
- ▶ Herstellung: Epitaxie und Oxidation
- ▶ Problem: nicht plane Oberfläche

7. Si_3N_4 entfernen, Gateoxid bleibt SiO_2

8. Transistor Schwellspannungen „justieren“

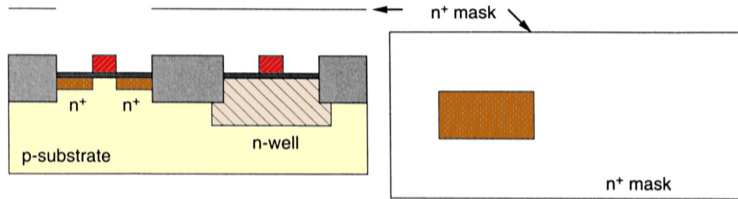
- ▶ Meist wird das Polysilizium zusätzlich n^+ dotiert (Grund: bessere Leitfähigkeit)
- ▶ Problem: $U_D(T_N) \approx 0,5 \dots 0,7V$
 $U_D(T_P) \approx -1,5 \dots -2,0V$
- ▶ Maske: n-Wanne, bzw. p-Wanne
- ▶ Herstellung: Epitaxie einer leicht negativ geladenen Schicht an der Substratoberfläche

9. Polysilizium Gate



- ▶ Herstellung: Epitaxie von Polysilizium, Ätzen nach Planarprozess

10. n^+ -Diffusion

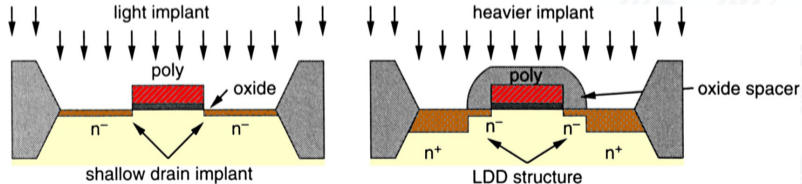


- ▶ Erzeugt Source und Drain der n-Kanal Transistoren
- ▶ Maskiert durch aktiven Bereich, n^+ -Maske und Polysilizium
- ▶ Dotiert auch das Polysilizium Gate leicht (s.o.)
- ▶ Herstellung: Ionenimplantation, durchdringt Gateoxid

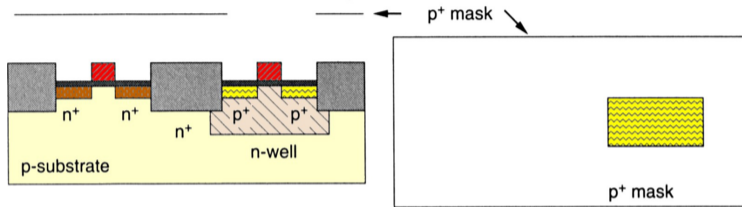
⇒ Selbstjustierung

Zusätzliche Schritte bei der Source/Drain Herstellung

- ▶ Problem „Hot-Carrier“ Effekte (schnelle Ladungsträger): Stoßionisation, Gateoxid wird durchdrungen . . .
- ▶ Lösung: z.B. LDD (**L**ightly **D**oped **D**rain)
 - „flaches“ n-LDD Implant
 - zusätzliches SiO_2 über Gate aufbringen (*spacer*)
 - „normales“ n^+ -Implant
 - Spacer SiO_2 entfernen



11. p⁺-Diffusion



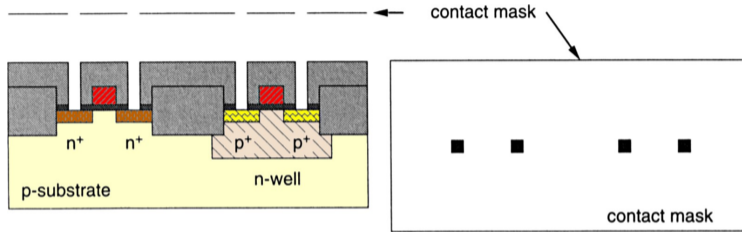
- ▶ Erzeugt Source und Drain der p-Kanal Transistoren
- ▶ Maskiert durch aktiven Bereich, p⁺-Maske und Polysilizium
- ▶ teilweise implizite p⁺-Maske = ¬ n⁺-Maske
- ▶ wenig schnelle Ladungsträger (Löcher), meist keine LDD-Schritte
- ▶ Herstellung: Ionenimplantation, durchdringt Gateoxid

⇒ Selbstjustierung

12. SiO₂ aufbringen, Feldoxid

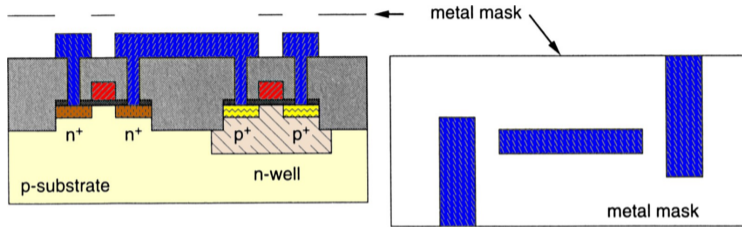
- ▶ Strukturen isolieren
- ▶ Herstellung: Epitaxie

13. Kontaktlöcher



- ▶ Verbindet (spätere) Metallisierung mit Polysilizium oder Diffusion
- ▶ Anschlüsse der Transistoren: Gate, Source, Drain
- ▶ Herstellung: Ätzprozess

14. Metallverbindung



- ▶ Erzeugt Anschlüsse im Bereich der Kontaktlöcher
- ▶ Herstellung: Metall aufdampfen, Ätzen nach Planarprozess

15. weitere Metalllagen

- ▶ Schritte: 12. bis 14. wiederholen
- ▶ Materialien: Wolfram, Kupfer früher: Aluminium
- ▶ aktuell: bis zu $13 \times$ Metall

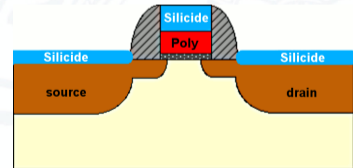
16. Passivierung

- ▶ Chipoberfläche abdecken, Plasmanitridschicht

17. Pad-Kontakte öffnen

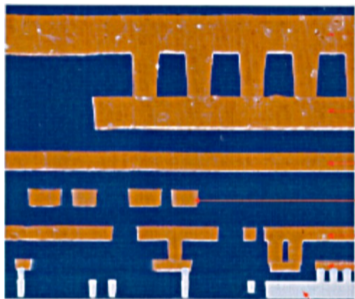
Zahlreiche Erweiterungen für Submikron CMOS-Prozesse

- ▶ „vergrabene“ Layer
 - ▶ verbessern elektrische Eigenschaften
 - ▶ Bipolar-Transistoren
 - ▶ Analog-Schaltungen
- ▶ Gate Spacer, seitlich SiO_2
- ▶ Silizidoberflächen: verringerter Kontaktwiderstand zu Metallisierung



CMOS Prozessschritte: Inverter (cont.)

► Kupfer Metallisierung



Copper 6

Copper 5

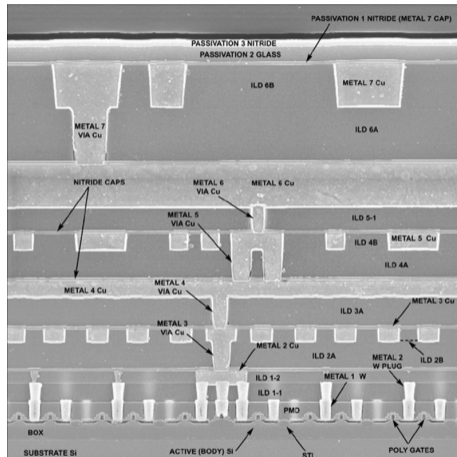
Copper 4

Copper 3

Copper 2

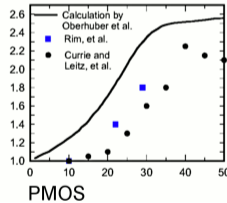
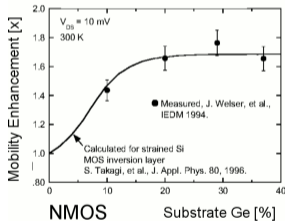
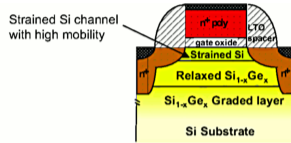
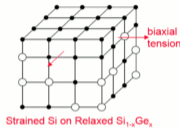
Copper 1

Tungsten
Local Interconnect



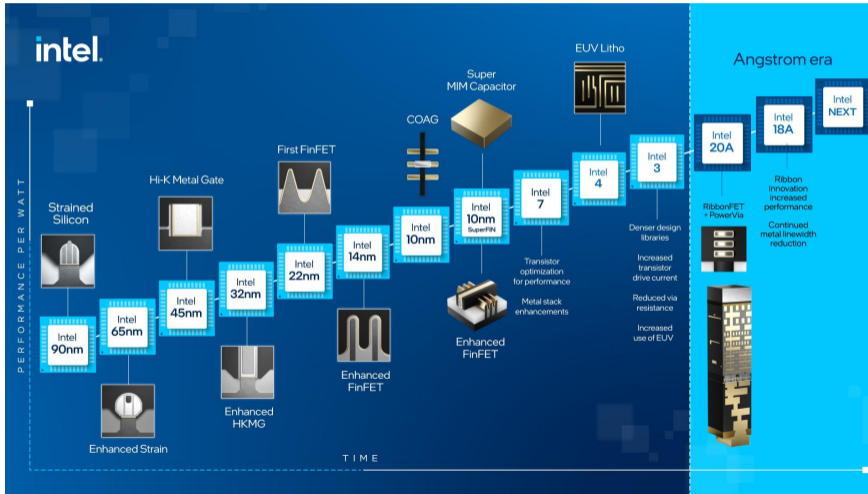
CMOS Prozessschritte: Inverter (cont.)

