



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Einführung in Computer Microsystems

1. Einleitung

Andreas Koch

FG Eingebettete Systeme und ihre Anwendungen
Informatik, TU Darmstadt

Sommersemester 2007



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Einleitung



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Mikroelektronik
- vulgo Chips
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte Hardware im Blick



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Mikroelektronik
- vulgo **Chips**
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte **Hardware** im Blick



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Mikroelektronik
- vulgo **Chips**
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte **Hardware** im Blick



- Mikroelektronik
- vulgo **Chips**
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte **Hardware** im Blick



- Mikroelektronik
- vulgo **Chips**
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte **Hardware** im Blick



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Mikroelektronik
- vulgo **Chips**
- Hier aber informatiker-kompatible Sicht
 - Weniger Differentialgleichungen auf Transistorebene
 - ... eher abstraktere Sicht der Dinge
- Aber immer echte **Hardware** im Blick



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - Desktop-Rechner, Server, Embedded
 - Versteckt: eingebettete Systeme
 - Mobiltelefone, Kameras, Videorekorder
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze

➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze

➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
 - Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
 - Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
 - Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
 - Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze

➔ Sollte man sich auch als Informatiker genauer angucken

- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
- Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
- Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze

➔ Sollte man sich auch als Informatiker genauer angucken

- Ursprung dieser Bedeutung?



- Wert in 2004: 213 Milliarden USD weltweit
 - Nahezu **überall** verbaut
 - Offensichtlich: Rechner
 - PC, Server, Supercomputer, ...
 - Versteckt: eingebettete Systeme
 - Autos, Fernseher, Herzschrittmacher, ...
 - Alleine in D: 9 Milliarden EUR Chip-Umsatz
 - Bildet aber Basis für 50x **größeren** Markt
 - 3 Millionen Arbeitsplätze
- ➔ Sollte man sich auch als Informatiker genauer angucken
- Ursprung dieser Bedeutung?



CMS

A. Koch

Intro

Entwicklung

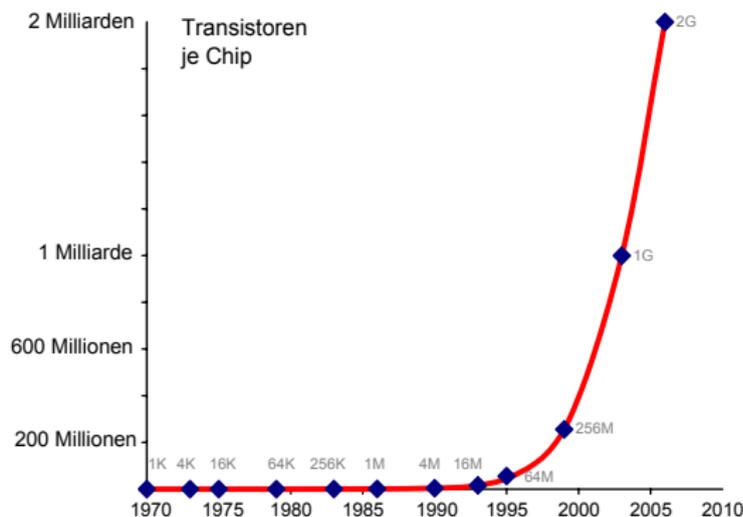
Orga

Entwurf

Methoden

Exponentielles Wachstum

Moore's Gesetz



Alle 18 Monate verdoppelt sich Anzahl auf einem Chip
praktisch herstellbarer Transistoren

Praktische Auswirkungen?

CMS

A. Koch

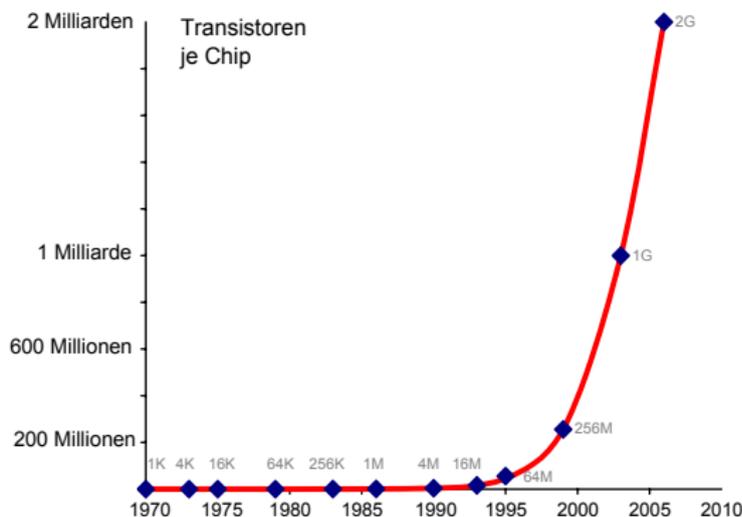
Intro

Entwicklung

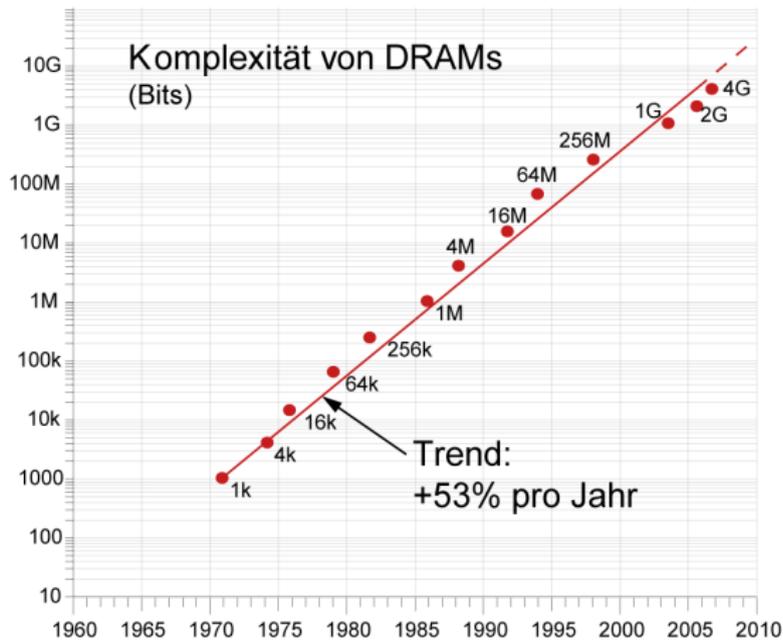
Orga

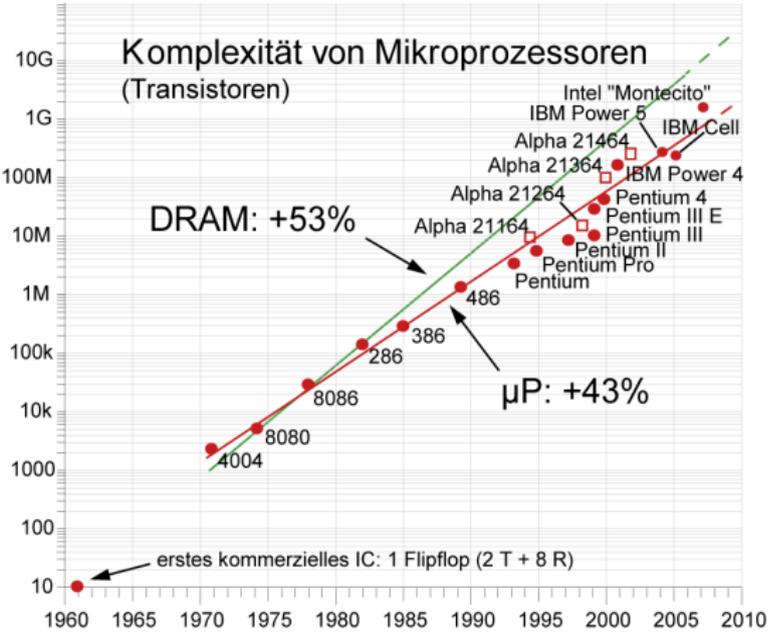
Entwurf

Methoden



Alle 18 Monate verdoppelt sich Anzahl auf einem Chip
praktisch herstellbarer Transistoren
Praktische Auswirkungen?





