

Rechnerstrukturen

Vorlesung im Sommersemester 2009

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



Kapitel 1: Grundlagen

1.2 Entwurf von Rechenanlagen - Entwurfsfragen

- Entwurf einer Rechenanlage
 - Ingenieurmäßige Aufgabe der Kompromissfindung zwischen
 - Zielsetzungen
 - Einsatzgebiet, Anwendungsbereich, Leistung, Verfügbarkeit ...
 - Randbedingungen
 - Technologie, Größe, Geld, Energieverbrauch, Umwelt, ...
 - Gestaltungsgrundsätzen
 - Modularität, Sparsamkeit, Fehlertoleranz ...
 - Anforderungen
 - Kompatibilität, Betriebssystemanforderungen, Standards

- Zielsetzungen

- Anwendungsbereiche:

- Technisch-wissenschaftlichen Bereich

- Strömungsmechanik
 - Materialforschung
 - ...

- Kommerzieller Bereich

- Datenbankanwendungen
 - WEB, Suchmaschinen
 - Optimierung von Geschäftsprozessen,
 - Unterstützung von Geschäftsentscheidungen (Risikoanalyse)
 - ...

- Eingebettete Systeme

- Verarbeitung digitaler Medien
 - Automatisierungstechnik
 - Automobil
 - Telekommunikation
 - ...

- Zielsetzungen

- Benutzerfreundlichkeit

- Beziehung zwischen einem Rechensystem und Nutzer:
 - Anwendungsprogrammierer:
 - » problemorientierte Erstellung von Anwendungen
 - Administration
 - » Verwaltung, Wartung verteilter Systemressourcen
 - Bedienung durch nicht-technisches Personal, Anwender
 - Gestaltung der Schnittstelle zwischen dem Rechensystem und seinem Benutzer
 - Ziel der Software-Entwicklung
 - Techniken der benutzerfreundlichen Oberfläche erst durch spezielle Hardware-Techniken möglich

- Zielsetzungen
 - **Verlässlichkeit / Robustheit**
 - Gewährleistung einer minimalen Verfügbarkeit des Systems
 - Bei Ausfällen von Komponenten muss ein betriebsfähiger Kern bereit sein
 - » Vielfach Verwendung redundanter Komponenten
 - Wichtig für sicherheitskritische Anwendungen

- Zielsetzungen
 - Erweiterbarkeit / Skalierbarkeit
 - Charakterisiert ein System, das in jedem Zustand die Erweiterung seiner Fähigkeiten zulässt, aber in jedem Zwischenzustand korrekt funktionsfähig ist
 - Installation weiterer Ausbaustufen durch Hinzufügen von Komponenten

- **Gestaltungsgrundsätze (nach Ungerer, 2000)**
 - **Konsistenz:**
 - Eigenschaft eines Systems mit folgerichtiger, schlüssiger Aufbau
 - Bei Kenntnis eines Teil des Systems muss der Rest vorhersagbar sein
 - Beispiel:
 - Bei Erweiterung des Befehlssatzes ist die Realisierung eines neuen Befehls weitgehend durch die bereits vorliegenden Befehle festgelegt
 - **Orthogonalität / Modularität**
 - Funktional unabhängige Teilelemente sind unabhängig voneinander spezifiziert und realisiert.
 - Beispiel:
 - Befehlssatz ist orthogonal, wenn sich jeder arithmetisch-logische Befehl mit jeder Adressierungsart kombinieren lässt

- **Gestaltungsgrundsätze**

- **Symmetrie**

- Eigenschaft eines Systems, mathematisch symmetrische Eigenschaften des Systems auch symmetrisch zu entwerfen
- Beispiel:
 - Die Verwendung der Subtraktion sollte in gleichartiger Weise erfolgen können, wie die Addition, d.h. mit denselben Daten- und Befehlsformaten

- **Angemessenheit**

- Die Elemente eines Systems sind angemessen, wenn ihre Funktionen bei der Lösung der vorgesehenen Problemstellung ausgeschöpft werden
- Beispiel:
 - Abstimmung von Prozessor und Speicher

• Gestaltungsgrundsätze

– Sparsamkeit

- Kosten des Systems möglichst gering halten. Sie ist in Abhängigkeit von der Technologie zu sehen
- Beispiel:
 - Kosten, Ausbeute bei der Chip-Herstellung

– Wiederverwendbarkeit

- Einsatz von Komponenten, die von vorne herein auf allgemeine Verwendbarkeit hin entwickelt wurden
- Beispiel:
 - Standardisierung von Bussen (PCI-Bus)
 - Einsatz von Standardkomponenten bei der Entwicklung von Höchstleistungsrechnern

- **Gestaltungsgrundsätze**

- **Transparenz**

- Ein System ist transparent, wenn verschiedene Funktionen des Gesamtsystems unsichtbar bleiben
 - Beispiel:
 - Prozesse im Hintergrund

