

11. Übung für die Vorlesung Rechnerorganisation

Sommersemester 2019

Abgabe: keine Abgabe!

Wichtige Hinweise zum Übungsbetrieb:

1. Dieser Übungszettel wird nicht mehr abgegeben und auch nicht für die Klausurzulassung gewertet, der Inhalt ist weiterhin prüfungsrelevant!
2. Die Aufgaben werden zusammen mit dem 10. Zettel in den Tutorien am 8.7. bzw. 9.7. besprochen.
3. Die Tutorien am 8.7. bzw. 9.7. beginnen und enden pünktlich zur vollen Stunde (s.t.) und dauern damit 120 Minuten.

Aufgabe 1. *Cache*

0 P.

Betrachten Sie einen 4-fach set associative Cache. Ein Block besteht aus 4 Worten, jedes Wort besteht aus 32 Bits. Die Gesamtspeicherkapazität beträgt 256 Bytes. Als Verdrängungsstrategie wird LRU (Least Recently Used) verwendet, d.h. der Block, auf den am längsten nicht mehr zugegriffen wurde, wird verdrängt.

1. Wie viele Sets hat der Cache?
2. Der Prozessor benutzt 32-Bit Byte-Adressen. Geben Sie an, welche Bits der Byte-Adressen für das Tag, die Set-, Word- und Byte-Adressierung verwendet werden.
3. Der Prozessor greift nacheinander auf folgende Byte-Adressen zu:
(in Hexadezimal-Darstellung angegeben)

00000014_{16} , 00000090_{16} , $000000D0_{16}$, $0000000C_{16}$, 00000018_{16} , 00000058_{16} , 00000028_{16} , 00000110_{16} .

Geben Sie alle Byte-Adressen an, die anschließend zu einem Hit führen, obwohl noch nie auf diese Adresse zugegriffen wurde.

Aufgabe 2. *Cache-Varianten*

0 P.

Ihr Programm greift nacheinander auf folgende Wort-Adressen zu:

2, 4, 1, 133, 36, 156, 133, 4

Ihr Cache-Baustein hat für die zu speichernden Daten eine Größe von 128 Bytes, sowie ausreichend Platz für die benötigten Verwaltungsbits. Ein Wort ist 32 Bit lang. Geben Sie für die verschiedenen Cacheorganisationen 1.-4. an, wie der Cache nach jedem Zugriff belegt ist, und ob es sich bei den Speicher-Zugriffen um eine Erstbelegung, einen *hit* oder um einen *miss* handelt. Wenn Daten aus dem Cache verdrängt werden müssen, so wird das am längsten nicht mehr benutzte Datum verdrängt (LRU-Strategie). Es genügt, wenn Sie nur Blocks bzw. Sets angeben, in denen sich etwas ändert.

Verwenden Sie Tabellen der Form

Adresse	Cache Block/ Set	hit/ miss/ Erstb.	Inhalt des Block bzw. Set					
			1	2	3	4	...	
2 = 000 00010	00010	E		M[2]				
4 = 000 00100								
1 = 000 00001								
133 = 100 00101								
36 = 001 00100								
156 = 100 11100								
133 = 100 00101								
4 = 000 00100								

Tabelle 1: Beispiel-Tabelle

1. direkte Adressierung (direct mapped - jede Speicheradresse kann auf genau einen Block abgebildet werden), jeder Block umfasst ein Wort
2. direkte Adressierung (direct mapped), jeder Block umfasst vier Worte
3. zweifach-assoziativ (2-way-set-associative - jedes Set enthält bis zu zwei Blöcke), jeder Block umfasst vier Worte
4. voll-assoziativ (fully associative - jede Speicheradresse kann auf jeden beliebigen Block abgebildet werden), jeder Block umfasst vier Worte.