Kapitel 8 – Entwurfs- und Architekturkonzepte

1. **VLSI-Entwurf**
2. Rechnerarchitektur

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Dr. Tobias Schubert, Dr. Ralf Wimmer

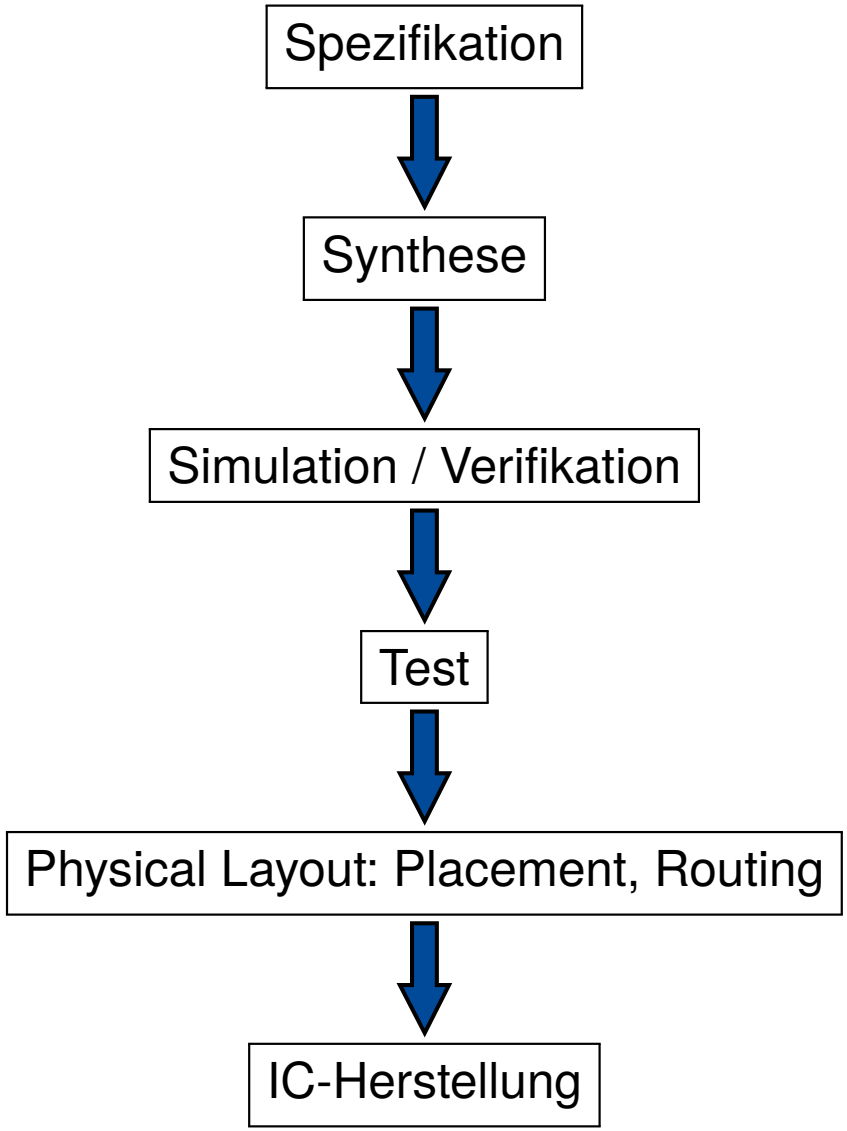
Professur für Rechnerarchitektur

WS 2016/17

