

Rechenanlagen

SS 2019

- **Vorlesung:** Mo 12:00-13:30, Fr 10:15-11:45
- **Dozent:** Prof. Dr. Reiner Kolla
- **Assistent:** M.Sc. Johannes Mühr



Organisatorische Anmerkungen

- Material zu Vorlesung und Übungen erhältlich über uniweite eLearning-Plattform **Wuecampus**
- Zum Login ist ein Novell-Account des Rechenzentrums notwendig. Informationen dazu und zur Anmeldung erhältlich über
<https://www.rz.uni-wuerzburg.de>
- Übungsanmeldung nach ***Losverfahren*** bis einschließlich **1.5.2019** über **Wuestudy** möglich
- Adresse um zum Kurs Rechenanlagen zu gelangen
<https://wuecampus2.uni-wuerzburg.de>

Zugangsschlüssel für den Kurs: ***FlipFlop***

Organisatorische Anmerkungen

- Über eLearning-Plattform Wuecampus finden Sie z.B. folgende Materialien:
 - **Skript zur Vorlesung (Powerpoint, PDF)**
Achtung: Skript wird ggf. nachträglich aktualisiert !
 - **Aufgabenblätter**
 - **FAQs, HOWTOs, Weblinks u.a.**
- **Übungsbetrieb** (1. Übung am 16.04.2018 - Organisatorisches)
- **Aufgabenblätter** (immer Montags)
 - wöchentlich, Bearbeitungszeit: 1 Woche (i.d.R.)
 - Zahl der Bearbeiter pro Lösung: min: 2 / max: 3
 - mindestens 2 mal „vorrechnen“ in der Übung und
 - 50% der Aufgabenpunkte

sind hinreichend für das Bestehen des **Tutoriums Rechenanlagen** bzw. das Erreichen des **Bonus**.
Das **Modul Rechenanlagen** wird durch eine Klausur geprüft.

Organisatorische Anmerkungen

➤ Programmieraufgaben

Wir empfehlen den Public Domain Simulator **GHDL** zur Simulation.

Download und Installation siehe Download- und HOWTO-Bereich des Kurses Rechenanlagen bei der eLearning-Plattform Moodle.

Aufgaben sind aber prinzipiell auch ohne Rechner lösbar, d.h. nur anhand des VL-Skripts und der Beispielprogramme. Erwartet werden keine zu 100% syntaktisch korrekten Quellcodes.

Abgabe von Programmieraufgaben: *in schriftlicher Form*

Bücherliste

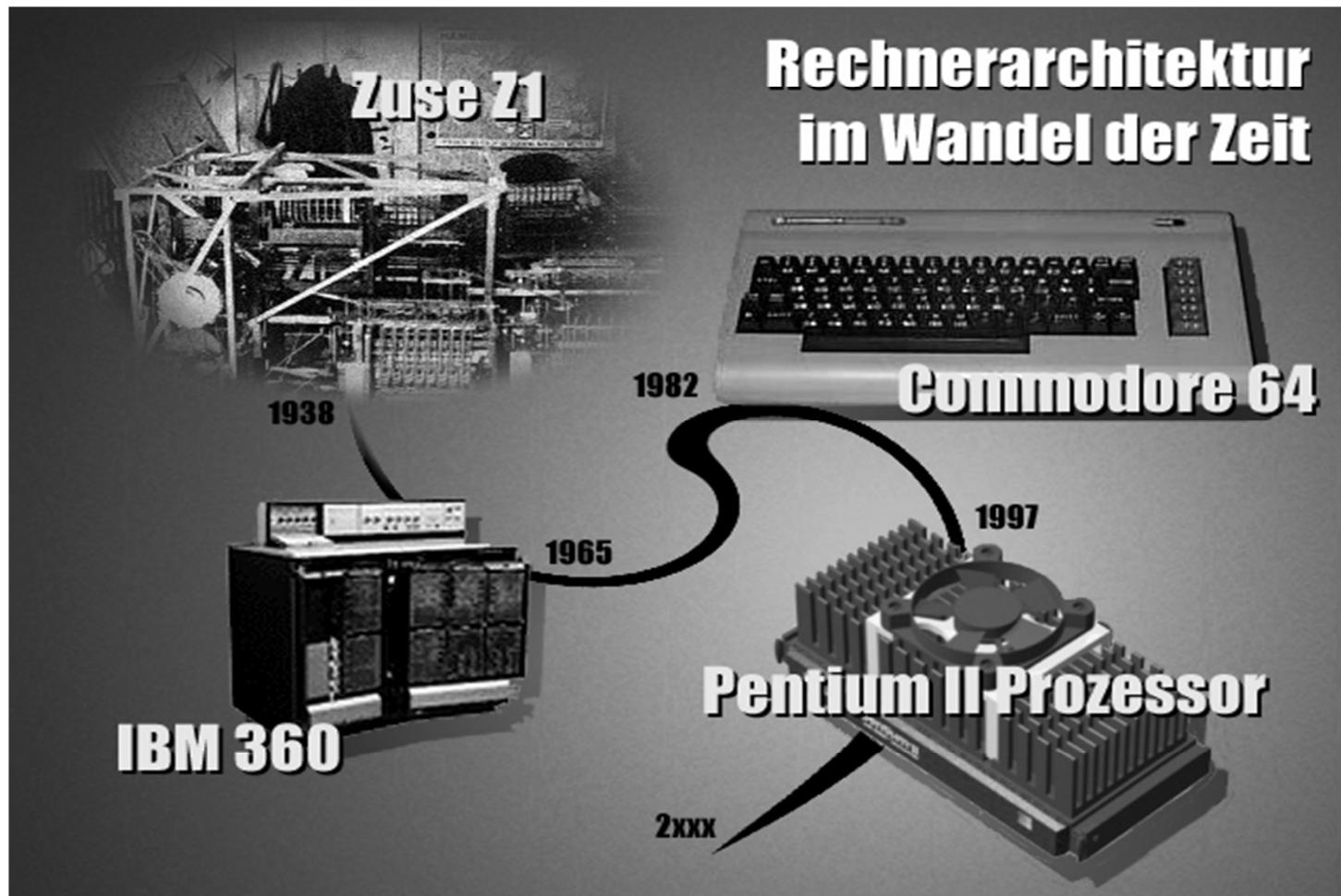
- I. **Structured Computer Organization**, 4th ed
A.S. Tanenbaum & J.R. Goodman, Prentice Hall
- II. **Computer Architecture, a Quantitative Approach**,
D. Patterson & J.L. Hennessy, Morgan Kaufmann Publishers Inc
- III. **Technische Informatik** Band 2: Entwurf digitaler Schaltungen, *Günter Kemnitz, Springer (hat VHDL als begleitende HDL)*