- **Virtualität**

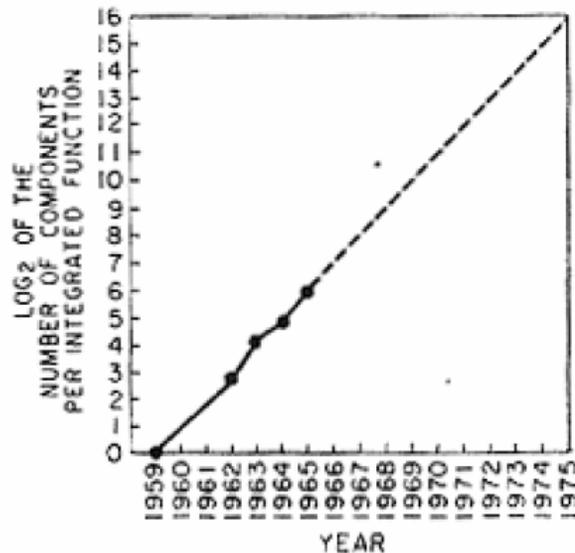
- Eigenschaft eines Systems, Funktionen anzubieten, die real gar nicht vorhanden sind
 - Beispiel:
 - Virtuelle Speicherverwaltung

- **Randbedingungen**
 - Technologische Entwicklungen

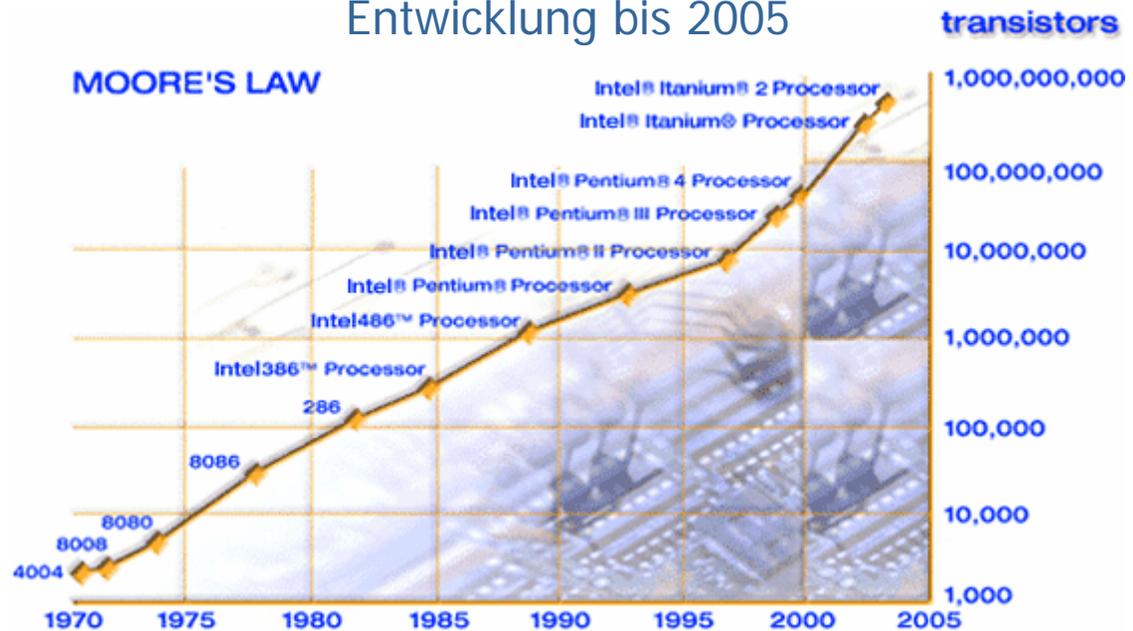
- **Moore's Law**

- Anzahl der Transistoren, die auf einem Chip integriert werden können, verdoppelt sich etwa alle 2 Jahre

