

# Zentralübung Rechnerstrukturen: Cache-Kohärenz

## 7. Übungsblatt – Musterlösung

### 1 Cache: Grundlagen

#### 1.1 Cacheleistung

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

b) Alternative A:

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

Alternative B:

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

## 1.2 Beweise

- a) Widerlegung durch Beispiel bei dem Aussage falsch ist:  
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 Caches

- b) Widerlegung durch Beispiel bei dem Aussage falsch ist:  
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.

### 1.3 Verständnisfragen

- a) Räumliche und zeitliche Lokalität
- b) Conflict-, Capacity-, Compulsory-Misses
- Im Mehrprozessorfall: Coherency-Miss
    - Unterscheidung in: False-Sharing-Miss und True-Sharing-Miss
- c)
- 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
    - ⇒ Signale brauchen länger und Zugriffszeit erhöht sich
    - ⇒ Vorteil von SRAM geht verloren

## 2 Cache-Kohärenzprotokolle

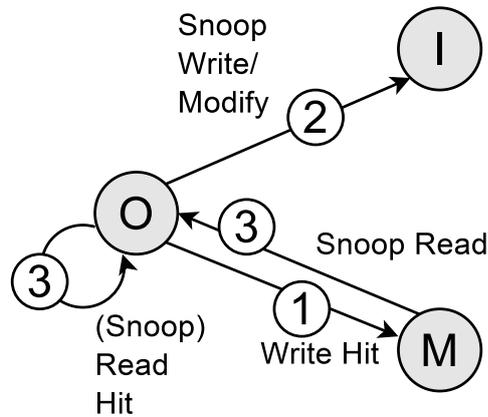
### 2.1 MESI

a) Tabelle:

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/					
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			

## 2.2 MOESI

a) Erweiterung des MESI-Zustandsautomats:



**1** Invalidate

**2** Retry

**3** Shared

b) Tabelle:

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	

c) MOESI: Lesend: 6 Zugriffe – Schreibend: 1 Zugriff

MESI: Lesend: 8 Zugriffe – Schreibend: 3 Zugriffe

Es würden zwei Lesezugriffe und zwei Schreibzugriffe eingespart → Leistungssteigerung vorhanden

## 2.3 Verständnisfragen

- a) 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
- b) In DSM-Systemen existiert kein gemeinsamer Speicherbus, den eine Snooping-Logik überwachen könnte.
- c) In DSM-Systemen kommen verzeichnisbasierte Cache-Kohärenzprotokolle zum Einsatz.

## 3 Klausuraufgaben

Die Musterlösungen zu den Klausuraufgaben finden Sie auf der Homepage zur Vorlesung Rechnerstrukturen. Weitere Erklärungen siehe in den Folien zur Übung.