Buch zu VHDL

Komplettkurs VHDL,
P. Molitor und J. Ritter,
Oldenbourg Verlag 2013



1.0 Prolog



Eniac (1946)



- 30 Tonnen, 3m hoch, 24m breit
- 18 000 Elektronenröhren
- Multiplikationszeit: 3ms

1943-50: Was man sich damals so dachte!

- **Thomas Watson (IBM)**
„I think there is a world market for maybe five computers.”

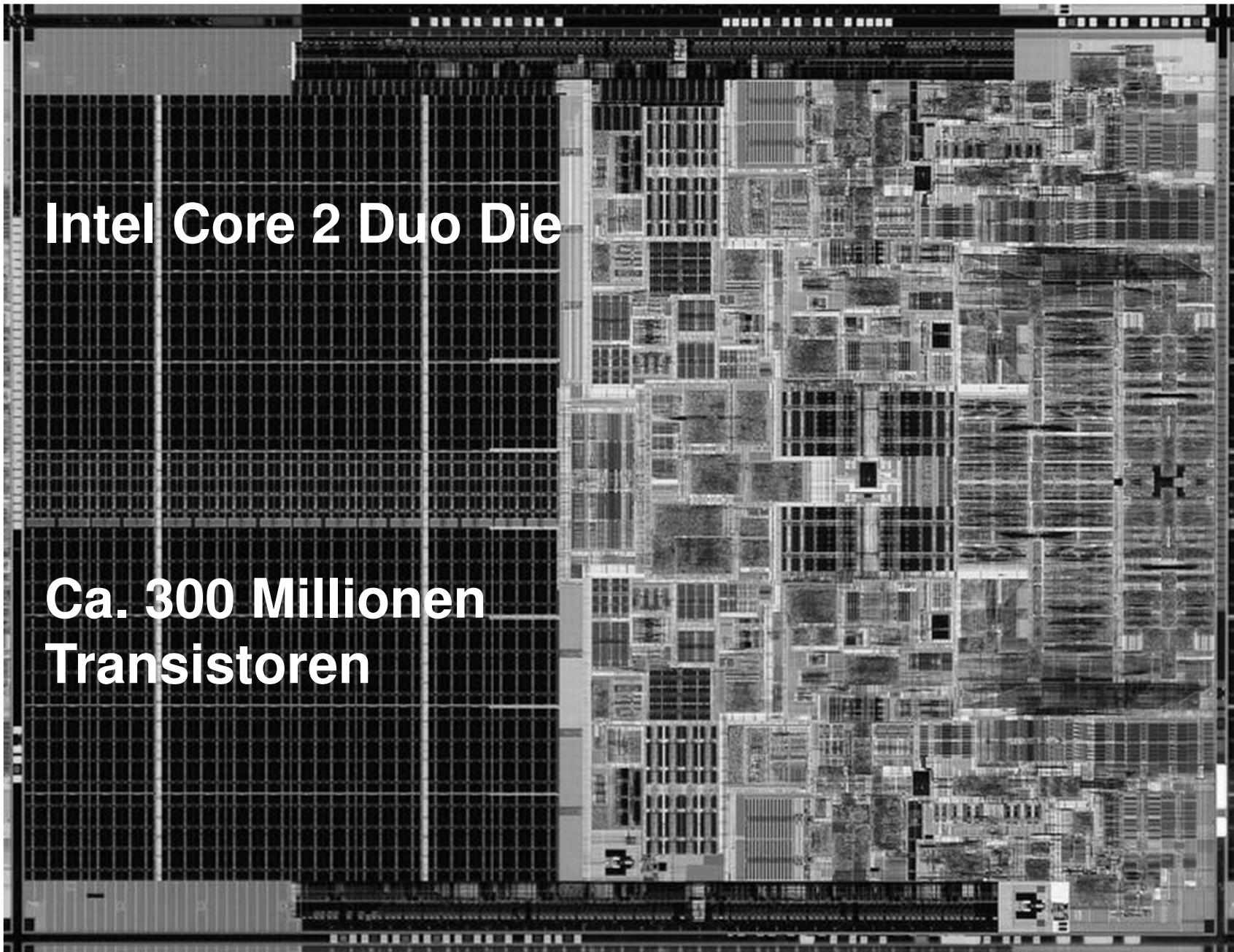
- **Popular Mechanics:**
„Computers in the future may weigh no more than 1.5 tons.”