- ▶ „gestrecktes“ Silizium: höhere Beweglichkeit

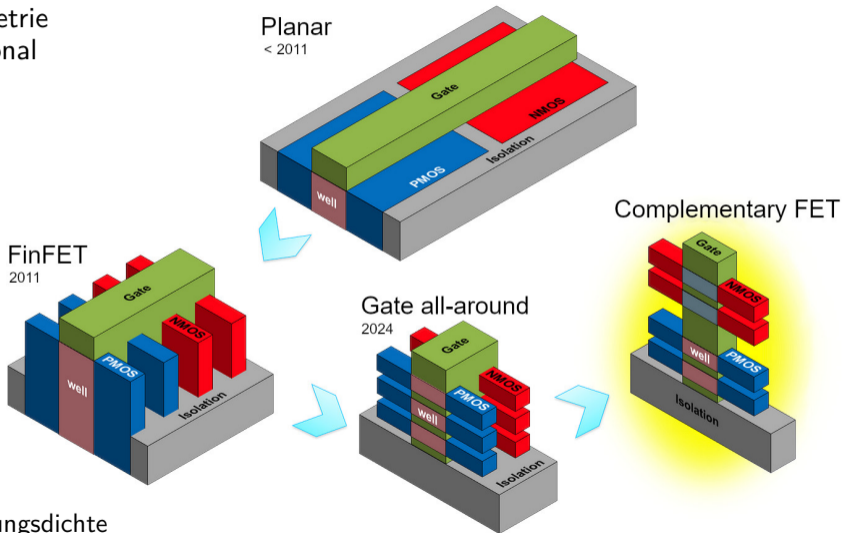


- ▶ high-k Dielektrika
 - ▶ Gate-Isolierung dicker, weniger Leckströme
 - ▶ Metall-Gate

Technischer Fortschritt – Prozesstechnologie



- ▶ Transistorgeometrie wird 3-dimensional



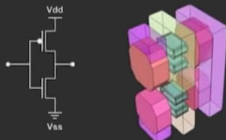
⇒ höhere Packungsdichte

CFET Density Scaling

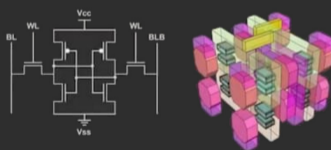
Density scaling ~ 1.5-2X

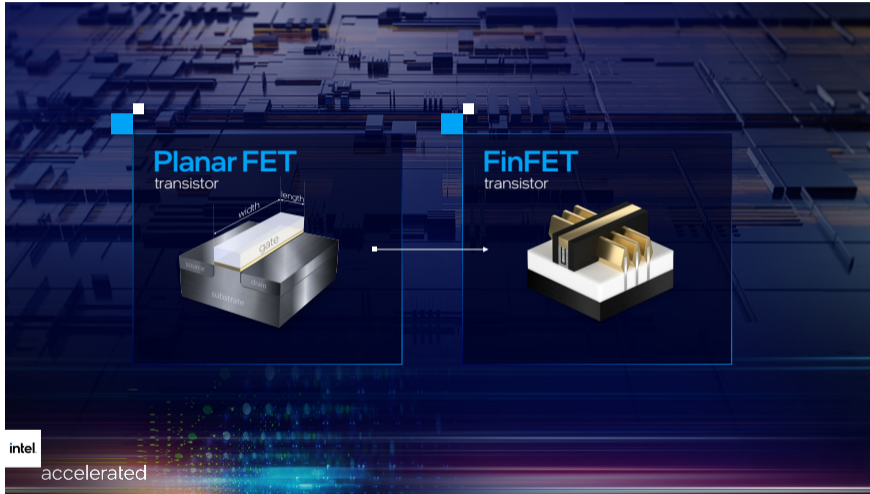


Inverter



SRAM



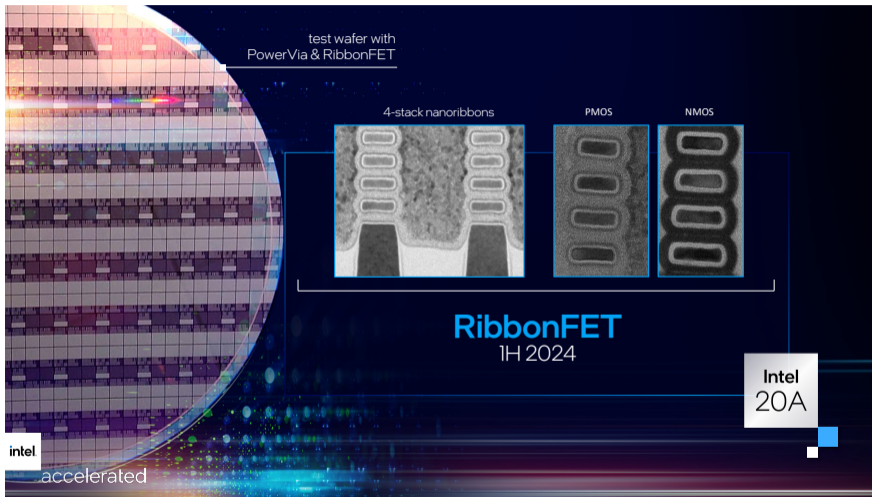


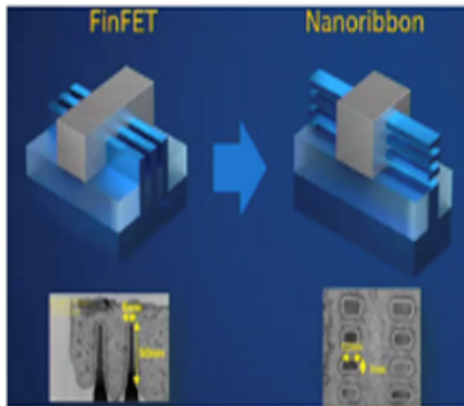
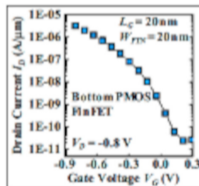
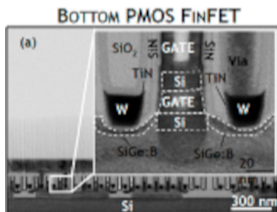
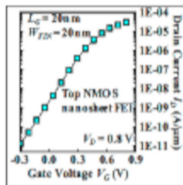
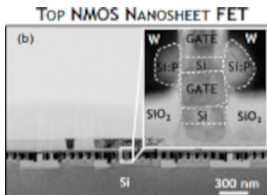
intel

accelerated

www.intel.com/content/www/us/en/newsroom/resources/press-kit-accelerated-event-2021.html



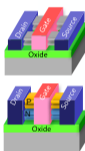
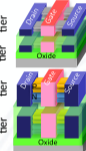
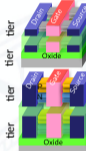
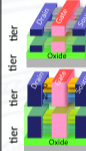
Trends / Forschung (cont.)





