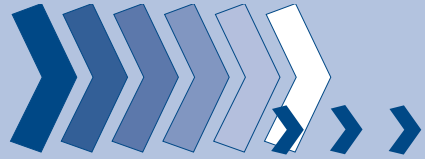


Übung 2: Befehls- und Datentypen, Rechenleistung

Technische Informatik II – Übung
Sommersemester 2008



1. Befehle

- Befehlsarten
- grundlegender Befehlsaufbau

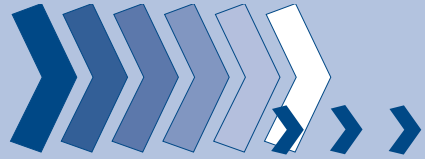
2. Grundlegende Datentypen und Benennung

3. Rechenleistung

- Leistungsindikatoren
- Rechenbeispiel

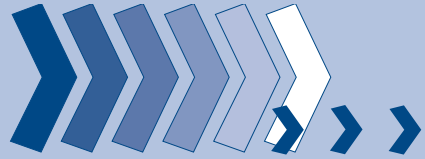
4. „echte“ Benchmarks

5. Projekte: Bilden von Teams und deren Anmeldung



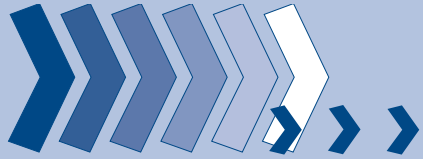
- „Geh' ins Bad“
- „Wasch die Tassen und die Teller ab und räume sie anschließend in den Schrank“
- „Geh' Einkaufen, wenn noch Geld da ist.“
- „Hole das Werkzeug aus dem Keller und repariere den Esszimmerstuhl“

→ *unterschiedliche Situationen erfordern unterschiedliche Befehlsarten*

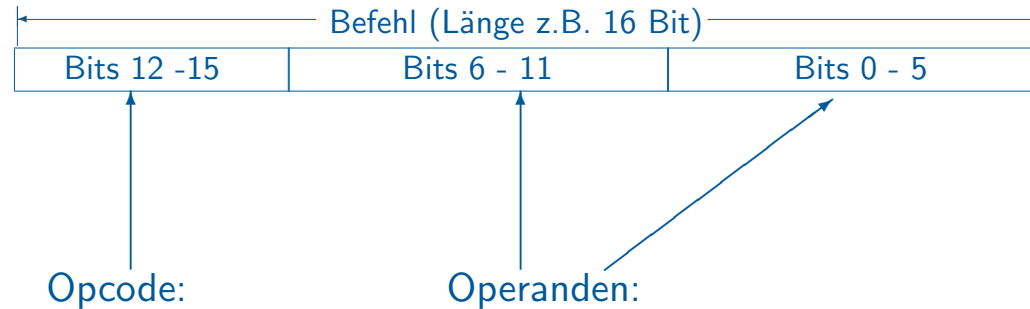


Einteilung nach Funktion / Operanden

- Arithmetische Befehle (Fließkomma, Integer)
 - mit einem Operanden: z.B. rotate left, negation, increment
 - mit zwei Operanden: z.B. add, mul, or, compare, load
 - mit drei und mehr Operanden: z.B. add, mul, load, rtlwm
- Datenbewege Befehle, Quelle und Ziel sind vielfältig wählbar: Register, Speicher, Externer Bus: z.B. lda, store byte, load double word
- Sprünge, Verzweigungen, mit/ohne Bedingung, relativ/direkt ... z.B. branch if zero, branch relative to REG, jump always



Aufbau eines Befehls



Befehl als solcher wird festgelegt: Addiere! Springe!

Hier werden meist nicht die Operanden selbst gespeichert sondern nur ihr Aufenthaltsort angegeben.

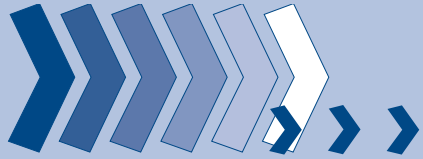
Unterschiedliche Adressierungsarten erlauben es Operanden von verschiedenen Plätzen zu holen

Beispiel der PDP-11:

0 1 1 0
add

S S S S S S
source

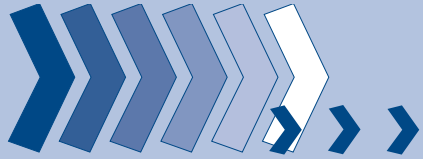
D D D D D D
destination



Datentypen und deren Namen

Länge	Intel, M68k, 16-Bitter	PowerPC, MIPS, Sparc	Gnu-C 32Bit	Gnu-C 64Bit
8 Bit	Byte	Byte	byte, char	byte, char
16 Bit	Word	Halfword	short	short
32 Bit	Doubleword	Word	int, long, <i>float</i>	int, <i>float</i>
64 Bit	Quadword	Doubleword	long long, <i>double</i>	long, <i>dou- ble</i>

Aber *Wort* als *Befehlswort*, Länge eines Befehls



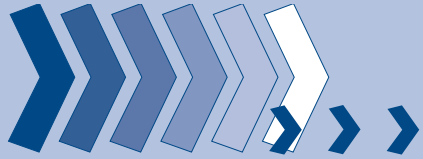
Leistungsindikatoren:

- Antwortzeit auf eine Eingabe
- Ausführungszeit einer Task

Antwortzeit kann gemessen werden, Vorhersage ist sehr schwierig, zudem schwankt die Antwortzeit je nach aktuellem Zustand des Systems erheblich.