Verbesserung der Fertigungsprozesse



CMS

A. Koch

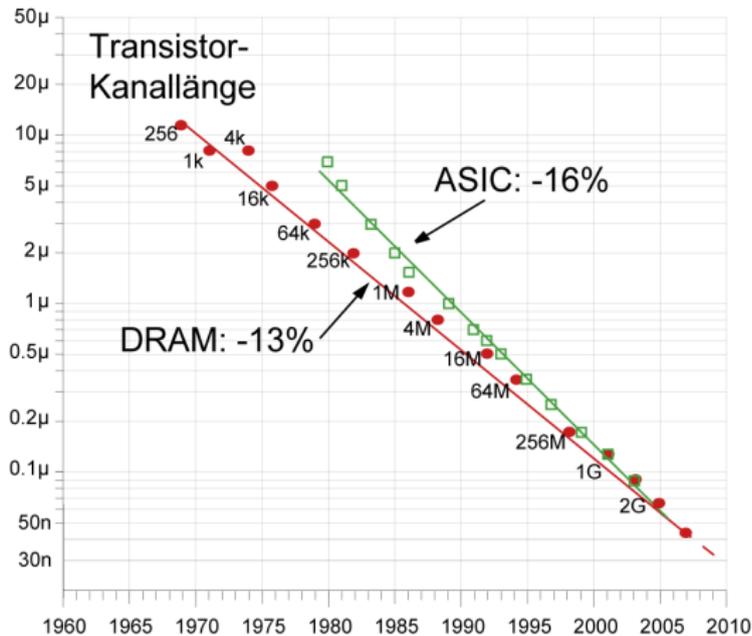
Intro

Entwicklung

Orga

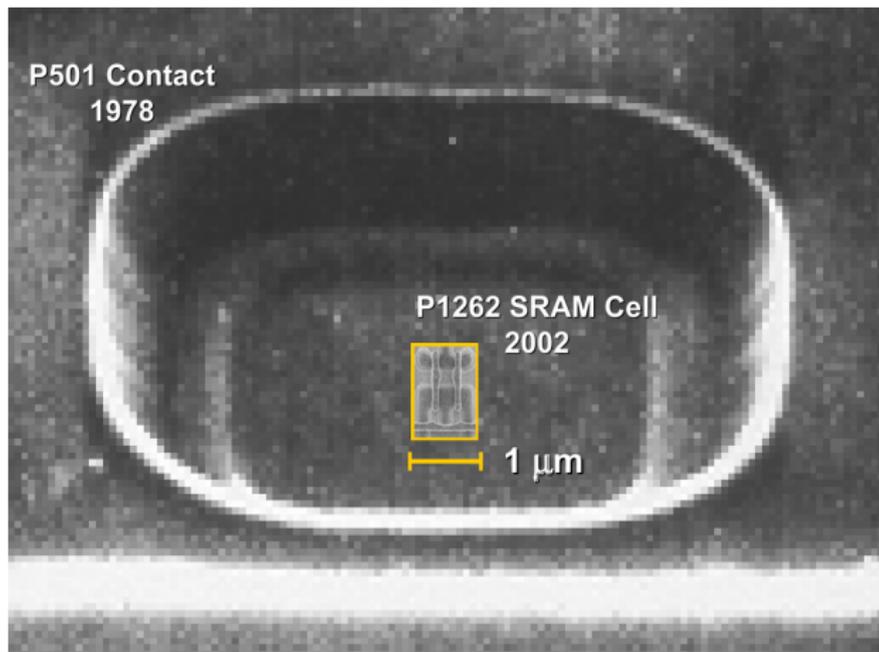
Entwurf

Methoden



Vergleich von Abmessungen

P1262: 90nm



CMS

A. Koch

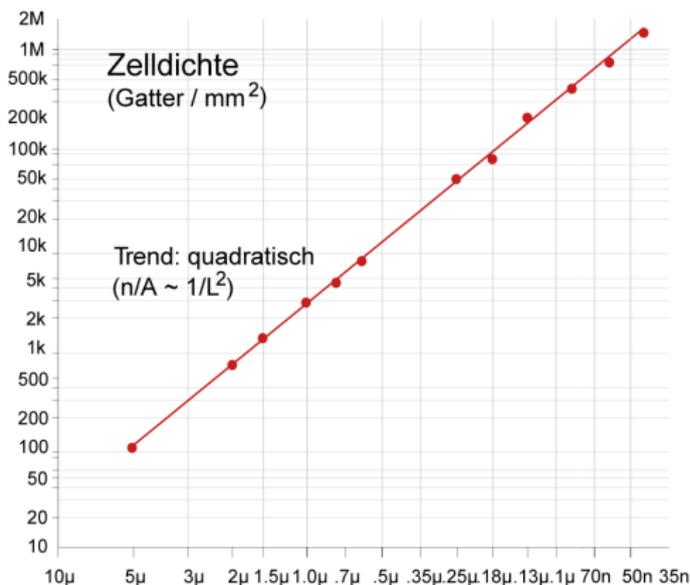
Intro

Entwicklung

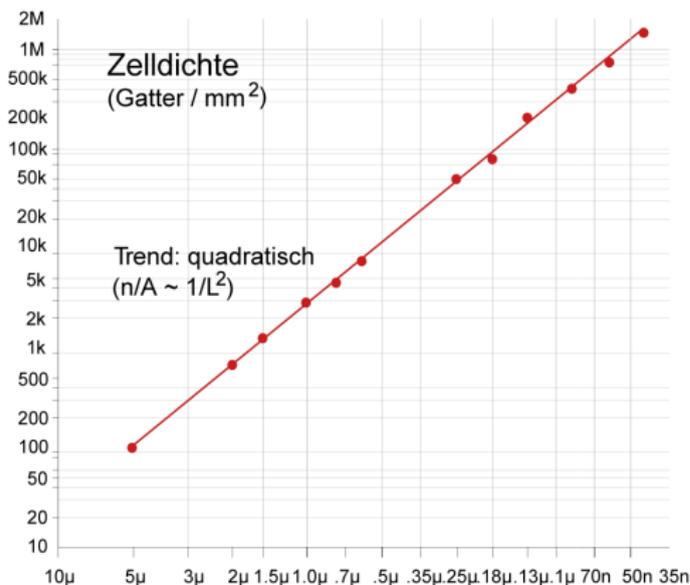
Orga

Entwurf

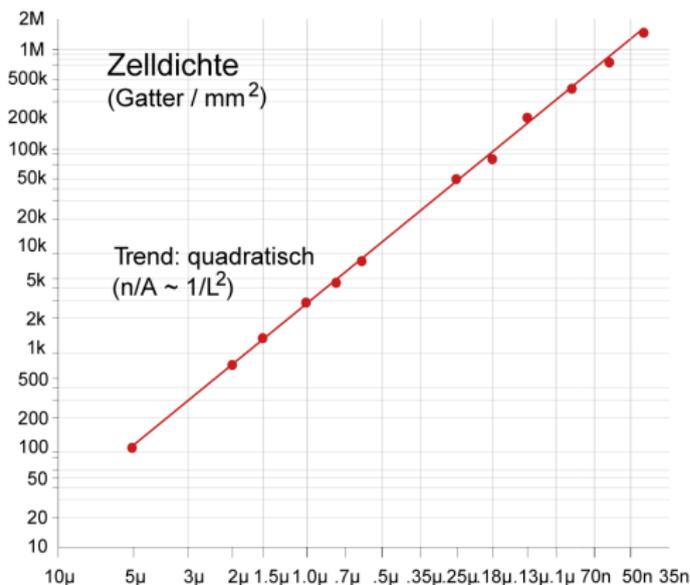
Methoden



- Transistoren schrumpfen um 13% jährlich
- Übrige Strukturen ungefähr auch
- Pro Flächeneinheit $1/0,87^2 = 33\%$ mehr Elemente



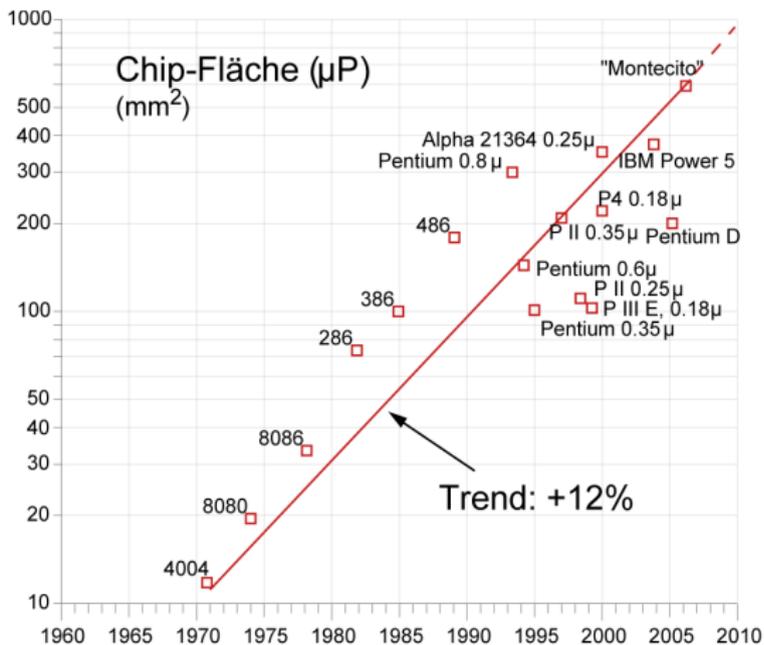
- Transistoren schrumpfen um 13% jährlich
- Übrige Strukturen ungefähr auch
- Pro Flächeneinheit $1/0,87^2 = 33\%$ mehr Elemente



- Transistoren schrumpfen um 13% jährlich
- Übrige Strukturen ungefähr auch
- Pro Flächeneinheit $1/0,87^2 = 33\%$ mehr Elemente

Chip-Größen

Nicht nur kleinere Strukturen, auch größere Chip-Flächen



CMS

A. Koch

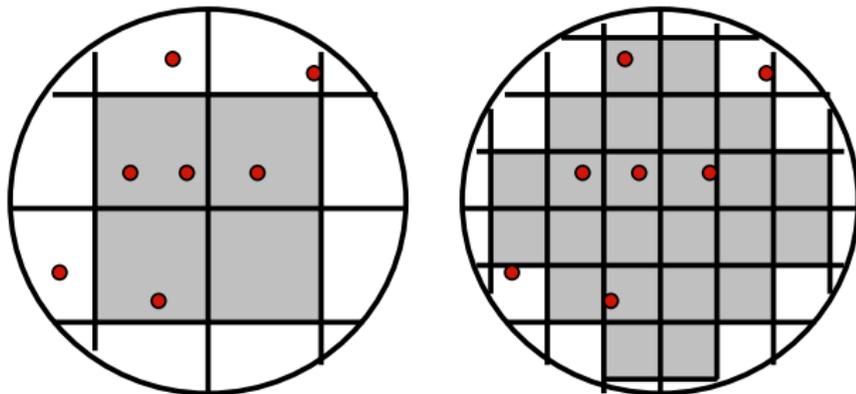
Intro

Entwicklung

Orga

Entwurf

Methoden



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

Quelle: J. Rabaey, Digital Integrated Circuits, 2003

CMS

A. Koch

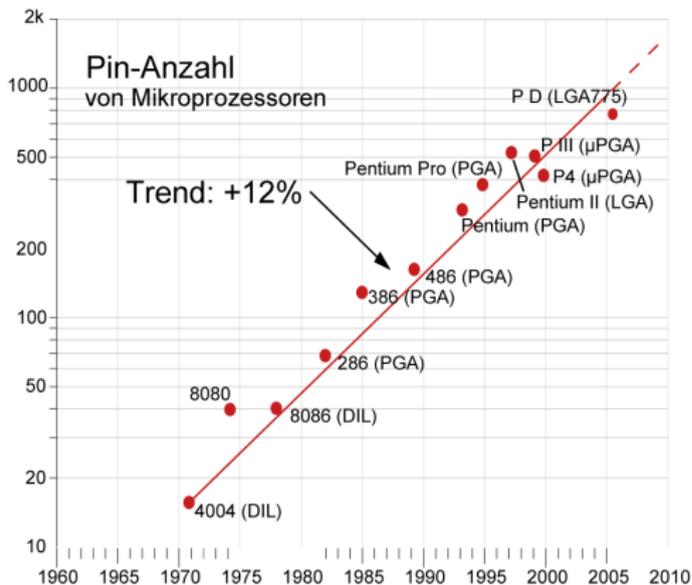
Intro

Entwicklung

Orga

Entwurf

Methoden



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

Taktfrequenz



CMS

A. Koch

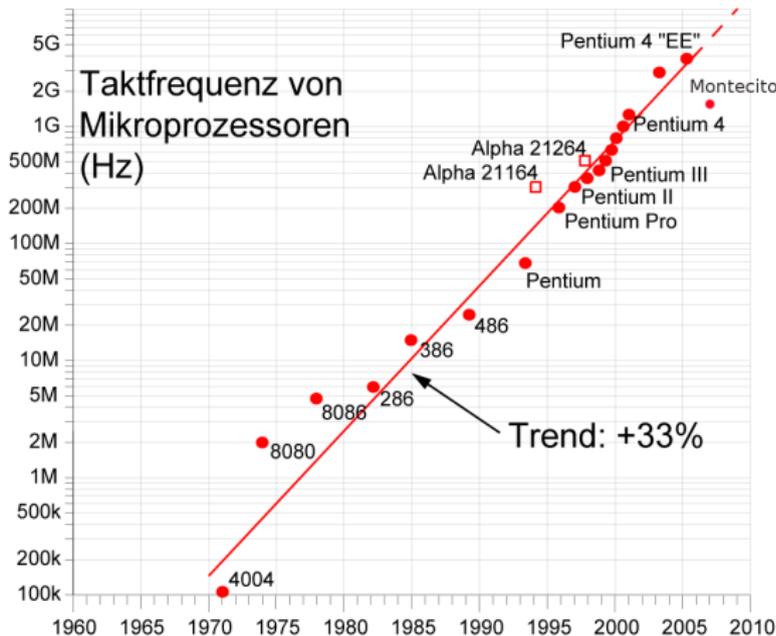
Intro

Entwicklung

Orga

Entwurf

Methoden



Taktfrequenz: Es kommt immer darauf an, wie weit man mit jedem Schritt kommt!



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Intel Pentium 4 EE
3,8 GHz Takt
11,5 SPECint 2006

Intel Montecito 9050
1,6 GHz Takt
14,5 SPECint 2006

Taktfrequenz: Es kommt immer darauf an, wie weit man mit jedem Schritt kommt!



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Intel Pentium 4 EE
3,8 GHz Takt
11,5 SPECint 2006

Intel Montecito 9050
1,6 GHz Takt
14,5 SPECint 2006

Taktfrequenz: Es kommt immer darauf an, wie weit man mit jedem Schritt kommt!



