Übungen zu Rechnerarchitektur (Cache Replacement (LRU))

Sommersemester 2017

Jian-Jia Chen

(Slides from Dipl.-Inf. Björn Bönninghoff)

Technische Universität Dortmund

Lehrstuhl Informatik 12

Entwurfsautomatisierung für Eingebettete Systeme

Cacheersetzungsstrategien

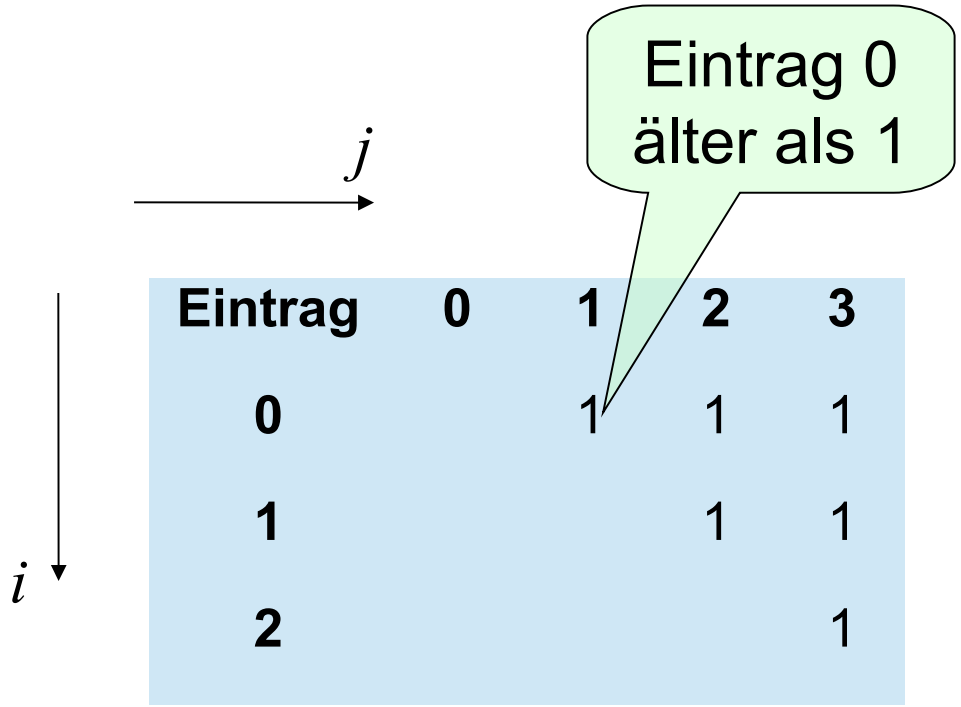
- **L1 Cache: 256Byte, 4 Wege, Blockgröße 16Byte, write-back, write-allocate**

  - Anzahl der Cachezeilen
    - → 16 Bytes/Block x 4 Blöcke/Zeile = 64Byte/Zeile
    - → 4 Zeilen

  - Aufteilung der Adressbits
    - Offset: 4 Bits für Byte-Adressierung von 16 = 2^4 Bytes
    - Index: 2 Bit für die Auswahl von 4 Zeilen
    - Tag: 26 Bits (die restlichen Bits)

Realisierung von LRU mittels Dreiecksmatrix (1)

- **Def: $f[i, j] = 1$, falls Zugriff auf i älter ist als der auf j und 0 sonst.**



Wegen Antisymmetrie nur Dreiecksmatrix $f[i, j]$, $i < j$ (d.h. Zeilenindex < Spaltenindex) zu speichern.

Dreiecksmatrizen

- **Zeile 2**

  Zugriffsfolge 2,3,1,0 (alt → neu)

Eintrag	0	1	2	3
0		**0**	**0**	**0**
1			**0**	**0**
2				**1**

Eintrag 1 älter als 0

Eintrag 2 älter als 3

- **Zeile 3**

  Zugriffsfolge 0,2,3,1

Eintrag	0	1	2	3
0		**1**	**1**	**1**
1			**0**	**0**
2				**1**

Dreiecksmatrizen

- **andere Zeilen**

  Zugriffsfolge 0,1,2,3

Eintrag	0	1	2	3
0		**1**	**1**	**1**
1			**1**	**1**
2				**1**

- **Beispiel: ungültiger Zustand**

Eintrag	0	1	2	3
0		**0**	**1**	**1**
1			**1**	**0**
2				**1**

Cacheersetzungsstrategien

b) Der Prozessor greift nun auf die folgenden Speicheradressen zu:

1. Write: 0x00493720
2. Write: 0x00493734
3. Read: 0x00471110
4. Read: 0x00493728
5. Read: 0x0049352C
6. Read: 0x00493734

Gehen Sie davon aus, dass zu Beginn der Zugriffssequenz die Tags in allen Cacheblöcken einer Null entsprechen. Geben Sie für jeden Zugriff an, ob ein Cache-Miss oder Cache-Hit vorliegt und welche Cachezeile geprüft wurde. Wie sieht nach einem Zugriff die Dreiecksmatrix der Zeile aus und welcher Block wird bei einem Cache-Miss ersetzt?

Speicherzugriff 1: Write 0x00493720

- $(00493720)_{16}$ =

$(0000\ 0000\ 0100\ 1001\ 0011\ 0111\ 0010\ 0000)_2$

- **Offset [3:0] = 0000**
- **Index [4:4] = 10**

→ **Tag [31:5] = 0000 0000 0100 1001 0011 0111 00**

Realisierung von LRU mittels Dreiecksmatrix (2)

Annahme: Initialisierung mit 1

Cache miss ☞ Überschreiben von *way* 0

Cache miss ☞ Überschreiben von *way* 1

Jüngster Eintrag jeweils gestrichelt

Ein-trag	**0**	**1**	**2**	**3**
0		1	1	1
1			1	1
2				1

Ein-trag	**0**	**1**	**2**	**3**
0		0	0	0
1			1	1
2				1

Ein-trag	**0**	**1**	**2**	**3**
0		1	0	0
1			0	0
2				1

Speicherzugriff 1: Write 0x00493720

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 10 Offset = 0000

	0	1	2	3				
0		1	1	1	?	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0		1	1	1	?	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0		0	0	0	?	?	?	?
1			0	0				
2				1				
	0	1	2	3				
0		1	1	1	?	?	?	?
1			0	0				
2				1				

Speicherzugriff 1: Write 0x00493720

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 10 Offset = 0000

Prüfe: Zeile 2

Cache Miss (Compulsory)

	0	1	2	3				
0	1	1	1	1	?	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0	1	1	1	1	?	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0	0	0	0	0	?	?	?	?
1			0	0				
2				1				
	0	1	2	3				
0	1	1	1	1	?	?	?	?
1			0	0				
2				1				

Speicherzugriff 1: Write 0x00493720

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 10 Offset = 0000

Prüfe: Zeile 2

Cache Miss (Compulsory)

Eintrag 2 wird ersetzt (write-allocate)

	0	1	2	3
0		1	1	1
1			1	1
2				1

?	?	?	?	
?	?	?	?	
?	?	0000 0000 0100 1001 0011 0111 00	?	
?	?	?	?	

Speicherzugriff 2: Write 0x00493734

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 11 Offset = 0100

Prüfe: Zeile 3

Cache Miss (Compulsory)

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 1 0 1 1 0 2 0	?	?	0000 0000 0100 1001 0011 0111 00	?
0 1 2 3 0 1 1 1 1 0 0 2 1	?	?	?	?

Speicherzugriff 2: Write 0x00493734

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 11 Offset = 0100

Prüfe: Zeile 3

Cache Miss (Compulsory)

Eintrag 0 wird ersetzt

	0	1	2	3
0	1	1	1	1
1			1	1
2				1

?	?	?	?
?	?	?	?
?	?	0000 0000 0100 1001 0011 0111 00	?
0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 3: Read 0x00471110

Tag = 0000 0000 0100 0111 0001 0001 00 Index = 01 Offset = 0000

Prüfe: Zeile 1

Cache Miss (Compulsory)

	0	1	2	3				
0		1	1	1	?	?	?	?
1			1	1				
2				1				
0	0	1	2	3	?	?	?	?
1		1	1	1				
2				1				
0	0	1	2	3	?	?	0000 0000 0100 1001 0011 0111 00	?
1		0	1	0				
2				0				
0	0	1	2	3	0000 0000 0100 1001 0011 0111 00	?	?	?
1		0	0	0				
2				1				

Speicherzugriff 3: Read 0x00471110

Tag = 0000 0000 0100 0111 0001 0001 00 Index = 01 Offset = 0000

Prüfe: Zeile 1

Cache Miss (Compulsory)

Eintrag 0 wird ersetzt

	0	1	2	3
0	1	1	1	1
1			1	1
2				1

?	?	?	?

