

# Rechnerstrukturen

Vorlesung im Sommersemester 2009

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



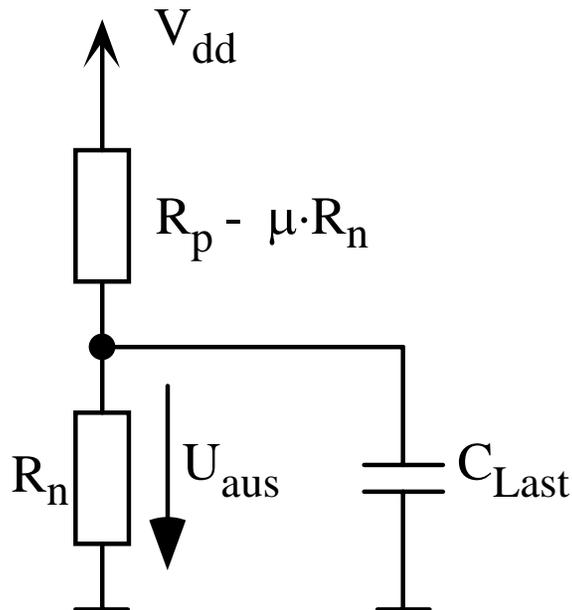
## Kapitel 1: Grundlagen

### 1.3 Entwurf von Rechenanlagen – Energiespartechniken

- 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)

- $P_{\text{static}}$ : Statischer Leistungsverbrauch

- $P_{\text{leakage}}$ : 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  $V_{\text{swing}}$  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
    - $V_{\text{dd}}$ : 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

- $C_{\text{eff}}$ : 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
  - $I_{\text{mean}}$ : 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**

- $P_{\text{leakage}}$ :

- 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**

## Kapitel 1: Grundlagen

### 1.4 Bewertung der Leistungsfähigkeit

- Ziele
  - Auswahl der Rechenanlage
  - Veränderung der Konfiguration einer bestehenden Anlage
  - Entwurf von Anlagen

- Was heißt es: Ein Rechner ist schneller als ein anderer Rechner?

- Der Benutzer eines Arbeitsplatzrechners:

- „Ein Rechner A ist schneller als ein Rechner B, wenn ein Programm auf A weniger Zeit benötigt.“
- Reduzierung der Antwortzeit (response time) oder Ausführungszeit (execution time)
  - Zeit zwischen dem Beginn und dem Ende eines Ereignisses, einer Aufgabe
- A ist  $n$ -mal schneller als B:

$$\frac{\text{Ausführungszeit(B)}}{\text{Ausführungszeit(A)}} = n$$

- Was heißt es: Ein Rechner ist schneller als ein anderer Rechner?
  - Der Rechenzentrumsleiter:
    - „Ein Rechner A ist schneller als ein Rechner B, wenn A in einer Stunde mehr Aufträge (Jobs) erledigt.“
    - Erhöhung des Durchsatzes (throughput)
      - Anzahl der ausgeführten Aufgaben in einem gegebenen Zeitintervall
    - Durchsatz von A ist  $m$ -mal höher als der von B:
      - Die Anzahl der erledigten Aufgaben auf A ist  $m$ -mal die Anzahl der erledigten Aufgaben auf B.

