

Einführung in Computer Microsystems

Sommersemester 2012 – 1. Block

Prof. Andreas Koch



TECHNISCHE
UNIVERSITÄT
DARMSTADT



1. Lernziele und Lerninhalte
2. Literatur
3. Einführung in Computer Microsystems
4. Hardware-Entwurf
5. Fundamentale Entwurfsmethoden
6. Zusammenfassung und Ausblick

Quelle: Modulhandbuch¹ *Einführung in Computer Microsystems*

- ▶ Vermittlung von umfassenden Kompetenzen (Theorie und Praxis) in grundlegenden Aspekten der technischen Informatik.
 - ▶ Modellierung und Design von endlichen Automaten und Datenpfaden sowie deren Simulation und Realisierung mittels Verilog HDL
 - ▶ Logik-Synthese
 - ▶ Modellierung von zeitbehafteten, parallelen Abläufen in Hardware
- ▶ z. B.:
 - ▶ Fundamentale Hardware-Strukturen und ihre Modellierung
 - ▶ Hardware-Beschreibungssprache Verilog HDL
 - ▶ Technik und Technologien von FPGAs
 - ▶ Simulation, Verifikation und Synthese
 - ▶ Modellierung endlicher Automaten, zeitbehaftete Abläufe
 - ▶ Hierarchische Automaten, Statecharts
 - ▶ Anwendungsgebiete für FPGAs - Ausblick

¹http://www.mhb.informatik.tu-darmstadt.de/scripte/ov_all.cgi?lang=de

- [Cil04] Ciletti, Michael D.: *Starter's Guide to Verilog 2001*.
Prentice Hall, 2004.
- [Cil10] Ciletti, Michael D.: *Advanced Digital Design with the Verilog HDL, 2. Auflage*.
Prentice Hall, 2010.
- [Kat94] Katz, Randy H.: *Contemporary Logic Design*.
Addison-Wesley Longman, 1994.

- ▶ Was sind **Microsystems**?
- ▶ Eine Definition ist: Microminiaturized and integrated systems based on microelectronics
- ▶ Im Deutschen wird der Begriff **Mikroelektronik** verwendet.
- ▶ Die Mikroelektronik ist ein Teilgebiet der Elektrotechnik und beschäftigt sich mit der Miniaturisierung von elektronischen Schaltungen.
- ▶ Merkmale sind
 - ▶ Integration (Begriff *Integrierte Schaltung*): Mehrere Bauelemente (Transistoren, Widerstände, Kondensatoren) werden in einem Fertigungsschritt auf einem Träger hergestellt.
 - ▶ Miniaturisierung: Die Abmessung der Bauelemente (Transistoren) wird immer kleiner.
- ▶ Weitere Entwicklungen wie die Mikrosystemtechnik sind daraus abgeleitet worden.

Am Anfang war das Relais und die Röhre...

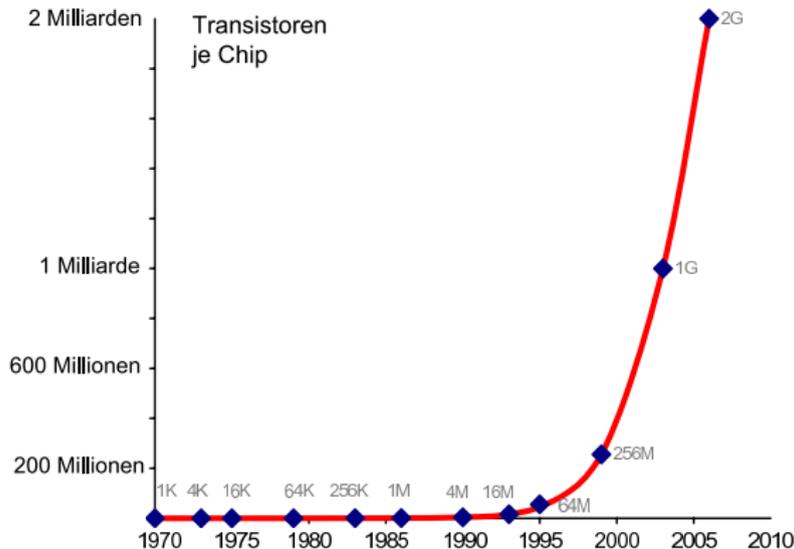
- ▶ 1947/48 Erfindung des Bipolartransistors durch Shockley, Bardeen, Braitain
- ▶ ab 1950 Grundlagenforschung auf dem Gebiet der Bipolarschaltungen
- ▶ ca. 1960 Entwicklung der Silizium-Planartechnik und erste *integrierte Schaltungen* mit ca. 10 Bauelementen
- ▶ ca. 1960 Konzeption des MOS-Transistors durch Khang und Atalla
- ▶ ca. 1970 erster Mikroprozessor
- ▶ und was waren bzw. sind die *Vorteile der integrierten Schaltungen*
 - ▶ Transistoren, Dioden und Widerstände sowie die Verbindung der Bausteine untereinander werden in einem gemeinsamen Herstellungsprozess in einem Silizium-Einkristall integriert.
 - ▶ Höhere Zuverlässigkeit, höhere Schaltgeschwindigkeiten, höhere Packungsdichte.

- ▶ Mikroelektronik umfaßt viele Gebiete:
 - ▶ Elektrotechnik: Transistorschaltungen/Differentialgleichungen
 - ▶ Physik, Verfahrenstechnik/Chemie
- ▶ Warum Mikroelektronik in der Informatik?
 - ▶ Grundverständnis kann nicht schaden
 - ▶ Es geht um Computer Microsystems
 - ▶ Zur Entwicklung von Mikroelektronik wird Software verwendet.
 - ▶ Die Anforderungen an den Entwurf (z. B. Logikminimierung, Platzierung von Bauelementen) werden mit Algorithmen gelöst.
- ▶ Weitere Veranstaltungen zur technischen Informatik dieses Semester
 - ▶ Praktikum Adaptive Computersysteme (FG ESA)
 - ▶ Praktikum Technische Informatik (FG ESA)
 - ▶ Seminar Technische Informatik (FG ESA)
 - ▶ Im FB18: Bei Profs. Evekings und Hofmann