CMS

A. Koch

Intro

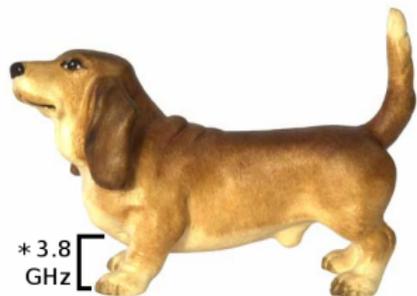
Entwicklung

Orga

Entwurf

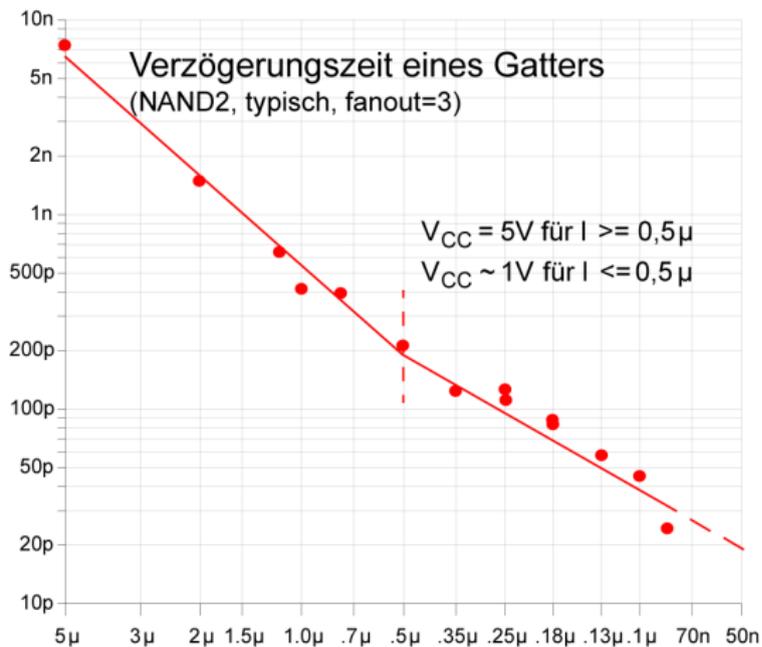
Methoden

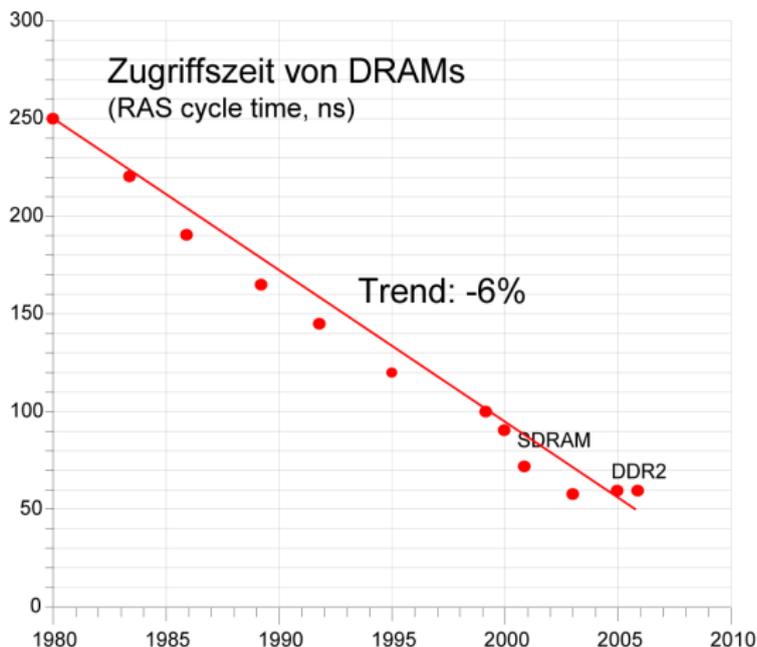
Intel Pentium 4 EE
3,8 GHz Takt
11,5 SPECint 2006



Intel Montecito 9050
1,6 GHz Takt
14,5 SPECint 2006

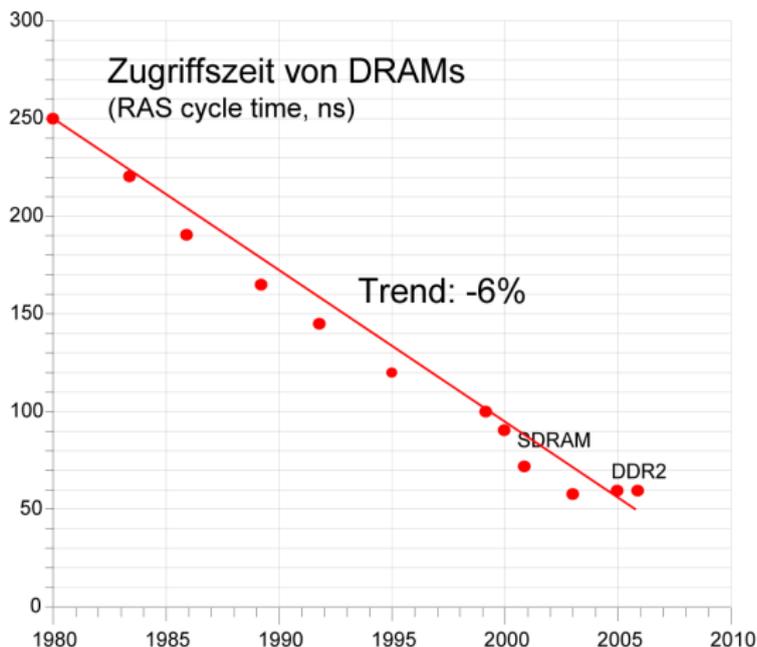






Problem: Gatterlaufzeiten im ps-Bereich

Hauptspeicher ist langsam!



Problem: Gatterlaufzeiten im ps-Bereich
Hauptspeicher ist **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!)**



- +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!)



- +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!)



- +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!)



- +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!)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- +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!)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- +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!)



CMS

A. Koch

Intro

Entwicklung

Orga

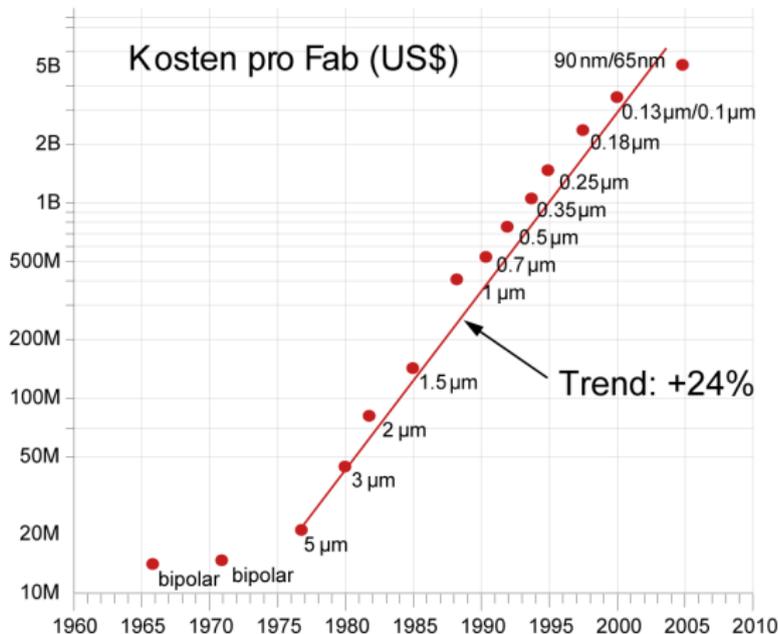
Entwurf

Methoden

- +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!)

Technik ist aber nicht alles

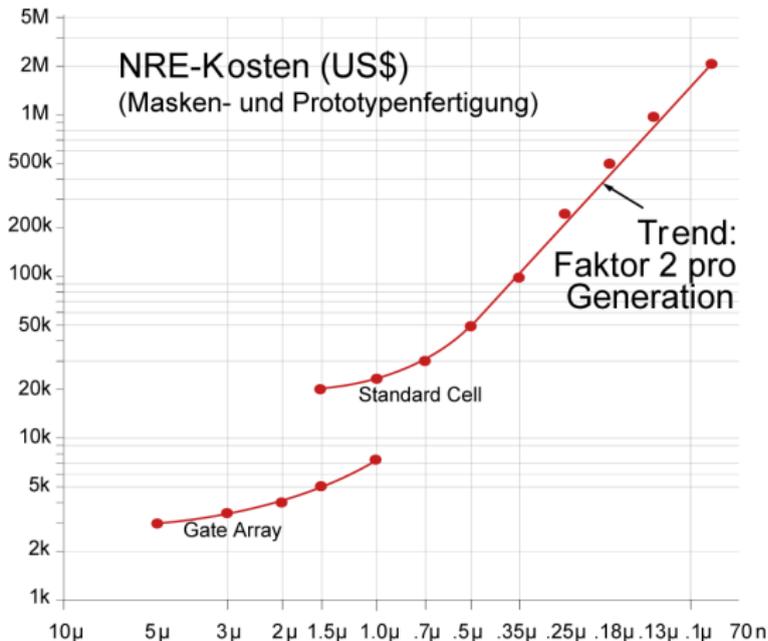
Was kostet der Spaß?



In 10 Jahren fast verzehnfachte Kosten!

Kosten eines Chip-Musters

Non Recurring Engineering-Kosten (NRE)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- **Fertigung auf älterer Technologie**
 - 180nm ... 110nm *sehr* weit verbreitet
- **Multi-Projekt-Chips**
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- **Programmierbare/konfigurierbare Schaltungen**
 - Keine photochemische Chip-Fertigung mehr erforderlich



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



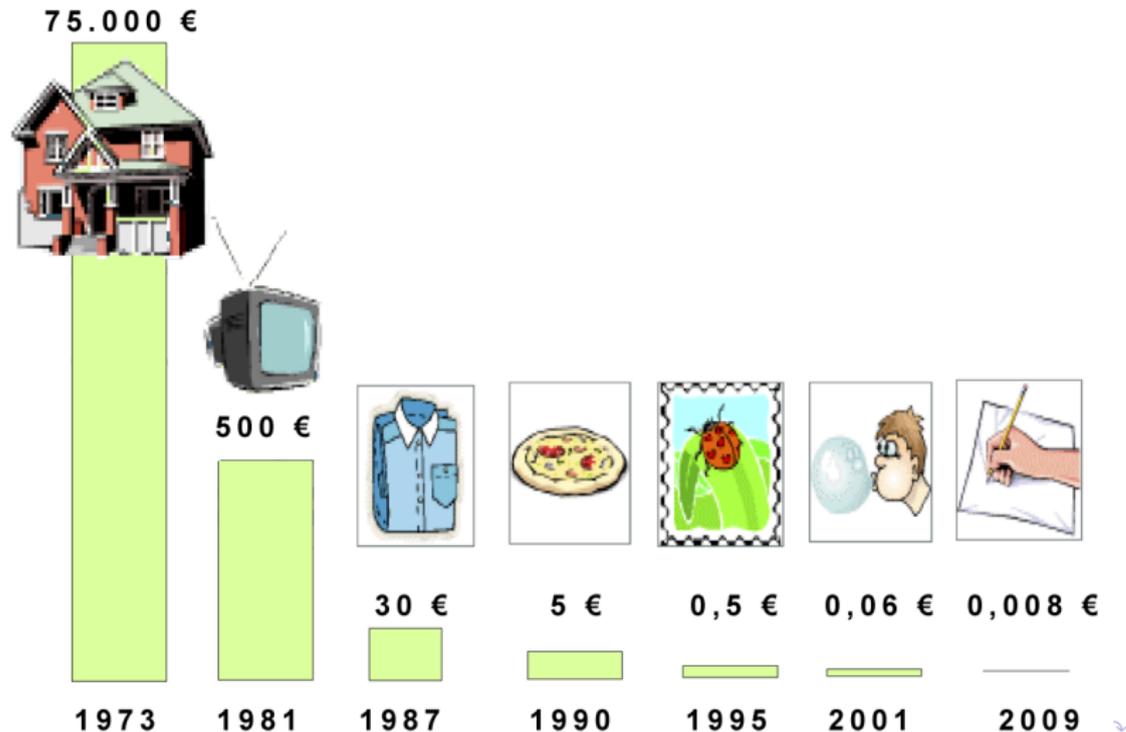
- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich



- Fertigung auf älterer Technologie
 - 180nm ... 110nm *sehr* weit verbreitet
- Multi-Projekt-Chips
 - **Sonderangebote** für EU Forschung und Lehre
 - UMC L130E: EUR 32.000/5mm x 5mm
 - UMC L90N: EUR 50.000/4mm x 4mm
 - Für ca. 20 Chips, Gehäuse kosten extra
- Programmierbare/konfigurierbare Schaltungen
 - Keine photochemische Chip-Fertigung mehr erforderlich

Kostensenkung durch Massenfertigung

Kosten für 1 Mb DRAM



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Organisatorisches



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Vorlesung Andreas Koch

Übung Holger Lange

Tutor Thomas Feller

Mi 14:00-15:00, E103

Mi 14:00-15:00, E106

TBA



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Nahtloser Anschluss an

- Technische Grundlagen der Informatik 1
 - Digitale Logik, Basisgatter
- Technische Grundlagen der Informatik 2
 - Hardware für Arithmetik, einfache Prozessoren



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Nahtloser Anschluss an

- Technische Grundlagen der Informatik 1
 - Digitale Logik, Basisgatter
- Technische Grundlagen der Informatik 2
 - Hardware für Arithmetik, einfache Prozessoren



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Nahtloser Anschluss an

- Technische Grundlagen der Informatik 1
 - Digitale Logik, Basisgatter
- Technische Grundlagen der Informatik 2
 - Hardware für Arithmetik, einfache Prozessoren



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Nahtloser Anschluss an

- Technische Grundlagen der Informatik 1
 - Digitale Logik, Basisgatter
- Technische Grundlagen der Informatik 2
 - Hardware für Arithmetik, einfache Prozessoren



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - VLSI Entwurf digitaler Schaltungen und Systeme (Prof. Krumm)
 - VLSI Hochgeschwindigkeiten (Prof. Krumm)
 - VLSI Entwurf digitaler Systeme und Bus-Systeme (Prof. Krumm)



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - VLSI (Prof. Dr. Grottel)
 - VLSI (Prof. Dr. Grottel)
 - VLSI (Prof. Dr. Grottel)



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



- Durchgehend **konkreter** Hardware-Bezug
- Praktische Erfahrung in Entwurf digitaler Schaltungen
 - Hier in Hardware-Beschreibungssprache **Verilog**
 - Aber **fundiierter** als das Reinschnuppern in TGD12
 - Nicht nur Simulation, sondern auch **Synthese**
- Weniger abstrakte Modelle als in den Vorjahren
 - Zeit reicht schlicht nicht
 - Diese Themen dann später in Spezialveranstaltungen
 - FG Integrierte Schaltungen und Systeme (Prof. Huss)
 - FG Rechnerarchitekturen (Prof. Hoffmann)
 - FG Eingebettete Systeme und ihre Anwendungen



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Veranstaltungen

- **Vorlesung** **Mi 9:50-11:20, C205**
- Übung **Do 14:25-15:10, C205**
 - Beginn ab dem 26.04.07
- Tutorensprechstunde zu Aufgaben **TBA**



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Veranstaltungen

- Vorlesung Mi 9:50-11:20, C205
- Übung Do 14:25-15:10, C205
 - Beginn ab dem 26.04.07
- Tutorensprechstunde zu Aufgaben TBA



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Veranstaltungen

- Vorlesung Mi 9:50-11:20, C205
- Übung Do 14:25-15:10, C205
 - Beginn ab dem 26.04.07
- Tutorensprechstunde zu Aufgaben TBA



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Veranstaltungen

- Vorlesung Mi 9:50-11:20, C205
- Übung Do 14:25-15:10, C205
 - Beginn ab dem 26.04.07
- Tutorensprechstunde zu Aufgaben TBA



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- **Wöchentlich**
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen

CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden



- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



- Wöchentlich
- Freiwillig
- Dieses Jahr auch noch keine Anrechnung auf Klausurpunkte
 - ... kommt hoffentlich nächstes Jahr!
- Bearbeitung aber trotzdem **dringend** empfohlen
- Werden eine Woche nach Ausgabe in Tafelübung **vorgerechnet**
- Bei Schwierigkeiten
 - Diskussion im Forum der Fachschaft
 - Persönliche Betreuung durch Tutor bei **konkreten** Problemen



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vorlesungsbegleitend
- Zwei Teilklausuren
- Datum steht noch **nicht** fest
- **1. Klausur** zur Vorlesungsmitte
 - 45min, erreichbar 45 Punkte
- **2. Klausur** zum Vorlesungsende
 - 90min, erreichbar 90 Punkte
- Beide Teilpunktzahlen werden addiert
- Sicher Bestanden bei Erreichen von 67,5 Punkten



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Zur **Raumplanung** unbedingt erforderlich

- Bis zum 26.4., 12:00 Uhr via WebReg
- Für Sie unverbindlich
- Ersetzt **nicht** möglicherweise erforderliche *echte* Prüfungsanmeldung
 - Z.B. im zentralen Prüfungssekretariat



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Zur **Raumplanung** unbedingt erforderlich

- Bis zum 26.4., 12:00 Uhr via WebReg
- Für Sie unverbindlich
- Ersetzt **nicht** möglicherweise erforderliche *echte* Prüfungsanmeldung
 - Z.B. im zentralen Prüfungssekretariat



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Zur **Raumplanung** unbedingt erforderlich

- Bis zum 26.4., 12:00 Uhr via WebReg
- Für Sie unverbindlich
- Ersetzt **nicht** möglicherweise erforderliche *echte* Prüfungsanmeldung
 - Z.B. im zentralen Prüfungssekretariat



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Zur **Raumplanung** unbedingt erforderlich

- Bis zum 26.4., 12:00 Uhr via WebReg
- Für Sie unverbindlich
- Ersetzt **nicht** möglicherweise erforderliche *echte* Prüfungsanmeldung
 - Z.B. im zentralen Prüfungssekretariat



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - Logik-Synthese
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Test Editor und -Werkzeug
 - Synthesizer
 - Synthesizer
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - http://www.xilinx.com/lit/gtw012/ise_webpack_download.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Test Editor und -Simulator
 - Synthesizer
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Test Editor und -Simulator
 - Synthesizer
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



- Diesmal mehr als nur Verilog-Simulator
- Nun auch Hardware-Erzeugung aus Verilog
 - **Logik-Synthese**
- Zunächst verwendetes Werkzeug
 - Xilinx ISE, enthält u.a.
 - Verilog-Text Editor und -Wizards
 - Simulator
 - Synthese
 - Auf RBG-Pool-Rechnern installiert
 - Freie Version ISE WebPack herunterladbar
 - Für Windows und x86-Linux
 - http://www.xilinx.com/ise/logic_design_prod/webpack.htm
- Bei Bedarf können auch noch andere Werkzeuge zum Einsatz kommen



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Hardware-Entwurf



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation ("re-spin")

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation ("re-spin")

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation ("re-spin")

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation (“re-spin”)

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation (“re-spin”)

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation (“re-spin”)

Wie in den Griff bekommen?



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Vergleichbar einem Puzzle mit **einer Milliarde** Teile
- Zusammensetzen unter hohem Zeitdruck
 - Time-to-Market (TTM)
- Ein **einzig**er Fehler kann zwei Millionen USD kosten
 - Erneute Chip-Fabrikation (“re-spin”)

Wie in den Griff bekommen?



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!

CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/VHDL
- Entkoppeln von Entwurf und technischer Realisierung
- Umsetzung idealerweise **automatisch** (Synthese)
 - Klappt aber noch nicht immer!



- **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/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.

$$y = f(x)$$

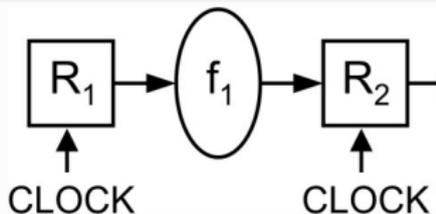
Systemebene

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);`



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Entwurfsebenen 1

Unterschiedlicher Abstraktionsgrad



Verhaltensebene

Was soll passieren?
Realisierung bleibt offen.

$$y = f(x)$$

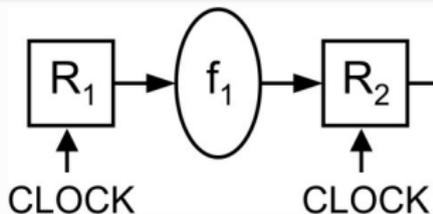
Systemebene

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);`



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Entwurfsebenen 1

Unterschiedlicher Abstraktionsgrad



Verhaltensebene

Was soll passieren?
Realisierung bleibt offen.

$$y = f(x)$$

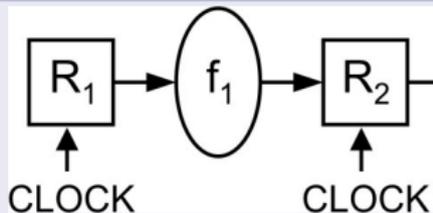
Systemebene

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);`



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

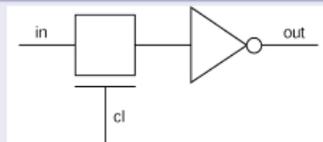
Methoden

Entwurfsebenen 2



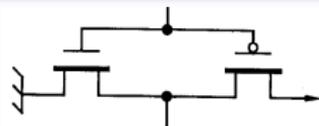
Logik- oder Gatterebene

Netze aus Gattern,
Flip-Flops, etc.



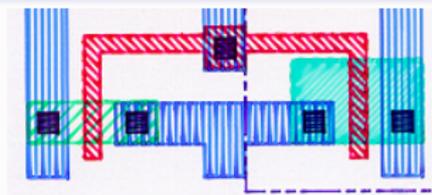
Transistorebene

Elektrischer Schaltplan



Layoutebene

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



CMS

A. Koch

Intro

Entwicklung

Orga

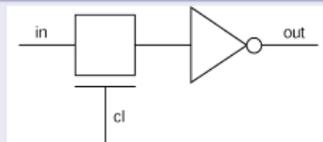
Entwurf

Methoden



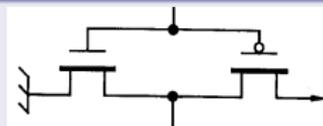
Logik- oder Gatterebene

Netze aus Gattern,
Flip-Flops, etc.



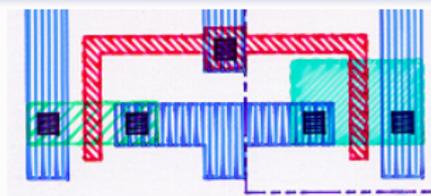
Transistorebene

Elektrischer Schaltplan



Layoutebene

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



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

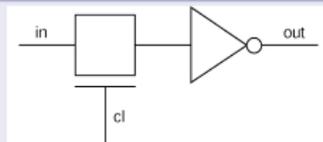
Methoden

Entwurfsebenen 2



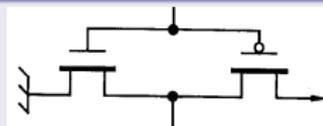
Logik- oder Gatterebene

Netze aus Gattern,
Flip-Flops, etc.



