

1. Einführung: Was ist Informatik?

- **Brockhaus Enzyklopädie in 24 Bänden (1997):**
 - Die Wissenschaft von der *systematischen* Verarbeitung von Informationen, besonders der automatischen Verarbeitung mithilfe von Computern.
- **Informatik (engl.: *computer science*)**
 - Kunstwort aus Information und Mathematik
 - Begriff ist in den 60er Jahren entstanden
 - ist nicht Programmieren
 - beschäftigt sich *nicht nur* mit den Digitalrechnern, wie wir sie heute kennen, sondern auch z.B. mit
 - DNA-Computing
 - Quantencomputer

Teilgebiete der Informatik

- **Technische Informatik**
- **Praktische Informatik**
- **Theoretische Informatik**
- **Angewandte Informatik**

Technische Informatik

- **Bereitstellung der Hardware**
 - Konstruktion von Rechnern, Speicherchips, Parallelrechnern, Peripheriegeräten, ...
 - Grenzen zur Elektrotechnik fließend
- **Berücksichtigung der Anforderungen der Programme**
 - z.B.
 - Verarbeitung von extrem vielen Daten in möglichst kurzer Zeit
 - ganz andere Hardwarelösung als zur Realisierung von z.B.
 - Steuerung von Waschmaschinen
 - Handys
 - Erzeugung von Grafiken
 - etc.

Praktische Informatik

- **Bereitstellung der Software, die auf den Rechnern läuft**
 - Brücke zwischen primitiven Operationen, die eine Maschine „versteht“ und den Anwendungsprogrammen, z.B.
 - Betriebssysteme
 - Netzwerke
 - Grafiksysteme
 - Mustererkennung
 - etc.
 - klassisches Gebiet: Compilerbau
 - Compiler übersetzen Programme einer höheren Programmiersprache in die Maschinensprache
 - dadurch kann man Programme in einer höheren Programmiersprache formulieren, die weitgehend unabhängig ist von der verwendeten Hardware, die nur eine spezielle Maschinensprache versteht

Theoretische Informatik

- **Theoretische Grundlagen der Informatik**
 - theoretische Erkenntnisse können sehr schnell in die Praxis umgesetzt werden
 - es entstehen Werkzeuge (engl.: *tools*), die an die Theorie angelehnt sind
 - Beispiel Compilerbau
 - anfangs (ohne Theorie der formalen Sprachen und Automatentheorie) benötigte man ca. 25 Personen-Jahre (Anzahl Personen mal Anzahl Jahre) für die Entwicklung eines einfachen Compilers
 - heute schafft das dank der entstandenen Werkzeuge ein Studierender allein im Praktikum
 - Beispiel Softwareentwicklung
 - gut durchdachte, theoretisch abgesicherte Entwürfe erweisen sich auch für hochkomplexe Software als leichter wartbar und erweiterbar
 - Software, die im Hauruck-Verfahren entstanden ist, stößt stattdessen schnell an ihre Grenzen

Theoretische Informatik (2)

- **Beispiel FORTRAN**
 - eine der ersten höheren Programmiersprachen
 - wurde nicht nach den Erkenntnissen der theoretischen Informatik entworfen
 - viele Regeln, viele Ausnahmen
 - Compilerbau erschwert
 - Programmentwicklung erschwert
- **besser PASCAL**
 - sauber strukturierte Sprache
 - war ursprünglich nur für die Lehre vorgesehen
 - wurde plötzlich überall eingesetzt
 - bis heute als Delphi: objektorientierte Erweiterung von Pascal
- **heute eher**
 - C++, Java zur Entwicklung von Software
 - VHDL, Verilog, SystemVerilog zur Entwicklung von Hardware
 - SystemC zur Entwicklung von Hardware und Software

Angewandte Informatik

- **Einsatz von Rechnern in den verschiedenen Anwendungsbereichen**
 - spezialisierte Programme für verschiedenste Anwendungen
 - Konzepte entwickeln, die in verschiedenen Bereichen zur Anwendung kommen
 - Textverarbeitung
 - Tabellenkalkulation
 - Simulation von Wirtschaftsprozessen
 - Computerspiele
 - etc.