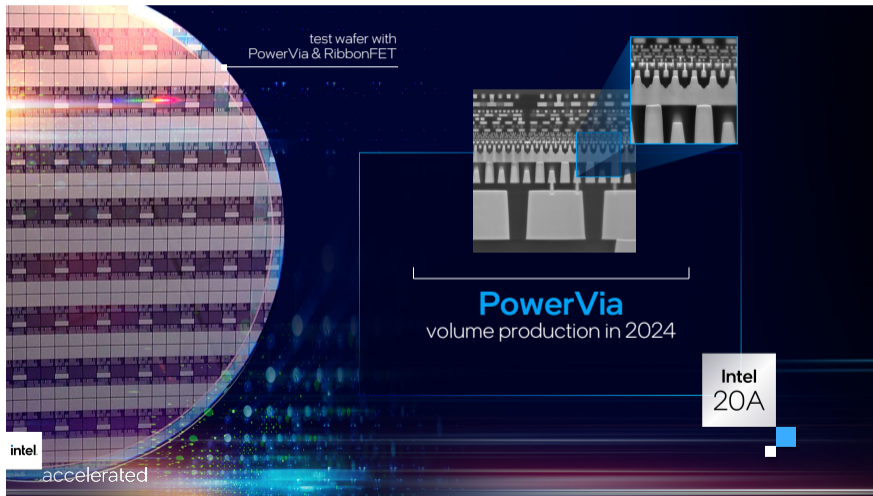
- ▶ IRDS: irds.ieee.org/editions/2022
- ▶ IMEC: www.imec-int.com/en/reading-room imec-publications.be
- ▶ Intel: www.intel.com/content/www/us/en/newsroom

Table MM01 - More Moore - Logic Core Device Technology Roadmap

YEAR OF PRODUCTION	2022	2025	2028	2031	2034	2037
Logic industry "Node Range" Labeling	G48M24	G45M20	G42M16	G40M16/T2	G38M16/T4	G38M16/T6
Fine-pitch 3D integration scheme	"3nm"	"2nm"	"1.5nm"	"1.0nm eq"	"0.7nm eq"	"0.5nm eq"
Logic device structure options	Stacking	Stacking	Stacking	3DVLSI	3DVLSI	3DVLSI
Logic device structure options	finFET LGAA	LGAA	LGAA CFET-SRAM	LGAA-3D CFET-SRAM	LGAA-3D CFET-SRAM	LGAA-3D CFET-SRAM
Platform device for logic	finFET	LGAA	LGAA CFET-SRAM	LGAA-3D CFET-SRAM-3D	LGAA-3D CFET-SRAM-3D	LGAA-3D CFET-SRAM-3D
						
V _{dd} (V)	0,70	0,65	0,65	0,60	0,60	0,60
Gate length (nm)	16	14	12	12	12	12
Number of stacked tiers [1]	1	1	1	2	4	6
Number of stacked nanosheets in logic device [1]	1	3	3	4	4	4
Number of stacked nanosheets in SRAM device [1]	1	3	6	8	8	8
Maximum number of stacked nanosheets in one device [1]	8	8	16	16	32	32
Digital block area scaling	1,00	0,74	0,55	0,26	0,13	0,08
Digital block energy scaling	1,00	0,81	0,72	0,56	0,50	0,49
#MAC units in SoC - based on integration capacity	8192	11038	14980	30966	65191	108652
Cell height (nm) - HD	144	114	90	80	80	72
CPU frequency (GHz)	3,18	3,28	3,36	3,42	3,47	3,50
CPU frequency at constant power density (GHz)	3,18	3,17	2,79	1,49	0,71	0,44
Power density scaling	1,00	1,03	1,20	2,29	4,85	7,99
TOPS/mm ² scaling	1,00	1,39	1,93	4,07	8,68	14,62
TOPS/W scaling	1,00	1,23	1,39	1,79	1,99	2,03
TOPS/mm ² * TOPS/W	1,00	1,71	2,70	7,29	17,24	29,72

[IRDS22]

► rückseitige Spannungsversorgung



▶ 3D gestapelte ICs

test wafer with Foveros Omni

top die

base die

Cu columns

Foveros Omni
enables flexible design
with maximum performance

- TSV penalty minimized
- power and IO optimization
- high bandwidth interconnects

intel
accelerated

The diagram illustrates the Foveros Omni 3D stacked IC technology. On the left, a circular inset shows a test wafer with a grid of dies. The main diagram shows a cross-section of the stack with a top die and a base die connected by copper (Cu) columns. Below this, a central box highlights the technology's benefits: enabling flexible design with maximum performance, minimizing TSV penalty, optimizing power and IO, and providing high bandwidth interconnects. The Intel logo and 'accelerated' branding are visible at the bottom left.

The diagram illustrates the Foveros Direct technology. It features a central cross-sectional view of a top die bonded to a base die. The top die is shown with a grid of bumps. A blue callout box on the right provides a magnified view of the bonding interface, showing the top die, a bonded Cu interconnect, and the base die. Below the central view, the text 'Foveros Direct' is followed by 'direct copper-to-copper bonding which enables low resistance interconnects'. Two blue boxes below this text state 'bump density increases to 10K/mm²' and 'functional block level partitioning'. The Intel logo and the word 'accelerated' are visible in the bottom left corner of the slide.

top die

base die

top die

bonded Cu interconnect

base die

Foveros Direct
direct copper-to-copper bonding
which enables low resistance interconnects

bump density
increases to 10K/mm²

functional block
level partitioning

intel
accelerated

Zukünftige Technologien

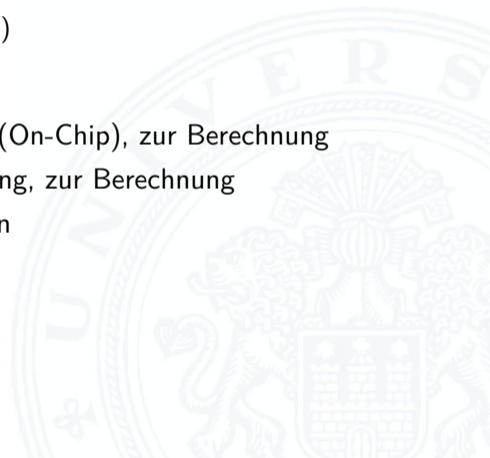
▶ neue Speicher

- ▶ Ferroelectric RAM (FeRAM),
- ▶ Spin-transfer torque-magnetic RAM (STT-MRAM)
- ▶ Resistive RAM (ReRAM)

...

- ▶ Photonik: Licht zur Datenübertragung (On-Chip), zur Berechnung
- ▶ Spintronik: Kernspin zur Datenspeicherung, zur Berechnung
- ▶ Quantencomputing: neue Programmierparadigmen

...

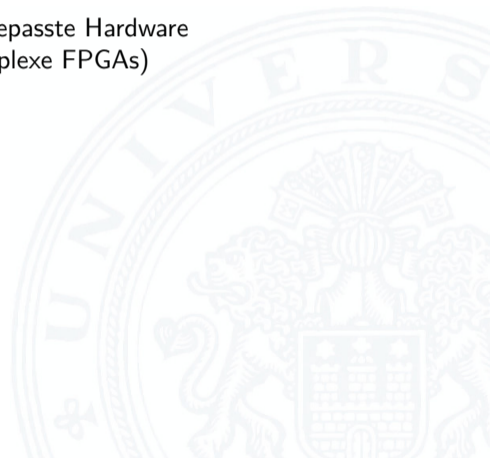




Kompromiss zwischen fest aufgebauter Hardware und Software-basierten Lösungen

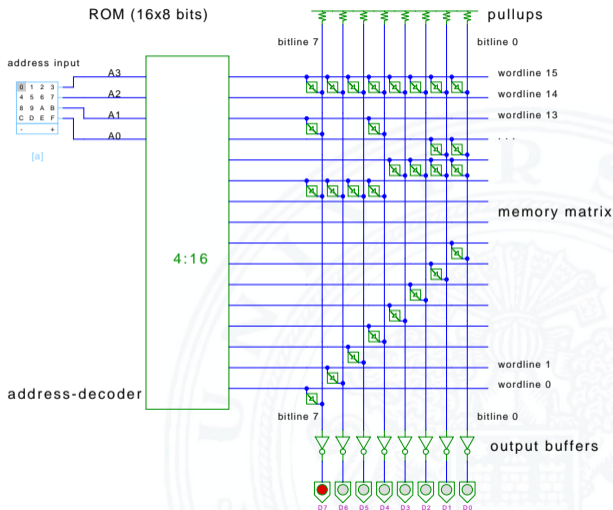
- ▶ Realisierung anwendungsspezifischer Funktionen und Systeme
 - ▶ vom Anwender (evtl. mehrfach) programmierbar
 - ▶ gute bis sehr gute Performanz \Rightarrow an Aufgabe angepasste Hardware
 - ▶ Entwurfsaufwand: einfach (PROM) ... hoch (komplexe FPGAs)

- ▶ Klassifikation nach Struktur und Komplexität
 - ▶ PROM Programmable Read-Only Memory
 - ▶ PAL Programmable Array Logic
 - ▶ GAL Generic Array Logic
 - ▶ PLA Programmable Logic Array
 - ▶ CPLD Complex Programmable Logic Device
 - ▶ FPGA Field-Programmable Gate Array
 - ▶ ...