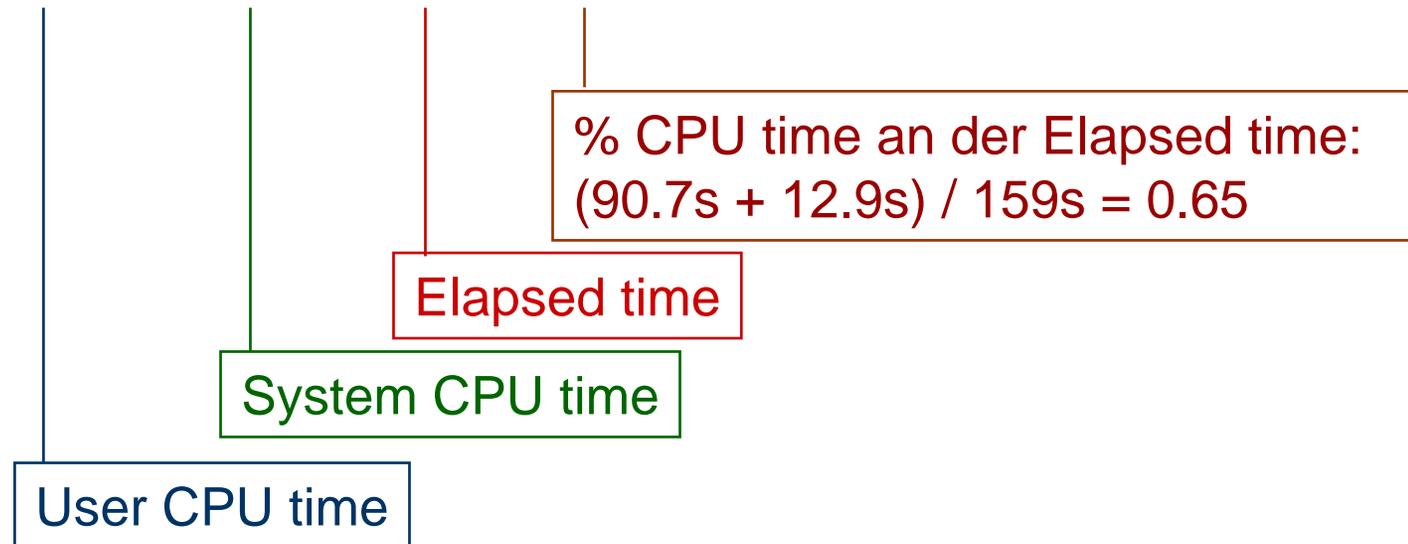
- **Ausführungszeit (execution time)**
  - Wall-clock time, response time, elapsed time
    - Latenzzeit für die Ausführung einer Aufgabe
      - Schließt den Speicher- und Plattenzugriff, Ein-/Ausgabe etc. mit ein.
  - CPU Time
    - Zeit, in der die CPU arbeitet
      - User CPU Time: Zeit, in der die CPU ein Programm ausführt
      - System CPU Time: Zeit, in der die CPU Betriebssystemaufgaben ausführt, die von einem Programm angefordert werden

- Ausführungszeit (execution time)

- CPU Time

- Beispiel: UNIX `time` Kommando:

90.7u, 12.9s, 2:39 65%



- Verfahren
  - Auswertung von Hardwaremaßen und Parametern
  - Laufzeitmessungen bestehender Programme
  - Messungen während des Betriebs von Anlagen
  - Modelltheoretische Verfahren

- Gleichung für die Leistung der CPU
  - Der Rechner läuft mit fester Taktrate, angegeben durch
    - Dauer eines Taktzyklus (z. B. 1ns)
    - Taktfrequenz (z. B. 1 GHz)
  - Die CPU-Zeit einer Programmausführung kann dargestellt werden mit

$$\begin{aligned} \text{CPU-Zeit} &= \text{Anzahl Taktzyklen für das Programm} * \text{Taktzyklusdauer} = \\ &= \frac{\text{Anzahl Taktzyklen für das Programm}}{\text{Taktfrequenz}} \end{aligned}$$

- Gleichung für die Leistung der CPU
  - Einführung der Maßzahl CPI (clock cycles per instruction)
    - Mittlere Anzahl der Taktzyklen pro Befehl
    - mit IC (instruction count), der Anzahl der ausgeführten Befehle eines Programms

$$\text{CPI} = \text{Anzahl Taktzyklen für das Programm} / \text{IC}$$

- Damit

$$\begin{aligned} \text{CPU-Zeit} &= \text{IC} \times \text{CPI} \times \text{Taktdauer} \\ &= \text{IC} \times \text{CPI} / \text{Taktfrequenz} \end{aligned}$$