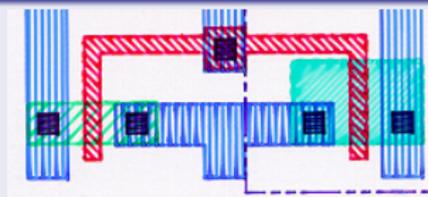
Transistorebene

Elektrischer Schaltplan



Layoutebene

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



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden



```
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



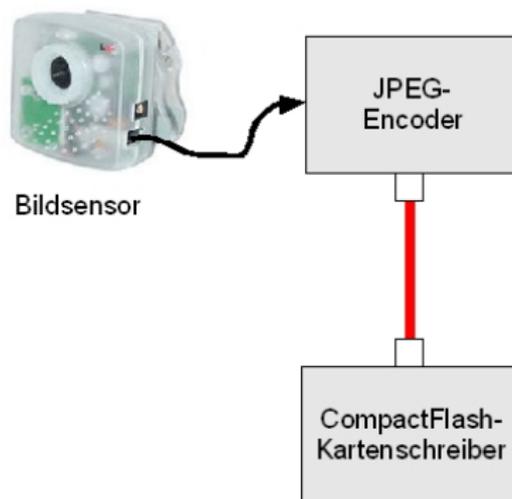
```
module MULT_SCALED2(  
  input wire [15:0] a, b,  
  output wire [32:0] prod  
);  
  
  assign prod = a * b * 2;  
  
