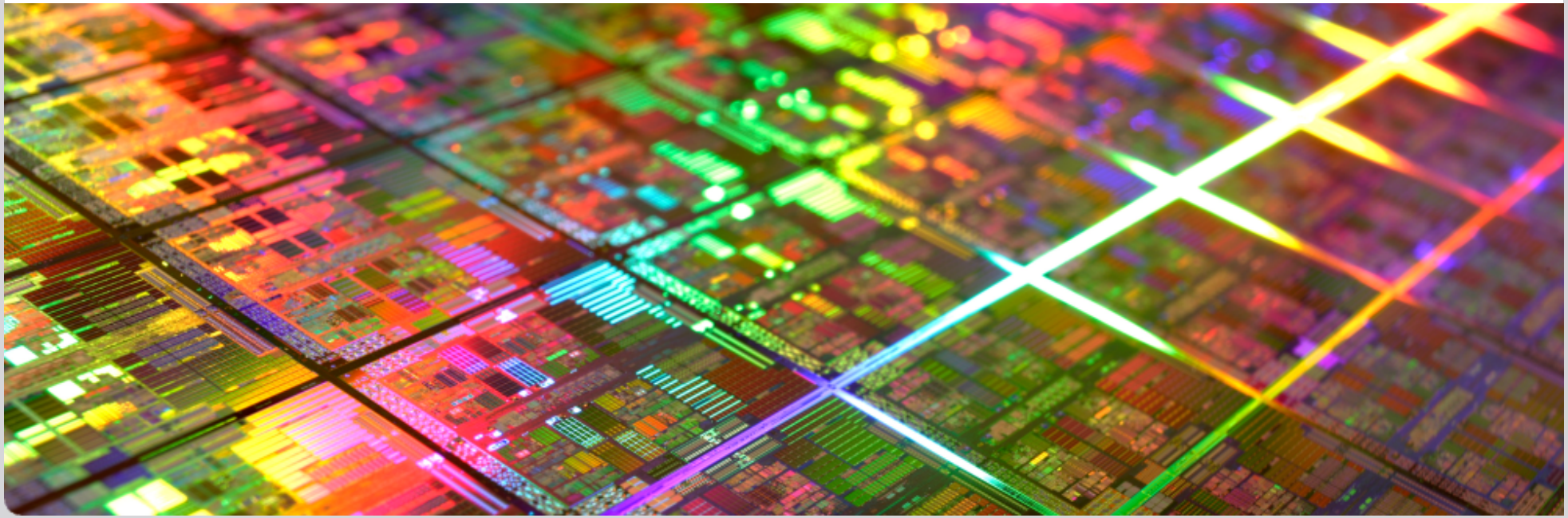
Karlsruher Institut für Technologie

Zentralübung Rechnerstrukturen im SS 2014
Cache-Kohärenz

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

15. Juli 2014



Überblick

Karlsruher Institut für Technologie

Inhalt der 7. Übung

- Motivation für Caches
- Cacheorganisation
- Eigenschaften von Caches
- Cachekoheränzprotokolle
  - MESI
  - MOESI
- DSM-Systeme
- Organisatorisches zur Klausur

Cache - Motivation

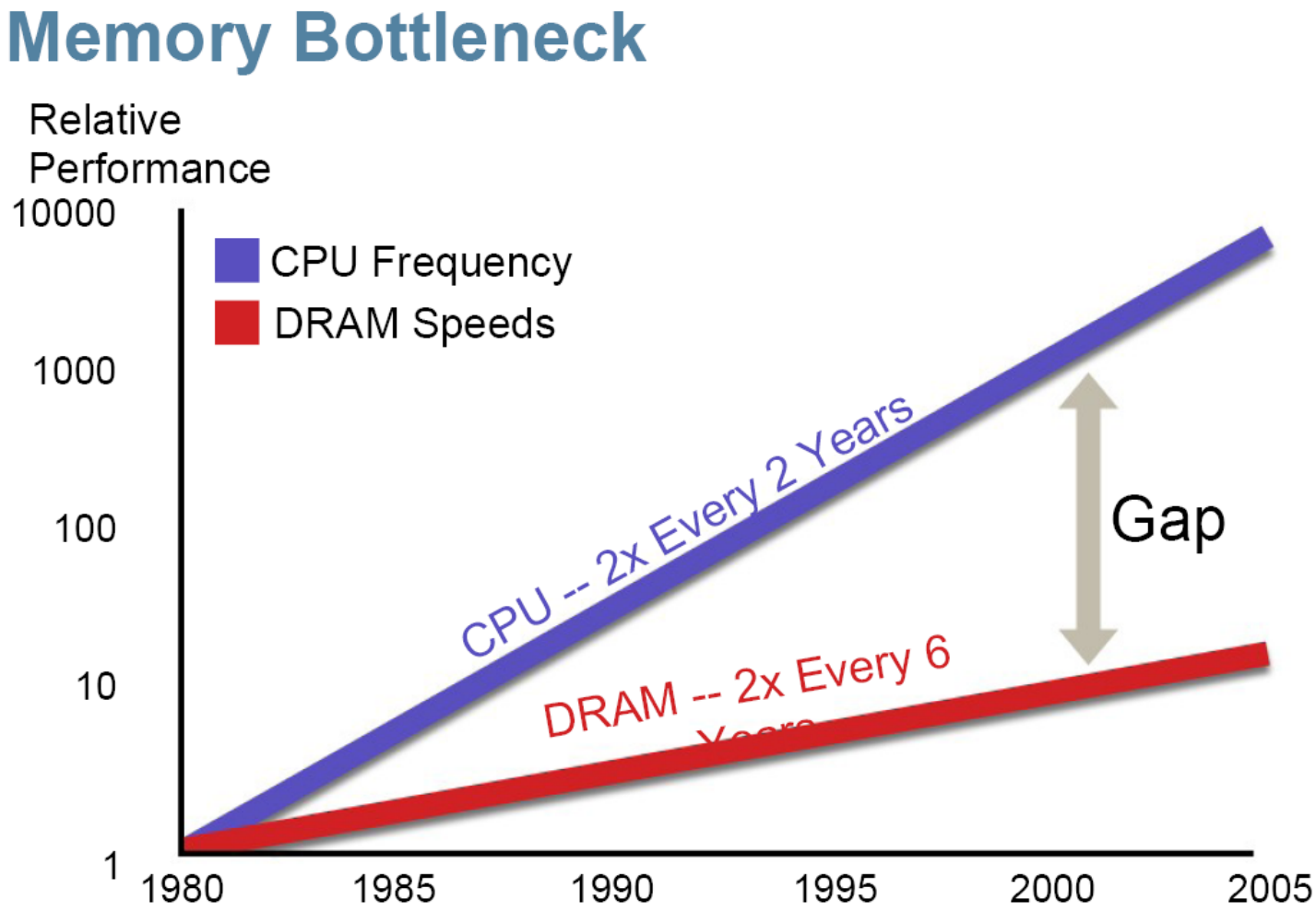


Abbildung : Quelle: http://blogs.sun.com/toddjobson/resource/

Cache - Motivation (forts.)

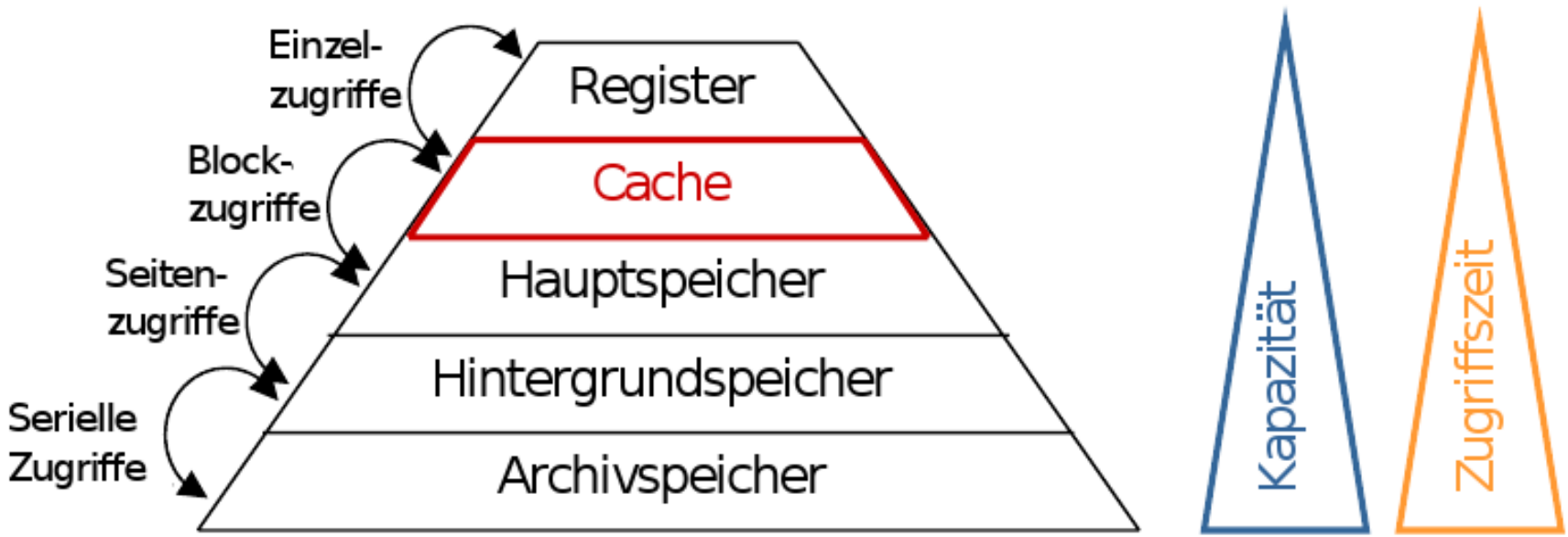


Abbildung : Speicherhierarchie

- Ausnutzung der räumlichen und zeitlichen Lokalität von Anwendungen
- 90/10 - Regel

Cache - Parameter

Organisation

- Direkt abgebildet (direct mapped)
- Satzassoziativ (set associative)
- Vollassoziativ (fully associative)

Hierarchie

- Mehrere Level (L1, L2, L3)
- im Multiprozessorfall: private vs. shared
- inclusive vs. exclusive

Größe

- Cacheline-Size
- # Cachelines bzw. Assoziativität * # Sätzen

Cache - Parameter (forts.)

Ersetzungstrategie

- least recently used (LRU)
- least frequently used (LFU)
- Round Robin
- Random

Schreibstrategie

- write-through vs. write-back
- write-allocation vs. no write-allocation

Cachekohärenzprotokoll (im Multiprozessorfall)

- Busbasiert
- Verzeichnisbasiert

Cache - Eigenschaften

- Bei Caches kann zwischen unterschiedlichen auftretenden Fehlzugriffen unterschieden werden

- Compulsory Miss
  - Miss, der durch ersten Zugriff auf ein Datum entsteht
  - Daten also noch nicht im Cache und daher Miss nicht vermeidbar

- Capacity Miss
  - Kapazität des Caches zu klein, weshalb bereits gecachtes Datum aus Cache entfernt werden musste

Cache - Eigenschaften (forts.)

- Conflict Miss
  - Set zu klein und bereits gecachtes Datum wurde aus Set entfernt (Cache nicht zwangsläufig voll)
  - Nur bei nicht-vollassoziativem Cache

- Coherency Miss
  - Nur bei Multiprozessor-Systemen mit Kohärenzprotokoll
  - Unterscheidung zwischen False- und True-Sharing
  - False-Sharing, falls nicht eigentliches Wort sondern anderes Wort in Cache-Block geändert wurde

Aufgabe 1.1: Cacheleistung

- Hit-Rate r_H
- Miss-Rate $r_M = 1 - r_H$
- Zugriffszeit bei Hit t_H
- Zugriffszeit Hauptspeicher t_{Mem} (falls Miss)

Mittlere Zugriffszeit

$$t_a = \underbrace{r_H * t_H}_{Hit} + \underbrace{r_M * t_{Mem}}_{Miss}$$

Mehrstufige Hierarchie: Konkretisierung des Miss-Zweigs

$$t_a = \underbrace{r_{H1} * t_{L1}}_{Hit\ L1} + \underbrace{r_{M1} * (\underbrace{r_{H2} * t_{L2}}_{Hit\ L2} + \underbrace{r_{M2} * t_{Mem}}_{Miss\ L2})}_{Miss\ L1}$$

Aufgabe 1.1: Cacheleistung

Bei dem Entwurf eines Systems stehen zwei Entwurfsalternativen zur Auswahl. In beiden Entwurfsalternativen kommt eine zwei-stufige Cache-Hierarchie zum Einsatz. Entwurfsalternative A hat ein kleinen L1-Cache mit einer Zugriffszeit von $t_{A-L1} = 10\,ns$, sowie einen L2-Cache mit einer Zugriffszeit von $t_{A-L2} = 30\,ns$. In Entwurfsalternative B ein größerer L1-Cache mit einer Zugriffszeit von $t_{B-L1} = 12\,ns$ zum Einsatz, sowie ein L2-Cache mit einer Zugriffszeit von $t_{B-L2} = 25\,ns$. Die Zugriffszeit des Hauptspeichers sei in beiden Entwurfsalternativen gleich und betrage $t_{Mem} = 100\,ns$.

a) Geben Sie eine Formel zur Berechung t_a der mittleren Zugriffszeit in einer zwei stufigen Cache-Hierarchie an.