- Gleichung für die Leistung der CPU
  - Konsequenz
    - CPU-Leistung ist bestimmt durch
      - Taktfrequenz
        - » Abhängig von der Halbleitertechnologie
      - CPI
        - » Abhängig von der Implementierung, ISA (Befehlssatzarchitektur), Programm
      - IC
        - » Abhängig von ISA, Compilertechnologie, Programm

- Maßzahlen für die Operationsgeschwindigkeit
  - Analog lässt sich für eine Programmausführung definieren:
    - MIPS: Millions of Instructions Per Second

$$\text{MIPS} = \frac{\text{Anzahl der ausgeführten Instruktionen}}{10^6 \times \text{Ausführungszeit}}$$

- MFLOPS: Millions of Floating Point Operations Per Second

$$\text{MFLOPS} = \frac{\text{Anzahl der ausgeführten Gleitkommaoperationen}}{10^6 \times \text{Ausführungszeit}}$$

- Probleme mit diesen Maßzahlen
  - Abhängigkeit von ISA und ausgeführter Befehlssequenz
  - Problem:
    - Vergleich von Rechnern mit unterschiedlicher ISA
    - Unterschiedliche MIPS/MFLOPS-Zahlen bei verschiedenen Programmen
    - MIPS kann umgekehrt zur Leistung variieren
      - Beispiel: Gleitkommarechnung in Hardware bzw. mit Software-Routinen
    - MIPS/MFLOPS Angaben von Herstellern oft nur best-case-Annahme: theoretische Maximalleistung

- **Zusammenfassung**

- Vergleich von Rechnern bezüglich ihrer Leistung ohne großen Aufwand
- Maßzahlen bewerten nur spezielle Aspekte
- Kritische Betrachtung der Leistungsangabe unbedingt notwendig!
- Angabe einer hypothetische Maximalleistung!

- **Benchmarks**

- Bewertung der Leistungsfähigkeit aufgrund von Messungen

- Programm oder Programmsammlung im Quellcode
- Übersetzung notwendig
- Messung der Ausführungszeiten
- In die Bewertung fließt „Güte“ des Compiler und Betriebssoftware ein
- Zugriff auf die Maschinen notwendig

