

Digitale Systeme SS2015

Übungsblatt 3

Abgabe in der Woche ab 29.06.2015 zu Beginn der Übungen

Geben Sie ein Exemplar der Lösungen pro Gruppe ab. Die Gruppengröße muss 3 oder 4 betragen. Denken Sie daran, die Immatrikulationsnummern aller Beteiligten auf das Übungsblatt zu schreiben

Aufgabe 1

(3 Punkte)

Ein Cache mit 8 Cache-Frames sei als direct-mapped Cache organisiert. Folgende Zugriffssequenz auf Hauptspeicherblöcke werde beobachtet:

2 5 0 13 2 5 10 8 0 4 5 2

(alle Betrachtungen beziehen sich auf anfänglich leere caches)

- (a) Welche Trefferquote ergibt sich bei oben genannter Organisation des Caches?

Verwenden Sie zur Verdeutlichung der Vorgänge die vorgegeben Tabellen.

direct mapped							
0	1	2	3	4	5	6	7

Lösung:

direct mapped							
0	1	2	3	4	5	6	7
		2					
					5		
0							
					13		
		(2)					
					5		
		10					
8							
0							
				4			
					(5)		
		2					

Trefferquote: ca. 17%

Wie erhöht/verringert sich die Trefferquote, wenn der Cache als
 (b) set-assoziativer Cache mit Set-Größe 4 und FIFO-Zugriff

set-assoziativ (2 Sets)							
0	1	2	3	0	1	2	3

Lösung:

set-assoziativ (2 Sets)							
0	1	2	3	0	1	2	3
2							
				5			
0							
				13			
(2)							
				(5)			
		10					
			8				
(0)							
4							
				(5)			
2							

Trefferquote: ca. 42%

(c) vollassoziativer Cache mit LRU-Zugriff

arbeiten würde?

vollassoziativ							
0	1	2	3	4	5	6	7

Lösung:

vollassoziativ							
0	1	2	3	4	5	6	7
2							
	5						
		0					
			13				
(2)							
	(5)						
				10			
					8		
		(0)					
						4	
	(5)						
(2)							

Trefferquote: ca. 42%

Aufgabe 2

(3 Punkte)

Gegeben sei ein Writeback-Cache mit einer Cachelinegröße von 32 Byte. Der Cache sei 1 MByte groß. Der Hauptspeicher sei 4 GByte groß und byteadressiert.

- (a) Wie viele Bit müssen für Steuerinformationen (Tag, Statusinformation) für jeden Eintrag bereitstehen, wenn ein 4-fach-setassoziativer Cache zum Einsatz kommt?
- (b) Wie viele Bit müssen für Steuerinformationen für jeden Eintrag bereitstehen, wenn ein voll assoziativer Cache zum Einsatz kommt?
- (c) Wie viele Bit müssen für Steuerinformationen für jeden Eintrag bereitstehen, wenn ein direct mapped Cache zum Einsatz kommt?

Lösung:

- (a) Um jedes einzelne Byte des 4 GByte großen Speichers zu adressieren, sind 32 Bit Adressen notwendig. Von diesen Adressbits werden als Tag nur eine geringere Anzahl gespeichert.

Bei der Adressierung im set-assoziativen Cache werden einige Adressbits für die Auswahl eines Sets benötigt, die dann nicht mehr zum Tag gehören.

Durch die Adressierung von 32-Byte großen Blöcken entfallen nochmals 5 Bit, da im Cache Blöcke organisiert werden.

Die Anzahl der Sets ergibt sich nach folgender Formel:

$$\begin{aligned}\# \text{ Anzahl Sets} &= \frac{\text{Cachegröße}}{\text{Blockgröße} \cdot \text{Blöcke pro Set}} = \frac{1 \text{ MByte}}{32 \text{ Byte} \cdot 4} \\ &= \frac{2^{20} \text{ Byte}}{2^5 \text{ Byte} \cdot 2^2} = 2^{13} = 8192\end{aligned}$$

Es werden also 13 Bit für die Set-Adressierung verwendet, die nicht im Tag verwendet werden müssen. Für einen Write-Through-Cache benötigt man zusätzlich 1 Bit als Valid/Invalid Bit. Bei einem Write-Back-Cache braucht man sogar 2 zusätzliche Bits für Valid/Invalid und Clean/Dirty, als Statusinformationen. Demnach ergeben sich $32 - 13 - 5 + 1 = 15$ Bit für den Write-Through-Cache und $32 - 13 - 5 + 2 = 16$ Bit für den Write-Back-Cache.