Antwort: $t_a = \underbrace{r_{H1} * t_{L1}}_{Hit\ L1} + \underbrace{r_{M1} * (\underbrace{r_{H2} * t_{L2}}_{Hit\ L2} + \underbrace{r_{M2} * t_{Mem}}_{Miss\ L2})}_{Miss\ L1}$

Aufgabe 1.1: Cacheleistung

Bei dem Entwurf eines Systems stehen zwei Entwurfsalternativen zur Auswahl. In beiden Entwurfsalternativen kommt eine zwei-stufige Cache-Hierarchie zum Einsatz. Entwurfsalternative A hat ein kleinen L1-Cache mit einer Zugriffszeit von $t_{A-L1} = 10\,ns$, sowie einen L2-Cache mit einer Zugriffszeit von $t_{A-L2} = 30\,ns$. In Entwurfsalternative B ein größerer L1-Cache mit einer Zugriffszeit von $t_{B-L1} = 12\,ns$ zum Einsatz, sowie ein L2-Cache mit einer Zugriffszeit von $t_{B-L2} = 25\,ns$. Die Zugriffszeit des Hauptspeichers sei in beiden Entwurfsalternativen gleich und betrage $t_{Mem} = 100\,ns$.

a) Geben Sie eine Formel zur Berechung t_a der mittleren Zugriffszeit in einer zwei stufigen Cache-Hierarchie an.

Antwort: $t_a = \underbrace{r_{H1} * t_{L1}}_{Hit\ L1} + \underbrace{r_{M1} * (\underbrace{r_{H2} * t_{L2}}_{Hit\ L2} + \underbrace{r_{M2} * t_{Mem}}_{Miss\ L2})}_{Miss\ L1}$

Aufgabe 1.1: Cacheleistung

Karlsruher Institut für Technologie

Bei dem Entwurf eines Systems stehen zwei Entwurfsalternativen zur Auswahl. In beiden Entwurfsalternativen kommt eine zwei-stufige Cache-Hierarchie zum Einsatz. Entwurfsalternative A hat ein kleinen L1-Cache mit einer Zugriffszeit von $t_{A-L1} = 10\,ns$, sowie einen L2-Cache mit einer Zugriffszeit von $t_{A-L2} = 30\,ns$. In Entwurfsalternative B ein größerer L1-Cache mit einer Zugriffszeit von $t_{B-L1} = 12\,ns$ zum Einsatz, sowie ein L2-Cache mit einer Zugriffszeit von $t_{B-L2} = 25\,ns$. Die Zugriffszeit des Hauptspeichers sei in beiden Entwurfsalternativen gleich und betrage $t_{Mem} = 100\,ns$.

a) Geben Sie eine Formel zur Berechung t_a der mittleren Zugriffszeit in einer zwei stufigen Cache-Hierarchie an.

Antwort:

$$t_a = \underbrace{r_{H1} * t_{L1}}_{\text{Hit L1}} + \underbrace{r_{M1} * (\underbrace{r_{H2} * t_{L2}}_{\text{Hit L2}} + \underbrace{r_{M2} * t_{Mem}}_{\text{Miss L2}})}_{\text{Miss L1}}$$

Aufgabe 1.1: Cacheleistung

Karlsruher Institut für Technologie

$t_{A-L1} = 10\,ns$, $t_{A-L2} = 30\,ns$, $t_{B-L1} = 12\,ns$, $t_{B-L2} = 25\,ns$, $t_{Mem} = 100\,ns$.

Bei der Evaluation wurden folgende Hit-Raten gemessen:

- Alternative A: $r_{A-L1} = 70\,\%$, sowie $r_{A-L2} = 40\,\%$
- Alternative B: $r_{B-L1} = 75\,\%$, sowie $r_{B-L2} = 35\,\%$

Alternative A

$$t_a = 70\% * 10\,ns + (1 - 70\%) * (40\% * 30\,ns + (1 - 40\%) * 100\,ns)$$
$$t_a = 7\,ns + 0,3 * (12\,ns + 60\,ns) = 28,6\,ns$$

Alternative B

$$t_a = 75\% * 12\,ns + (1 - 75\%) * (35\% * 25\,ns + (1 - 35\%) * 100ns)$$
$$t_a = 9\,ns + 0,25 * (8,75\,ns + 65\,ns) = 27,4375\,ns$$

Die durchschnittliche Zugriffszeit in **Entwurfsalternative B** ist geringer, somit ist diese Entwurfsalternative zu wählen.

Aufgabe 1.1: Cacheleistung

Karlsruher Institut für Technologie

$t_{A-L1} = 10\,ns$, $t_{A-L2} = 30\,ns$, $t_{B-L1} = 12\,ns$, $t_{B-L2} = 25\,ns$, $t_{Mem} = 100\,ns$.

Bei der Evaluation wurden folgende Hit-Raten gemessen:

- Alternative A: $r_{A-L1} = 70\,\%$, sowie $r_{A-L2} = 40\,\%$
- Alternative B: $r_{B-L1} = 75\,\%$, sowie $r_{B-L2} = 35\,\%$

Alternative A

$$t_a = 70\% * 10\,ns + (1 - 70\%) * (40\% * 30\,ns + (1 - 40\%) * 100\,ns)$$
$$t_a = 7\,ns + 0,3 * (12\,ns + 60\,ns) = 28,6\,ns$$

Alternative B

$$t_a = 75\% * 12\,ns + (1 - 75\%) * (35\% * 25\,ns + (1 - 35\%) * 100ns)$$
$$t_a = 9\,ns + 0,25 * (8,75\,ns + 65\,ns) = 27,4375\,ns$$

Die durchschnittliche Zugriffszeit in **Entwurfsalternative B** ist geringer, somit ist diese Entwurfsalternative zu wählen.

Aufgabe 1.1: Cacheleistung

Karlsruher Institut für Technologie

$t_{A-L1} = 10\,ns$, $t_{A-L2} = 30\,ns$, $t_{B-L1} = 12\,ns$, $t_{B-L2} = 25\,ns$, $t_{Mem} = 100\,ns$.

Bei der Evaluation wurden folgende Hit-Raten gemessen:

- Alternative A: $r_{A-L1} = 70\,\%$, sowie $r_{A-L2} = 40\,\%$
- Alternative B: $r_{B-L1} = 75\,\%$, sowie $r_{B-L2} = 35\,\%$

Alternative A

$$t_a = 70\% * 10\,ns + (1 - 70\%) * (40\% * 30\,ns + (1 - 40\%) * 100\,ns)$$
$$t_a = 7\,ns + 0,3 * (12\,ns + 60\,ns) = 28,6\,ns$$

Alternative B

$$t_a = 75\% * 12\,ns + (1 - 75\%) * (35\% * 25\,ns + (1 - 35\%) * 100ns)$$
$$t_a = 9\,ns + 0,25 * (8,75\,ns + 65\,ns) = 27,4375\,ns$$

Die durchschnittliche Zugriffszeit in **Entwurfsalternative B** ist geringer, somit ist diese Entwurfsalternative zu wählen.

Aufgabe 1.2: Beweise

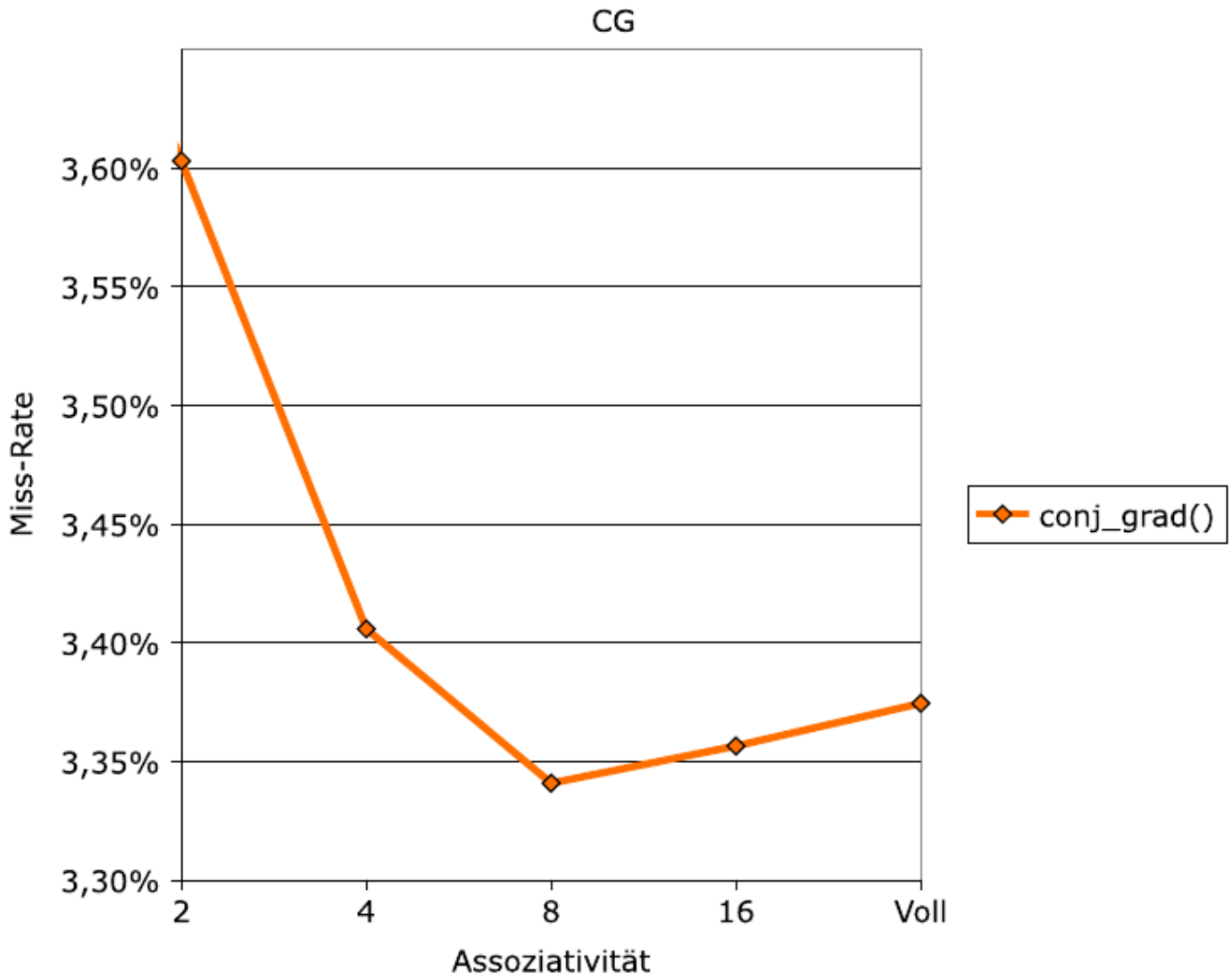
Karlsruher Institut für Technologie

Beweisen oder widerlegen Sie folgende Behauptungen:
Hinweis: Gehen Sie in ihren Überlegungen davon aus, dass die Caches gemäß Least-Recently-Used (LRU)-Strategie verdrängen.

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Miss-Rate.