Vorhersage 1965



Entwicklung bis 2005

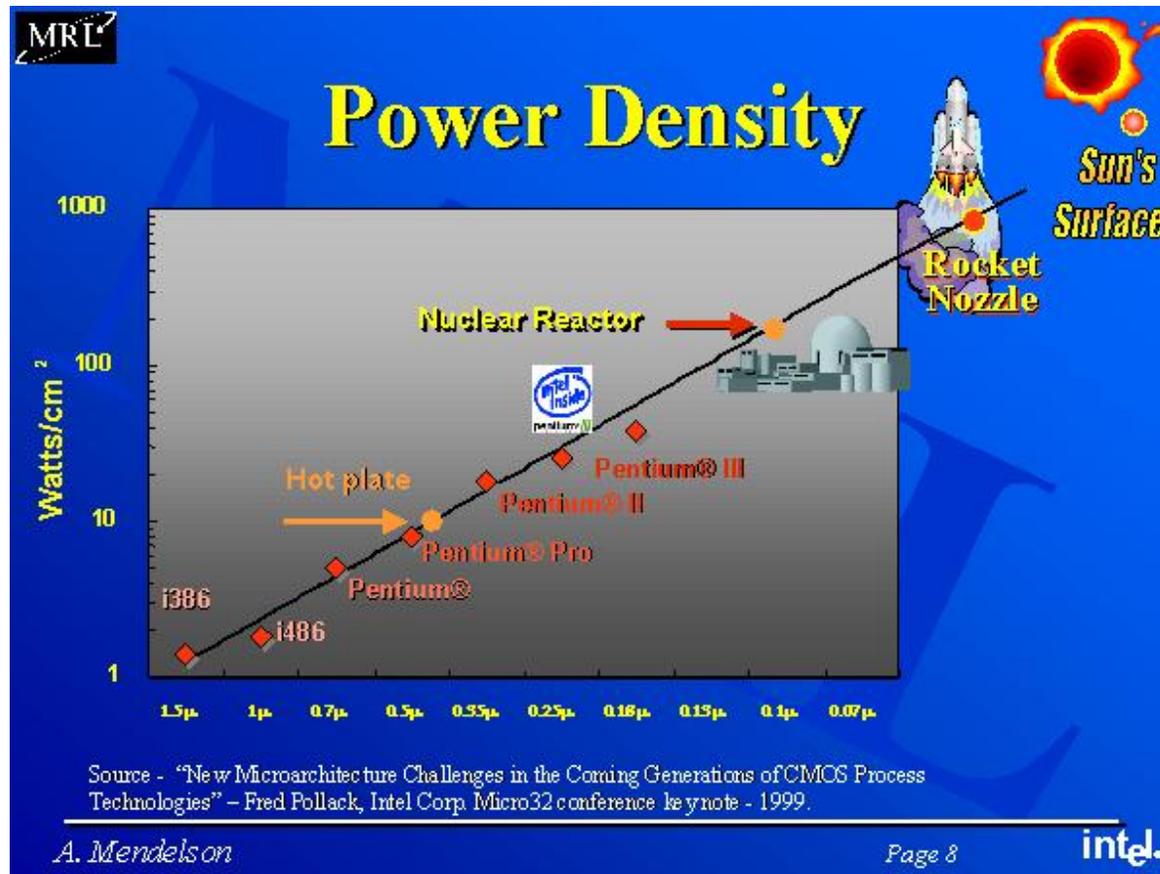


Gordon E. Moore: Craming More Components Onto Integrated Circuits. Electronics, Vol. 38, No.8, April 19, 1965

<http://www.intel.com/research/silicon/mooreslaw.htm>

- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Entwicklung der Halbleitertechnologie (Henn./Patt. ,03)**
 - Integrationsdichte
 - » Verbesserung um etwa 35% pro Jahr
 - Chipfläche
 - » Erhöhung um etwa 10% – 20% pro Jahr (schwieriger vorherzusagen)
 - Anzahl der Transistoren auf einem Chip
 - » Steigerung um etwa 55% pro Jahr
 - Vorhersagen durch die Semiconductor Industry Association (SIA), (<http://www.sia-online.org>)
 - » International Technology Roadmap for Semiconductors 2003 Edition (<http://public.itrs.net>)
 - » 2004 Update
 - **Folgerung:**
 - mehrere Milliarden Transistoren auf einem Chip in wenigen Jahren

- **Randbedingungen**
 - Technologische Entwicklungen

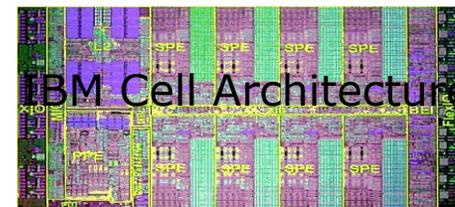


Elektrische Leistung wird in Wärme umgewandelt.

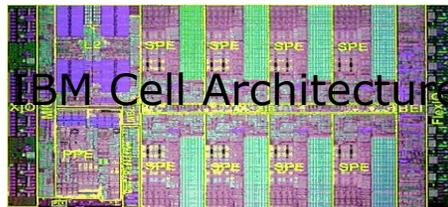
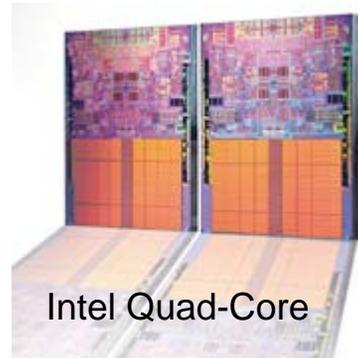
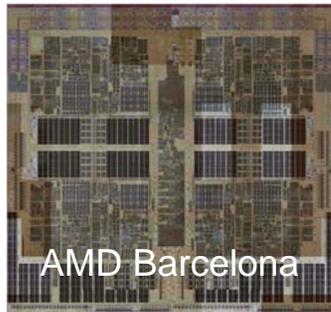
Problem:
Prozessortemperatur begrenzt die Verarbeitungsgeschwindigkeit von Prozessoren!

(Genauere Erläuterung des Zusammenhangs später!)

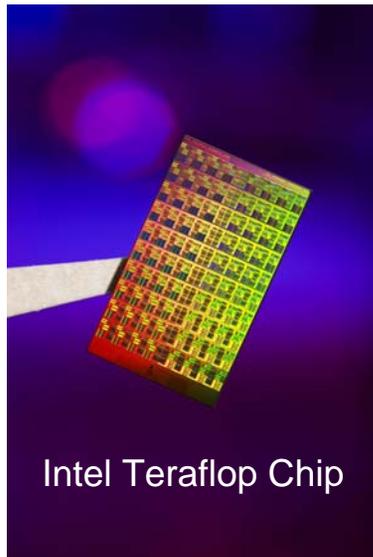
- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Leistungsdichte: Folgerungen**
 - Leistungssteigerung durch Parallelverarbeitung, auch auf Chip-Ebene:
 - » Ausschöpfen der Leistung durch höhere Integration der Chip-, Modul- und Systemebenen
 - » Sinnvolles Zusammenwirken von HW/SW
 - **Chip-Multiprozessoren (CMPs):**
 - » Integration mehrerer Prozessorkerne auf einem Chip:



- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Chip-Multiprozessoren (CMPs), Multi-core:**
 - Stand heute: 4-8 Prozessorkerne auf einem Chip



- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Chip-Multiprozessoren (CMPs), Multicore:**
 - Zukünftige Entwicklung: mehrere hundert Prozessorkerne auf einem Chip



Intel Teraflop Chip

- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Entwicklung der DRAM-Technologie (Henn./Patt. ,03)**
 - **Integrationsdichte**
 - » Verbesserung um etwa 40% - 60% pro Jahr
 - **Zykluszeit**
 - » Nur langsame Verbesserung: hat in 10 Jahren um etwa 1/3 abgenommen
 - **Bandbreite**
 - » Steigerung doppelt so hoch wie bei der Latenz

- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Entwicklung der Magnetplattentechnologie (Henn./Patt. ,03)**
 - **Speicherdichte**
 - » **Verbesserung um etwa 100% pro Jahr**
 - » **Bis 1990 Steigerung um etwa 30% pro Jahr**
 - **Zugriffszeit**
 - » **Verbesserung um etwa 1/3 in 10 Jahren**

- **Randbedingungen**

- **Technologische Entwicklungen**

- **Entwicklung der Netzwerktechnologie (Henn./Patt. ,03)**

- Leistung des Netzwerks hängt von der Leistung der Switches und der Leistung des Übertragungssystems ab

- **Beobachtungen:**

- Die Anzahl der logischen Netzwerkverbindungen wächst mit der Anzahl N der Internetknoten um $N \times (N-1)/2$ (Metcalfe's Law)
 - Steigerung der Bandbreite um den Faktor 1000 in 15 Jahren von 1MB Ethernet – 10GBit Ethernet

- **Folgerung:**

- Leistungsfähigkeit des Netzwerks wächst schneller als die Rechenleistung

- **Randbedingungen**
 - **Technologische Entwicklungen**
 - **Folgerungen**
 - Drastisch steigende Komplexität der Systeme
 - Verteilte Systemressourcen
 - Selbstorganisierende Systeme
 - » Selbst-konfigurierend
 - » Selbst-optimierend
 - » Selbst-heilend
 - » Selbst-schützend
 - » ...
 - Beispiel: Forschungsprojekt DodOrg an Uni KA

Kapitel 1: Grundlagen

1.3 Entwurf von Rechenanlagen – Energiespartechniken

- Motivation

- Mobile Geräte

- verfügbare Energiemenge durch Batterien und Akkumulatoren begrenzt
 - möglichst lange mit vorhandener Energie auskommen
 - möglichst wenig Energie soll in Wärme umgesetzt werden, um eine Überhitzung zu vermeiden

- Green IT

- Rechnerhersteller bieten „green HW“ an:
 - niedriger Energieverbrauch
 - ökologische Produktion
 - einfaches Recycling