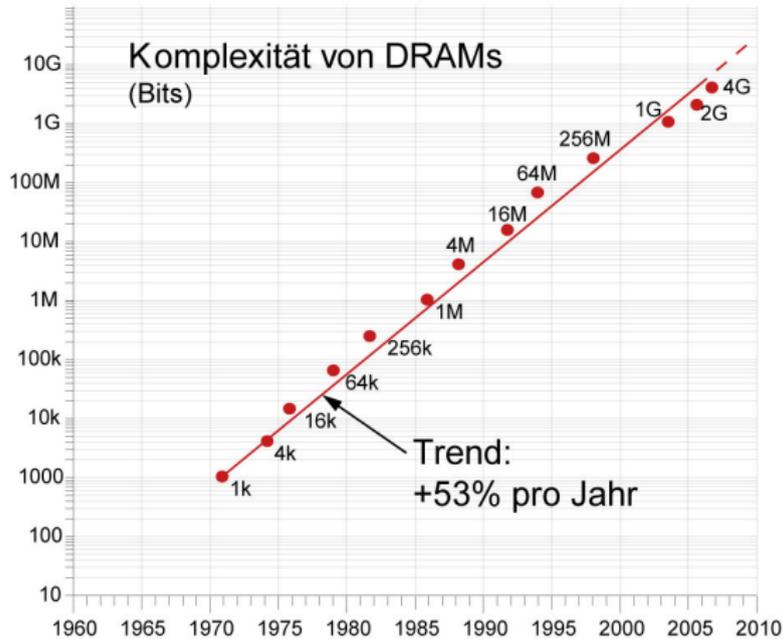
- ▶ Integrierte Schaltungen (ICs), oft auch nur als **Chips** bezeichnet.
- ▶ Chipmarkt: in 2004 Umsatz von 213 Milliarden USD weltweit für 2010 Umsatz von 246 Milliarden USD prognostiziert²
- ▶ Nahezu überall verbaut
 - ▶ Offensichtlich: Rechner
 - ▶ PC, Server, Supercomputer, ...
 - ▶ Versteckt: eingebettete Systeme
 - ▶ Autos, Fernseher, Herzschrittmacher, ...
- ▶ Alleine in Deutschland: 9 Milliarden EUR Chip-Umsatz
 - ▶ Bildet aber Basis für 50x größeren Markt
 - ▶ 3 Millionen Arbeitsplätze
- ▶ Sollte man sich auch als Informatiker genauer anschauen!
- ▶ Ursprung dieser Bedeutung?

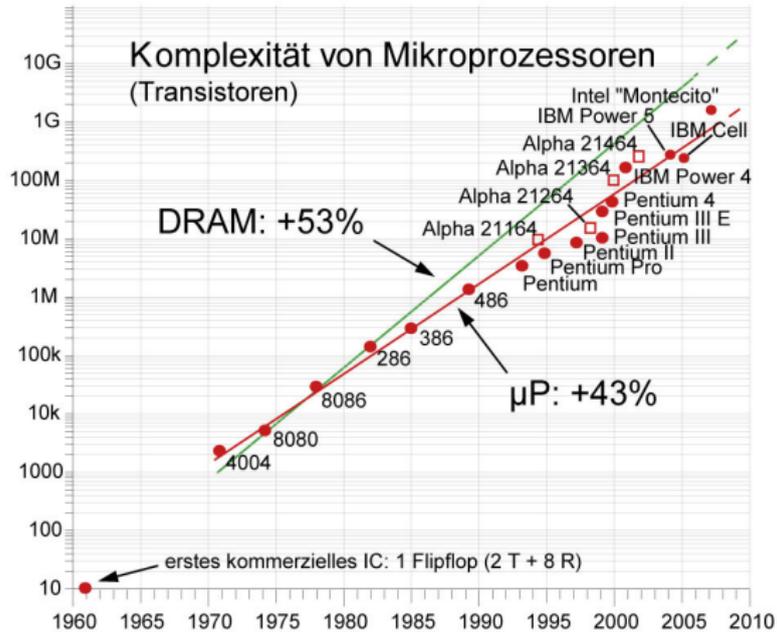
²Financial Times Deutschland: <http://www.ftd.de/it-medien/it-telekommunikation/halbleiterhersteller-auftragsflut-schwappt-ueber-chipbranche/50050042.html>

Moore's Gesetz - Exponentielles Wachstum

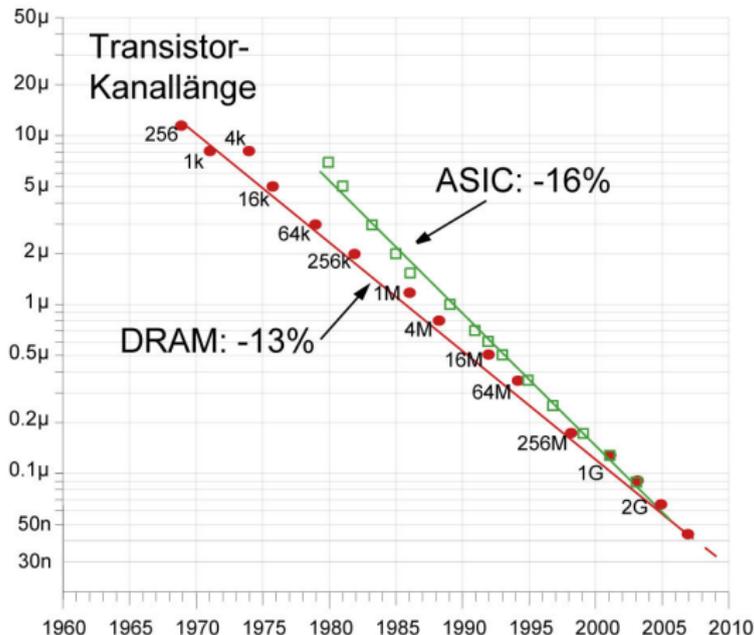


Alle 18 Monate verdoppelt sich die Anzahl der Transistoren auf einem Chip. Dies ist allerdings eine abgewandelte Auslegung.