Aufgabe 1.2: Beweise - Motivation



Aufgabe 1.2: Beweise

Karlsruher Institut für Technologie

Beweisen oder widerlegen Sie folgende Behauptungen:
Hinweis: Gehen Sie in ihren Überlegungen davon aus, dass die Caches gemäß Least-Recently-Used (LRU)-Strategie verdrängen.

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Miss-Rate.

Widerlegung:

Finden einer Folge von Cache-Belegungen bei der Aussage falsch

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

4-fach assoziativ

Folge:

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

A	A
B	B

C
D

4-fach assoziativ

A	A
B	B
C	
D	

Folge: *ABCDAB*

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

A	A	E
B	B	F

C
D

4-fach assoziativ

A	A
B	B
C	E
D	F

Folge: *ABCDABEF*

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

A	A	E
B	B	F

C	C	
D	D	

4-fach assoziativ

A	A	C
B	B	D
C	E	
D	F	

Folge: *ABCDABEFCD*

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

A	A	E	A	E
B	B	F	B	F

C	C
D	D

4-fach assoziativ

A	A	C	E
B	B	D	F
C	E	A	
D	F	B	

Folge: *ABCDABEFCDABEF*

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

2-fach assoziativ

A	A	E	A	E
B	B	F	B	F

C	C	C		
D	D	D		

4-fach assoziativ

A	A	C	E
B	B	D	F
C	E	A	C
D	F	B	D

Folge: $ABCD(ABEFCD)^n$

Voraussetzung: Speicherzugriffe A, B, E, F werden auf den ersten Satz abgebildet
C und D auf den zweiten Satz des 2-fach assoziativen Cache

Aufgabe 1.2: Beweise (forts.)

Karlsruher Institut für Technologie

Beweisen oder widerlegen Sie folgende Behauptungen:
Hinweis: Gehen Sie in ihren Überlegungen davon aus, dass die Caches gemäß Least-Recently-Used (LRU)-Strategie verdrängen.

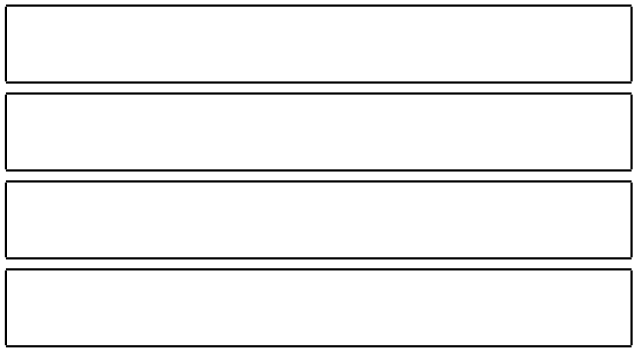
a) Eine Erhöhung der Assoziativität eines Caches zieht immer eine Verringerung der Miss-Rate nach sich.

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Miss-Rate.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped



Vollassoziativ

Folge:

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped

A
B
C
D

Vollassoziativ

A
B
C
D

Folge: *ABCD*

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped

A
B
C
D H

Vollassoziativ

A	H
B	
C	
D	

Folge: *ABCDH*

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped

A	A
B	B
C	C
D	H

Vollassoziativ

A	H
B	A
C	B
D	C

Folge: *ABCDHABC*

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped

A	A
B	B
C	C
D	H D H

Vollassoziativ

A	H	D
B	A	H
C	B	
D	C	

Folge: $(ABCDH)^n$

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.2: Beweise (forts.)

b) Vollassoziative Caches haben satzassoziativen Caches gegenüber immer eine niedrigere Missrate.

Direct Mapped

A	A	A	
B	B	B	
C	C	C	
D	H	D	H

Vollassoziativ

A	H	D	C
B	A	H	
C	B	A	
D	C	B	

Folge: $(ABCDH)^n$

Voraussetzung: A wird auf die erste, B auf die zweite, C auf die dritte und D und H auf die vierte Cachezeile des DM-Cache abgebildet.

Aufgabe 1.3 - Verständnisfragen

a) Welche Eigenschaft von Anwendungen werden von Caches ausgenutzt?

Antworten:

a) Räumliche und Zeitliche Lokalität

Aufgabe 1.3 - Verständnisfragen

a) Welche Eigenschaft von Anwendungen werden von Caches ausgenutzt?

Antworten:

a) Räumliche und Zeitliche Lokalität

Aufgabe 1.3 - Verständnisfragen (forts.)

b) Welche Arten von Cache-Misses können unterschieden werden?

Antworten:

b) Conflict-, Capacity-, Compulsory-Misses

Im Mehrprozessorfall:

- Coherency-Miss
- Unterscheidung in: False-Sharing-Miss bzw. True-Sharing-Miss

Aufgabe 1.3 - Verständnisfragen (forts.)

Karlsruher Institut für Technologie

b) Welche Arten von Cache-Misses können unterschieden werden?

Antworten:

b) Conflict-, Capacity-, Compulsory-Misses

Im Mehrprozessorfall:

- Coherency-Miss
- Unterscheidung in: False-Sharing-Miss bzw. True-Sharing-Miss

Aufgabe 1.3 c) - Motivation

- Cache-Speicher zeichnet sich durch seine schnellen Zugriffszeiten Ausblick

⇒ Verwendet daher im Gegensatz zum Hauptspeicher static RAM Speicherbausteine

- SRAM verbraucht deutlich mehr Chipfläche als Speicherbausteine des Hauptspeichers

⇒ Deutlich geringere Datenkapazität bei gleichem Flächenverbrauch

⇒ Deutlich höhere Herstellungskosten pro Bit Speicher

Aufgabe 1.3 - Verständnisfragen (forts.)

Karlsruher Institut für Technologie

c) Warum wird der Cache-Speicher nicht deutlich größer (Größenbereich 100 MB - mehrere GB) angelegt?

Antworten:

- Benötigt deutlich größeren Die, wodurch der Yield bei der Herstellung sinkt
  - Herstellungskosten steigen
- Für Desktop-PCs auf Grund der erhöhten Herstellungskosten zu teuer
  - Kein ausreichendes Argument für Nicht-Anwendung
- SRAM Bausteine benötigen große Fläche für den angestrebten Größenbereich
  - Signale müssen Fläche durchlaufen, also deutlich längere Signalwege

  $\Rightarrow$ Signale brauchen länger und Zugriffszeit erhöht sich

  $\Rightarrow$ Vorteil von SRAM geht verloren

Aufgabe 1.3 - Verständnisfragen (forts.)

c) Warum wird der Cache-Speicher nicht deutlich größer (Größenbereich 100 MB - mehrere GB) angelegt?

Antworten:

- Benötigt deutlich größeren Die, wodurch der Yield bei der Herstellung sinkt
  - Herstellungskosten steigen
- Für Desktop-PCs auf Grund der erhöhten Herstellungskosten zu teuer
  - Kein ausreichendes Argument für Nicht-Anwendung
- SRAM Bausteine benötigen große Fläche für den angestrebten Größenbereich
  - Signale müssen Fläche durchlaufen, also deutlich längere Signalwege

  $\Rightarrow$ Signale brauchen länger und Zugriffszeit erhöht sich

  $\Rightarrow$ Vorteil von SRAM geht verloren

Caches in Multiprozessorsystemen

Karlsruher Institut für Technologie

Problem bei Verwendung von Caches in MP-Systemen

- Wahrung der Konsistenz
- CPUs operieren auf lokalen Kopien
- Daten in den Caches können sich von den Daten im Hauptspeicher unterscheiden

Kohärenzprotokolle

Bus-Snooping-basiert

- Write-Invalidate Protokoll
  - arbeitet auf Cacheline-Ebene
  - benötigt nur ein Invalidate
  - Vertreter
    - MSI
    - MESI
    - MOESI
- Write-Update Protokoll
  - arbeitet auf Wort-Ebene
  - u. U. mehrere Update-Broadcasts nötig

Verzeichnisbasiert

Wird in DSM-Systemen benötigt.

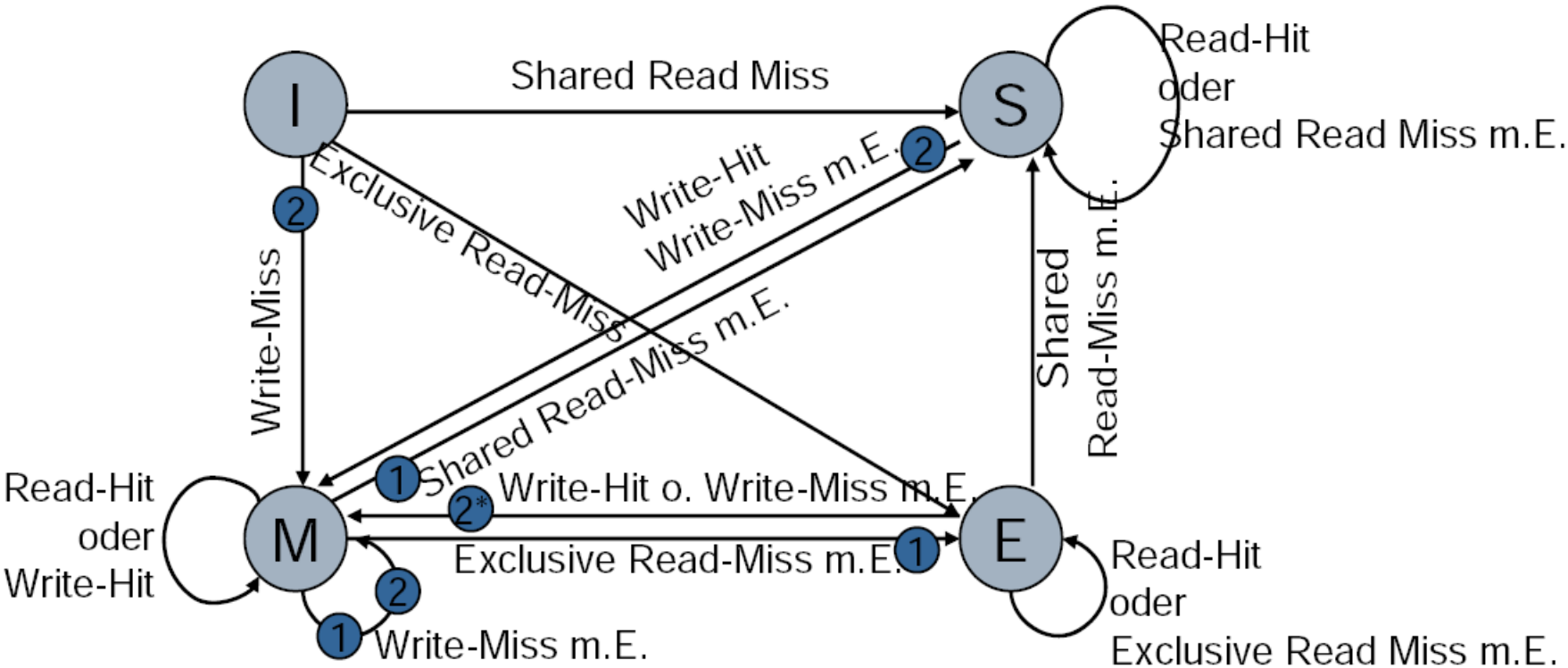
Bus-Snooping

- Beobachtung des Speicherbusses hinsichtlich Speicherzugriffe anderer Bus-Master
- Bus-Snooping erfordert einen **globalen Speicherbus**
- Jeder Cache muss um sog. **Snoop-Logik** und **Steuersignale** zu anderen Caches erweitert werden.
  - Invalidate-Signal
  - Shared-Signal
  - Retry-Signal

Das MESI-Protokoll

- Write-Invalidate Protokoll
- MESI: Zustandsautomat mit 4 Zuständen
  - M: exclusive Modified
  - E: Exclusive unmodified
  - S: Shared unmodified
  - I: Invalid
- Jede Cachezeile befindet sich in einem dieser Zustände
- d.h. jede Cachezeile wird um **2 Zustandsbits** erweitert
- Zustandsübergänge werden durch lokale und entfernte Speicherzugriffe ausgelöst