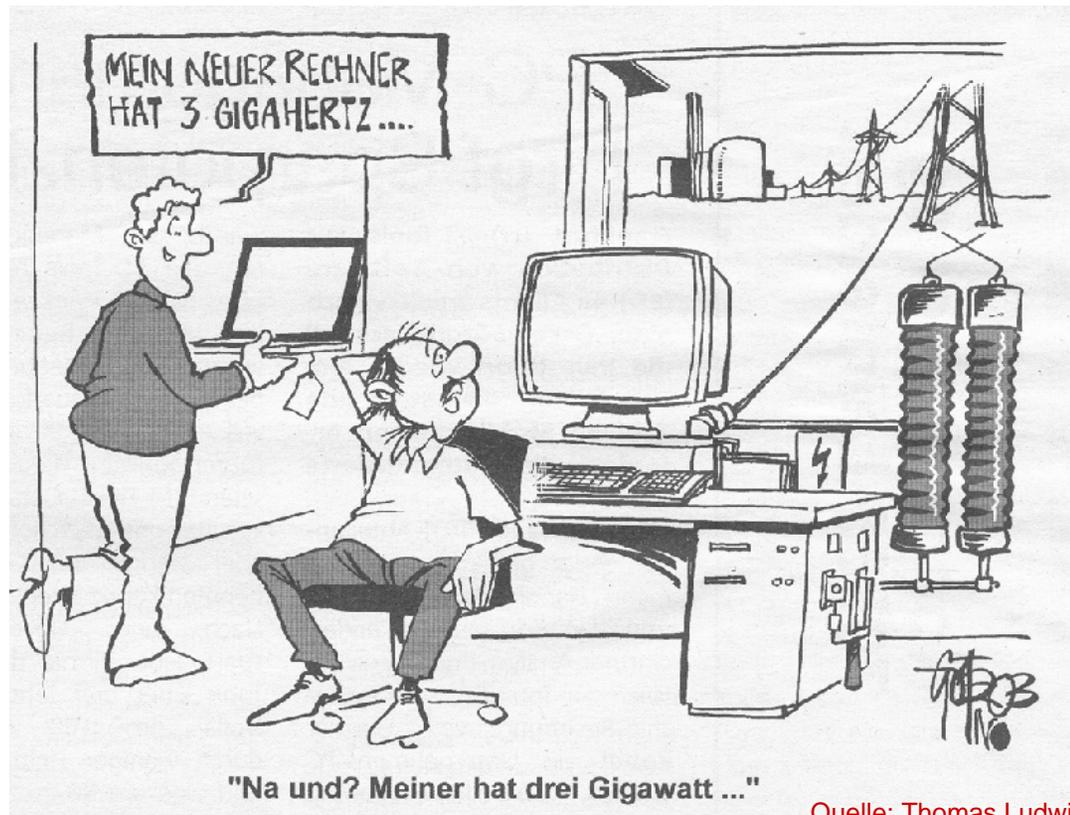
• Motivation

– Einige Fakten zum Energieverbrauch

- Beobachtungen seit 1992:
 - Steigerung der Rechenleistung: Faktor 10000
 - Steigerung der Rechenleistung/Watt: Faktor 300
- Leistungsverbrauch von Rechenzentren
 - im Bereich zwischen 0,5 MW und 15 MW, oder höher
- Laufender Betrieb mit 1 MW bedeutet:
 - $\sim 8.700.00$ kWh/Jahr
- mit Energiekosten von 0,20 €/kWh
 - $\approx 1.750.000$ €/y
 - $\approx 5.,000$ €/d
 - ≈ 200 €/h
- CO₂-äquivalent mit 500 g/kWh
 - 4,350,000 kg CO₂
 - approx. 3,000 2-Personenhaushalte oder
 - approx. 1,000 Automobile mit $\sim 15,000$ km/Jahr

Quelle: Thomas Ludwig, Intel HPC Roundtable, Munich, 04/2008

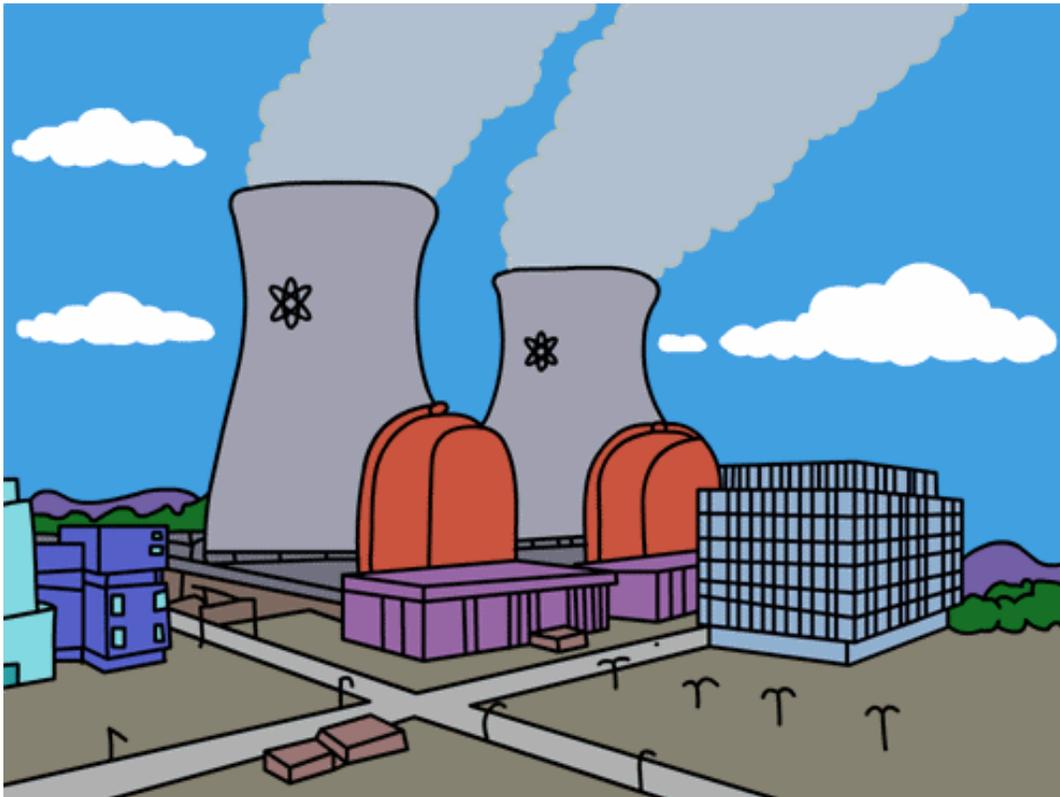
- Motivation
 - HPC-Bereich



Quelle: Thomas Ludwig, Intel HPC Roundtable, Munich, 04/2008

- Motivation

- HPC-Bereich: SCC in Karlsruhe 2015 ???



- **Motivation**

- HPC-Bereich: Rangliste unter www.green500.org

- Maß: MFlops/W

- Energie-effizientester Supercomputer:

- IBM Blue Gene/P ~ 350 MFlops/W

- Energie-effizienter Dinosaurier:

- Earth Simulator ~ 5 MFlops/W

- hoher Energiebedarf für die Kühlung:

- zusätzlich 50% - 70% der Leistung

• Definitionen

- Leistung bezeichnet den Energiefluss pro Zeit
- Zusammenhang zwischen Energie E , Leistung P und Zeit t :

$$P = E/t \text{ bzw. } E = P * t$$

- Auf elektrische Geräte übertragen:
 - Leistung bezeichnet die aufgenommene bzw. verbrauchte Energie pro Zeit
- Verbale Unterscheidung der Rechenleistung von der elektrischen Leistung:
 - Leistungsaufnahme oder Verlustleistung

- Ziele beim Entwurf

- Verringerung des Energieverbrauches

- Erhöhung der Betriebszeit eines batteriebetriebenen Gerätes

- Reduktion der Temperatur

- Reduktion der Leistungsaufnahme (Verlustleistung)