endmodule
```

Keine Angaben über

- **Art** des Multiplizierers (seriell, parallel, seriell/parallel)
- Zeitverhalten

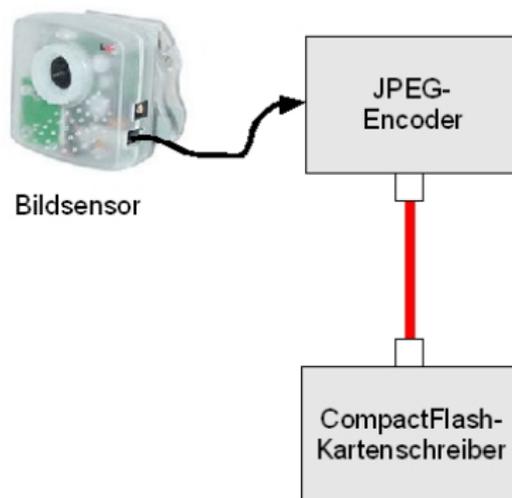


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



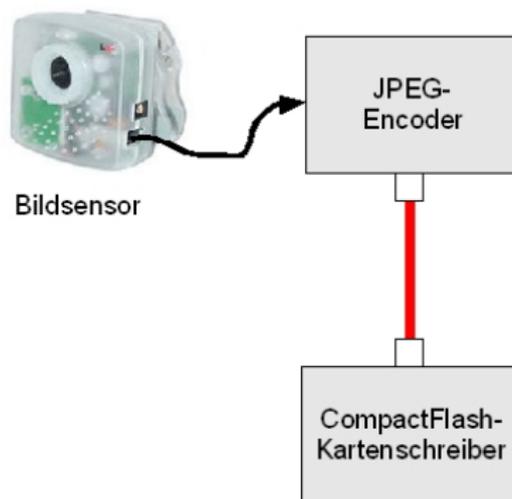


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



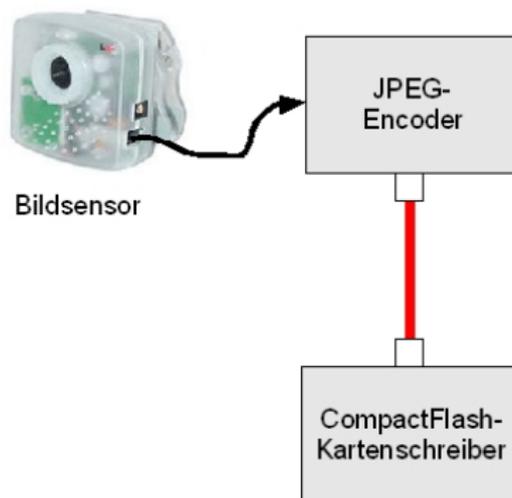


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



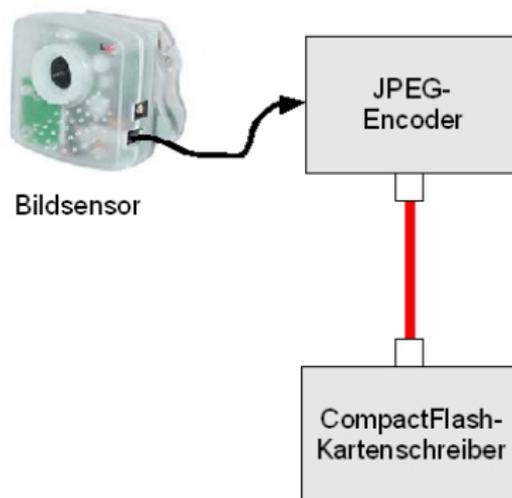


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



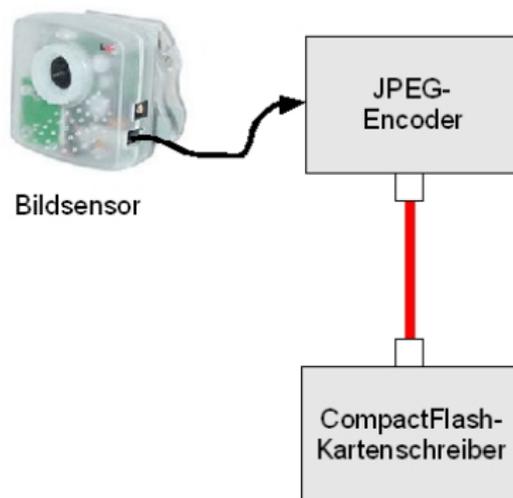


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



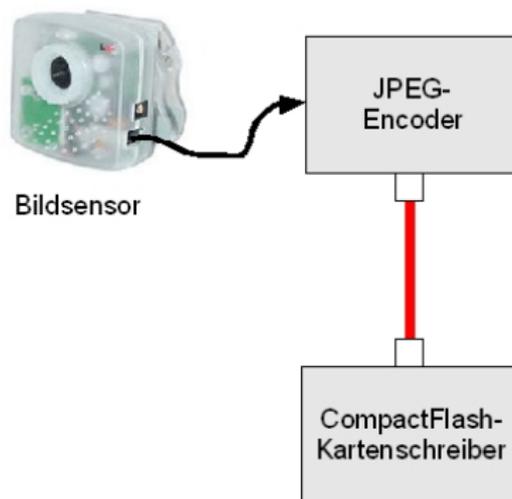


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



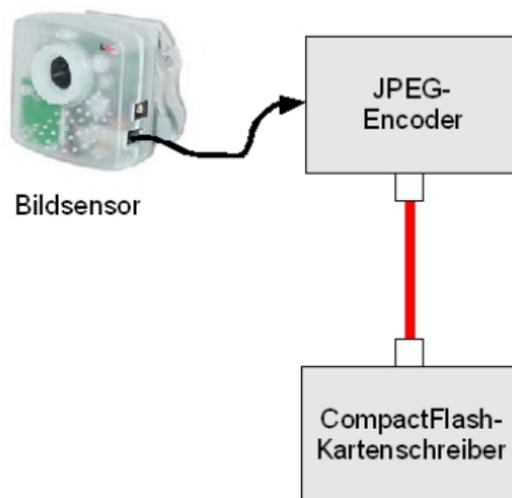


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



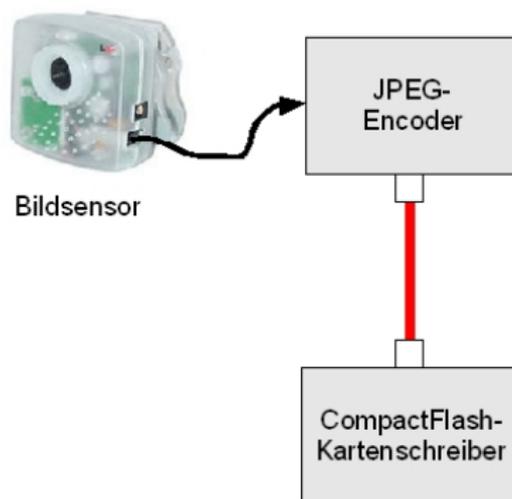


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)



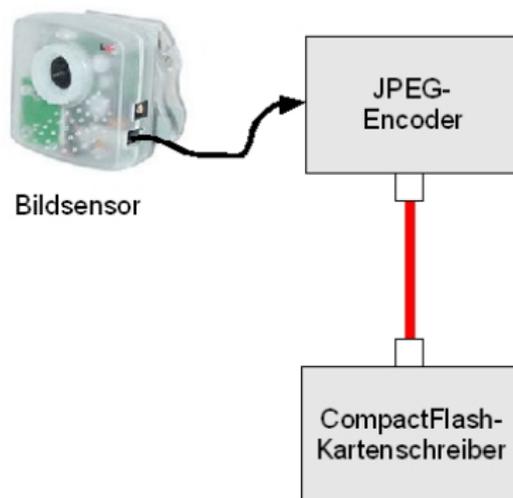


- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)





- Größere Komponenten
 - Details nicht betrachtet oder sind noch unbekannt
- Grobmodellierung
 - Struktur
 - Kommunikation
 - Zeit
 - Daten
 - Algorithmen
- Spezielle Beschreibungssprachen
 - System-C (aufgebohrtes C++)





CMS

A. Koch

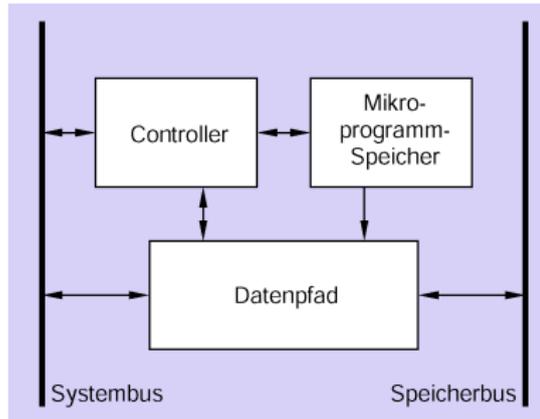
Intro

Entwicklung

Orga

Entwurf

Methoden





CMS

A. Koch

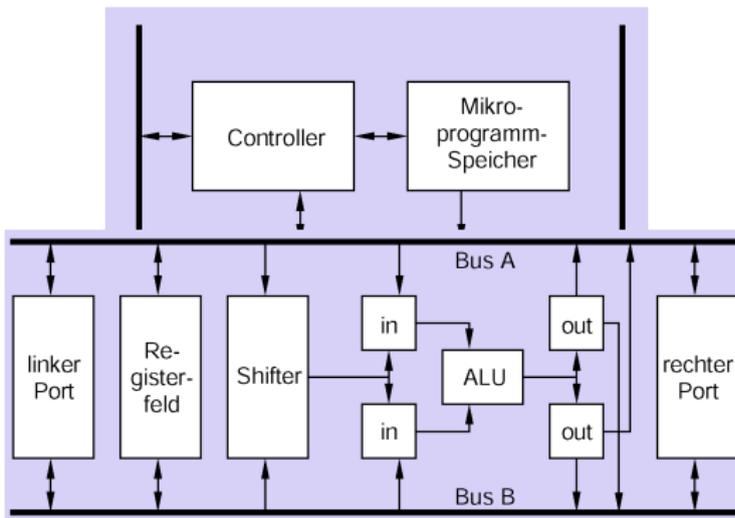
Intro

Entwicklung

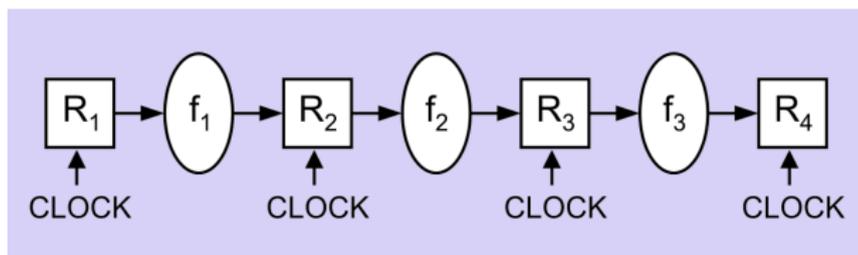
Orga

Entwurf

Methoden



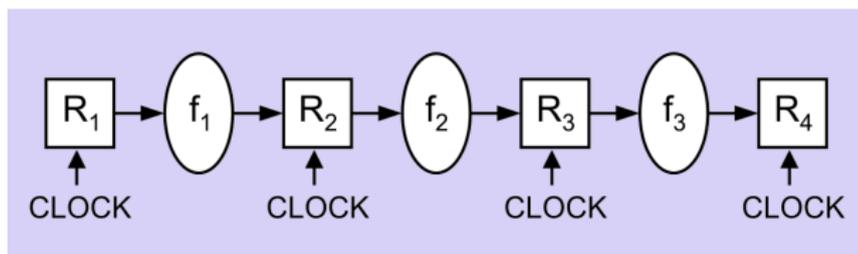
Register-Transfer-Ebene (RTL)



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

- Sehr wichtige Entwurfsebene
- Fließbandverarbeitung (Pipelines) / Automatenetze
- Effiziente Umsetzung in Hardware **automatisch** möglich

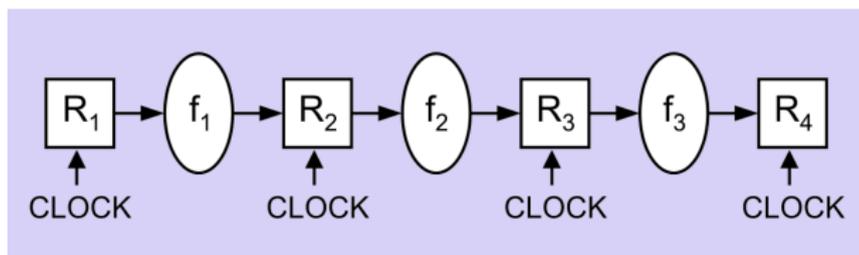
Register-Transfer-Ebene (RTL)



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

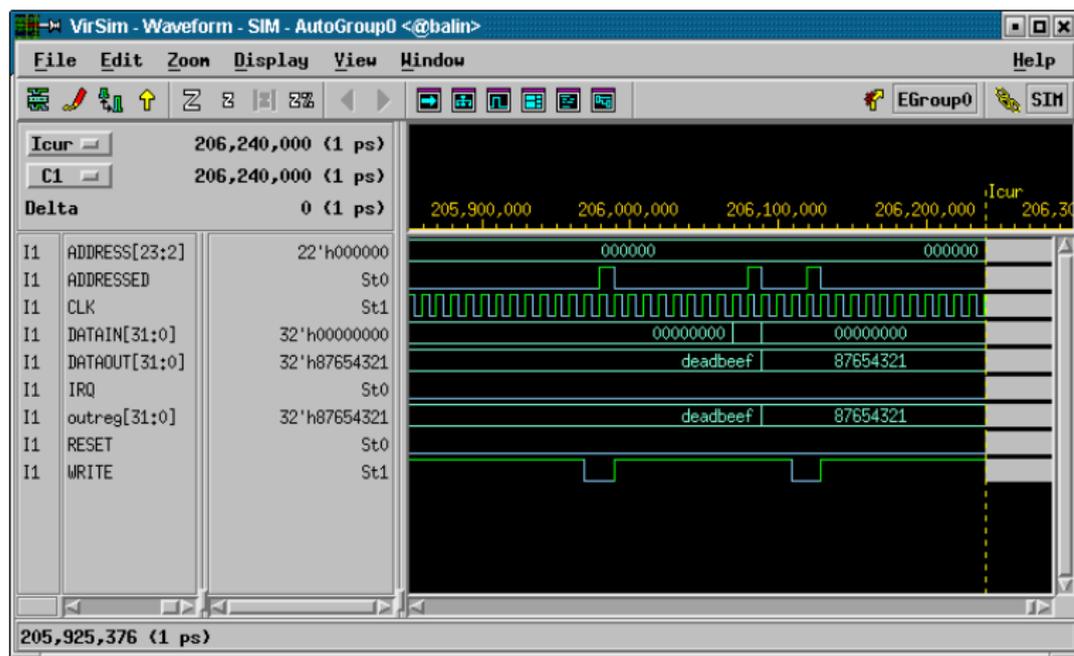
- Sehr wichtige Entwurfsebene
- Fließbandverarbeitung (Pipelines) / Automatenetze
- Effiziente Umsetzung in Hardware **automatisch** möglich

Register-Transfer-Ebene (RTL)



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

- Sehr wichtige Entwurfsebene
- Fließbandverarbeitung (Pipelines) / Automatenetze
- Effiziente Umsetzung in Hardware **automatisch** möglich



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

CMS

A. Koch

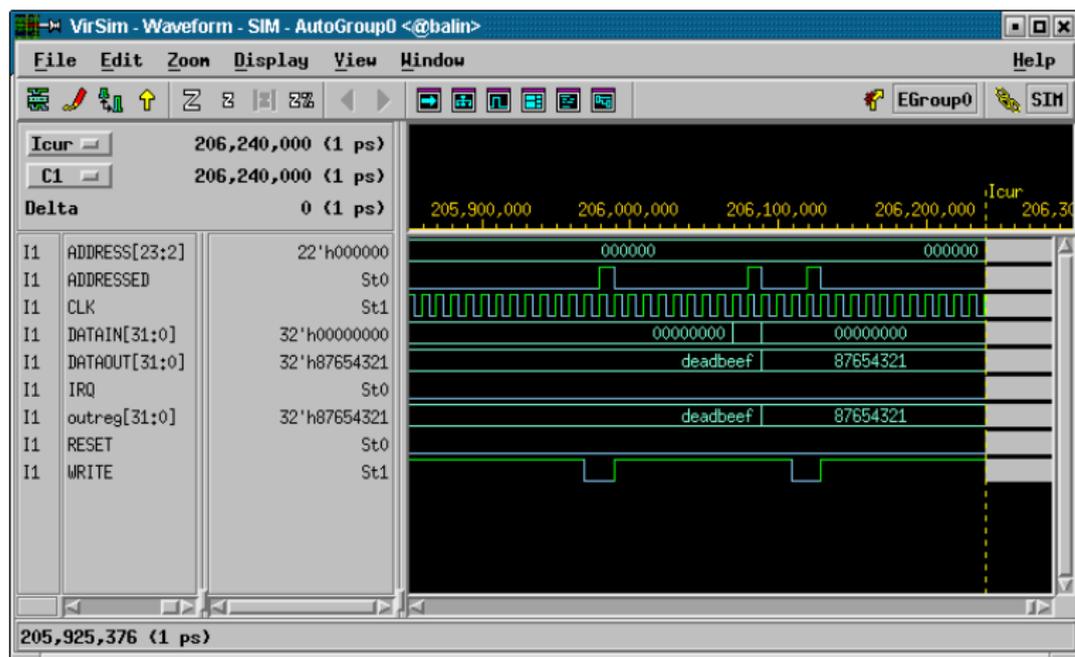
Intro

Entwicklung

Orga

Entwurf

Methoden



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

CMS

A. Koch

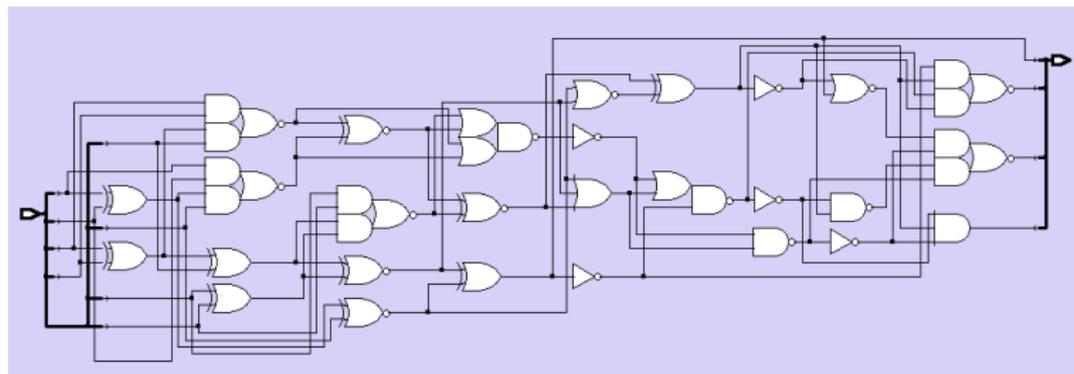
Intro

Entwicklung

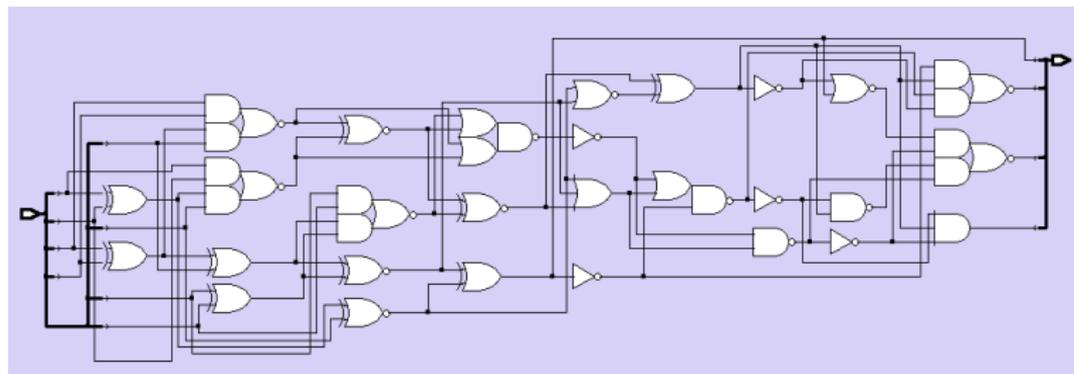
Orga

Entwurf

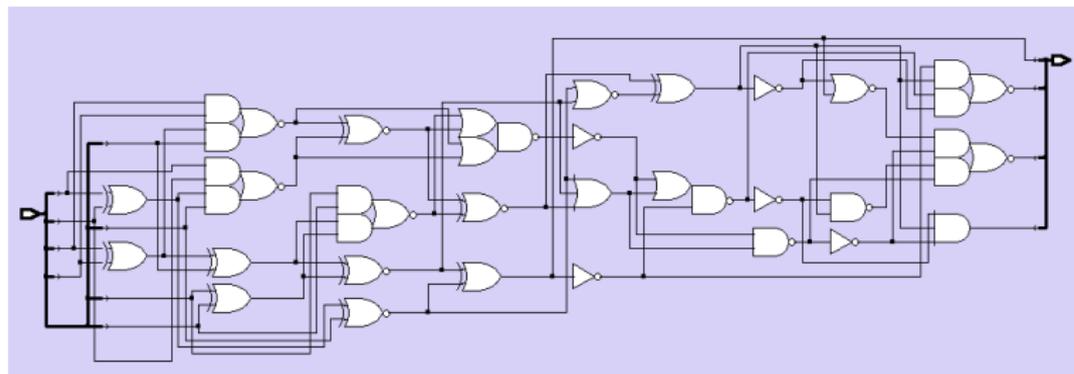
Methoden



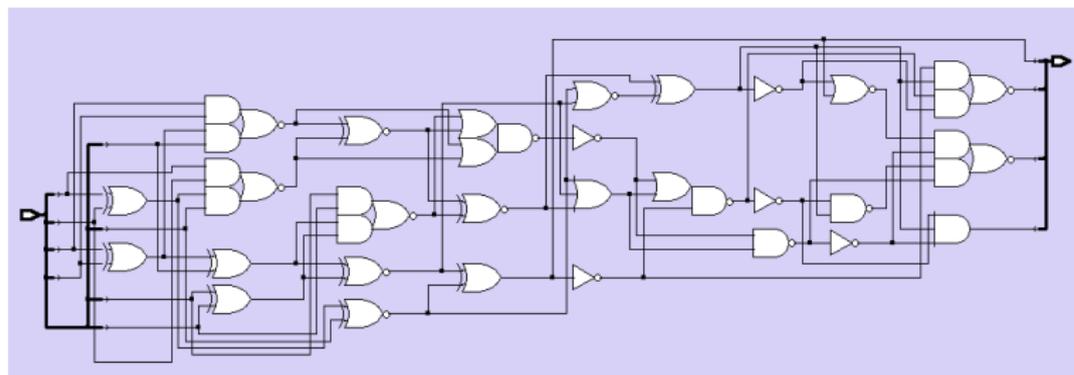
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



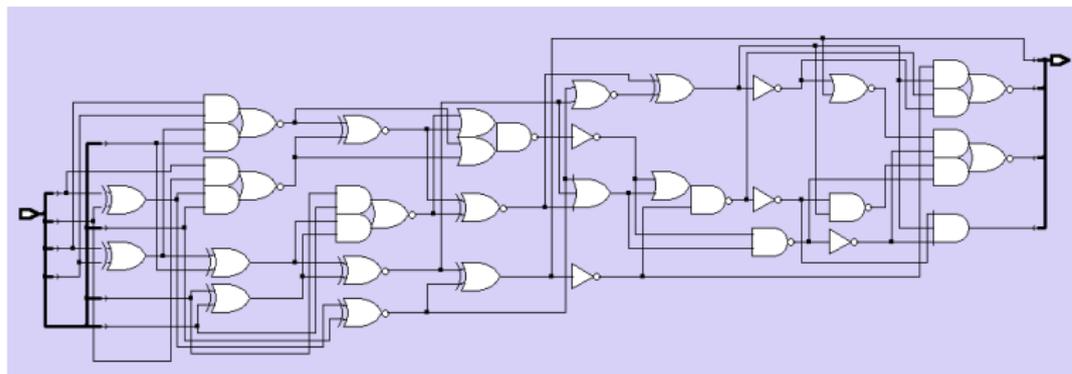
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



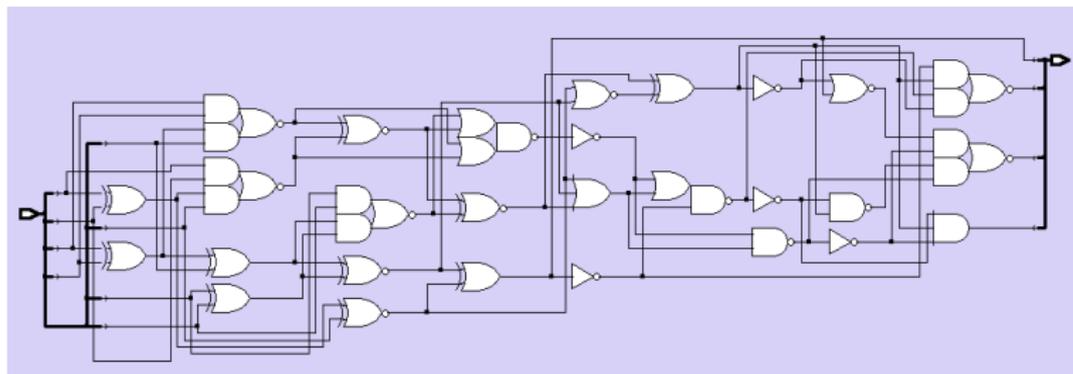
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



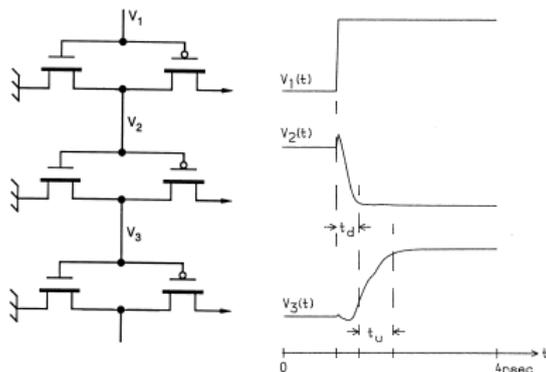