MESI-Zustandsautomat (1)



m.E.: mit Ersetzung

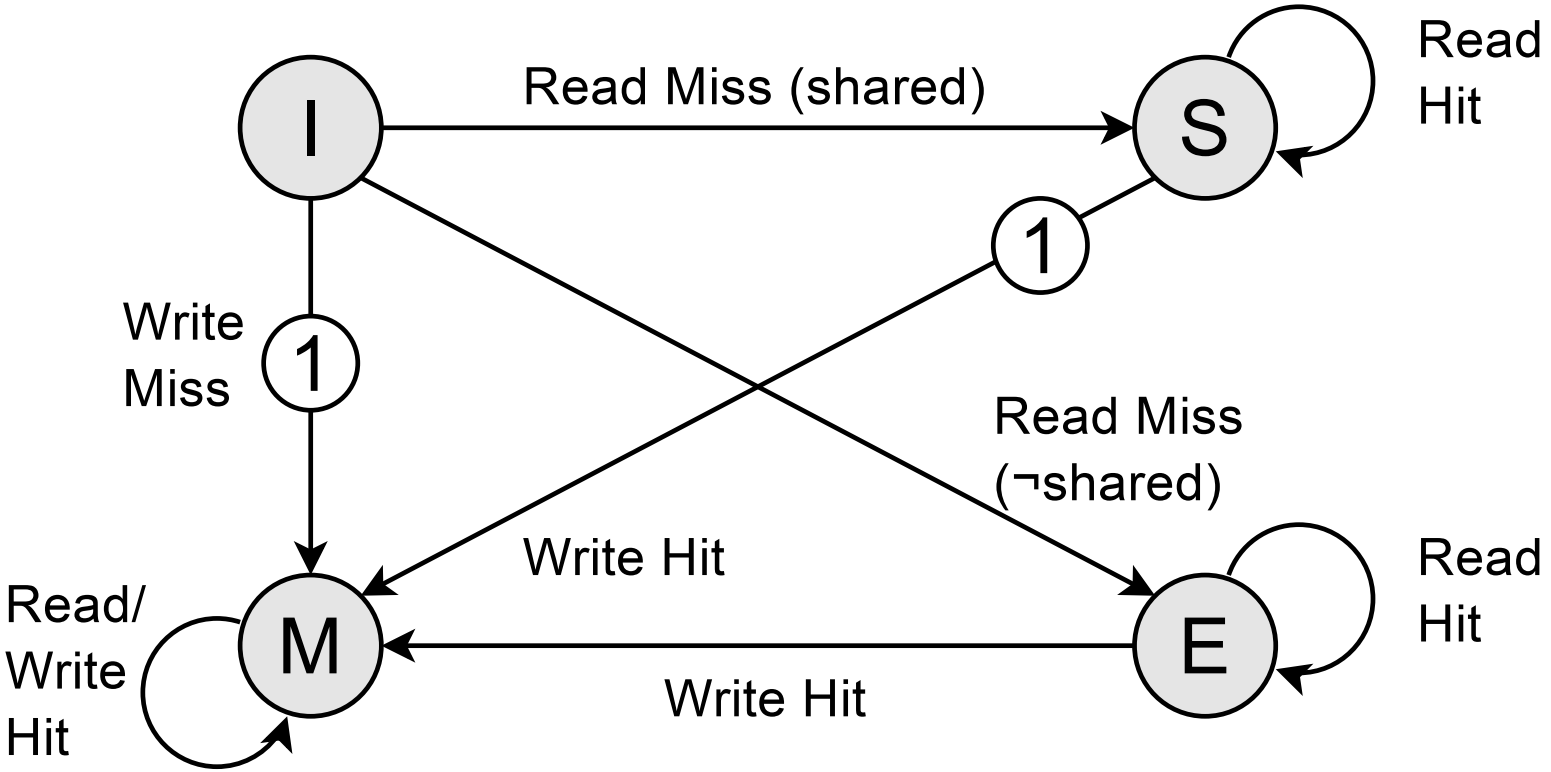
1 Cache-Zeile wird in den Hauptspeicher zurückkopiert (Line-Flush)

2 Cache-Zeilen in den anderen Caches mit gleicher Blockadresse werden invalidiert

2* Wie 2, gilt jedoch nur für Write-Miss mit Ersetzung

MESI-Zustandsautomat (2)

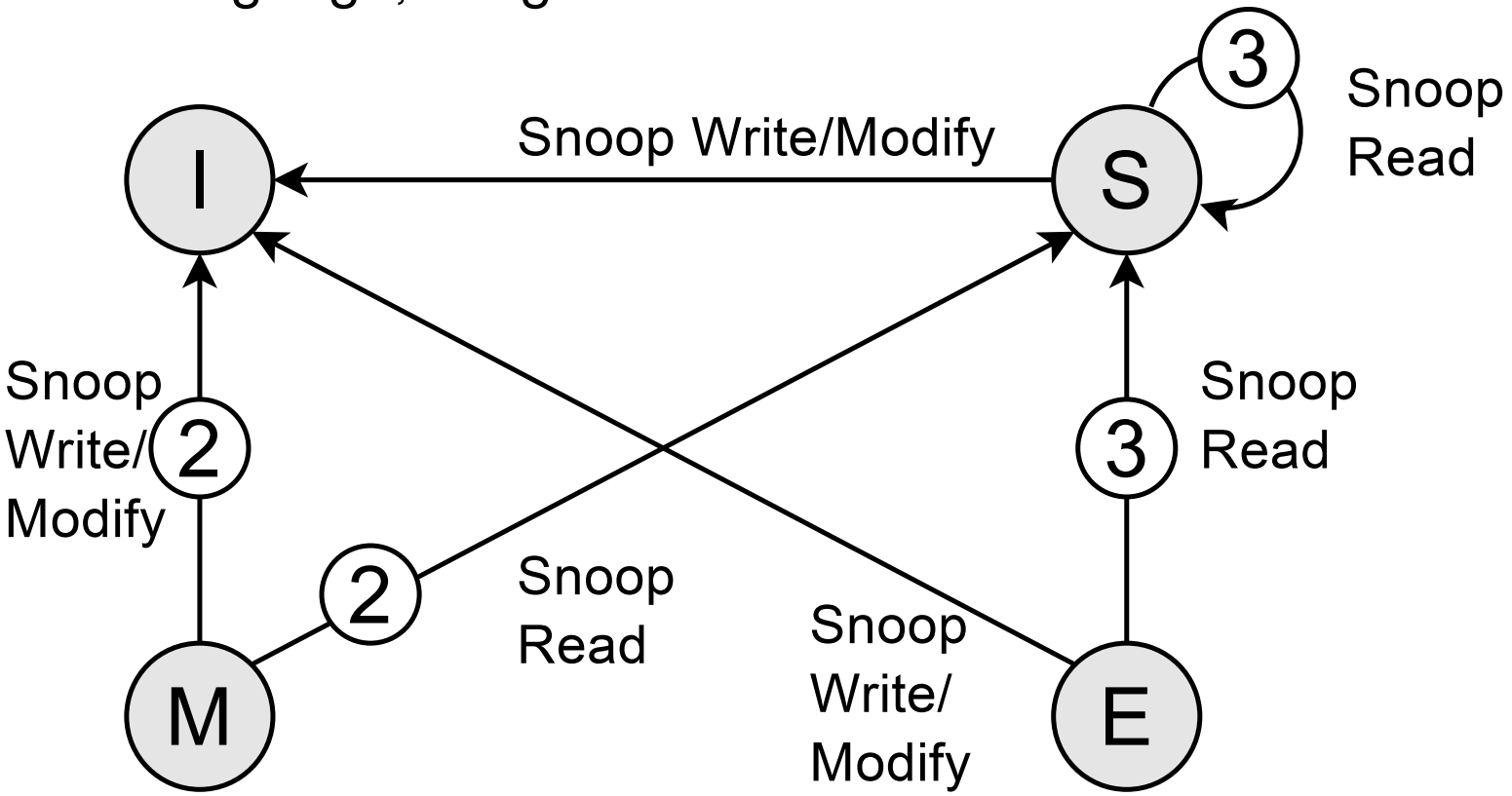
Zustandübergänge, ausgelöst durch eigene Aktionen:



1 Cache-Zeilen in den anderen Caches mit gleicher Blockadresse werden invalidiert

MESI-Zustandsautomat (3)

Zustandsübergänge, ausgelöst durch andere Prozessoren:



2 Retry-Signal wird aktiviert, Cache-Zeile in den Hauptspeicher kopiert, dann Signal deaktiviert

3 Shared-Signal wird aktiviert

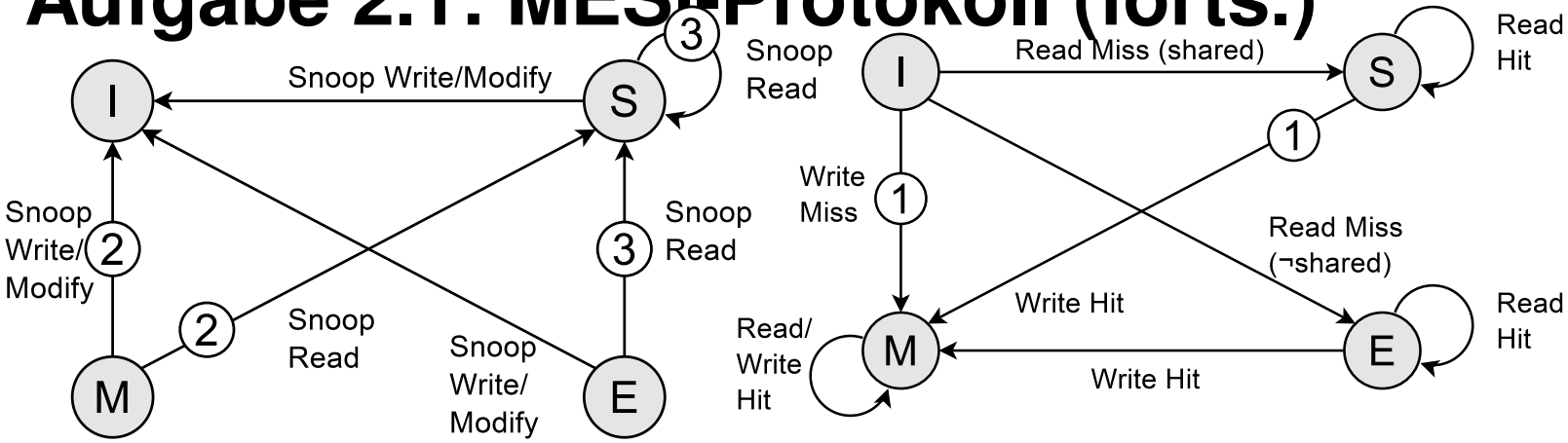
Aufgabe 2.1: MESI-Protokoll

Karlsruher Institut für Technologie

Ein Dreiprozessorsystem sei speichergekoppelt. Die Caches haben je eine Größe von zwei Cachezeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung des Caches erfolgt von der niedrigsten Cachezeile aufwärts, sofern noch freie Zeilen zur Verfügung stehen, anderenfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll komme das MESI-Protokoll zum Einsatz.

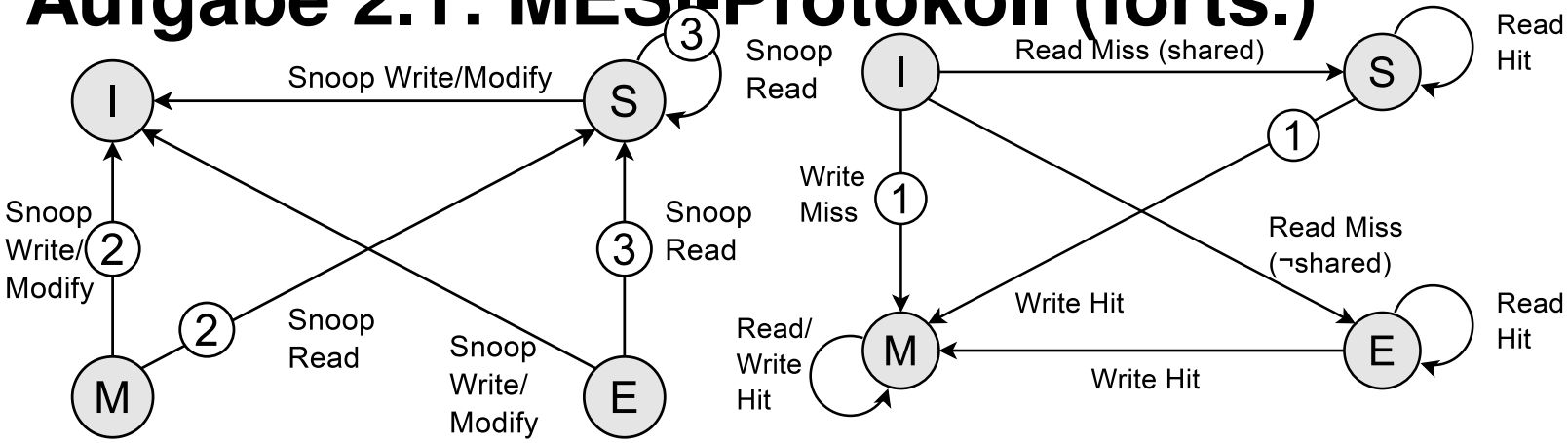
- Vervollständigen Sie die gegebene Tabelle: Geben Sie jeweils Inhalt der Cache-Zeile und MESI-Zustand an.