2000 -- und heute!

- ENIAC on a chip
- Laptop = alle Rechner der Welt von 1950

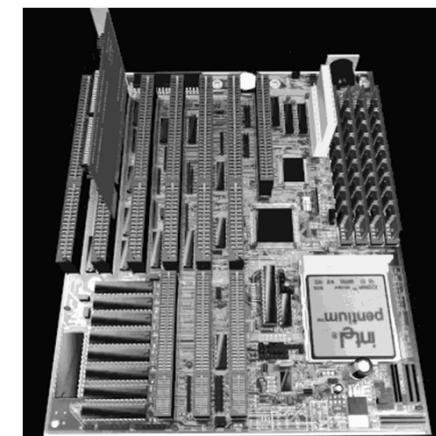
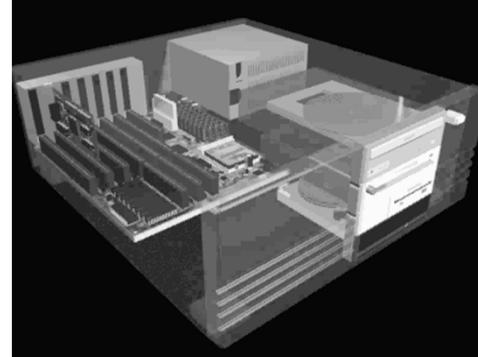
Intel Core 2 Duo Die

**Ca. 300 Millionen
Transistoren**



Lernziel in dieser Vorlesung

"Wie werden Rechner aufgebaut, warum werden sie so aufgebaut und wie funktionieren sie?"



Lernziel in dieser Vorlesung -- ff

Rechner werden heutzutage durch **höchstintegrierte Schaltungen** realisiert.

Frage

Prozessor =

Maskendaten für einen Halbleiterfertigungsprozess

+ Gehäusespezifikation

+ Bondplan

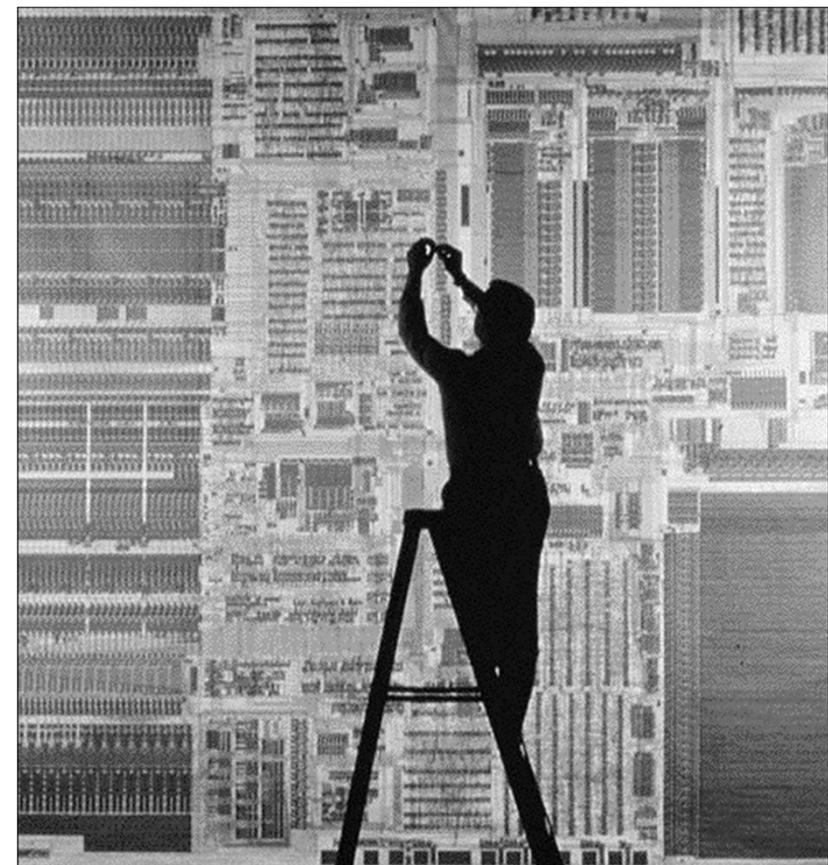
Der Entwurf von Rechnern erfordert die Benutzung vieler Softwarewerkzeuge. Aufbau und Arbeitsweise von Rechenanlagen sind demnach nur noch im Zusammenhang mit diesen Werkzeugen zu verstehen. Wir tragen dieser Tatsache in dieser Vorlesung Rechnung.

