

Aufgabe 1: Ihre Kreativität ist gefragt!

Um die Qualität der Lehrveranstaltung hoch zu halten, sind wir stets auf der Suche nach guten neuen Aufgabenstellungen. Bei dieser Aufgabe haben Sie die Möglichkeit, sich selbst eine derartige Aufgabenstellung zu überlegen, wobei die folgenden Voraussetzungen erfüllt werden müssen:

- Die Aufgabenstellung soll – dem Übungstag entsprechend – aus dem folgenden Stoffgebiet stammen:
 Montag – *Speicher*
 Mittwoch – *Schaltwerke*
 Donnerstag – *Mikroprozessoren*
 Freitag – *Pipelining*
- Für die Lösung der Aufgabenstellung sollte man ca. 15 Minuten benötigen.
- Es muss sich um eine neue Aufgabenstellung handeln. Eine abgeschriebene Lösung – sei es von einem alten Prüfungsbeispiel (egal von welcher LVA) oder von einer/einem anderen Studierenden – wird nicht akzeptiert! Werden bei einer bereits existierenden Aufgabenstellung nur Zahlen, Ausdrücke, Caching-Strategie, etc. variiert, gilt die Aufgabenstellung ebenfalls nicht als neu und wird somit nicht angerechnet.

Hinweis: Interessehalber werden diesmal wirklich alle abgegebenen Lösungen betrachtet.

Optional: Sie haben weitere Ideen für Übungsaufgaben? Wenn ja, schicken Sie uns diese bitte einzeln und inklusive Lösung an tgi-beispiele@auto.tuwien.ac.at – die besten Einsendungen werden mit einem Arbeitsplus belohnt! :-)

Aufgabe 2: Direct Mapped Cache

Gegeben ist ein Prozessor mit einer Adresslänge von 10 Bit und einer Datenwortlänge von 8 Bit.

Die Verwaltung des integrierten *Direct Mapped Caches* gliedert die Adressen des Prozessors in 5 Bit *Tag*, 2 Bit *Index* und 3 Bit *Offset*.

- Skizzieren Sie die Struktur des Caches. Wieviele Datenblöcke umfasst er und wieviele Datenwörter können pro Datenblock gespeichert werden? Überlegen Sie außerdem, welche Informationen zusätzlich zu den Daten für die Verwaltung des Caches gespeichert werden müssen.
- Zeigen Sie, wie sich die Verwaltungsdaten des gegebenen, anfangs leeren Caches verändern, wenn der Prozessor in der angegebenen Reihenfolge diese Speicherzugriffe ausführt:

$$(1F3)_{16} - (21D)_{16} - (1F4)_{16} - (3A7)_{16} - (1F5)_{16} - (0DD)_{16} - (21C)_{16}$$

Geben Sie *vor und nach* jedem Speicherzugriff die aktuell gespeicherten Werte von *Tag* und *Valid-Bit* im gesamten Cache an. Geben Sie außerdem zu jedem Speicherzugriff an, ob er ein *hit* oder *miss* ist.

Aufgabe 3: Full Associative Cache

Auf einem Prozessor mit einer Adresslänge von 8 Bit wird folgende Befehlssequenz ausgeführt (alle Konstanten sind im Hexadezimalsystem angegeben!):

```

1:      R0 ← 71
2:      R1 ← memory[(R0)+]
3:  memory[1A] ← R1
4:      R2 ← memory[(R0)+]
5:      R3 ← memory[(R0)+]
6:      R4 ← D8
7:  memory[(R4)] ← R0
8:  memory[7F] ← R2
9:      R2 ← memory[R4 + 6]
10:     R5 ← memory[-(R0)]

```

- a) Bestimmen Sie die Datenadressen, die sich aus dem Ablauf der Befehlssequenz ergeben, und tragen Sie diese in unten stehende Tabelle ein.
- b) Es wird angenommen, dass der Prozessor einen *Full Associative Cache* bestehend aus 2 Blöcken zu je 8 Datenwörtern besitzt. Die verwendete Ersetzungsstrategie ist *Least Recently Used*. Lese- und Schreibzugriffe sollen dabei gleich behandelt werden, d.h. der betreffende Datenblock wird in jedem Fall in den Cache geladen. Bestimmen Sie für jede Adresse *Tag* und *Offset*. Tragen Sie diese in die Tabelle ein und geben Sie außerdem an, ob der jeweilige Speicherzugriff ein *hit* oder ein *miss* ist. Tragen Sie in jeder Zeile den Zustand des Caches *nach* dem Speicherzugriff ein.

Zeit	Adresse	Tag	Offset	hit/ miss	Block 0			Block 1		
					Tag	valid	Letzter Zugriff	Tag	valid	Letzter Zugriff
0	—	—	—	—	00000	0	0	00000	0	0
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

- c) Berechnen Sie die *miss-rate* der oben angeführten Speicherzugriffssequenz.