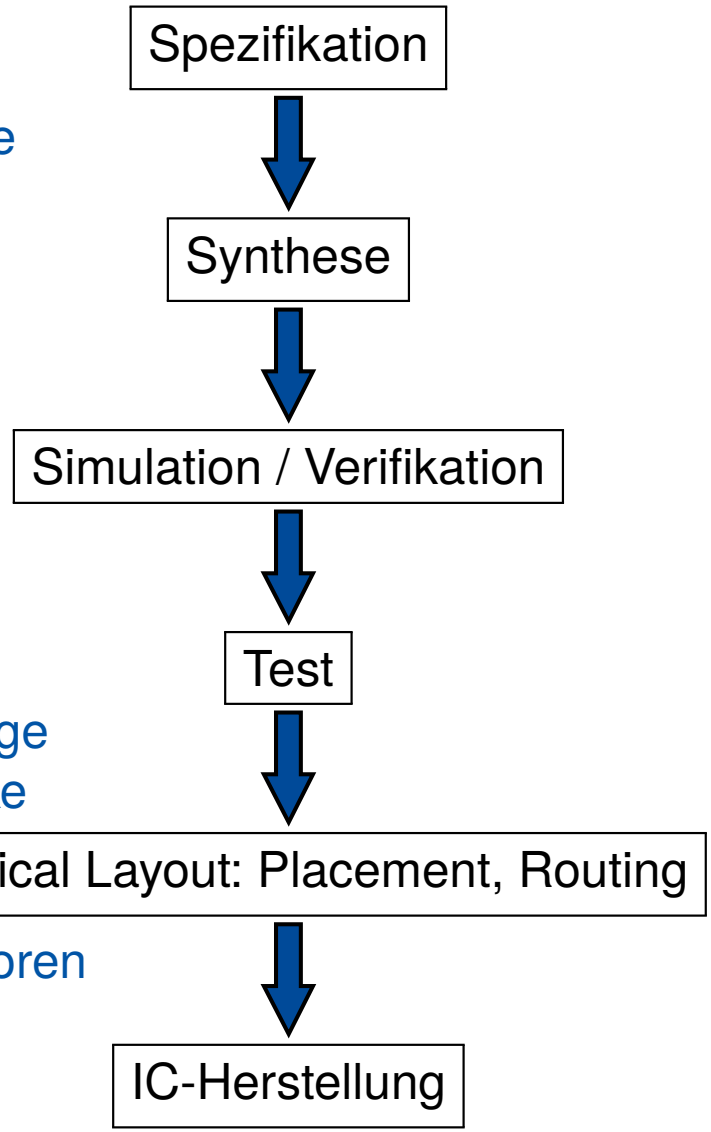
VLSI-Entwurf

- Im Schnelldurchgang ...
- VLSI: Very Large-Scale Integration
- Entwurfs-Flow kann für Mikroprozessoren oder beliebige andere Schaltungen verwendet werden!

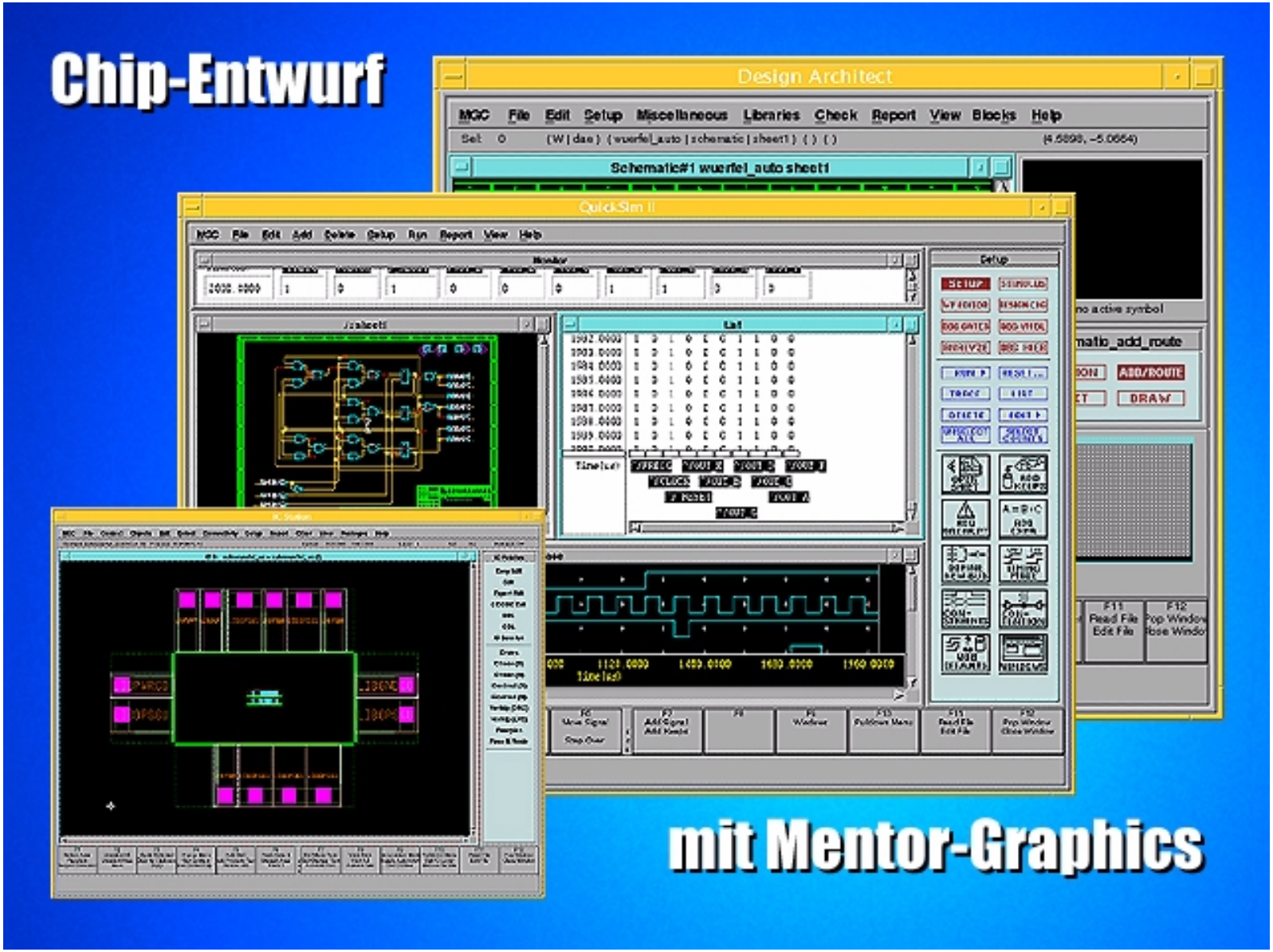
Übersicht (1/6)