- **Kernels**

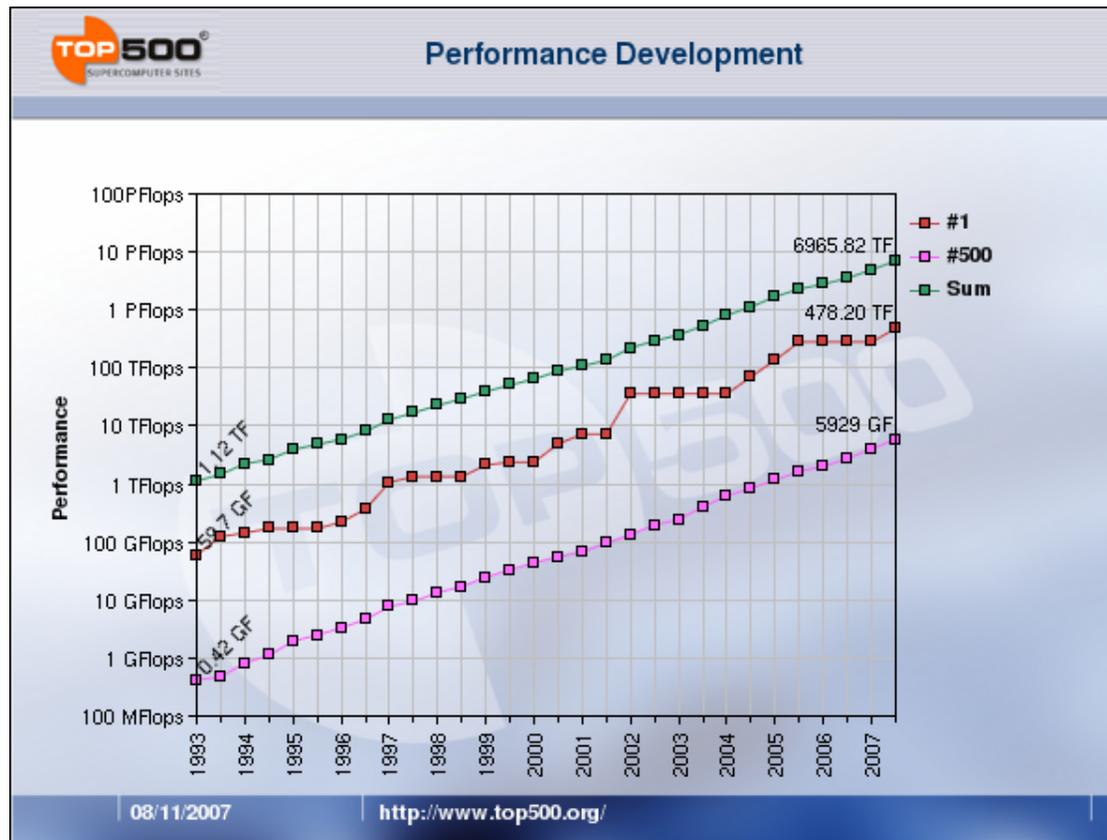
- Rechenintensive Teile realer Programme
- Vorwiegend numerische Algorithmen
- **Beispiele:**
  - **Lawrence Livermore Loops:**
    - Zur Bewertung vektorisierender Compiler
  - **LINPACK Softwarepaket:**
    - Lösung eines Systems linearer Gleichungen
    - Verwendet für die TOP500 Liste (<http://www.top500.org>)
  - **BLAS (Basic Linear Algebra Subprograms)**
- Wenig Aufwand, aber nur bedingt aussagekräftig

- **Kernels**
  - LINPACK Softwarepaket:
    - Verwendet für die TOP500 Liste
      - <http://www.top500.org>

- **Kernels**

- LINPACK Softwarepaket:

- Verwendet für die TOP500 Liste



- **Standardisierte Benchmarks**
  - Ziel: Vergleichbarkeit von Rechnern (inkl. Betriebssystem und Compiler)
  - Anforderungen:
    - Gute Portierbarkeit
    - Repräsentativ für typische Nutzung der Rechner
  - Sammlung von Benchmark-Programmen (Benchmark Suites)
    - Ausgeglichene Bewertung durch die unterschiedlichen Eigenschaften der Programme

- Standardisierte Benchmarks
  - Standardisierungsorganisationen
    - TPC (Transaction Processing Performance Council)
      - Mitte der 80'er Jahre, <http://www.tpc.org>
      - Zusammenschluss von Datenbank- und Rechnerherstellern
      - Ziel: Bewertung von Datenbanksystemen

- **Standardisierte Benchmarks**
  - Standardisierungsorganisationen
    - EEMBC (Embedded Microprocessor Benchmark Consortium)
      - <http://www.eembc.org>
      - Anwendungen aus dem Bereich Eingebettete Systeme
      - Automotive/Industrial (Automatisierungstechnik)
        - » 6 Microbenchmarks (arithmetische Operationen, Pointer, Speicherleistung, Matrizenarithmetik, Tabellensuche, Bitmanipulation), 5 Steuerungsprogramme aus dem Automobilbereich, 5 Filter und FFT-Benchmarks
      - Consumer (Unterhaltungselektronik)
        - » 5 Multimedia-Benchmarks (JPEG compress/decompress, Filtering, RGB Conversions)
      - Networking (Netzwerk)
        - » Kürzeste-Wege-Berechnung, IP Routing, Paketfluss-Operationen
      - Office Automation
        - » Graphik- und Text-Benchmarks, (Bézier-Kurvenberechnung, Dithering, Bildrotation, Textverarbeitung)
      - Telecommunication
        - » Filtering und DSP-Anwendungen

