Aufgabe 1: Grundlegende Cachebegriffe

a) Erklären Sie die folgenden Begriffe:
Write through, Write back, Write allocate, Write to memory

b) Welche Cache-Arten kennen Sie?
Was bedeuten die Begriffe ***virtuelle*** und ***reale*** Caches?

c) Sie haben einen Rechner mit einem Cache-Speicher zwischen CPU und Hauptspeicher. Die Zugriffe auf die jeweiligen Speicher benötigen

Zugriff	Anzahl Takte
Cache-Tag suchen	1
Aus dem Cache lesen	1
In den Cache schreiben	1
Aus dem Hauptspeicher lesen	5
In den Hauptspeicher schreiben	5

Ermitteln Sie für die verschiedenen Kombinationen der Strategien aus Aufgabe a) die jeweils benötigten Takte bei Read/Write-Hit/Miss.

d) Ermitteln Sie für folgendes Programm die jeweiligen Cache-Inhalte und die Interaktionen mit dem Hauptspeicher. Dabei soll die Cache-Architektur ein direct-mapped Cache mit 4 Cache-Lines sein. Eine Cache-Line ist 4 Bytes lang. Jede Operation arbeitet auf einem 16-Bit-Wort. Eine Adresse ist durch die verwendete Hash-Funktion in folgende Komponenten unterteilt:

Bits 4-31	Bits 2+3	Bits 0+1
Tag	Cacheline	Index in der Cacheline

```
Programm: load   2718
          load   271c
          load   271A
          store  271E
          store  271C
          load   1128
          load   112E
          store  0000
          store  0002
```

Lösung Aufgabe 1:

a) *Write-through* und *Write-back* sind Strategien zum Schreiben von Daten, die sich bereits im Cache befinden (*Cache-Hit*).

Bei Write-back wird der Schreibzugriff ausschließlich im Cache ausgeführt und dabei der Inhalt der entsprechenden Cache-Line als "dirty" markiert. Muß eine "dirty" Cache-Line ersetzt werden, so wird zunächst die Cache-Line in den Hauptspeicher zurückgeschrieben, bevor die zu ladenden Daten in den Cache transferiert werden können.

Bei einer Write-through-Strategie werden die Daten sowohl in den Cache als auch in den Hauptspeicher geschrieben. Der Schreibvorgang ist abgeschlossen, wenn die Daten in den Hauptspeicher geschrieben wurden.

Die Strategien *Write-allocate* und *Write-to-memory* werden eingesetzt, wenn die zu schreibenden Daten sich nicht im Cache befinden - der Schreibzugriff also einen *Cache-miss* verursacht.

Bei Write-to-memory wird der Schreibzugriff nur im Hauptspeicher durchgeführt, wobei der Inhalt des Caches sich nicht ändert.
Bei Write-allocate wird zunächst die passende Cache-Line geladen und anschließend ein normaler Schreibzugriff wie bei Cache-Hit durchgeführt.

b) Folgende Cache-Architekturen wurden vorgestellt:

- Direct-mapped Cache
- Two-way-associative Cache
- N-way-associative Cache
- Full-associative Cache

Bei virtuellen Caches werden virtuelle Adressen zur Indexierung innerhalb des Caches verwendet. Bei realen Caches dagegen werden physikalische Adressen eingesetzt, wodurch jeder Cache-Zugriff eine Adressauflösung durch die MMU benötigt.

c)

	Read		Write	
	Hit	Miss	Hit	Miss
write through+write allocate	2	6	5	10
write through+write memory	2	6	5	5
write back+write allocate	2	6/11	2	7/12
write back+write memory	2	6/11	2	6

d)

Befehl	write through		write back	
	w-a	w-m	w-a	w-m
load 2718	6	6	6	6
load 271C	6	6	6	6
load 271A	2	2	2	2
store 271E	5	5	2	2
store 271C	5	5	2	2
load 1128	6	6	6	6
load 112E	6	6	11	11
store 0000	10	5	7	6
store 0002	5	5	2	6

271	1111 2222

271	1111 2222
271	~~3333~~>2222 ~~4444~~>2222

112	5555 5555
271	2222 2222

112	5555 5555
112	6666 7777

000	~~8888~~>7777 ~~9999~~>7777
112	5555 5555
112	7777 7777

112	5555 5555
112	7777 7777