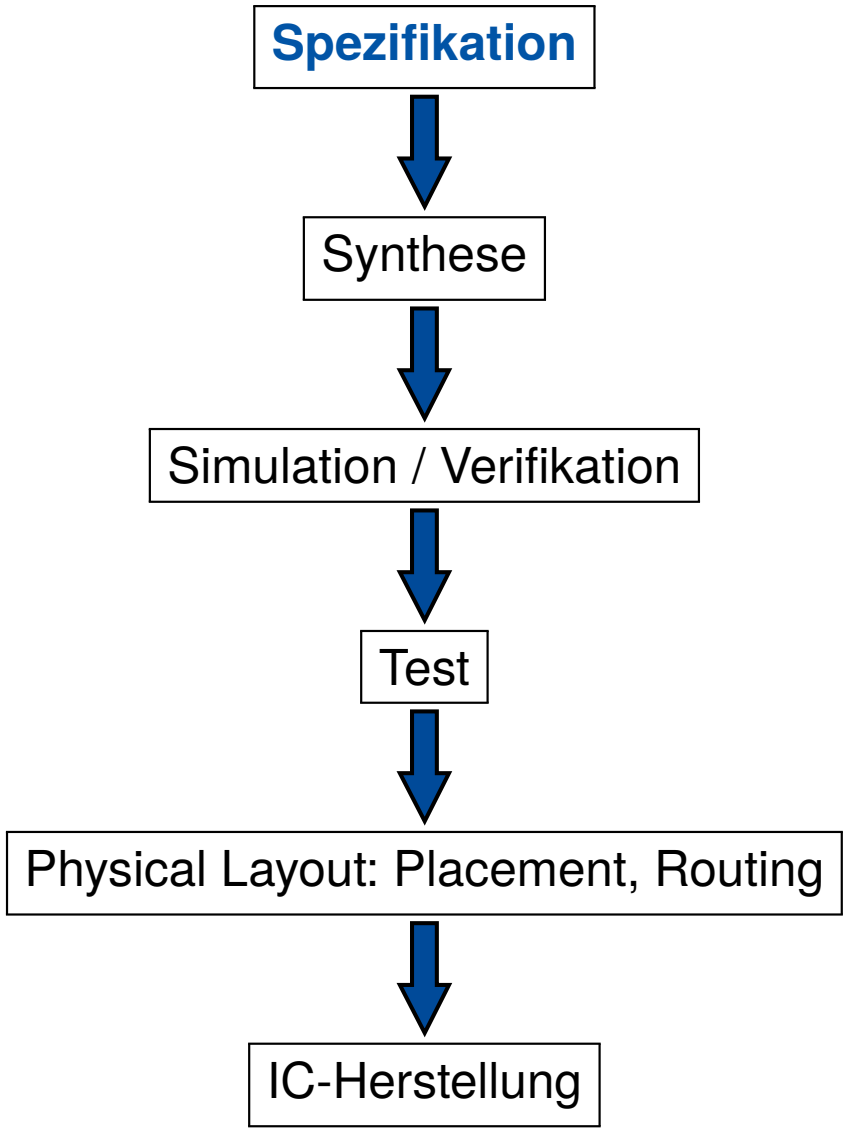
Übersicht (1/6)