- **Standardisierte Benchmarks**
  - Standardisierungsorganisationen
    - SPEC (Standard Performance Evaluation Corporation)
      - Gegründet 1988, <http://www.spec.org>
      - Zusammenschluss von mehr als 40 Firmen (Rechnerhersteller)
      - Festlegung von Richtlinien für eine gemeinsame Rechnerbewertung

- **Verschiedene Untergruppen in SPEC**
  - Open Systems Group (OSG)
    - SPEC CPU 2000: Prozessorleistung
    - SPEC JVM98: Vergleich von Java Virtual Machine Client Plattformen
    - SPEC MAIL2001: Mail Server Benchmark
    - SPEC SFS 2.0: System: File Server
    - SPEC WEB99: Test für WWW Server

- **Verschiedene Untergruppen in SPEC**
  - High Performance Group (HPG)
    - SPEC<sub>hpc96</sub>: Industrielle große Anwendungen
  - Graphics Performance Characterization Group (GPC)
    - Zusammenschluss mit SPEC<sub>copc</sub>: OpenGL Leistungsbewertung

- **SPEC CPU 2000**
  - 12 nichtnumerische Programme in C/C++ (CINT2000)
  - 14 numerische Programme in FORTRAN/C (CFP2000)
  - Strenge, genau festgelegte Regeln
    - Ab CPU95: vollautomatische Messung und Protokollierung
  - Regelmäßige Aktualisierungen (CPU92, CPU95, CPU2000)
    - Laufzeiten werden zu kurz
    - Caches werden größer: größere Datensätze
    - Mehr Praxisnähe: Programme mit schlechterer Datenlokalität

- SPEC CINT2000**

Benchmark	Referenzzeit	Bemerkung
164.gzip	1400	Datenkompression
175.vpr	1400	FPGA Layout und Routing
176.gcc	1100	C Compiler
181.mcf	1800	Lineare Optimierung, KFZ-Scheduling
186.crafty	1000	Schachprogramm
197.parser	1800	Sprachverarbeitung, Parser für Englisch
252.eon	1300	Ray Tracing
253.perlbmk	1800	Perl
256.gap	1100	Gruppentheorie, Interpreter
255.vortex	1900	Datenbank, objektorientiert
256.bzip2	1500	Datenkompression
300.twolf	3000	IC Platzierung und Routing

Referenzzeit bezieht sich auf Sun Ultra10 (300MHz, 256MB)

- SPEC CFP2000**

Benchmark	Referenzzeit	Bemerkung
168.wupwize	1600	Quantenphysik
171.swim	3100	Gleichungslösung, Flachwasser-Modellierung
172.mgrid	1800	3D-Mehrgitterverfahren
173.applu	2100	Lösung parabolischer u. elliptischer part. DG
177.mesa	1400	3D Graphikbibliothek
178.galgel	2900	Strömungsmechanik
179.art	2600	Bildererkennung, neuronales Netzwerk Simulation

- SPEC CFP2000**

Benchmark	Referenzzeit	Bemerkung
183.equake	1300	Seismografische Wellenausbreitung, Finite Elemente Simulation
187.facerec	1900	Bildverarbeitung, Gesichtserkennung
188.amp	2200	Chemie, Molekulardynamik-Sim.
189.lucas	2000	Zahlentheorie, Mersenne-Primzahltest
191.fma3d	2100	Finite Elemente: Crash-Simulation
200.sixtrack	1100	Teilchenbeschleunigermodell
300.apsi	2600	Sim. Schadstoffausbreitung