Aufgabe 4: Peripherie - Monitor

Ein Monitor mit einer Auflösung von 2000x1000 Pixel verwendet für die Datenübertragung USB 3.0 mit einer maximalen Datenrate von 5 Gbit pro Sekunde ($1 \text{ Gbit/s} \hat{=} 1000 \text{ MBit/s}$). Für jeden Pixel wird je 1 Byte für die Farben rot, grün und blau sowie 1 Byte für die Helligkeit übertragen.

- a) Berechnen Sie die Datenrate, die benötigt wird, um 50 Bilder pro Sekunde zu übertragen.
- b) Wieviele Bilder können über USB 3.0 maximal pro Sekunde übertragen werden?
- c) Ist es möglich mit USB 3.0 3D-Bilder in dieser Auflösung zu übertragen (pro Auge und Sekunde 50 Bilder)?

Aufgabe 5: Interleaved Memory

Sie arbeiten auf einem System mit 8-fach *Interleaved Memory*. Die Adressen sind dabei so verschränkt, dass aufeinanderfolgende Speicherworte jeweils in einer anderen Speicherbank liegen. Die Geschwindigkeit in einem solchen System ist natürlich umso höher, je besser eine Speicherzugriffssequenz *Memory Interleaving* ausnutzt.

Geben Sie an, wie sich die Speicheradressen auf die einzelnen Bänke verteilen. Reichen Sie außerdem die folgenden Sequenzen nach der Geschwindigkeit, mit der die Datenwörter zur Verfügung stehen (absteigend von der schnellsten zur langsamsten). Begründen Sie, warum die einzelnen Sequenzen *Memory Interleaving* gut oder weniger gut ausnutzen.

Sequenz 1:	0	24	16	2	41	33	5	7
Sequenz 2:	9	10	11	12	17	26	35	44
Sequenz 3:	40	41	33	25	1	9	17	36
Sequenz 4:	16	17	25	26	27	35	36	37

Aufgabe 6: Speicherbandbreite

Speicherbandbreite ist ein oft verwendetes Maß für die Performance von Speicherbausteinen. Unter Speicherbandbreite (*Memory Bandwidth*) versteht man die Anzahl der gelesenen oder geschriebenen Bytes pro Zeiteinheit, wobei als Zeiteinheit entweder Sekunden oder Taktzyklen verwendet werden.

Beispiel: Ein System benötigt 128 Taktzyklen um 8 Byte aus dem Hauptspeicher zu übertragen. Entsprechend beträgt die Speicherbandbreite:

$$B = \frac{8}{128} = \frac{1}{16} \text{ Byte / Taktzyklus}$$

Nehmen Sie an, der Zugriff auf einen Speicherbaustein wird durch folgende charakteristische Zeiten bestimmt:

- 2 Taktzyklen für das Senden der Speicheradresse an den Baustein,
- 40 Taktzyklen für die Verarbeitung der Speicheranforderung im Speicherbaustein und
- 2 Taktzyklen zum Übertragen der Daten.

Somit dauert es $2 + 40 + 2 = 44$ Taktzyklen, um ein Speicherwort aus einem Speicherbaustein zu lesen.

Betrachten Sie die folgenden drei Speicherkonfigurationen, wobei *Speicherwortbreite* die kleinste im Speicherbaustein adressierbare und über den Datenbus übertragbare Einheit bezeichnet:

- Speicher mit einer Speicherwortbreite von 2 Byte (Abbildung a. – siehe nächste Seite)
- Speicher mit einer Speicherwortbreite von 8 Byte (Abbildung b.)
- 4-fach *Interleaved Memory* aus Bausteinen mit einer Speicherwortbreite von 2 Byte (Abbildung c.)

Führen Sie damit folgende Untersuchungen durch, wobei Sie die oben angegebenen charakteristischen Zeiten zugrundelegen:

- a) Berechnen Sie jeweils die Zeit (gemessen in Taktzyklen), die in den jeweiligen Speicherkonfigurationen notwendig ist, um 4 aufeinanderfolgende Worte zu je 2 Byte zu lesen. Berechnen Sie außerdem die erzielte Speicherbandbreite.
- b) Nehmen Sie an, dass die 4 Speicherworte zu je 2 Byte nicht aus unmittelbar aufeinanderfolgenden Adressen, sondern von den Wort-Adressen 0, 2, 4, 7 gelesen werden. Wie lange dauert es, bis alle Datenworte gelesen sind? Wie groß ist die Speicherbandbreite?