Chip-Entwurf



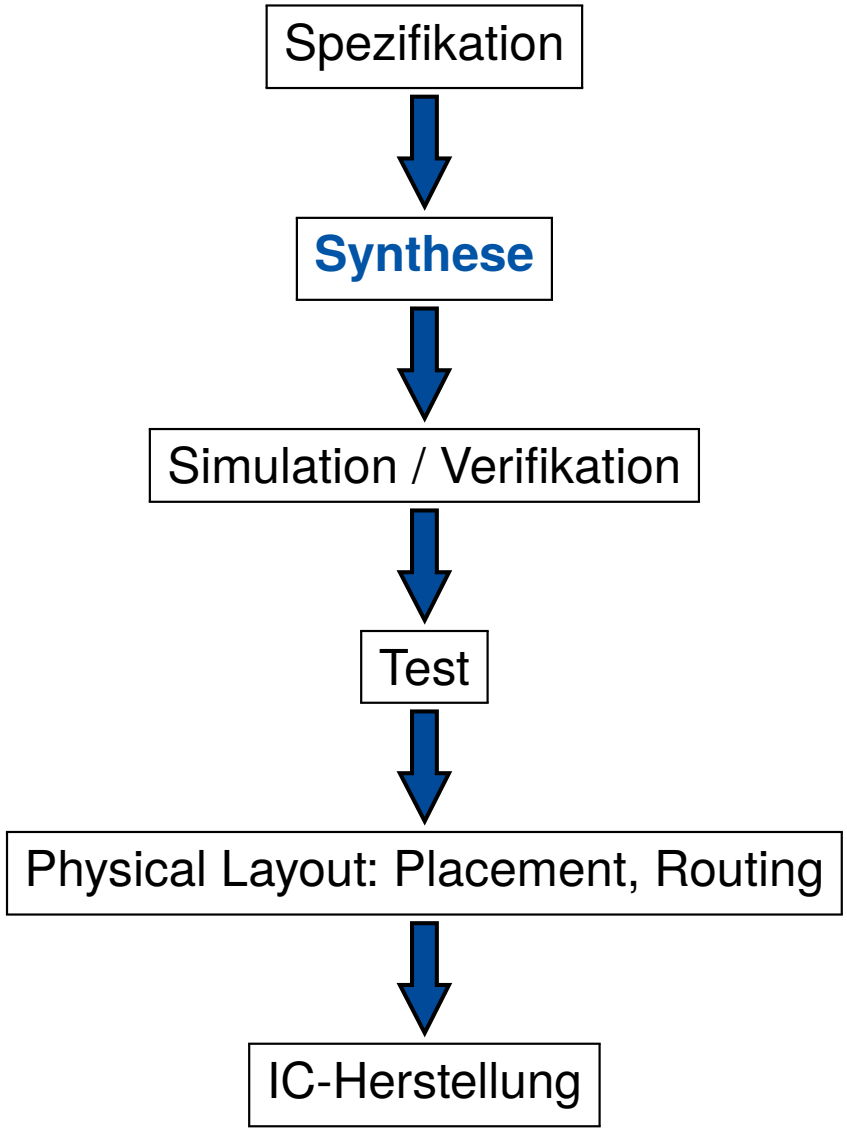
Übersicht (2/6)



Spezifikation

- Ein digitales System kann auf verschiedene Weise beschrieben werden:
  - Beschreibung als Grafik (z.B. als Schaltplaneingabe),
  - Textuelle Beschreibung als Netzliste (z.B. EDIF),
  - Textuelle Beschreibung auf Basis einer Hardwarebeschreibungssprache (z.B. VHDL, Verilog)
  - ... oder auf noch höherer Abstraktionsebene.

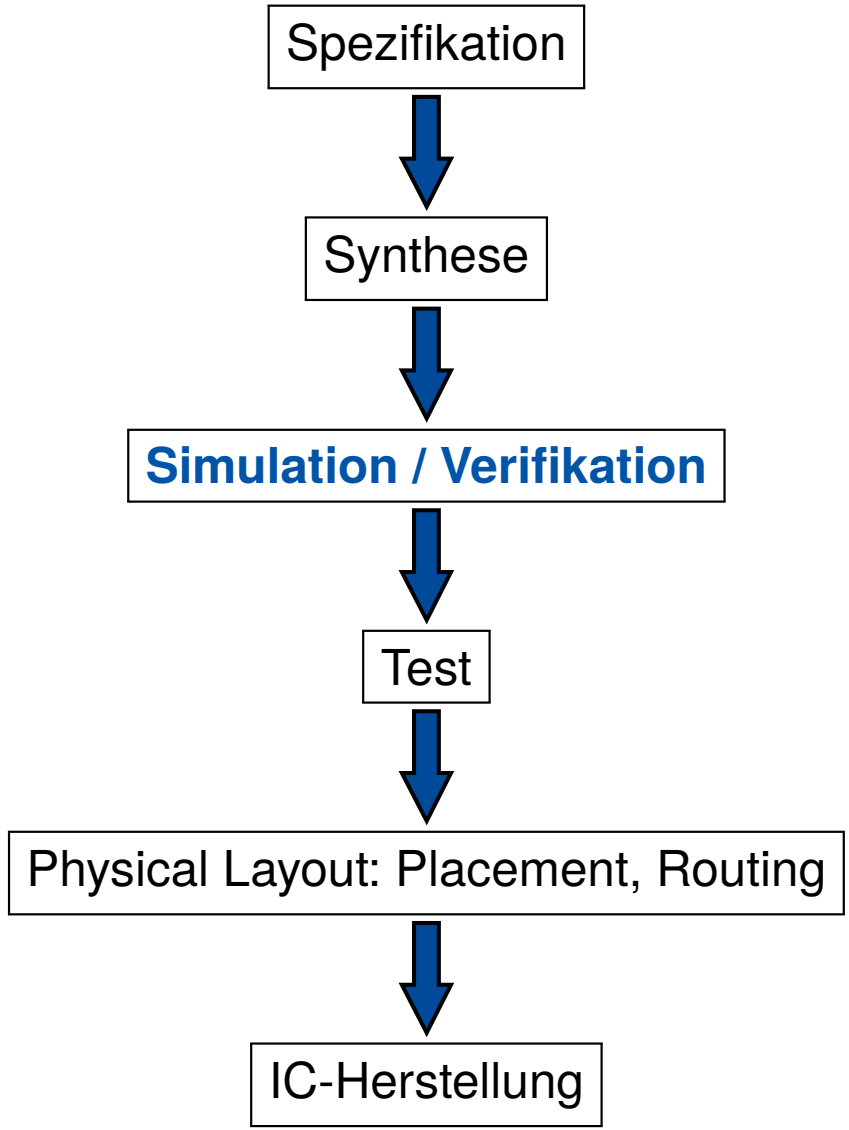
Übersicht (3/6)