Aufgabe 2.1: MESI-Protokoll (forts.)



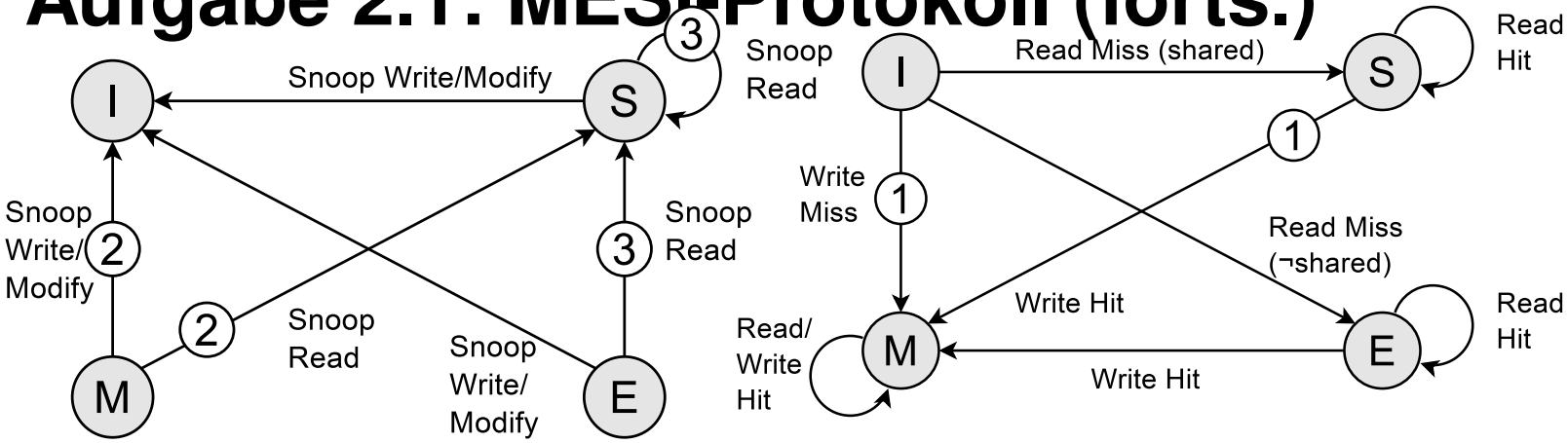
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6						
2	rd 2						
1	rd 4						
3	rd 4						
2	rd 3						
3	wr 7						
1	wr 4						
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



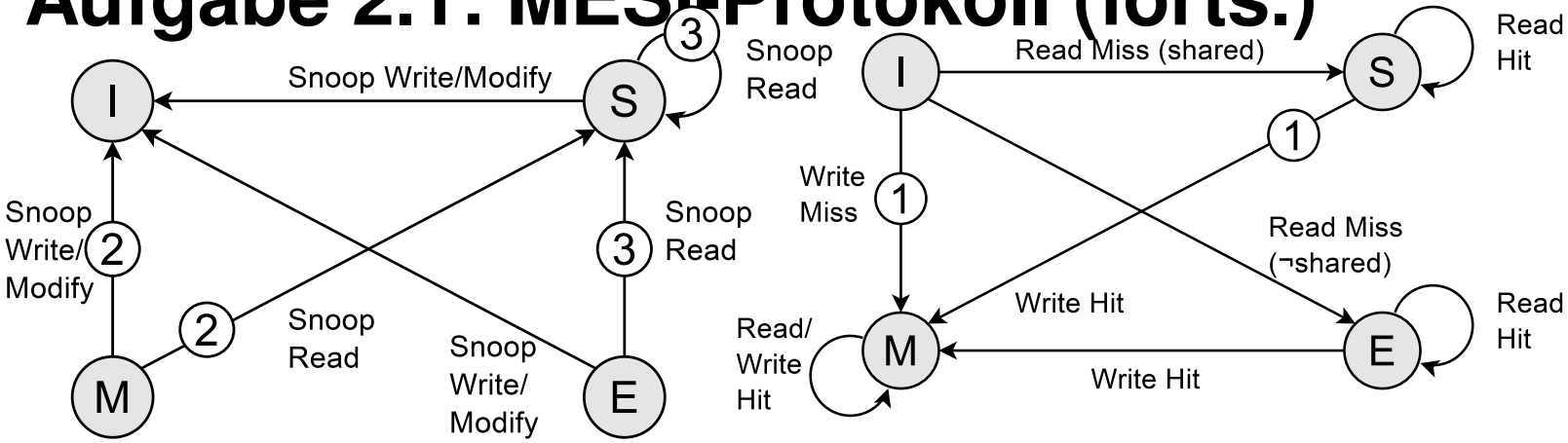
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4						
2	rd 3						
3	wr 7						
1	wr 4						
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



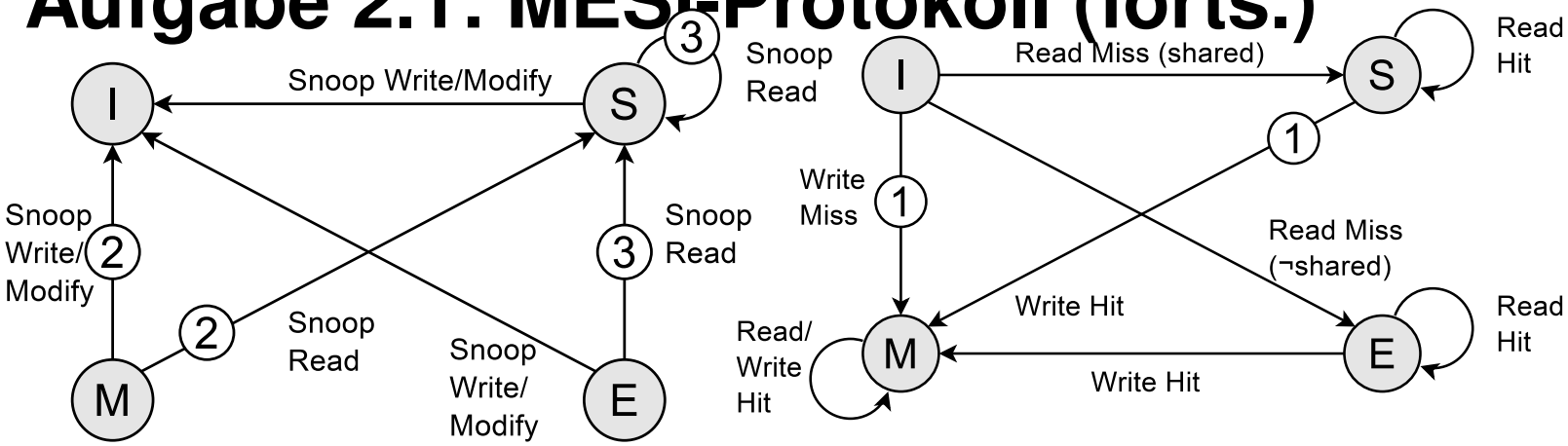
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3						
3	wr 7						
1	wr 4						
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



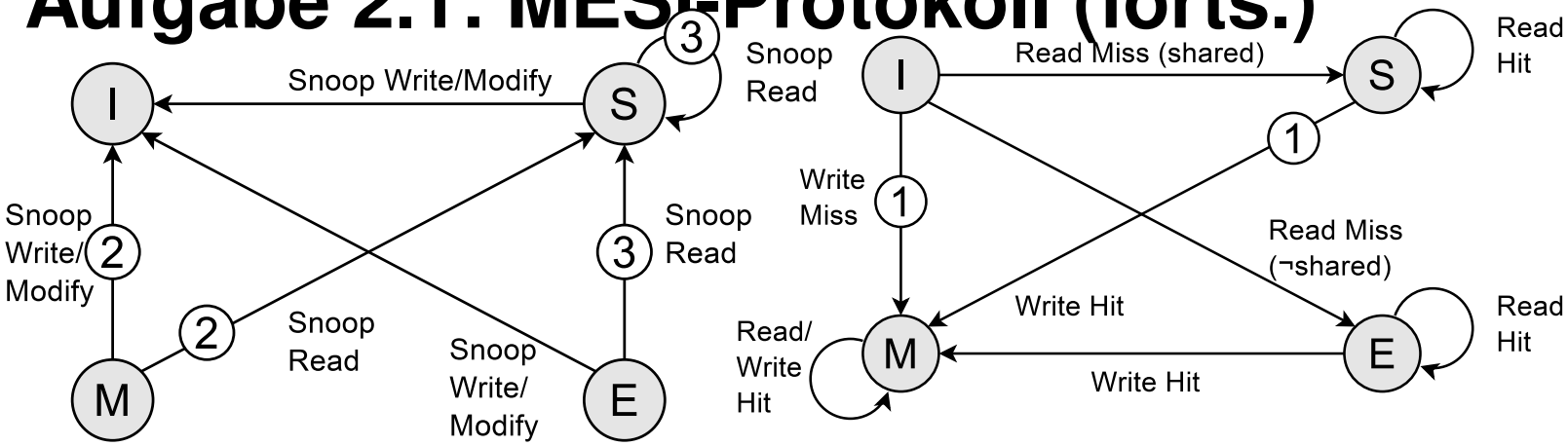
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4						
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



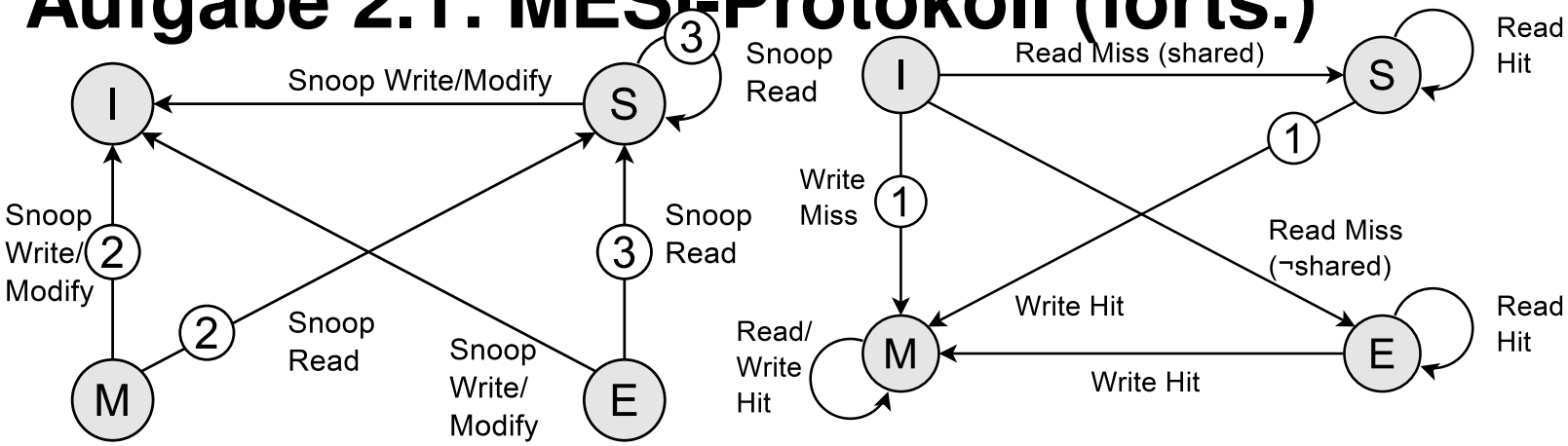
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7						
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



