Fachbereich Medienproduktion

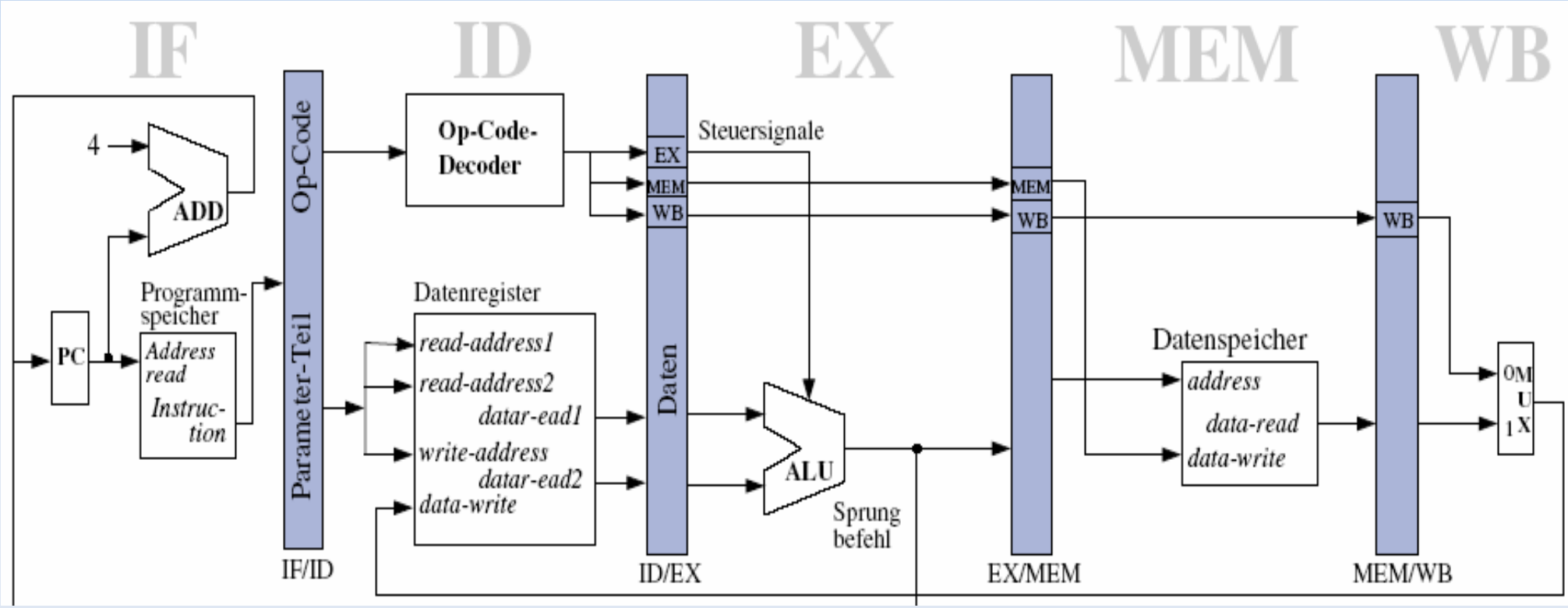
Herzlich willkommen zur Vorlesung im Studienfach:

Grundlagen der Informatik I

Pipelining (Fließbandverarbeitung)

- Instruction fetch
  - Nächster Befehl wird aus dem Speicher geholt
- Instruction decode
  - Dekodieren des Befehls und Holen der Operanden aus den Register
- Execute / address calculation
  - Ausführen der Operanden und Berechnen der Adresse
- Memory Access (optional)
  - Abspeichern des Ergebnisses
- Write Back (optional)
  - Schreiben der Ergebnisse in die Register

RISC Prozessor mit 5-stufigem Pipelining



© Pearson Studium 2006; Grundlagen der Informatik

Abarbeitung der Befehle

Abarbeitung der Befehle bei fünfstufigem Pipelining

Befehl 1:	ID	IF	EX	MEM	WB					
Befehl 2:		ID	IF	EX	MEM	WB				
Befehl 3:			ID	IF	EX	MEM	WB			
Befehl 4:				ID	IF	EX	MEM	WB		
Befehl 5:					ID	IF	EX	MEM	WB	
Befehl 6:						ID	IF	EX	MEM	WB
Takt:	1	2	3	4	5	6	7	8	9	10

