# Technische Grundlagen der Informatik – Kapitel 8



Prof. Dr. Jan Peters
Fachgebiet Intelligente Autonome Systeme (IAS)
Fachbereich Informatik

WS 14/15



# Meisenantworten ... vielen Dank für die Hinweise!



- VHDL Klausurelevant? Ja...
- Synthaxübersicht MIPS? SLT? Herke hat eine online gestellt.
- Schrift im Assembler größer? Versuche ich!
- Folien vor Vorlesung online? Versuche ich!
- Reload Fehler Java-Skript Hint? Eingebaut....



# **Kapitel 8: Themen**



- Einleitung
- Zugriffszeitvergleich von Speichersystemen
- Caches
- Virtueller Speicher
- Speichereinblendung von Ein-/Ausgabegeräten
- Zusammenfassung

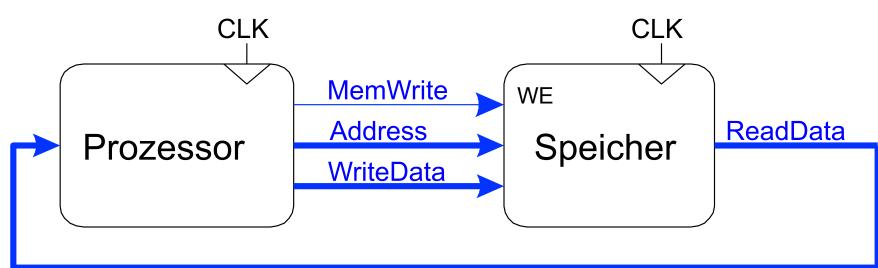


# **Einleitung**



- Rechenleistung hängt ab von:
  - Prozessorleistung
  - Leistung des Speichersystems

### **Speicherschnittstelle**

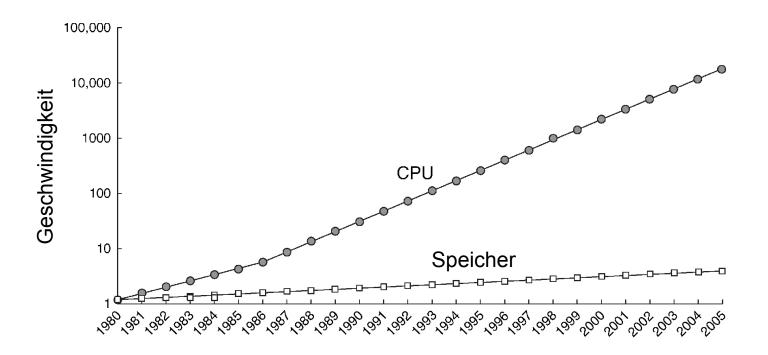




# **Einleitung**



- Annahme bisher in der Vorlesung: Speicherzugriffe dauern 1 Takt
- Ist aber seit den 1980'er Jahren nicht mehr wahr





# Herausforderungen beim Entwurf von Speichersystemen



- Speichersystem soll so schnell sein wie Prozessor
  - Zumindest dem Anschein nach ...
- Idealer Speicher:
  - Schnell
  - Billig
  - Hohes Fassungsvermögen

Praktisch: Nur zwei von drei Eigenschaften realisierbar!

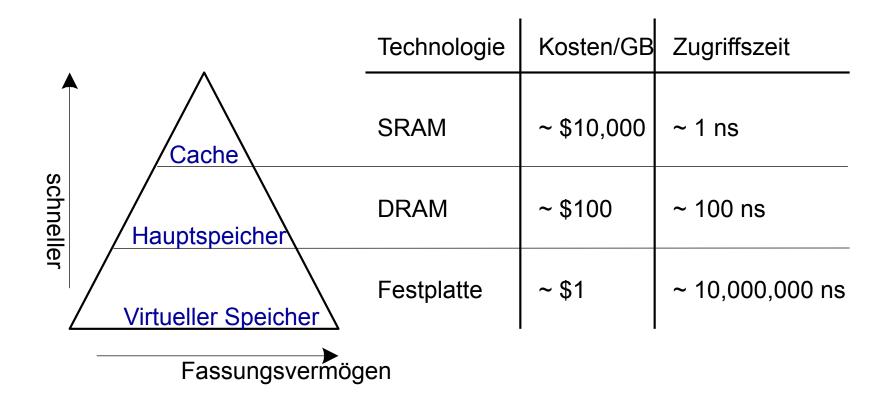
Verwende Hierarchie von Speichern





# Speicherhierarchie: Beispiele für Ebenen







#### Lokalität



Nutze Lokalität zur Beschleunigung von Speicherzugriffen aus

#### Zeitliche Lokalität:

- Ein gerade benutztes Datum wird wahrscheinlich bald wieder gebraucht
- Ausnutzung: gerade benutzte Daten in höheren Ebenen der Speicherhierarchie halten