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



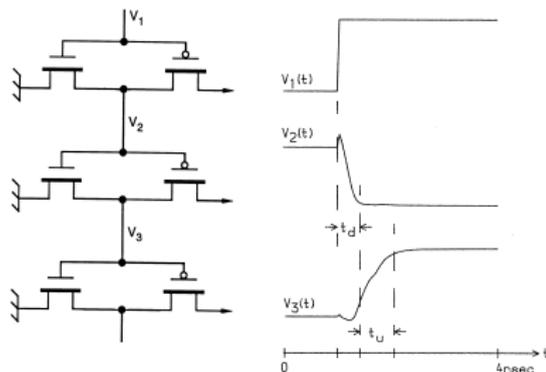
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



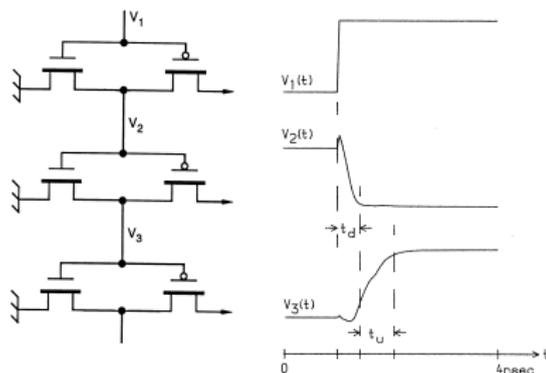
- Ergebnis der **Logiksynthese**
- **Generische** AND, NOR, Flip-Flops, etc.
- Zeitverhalten abschätzbar (aber noch sehr **ungenau!**)
- Noch keine **endgültige** Hardware-Beschreibung
 - Hängt von konkreter **Zieltechnologie** ab
 - ASIC, FPGA, Gate-Array, ...



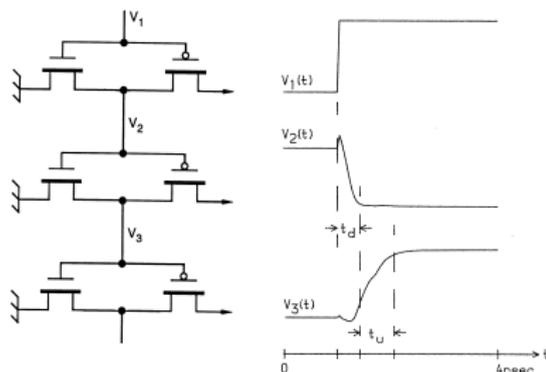
- **Schaltpläne** aus Transistoren, Widerständen, etc.
- Beim Digitalschaltungsentwurf verborgen
- Ausnahmen
 - Analoge Teilschaltungen
 - Full-Custom-Entwurf
- Analogsimulation mit exakten **Schaltzeiten**



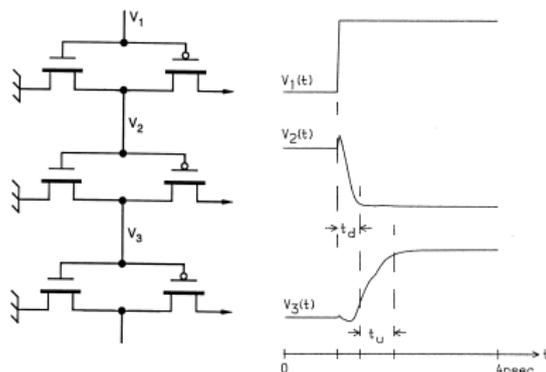
- **Schaltpläne** aus Transistoren, Widerständen, etc.
- Beim Digitalschaltungsentwurf verborgen
- Ausnahmen
 - Analoge Teilschaltungen
 - Full-Custom-Entwurf
- Analogsimulation mit exakten **Schaltzeiten**



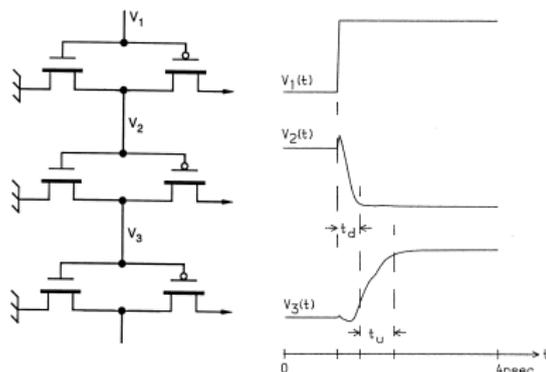
- **Schaltpläne** aus Transistoren, Widerständen, etc.
- Beim Digitalschaltungsentwurf verborgen
- Ausnahmen
 - Analoge Teilschaltungen
 - Full-Custom-Entwurf
- Analogsimulation mit exakten **Schaltzeiten**



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



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



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

Layout-Ebene

Beschreibt **exakt** Transistoren und Leitungen, hier: Inverter



CMS

A. Koch

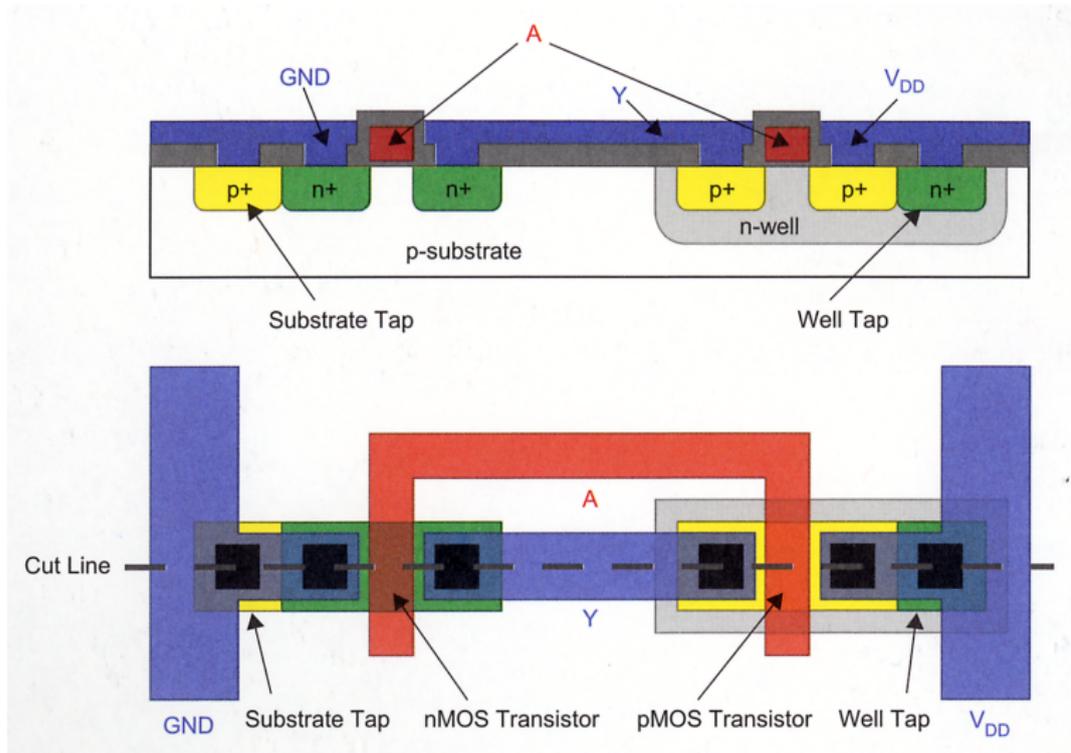
Intro

Entwicklung

Orga

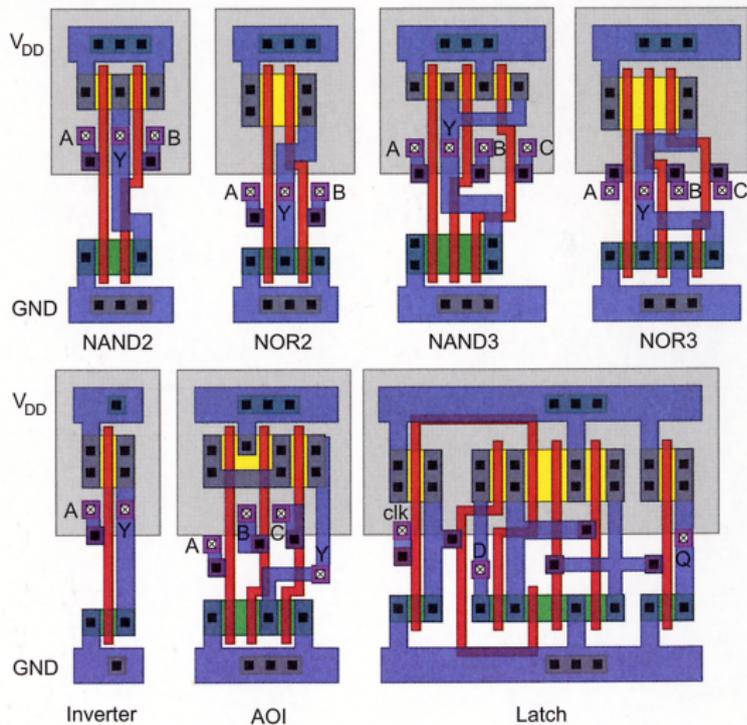
Entwurf

Methoden



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)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- 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)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- 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)



- 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)



- 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)



- 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)



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

Fundamentale Entwurfsmethoden



- **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 hochentwickelte Werkzeuge
 - Wird in der Regel nicht von Internetauth gemacht



- **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 ist ein **Prozess** (nicht ein **Produkt**)
- Wird in der Regel nicht von **Internetauto** gemacht



- **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 wird in mehreren Schritten realisiert
→ Wird in der Regel nicht von Mensch zu Mensch gemacht



- **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



- **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



- **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



- **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



- **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
 - Wird in der Regel nicht von Informatikern gemacht ☹



- **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