© Pearson Studium 2006; Grundlagen der Informatik

Fragen

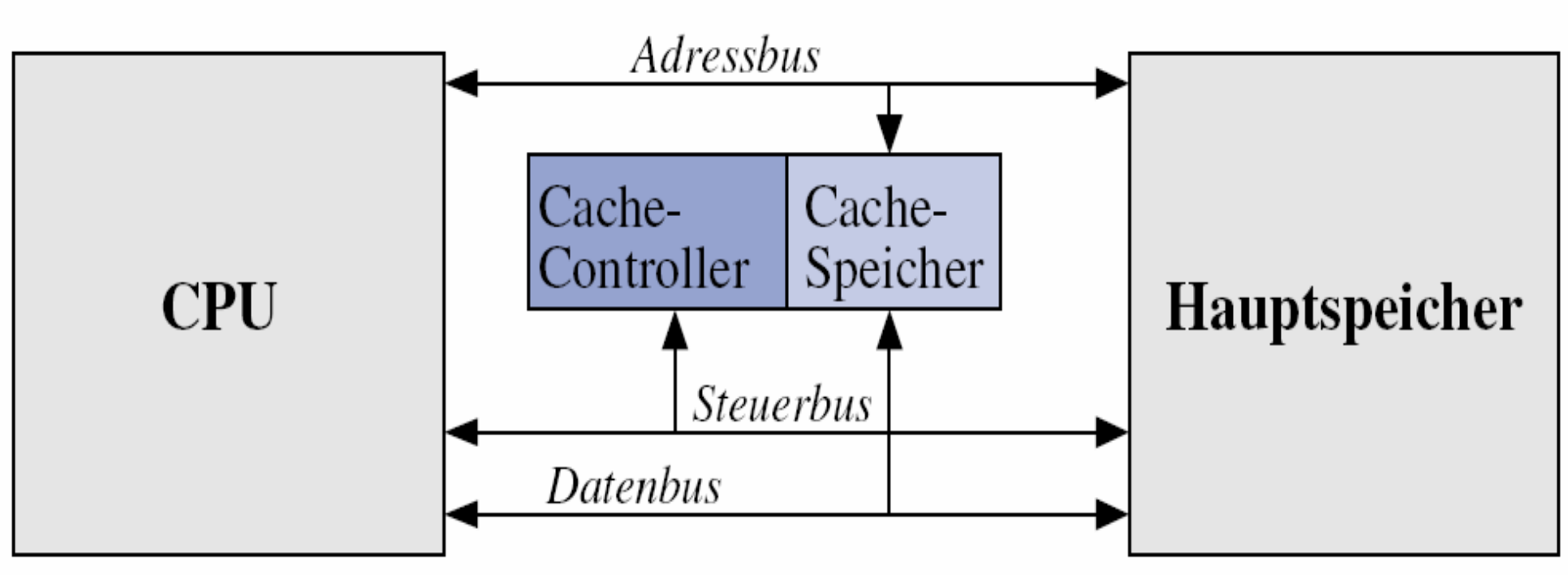
Caches

- Der Cache befindet sich in der Speicherhierarchie zwischen CPU internen Registern und dem Hauptspeicher
  - Direkt in der CPU (on chip cache oder first level cache)
  - Außerhalb der CPU (second level cache)
- Auf den Cache Speicher kann nicht von „außen“ durch den Benutzer zugegriffen werden
- Cache Speicher schließen die Geschwindigkeitslücke zwischen Registern (SRAM) und den Hauptspeicher (DRAM)

Lokalitäts-Prinzip

- Mit sehr großer Wahrscheinlichkeit wird als nächstes auf eine Adresse zugegriffen, die nahe an der Adresse liegt, auf die zuletzt zugegriffen wurde
- Ganze Blöcke vom Hauptspeicher in den Cache kopieren, damit auch noch die benachbarten Werte im Cache stehen
- Gilt auch, wenn auf ein gleiches Datum in kurzer Zeit mehrfach zugegriffen wird

