

# Rechnerstrukturen

Vorlesung im Sommersemester 2008

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



## Kapitel 1: Grundlagen

### 1.2 Einführung, Entwurfsfragen

- **Energiespar-Techniken**

- 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

## • Energiespar-Techniken

### – 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

- **Energiespar-Techniken**

- 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:  $C * a$

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

- **Energiespar-Techniken**

- 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

- **Energiespar-Techniken**

- 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.

- **Energiespar-Techniken**

- 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  $P \sim f^3$

- **Energiespar-Techniken**

- 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

### 1.3 Einführung: Bewertung der Leistungsfähigkeit

## Kapitel 1: Grundlagen

### 1.3 Einführung: 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.

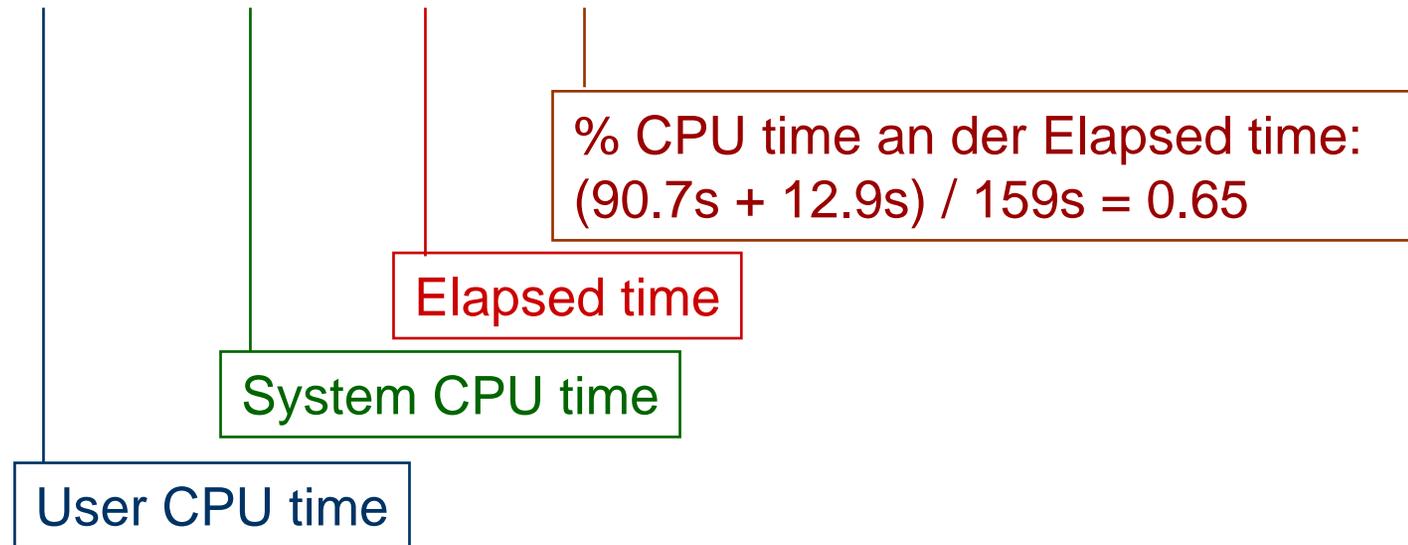
- **Ausführungszeit (execution time)**
  - Wall-clock time, response time, elapsed time
    - Latenzzeit für die Ausführung einer Aufgabe
      - Schließt den Speicher- und Plattenzugriff, Ein-/Ausgabe etc. mit ein.
  - CPU Time
    - Zeit, in der die CPU arbeitet
      - User CPU Time: Zeit, in der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