#### Räumliche Lokalität:

- Um ein benutztes Datum herum liegende Daten werden wahrscheinlich auch bald gebraucht
- Ausnutzung: Beim Zugriff auf ein Datum auch benachbarte Daten in höhere Ebenen der Speicherhierarchie bringen



# Zugriffszeit eines Speichersystems



- Treffer (hit): Datum wird auf dieser Ebene der Speicherhierachie gefunden
- Verfehlt (miss): nicht gefunden (suche auf tieferer Ebene)

Hit-Rate = # hits / # Speicherzugriffe

= 1 – Miss Rate

**Miss-Rate** (*MR*) = # misses / # Speicherzugriffe

= 1 – Hit Rate

 Durchschnittliche Speicherzugriffszeit (average memory access time, AMAT): Durchschnittliche Zeit die der Prozessor braucht, um auf ein Datum zuzugreifen

$$AMAT = t_{cache} + MR_{cache} (t_{MM} + MR_{MM} t_{VM})$$

MM = main memory, HauptspeicherVM = virtual memory, Virtueller Speicher



# Beispiel 1: Zugriffszeit eines Speichersystems



- Ein Programm hat 2.000 Load- und Store-Befehle
- 1.250 der Daten werden im Cache vorgefunden
- Der Rest kommt aus anderen Ebenen der Speicherhierarchie
- Was sind die Hit- und Miss-Rates des Caches?



# Beispiel 1: Zugriffszeit eines Speichersystems



- Ein Programm hat 2.000 Load- und Store-Befehle
- 1.250 der Datenpunkte werden im Cache vorgefunden
- Der Rest kommt aus dem Hauptspeicher
- Was sind die Hit- und Miss-Rates des Caches?

Hit Rate = 1250/2000 = 0,625

Miss Rate = 750/2000 = 0.375 = 1 - Hit Rate



# Beispiel 2: Zugriffszeit eines Speichersystems



- Annahme: Prozessor hat zwei Hierarchieebenen:
  - Cache
  - Hauptspeicher
- *t*<sub>cache</sub> = 1 Takt
- *t<sub>MM</sub>* = 100 Takte

• Wie lang ist die AMAT des Programmes aus Beispiel 1?



# Beispiel 2: Zugriffszeit eines Speichersystems



- Annahme: Prozessor hat zwei Hierarchieebenen:
  - Cache
  - Hauptspeicher
- *t*<sub>cache</sub> = 1 Takt
- $t_{MM}$  = 100 Takte

• Wie lang ist die AMAT des Programmes aus Beispiel 1?

**AMAT** = 
$$t_{\text{cache}} + MR_{\text{cache}} t_{MM}$$
  
=  $[1 + 0.375 * 100]$  Takte  
= **38.5 Takte**



## Gene Amdahl, 1922 -



- Amdahls Gesetz: Die Optimierung eines Subsystems bringt nur dann etwas, wenn das Subsystem tatsächlich großen Einfluss auf die Gesamtrechenleistung hat
- Gründete drei Firmen, darunter auch die Amdahl Corporation in 1970
  - IBM-kompatible Großrechner
    - I.d.R. schneller und/oder billiger





#### LEHRE WIKI FRAGE



Bitte jetzt auf LEHRE WIKI eine Frage beantworten!



#### Cache



#### Geheimes Lager

- Höchste Ebene der Speicherhierarchie
- Schnell (oft Zugriffszeit ~ 1 Takt)
- Stellt dem Prozessor idealerweise die meisten benötigten Daten zur Verfügung
- Speichert (i.d.R.) die zuletzt benutzten Daten



# Aufbau von Caches: Entwurfsentscheidungen



- Welche Daten werden im Cache gehalten?
- Wie werden die Daten gefunden?
- Wie werden Daten ersetzt?