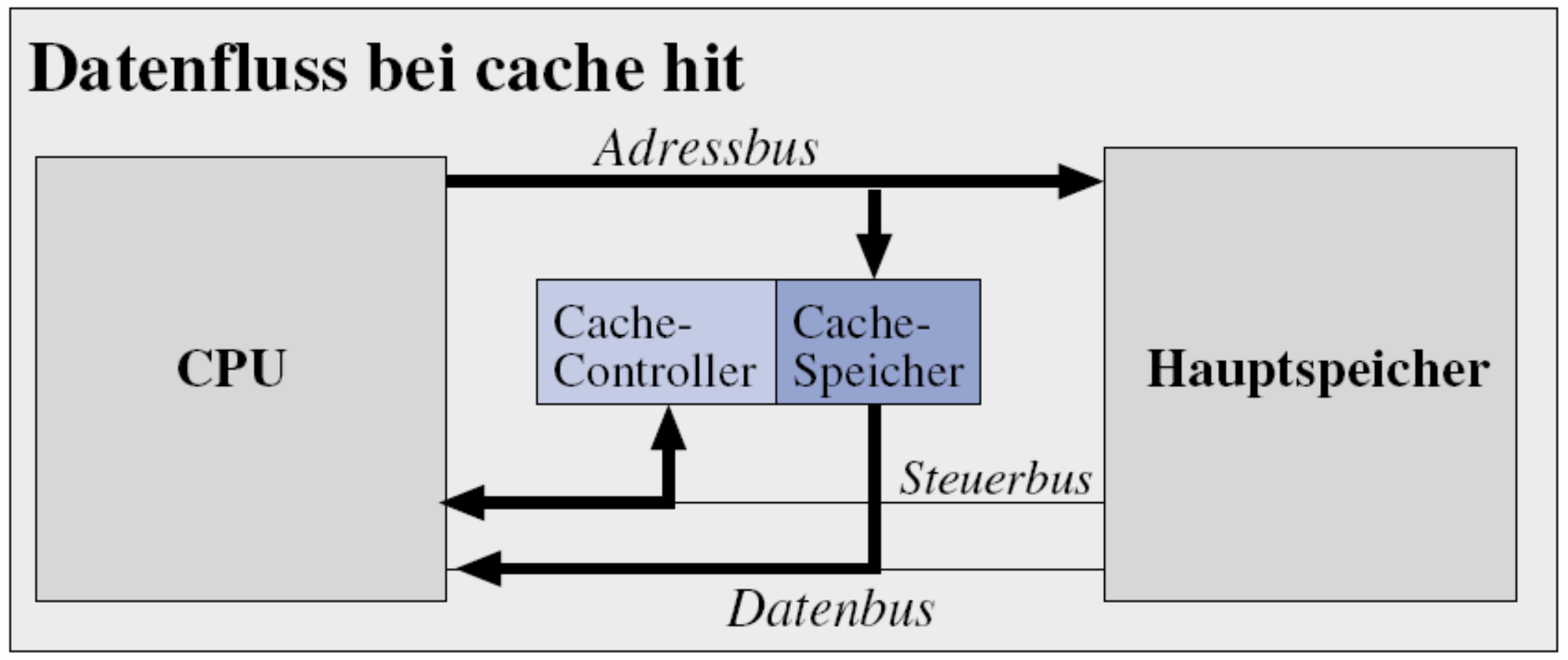
Cache Controller



© Pearson Studium 2006; Grundlagen der Informatik

Normalerweise gibt es in der CPU zwei Caches: Einer für Befehle (Instruction cache) und einer für Daten (Data cache).