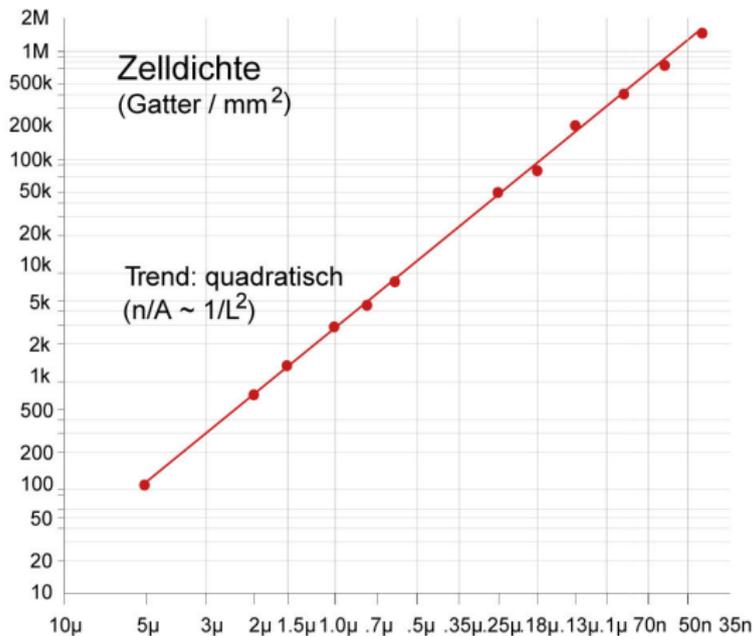




Verbesserung der Fertigungsprozesse Transistor-Kanallängen

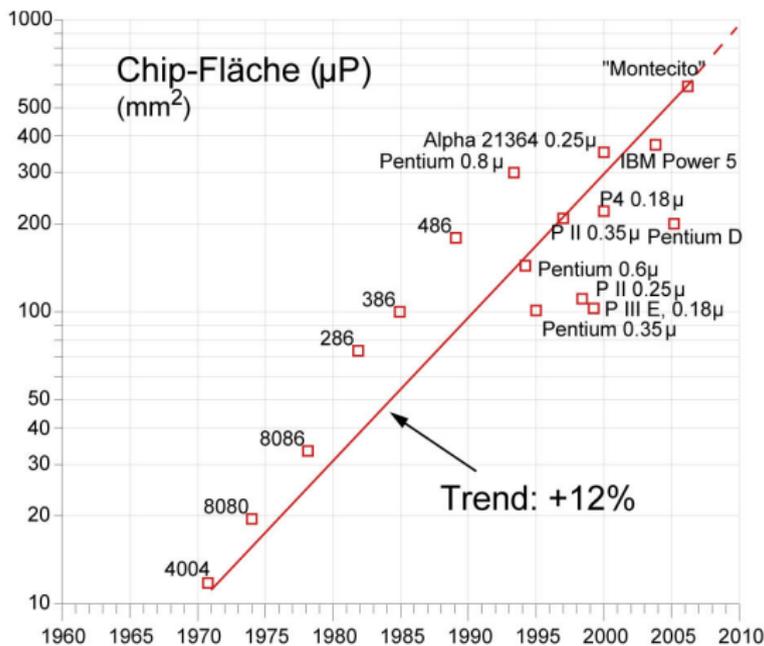


Verbesserung der Fertigungsprozesse Auswirkungen - Transistoren schrumpfen um 13% jährlich



► pro Flächeneinheit $1/0,87^2 = 33\%$ mehr Elemente

Nicht nur kleinere Strukturen, auch größere Chip-Flächen



Ausbeute

Effekte der Chip-Größe

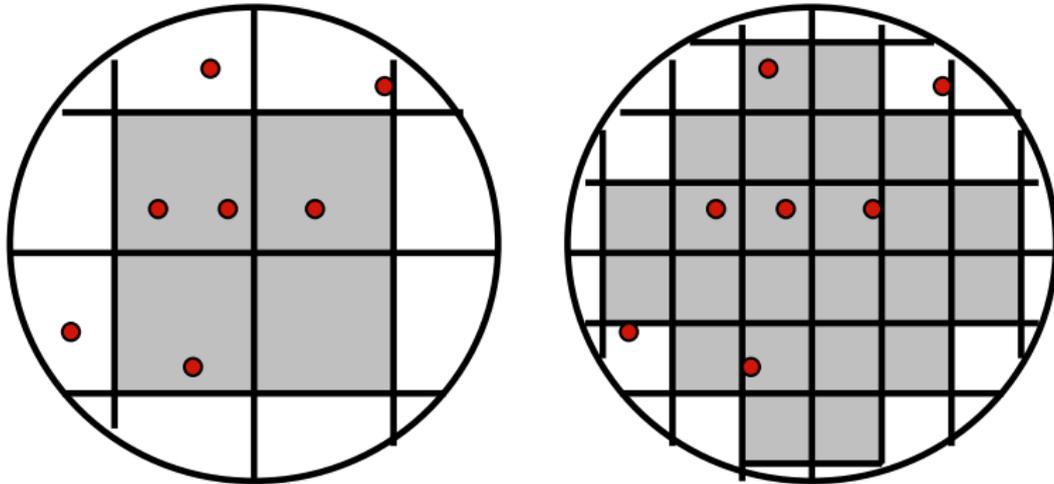


Abbildung: Quelle: J. Rabaey, Digital Integrated Circuits

Heute zuverlässig erreichbar: Nur ca. 1 Fehler pro cm^2 .

Beispiel Cell-Prozessor - Layout

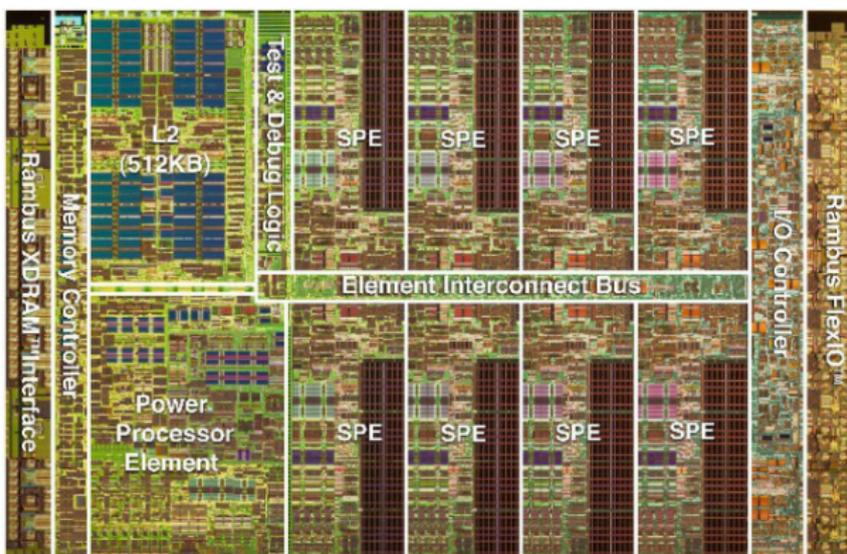
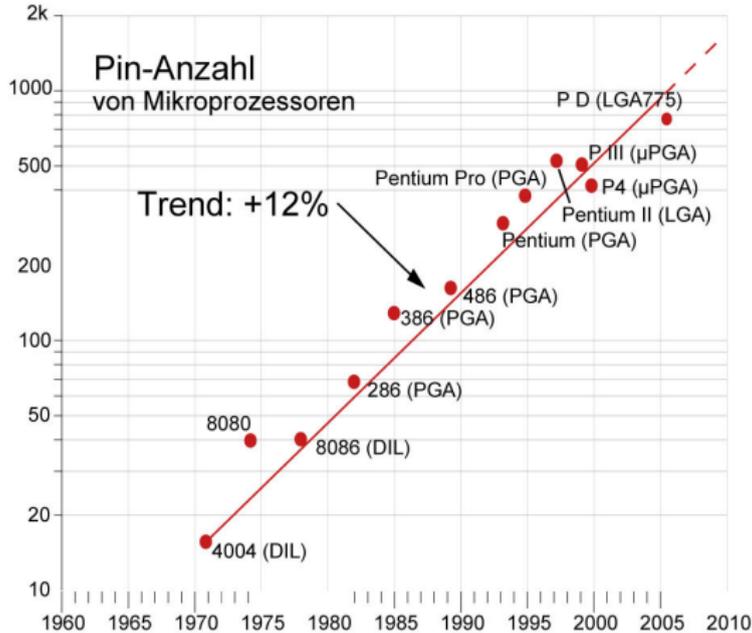