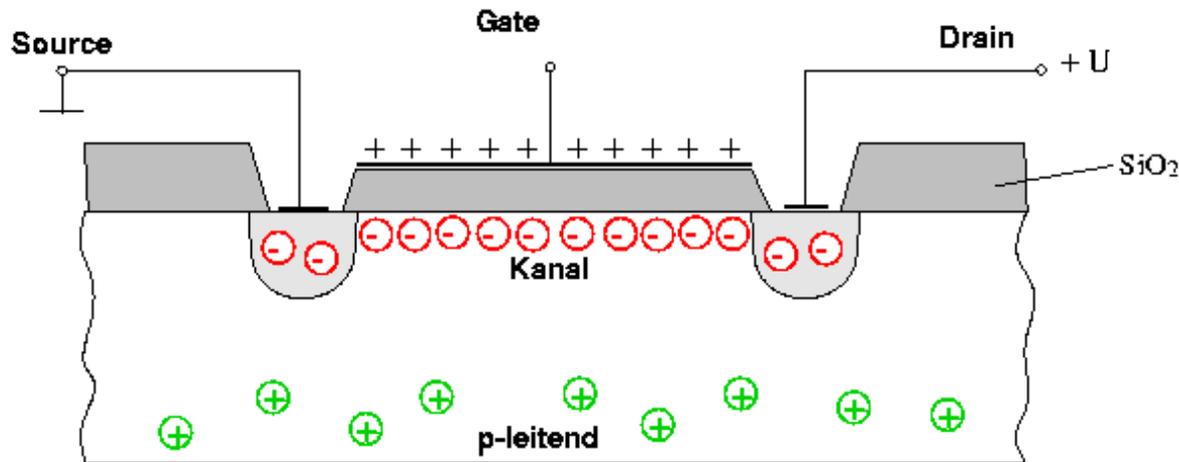
- wichtig bei Hochleistungsmikroprozessoren

- Prozessortemperatur begrenzt möglicherweise die Verarbeitungsgeschwindigkeit
- als Vergleichsmaß wird die auf die Leistungsaufnahme normierte Verarbeitungsgeschwindigkeit verwendet (MIPS/Watt)

• Grundlagen

– Selbstsperrende nMOS-Transistoren (siehe TI I)

Eine Spannung U_{GS} zwischen Gate und Source hat folgende Wirkung:

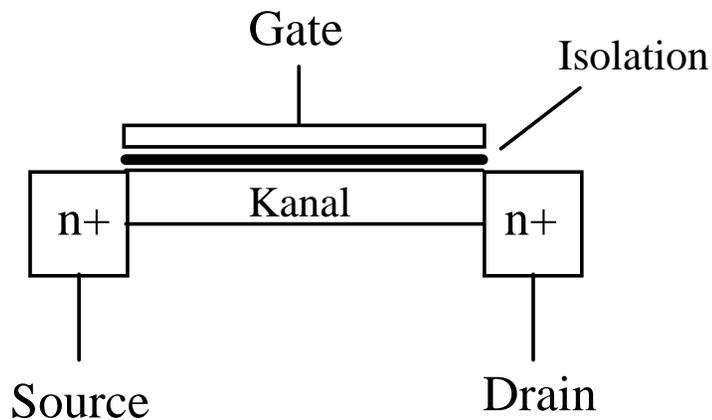


$$U_{GS} = +U$$

Positive Ladungsträger auf der Gate-Elektrode, die negative Ladungsträger unter der Isolationsschicht induzieren.

• Grundlagen

– Selbstsperrende nMOS-Transistoren (siehe TI I)



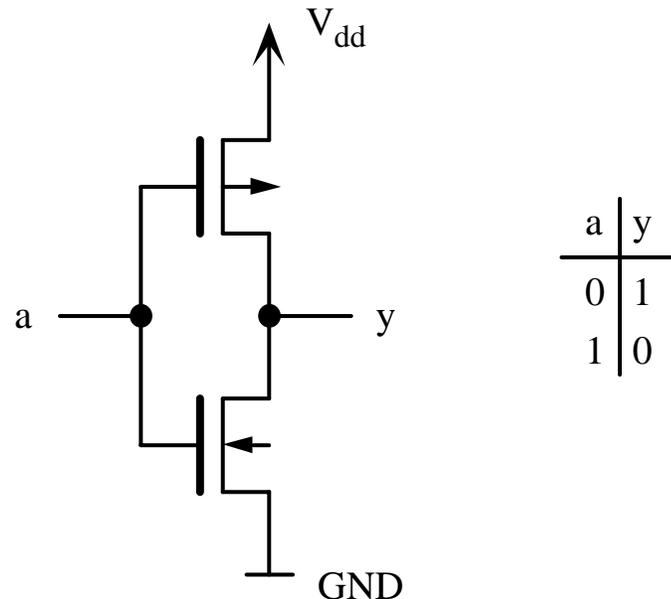
- der MOS Transistor arbeitet zur Steuerung der Strecke zwischen Source und Drain allein mit elektrischen Feldern (praktisch kein Stromfluss am Gate)
- je nach Spannung am Gate und dem daraus resultierenden Feld im Kanal können Ladungsträger den Kanal passieren oder nicht.