Synthese

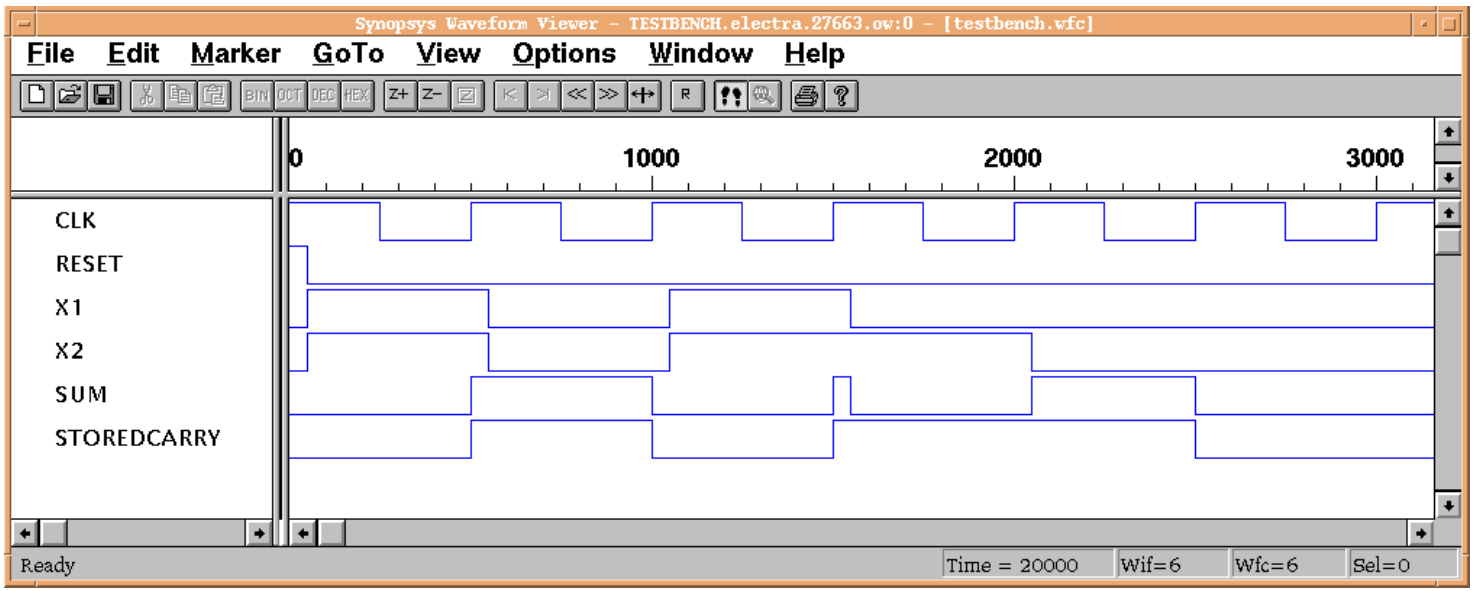
- Aufgabe der Synthese ist das Umsetzen einer Spezifikation in eine (Schaltkreis-)Realisierung.
- Syntheseschritte:
  - High-Level-Synthese (Rechnerarchitektur)
  - Logiksynthese, meist noch technologieunabhängig (siehe auch Kap. 3.3 - Zweistufige Logiksynthese mit Quine/McCluskey für PLAs)
  - Technologiemapping: Abbildung einer technologieunabhängigen Netzliste auf eine Bauteilebibliothek eines Halbleiterherstellers.