PROM: Programmable Read-Only Memory

- ▶ UND-ODER Struktur
- ▶ UND-Array
 - ▶ fest
 - ▶ voll auscodiert: 2^n Terme
- ▶ ODER-Terme
 - ▶ programmierbar
- ▶ Hades Beispiel: $n = 4$, 16×8 bit
- ▶ Erweiterungen
 - ▶ „LUT“ (look-up table)
 - ▶ mehrfach programmierbar
 - EPROM (erasable)
 - EEPROM (electric erasable)

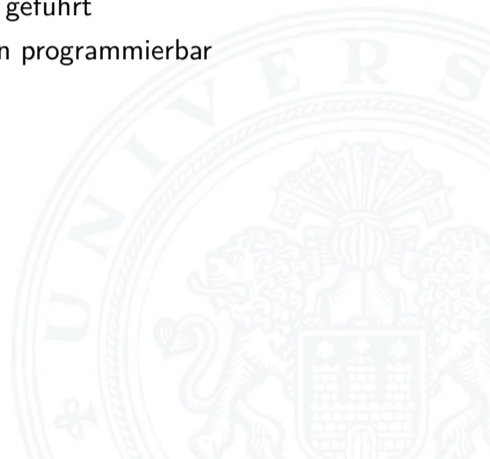


[HenHA] Hades Demo: 40-memories/20-rom/rom



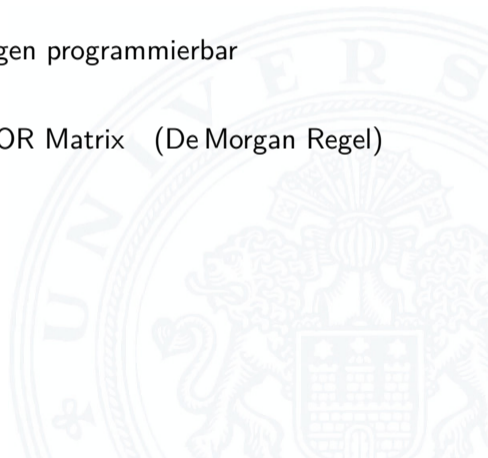
- ▶ disjunktive Form: UND-ODER Struktur
- ▶ UND-Ausgänge fest an die ODER-Eingänge angeschlossen
- ▶ Eingänge direkt und invertiert in die UND-Terme geführt
- ▶ Verknüpfungen der Eingänge zu den UND-Termen programmierbar

- ▶ heute durch GAL ersetzt (s.u.)

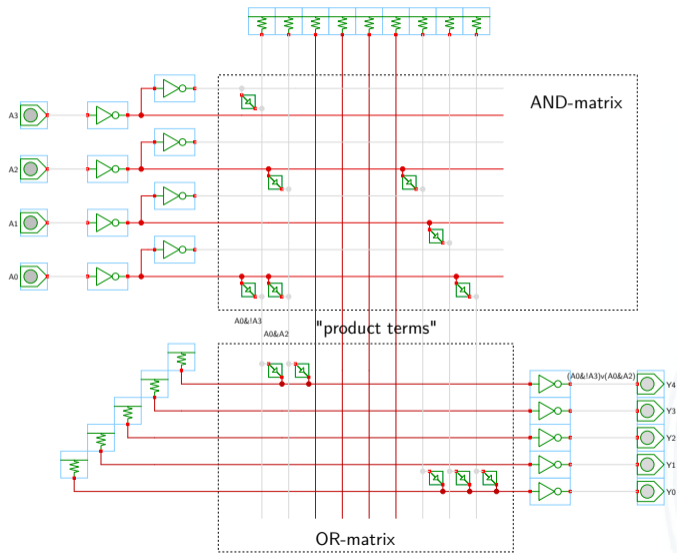




- ▶ disjunktive Form: logische UND-ODER Struktur
 - ▶ Eingänge direkt und invertiert in die UND-Terme geführt
 - ▶ Verknüpfungen Eingänge UND-Terme
 - ▶ Verknüpfungen UND-Ausgänge zu ODER-Eingängen programmierbar
-
- + in NMOS-Technologie sehr platzsparend: NOR-NOR Matrix (De Morgan Regel)
 - statischer Stromverbrauch
 - in CMOS-Technologie kaum noch verwendet



PLA: Programmable Logic Array (cont.)

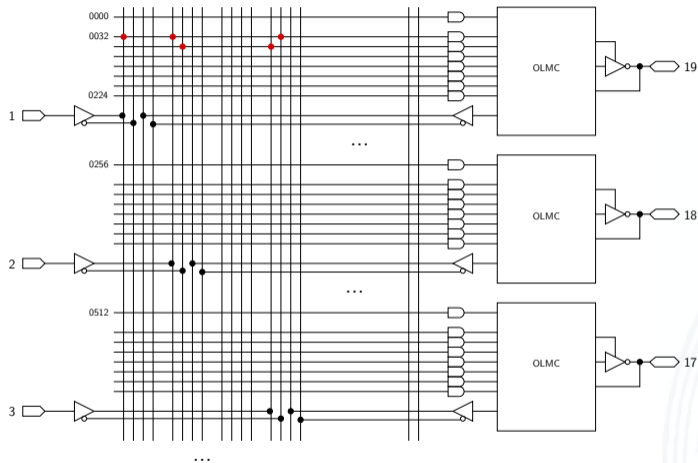




- ▶ disjunktive UND-ODER Struktur
- ▶ externe Eingänge und Ausgangswerte direkt/invertiert
- ▶ „Fuses“ verbinden Eingangswerte mit den AND-Termen

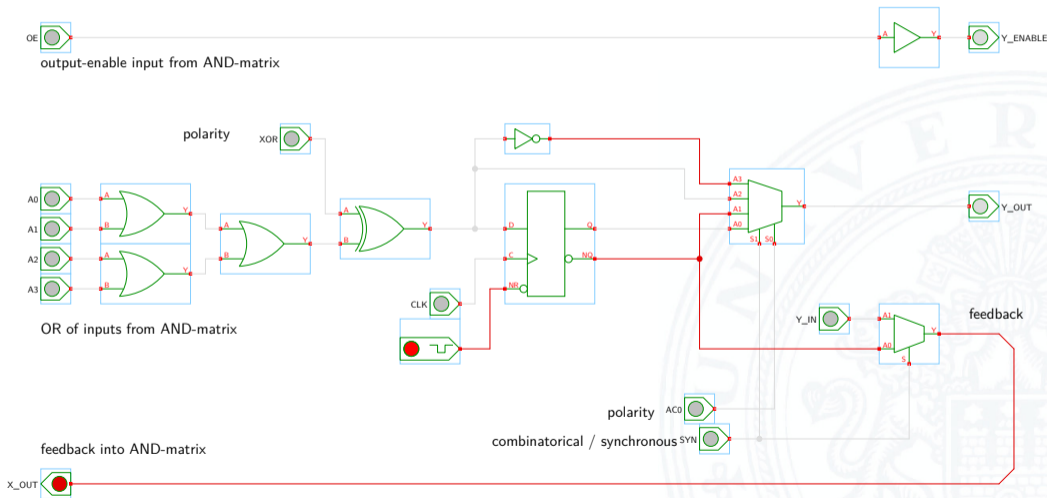
- ▶ programmierbare Ausgabezellen (OLMC) mit je einem D-Flipflop
- ▶ Output-Enable über AND-OR Matrix steuerbar
- ▶ drei Optionen
 - ▶ synchron/kombinatorisch (Flipflop nutzen oder umgehen)
 - ▶ Polarität des Eingangs (D oder \overline{D} speichern)
 - ▶ Polarität des Ausgangs (Q oder \overline{Q} ausgeben)
- ▶ Rückkopplungen für komplexere Logik oder (einfache) Automaten
- ▶ Beispiel: GAL16V8 mit 8 Ausgabezellen, je 7+1 OR-Terme pro Ausgabezelle, 32 Eingänge pro Term

GAL: Blockschaltbild (Ausschnitt)

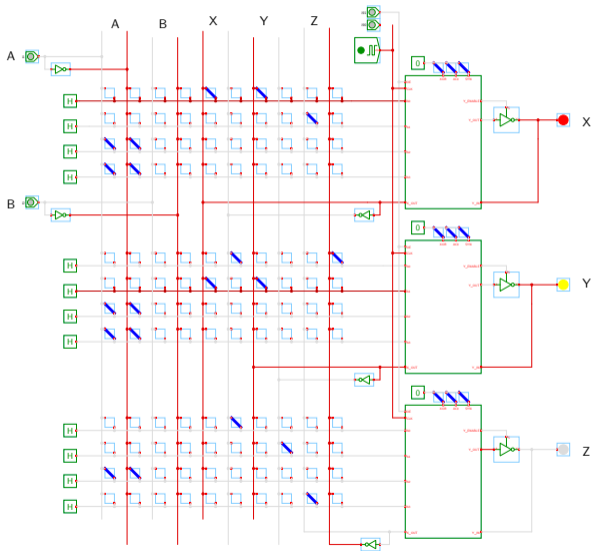


- ▶ programmierbare Sicherungen durchnummeriert
- ▶ kompakte Darstellung der UND-Terme: je eine Zeile
- ▶ Beispiel: zweiter Term (Zeile 0032) $y = 1 \vee 2 \vee \bar{3}$

OLMC: Output-Logic-Macrocell



GAL: Beispiel Ampel





Sammelbegriff für „komplexere“ anwenderprogrammierbare Schaltungen

- ▶ Matrix von kleineren programmierbaren Zellen, typisch:
 - ▶ SRAM als Lookup für Funktionen
 - ▶ programmierbare Register
 - ▶ Carry-Lookahead Logik
- ▶ Multiplexer-Netzwerk als programmierbare Verbindung
- ▶ zusätzliche „Makrozellen“
 - ▶ Multiplizierer, FP-Recheneinheiten, KI-Beschleuniger
 - ▶ eingebettete Prozessorkerne
 - ▶ A/D + D/A Wandler
- ▶ IO-Zellen
 - ▶ schnelle serielle Kommunikation
 - ▶ PLLs (programmierbare Taktgeneratoren)
- ▶ generierte Komponenten: ROM, RAM, FIFO ...