- (b) Eine Cacheline kann nicht durch Berechnung festgelegt werden. Die letzten 5 Bit müssen wieder nicht gespeichert werden. Der Rest der insgesamt 32 Bit langen Speicheradresse stellt das also Tag dar und muss gespeichert werden. Zusätzlich sind auch hier wieder 2 Bit für valid/invalid bzw. clean/dirty notwendig. Es ergeben sich $32 - 5 + 2 = 29$ Bit für den Write-Back-Cache.
- (c) Durch die direkte Adressierung einer der 2^{15} Cachelines entfallen 15 Bit der Speicheradresse als Tag. Auch hier sind außerdem die letzten 5 Bit nicht erforderlich. Trotzdem sind wieder die 2 Bit Statusinformation nötig. Es ergibt sich also eine Anzahl an Tagbits von $32 - 15 - 5 + 2 = 14$.

Aufgabe 3

(4 Punkte)

In einem Rechner, der virtuellen Speicher mittels Paging realisiert, generiert die CPU diese Folge logischer Adressen (als Dezimalzahl angegeben) :

901, 2047, 1100, 1199, 912, 178, 3389

Nehmen Sie an:

- die Seitengröße sei 512 Byte
- die CPU generiert 12 Bit lange logische Adressen
- der Hauptspeicher sei byteweise adressierbar und kann 4 Seiten aufnehmen

(a) Geben Sie für jede logische Adresse die Binärdarstellung, die logische Seitennummer und den Offset (beides dezimal) an.

Adresse	Adresse binär	Seite	Offset
901			
2047			
1100			
1199			
912			
178			
3389			

Lösung: Eine Möglichkeit ist es die Adressen in die Binärform umwandeln: die obere 3 Bits geben dann die Seitennummern an und der Rest definiert der Offset.

Adresse	Adresse binär	Seite	Offset
901	001 110000101	1	389
2047	011 111111111	3	511
1100	010 001001100	2	76
1199	010 010101111	2	175
912	001 110010000	1	400
178	000 010110010	0	178
3389	110 100111101	6	317

(b) Stellen Sie den Inhalt der Seitentabelle der MMU nach Abschluss aller, in der oben angegebenen Reihenfolge erfolgten Speicherzugriffe dar.

Sie brauchen nur die auf jeden Fall notwendigen Einträge angeben.

Alle page frames seien zu Beginn unbelegt und werden mit aufsteigender Seitennummer verwendet, d.h. der zuerst angesprochenen logischen Seite wird die physische Seite 0, der danach angesprochenen logischen Seite die physische Seite 1 usw. zugeordnet. Wenn alle page frames in Verwendung sind, wird die LRU-Ersetzungsstrategie angewendet. Stellen Sie alle Angaben binär dar.

Lösung: Als erstes wird die logische Seite 1 angesprochen, ihr wird die physische Seite 0 zugeordnet. Als nächstes Adressierung innerhalb LS 3 und Zuordnung von PS 1 usw. Logische Seiten 1 und 2 werden zwei mal angesprochen, deshalb erfolgt die Zuordnung nur bei jeweils ersten Mal, bei den jeweils zweiten Zugriffen ergibt sich ein "Hit". Als letztes wird logische Seite 6 angesprochen. Es ist kein Page Frame mehr frei, so das nun die physische Seite 1 der logischen Seite 6 zugeordnet wird, da die logische Seite 3 am längsten nicht verwendet wurde.

Seitentabelle:

Page Frame Nummer	Valid Flag
11	1
00	1
10	1
01	0
	0
	0
01	1
	0

- (c) Geben Sie zu jeder logischen Adresse die verwendete(n) physische(n) Speicheradresse(n) an. Stellen Sie alle Angaben binär dar.

Log. Adresse	Phys. Adresse(n)

Lösung: Generierte physische Adressen:

Log. Adresse	Phys. Adresse
001 110000101	00 110000101
011 111111111	01 111111111
010 001001100	10 001001100
010 010101111	10 010101111
001 110010000	00 110010000
000 010110010	11 010110010
110 100111101	01 100111101