Übersicht (4/6)



Simulation

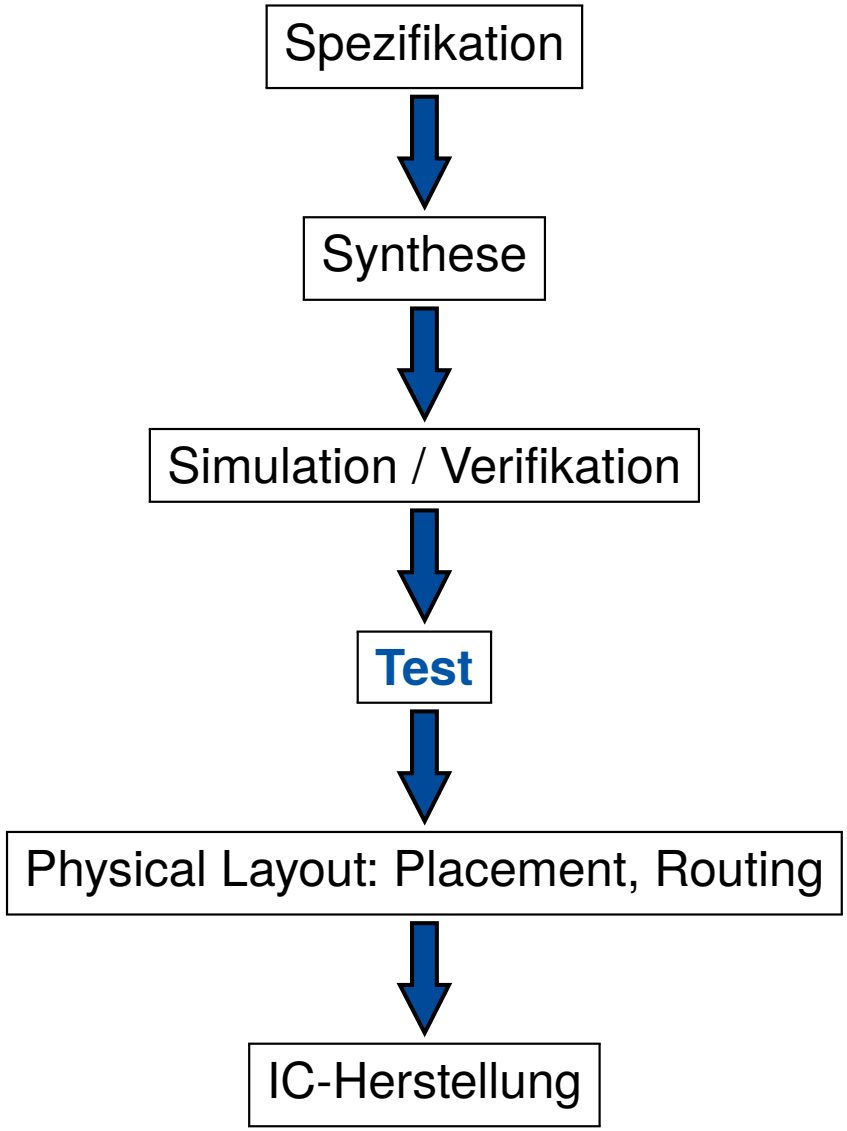
- Aufgabe von Simulation und Verifikation ist der Nachweis, dass Spezifikation und Realisierung übereinstimmen.
- Validierung durch Simulation (unvollständig!)



Verifikation

- Formale Verifikation ist der Beweis von Eigenschaften der Realisierung, durch:
  - Äquivalenztests (z.B. mit BDDs, siehe Kap. 7.3)
  - Model Checking (Nachweis von Eigenschaften, einfaches Beispiel siehe Kap. 7.3)
  - Automatentraversierung (z.B. endlicher Zustandsautomat, siehe Kap. 4.3)
- Diese und weitere Formalismen werden behandelt in der Vorlesung Rechnerarchitektur, sowie in diversen Spezialvorlesungen (Verifikation eingebetteter Systeme, Verifikation probabilistischer und hybrider Systeme, ...)