Beispiel Pen-Point-Computer

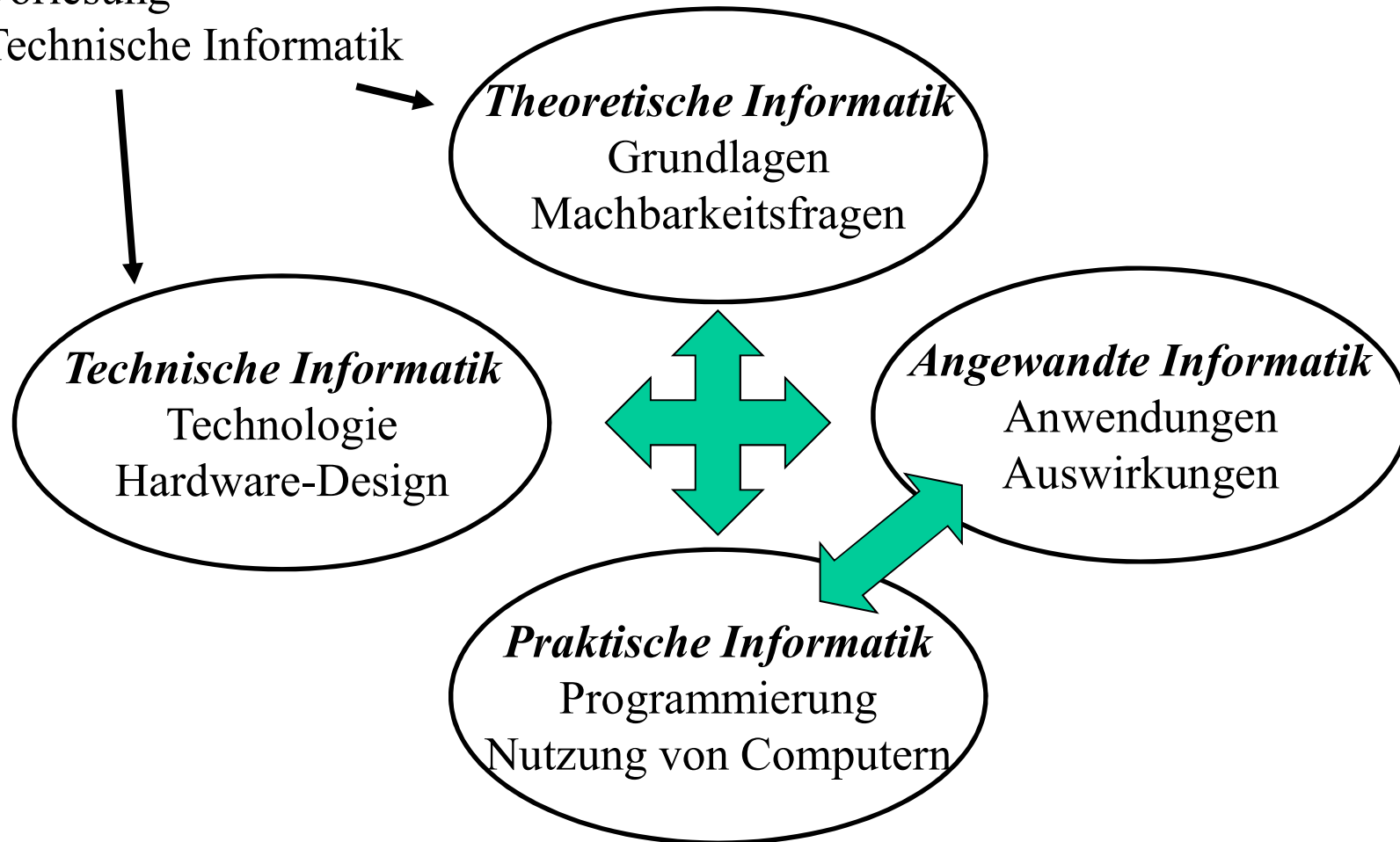
- **Touch-Screen plus Stift**
 - **Technische Informatik:**
 - Hardware bereitstellen
 - Designmethodik, Partitionierung (HW oder SW?)
 - **Praktische Informatik:**
 - Softwaregrundlagen, Betriebssystem
 - z.B. Handschrifterkennung
 - **Theoretische Informatik:**
 - theoretische Grundlagen
 - z.B. statistische Lerntheorie für Handschrifterkennung
 - **Angewandte Informatik:**
 - Einsatzmöglichkeiten entwickeln
 - z.B. Lagerhaltung, Terminkalender, etc.