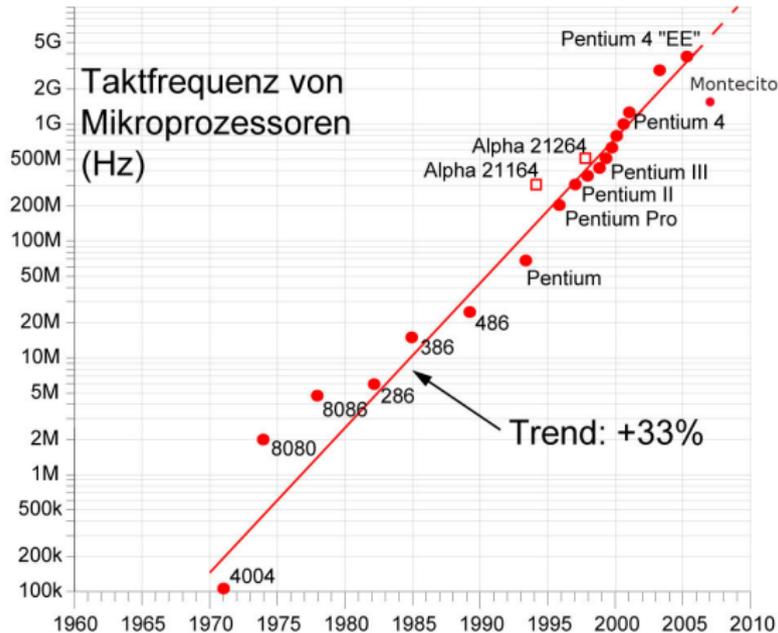
Abbildung: Layout (Die) des Cell-Prozessors, Quelle: IBM

In der PlayStation 3 werden nur sechs SPEs verwendet.



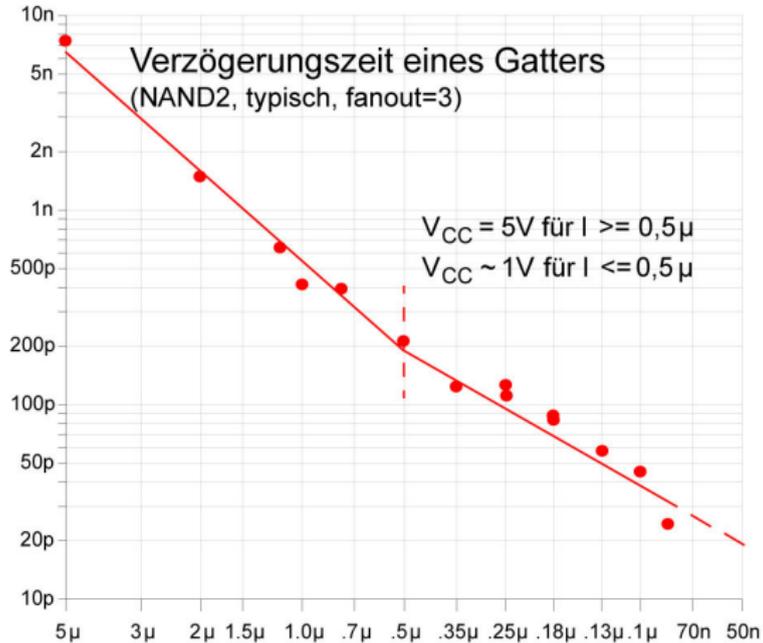
Problem: Komplexität (+53% p. a.) wächst stärker als Kommunikationsmöglichkeit

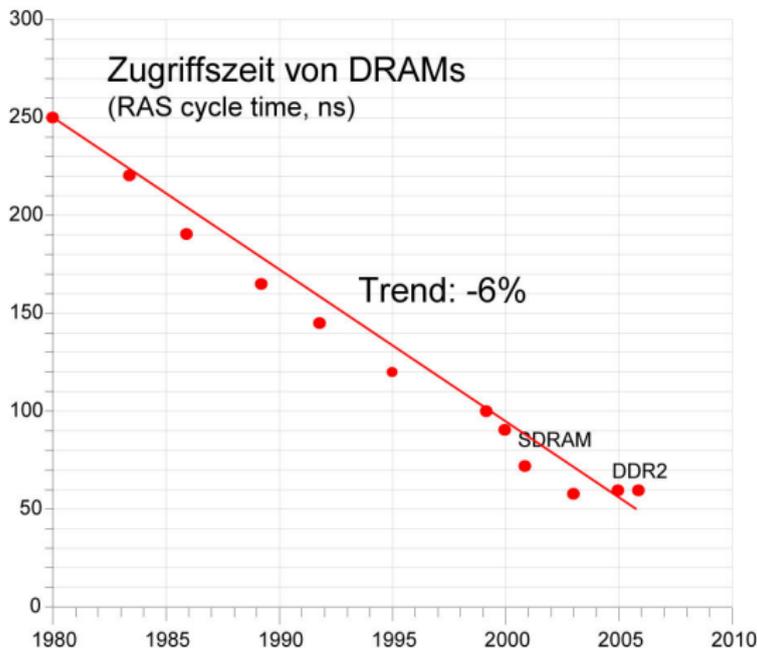
Taktfrequenz - Entwicklung





- ▶ Leistungssteigerung wurde lange Zeit, durch Erhöhen der Taktfrequenz erreicht. Aktuell liegt der Prozessortakt vieler Mikroprozessoren bei 3.x GHz.
 - ▶ Intel Pentium 4 EE, 3,8 GHz Takt, 11,5 SPECint2006
 - ▶ Intel Montecito 9050, 1,6 GHz Takt, 14,5 SPECint2006
- ▶ Bedingt durch die Technologie (CMOS-Technologie) steigt der Leistungsumsatz der Prozessoren mit dem Takt ($P \approx U^2 \cdot f \cdot C_L$).
- ▶ Die entstehende Wärme ist nur mit großem Aufwand abzutransportieren.
- ▶ Parallelrechner:
 - ▶ Integration mehrerer CPUs auf einem Chip (aktuell: 2-12 CPU-Kerne von Intel und AMD verfügbar), Cell B. E. Prozessor von IBM
 - ▶ Massiv parallele Systeme mit mehreren tausend Prozessoren, Spezialarchitekturen z. B. Vektorrechner