 - Wird in der Regel nicht von Informatikern gemacht ☹



- **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!
 - Wird in der Regel nicht von Informatikern gemacht ☺



- **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!
 - Wird in der Regel nicht von Informatikern gemacht ☺

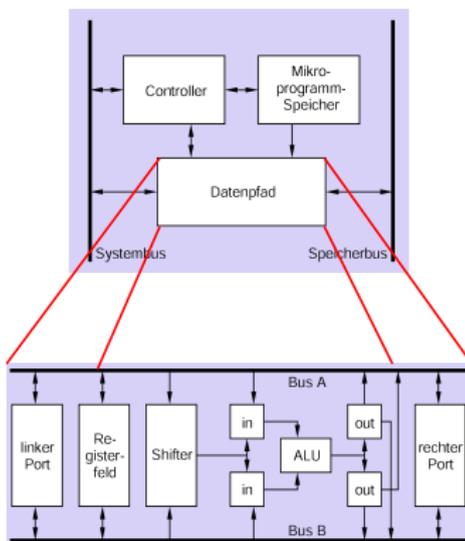


- **Weglassen** für die *aktuelle* Beschreibung *unwichtiger* Details
- Arbeiten auf unterschiedlichen Ebenen
 - Von ungenau bis sehr genau
 - Verhaltensebene, . . . , Layout-Ebene
- Beispiel: Entwurf eines MP3-Encoder-Chips
 - Manuell von funktionaler bis RTL-Ebene
 - Andere Schritte i.d.R. automatisch
- Beispiel: Entwurf von Empfangsteil für UMTS-Handy
 - Manuell von funktionaler bis **Layout**-Ebene
 - Sehr **komplizierte** Chips
 - Entwurf erfordert hochspezialisierte Kenntnisse!
 - Wird in der Regel nicht von Informatikern gemacht ☺

Hierarchische Zerlegung



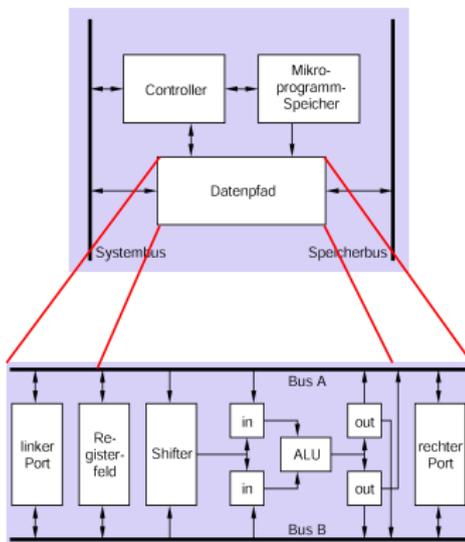
- Aufteilen eines Problems in kleinere Unterprobleme
 - Ganz alte Idee: divide et impera (Philip II, -381 ... -335)
- Auch **rekursiv** anwendbar
- Damit entsteht eine **Hierarchie** von Zerlegungen



Hierarchische Zerlegung



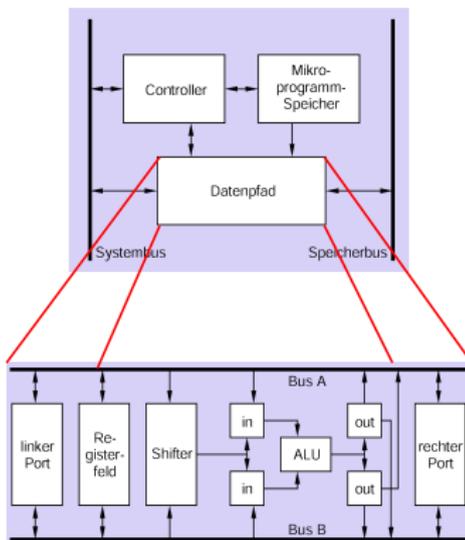
- Aufteilen eines Problems in kleinere Unterprobleme
 - Ganz alte Idee: divide et impera (Philip II, -381 ... -335)
- Auch **rekursiv** anwendbar
- Damit entsteht eine **Hierarchie** von Zerlegungen



Hierarchische Zerlegung



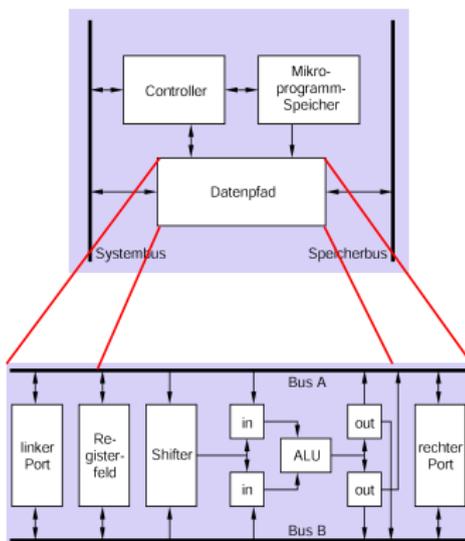
- Aufteilen eines Problems in kleinere Unterprobleme
 - Ganz alte Idee: divide et impera (Philip II, -381 ... -335)
- Auch **rekursiv** anwendbar
- Damit entsteht eine **Hierarchie** von Zerlegungen



Hierarchische Zerlegung



- Aufteilen eines Problems in kleinere Unterprobleme
 - Ganz alte Idee: divide et impera (Philip II, -381 ... -335)
- Auch **rekursiv** anwendbar
- Damit entsteht eine **Hierarchie** von Zerlegungen



Reguläre Zerlegung

Gezielte **Vervielfältigung** von Komponenten



CMS

A. Koch

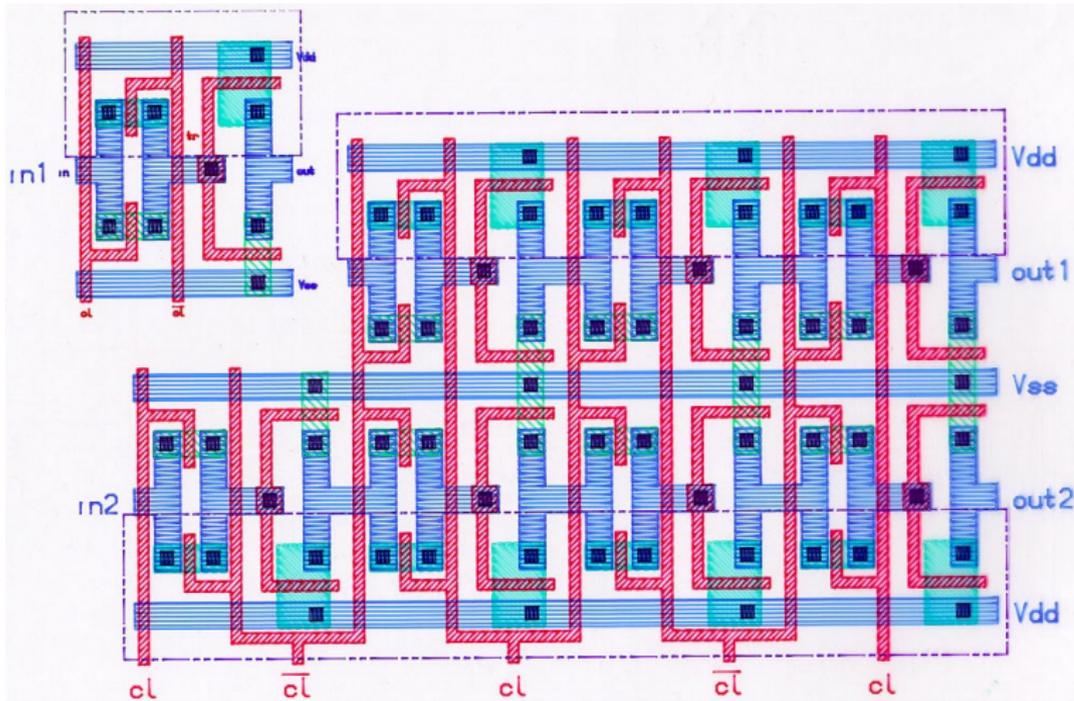
Intro

Entwicklung

Orga

Entwurf

Methoden





CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- **Entwicklungen in der Mikroelektronik**
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- **Entwicklungen in der Mikroelektronik**
- **Entwurfsebenen**
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Entwicklungen in der Mikroelektronik
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Entwicklungen in der Mikroelektronik
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Entwicklungen in der Mikroelektronik
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Entwicklungen in der Mikroelektronik
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog 2001



CMS

A. Koch

Intro

Entwicklung

Orga

Entwurf

Methoden

- Entwicklungen in der Mikroelektronik
- Entwurfsebenen
- Fundamentale Entwurfstechniken

Ausblick auf nächsten Block

- Abstieg in Verilog, aufbauend auf TGD12
- Konzentration auf Untermenge für Hardware-Synthese
- Neuere Sprachversion als in TGD12: Verilog **2001**