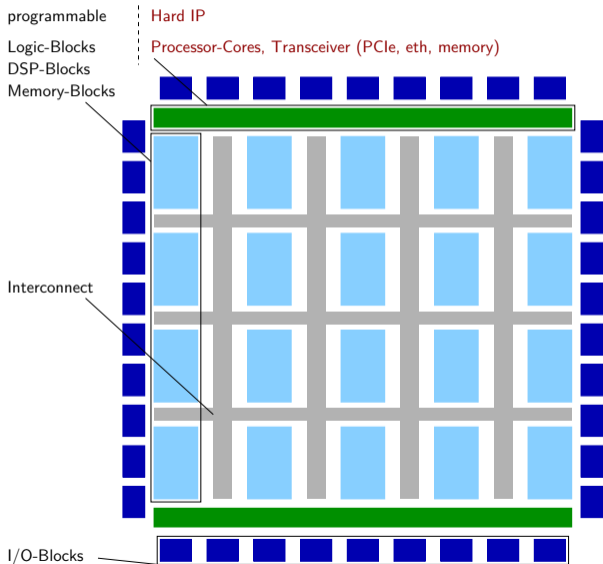


- ▶ fertige IP-Blöcke („*Intellectual Property*“), wie Programmbibliotheken nutzen
 - ▶ Netzwerkprotokolle (Ethernet ...)
 - ▶ Bussysteme (PCIe ...)
 - ▶ Prozessor- und DSP-Kerne
 - ▶ Speichercontroller ...
- ▶ Komplexität
 - ▶ $\approx 3\,000$ nutzbare I/O
 - ▶ ≈ 40 Mio. Register (1-bit)
 - ▶ $\approx 2,0$ GHz, 7 nm Prozesstechnik
 - ▶ ≈ 92 Mrd. Transistoren
- ▶ Hersteller
 - ▶ Xilinx Inc. (Advanced Micro Devices, Inc.)
 - ▶ Intel Corporation (ex. Altera)
 - ▶ Microchip Technology Inc.
 - ▶ Lattice Semiconductor Corporation
 - ▶ Achronix Semiconductor Corporation ...

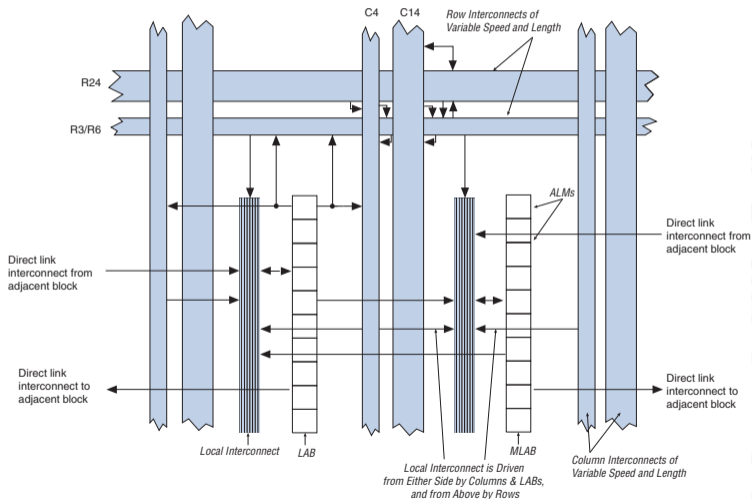




FPGA Struktur: programmierbare + fest vorgebene Bereiche

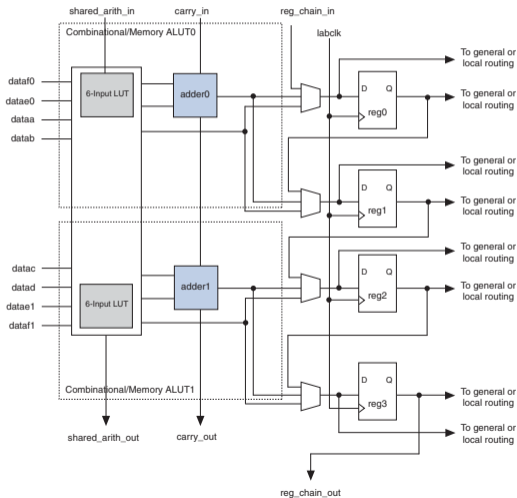


► Verbindungsnetzwerk



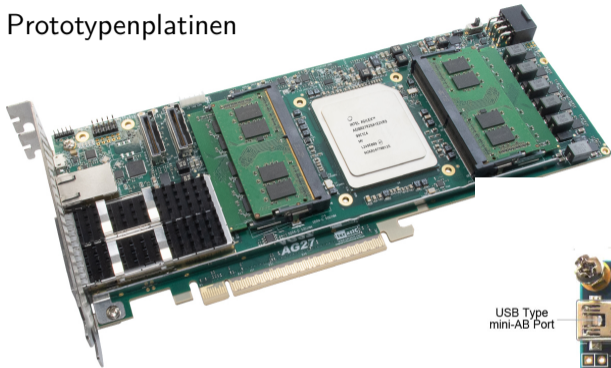
[Intel FPGA]

► programmierbarer Block



[Intel FPGA]

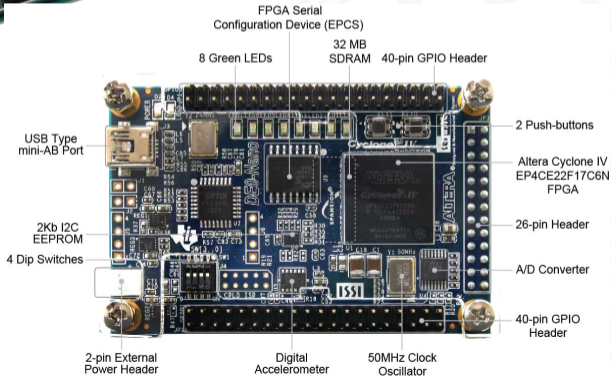
► Prototypenplatten



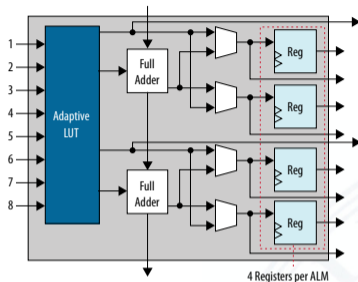
z.B.: Terasic Inc. www.terasic.com.tw

von a.A. (> 15 000 \$)

bis < 100 \$

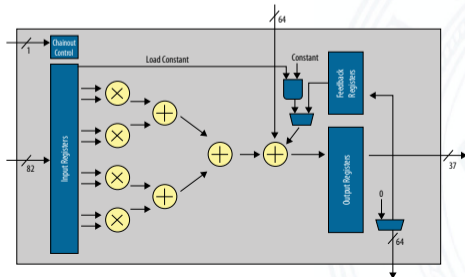


► Logic-Block

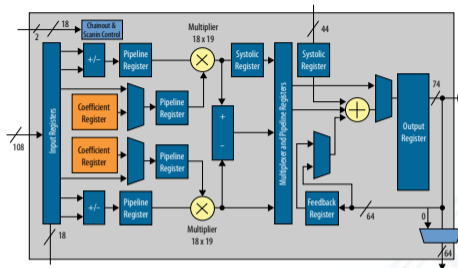


[Intel Agilex]