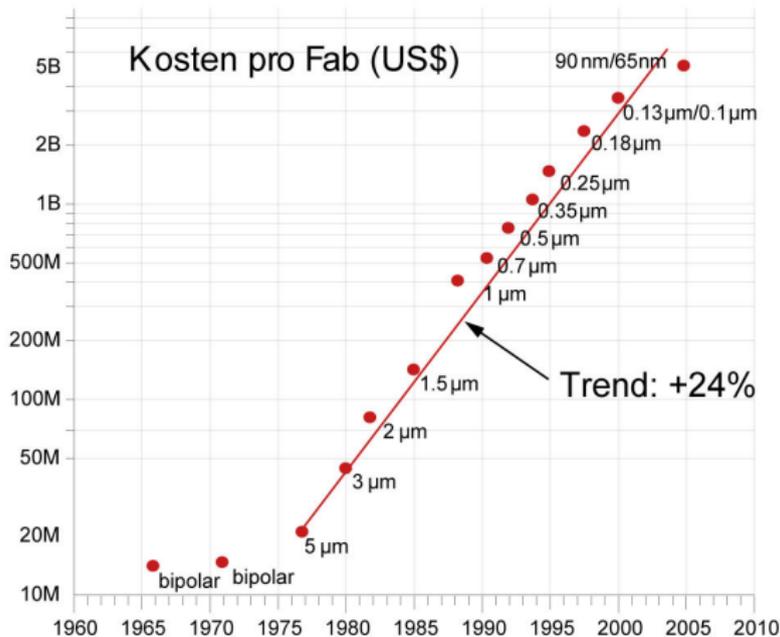




Gatterlaufzeiten liegen im ps-Bereich. Hauptspeicher ist vergleichsweise langsam.

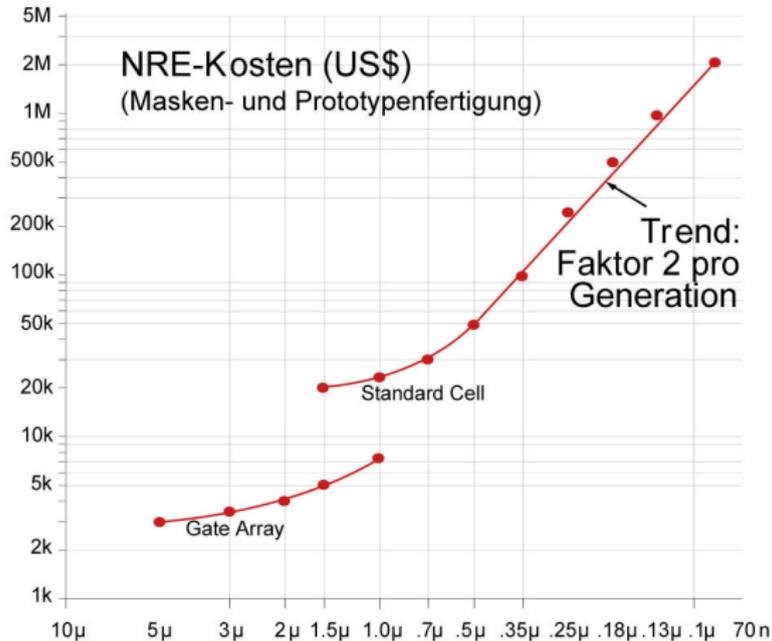


- ▶ +53% p. a. Chip-Komplexität
 - ▶ Zahl der Transistoren, Speichergröße
- ▶ +33% p. a. Packungsdichte
 - ▶ Elemente/Flächeneinheit
- ▶ +33% p. a. Taktfrequenz
- ▶ +12% p. a. Chip-Fläche
- ▶ +12% p. a. mehr Pins (Flaschenhals!)
- ▶ -6% p. a. Speicherzugriffszeit (Flaschenhals!)



In zehn Jahren haben sich die Kosten für eine Fabrik fast verzehnfacht.

Kostenentwicklung II

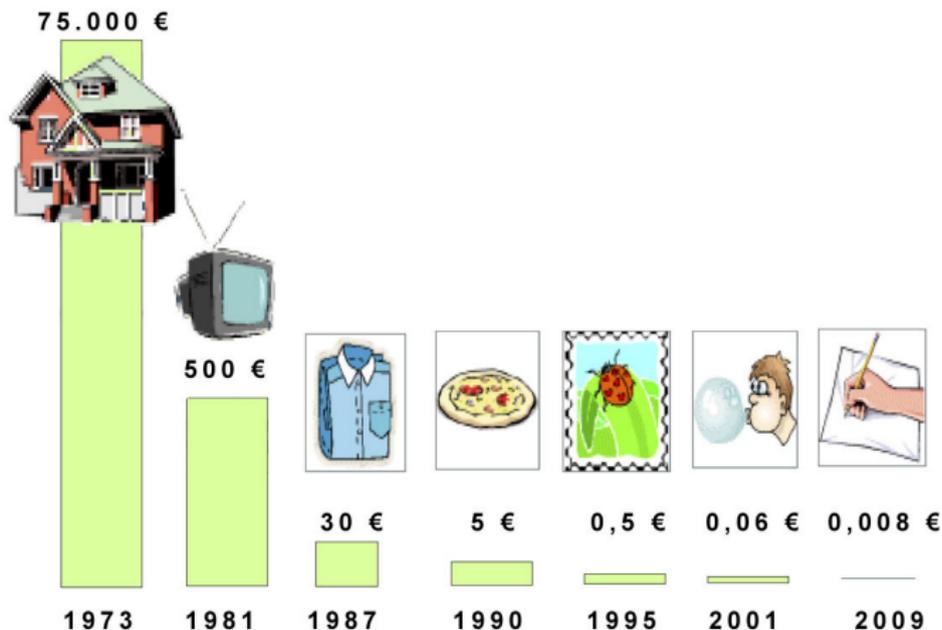




- ▶ Fertigung auf älterer Technologie
 - ▶ 180nm ... 110nm *sehr* weit verbreitet
- ▶ Multi-Projekt-Chips
 - ▶ Sonderangebote für EU Forschung und Lehre
 - ▶ UMC L130E: EUR 30.000/5mm x 5mm
 - ▶ UMC L90N: EUR 45.000/4mm x 4mm
 - ▶ UMC L65: EUR 45.000/4mm x 4mm
 - ▶ Für ca. 45 Chips, Gehäuse kosten extra
- ▶ Programmierbare/konfigurierbare Schaltungen
 - ▶ Keine photochemische Chip-Fertigung mehr erforderlich
 - ▶ Beispiele sind: PALs, PLAs, FPGAs
 - ▶ FPGAs werden gerne zum Rapid-Prototyping eingesetzt.

Kostensenkung durch Massenfertigung

Kosten für 1 Mb DRAM



- ▶ Vergleichbar einem Puzzle mit einer Milliarde Teile
- ▶ Zusammensetzen unter hohem Zeitdruck
 - ▶ Time-to-Market (TTM)
- ▶ Ein einziger Fehler kann Millionen USD kosten
 - ▶ Erneute Chip-Fabrikation (“re-spin”)
 - ▶ Intel Pentium Bug (FDIV): deutlicher Gewinneinbruch

Wie bekommt man die Komplexität eines solchen Entwurfs mit 100 Millionen Transistoren (und mehr) in den Griff?