Beziehungen der Teilgebiete

Vorlesung

Technische Informatik



Teilgebiete der Informatik (2)

- **Übergänge fließend**
 - Einteilung in diese vier Gebiete ist im deutschsprachigen Raum üblich (aber nicht unumstritten)
- **Weitere anwendungsbezogene Spezialgebiete ("Bindestrichinformatiken")**
 - Medizinische Informatik
 - Bio-Informatik
 - Medien-Informatik
 - Wirtschaftsinformatik
 - Linguistische Informatik, etc.

Ziele des Studiums der Informatik

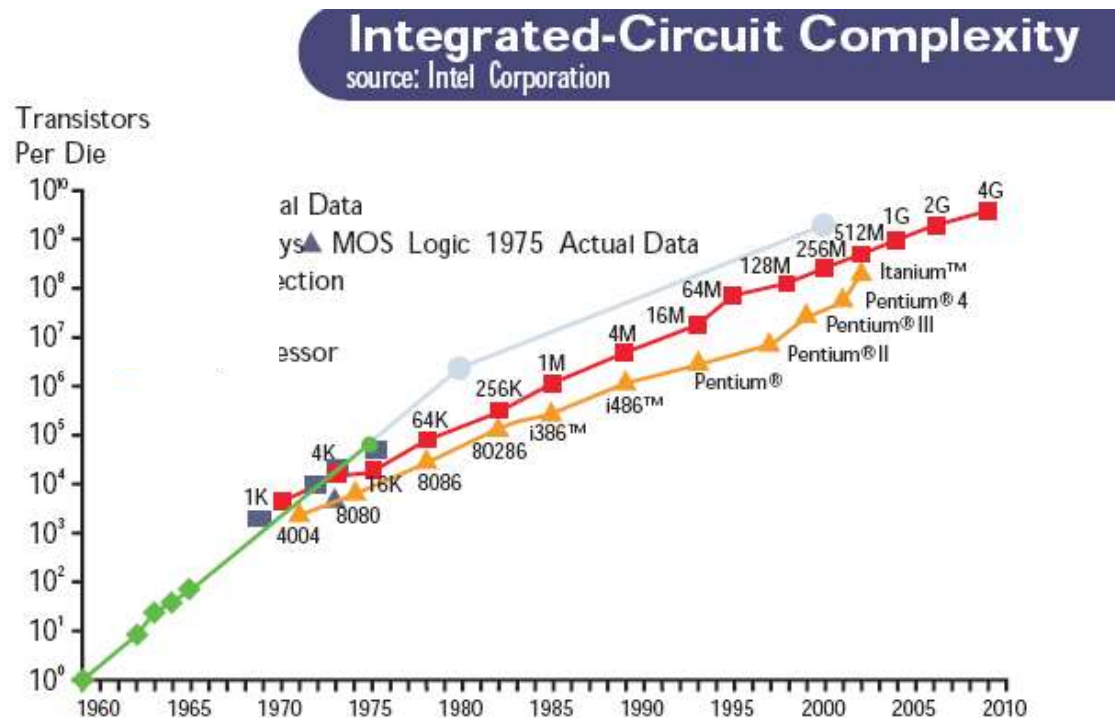
- **Informatik ist nicht identisch mit Programmieren!!!**
 - Ein Informatiker soll fähig sein
 - fachliche Zusammenhänge zu überblicken
 - Methoden der Informatik in der Berufspraxis einzusetzen
 - dem schnellen Wandel in der Informatik zu folgen
 - neue Entwicklungen kritisch zu bewerten
 - nach wissenschaftlichen Methoden selbständig zu arbeiten
- **Aber ohne Programmieren geht es auch nicht!!!!**
 - Implementieren und Testen von neuen Algorithmen
 - Modellieren von Softwaresystemen
 - selbst Hardware wird heute mit Programmiersprachen beschrieben, simuliert und synthetisiert (also der Schaltplan aus dem Programm generiert)