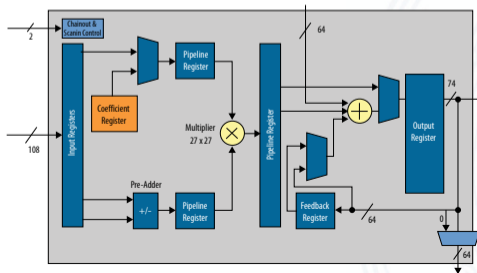
► DSP-Block



► DSP-Block



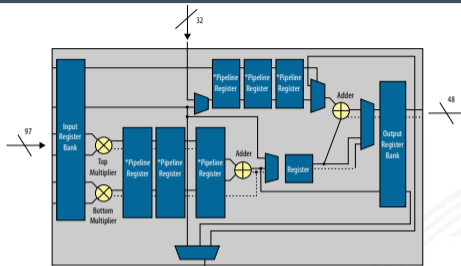
Standard Precision Fixed Point Mode



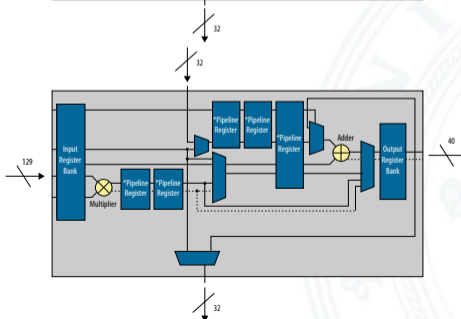
High Precision Fixed Point Mode

► DSP-Block

Half Precision Floating Point (16-bit)



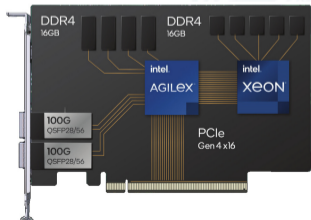
Single Precision Floating Point (32-bit)



► Intel Agilex

[Intel FPGA]

Intel Agilex F-Series Device Names	Logic Elements (LE)	eSRAM Blocks	eSRAM Mbits	M20K Blocks	M20K Mbits	MLAB Counts	MLAB Mbits	Variable Precision DSP Blocks	18x19 Multipliers
AGF 004	392,000	0	0	1,900	38	6644	4.3	1,640	2.3K
AGF 006	573,480	0	0	2,844	56	9720	6.2	1,640	3.3K
AGF 008	764,640	0	0	3,792	74	12960	8.3	2,296	4.6K
AGF 012	1,200,000	2	36	5,568	110	20338	13	4,000	8K
AGF 014	1,437,240	2	36	7,110	139	24,360	15.6	4,510	9K
AGF 022	2,200,000	0	0	11,616	210	37288	21	6,250	12.5K
AGF 027	2,692,760	0	0	13,272	259	45,640	29.2	8,736	17K



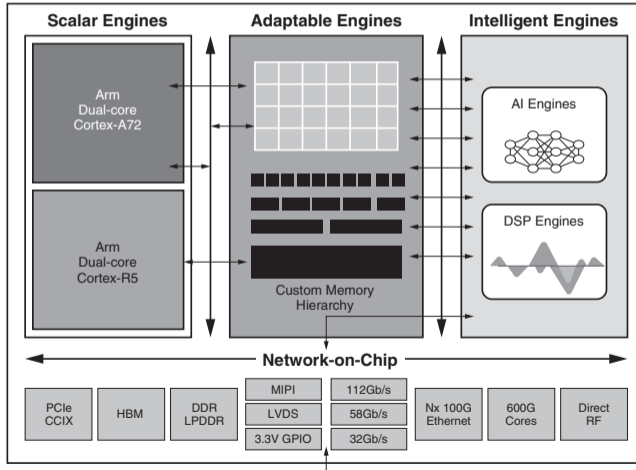
► Xilinx Versal

[AMD FPGA]

		VP1002	VP1052	VP1102	VP1202	VP1402	VP1502	VP2502	VP1552	VP1702	VP1802	VP2802	
Adaptable Engines	System Logic Cells (K)	833	1,186	1,575	1,969	2,233	3,763	3,738	3,837	5,558	7,352	7,326	
	LUTs	380,800	542,080	719,872	900,224	1,020,928	1,720,448	1,708,672	1,753,984	2,540,672	3,360,896	3,349,120	
	NoC Master / NoC Slave Ports	22	22	30	28	42	52	52	52	76	100	100	
Memory	Distributed RAM (Mb)	12	17	22	27	31	53	52	54	78	103	102	
	Total Block RAM (Mb)	19	26	49	47	70	89	89	89	132	174	174	
	UltraRAM (Mb)	97	138	127	190	181	366	366	366	541	717	717	
	Total PL Memory (Mb)	128	181	198	264	282	508	507	509	751	994	994	
	DDR Memory Controllers	2	2	3	4	3	4	4	4	4	4	4	
	DDR Bus Width	128	128	192	256	192	256	256	256	256	256	256	
Intelligent Engines	DSP Engines	1,140	1,572	1,904	3,984	2,672	7,440	7,392	7,392	10,896	14,352	14,304	
	AI Engines Tiles	-	-	-	-	-	-	472	-	-	-	472	
	AI Engine Data Memory (Mb)	-	-	-	-	-	-	118	-	-	-	118	
Scalar Engines	APU	Dual-core Arm® Cortex®-A72, 48KB/32KB L1 Cache w/ parity & ECC; 1MB L2 Cache w/ ECC											
	RPU	Dual-core Arm Cortex-R5F, 32KB/32KB L1 Cache, and 256KB TCM w/ECC											
	Memory	256KB On-Chip Memory w/ECC											
	Connectivity	Ethernet (x2); UART (x2); CAN-FD (x2); USB 2.0 (x1); SPI (x2); I2C (x2)											
Serial Transceivers	GTY Transceivers (32.75Gb/s)	20	20	-	-	-	-	-	-	-	-	-	
	GTYP Transceivers (32.75Gb/s)	-	-	8	28 ⁽¹⁾	8	28 ⁽¹⁾	28 ⁽¹⁾	68 ⁽¹⁾	28 ⁽¹⁾	28 ⁽¹⁾	28 ⁽¹⁾	
	GTM Transceivers ⁽²⁾ (58G (112G))	24 (12)	48 (24)	64 (32)	20 (10)	96 (64)	60 (30)	60 (30)	20 (10)	100 (50)	140 (70)	140 (70)	
Integrated Protocol IP	PCIe® w/DMA & CCIX (CPM4)	2 x Gen4x4	2 x Gen4x4	-	-	-	-	-	-	-	-	-	
	PCIe w/DMA & CCIX (CPM5)	-	-	-	2 x Gen5x8	-	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	2 x Gen5x8	
	PCI Express	1 x Gen4x8	1 x Gen4x8	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	8 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	
	100G Multirate Ethernet MAC	3	5	6	2	6	4	4	4	6	8	8	
	600G Ethernet MAC	2	3	7	1	11	3	3	1	5	7	7	
	600G Interlaken	1	2	0	0	0	1	1	0	2	3	3	
Ordering Information	400G High-Speed Crypto Engine	1	1	3	1	4	2	2	2	3	4	4	
	Extended ⁽³⁾	-1MSE, -1LSE, -2MSE, -2MLE, -2LSE, -2LLE					-1MSE, -1LSE, -2MSE, -2MLE, -2LSE, -2LLE, -3HSE						
	Industrial ⁽³⁾	-1MSI, -1MLI, -1LSI, -1LLI, -2MSI, -2MLI, -2LSI, -2HSI					-1MSI, -1MLI, -1LSI, -1LLI, -2MSI, -2MLI						

⇒ FPGAs gehören zu den komplexesten ICs (≈ 92 Milliarden Transistoren)

- ▶ Rechenbeschleuniger in Workstations und PCs
- ▶ dynamische (Teil-) Konfiguration zur Programmlaufzeit



Xilinx Versal ACAP
(Adaptive Compute Acceleration Platform)
[Xilinx Versal]



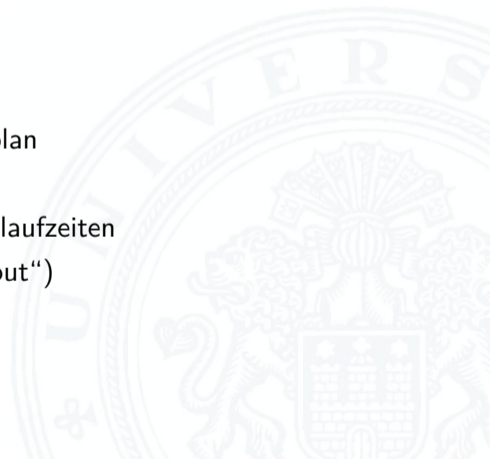
besonders anspruchsvoller Bereich der Informatik

- ▶ Halbleiterfertigung benötigt vorab sämtliche Geometriedaten
 - ▶ spätere Änderungen eines Chips nicht möglich
 - ▶ Durchlauf aller Fertigungsschritte dauert Wochen bis Monate
 - ▶ Entwürfe müssen komplett fehlerfrei sein
 - ▶ spezielle Hardware-/System-Beschreibungssprachen
 - ▶ Simulation des Gesamtsystems
 - ▶ Analyse des Zeitverhaltens
 - ▶ ggf. Emulation/Prototyping mit FPGAs
 - ▶ Kombination von Hardware- oder Softwarerealisierung von Teilfunktionen
- ⇒ **HW/SW-Codesign**