- ▶ Abstraktere Vorgehensweisen
- ▶ Beschreibe
 - ▶ ... nicht mehr einzelne Transistoren
 - ▶ ... sondern komplette Systeme
- ▶ Vergleichbar Software-Entwicklung
 - ▶ ... statt Assembler
 - ▶ ... Beschreibung von Systemen als interagierende Komponenten (service-oriented architectures)
- ▶ Mittel der Wahl: (Hardware)-Beschreibungssprachen
 - ▶ Sehr abstrakt: MATLAB/Simulink (Signalverarbeitung)
 - ▶ Abstrakt: SystemC
 - ▶ Noch recht hardware-nah: Verilog HDL - VHDL
- ▶ Entkoppeln von Entwurf und technischer Realisierung
- ▶ Umsetzung idealerweise automatisch (Synthese)
 - ▶ Klappt aber noch nicht immer!

Entwurfsebenen 1

Unterschiedlicher Abstraktionsgrad

▶ **Verhaltensebene**

Was soll passieren? Realisierung bleibt offen.

$$y = f(x)$$

▶ **Systemebene**

Grobe Aufteilung von Struktur, Zeit, Daten und Kommunikation CPU, FPGA, DRAM,

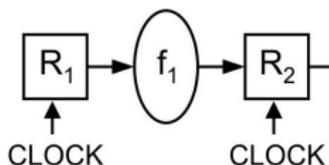
4 Busse, 32b Integer

▶ **Register-Transfer-Ebene**

Synchron, getaktet

always @(posedge CLOCK)

```
R2 <= f1(R1);
```

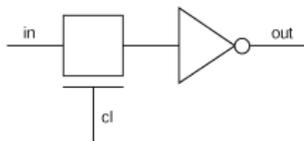


Entwurfsebenen 2

Unterschiedlicher Abstraktionsgrad

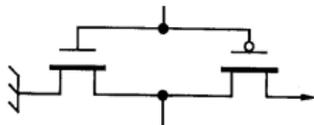
► Logik- oder Gatterebene

Netze aus Gattern, Flip-Flops, etc.



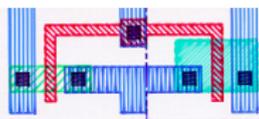
► Transistorebene

Elektrischer Schaltplan



► Layoutebene

Maßstabgetreue geometrische Anordnung des Chips mit verschiedenen Schichten (3D)

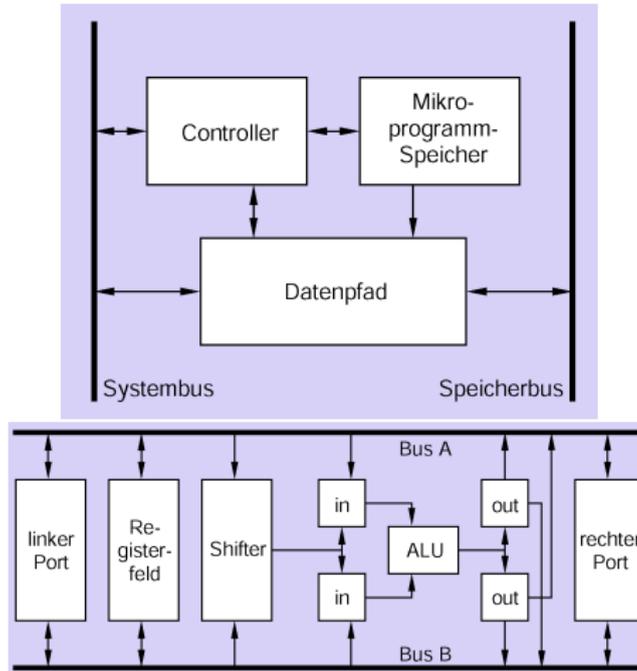


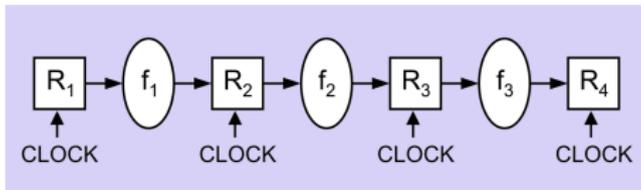
- ▶ Verilog HDL unterstützt auch die Beschreibung des Verhaltens.

```
module MULT_SCALED2(  
  input  wire [15:0] a, b,  
  output wire [32:0] prod  
);  
  
  assign prod = a * b * 2;  
  
endmodule
```

- ▶ Keine Angaben über
 - ▶ Art des Multiplizierers (seriell, parallel, seriell/parallel)
 - ▶ Zeitverhalten

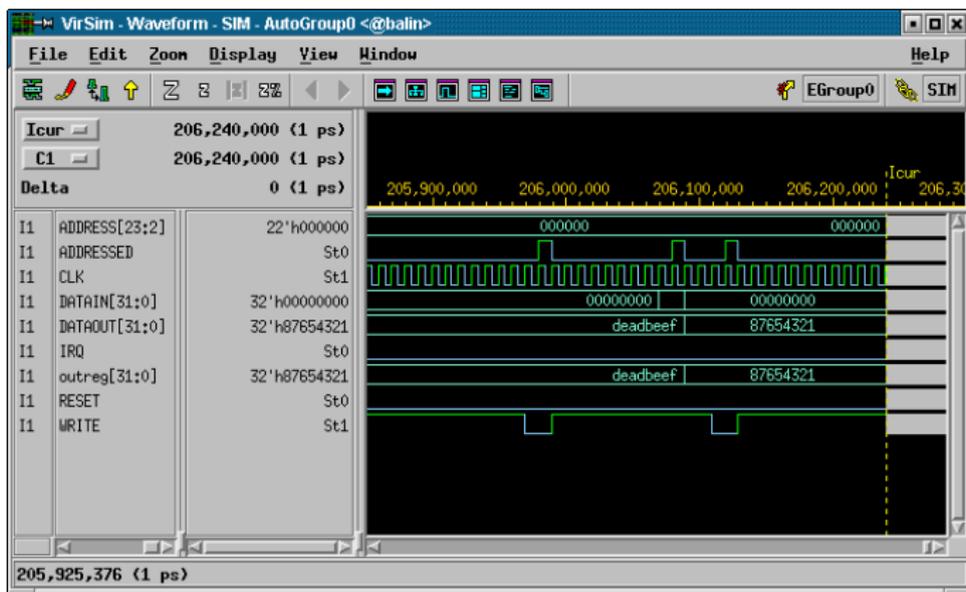
Systemebene Komponenten eines Mikrokontrollers