Problem: First time success

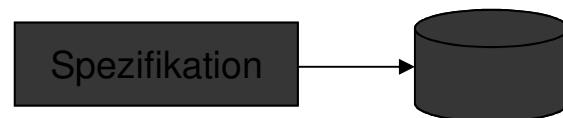
Stelle sicher, dass die erzeugten Maskendaten von Anfang an ein korrekt arbeitendes System liefern.

Hohe Anforderungen an die Überprüfung und Sicherstellung der Korrektheit vor der Anfertigung des ersten physischen Exemplars.

(so wird das nichts!)



Vorgehensweise beim Entwurf -- Designflow

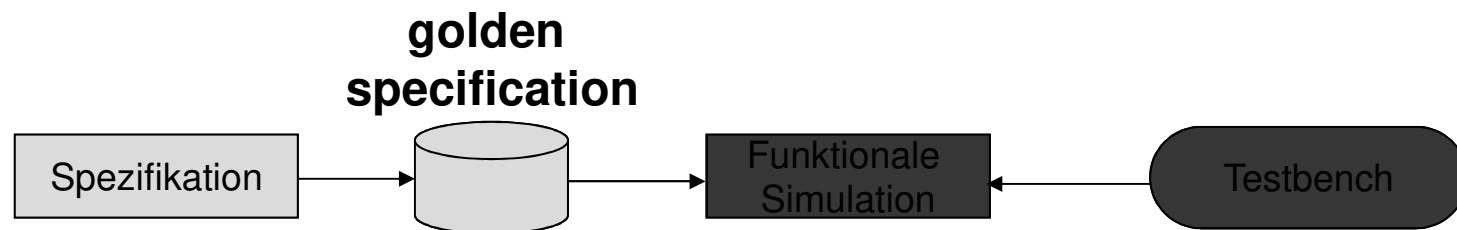


Spezifikation

- **Schaltplaneingabe (Schematic Entry)**
 - Graphikeditor zur Eingabe einer Entwurfsskizze der Schaltung
 - Herstellung der Verbindungen von Blöcken
 - Strukturbeschreibung
- **Beschreibung mittels einer Hardwarebeschreibungssprache (HDL)**
 - Hardware wird wie Software durch Programme spezifiziert
 - Struktur- oder Verhaltensbeschreibung

In dieser Vorlesung werden wir begleitend zum Stoff eine Hardwarebeschreibungssprache (Hardware Description Language = HDL) einführen. Sie ist eine echte Teilmenge des IEEE Standards **VHDL93** (VHSIC (= very high speed integrated circuit) Hardware Description Language). Wir nennen sie **WüHDL**.