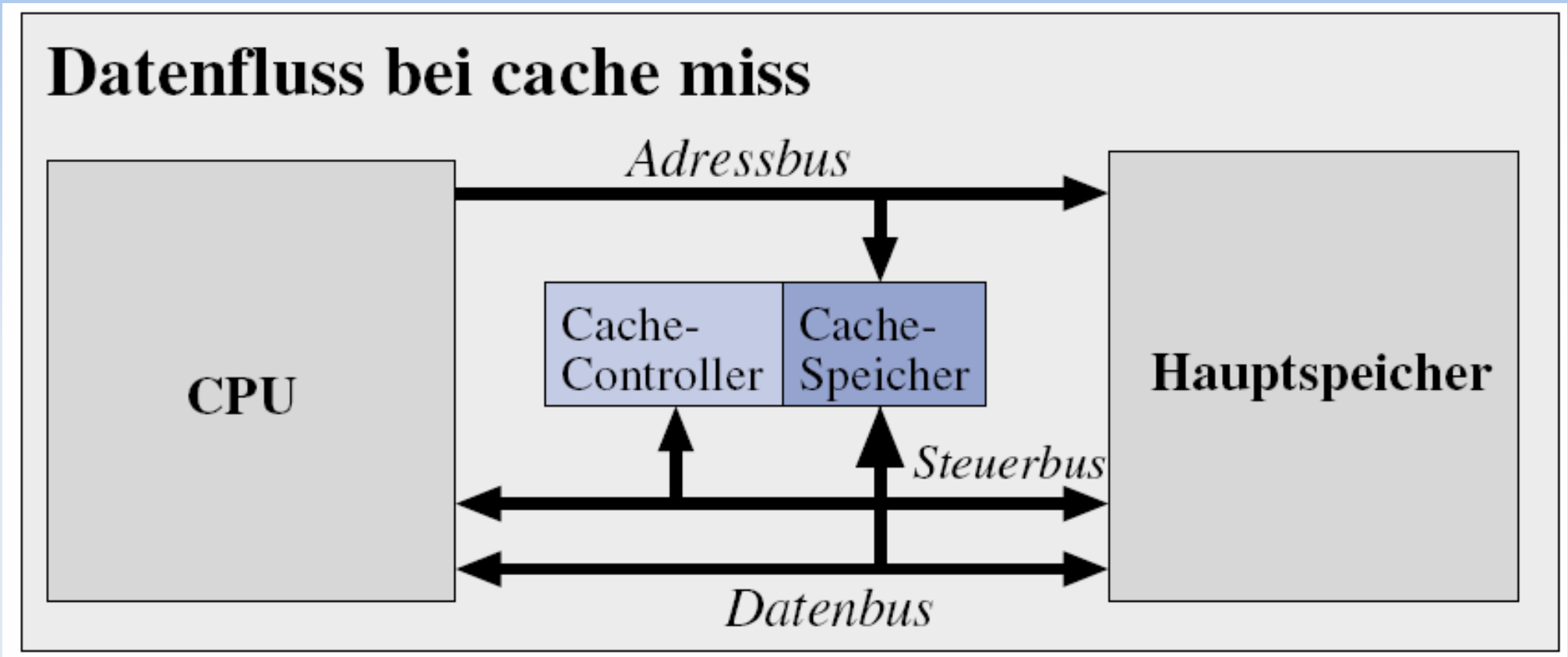
Cache hit - Lesezugriff



© Pearson Studium 2006; Grundlagen der Informatik

Datum (Kopie) ist bereits im Cache vorhanden.