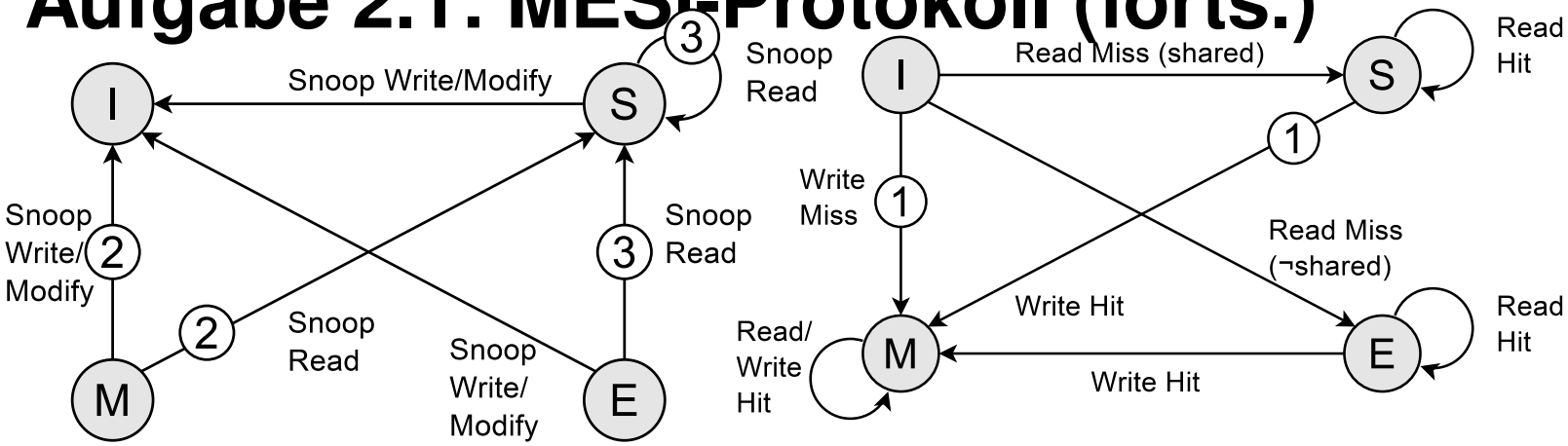
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7			7/S			7/S
3	wr 5						
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



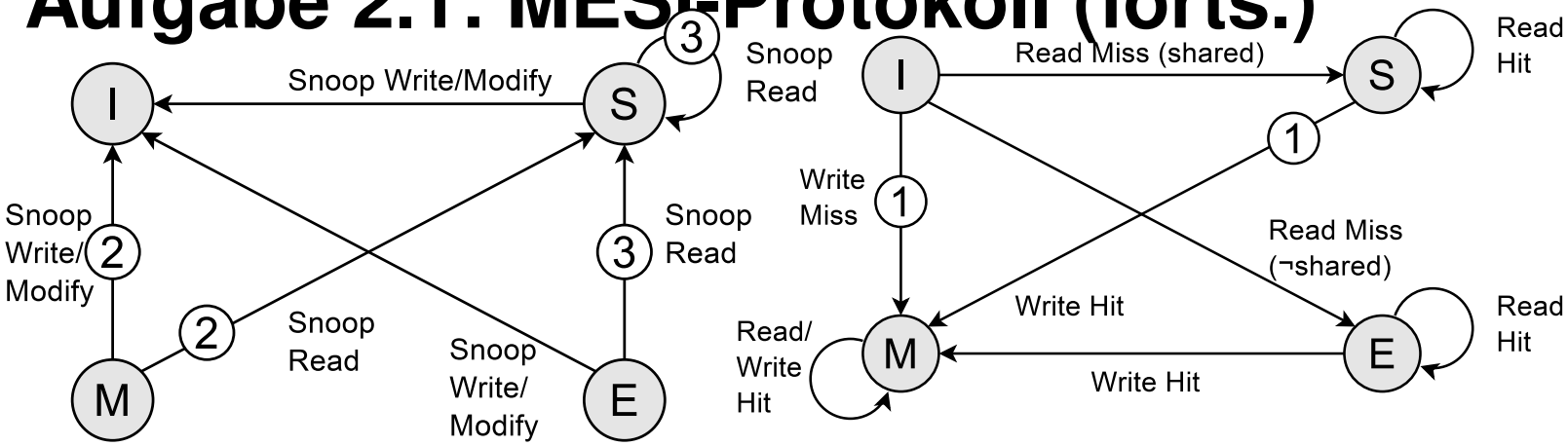
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7			7/S			7/S
3	wr 5					5/M	
1	rd 3						
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



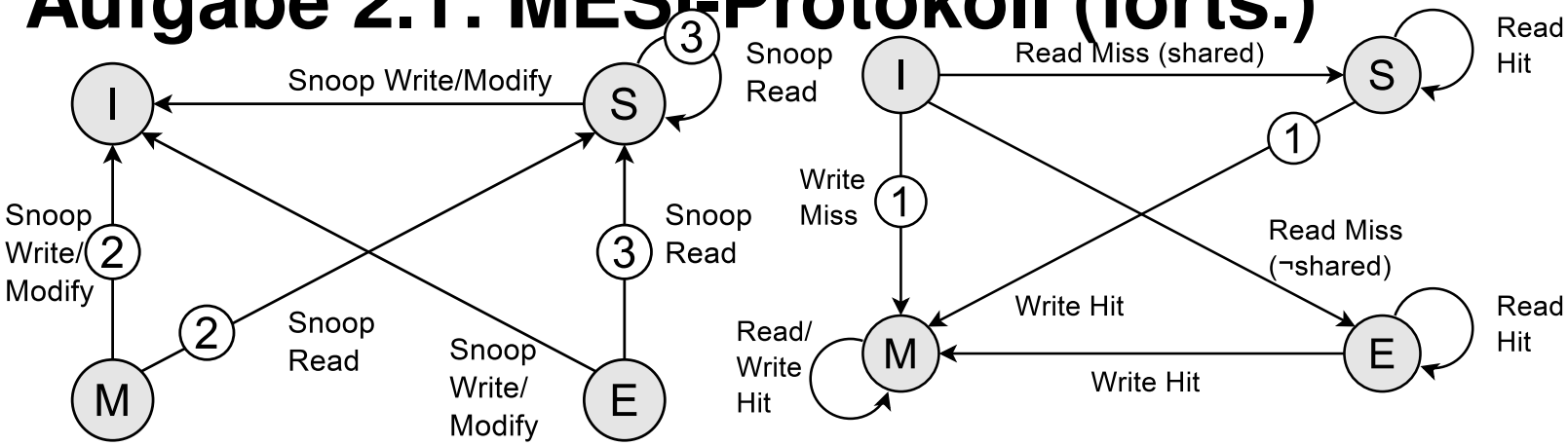
Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7			7/S			7/S
3	wr 5					5/M	
1	rd 3	3/S			3/S		
3	wr 3						
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7			7/S			7/S
3	wr 5					5/M	
1	rd 3	3/S			3/S		
3	wr 3	3/I			3/I		3/M
2	wr 7						

Aufgabe 2.1: MESI-Protokoll (forts.)



Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 6	6/E					
2	rd 2			2/E			
1	rd 4		4/E				
3	rd 4		4/S			4/S	
2	rd 3				3/E		
3	wr 7						7/M
1	wr 4		4/M			4/I	
2	rd 7			7/S			7/S
3	wr 5					5/M	
1	rd 3	3/S			3/S		
3	wr 3	3/I			3/I		3/M
2	wr 7			7/M			

MOESI-Protokoll

- Erweiterung des MESI-Protokolls um einen weiteren Zustand
- Neuer Zustand O: Owned (shared modified)

Vorteil

- Durch Cache-Cache-Transfers werden 2 Hauptspeicherzugriffe eingespart

Nachteile

- Komplexere Logik notwendig
- 3 Zustands-Bits nötig

MOESI Zustandsautomat

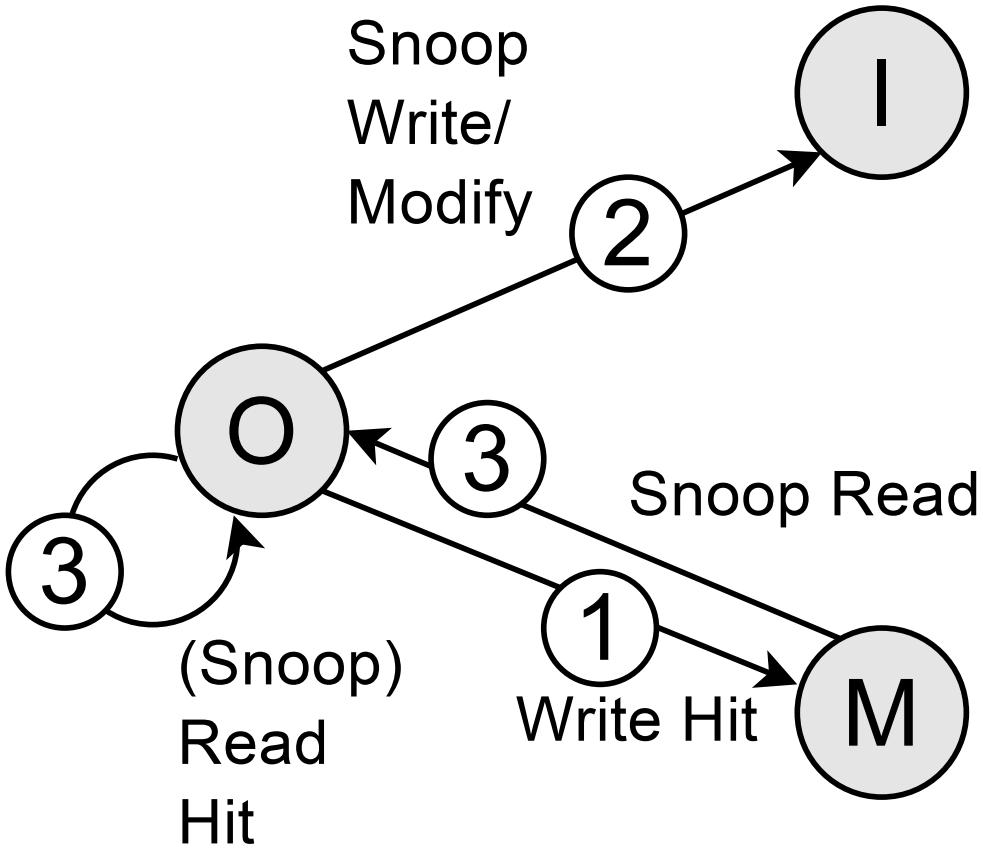
Karlsruher Institut für Technologie

Wird ein in einem Cache bereits modifiziertes Datum von einem weiteren Cache gelesen, so wechselt der Cache mit dem modifizierten Datum von Zustand **M** in den Zustand **O**. Der lesende Cache übernimmt das Datum aus dem Cache des ersten Prozessors und lagert das Datum, gemäß MESI-Protokoll, als Shared **S** markiert ein. Werden die Daten in Caches mit dem Zustand **S** modifiziert, so wird das Datum in den anderen Caches invalidiert (Zustand **I**) und in den Zustand **M** gewechselt.

a) Erweitern Sie den aus der Vorlesung bekannte MESI-Zustandsautomat um die oben beschriebene Erweiterungen zum MOESI-Zustandsautomat.

MOESI Zustandsautomat

Zusätzliche Kanten für O-Zustand:



1 Invalidate
2 Retry
3 Shared

Aufgabe 2.2: MOESI-Protokoll

Karlsruher Institut für Technologie

Ein Dreiprozessorsystem sei speichergekoppelt. Die Caches haben je eine Größe von zwei Cachezeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung des Caches erfolgt von der niedrigsten Cachezeile aufwärts, sofern noch freie Zeilen zur Verfügung stehen, andernfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll komme das MOESI-Protokoll zum Einsatz. Der Cache sei initial leer.

b) Vervollständigen sie die gegebene Tabelle: Geben Sie jeweils Inhalt der Cache-Zeile und MOESI-Zustand an.