- **Neueste Version:**
  - [SPEC CINT2006 Benchmarks.htm](#)
  - [SPEC CFP2006 Benchmark Descriptions.htm](#)

- Mögliche Kategorien

	<b>Geschwindigkeit</b>	<b>Durchsatz</b>
<b>Aggressive Optimierung</b>	SPECint2000	SPECint_rate2000
	SPECfp2000	SPECfp_rate2000
<b>Konservative Optimierung</b>	SPECint_base2000	SPECint_rate_base2000
	SPECfp_base2000	SPECfp_rate_base2000

- SPEC CPU Benchmark-Metrik: Geschwindigkeit

$$\text{SPECratio} = \frac{\text{Referenzzeit}_x}{\text{Laufzeit}_x \text{ auf Testsystem}}$$

Benchmark x

– Endwerte: je ein geometrisches Mittel der SPECratio's über alle CINT2000 und CFP2000 Benchmarks

- SPECint2000, SPECfp2000
  - Aggressive, individuelle Optimierungen erlaubt
- SPECint\_base2000, SPECfp\_base2000
  - Nur mit konservativer Standardoptimierung
  - Identische Compileroptionen für alle Programme

- SPEC CPU Benchmark-Metrik
  - Warum geometrisches Mittel:

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

mit *Execution time ratio*<sub>*i*</sub>:  
 Ausführungszeit der Programms *i*  
 einer Last von *n* Programmen,  
 normalisiert bezüglich der  
 Referenzmaschine

- Eigenschaft des geometrischen Mittels:

$$\frac{\text{geometrisches Mittel } (X_i)}{\text{geometrisches Mittel } (Y_i)} = \text{geometrisches Mittel} \left( \frac{X_i}{Y_i} \right)$$

- Geometrisches Mittel ist konsistent, unabhängig von Referenzmaschine!

## • SPEC CPU Benchmark-Metrik: Durchsatz

$$\text{SPECrate}_x = \frac{\text{Sekunden pro Stunde}}{\text{Laufzeit}_x \text{ von } n_x \text{ Kopien auf Testsystem}} \times \frac{\text{Referenzzeit}_x}{\text{längste Laufzeit}}$$

– Endwerte: je ein **geometrisches Mittel** der **SPECrate's** über alle CINT2000 bzw. CFP2000 Benchmarks

- SPECint\_rate2000, SPECfp\_rate2000
- SPECint\_base2000, SPECfp\_base2000
- $n_x$  kann frei gewählt werden, muss aber dokumentiert werden