➔ **MOS Transistor als Schalter**

pMOS entgegengesetzte Dotierungen

• Grundlagen

– CMOS-Schaltung: Beispiel Inverter



Funktionsweise:

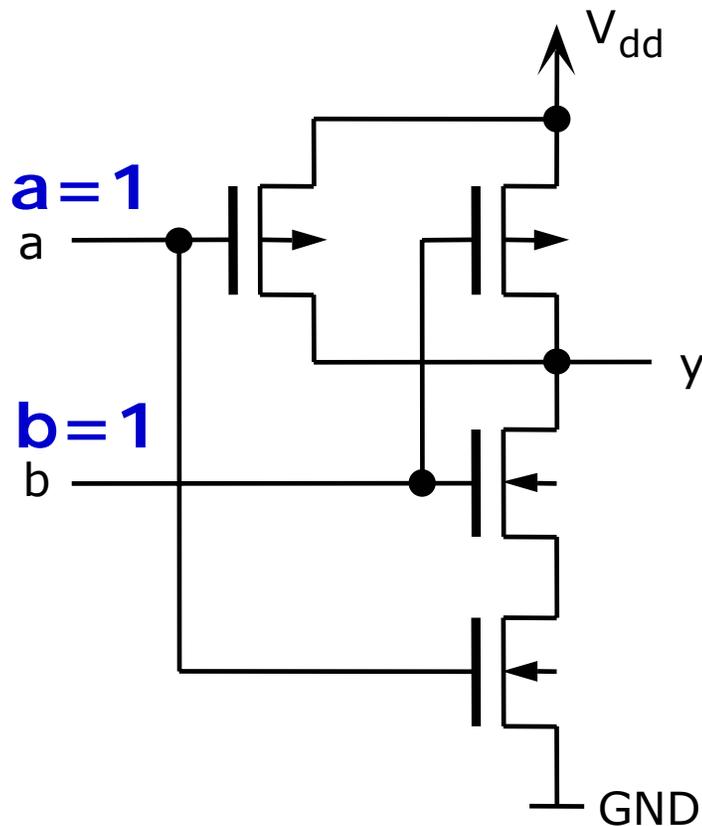
- Ist $a = 0$, so wird der nMOS Transistor gesperrt, der pMOS Transistor leitet
 → am Ausgang liegt $V_{dd} = 1$
- Ist $a = 1$, so leitet der nMOS Transistor, der pMOS Transistor ist gesperrt
 → am Ausgang liegt $GND = 0$

Weder für $a=1$ noch für $a=0$ existiert ein leitender Pfad von V_{dd} zu GND \Rightarrow kein Stromverbrauch bei konstanten Eingangsvariablen.

Stromverbrauch nur beim Übergang.

• Grundlagen

• CMOS-Schaltung: NAND



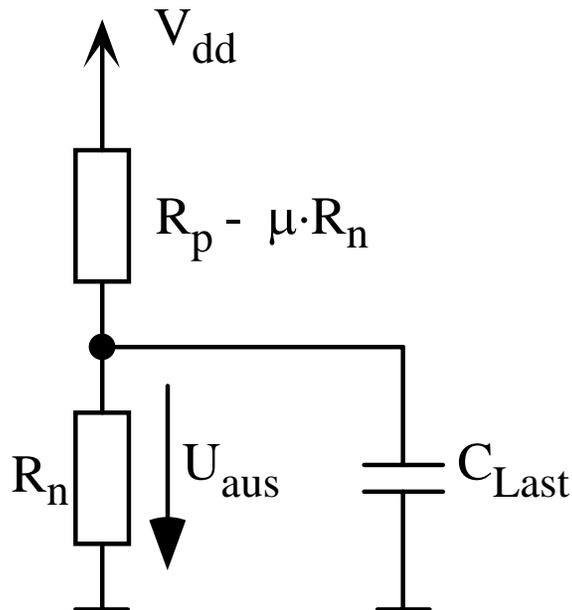
a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

- Nur wenn $a = 1$ und $b = 1$ sind beide nMOS Transistoren leitend
- ➔ am Ausgang liegt $\text{GND} = 0$

• Grundlagen

– Transistor als Schalter

- Ersatzschaltbild: realitätsnäheres Bild mit Widerständen und Kapazitäten



R_p : Widerstand des p-Netzes leitet das p-Netz, so ist R_p klein, ansonsten groß

R_n : Widerstand des n-Netzes leitet das n-Netz, so ist R_n klein, ansonsten groß

C_{Last} : Lastkapazität der am Ausgang angeschlossenen Leitungen und weiteren Schaltungen

• Grundlagen

– Leistungsaufnahme bei CMOS-Schaltungen

- $P_{\text{total}} = P_{\text{switching}} + P_{\text{shortcircuit}} + P_{\text{static}} + P_{\text{leakage}}$

- Leistungsverbrauch bei Zustandsänderung

- $P_{\text{switching}}$: Laden oder Schalten einer kapazitiven Last
- $P_{\text{shortcircuit}}$: Leistungsverbrauch während des Übergangs am Ausgang in einem CMOS Gatter, wenn sich die Eingänge ändern

- Statischer Leistungsverbrauch (unabhängig von Zustandsänderungen)

- P_{static} : Statischer Leistungsverbrauch
- P_{leakage} : Leistungsverbrauch durch Kriechströme

• Grundlagen

– Leistungsaufnahme bei CMOS-Schaltungen

- $P_{\text{switching}}$: Wesentlicher Anteil am Leistungsverbrauch

- Entsteht beim Laden und Entladen der Lastkapazitäten C
- Beim Übergang von $0 \rightarrow 1$ ziehen die Kapazitäten einen Spannungswechsel V_{swing} nach sich
- Der Leistungsaufnahme $P_{\text{switching}}$ errechnet sich zu:

- $P_{\text{switching}} = C * V_{\text{dd}} * V_{\text{swing}} * a * f$, mit

- a : Aktivitätsgewichtsfaktor; Faktor der Frequenz, mit der die Übergänge erfolgen dürfen ($0 < a < 1$)
- f : Frequenz
- V_{dd} : Versorgungsspannung

- Grundlagen

- Leistungsaufnahme bei CMOS-Schaltungen

- $P_{\text{switching}}$: Wesentlicher Anteil am Leistungsverbrauch

- Vereinfacht:

- $P_{\text{switching}} = C_{\text{eff}} * V_{\text{dd}}^2 * f$, mit

- C_{eff} : effektive Kapazität: $C * a$

- $V_{\text{dd}} = V_{\text{swing}}$

- Grundlagen

- Leistungsaufnahme bei CMOS-Schaltungen

- $P_{\text{shortcircuit}}$: Während des Wechsel des Eingangssignals tritt eine überlappte Leitfähigkeit der nMOS und pMOS-Transistoren auf, die einen CMOS-Transistorgatter ausmachen
 - Vereinfacht:
 - $P_{\text{switching}} = I_{\text{mean}} * V_{\text{dd}}$, mit
 - I_{mean} : mittlerer Strom während des Wechsels des Eingangssignals
 - Kann für ein Gatter mit kurzen Eingangsflanken minimiert werden, auf Kosten von langen Übergangszeiten am Ausgang
 - Für eine Menge von Gatter wird versucht, gleiche Zeiten für steigende und fallende Flanken am Eingang und am Ausgang zu erhalten

• Grundlagen

– Leistungsaufnahme bei CMOS-Schaltungen

• P_{leakage} :

- Bei realen CMOS-Schaltungen kommt zu dem Stromfluss beim Wechsel des logischen Pegels ein weiterer ständiger Stromfluss hinzu: Leckströme
- Leckströme entstehen, da die Widerstände zwischen den Leiterbahnen der integrierten Schaltkreise nicht unendlich hoch sind.
- Leckströme wachsen mit zunehmender Integrationsdichte
- bei Integrationsdichten mit Strukturen kleiner als 100 nm kann die Leistungsaufnahme aufgrund von Leckströmen nicht mehr vernachlässigt werden.

• Grundlagen

– Leistungsaufnahme bei CMOS-Schaltungen

• Zusammenfassung:

- unter idealen Voraussetzungen ist $P \sim f$, d.h. Reduktion der Taktfrequenz bedeutet Reduktion der Leistungsaufnahme, aber eine Verlangsamung der Ausführungsgeschwindigkeit
- unter idealen Voraussetzungen ist $P \sim V_{dd}^2$, d.h. eine Reduktion der Versorgungsspannung um beispielsweise 70% bedeutet eine Halbierung der Leistungsaufnahme. Bei Beibehaltung der Taktfrequenz keine Verlangsamung der Ausführungsgeschwindigkeit!
- Aber: Versorgungsspannung und Taktfrequenz sind keine voneinander unabhängige Größen: je geringer die Versorgungsspannung desto geringer die maximale Frequenz. Näherungsweise kann ein linearer Zusammenhang angenommen werden: $f \sim V_{dd}$
- Kubus-Regel: $P \sim V_{dd}^3$ oder $P \sim f^3$

- Grundlagen

- Energieverbrauch bei CMOS-Schaltungen

- unter idealen Voraussetzungen ist für eine konstante Zeit t_k der Energieverbrauch E proportional zur Taktfrequenz f :
 $E \sim f$
- unter idealen Voraussetzungen ist bezogen auf eine zu erfüllende Aufgabe (z.B. Durchführung einer Berechnung) die dafür benötigte Zeit t_a umgekehrt proportional zur Taktfrequenz. Damit wird der Energieverbrauch zur Erfüllung einer Aufgabe unabhängig von der Taktfrequenz.
- unter Berücksichtigung der statischen Leistungsaufnahme wächst der Energieverbrauch bezogen auf eine zu erfüllende Aufgabe mit abnehmender Taktfrequenz!
 - Dies wird verursacht durch den statischen Teil der Leistungsaufnahme, der umso längere Zeit anliegt, je länger die Ausführung der Aufgabe durch Verringerung der Taktfrequenz anliegt.

- **Energiespar-Techniken**

- Senkung der Leistungsaufnahme ohne Einbußen in der Verarbeitungsgeschwindigkeit und damit auch Senkung des Energiebedarfs für die Bearbeitung einer Aufgabe
 - **Optimierung der Systemarchitektur**
 - Sinnvolles Zusammenwirken aller Komponenten einer Systemarchitektur (HW, Betriebssystem, Kommunikationsschnittstelle, Middleware, Anwendung), um unnötigen Energieverbrauch zu erkennen und zu vermeiden
 - **Energieoptimierung für Desktop- und Serversysteme**
 - Einsatz von Multicore-CPU's
 - » Ausnützen der Parallelverarbeitung anstelle Erhöhung der Taktfrequenz
 - Einsatz energiesparender spezialisierter Prozessorkerne (Koprozessoren)
 - **Energiespartechniken auf den verschiedenen Ebenen des Entwurfs**