Aufgabe 2.2: MOESI-Protokoll

Karlsruher Institut für Technologie

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4						
3	rd 4						
2	rd 3						
2	wr 5						
1	rd 2						
2	wr 4						
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Karlsruher Institut für Technologie

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3						
2	wr 5						
1	rd 2						
2	wr 4						
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2						
2	wr 4						
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4						
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1						
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1				1/E		
3	rd 4						
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1				1/E		
3	rd 4			4/O		4/S	
3	rd 3						
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1				1/E		
3	rd 4			4/O		4/S	
3	rd 3						3/E
1	wr 1						
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1				1/E		
3	rd 4			4/O		4/S	
3	rd 3						3/E
1	wr 1	1/M			1/I		
3	rd 1						

Aufgabe 2.2: MOESI-Protokoll

Karlsruher Institut für Technologie

Proz.	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
1	rd 4	4/E					
3	rd 4	4/S				4/S	
2	rd 3			3/E			
2	wr 5				5/M		
1	rd 2		2/E				
2	wr 4	4/I		4/M		4/I	
2	rd 1				1/E		
3	rd 4			4/O		4/S	
3	rd 3						3/E
1	wr 1	1/M			1/I		
3	rd 1	1/O				1/S	

MOESI-Zustandsautomat

Karlsruher Institut für Technologie

Aufgabe 2.2 - Teilaufgabe c

Wie viele Hauptspeicherzugriffe werden durch diese Speicherzugriffsfolge verursacht? Führt hier die Verwendung des MOESI gegenüber des MESI-Protokolls zu einer Leistungssteigerung und wenn ja, warum?

Antwort

MOESI: Lesend: 6 Zugriffe – Schreibend: 1 Zugriff
MESI: Lesend: 8 Zugriffe – Schreibend: 3 Zugriffe

Es würden zwei Lesezugriffe und zwei Schreibzugriffe eingespart → Leistungssteigerung vorhanden

MOESI-Zustandsautomat

Karlsruher Institut für Technologie

Aufgabe 2.2 - Teilaufgabe c

Wie viele Hauptspeicherzugriffe werden durch diese Speicherzugriffsfolge verursacht? Führt hier die Verwendung des MOESI gegenüber des MESI-Protokolls zu einer Leistungssteigerung und wenn ja, warum?

Antwort

MOESI: Lesend: 6 Zugriffe – Schreibend: 1 Zugriff
MESI: Lesend: 8 Zugriffe – Schreibend: 3 Zugriffe

Es würden zwei Lesezugriffe und zwei Schreibzugriffe eingespart $\rightarrow$ Leistungssteigerung vorhanden

Aufgabe 2.3: Verständnisfragen

a) Warum existiert im MOESI-Protokoll kein Zustandsübergang vom Zustand **S** nach Zustand **O**?

Antwort: Ein Zustandsübergang von **S** nach **O** kann es in einem Write-Invalidate-Protokoll nicht geben. Gemäß einem Write-Invalidate-Protokoll werden bei Veränderung eines geteilten Datums, die Daten in den entfernten Caches invalidiert und demzufolge dem Datum der Zustand **M** zugewiesen.

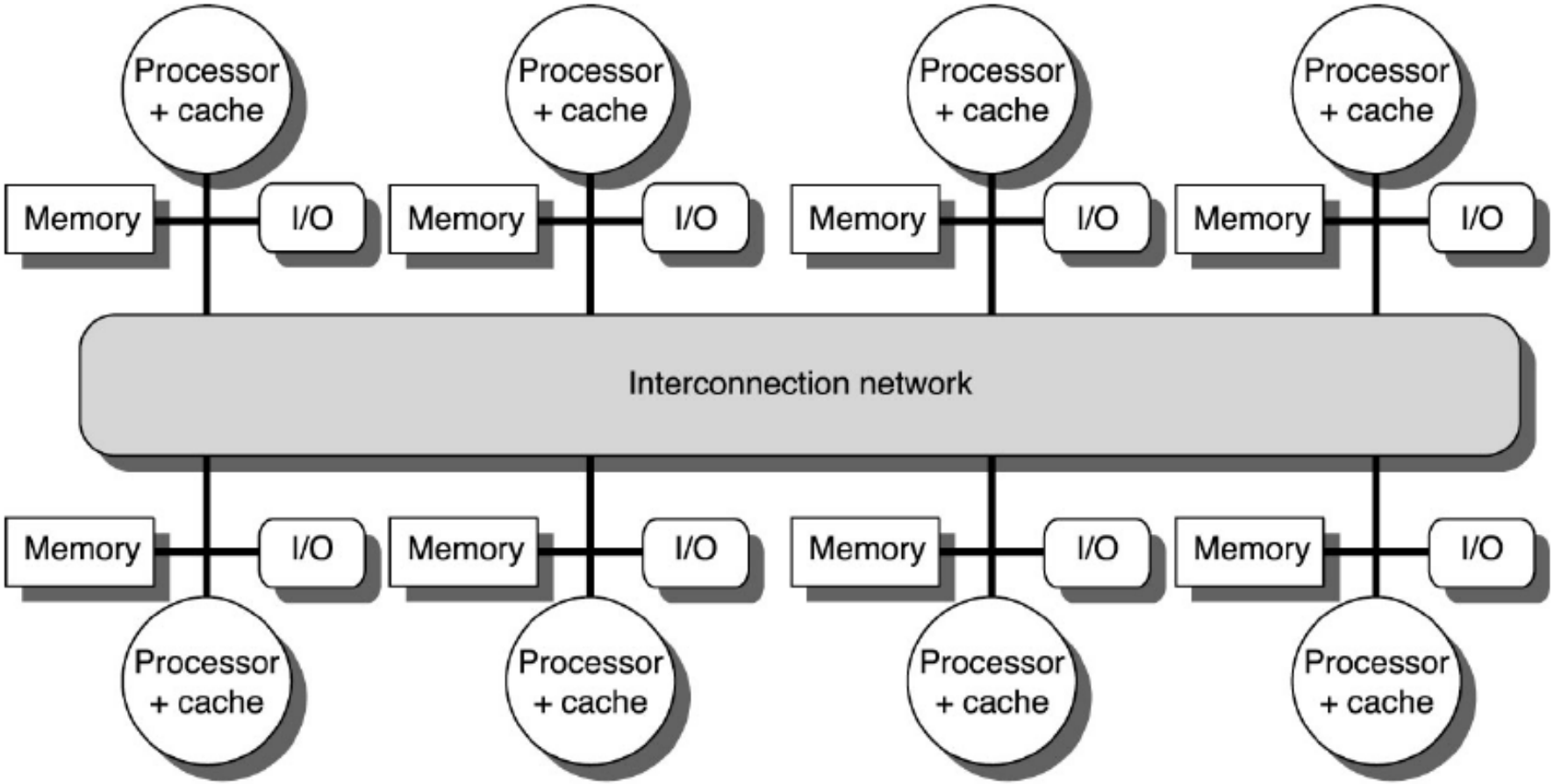
Der Wechsel von Zustand **S** in den Zustand **O**, müsste eine Aktualisierung des Datum in den entfernten Caches nach sich ziehen. Diese Vorgehensweise entspricht einem Write-Update-Protokoll

Aufgabe 2.3: Verständnisfragen

a) Warum existiert im MOESI-Protokoll kein Zustandsübergang vom Zustand **S** nach Zustand **O**?
Antwort: Ein Zustandsübergang von **S** nach **O** kann es in einem Write-Invalidate-Protokoll nicht geben. Gemäß einem Write-Invalidate-Protokoll werden bei Veränderung eines geteilten Datums, die Daten in den entfernten Caches invalidiert und demzufolge dem Datum der Zustand **M** zugewiesen.

Der Wechsel von Zustand **S** in den Zustand **O**, müsste eine Aktualisierung des Datum in den entfernten Caches nach sich ziehen. Diese Vorgehensweise entspricht einem Write-Update-Protokoll

Distributed-Shared-Memory (DSM)-System

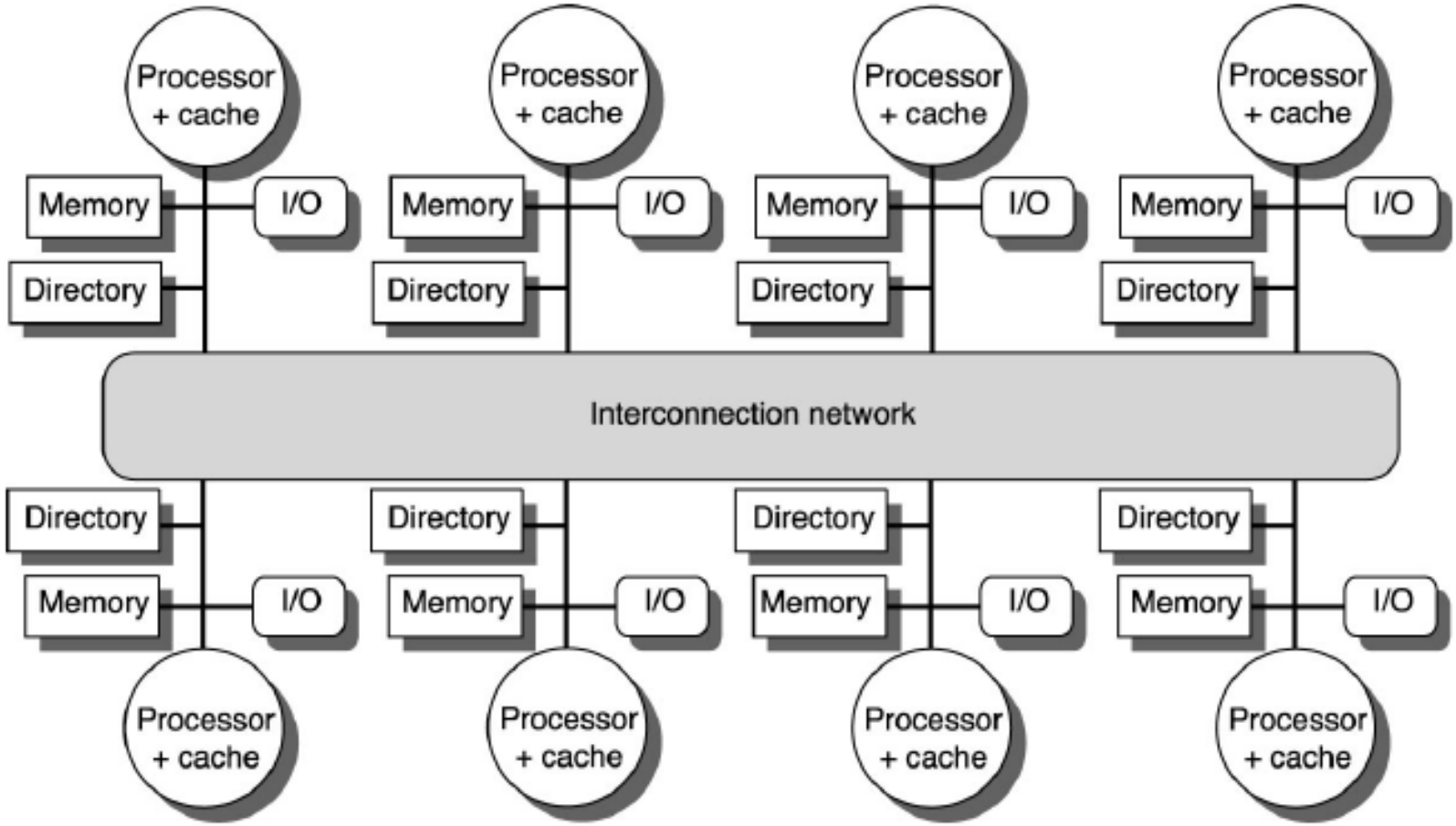


Kohärenz in DSM-Systemen

Einfachste Methode zur Wahrung der Kohärenz in DSM-Systemen:

- Nur Speicherzugriffe auf den lokalen Speicher werden in den Cache geladen

Verzeichnisbasierte Cache-Kohärenzprotokolle



Verzeichnisbasierte Cache-Kohärenzprotokolle

- DSM-Systeme haben kein gemeinsamer Speicherbus, den eine Snooping-Logik überwachen könnte
- Herstellen der Cache-Kohärenz über Verzeichnistabellen
- Implementierung in Hard- oder Software möglich
- Die Tabelle protokolliert für jeden Blockrahmen, ob dieser in den lokalen oder einem entfernten Cache-Speicher als Cache-Block übertragen worden ist.
- Zustände werden analog zu den MESI-Zuständen definiert

Aufgabe 2.3: Verständnisfragen

b) Warum läßt sich MESI nicht in Distributed Shared Memory (DSM) Systemen einsetzen?

Antwort:

b) In DSM-Systemen existiert kein gemeinsamer Speicherbus, den eine Snooping-Logik überwachen könnte.