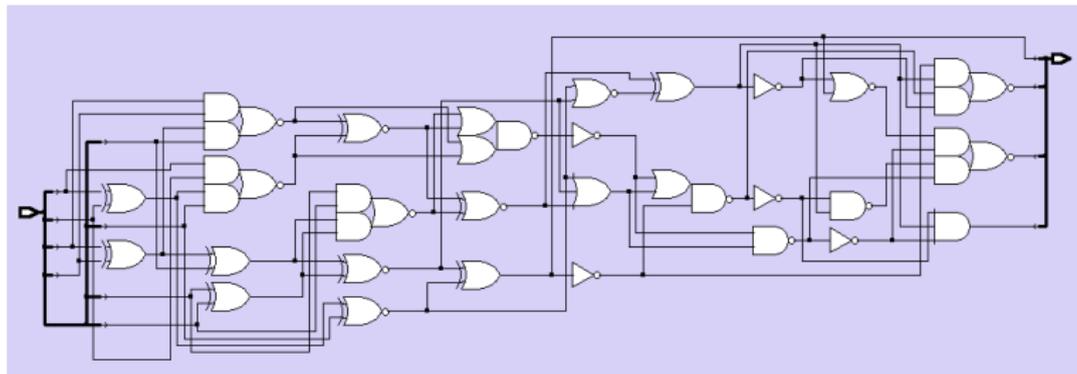


```
always @(posedge CLOCK) // mit jeder steigenden  
begin // Taktflanke  
    R2 <= f1 (R1); // Register-Transfer von  
    R3 <= f2 (R2); // Ri durch fi nach Ri+1  
    R4 <= f3 (R3);  
end
```

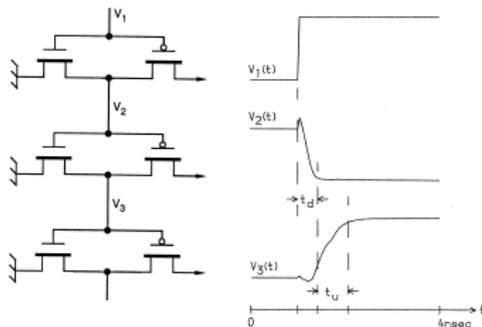
- ▶ Sehr wichtige Entwurfsebene
- ▶ Fließbandverarbeitung (Pipelines)/Automatennetze
- ▶ Effiziente Umsetzung in Hardware automatisch möglich



- ▶ Digitalsimulation noch ohne reale Verzögerungszeiten
- ▶ Alternativ auch Textausgabe möglich



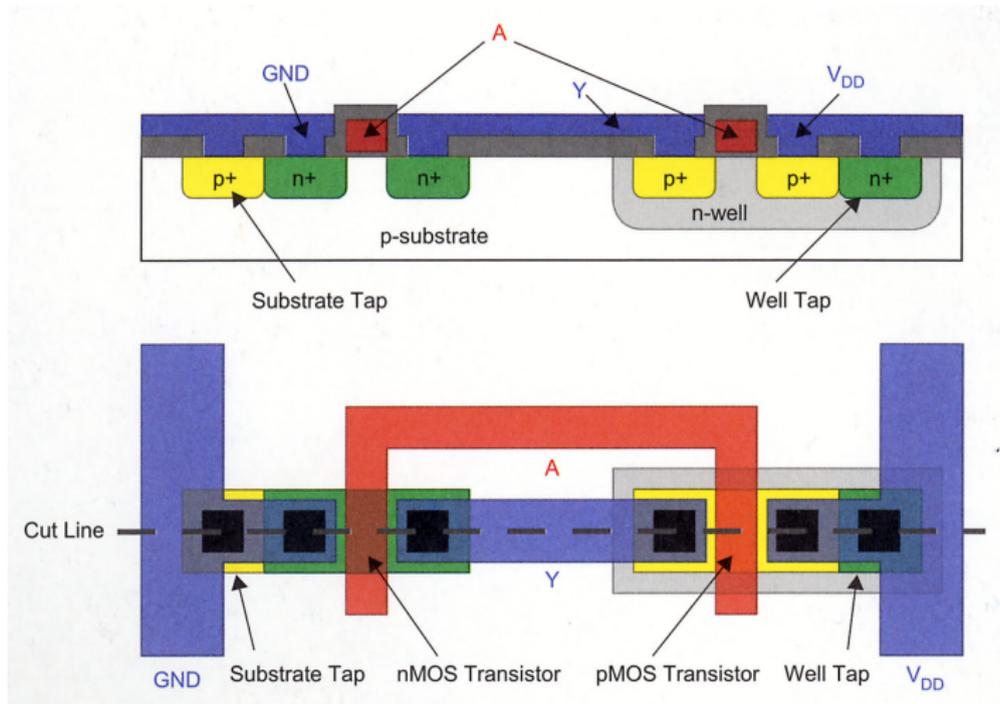
- ▶ Ergebnis der Logiksynthese
- ▶ Generische AND, NOR, Flip-Flops, etc.
- ▶ Zeitverhalten abschätzbar (aber noch sehr ungenau)
- ▶ Noch keine endgültige Hardware-Beschreibung
 - ▶ Hängt von konkreter Zieltechnologie ab
 - ▶ ASIC, FPGA, Gate-Array, ...



- ▶ Schaltpläne aus Transistoren, Widerständen, etc.
- ▶ Beim Digitalschaltungsentwurf verborgen
- ▶ Ausnahmen
 - ▶ Analoge Teilschaltungen
 - ▶ Full-Custom-Entwurf
- ▶ Analogsimulation mit Schaltzeiten

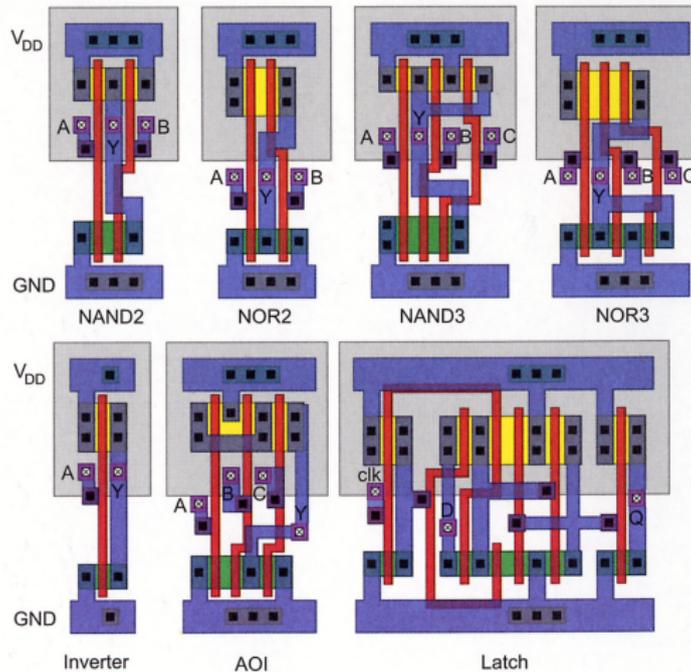
Layout-Ebene

Geometrien der Transistoren und Leitungen



Layout-Ebene

Einige Basisgatter (=Zellen)



