Aufgabe 1: Grundlegende Cachebegriffe

a) 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 1b) die jeweils benötigten Takte bei Read/Write-Hit/Miss.

Aufgabe 2: Virtuelle Caches

a) Was wird unter den Begriffen Ambiguities und Aliases verstanden?

b) Welche Verbesserungsmöglichkeiten gibt es bei virtuellen Caches?

c) Diskutieren Sie die Vorgehensweise bei der Verwaltung von Zugriffsrechten bei virtuellen Caches.

Lösung Aufgabe 2:

a) In den meisten Fällen benutzen write-back caches write-allocate, da der Mehraufwand

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

für Misses bei write-through konstant wäre.

Lösung Aufgabe 2:

a) Ambiguities (Mehrdeutigkeiten) treten auf, wenn *verschiedene physikalische* Adressen auf die *gleiche virtuelle* Adresse abgebildet werden. Die Folge ist, daß ein virtueller Cache diese physikalischen Adressen nicht unterscheiden kann und deshalb bei einem Zugriff auf die unterschiedlichen physikalischen Adressen nur die Daten einer Adresse liefert. Dieser Fall tritt potentiell bei jedem Prozesswechsel auf.

Aliase (Synonyme, Bedeutungsgleicheit) entstehen, wenn *verschiedene virtuelle* Adressen auf die *gleiche physikalische* Adresse abgebildet werden. Durch die unterschiedlichen virtuellen Adressen können die Daten in unterschiedlichen Bereichen des Caches abgelegt werden. Dadurch sind inkonsistente Zustände möglich. Dieser Fall tritt typischerweise bei Shared-Memory auf.

b) Verbesserungen:

1. Tags (z.B. Kachelnummern)
2. Prozeßschlüssel
3. Physikalische Adressen als Tags

d) Erzeugt der Zugriff auf einen virtuellen Cache einen Write-Miss, so können die Zugriffsrechte durch die MMU geprüft werden. Bei einem Write-Hit unter der Strategie Write-Through ist diese Vorgehensweise ebenfalls möglich, da auf jeden Fall in den Hauptspeicher geschrieben werden muß. Anders dagegen bei einem Cache mit Write-Back-Strategie: wird hier ein Write-Hit erzeugt, so wird bei gesetztem Modify-Bit davon ausgegegangen, daß die Daten der Cache-Line bereits schreibende Zugriffsrechte besitzen, und die Daten in den Cache geschrieben. Ist das Modify-Bit nicht gesetzt, kann nicht festgestellt werden, ob die Daten schreibende Zugriffsrechte besitzen - in diesem Fall muss über die MMU das Zugriffsrecht explizit geprüft werden. Eine alternative Möglichkeit besteht darin, das Zugriffsrecht zum Schreiben auf Daten in ein eigenes Bit der Cacheline zu übertragen. Dadurch kann im Cache selbst eine Überprüfung stattfinden.

Werden die Zugriffsrechte der Daten eines Prozesses während der Ausführung geändert, muß der Cache geleert werden, da sich noch Daten mit den alten Zugriffsrechten im Cache befinden können.

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

	Read		Write	
	Hit	Miss	Hit	Miss
write through+write allocate				
write through+write memory				
write back+write allocate				
write back+write memory				