Übersicht (5/6)



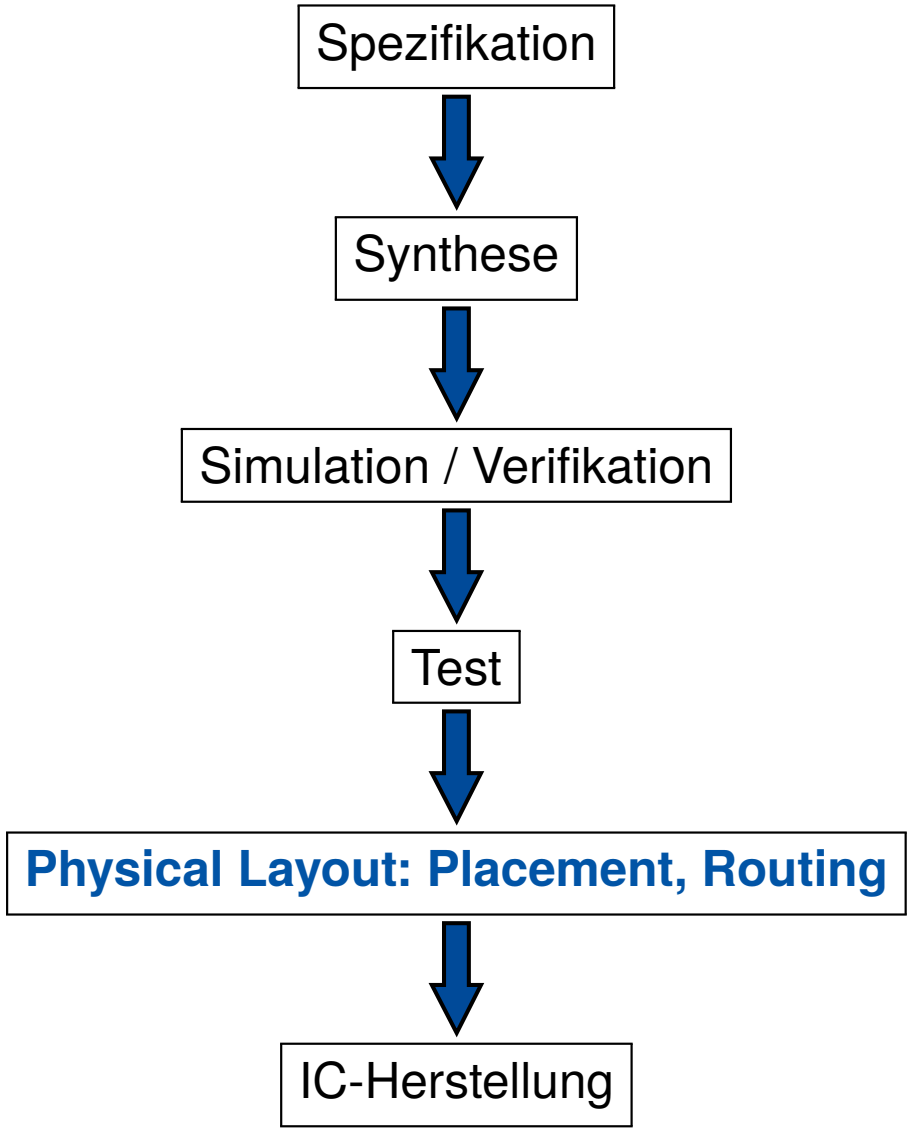
Testen (1/2)

- Aufgabe des Testens ist die Überprüfung eines Chips auf Fehler nach der Fertigung des Chips.
- **Motivation**: Es ist nicht ungewöhnlich, dass $60 - 70\%$ der gefertigten Chips fehlerhaft sind.
- **Ziel**: Aussortieren der fehlerhaften Chips.

Testen (2/2)

- Aufgabe des Testens ist die Überprüfung eines Chips auf Fehler nach der Fertigung des Chips.
- Daher vor der Fertigung der Chips: Maßnahmen zur Vorbereitung bzw. Erleichterung des Testens, z.B.
  - Berechnung von Testmustern für den späteren Fertigungstest
    - Testen der Funktionalität
    - Testen der Geschwindigkeit
    - ATPG = „Automatic Test Pattern Generation”
    - Fehlersimulation: Bewertung von gegebenen Testmustern
  - Designänderungen, um „Testbarkeit” und „Zuverlässigkeit” zu verbessern. (DFT = „Design for testability”)
  - Zusatzhardware zum Testen (BIST = „Built-in Self-Test”).
- (Ausführlich in Rechnerarchitektur und Spezialvorlesung Test und Zuverlässigkeit).

