



## Exercise 10

23. Juni 2015

Abgabe: 30. Juni 2015, 10.00 Uhr

### Problem 10.1: Cache-Design

Normalerweise läuft eine Arbeitsspeicher-Leseanfrage so ab, dass der Prozessor eine Anfrage an den Cache schickt und nur bei einem Cache-Miss anschließend beim Arbeitsspeicher nachfragt. Um die Verzögerung bei Read-Misses etwas zu reduzieren, kann der Prozessor eine Leseanfrage auch gleichzeitig an den Cache und an den Hauptspeicher richten.

Durch den zusätzlichen Verkehr auf dem Bus steht bei der zweiten Strategie für den Cache nur noch ein geringeres Hardware Budget zur Verfügung, wodurch sich die Trefferrate  $r'_h$  im Vergleich zur Trefferrate  $r_h$  der ersten Strategie verringert.

1. Geben Sie für beide Strategien die mittlere Zugriffszeit bei Leseanfragen in Abhängigkeit von Trefferrate  $r_h$ , Cache-Zugriffszeit  $t_C$  und Arbeitsspeicher-Zugriffszeit  $t_M$  an. 3 Points
2. Wie groß muss die reduzierte Trefferrate  $r'_h$  mindestens sein, damit die zweite Strategie besser ist? 2 Points
3. Ist es sinnvoller, bei einem Direct-Mapped-Cache die höchstwertigen oder niedrigstwertigen Bits der Adresse als Index zu verwenden? Begründen Sie ihre Antwort! 2 Points

## Problem 10.2: Cache

1. Warum werden in Computern Caches verwendet? 1 Points
2. Was ist der Unterschied zwischen direkt abgebildeten (direct-mapped) Caches und voll-assoziativen (fullassociative) Caches? Nennen Sie je einen Vorteil der beiden Varianten! 1 Points
3. Nennen Sie drei mögliche, realistische Verdrängungsstrategien bei einem voll-assoziativem (full-associative) Cache. 1 Points
4. Vervollständigen Sie die Tabelle 1 auf der nächsten Seite, die einen direkt abgebildeten (direct-mapped) Cache mit 4 Blöcken beschreibt. 10 Points
  - a) Geben Sie für jede Anfrage an, ob es zu einem Cache-Hit oder einem Cache-Miss kommt und tragen Sie ggf. neu in den Cache kommende Werte entsprechend ein!
  - b) Setzen Sie an den entsprechenden Stellen das Dirty Bit und geben Sie an, wenn ein Wert aus dem Cache in den Arbeitsspeicher zurückgeschrieben werden muss!

### Hint:

Die Reihenfolge der Befehle ist zu beachten!

Total: 

---

20 Points

Die Einträge in jeder Zelle haben das Format „< Adresse >:< Datenwert >“. Im Cache veränderte Werte werden erst in den Arbeitsspeicher zurückgeschrieben, wenn die modifizierte Zelle aus dem Cache verdrängt wird (write back); entsprechende Zeilen sind mit einem Stern („\*“) markiert. Bei write-misses wird der entsprechende Wert in den Cache geladen (write-allocate) und ebenfalls erst in den Arbeitsspeicher geschrieben, wenn die Zelle aus dem Cache verdrängt wird.

Nehmen Sie für Ihnen unbekannte Werte, die aus dem Arbeitsspeicher geladen werden sollen, an, dass dort an Adresse  $i$  der Wert  $11 \cdot i$  steht, an Adresse 4 also der Wert 44.

#	Anfrage	Zelle 0	Zelle 1	Zelle 2	Zelle 3	Kommentar
	<i>urspr. Belegung</i>	8:88	5:50(*)	2:22	7:70(*)	
1	MEM[1] = 10	8:88	1:10(*)	2:22	7:70(*)	write miss, write back
2	read MEM[4]	4:44	1:10(*)	2:22	7:70(*)	read miss
3	read MEM[7]					
4	read MEM[3]					
5	MEM[2] = 20					
6	MEM[0] = 0					
7	MEM[3] = 30					
8	read MEM[0]					
9	read MEM[6]					
10	read MEM[5]					

Tabelle 1: Cache-Belegung