Aufgabe 2.3: Verständnisfragen

b) Warum läßt sich MESI nicht in Distributed Shared Memory (DSM) Systemen einsetzen?

Antwort:

b) In DSM-Systemen existiert kein gemeinsamer Speicherbus, den eine Snooping-Logik überwachen könnte.

Aufgabe 2.3: Verständnisfragen

c) Welche Protokolle stellen in DSM-Systemen die Cache-Kohärenz sicher?

Antwort:

c) In DSM-Systemen kommen verzeichnisbasierte Cache-Kohärenzprotokolle zum Einsatz.

Aufgabe 2.3: Verständnisfragen

c) Welche Protokolle stellen in DSM-Systemen die Cache-Kohärenz sicher?

Antwort:

c) In DSM-Systemen kommen verzeichnisbasierte Cache-Kohärenzprotokolle zum Einsatz.

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

Sie sollen prüfen ob es vorteilhaft wäre eine bestehende Speicherhierarchie (Variante A) durch eine veränderte Variante B zu ersetzen. In der Variante A findet der Zugriff auf die nächste Hierarchieebene parallel zur ersten Ebene statt, während Variante B diese nacheinander anfragt (sequentiell).

	Variante A	Variante B
Zugriffszeit L1	4 ns	6 ns
Hitrate L1	80%	80%
Zugriffszeit L2	20 ns	19 ns
Hitrate L2	80%	75%
Zugriffszeit Hauptspeicher	100 ns	100 ns

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

d) **Um eine fundierte Aussage treffen zu können, berechnen Sie die durchschnittliche Antwortzeit für beide Varianten. Welche Variante empfehlen Sie und warum?**

- Formel für parallele Zugriffe auf Hierarchieebene $\Rightarrow$ s. Folie 9
- Formel für sequentielle Zugriffe:

$$t_a = \underbrace{r_{H1} * t_{L1}}_{\text{Hit L1}} + \underbrace{r_{M1} * (\underbrace{r_{H2} * (t_{L1} + t_{L2})}_{\text{Hit L2}} + \underbrace{r_{M2} * (t_{L1} + t_{L2} + t_{Mem})}_{\text{Miss L2}})}_{\text{Miss L1}}$$

Alternative A:

$$t_a = 0,8 * 4ns + 0,2 * (0,8 * 20ns + 0,2 * 100ns)$$
$$= 3,2ns + 0,2 * (16ns + 20ns)$$
$$= 3,2ns + 7,2ns = 10,4ns$$

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

d) **Um eine fundierte Aussage treffen zu können, berechnen Sie die durchschnittliche Antwortzeit für beide Varianten. Welche Variante empfehlen Sie und warum?**

- Formel für parallele Zugriffe auf Hierarchieebene $\Rightarrow$ s. Folie 9
- Formel für sequentielle Zugriffe:

$$t_a = \underbrace{r_{H1} * t_{L1}}_{\text{Hit L1}} + \underbrace{r_{M1} * (\underbrace{r_{H2} * (t_{L1} + t_{L2})}_{\text{Hit L2}} + \underbrace{r_{M2} * (t_{L1} + t_{L2} + t_{Mem})}_{\text{Miss L2}})}_{\text{Miss L1}}$$

Alternative A:

$t_a = 0,8 * 4ns + 0,2 * (0,8 * 20ns + 0,2 * 100ns)$

$= 3,2ns + 0,2 * (16ns + 20ns)$

$= 3,2ns + 7,2ns = 10,4ns$

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

d) **Um eine fundierte Aussage treffen zu können, berechnen Sie die durchschnittliche Antwortzeit für beide Varianten. Welche Variante empfehlen Sie und warum?**

- Formel für parallele Zugriffe auf Hierarchieebene $\Rightarrow$ s. Folie 9
- Formel für sequentielle Zugriffe:

$$t_a = \underbrace{r_{H1} * t_{L1}}_{\text{Hit L1}} + \underbrace{r_{M1} * (\underbrace{r_{H2} * (t_{L1} + t_{L2})}_{\text{Hit L2}} + \underbrace{r_{M2} * (t_{L1} + t_{L2} + t_{Mem})}_{\text{Miss L2}})}_{\text{Miss L1}}$$

Alternative A:

$$t_a = 0,8 * 4ns + 0,2 * (0,8 * 20ns + 0,2 * 100ns)$$
$$= 3,2ns + 0,2 * (16ns + 20ns)$$
$$= 3,2ns + 7,2ns = 10,4ns$$

Klausur WS 13/14 Cache-Leistung

Aufgabe 4

d) **Um eine fundierte Aussage treffen zu können, berechnen Sie die durchschnittliche Antwortzeit für beide Varianten. Welche Variante empfehlen Sie und warum?**

- Formel für parallele Zugriffe auf Hierarchieebene $\Rightarrow$ s. Folie 9
- Formel für sequentielle Zugriffe:

$$t_a = \underbrace{r_{H1} * t_{L1}}_{\text{Hit L1}} + \underbrace{r_{M1} * (\underbrace{r_{H2} * (t_{L1} + t_{L2})}_{\text{Hit L2}} + \underbrace{r_{M2} * (t_{L1} + t_{L2} + t_{Mem})}_{\text{Miss L2}})}_{\text{Miss L1}}$$

Alternative B:

$t_a = 0,8 * 6ns + 0,2 * (0,75 * 25ns + 0,25 * 125ns)$
$= 4,8ns + 0,2 * (\frac{75}{4}ns + \frac{125}{4}ns)$
$= 4,8ns + 0,2 * 50ns = 14,8ns$

Somit ist B langsamer, wodurch sich ein Wechsel nicht lohnt.

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

e) **Das System soll nun zu einem Multiprozessorsystem mit gemeinsam verwendetem Speicher ausgebaut werden, indem die Prozessoren mittels eines Busses verbunden werden. Beziehen Sie diesen Gesichtspunkt nun in die vorherige Bewertung der Ergebnisse der zwei Varianten mit ein. Verändert dieser Aspekt den Schluß aus ihrer Bewertung? Begründen Sie.**

- Hauptspeicheranfragen aller Prozessoren nun über gemeinsamen Bus

$\Rightarrow$ Unnötige Anfragen an Hauptspeicher möglichst vermeiden

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4

e) **Das System soll nun zu einem Multiprozessorsystem mit gemeinsam verwendetem Speicher ausgebaut werden, indem die Prozessoren mittels eines Busses verbunden werden. Beziehen Sie diesen Gesichtspunkt nun in die vorherige Bewertung der Ergebnisse der zwei Varianten mit ein. Verändert dieser Aspekt den Schluß aus ihrer Bewertung? Begründen Sie.**

- Hauptspeicheranfragen aller Prozessoren nun über gemeinsamen Bus

$\Rightarrow$ Unnötige Anfragen an Hauptspeicher möglichst vermeiden

Klausur WS 13/14 Cache-Leistung

Karlsruher Institut für Technologie

Aufgabe 4 - Antwort

e) Für ein Multiprozessorsystem mit gemeinsamem Speicher lohnt sich ein Überdenken der Bewertung, da bei Variante A im Fall eines Cache-Hits in den höheren Ebenen immer Anfragen an den Hauptspeicher gestartet werden, die dann später wieder abgebrochen werden. Diese blockieren den Bus und verhindern so, dass notwendige Daten zu anderen Prozessoren transferiert werden. Somit kann man argumentieren, dass die längere Antwortzeit der Variante B toleriert werden kann, da hier keine unnötigen Anfragen an den Hauptspeicher auftreten.

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Aufgabe 3

Ein Zweiprozessorsystem sei speichergekoppelt. Die Caches haben je eine Größe von drei Cache-Zeilen, welche je genau ein Speicherwort aufnehmen können. Die Füllung erfolgt von der niedrigsten Zeile aufwärts, sofern noch freie Zeilen zur Verfügung stehen, andernfalls wird gemäß LRU-Strategie verdrängt. Als Cache-Kohärenzprotokoll käme MESI zum Einsatz.

a) **Vervollständigen Sie die auf dem Lösungsblatt angegebene Tabelle: Geben Sie jeweils Inhalt der Cache-Zeile und MESI-Zustand an.**

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6						
2	rd 2						
1	wr 6						
1	wr 4						
2	rd 3						
1	rd 5						
2	rd 6						
2	wr 5						

Klausur SS 13 Cache-Kohärenzprotokoll

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6						
1	wr 4						
2	rd 3						
1	rd 5						
2	rd 6						
2	wr 5						

Klausur SS 13 Cache-Kohärenzprotokoll

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6	6/M					
1	wr 4		4/M				
2	rd 3						
1	rd 5						
2	rd 6						
2	wr 5						

Klausur SS 13 Cache-Kohärenzprotokoll

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6	6/M					
1	wr 4		4/M				
2	rd 3					3/E	
1	rd 5			5/E			
2	rd 6						
2	wr 5						

Klausur SS 13 Cache-Kohärenzprotokoll

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6	6/M					
1	wr 4		4/M				
2	rd 3					3/E	
1	rd 5			5/E			
2	rd 6	6/S					6/S
2	wr 5						

Klausur SS 13 Cache-Kohärenzprotokoll

Proz.	Aktion	Prozessor 1			Prozessor 2		
		Line 1	Line 2	Line 3	Line 1	Line 2	Line 3
1	rd 6	6/E					
2	rd 2				2/E		
1	wr 6	6/M					
1	wr 4		4/M				
2	rd 3					3/E	
1	rd 5			5/E			
2	rd 6	6/S					6/S
2	wr 5			5/I	5/M		

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Aufgabe 3

b) **Auf welchem Konzept basiert das MESI-Kohärenzprotokoll und was ist hierfür die technische Grundvoraussetzung?**

Bus-Snooping, welches einen gemeinsamen Datenbus voraussetzt.

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Aufgabe 3

b) **Auf welchem Konzept basiert das MESI-Kohärenzprotokoll und was ist hierfür die technische Grundvoraussetzung?**

Bus-Snooping, welches einen gemeinsamen Datenbus voraussetzt.

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Aufgabe 3

c) **Um einen Prozessor in einem bestehenden SMP-System mit MESI-Kohärenzprotokoll einsetzen zu können, müssen der Cache sowie der Cache-Controller des Prozessors zusätzliche Voraussetzungen erfüllen. Nennen Sie zwei der notwendigen Eigenschaften.**

Mögliche Antworten:

- 2 Zustandsbits in jeder Cache-Zeile zur Speicherung des MESI-Zustands
- Bus-Monitor zur Überwachung des Speicherbusses
- Retry, Abort und Invalidate Signale zu weiteren Prozessoren
- Erweiterung des Cache-Controllers um den MESI-Zustandsautomat

Klausur SS 13 Cache-Kohärenzprotokoll

Karlsruher Institut für Technologie

Aufgabe 3

c) **Um einen Prozessor in einem bestehenden SMP-System mit MESI-Kohärenzprotokoll einsetzen zu können, müssen der Cache sowie der Cache-Controller des Prozessors zusätzliche Voraussetzungen erfüllen. Nennen Sie zwei der notwendigen Eigenschaften.**

Mögliche Antworten:

- 2 Zustandsbits in jeder Cache-Zeile zur Speicherung des MESI-Zustands
- Bus-Monitor zur Überwachung des Speicherbusses
- Retry, Abort und Invalidate Signale zu weiteren Prozessoren
- Erweiterung des Cache-Controllers um den MESI-Zustandsautomat

Klausur Rechnerstrukturen

Karlsruher Institut für Technologie

Klausur am 06.08.2014, 11:00 Uhr

- Anmeldung ist online möglich via QISPOS!
- Anmeldeschluss ist der 29.07.2014
- **Danach besteht kein Anrecht mehr auf Klausurteilnahme!**

- Abmeldung online bis 29.07.2014
- **Hinweise auf der RS-Homepage beachten!**

- Erasmus-Studenten,. . .

$\Rightarrow$ Bitte Prüfungzulassung bei mir persönlich im Büro oder nach den Übungen abgeben!

Ausblick

Übung #8 – 18.07.2014

- **Vektorrechner**

- **Kurzer Überblick über den Inhalt aller Übungen**
- **Fragen/Antworten zur Klausur**

Karlsruher Institut für Technologie

Fragen?

Karlsruher Institut für Technologie

Zentralübung Rechnerstrukturen im SS 2014
Cache-Kohärenz

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

15. Juli 2014