Wasserfallmodell

- ▶ Lastenheft
- ▶ Verhaltensmodell (Software)
- ▶ Aufteilung in HW- und SW-Komponenten
- ▶ funktionale Simulation/Emulation und Test
- ▶ Synthese oder manueller Entwurf der HW, Floorplan
- ▶ Generieren der „Netzliste“ (logische Struktur)
- ▶ Simulation mit Überprüfung der Gatter-/Leitungslaufzeiten
- ▶ Generieren und Optimierung des Layouts („Tapeout“)

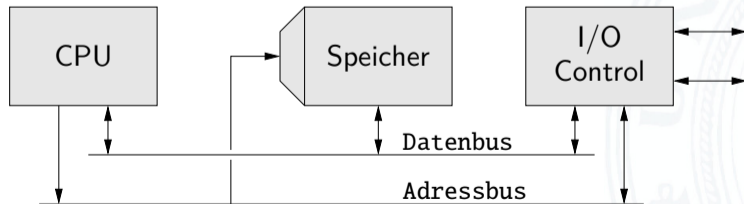


Hardware Abstraktionsebenen

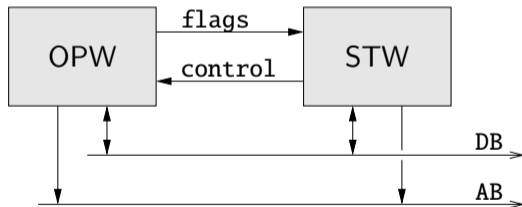
– keine einheitliche Bezeichnung in der Literatur

▶ **Architekturebene**

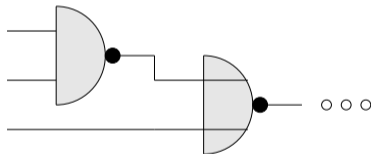
- ▶ Funktion/Verhalten Leistungsanforderungen
- ▶ Struktur Netzwerk
 aus Prozessoren, Speicher, Busse, Controller ...
- ▶ Nachrichten Programme, Protokolle
- ▶ Geometrie Systempartitionierung



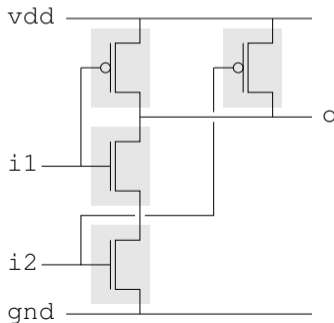
- ▶ Hauptblockebene (Algorithmenebene, funktionale Ebene)
 - ▶ Funktion/Verhalten Algorithmen, formale Funktionsmodelle
 - ▶ Struktur Blockschaltbild
 - aus Hardwaremodule, Busse ...
 - ▶ Nachrichten Protokolle
 - ▶ Geometrie Cluster



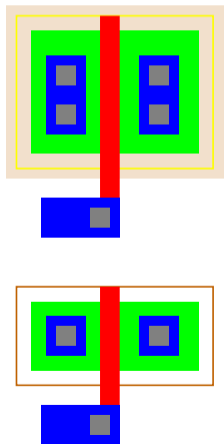
- ▶ Logikebene (Schaltwerkebene)
 - ▶ Funktion/Verhalten Boole'sche Gleichungen
 - ▶ Struktur Gatternetzliste, Schematic
aus Gatter, Flipflops, Latches ...
 - ▶ Nachrichten Bit
 - ▶ Geometrie Moduln



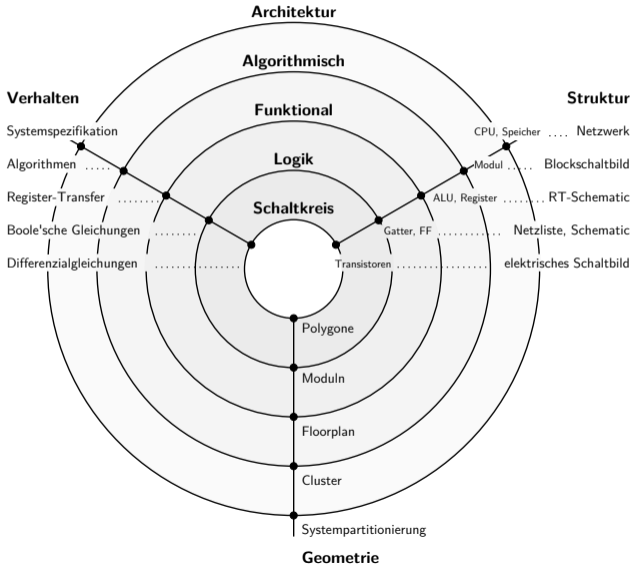
- ▶ elektrische Ebene (Schaltkreisebene)
 - ▶ Funktion/Verhalten Differentialgleichungen
 - ▶ Struktur elektrisches Schaltbild
aus Transistoren, Kondensatoren ...
 - ▶ Nachrichten Ströme, Spannungen
 - ▶ Geometrie Polygone, Layout → physische Ebene



- ▶ physische Ebene (geometrische Ebene)
 - ▶ Funktion/Verhalten partielle DGL
 - ▶ Struktur Dotierungsprofile



Y-Diagramm





drei unterschiedliche Aspekte/Dimensionen:

- 1 Verhalten
- 2 Struktur (logisch)
- 3 Geometrie (physisch)

- ▶ Start möglichst abstrakt, als Verhaltensbeschreibung
- ▶ Ende des Entwurfsprozesses ist das vollständige IC Layout für die Chipfertigung, Prüfmuster für Tests auf Fertigungsfehler oder Selbsttestmechanismen im IC
- ... und die (erfolgreich) simulierte Netzliste mit Gatter- und Leitungsverzögerungen

- ▶ Entwurfsprogramme („EDA“, *Electronic Design Automation*) dabei notwendig: setzen Verhalten in Struktur und Struktur in Geometrien um, ...



Was ist die „beste“ Realisierung einer gewünschten Funktionalität?

- ▶ viele konkurrierende Kriterien
 - ▶ Performanz: Durchsatz, Taktfrequenz, Reaktionszeit, ...
 - ▶ Leistungsaufnahme, Abwärme, ...
 - ▶ Chipfläche, Größe, Umgebungsbedingungen (Temperatur, Vibration, Strahlung)
 - ▶ Stückkosten vs. Entwurfsaufwand und Entwurfskosten
 - ▶ Zeitbedarf bis zur ersten Auslieferung und ggf. für Designänderungen
 - ▶ Schutz von Intellectual-Property
 - ▶ ...

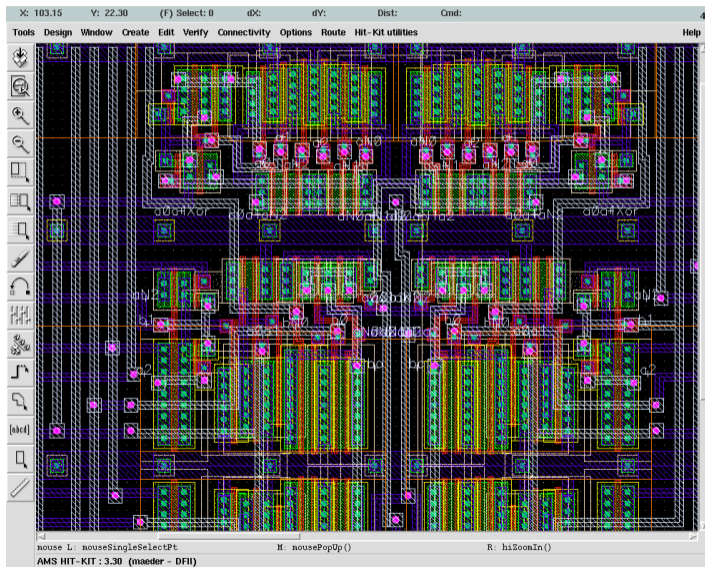
- ▶ vier gängige Varianten
 - ▶ Full-custom Schaltungen
 - ▶ Semi-custom Bausteine: Standardzellen, Gate-Arrays
 - ▶ Anwenderprogrammierbare Bausteine: FPGA, PAL/GAL, ROM
 - ▶ Software auf von-Neumann Rechner: RAM, ROM



- ▶ vollständiger Entwurf der gesamten Geometrie eines Chips
 - ▶ jeder Transistor einzeln „maßgeschneidert“ und platziert
 - ▶ vorgegeben sind lediglich die Entwurfsregeln (*design-rules*) des Herstellungsprozesses (Strukturbreite, Mindestabstände usw.)
 - ▶ oft Verwendung von Teilschaltungen/Makros des Herstellers

 - ▶ minimale Fläche, beste Performanz, kleinster Stromverbrauch
 - ▶ geringste Stückkosten bei der Produktion
 - ▶ aber höchste Entwurfs- und Maskenkosten
 - ▶ erste Prototypen erst nach Durchlaufen aller Fertigungsschritte
- ⇒ nur bei Massenprodukten wirtschaftlich \gg 100 000 Stück
z.B. Speicherbausteine (SRAM, DRAM), Prozessoren, GPUs, Mikrocontroller etc.