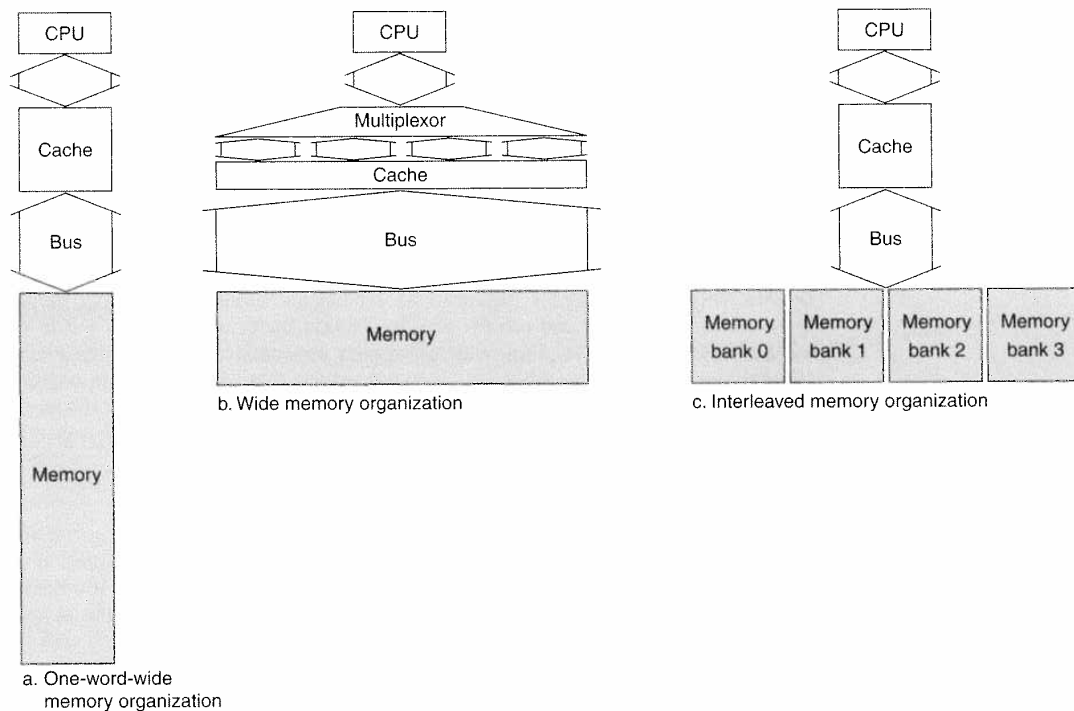


FIGURE 7.11 The primary method of achieving higher memory bandwidth is to increase the physical or logical width of the memory system. In this figure, memory bandwidth is improved two ways. The simplest design, (a), uses a memory where all components are one word wide; (b) shows a wider memory, bus, and cache; while (c) shows a narrow bus and cache with an interleaved memory. In (b), the logic between the cache and processor consists of a multiplexor used on reads and control logic to update the appropriate words of the cache on writes.

Bildquelle: Hennessy, Patterson, Computer Organization and Design, Morgan Kaufmann, 3rd Edition

Hinweis für Interessierte: In Wikipedia finden Sie unter http://en.wikipedia.org/wiki/List_of_device_bandwidths eine Liste mit realen Speicherbandbreiten. Weitere Quellen sind die Seiten der Speicherhersteller.

Aufgabe 7: Virtuelle Adressierung

Sie arbeiten auf einem Betriebssystem mit 32 Bit virtuellem Adressraum und 1 KB ($1 \text{ KB} \hat{=} 1024 \text{ Byte}$) großen Pages. Nehmen Sie an, dass 25% des adressierbaren Speichers im System tatsächlich physisch verbaut sind.

- Wie groß ist der Offset?
- Wieviel Speicher ist im System physisch verbaut?
- Wie lang sind die physikalischen Adressen?
- Wieviele Bit hat die Frame Nummer?
- Wie groß ist die Page-Table maximal, wenn jeder Eintrag noch zusätzlich 4 Bit an Zusatzinformation (z.B. *Permission Bits*) enthält?

Aufgabe 8: Virtual Memory

Gegeben ist ein System mit einem Prozessor, der einen Adressbereich von 16 KB ($1 \text{ KB} \hat{=} 1024 \text{ Byte}$) hat. Im System sind 16 KB Speicher physikalisch vorhanden. Die virtuelle Speicherverwaltung arbeitet mit 4 KB großen Page-Frames, wobei der physikalische Speicherbereich von 0 KB bis 4 KB dauerhaft belegt ist (z.B. durch das Betriebssystem) und daher für die übrigen Prozesse nicht zur Verfügung steht.