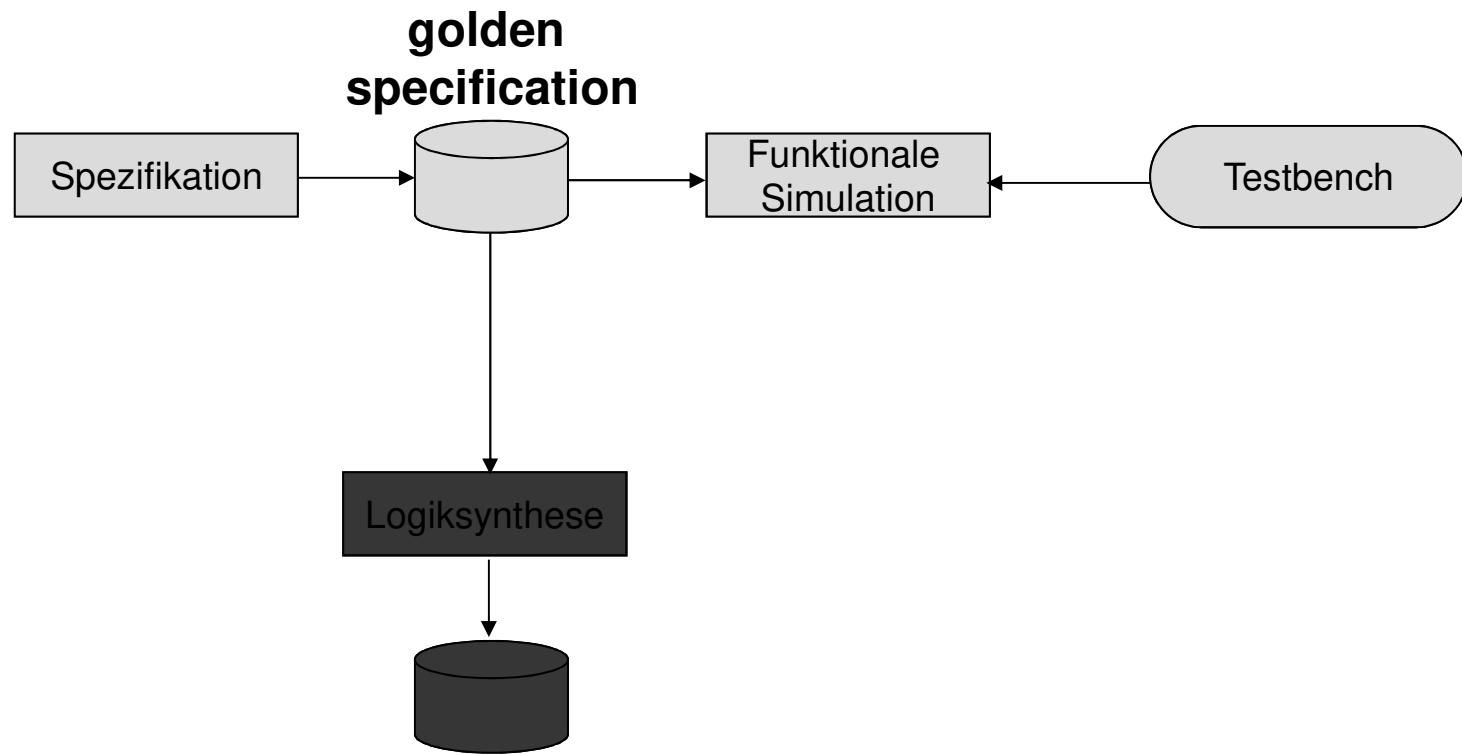
Designflow



Korrektheitsprüfung durch Simulation

- **Funktionale Simulation der Spezifikation**
= Überprüfung der logischen Korrektheit der Spezifikation
- **Vorgehensweise:**
 - Anlegen ausgewählter Eingabemuster, den **Stimuli**
 - Ist / Soll-Vergleich
 - Die Simulation wird üblicherweise durch ein Programm, die **Testbench**, gesteuert

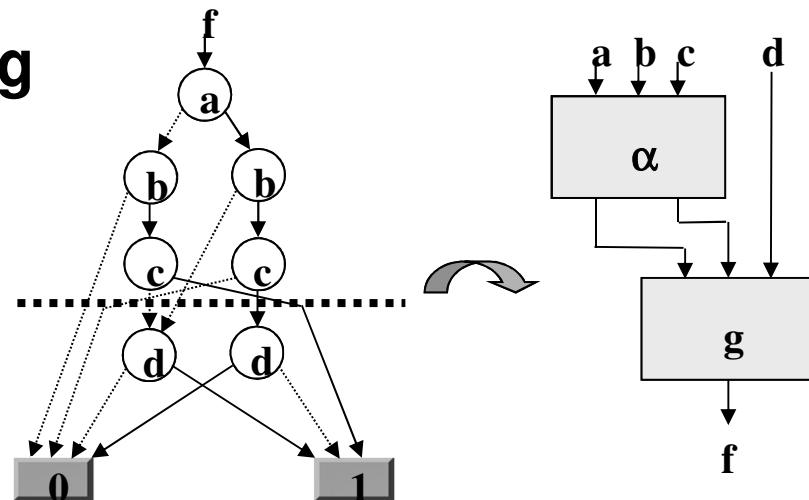
Designflow



Logiksynthese und -minimierung

Versuch, die Logik der Schaltung

- zu generieren, bzw.
- zu minimieren.