Full-custom / „Vollkunden-Entwurf“ (cont.)



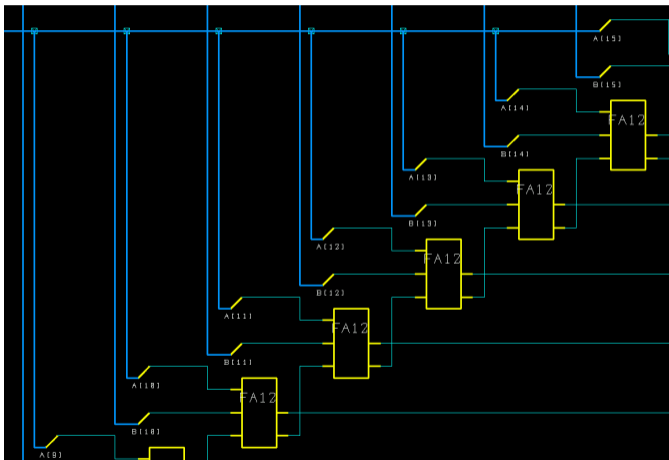


- ▶ Entwurf der Schaltung mit vorhandenen Grundkomponenten
 - ▶ Basisbibliothek mit Gattern und Flipflops
 - ▶ teilweise (konfigurierbare) ALUs, Multiplizierer
 - ▶ Generatoren für Speicher
 - ▶ Entwurfsregeln sind der Bibliothek berücksichtigt
 - ▶ Platzierung und Verdrahtung der Komponenten

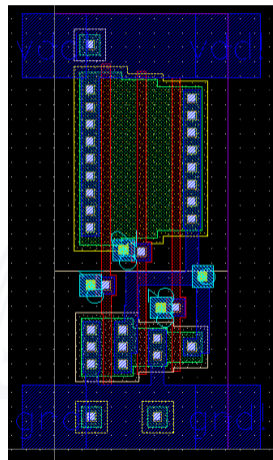
 - ▶ kleine Chipfläche, gute Performanz, niedriger Stromverbrauch
 - ▶ geringe Stückkosten
 - ▶ hohe Maskenkosten (alle Masken erforderlich)
 - ▶ erste Prototypen erst nach Durchlaufen aller Fertigungsschritte
- ⇒ nur bei größeren Stückzahlen wirtschaftlich \gg 10 000 Stück

Semi-custom: Standardzell-Entwurf (cont.)

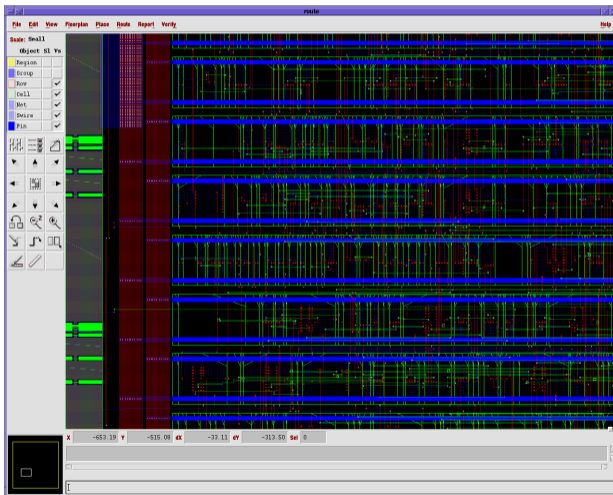
Schematic



Zell-Layout



Standardzell Layout





- ▶ „vorgefertigte“ Schaltung mit Gattern/Transistoren an festen Positionen
- ▶ Entwurf durch Verdrahten der vorhandenen Transistoren
- ▶ überzählige Transistoren werden nicht angeschlossen

- ▶ mittlere Chipfläche, Performanz und Stromverbrauch
- ▶ mittlere Stückkosten
- ▶ mittlere Maskenkosten (nur Verdrahtung kundenspezifisch)
- ▶ Prototypen schneller verfügbar als Standardzellrealisierung, es fehlen nur die Fertigungsschritte der Verdrahtung (= Metalllayer und Kontakte)

- ⇒ ab mittleren Stückzahlen wirtschaftlich $> 1\,000$ Stück
 - ▶ werden von großen FPGAs verdrängt



- ▶ mehrere Millionen konfigurierbarer Funktionsblöcken (LUTs + Register)
 - ▶ Verschaltung dieser Blöcke vom Anwender programmierbar
 - ▶ Entwurfsprogramme setzen Beschreibung des Anwenders auf die Hardware-Blöcke und deren Verschaltung um
 - ▶ spezielle IP-Komponenten, Prozessorkerne etc.
 - ▶ entsprechen > 20 Mio. Gatter-Äquivalente
 - ▶ Taktfrequenzen bis in GHz, Bereich, typisch 100 MHz-Bereich
 - ▶ Hersteller: Xilinx (jetzt AMD), Altera (jetzt Intel), Microchip, Lattice, Achronix ...

 - ▶ nicht benutzte Blöcke liegen brach
 - ▶ Schaltung kann in Minuten (im System) neu programmiert/verbessert werden
- ⇒ optimal für geringe Stückzahlen, ca. 10 ... 1 000 Stück



Ideen für einen Mikrochip? Spezialhardware für maschinelles Lernen, Bild- / Signalverarbeitung, „embedded Control“, Parallelverarbeitung usw.

- ▶ Hereinschnuppern: *Projekt 64-189 Entwurf eines Mikrorechners*
 - ▶ eigenen Prozessor mit Befehlssatz etc. entwerfen und auf einer FPGA Prototypenplatine realisieren
 - ▶ Demo-Boards von Intel (Altera) oder Xilinx und Entwurfssoftware sind bei uns am Fachbereich verfügbar
- ⇒ einfach bei TAMS vorbeischaun



- [Mäd11] A. Mäder: *Vorlesung: Rechnerarchitektur und Mikrosystemtechnik*.
Universität Hamburg, FB Informatik, 2011, Vorlesungsfolien.
tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/ram
- [HenHA] N. Hendrich: *HADES — HAMBURG DESIGN SYSTEM*.
Universität Hamburg, FB Informatik, Lehrmaterial.
tams.informatik.uni-hamburg.de/applets/hades/webdemos
- [HenCM] N. Hendrich: *CMOS GATES DEMONSTRATION*.
Universität Hamburg, FB Informatik, Lehrmaterial.
tams.informatik.uni-hamburg.de/applets/cmos
- [IRDS22] *International Roadmap for Devices and Systems (IRDS) 2022 Edition*.
IEEE International Roadmap for Devices and Systems, 2022.
irds.ieee.org/editions/2022



- [Rei98] N. Reifschneider: *CAE-gestützte IC-Entwurfsmethoden*.
Prentice Hall, 1998. ISBN 978-3-8272-9550-7
www.ibdr.de/Private/caemain.htm
- [WE93] N.H.E. Weste, K. Eshraghian:
Principles of CMOS VLSI design – A systems perspective.
2nd edition, Addison-Wesley, 1993. ISBN 978-0-201-53376-7
- [MC80] C. Mead, L. Conway: *Introduction to VLSI systems*.
2nd edition, Addison-Wesley, 1980. ISBN 978-0-201-04358-7
- [SS04] W. Schiffmann, R. Schmitz:
Technische Informatik 1 – Grundlagen der digitalen Elektronik.
5. Auflage, Springer-Verlag, 2004. ISBN 978-3-540-40418-7

- [TSG19] U. Tietze, C. Schenk, E. Gamm: *Halbleiter-Schaltungstechnik*.
16. Auflage, Springer-Verlag, 2019. ISBN 978-3-662-48553-8
www.tietze-schenk.de
- [GK83] D.D. Gajski, R.H. Kuhn: *Guest Editors' Introduction: New VLSI Tools*.
in: *IEEE Computer* 16 (1983), December, Nr. 12, S. 11-14. ISSN 0018-9162
- [Intel FPGA] *Intel FPGAs und programmierbare Geräte*
Intel Corp.; Santa Clara, CA. (Altera Corp.; San Jose, CA.)
www.intel.de/content/www/de/de/products/programmable.html
www.altera.com
- [AMD FPGA] *AMD Xilinx – Devices*
Advanced Micro Devices, Inc.; Santa Clara, CA. (Xilinx Inc.; San Jose, CA.)
www.xilinx.com/products/silicon-devices.html
www.amd.com