Cache miss - Lesezugriff

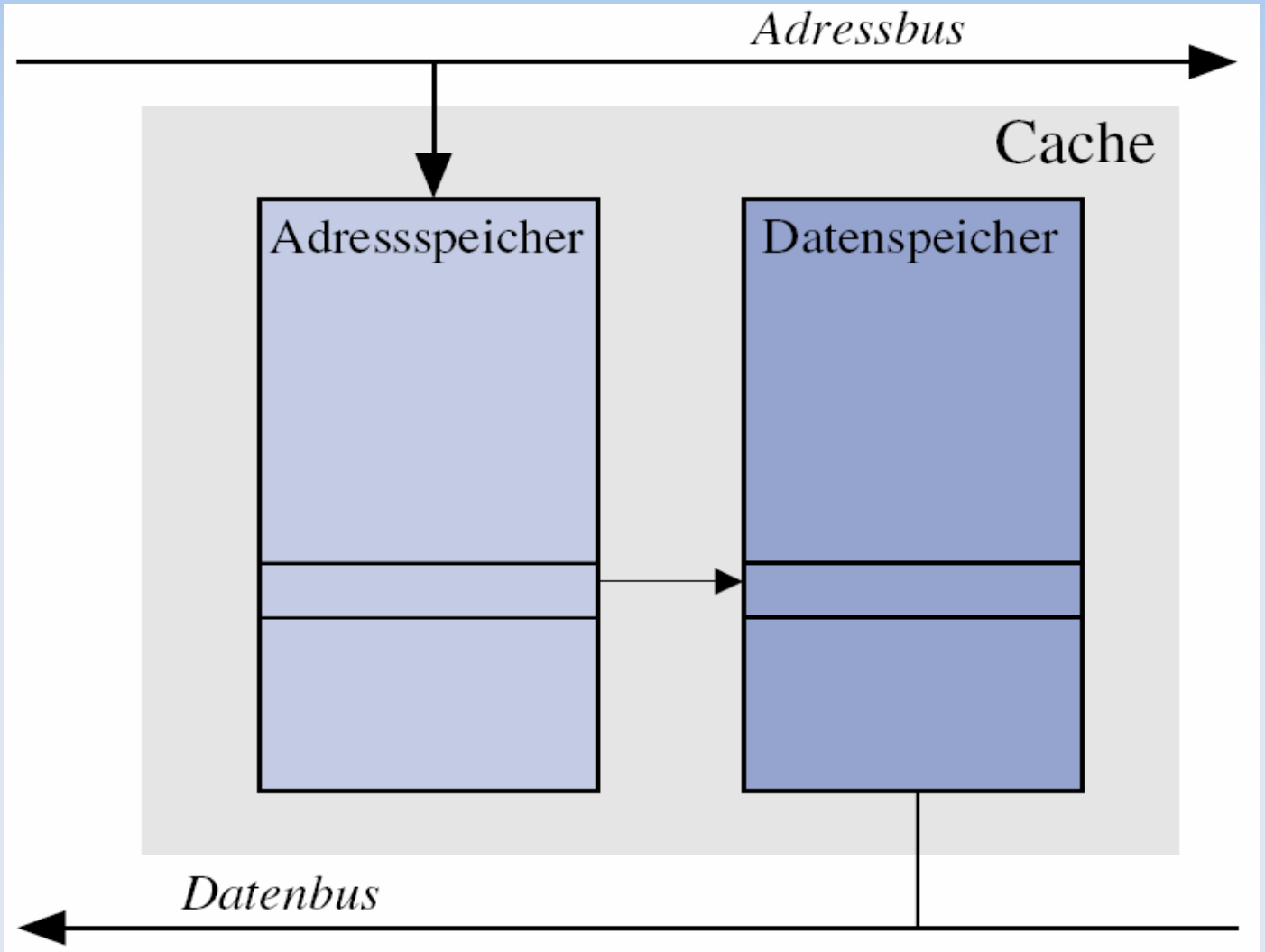


© Pearson Studium 2006; Grundlagen der Informatik

Datum (Kopie) ist nicht im Cache vorhanden.

Vollassoziative Caches

Der Cache enthält neben den Daten auch deren Hauptspeicher-Adresse



© Pearson Studium 2006; Grundlagen der Informatik

Vollassoziative Caches - Strategien

- Bei vollassoziativen Caches wird die gesuchte Adresse parallel mit allen Einträgen des Cache Adressspeichers verglichen
  - Erfordert zusätzlichen Schaltungsaufwand (teuer)
- Problem: Cache-miss bei vollassoziativen Cache, der bereits voll ist
  - Welches Datum im Cache durch das neue ersetzen?
  - Kein Datum verdrängen, das bald benötigt wird
- 3 Strategien denkbar
  - LRU (last recently used)
  - LFU (last frequently used)
  - FIFO (first in first out)

Fragen

Write through - Schreibzugriff

- Bei cache-hit wird sowohl die entsprechende Hauptspeicherzelle als auch deren Kopie im Cache sofort aktualisiert
- Bei cache-miss wird nur die entsprechende Hauptspeicherzelle aktualisiert und der Cache bleibt unverändert
- Daten-Kohärenz ist zu jedem Zeitpunkt gegeben
- Sehr zeitaufwändig, da in jedem Fall auf den Hauptspeicher zugegriffen werden muss