Unterscheidung zwischen

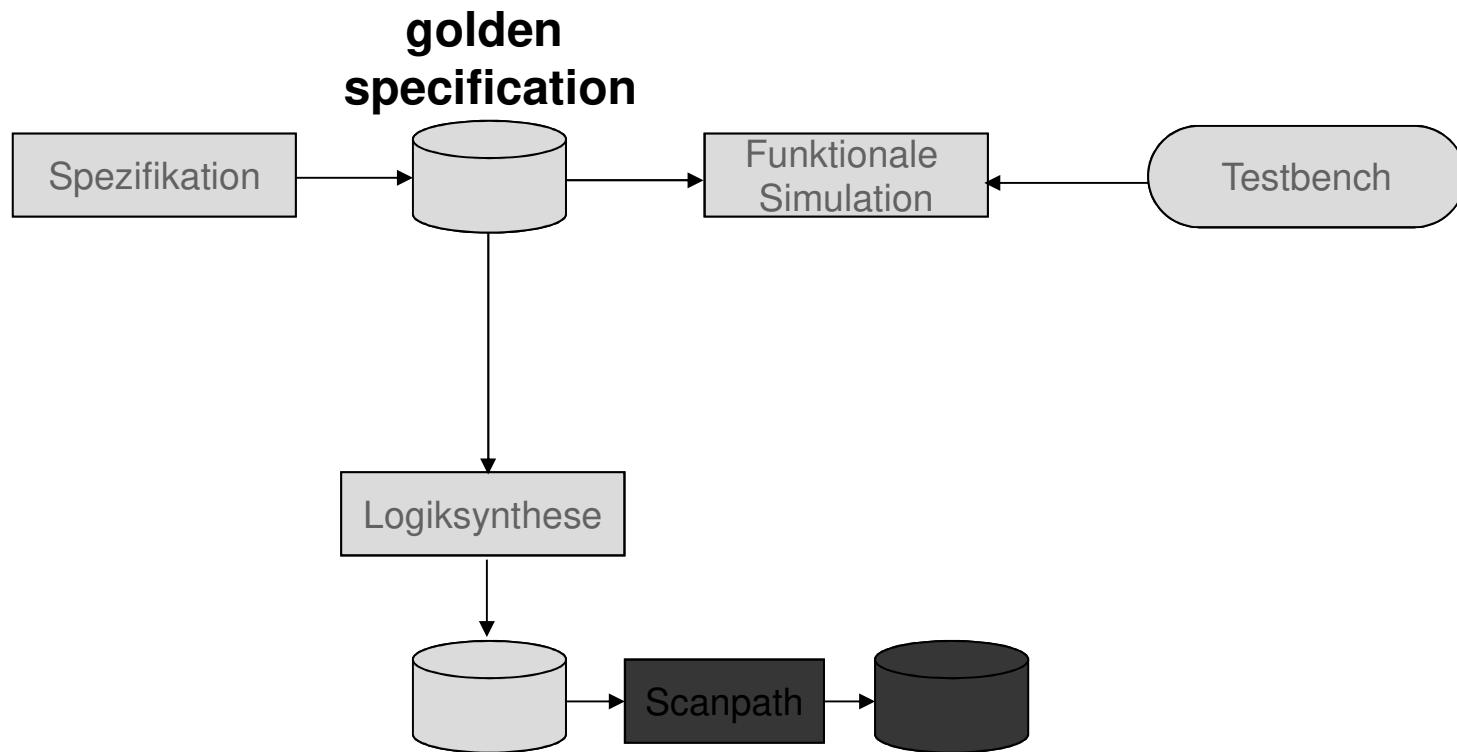
- **kombinatorischer Synthese**

= Realisierung einer Booleschen Funktion

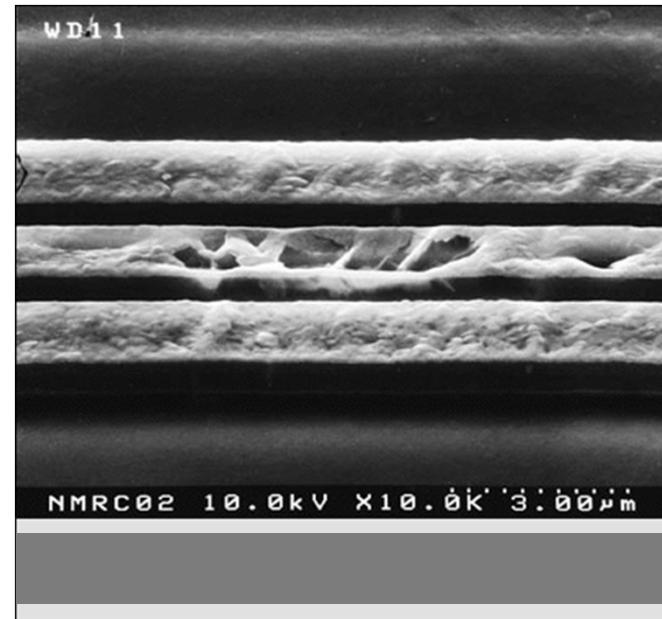
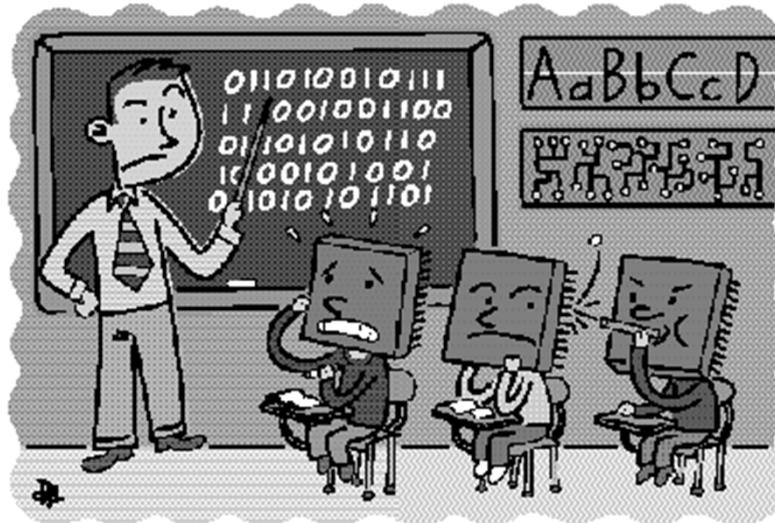
- **sequentieller Synthese**