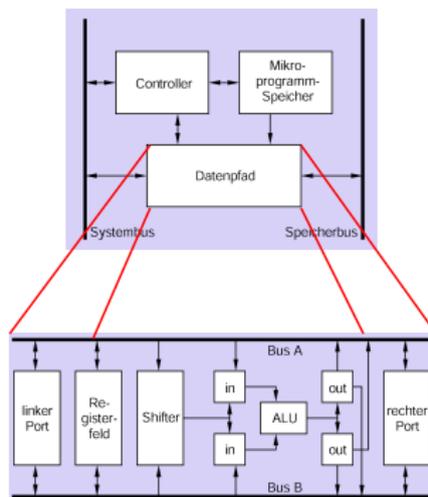


- ▶ Maßstabsgetreue Darstellung des endgültigen Chips
- ▶ Ergebnis von Platzieren und Verdrahten
 - ▶ Transistoren und Leitungen als Polygone
 - ▶ Abmessungen haben Einfluß auf elektrische Eigenschaften
 - ▶ Nun genaues Zeitverhalten bekannt
 - ▶ Weitergabe an Halbleiterhersteller (Tape-Out)

- ▶ Weglassen für die *aktuelle* Beschreibung *unwichtiger* Details
- ▶ Arbeiten auf unterschiedlichen Ebenen
 - ▶ Von ungenau bis sehr genau
 - ▶ Verhaltensebene, . . . , Layout-Ebene
- ▶ Beispiel: Entwurf eines MP3-Encoder-Chips
 - ▶ Manuell von funktionaler bis RTL-Ebene
 - ▶ Andere Schritte i. d. R. automatisch
- ▶ Beispiel: Entwurf von Empfangsteil für UMTS-Handy
 - ▶ Manuell von funktionaler bis Layout-Ebene
 - ▶ Sehr komplizierte Chips
 - ▶ Entwurf erfordert hochspezialisierte Kenntnisse.
 - ▶ Hochfrequenztechnik-Schaltungen sind schwierig zu modellieren.

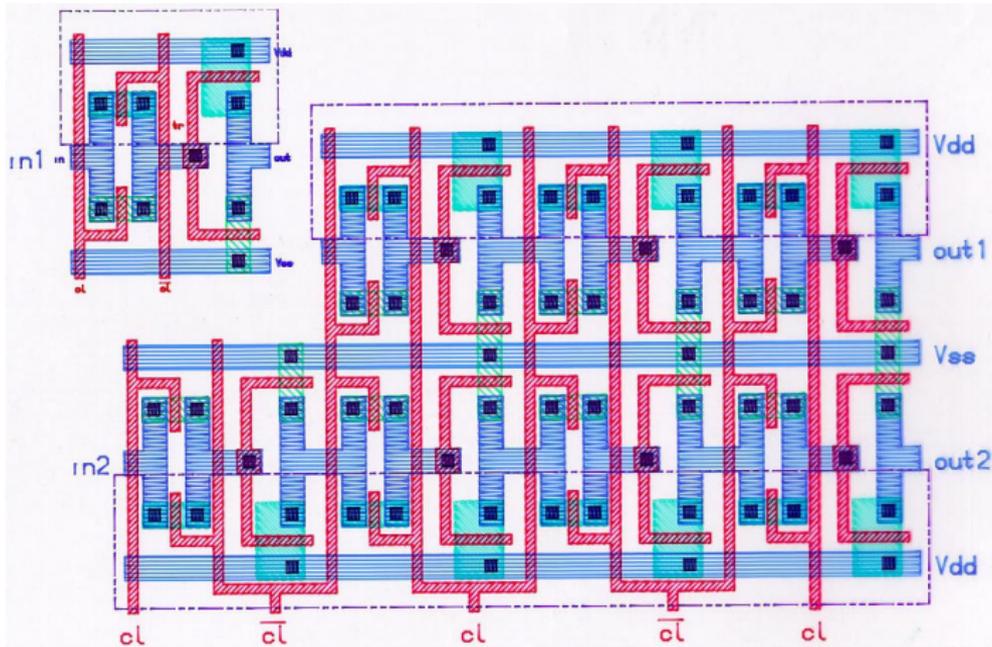
Hierarchische Zerlegung - Aufteilen eines Problems in kleinere Unterprobleme

- ▶ Alte Idee: divide et impera (Philip II, -381 ... -335)
- ▶ Auch rekursiv anwendbar
- ▶ Damit entsteht eine Hierarchie von Zerlegungen



Reguläre Zerlegung

Gezielte Vervielfältigung von Komponenten



- ▶ Top-down-Entwurf (Detaillierungsprozeß)
- ▶ Bottom-up-Entwurf (Kompositionsprozeß)
- ▶ Meet-in-the-Middle-Entwurf
- ▶ Diese Methoden existieren auch im Softwareentwurf.



- ▶ Der Schaltungsentwickler beginnt mit seinem Wissen über die zu realisierende Gesamtfunktion, die danach in kleinere Teilfunktionen (Teileinheiten, Komponenten) und ein Verbindungsnetz (Verbindung der Schnittstellen durch Signale) zerlegt wird.
- ▶ Die Zerlegung kann hinsichtlich verschiedener Kriterien, wie Kosten, Geschwindigkeit oder Chipfläche erfolgen.
- ▶ Der Prozeß endet, wenn Basis-Funktionen verwendet werden können.
- ▶ Der Top-down-Entwurf setzt voraus, dass sich Teilsysteme zunächst abstrakt, nämlich als „black boxes“ beschreiben lassen. Die nach außen hin sichtbare Funktionalität der Teileinheiten muß abstrakt definierbar sein, ohne dass auf die Details ihrer internen Realisierung näher eingegangen werden muß. Damit ist es möglich, das Zusammenwirken von Systemkomponenten zu evaluieren bevor diese vollständig auf Logik-Ebene entworfen wurden.



- ▶ Bottom-up: Hier wird von den verfügbaren Primitiven (z. B. TTL-Gatter) ausgegangen, welche in einer Bibliothek abgelegt sind. Aus den Bibliothekselementen werden komplexere Komponenten gebildet, die ihrerseits auf der nächst höheren Ebene als (elementare) Bausteine eingesetzt werden können.
- ▶ Meet-in-the-Middle: Der Meet-in-the-middle-Entwurf vereinigt die Top-down- und die Bottom-up-Vorgehensweise, d. h. man beachtet beim Vorgehen von der einen Seite des Entwurfsprozesses (z. B. Top) die Auswirkungen an der anderen

- ▶ Einführung
- ▶ Entwicklungen in der Mikroelektronik
- ▶ Entwurfsebenen
- ▶ Fundamentale Entwurfstechniken

Nächste Vorlesung behandelt

- ▶ Konzepte von Hardwarebeschreibungssprachen
- ▶ Verilog HDL