Schwerpunkt hier auf Loads, Stores werden aber ähnlich gehandhabt



# Welche Daten werden im Cache gehalten?



- Idealfall: Cache "ahnt" im Voraus, welche Daten der Prozessor benötigen wird und hält diese bereit
- Praxis: Prophezeiungen in der Regel ungenau
- Basiere Vorhersagen auf bisherigem Verhalten
  - Zeitliche Lokalität: kopiere gerade benutzte Daten in den Cache. Bei nächster Verwendung werden die Daten im Cache gefunden (Cache Hit).
  - Räumliche Lokalität: kopiere benachbarte Daten auch in den Cache
    - Blockgröße: Anzahl von Bytes, die immer zusammen in den Cache kopiert werden



# **Caches: Begriffe**



- Kapazität (capacity, C):
  - Anzahl der im Cache speicherbaren Bytes
- Blockgröße (block size, b):
  - Anzahl der auf einen Satz in den Cache geladenen Bytes
- Blockanzahl (B = C/b):
  - Anzahl von Blöcken im Cache: B = C/b
- Assoziativitätsgrad (degree of associativity, N):
  - Anzahl von Blöcken in einer Assoziativitätsmenge (kurz: Menge)
- Anzahl von Assoziativitätsmengen (S = B/N):
  - Jede Speicheradresse wird auf genau eine Assoziativitätsmenge abgebildet



# Wie werden Daten gefunden?

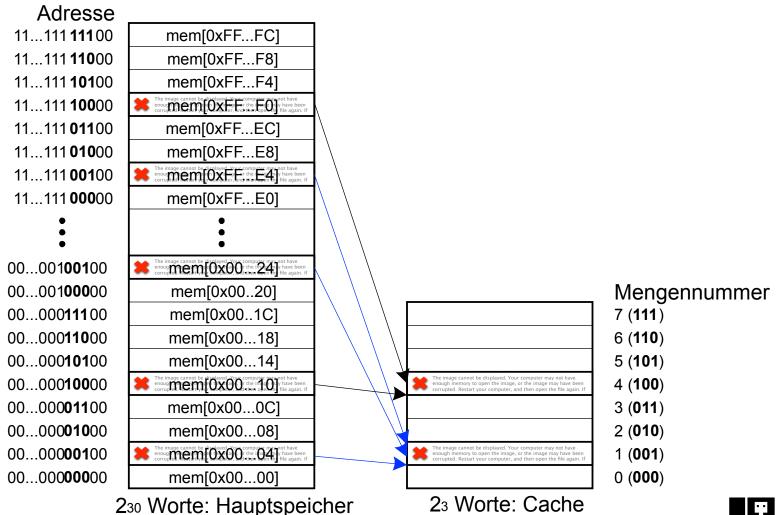


- Cache ist organisiert als S Mengen (sets)
- Jede Speicheradresse wird genau auf eine Menge abgebildet
- Caches werden klassifiziert nach Anzahl von Blöcken in einer Menge:
  - Direktabgebildet (direct mapped): Ein Block pro Menge
  - N-fach Mengenassoziativ (N-way set associative): N Blöcke pro Menge
  - Vollassoziativ (fully associative): Alle Cache Blocks in einer Menge
- Beispielorganisationen für einen Cache mit:
  - Kapazität C = 8 Worte
  - Blockgröße *b* = 1 Wort
  - Damit ist Blockanzahl B = 8