= Realisierung eines Endlichen Automaten

Designflow

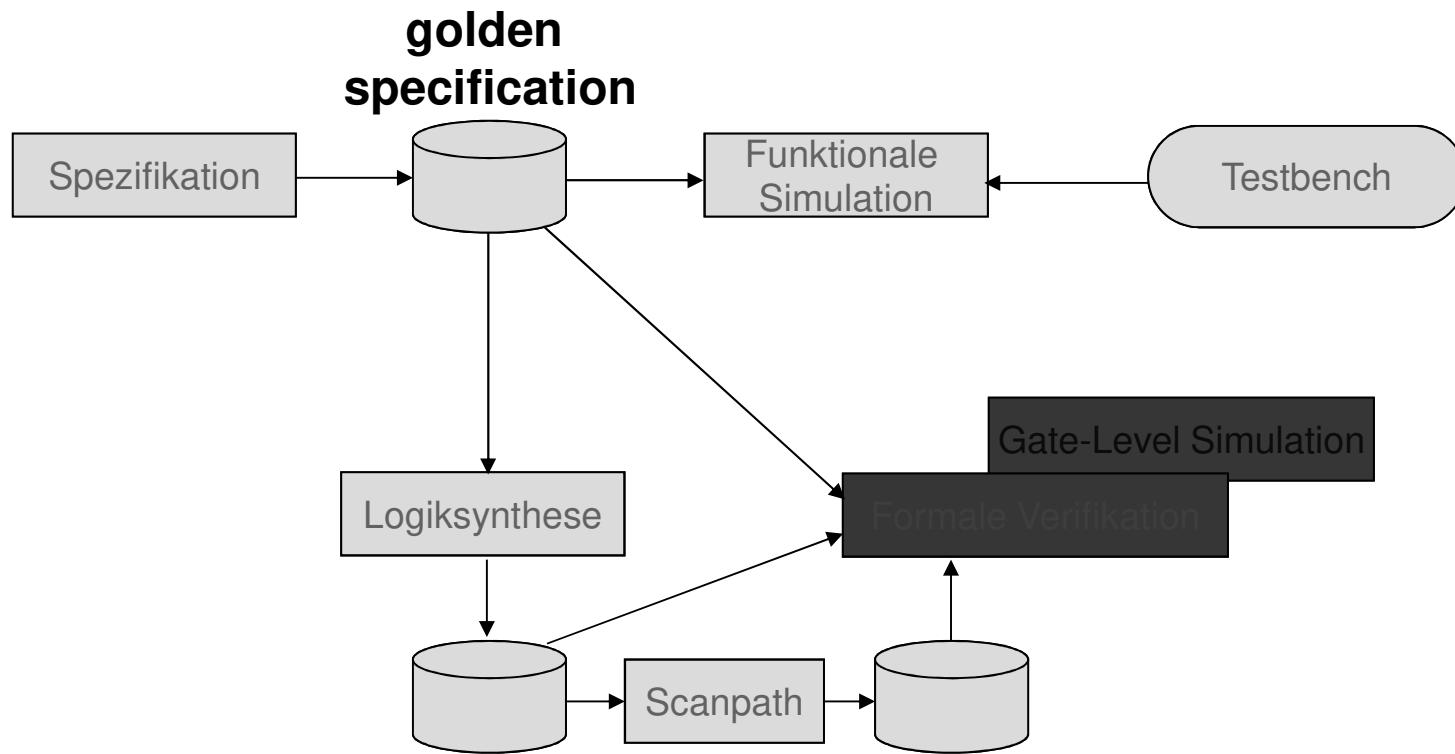


Testmustererzeugung

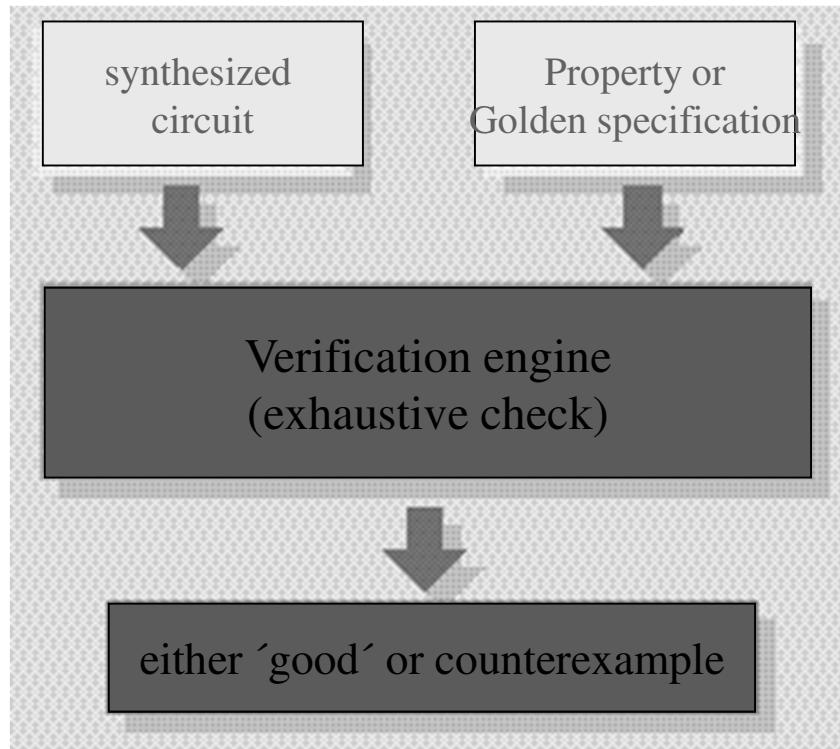


- Überprüfung, ob Spezifikation effizienten Test auf Fabrikationsfehler erlaubt
- Berechnung eines (fast) vollständigen Tests,
 - Menge von Eingabemuster, die alle Fabrikationsfehler (Fehler eines gegebenen Fehlermodells) entdeckt.

Designflow