Historie: Rechnertechnologien

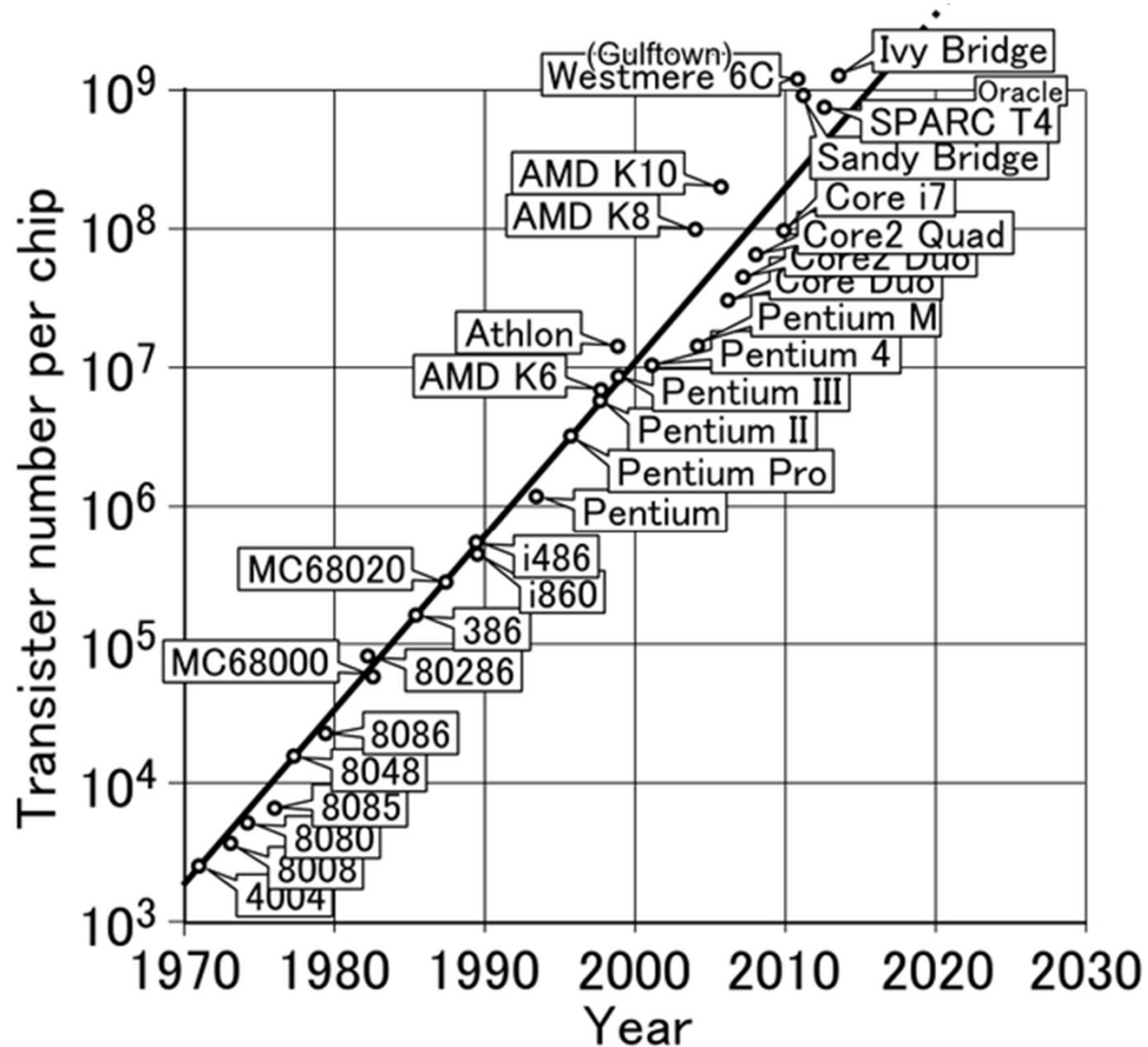
- **1. Generation: 1940 – 1954**
 - Relais: Schaltzeit 10^{-1} s
 - Vakuumröhre: Schaltzeit 10^{-4} s
- **2. Generation: 1955 – 1964**
 - Transistor: Schaltzeit 10^{-6} s
- **3. und 4. Generation: ab 1965**
 - Integrierter Schaltkreis: Schaltzeit $<10^{-9}$ s

Moore's Law

- **Gordon Moore (Mitbegründer von Intel)**
 - machte 1965 berühmte Beobachtung
 - Alle 12 bis 24 Monate verdoppelt sich die Anzahl der Transistoren pro integrierter Schaltung
 - Trend existiert bis heute



Moore's Law (2)



Die gute Nachricht

- **selbst die komplexesten Computerarchitekturen basieren auf ganz einfachen Prinzipien**
- **und die versuchen wir in dieser Vorlesung zu verstehen**