Ausführungszeit = CPU-Zeit:

$$\text{Ausfuehrungszeit} = \text{Befehlsanzahl} \cdot \frac{(\ominus)\text{Takte}}{\text{Befehl}} \cdot \frac{\text{Zeit}}{\text{Takt}}$$



Rechenleistung: Leistungsindikatoren II

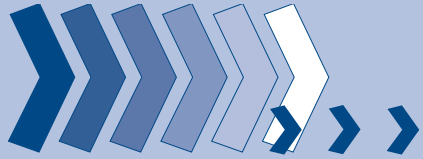
Cycles per Instruktion *CPI* (dt: Takte pro (Maschinen-)Befehl)

Die Dauer $CPI(i)$ jedes einzelnen Befehls i sei bekannt.

$$CPI = \frac{\sum_i (CPI(i) \cdot Anzahl(i))}{Befehlsanzahl}$$

Millions of Instructions per Second: MIPS

$$MIPS = \frac{Befehlsanzahl}{Ausfuehrungszeit \cdot 10^6} = \frac{Taktfrequenz}{CPI \cdot 10^6}$$



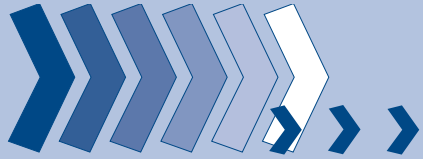
Rechenleistung: Beispiel I

rel. Häufigkeit	Beschreibung	Dauer
20%	load from memory	3 Takte
15%	store to memory	3 Takte
30%	simple math	1 Takt
7%	complex math	22 Takte
13%	compare	2 Takte
15%	branch	4 Takte

Berechnen Sie Ausführungszeit, MIPS, CPI mit und ohne Befehle der Sorte complex math. Der Prozessortakt beträgt 750MHz.

$$\text{Ausfuehrungszeit} = \text{Befehlsanzahl} \cdot \frac{(\otimes)\text{Takte}}{\text{Befehl}} \cdot \frac{\text{Zeit}}{\text{Takt}}$$

$$\text{CPI} = \frac{\sum_i (\text{CPI}(i) \cdot \text{Anzahl}(i))}{\text{Befehlsanzahl}}; \quad \text{MIPS} = \frac{\text{Taktfrequenz}}{\text{CPI} \cdot 10^6}$$



Rechenleistung Beispiel II

$$CPI : 0.35 \cdot 3 + 0.30 + 0.07 \cdot 22 + 0.13 \cdot 2 + 0.15 \cdot 4 = 3.75 \text{ CPI}$$

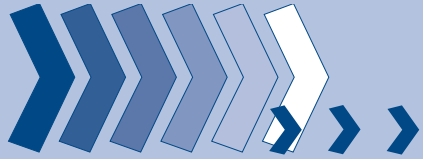
$$MIPS : \frac{750MHz}{3.75} = 200 \text{ MIPS}$$

$$Abarbeitungszeit : \frac{1}{750MHz} \cdot 3.75 = 5 \text{ ns}$$

$$CPI : \frac{0.35 \cdot 3 + 0.30 + 0.13 \cdot 2 + 0.15 \cdot 4}{0.93} = 2.376 \text{ CPI}$$

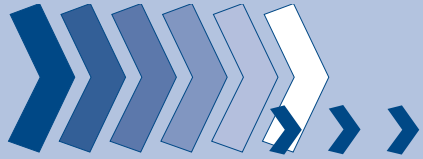
$$MIPS : \frac{750MHz}{2.376} = 315.6 \text{ MIPS}$$

$$Abarbeitungszeit : \frac{1}{750MHz} \cdot 2.376 = 3.179 \text{ ns}$$



Rechenleistung: Erkenntnisse

- *MIPS*, *CPI* und Abarbeitungszeit können von Programm zu Programm erheblich schwanken.
- die Messungen sollten mit Programmen gemacht werden, die den Nutzer interessieren
- die Gewichtung der Teil – Programme sollte der Realität entsprechen
- Rechenleistung hängt nicht nur vom Prozessor allein ab, sondern auch von anderen Faktoren wie Speicherzugriffszeit, I/O Durchsatz u.ä.



Spec CPU2000

- reine CPU– und RAM – Geschwindigkeit soll getestet werden
- Referenzmaschine *Sun Ultra5_10* mit 300 MHz und 256 MByte RAM erreicht in allen Disziplinen 100 Punkte
- Teilung in 2 Unterbenchmarks: Spec FP2000 und Spec INT2000 mit 14 bzw. 12 Einzeltests
- Punktzahl für einen Einzelwert ergibt sich aus $\frac{\text{Referenzzeit}}{\text{gemessene Laufzeit}} \cdot 100$
- Gesamtwert ist das geometrische Mittel der Einzeltests i : $\sqrt[n]{\prod_i \text{val}(i)}$