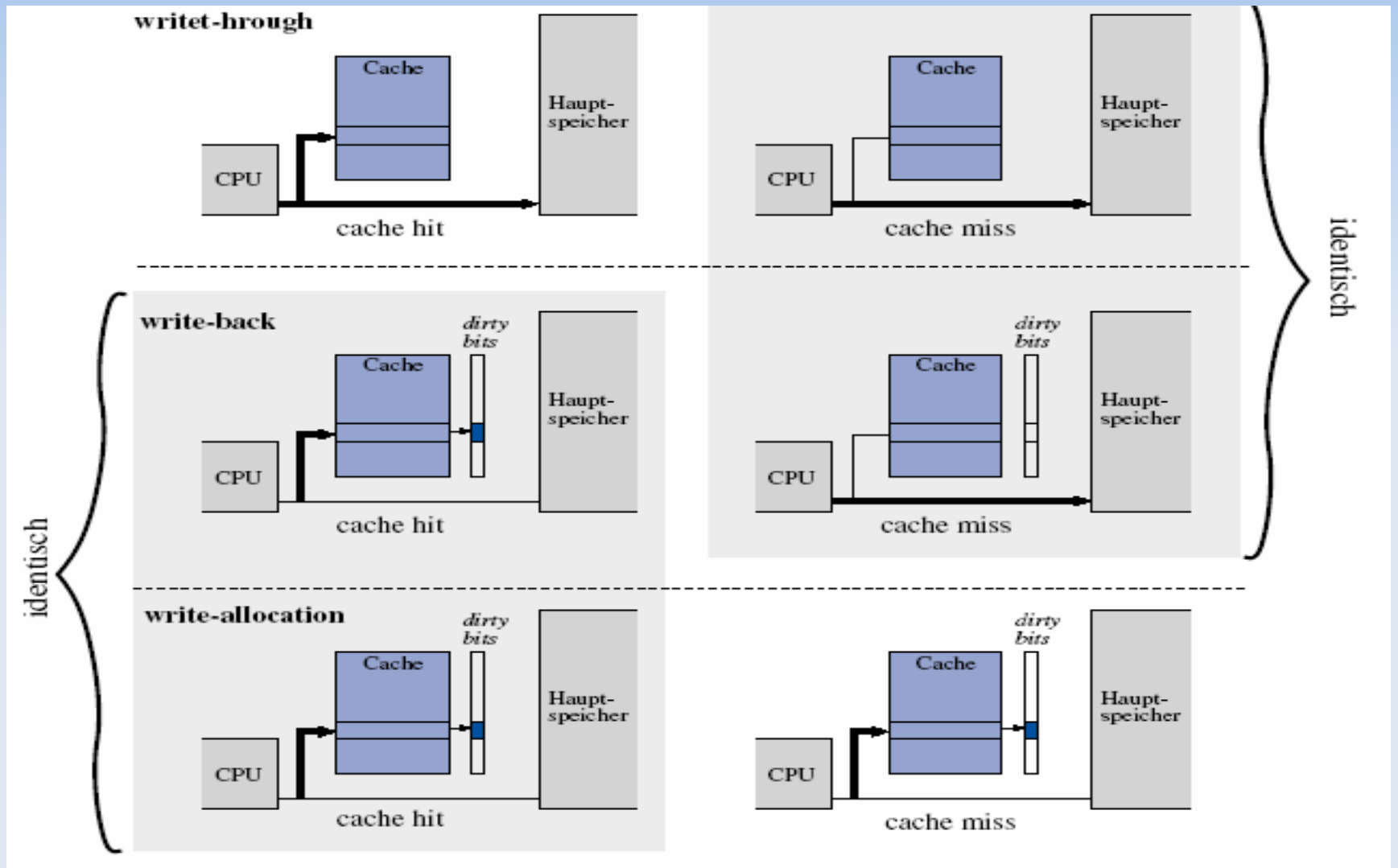
Write back - Schreibzugriff

- Bei cache-hit wird nur die Kopie im Cache aktualisiert, nicht aber die entsprechende Hauptspeicherzelle. Durch ein „dirty bit“ wird die Zelle als inkonsistent zum Hauptspeicher markiert
- Bei späterer Veränderung dieser Zelle oder Programmende werden alle Cache Zellen in den Hauptspeicher zurückgeschrieben, für die das „dirty bit“ gesetzt wurde
- Bei cache-miss wird nur die entsprechende Hauptspeicherzelle aktualisiert und der Cache bleibt unverändert

Write allocation - Schreibzugriff

- Bei cache-hit wird nur die Kopie im Cache aktualisiert, nicht aber die entsprechende Hauptspeicherzelle. Durch ein „dirty bit“ wird die Zelle als inkonsistent zum Hauptspeicher markiert
- Bei cache-miss wird das Datum nicht im Hauptspeicher sondern nur im Cache aktualisiert. Durch ein „dirty bit“ wird die Zelle als inkonsistent zum Hauptspeicher markiert
- Bei späterer Veränderung dieser Zelle oder Programmende werden alle Cache Zellen in den Hauptspeicher zurückgeschrieben, für die das „dirty bit“ gesetzt wurde

Gegenüberstellung



© Pearson Studium 2006; Grundlagen der Informatik

Vielen Dank für Ihre Aufmerksamkeit!