

## Aufgabenblatt 9: Cache Verdrängungsverfahren

*LRU-Verfahren:* Bei einem Treffer in der oder bei einer Einlagerung in die  $i$ -te Cache-Line eines Set wird die  $i$ -te Spalte der LRU-Matrix mit '1' und die  $i$ -te Zeile mit '0' beschrieben. Muss Platz im Set geschaffen werden, wird die Cache-Line verdrängt, deren Zeile (binär interpretiert) die grösste Zahl (oder deren Spalte die kleinste Zahl) enthält.

Die LRU-Matrix ist eine quadratische Matrix mit der Dimension der Assoziativität.

### Aufgabe 1: LRU-Verfahren

Gegeben sei ein 4-fach assoziativer Cache mit insgesamt 32 Cache Lines á 1 Byte. Mit Hilfe des LRU-Algorithmus soll die Cache Line ausgewählt werden, in die die Daten eingelagert werden. Alle Sets besitzen eine eigene LRU-Matrix. Dargestellt ist hier nur die Historie der Matrix eines Sets! Überlegen Sie zuerst anhand der Zugriffsreihenfolge welches Set gemeint ist.

Spielen Sie dann die unten angegebene Zugriffsreihenfolge durch und geben Sie jeweils die Belegung der Matrix (nach dem Speicherzugriff) des LRU-Algorithmus und den Speicherort (welchen der Assoziativspeicher und darin welche Cache Line) an, in den der Wert eingelagert wurde.

Zugriffsreihenfolge auf Adresse: 0, 8, 16, 24, 16, 0, 32, 48

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 0

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 16

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 8

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 0

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 16

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 32

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 24

Cache Line	0	1	2	3
0				
1				
2				
3				

nach Zugriff auf Adresse 48

## Aufgabe 2: Programmierung

Im folgenden soll ein System mit einem Direct Mapped Cache betrachtet werden. Es sollen Daten einer Liste mit  $m$  Elementen in einer Schleife bearbeitet werden, wobei jedes Element 100 mal betrachtet werden muss ( $O(m \cdot 100)$ ).

Annahme: der Cache ist 16 KByte groß, der Code der inneren Schleife ist 1 KByte groß und jedes Element ist 512 Byte groß, eine Cache Line besteht aus nur einem Byte.

- a) Wie müssen die Schleifen geschachtelt sein, damit der Cache ausgenutzt wird? Überlegen Sie dazu, welche Daten im Cache gehalten werden können und welche zu einem Cache-Miss führen werden. Geben Sie die resultierenden Schleifen in einem Pseudo-Code an.
- b) Wie sollten Daten und Programm im Speicher angeordnet sein, damit der Cache optimal genutzt wird? Überlegen Sie dazu, wie die Speicherbereiche des Cache auf den Hauptspeicher abgebildet werden.
- c) Wie sollten die Daten und das Programm im Speicher angeordnet werden, wenn nun ein 4-fach assoziativer Cache verwendet wird?