Sie starten auf dem System einen Prozess P , der in der nachfolgend angegebenen Reihenfolge auf den (virtuellen) Speicher zugreift: $(3E8F)_{16}$, $(0070)_{16}$, $(0123)_{16}$, $(20C3)_{16}$, $(1000)_{16}$ und $(10CC)_{16}$.

- Geben Sie die Struktur und Größe der Page Table und der virtuellen Adressen an. Geben Sie auch an, wieviele Einträge die Page Table besitzt und welche Informationen darin gespeichert werden.
- Geben Sie für jede der oben angegebenen (virtuellen) Adressen die Zuordnung zu den Zeilen der Page-Table an.
- Zeigen Sie, wie der Prozess P den Speicher nutzt, wenn P hintereinander die oben angeführten Adresszugriffe ausführt. Gehen Sie dabei von einer *Least Recently Used* Ersetzungsstrategie aus. Nehmen Sie außerdem an, dass die Page-Table am Beginn leer ist.

Stellen Sie dar, wie sich die Einträge der Page-Table in der zeitlichen Abfolge der angegebenen Speicherzugriffe verändert. Welche der Speicheraufrufe erzeugen einen Page-Fault, welche Pages werden in welchen Page-Frame des physikalischen Speichers geladen und welche Pages werden auf den Massenspeicher ausgelagert?

Aufgabe 9: Paging

Sie arbeiten mit einem Betriebssystem, das 8 Bit virtuelle Adressen verwendet. 4 Bit davon gehören zum Offset. Die physikalischen Adressen sind 6 Bit lang, es gibt also 4 Frames.

Es laufen 2 Programme parallel zueinander. Jedes Programm besitzt eine eigene Page-Table, die nach dem FIFO Prinzip arbeitet. Die Programme werden wie folgt ausgeführt (insgesamt 16 Takte):

P1,P2,P1,P2,P2,P1,P1,P1,P1,P2,P2,P2,P1,P1,P2,P2

Die Programme P1 und P2 greifen pro Abarbeitungsschritt (Dauer = 1 Takt) auf folgende virtuelle Adressen (Adressangaben sind hexadezimal) zu:

Schritt	P1	P2
1	A0	B1
2	00	01
3	01	03
4	11	10
5	FF	13
6	FE	FE
7	FC	FF
8	00	00

P1 greift also im 1. Takt auf die Adresse A0 zu, im 2. Takt greift P2 auf B1 zu. Danach ist wieder P1 mit 00 an der Reihe, usw.

- Tragen Sie in nachfolgender Tabelle (nächste Seite) ein, welche Page in welchem Frame liegt und zu welchem Programm sie gehört. Tritt ein Page-Fault auf, aktualisieren Sie den Inhalt des entsprechenden Frames (vgl. Beispiel). Liegt die Page bereits im physischen Speicher, tragen Sie beim entsprechenden Frame 'H' ein.

Beispiel: Nach dem ersten Takt liegt im Frame 0 die Page A von P1, im 2. Takt wird im Frame 1 die Page B von P2 abgelegt.

Takt	Schritt		Frame-Nr			
	P1	P2	0	1	2	3
1	1		P1: A			
2		1		P2: B		
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

b) Zeichnen Sie die Page-Tables von P1 und P2 nach Abarbeitung aller Schritte.

Aufgabe 10: Datenübertragung im Netzwerk

In einem Netzwerk sollen 1000 Byte Daten in 4 Paketen übertragen werden. Sie verwenden folgende Protokolle:

IP – Ethernet – TCP – HTTP

a) Ordnen Sie die Protokolle den Layern im OSI-Modell zu:

OSI-Layer	Protokoll
Network (2)	
Internet (3)	
Transport (4)	
Application (5-7)	

- b) Es gelten folgende Header-Größen: IP 20 Byte, TCP 24 Byte, Ethernet 16 Byte, HTTP 20 Byte. Wieviel Prozent Overhead (= Overhead / Gesamtpaketgröße) fällt pro Paket für die Datenübertragung an?
- c) Nehmen Sie an, Sie können die Pakete im Netzwerk mit 10 kb/s ($1 \text{ kb/s} \hat{=} 1000 \text{ b/s}$) übertragen. Wie lange dauert die Übertragung mit bzw. ohne Overhead? Wie hoch ist die Nutzdatenrate?
Hinweis: Die *Nutzdatenrate* gibt den Teil der gesamten Datenrate an, der nach Abzug von Overhead noch an tatsächlich nutzbarer Kapazität des Mediums für die eigentlichen Nutzdaten übrig bleibt.
- d) Sie verbinden sich von unterwegs aus über eine mittels VPN (*Virtual Private Network*) gesicherte WLAN Verbindung zum Netzwerk und können die Pakete mit 10kb/s übertragen. Welche Konsequenzen ergeben sich durch Ihre Verbindung in Bezug auf die Übertragungszeit?