Processor	Alpha 21364	AMD Athlon XP	HP PA-8700	IBM Power 4+	Intel Itanium 2	Intel XeonMP	Intel Xeon	MIPS R14000	Sun UltraSPARC III
System or Motherboard	Alpha GS1280/7	ASUS A7N8X	HP9000 C3750	pSeries 650 6M2	HP RX2600	Dell PwrEdg 6650	Dell Prec. 350	SGI 3200	Sun Blade 2050
Clock Rate	1.15GHz	2.17GHz	870MHz	1.45GHz	1.0GHz	2.0GHz	3.06GHz	600MHz	1.05GHz
External Cache	None	None	None	16MB	None	None	None	8MB	8MB
164.gzip	583	1,026	588	673	583	758	1,138	322	433
175.vpr	822	653	688	902	704	625	606	572	460
176.gcc	859	755	906	914	1,014	1,100	1,236	445	577
181.mcf	712	420	494	1,391	834	599	773	783	659
186.crafty	982	1,292	751	884	781	712	1,179	502	558
197.parser	514	905	495	381	660	778	1,025	409	488
252.eon	958	1,483	592	1,150	1,004	920	1,387	507	527
253.perlbnk	768	1,306	619	712	815	952	1,381	367	540
254.gap	636	1,059	339	936	680	722	1,417	308	372
255.vortex	1,094	1,608	1,196	1,428	1,193	1,118	1,658	679	738
256.bzip2	824	840	534	965	759	712	856	493	629
300.twolf	1,018	887	911	1,198	880	1,009	900	645	570
<b>SPECint_base2000</b>	<b>795</b>	<b>960</b>	<b>642</b>	<b>909</b>	<b>810</b>	<b>816</b>	<b>1,085</b>	<b>483</b>	<b>537</b>
168.wupside	883	1,131	446	1,532	1,003	816	1,406	434	659
171.swim	3,590	1,006	931	1,417	3,205	848	1,837	529	980
172.mgrid	708	799	621	850	1,720	449	1,047	379	487
173.applu	1,518	654	702	979	2,033	496	1,168	381	310
177.mesa	928	1,103	694	737	642	814	1,165	425	543
178.galgel	2,105	738	1,603	3,186	2,505	1,200	1,536	1,398	1,713
179.art	2,014	495	670	1,864	4,226	1,147	716	1,436	9,389
183.quake	519	730	413	2,098	1,871	449	1,291	347	645
187.facerec	1,105	1,008	430	1,515	1,152	762	1,315	647	958
188.amp	735	587	553	923	788	729	644	573	509
189.lucas	1,522	853	448	1,306	1,206	682	1,522	442	371
191.fma3d	1,019	850	404	898	747	551	1,089	306	400
200.sixtrack	469	538	471	621	894	376	564	298	366
301.aspi	1,242	705	696	966	678	695	833	406	471
<b>SPECfp_base2000</b>	<b>1,124</b>	<b>776</b>	<b>600</b>	<b>1,221</b>	<b>1,356</b>	<b>677</b>	<b>1,092</b>	<b>499</b>	<b>701</b>

Quelle: MDR, Microprocessor Report, April 28, 2003

Processor	AMD 1-core Opteron 854	Intel 1-core Xeon	AMD 2-core Opteron 8224SE	Intel 2-core Xeon 5160	AMD 4-core Opteron 8360SE	Intel 4-core Xeon X7350	Intel 4-core Core 2 Quad QX9650
Bit-width	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit	32/64-bit
Cores/chip x Threads/core	1 x 1	1 x 2	2 x 1	2 x 1	4 x 1	4 x 1	4 x 1
Clock Rate	2.80GHz	3.80GHz	3.20GHz	3.03GHz	2.50GHz	2.93GHz	3.00GHz
Cache: L1-L2-L3 - I/D or Unified	64K/64K - 1M - N/A	12K/16K - 2M - N/A	2 x 64K/64K - 2 x 1M - NA	2 x 32K/32K - 4M - NA	4 x 64K/64K - 4 x 512K - 2M	4 x 32K/32K - 2 x 4M - NA	4 x 32K/32K - 2 x 6M - NA
Execution Rate/Core	3 instructions	3 instructions	3 instructions	1 complex + 3 simple	3 instructions	1 complex + 3 simple	1 complex + 3 simple
Pipeline Stages	12 int / 17 fp	31	12 int / 17 fp	14	12 int / 17 fp	14	14
Out of Order	72	126	72	96	72	96	96
Memory bus	6.4 GB/s	800 MHz	10.6 GB/s	1333 MHz	10.6 GB/s	1066 MT/s	1333 MHz
Package	uPGA 940	LGA-775	LGA-1207	LGA-771	LGA-1207	LGA-771	LGA-775
IC Process	90nm 9M	90nm 7M	90nm 9M	65nm 8M	65nm 8M	65nm 8M	45nm
Die Size	106mm <sup>2</sup>	109mm <sup>2</sup>	227mm <sup>2</sup>	143mm <sup>2</sup>	283mm <sup>2</sup>	2 x 143mm <sup>2</sup>	2 x 107mm <sup>2</sup>
Transistors	120M	169M	233M	291M	463M	2 x 291M	2 x 410M
List Price (intro)	\$1,514	\$903	\$2,149	\$851	N/A	\$2,301	\$999
Power (Max)	93W	110W	120W	80W	120W	130W	130W
Availability	3Q05	3Q05	3Q07	3Q06	1Q08	3Q07	4Q07
Scalability	2-4 chips	1-2 chips	1-4 chips	1-2 chips	2-4 chips	1-4 chips	1 chip
SPECint/fp2006 [cores]	11.2/12.1 [2]	11.4/11.7 [2]	14.1/14.2 [8]	19.7*/18.3* [4]	N/A	21.7*/18.9* [16]	22.3*/21.4* [4]
SPECint/fp2006_rate [cores]	41.4/45.6 [4]	20.9/18.8 [2]	105/96.7 [8]	60.8/45.1 [4]	163/149 [16]	184*/108* [16]	69.0*/49.9 [4]