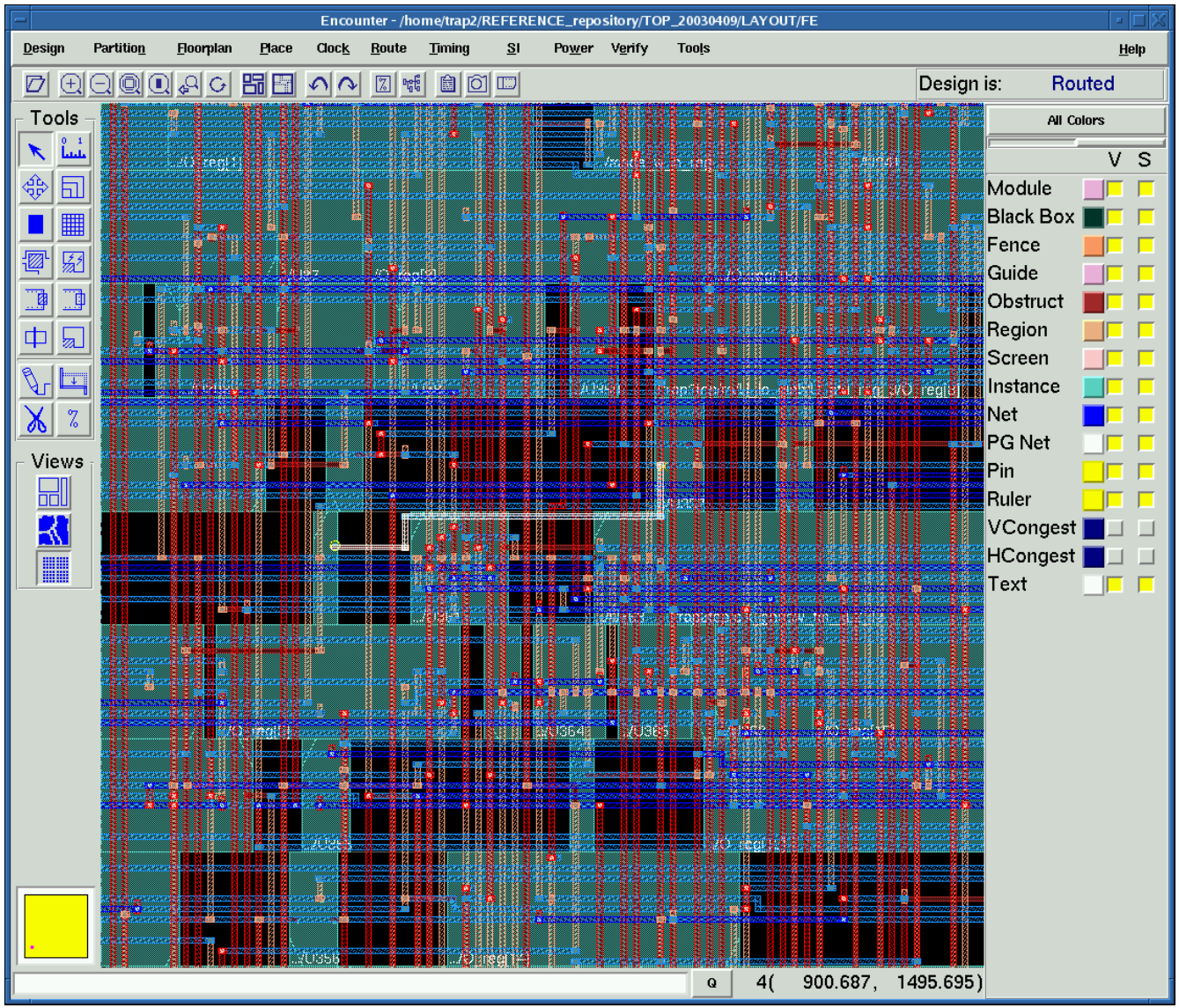
Übersicht (6/6)



Placement, Routing

- Die erzeugte Netzliste wird verwendet, um die Elemente der Schaltung zu platzieren und zu verdrahten.
- 1. Schritt:
  Geometrisches Platzieren vorgefertigter Standardzellen.
- 2. Schritt:
  Verdrahtung der Zellen, d.h. Herstellen leitender Verbindungen zwischen Standardzellen gemäß der Netzliste.
- Danach:
  Erzeugen von Maskendaten, verwendet zum Belichten der Chipherstellung.

Beispiel: Placement, Routing



UNI FREIBURG

Zusammenfassung

- Herausforderungen:
  - Chips mit Millionen/Milliarden von Transistoren in annehmbarer Zeit entwerfen,
  - gleichzeitig Entwurfskorrektheit garantieren,
  - neue Problemstellungen durch immer kleinere Strukturgrößen („nano scale”-Entwürfe),
  - ganze Systeme auf einem Chip.
- Designaufgabe bleibt beherrschbar bei
  - hierarchischem Vorgehen und Wiederverwenden vorhandener Entwürfe (Design Re-Use),
  - Entwurfsautomatisierung,
  - ausreichendem Aufwand für Entwurfskorrektheit.