	0	1	2	3
0	0	0	0	0
1			1	1
2				1

0000 0000 0100 0111 0001 0001 00	?	?	?

	0	1	2	3
0	0	1	0	0
1			1	0
2				0

?	?	0000 0000 0100 1001 0011 0111 00	?

	0	1	2	3
0	0	0	0	0
1			0	0
2				1

0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 4: Read 0x00493728

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 10 Offset = 1000

Prüfe: Zeile 2

Cache Hit (Tag in Eintrag 2 passt (Zugriff 1))

	0	1	2	3				
0		1	1	1	?	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0		0	0	0	0000 0000 0100 0111 0001 0001 00	?	?	?
1			1	1				
2				1				
	0	1	2	3				
0		0	1	0	?	?	0000 0000 0100 1001 0011 0111 00	?
1			1	0				
2				0				
	0	1	2	3				
0		0	0	0	0000 0000 0100 1001 0011 0111 00	?	?	?
1			0	0				
2				1				

Speicherzugriff 4: Read 0x00493728

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 10 Offset = 1000

Prüfe: Zeile 2

Cache Hit (Tag in Eintrag 2 passt (Zugriff 1))

Eintrag 2 aktualisieren

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 0 0 1 1 1 2 1	0000 0000 0100 0111 0001 0001 00	?	?	?
0 1 2 3 0 0 1 0 1 1 0 2 0	?	?	0000 0000 0100 1001 0011 0111 00	?
0 1 2 3 0 0 0 0 1 0 0 2 1	0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 5: Read 0x0049352C

Tag = 0000 0000 0100 1001 0011 0101 00 Index = 10 Offset = 1100

Prüfe: Zeile 2

Cache Miss (Compulsory)

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 0 0 1 1 1 2 1	0000 0000 0100 0111 0001 0001 00	?	?	?
0 1 2 3 0 0 1 0 1 1 0 2 0	?	?	0000 0000 0100 1001 0011 0111 00	?
0 1 2 3 0 0 0 0 1 0 0 2 1	0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 5: Read 0x0049352C

Tag = 0000 0000 0100 1001 0011 0101 00 Index = 10 Offset = 1100

Prüfe: Zeile 2

Cache Miss (Compulsory)

Eintrag 3 wird ersetzt

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 0 0 1 1 1 2 1	0000 0000 0100 0111 0001 0001 00	?	?	?
0 1 2 3 0 0 1 1 1 1 1 2 1	?	?	0000 0000 0100 1001 0011 0111 00	0000 0000 0100 1001 0011 0101 00
0 1 2 3 0 0 0 0 1 0 0 2 1	0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 6: Read 0x00493734

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 11 Offset = 1100

Prüfe: Zeile 3

Cache Hit (Tag in Eintrag 0 passt (Zugriff 2))

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 0 0 1 1 1 2 1	0000 0000 0100 0111 0001 0001 00	?	?	?
0 1 2 3 0 0 1 1 1 1 1 2 1	?	?	0000 0000 0100 1001 0011 0111 00	0000 0000 0100 1001 0011 0101 00
0 1 2 3 0 0 0 0 1 0 0 2 1	0000 0000 0100 1001 0011 0111 00	?	?	?

Speicherzugriff 6: Read 0x00493734

Tag = 0000 0000 0100 1001 0011 0111 00 Index = 11 Offset = 1100

Prüfe: Zeile 3

Cache Hit (Tag in Eintrag 0 passt (Zugriff 2))

Eintrag 0 aktualisieren

0 1 2 3 0 1 1 1 1 1 1 2 1	?	?	?	?
0 1 2 3 0 0 0 0 1 1 1 2 1	0000 0000 0100 0111 0001 0001 00	?	?	?
0 1 2 3 0 0 1 1 1 1 1 2 1	?	?	0000 0000 0100 1001 0011 0111 00	0000 0000 0100 1001 0011 0101 00
0 1 2 3 0 0 0 0 1 0 0 2 1	0000 0000 0100 1001 0011 0111 00	?	?	?