Quelle: MDR, Microprocessor Report, January 2008

Processor	Intel Itanium 2 9050	Intel Itanium 9150M	IBM Power6	IBM Power5+	Fujitsu SPARC64 VI	Sun UltraSPARC IV+	Sun UltraSPARC T2
Bit-width	64-bit	64-bit	64-bit	64-bit	64-bit	64-bit	64-bit
Cores/chip x Threads/core	2 x 2	2 x 2	2 x 2	2 x 2	2 x 2	2 x 1	8 x 8
Clock Rate	1.60GHz	1.67GHz	4.70GHz	2.20GHz	2.40GHz	1.95GHz	1.40GHz
Cache: L1-L2-L3 I/D or Unified	2 x 16K/16K - 1M/256K - 12M(on)	2 x 16K/16K - 1M/256K - 12M(on)	2 x 64K/64K - 2 x 4M - 32M(off)	2 x 64K/32K - 1.92M - 36M(off)	2 x 128K/128K - 5M - NA	2 x 64K/64K - 2M - 32M(off)	8 x 8K/16K - 4M - NA
Execution Rate/Core	6 issue	6 issue	7 issue	5 issue	4 issue	4 issue	16 issue
Pipeline Stages	8 stages	8 stages	13 stages	15 stages	15 stages	14 stages	8 int / 12 fp
Out of Order	None	None	"Limited"	200	64	None	None
Memory B/W	8.5GB/s	10.6GB/s	75GB/s	12.8GB/s	8GB/s	4.8GB/s	42.7GB/s
Package	mPGA-700	mPGA-700	N/A	MCM-5370 pins	412 I/O pins	FC-LGA 1368	1831 pins
IC Process	90nm 7M	90nm 7M	65nm 10m	90nm 10m	90nm 10M	90nm 9M	65nm
Die Size	596mm <sup>2</sup>	596mm <sup>2</sup>	341mm <sup>2</sup>	245mm <sup>2</sup>	421mm <sup>2</sup>	335mm <sup>2</sup>	342mm <sup>2</sup>
Transistors	1.72 billion	1.72 billion	790 million	276 million	540 million	295 million	503 million
List Price (intro)	\$3,692	\$3,692	N/A	N/A	N/A	N/A	N/A
Power (Max)	104W	104W	~100W	100W	120W	90W	84W
Availability	3Q06	4Q07	2Q07	4Q05	2Q07	3Q06	3Q07
Scalability	1-64 chips	8-128 chips	2-32 chips	1-32 chips	4-64 chips	1-72 chips	1 chip
SPECint/fp2006 [cores]	14.5/17.3 [2]	N/A	17.8/18.7 [1]	10.5/12.9 [1]	9.7/11.1 [32]	N/A	N/A
SPECint/fp2006_rate [cores]	1534/1671 [128]	1832/N/A [128]	420/379 [16]	197/229 [16]	1111/1160 [128]	1120/N/A [144]	73.1/58.1

Quelle: MDR, Microprocessor Report, January 2008

- Hennessy/Patterson: A Quantative Approach:  
Kap. 1.5 – 1.9