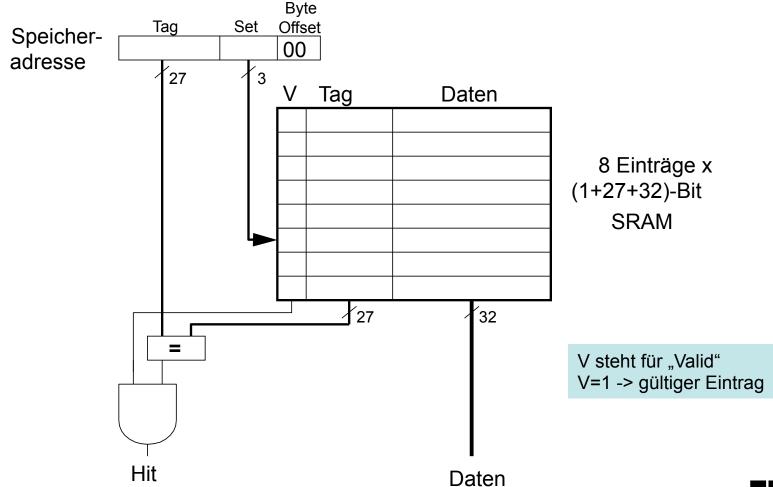
# Direktabgebildeter Cache





# Direktabgebildeter Cache: Hardware

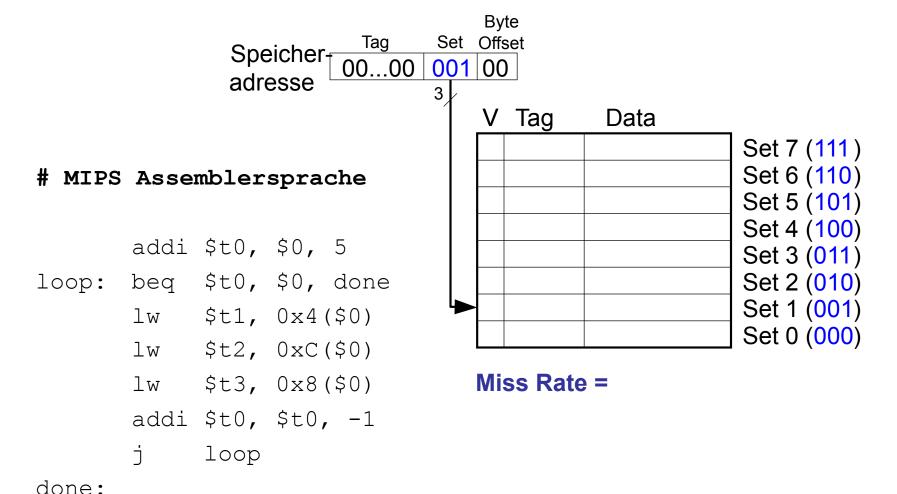






# Miss Rate eines direktabgebildeten Caches







# Miss Rate eines direktabgebildeten Caches

LEHRE WIKI TEST IN DREI FOLIEN



#### # MIPS Assemblersprache

addi \$t0, \$0, 5

loop: beq \$t0, \$0, done

lw \$t1, 0x4(\$0)

lw \$t2, 0xC(\$0)

lw \$t3, 0x8(\$0)

addi \$t0, \$t0, -1

j loop

_	V	Tag	Data	_
	0			Set 7 (111)
	0			Set 6 (110)
	0			Set 5 (101)
	0			Set 4 (100)
	1	0000	mem[0x000C]	Set 3 (011)
	1	0000	mem[0x0008]	Set 2 (010)
>	1	0000	mem[0x0004]	Set 1 (001)
	0			Set 0 (000)

Miss Rate = 3 / 15 = 20%

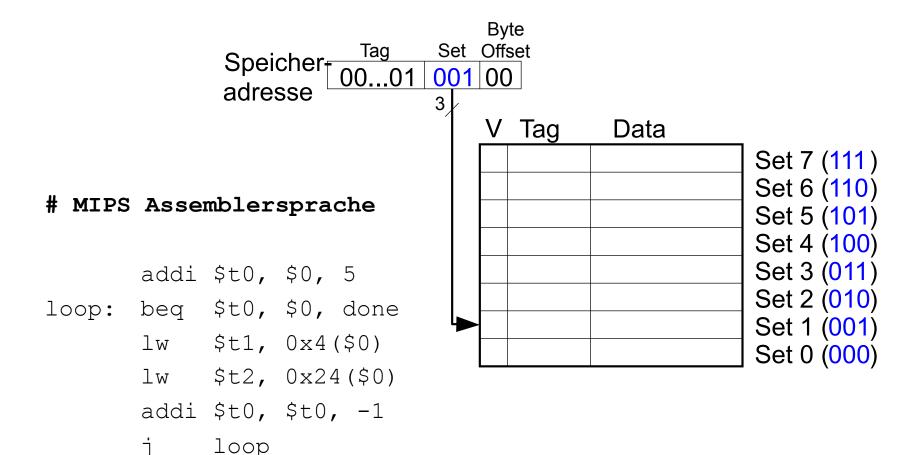
Temporale Lokalität
Unvermeidbare (compulsory) Misses

done:



# Konflikte bei direktabgebildeten Caches





done:

# Konflikte bei direktabgebildeten Caches





#### # MIPS Assemblersprache

addi \$t0, \$0, 5

loop: beq \$t0, \$0, done

lw \$t1, 0x4(\$0)

lw \$t2, 0x24(\$0)

addi \$t0, \$t0, -1

j loop

Tag Data Set 7 (111) 0 Set 6 (110) 0 Set 5 (101) 0 Set 4 (100) Set 3 (011) Set 2 (010) mem[0x00...04] 00...00 Set 1 (001) 00 01 Set 0 (000)

Miss Rate = 10/10 = 100%

done:

Sogenannte "Conflict Misses"



#### LEHRE WIKI FRAGE



Bitte jetzt auf LEHRE WIKI eine Frage beantworten!