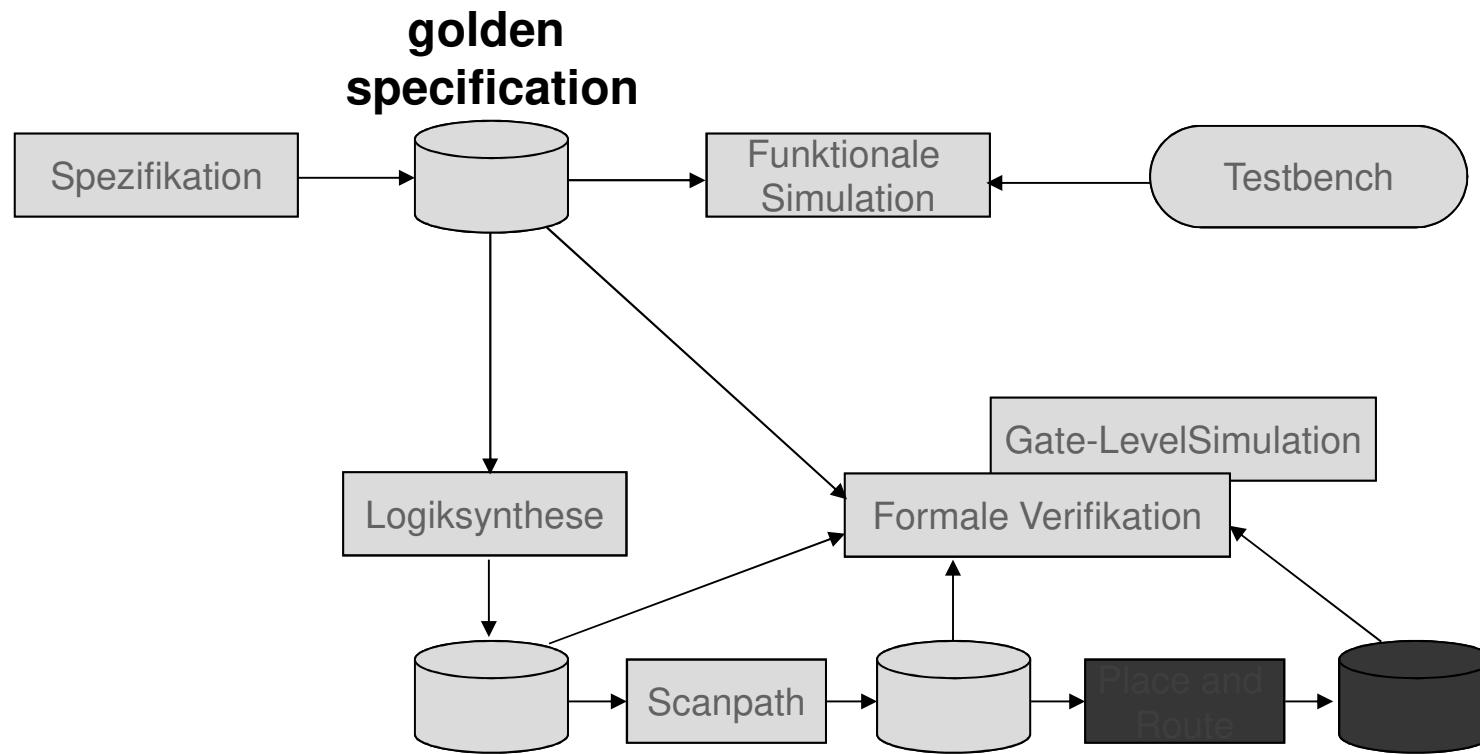
Formale Verifikation



oder:

Formaler Beweis, dass
Eigenschaft gilt oder der
goldenene Spezifikation
entspricht.

Designflow



Erstellung der Fertigungsdaten

- **Erstellung der Fertigungsunterlagen**
 - **Maskendaten: geometrische Daten, das Layout betreffend**
 - **Testprogramm für den Testautomaten**
 - **Bondplan: Zuordnung der Pads zu den Gehäusepins**