

**Aufgabe 1: Cache-Adressierung**

Gegeben sei ein Prozessor mit einer Adresslänge von 12 Bit und einem 1 KiB großem Cache. Die Blockgröße des Caches beträgt 16 Byte. Die Adressierung findet auf Datenwortebene statt, wobei ein Datenwort 2 Byte groß ist.

- a) Geben Sie die Anzahl der Bits an, die für den *Index*, *Tag* und *Offset* benötigt werden, wenn der Cache ein *direct mapped cache* ist.
  
  
  
  
  
  
  
  
  
  
- b) Geben Sie die Anzahl der Bits an, die für den *Index*, *Tag* und *Offset* benötigt werden, wenn der Cache ein *4-way set associative cache* ist.
  
  
  
  
  
  
  
  
  
  
- c) Angenommen, die beiden Caches aus den Unteraufgaben a) und b) werden jeweils beim Verarbeiten (Lesen und Schreiben) von 10 GiB Daten genutzt, und angenommen, sie verwenden beide die gleiche Ersetzungsstrategie. Wo könnten Effizienzunterschiede zwischen den zwei Cache-Architekturen liegen? Begründen Sie Ihre Antwort zumindest mit der Suchzeit und der Hit-Rate.
  
  
  
  
  
  
  
  
  
  
- d) Warum werden bei der Adressierung des *fully associative caches* keine *Index*-Bits benötigt? Würden Sie einen *fully associative cache* wählen, wenn Ihnen die Suchzeit oder die Hit-Rate wichtiger ist? Begründen Sie Ihre Antwort.

## Aufgabe 2: Cache-Storage

Gegeben ist ein Prozessor mit einer Adresslänge von 8 Bit. Der integrierte *2-way set associative cache* besitzt 4 Sets und eine Blockgröße von 4 Byte. Die kleinste adressierbare Einheit beträgt 1 Byte. Bei einem schreibenden Zugriff wird bei einem *Hit* das Verfahren *copy-back* verwendet, bei einem *Miss* das Verfahren *fetch-on-write*.

- a) Wie groß ist der Cache? Geben Sie Ihre Antwort in Byte an.
- b) Angenommen, im Cache gibt es bereits folgende Einträge:

	Block 0							Block 1						
	<i>V</i>	<i>D</i>	<i>Tag</i>	<i>Data</i>				<i>V</i>	<i>D</i>	<i>Tag</i>	<i>Data</i>			
<b>Set 0</b>	1	0	0xC	0xA	0xB	0xC	0xD							
<b>Set 1</b>														
<b>Set 2</b>	1	1	0xE	0xE	0xF	0xE	0xF							
<b>Set 3</b>														

*Cache (V:= valid bit, D:= dirty bit)*

Auf dem Prozessor werden nun sequenziell folgende Speicherzugriffe ausgeführt ( $w := write, r := read$ ):

$r(0xAE) - r(0xA8) - r(0xAF) - w(0xB0) \leftarrow 0x1 - w(0xB0) \leftarrow 0x5 - r(0xC3)$

- (i) Tragen Sie in die obige Cache-Tabelle die resultierenden Verwaltungsinformationen ( $V, D, Tag$ ) sowie die Daten ein, wie sie **nach** der Zugriffssequenz im Cache vorliegen. Beachten Sie dabei die oben angegebene *policy* für schreibende Zugriffe.

Data	Address
0xE	0xA8
0xF	0xA9
0x0	0xAA
0x1	0xAB
0x2	0xAC
0x3	0xAD
0x4	0xAE
0x5	0xAF
0x6	0xB0
0x7	0xB1
0x8	0xB2
0x9	0xB3

*Auszug aus dem Speicher (vor der obigen Zugriffssequenz)*

- (ii) Wurde während der Zugriffssequenz jemals auf den Hauptspeicher geschrieben? Falls ja, wann und an welche Adresse(n)?
- (iii) Welches Datum wurde mit dem letzten Befehl  $r(0xC3)$  aus dem Cache gelesen?

### Aufgabe 3: Cache-Replacement Policy

Gegeben ist ein *fully associative cache*, der aus 4 Blöcken besteht und in einem Block 8 Datenwörter halten kann, wobei ein Datenwort 1 Byte groß ist. Die Adressierung erfolgt auf Datenwortebene. Die Adresslänge beträgt 12 Bit.

Als Ersetzungsstrategie gilt *least recently used* (LRU). Bei schreibenden Zugriffen wird bei einem *Hit* das Verfahren *write-through* (WT) verwendet und bei einem *Miss* das Verfahren *write-around* (WA).

Nehmen Sie an, dass die folgenden Speicherzugriffe  $t_0$  bis  $t_{11}$  sequenziell unter der Verwendung des Caches stattfinden:

$t_0$	r(0x23D)	$t_4$	r(0xE06)	$t_8$	r(0xE39)
$t_1$	r(0xCC3)	$t_5$	r(0xCC4)	$t_9$	w(0x303)
$t_2$	w(0x575)	$t_6$	w(0x23F)	$t_{10}$	r(0x576)
$t_3$	r(0x571)	$t_7$	r(0x317)	$t_{11}$	w(0x304)

- a) Tragen Sie in die Tabelle die jeweils aktuellen Verwaltungsdaten des Caches zum Zeitpunkt  $t$  ein. Geben Sie weiters an, ob es sich um einen lesenden oder einen schreibenden Zugriff handelt (r/w) und ob ein *Hit* oder ein *Miss* vorliegt (h/m). In die Spalte „pol“ tragen Sie die relevante Ersetzungsstrategie ein (LRU, WA, WT).

					Block 0		Block 1		Block 2		Block 3	
$t$	Adresse	r/w	h/m	pol	Tag	V	Tag	V	Tag	V	Tag	V
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												

$r$ := read,  $w$ :=write,  $h$ :=hit,  $m$ :=miss,  $V$ :=valid bit,  $pol$ := policy (LRU, WA, WT)

- b) Geben Sie die Hit-Rate des Caches nach der Zugriffssequenz an.

#### Aufgabe 4: Multi-Core Theorie

Bearbeiten Sie folgende Aufgabenstellungen zum Thema Multi-Core.

a) Wieso sind spezielle Maschinenbefehle für eine Synchronisation in einem Multi-Core-System notwendig?

b) Erklären sie in Ihren eigenen Worten das Beispiel *Summe von einem Array mit 100000 Zahlen* aus der Vorlesung.

**Require:**  $c \dots$  Anzahl der Cores,  $P_n = \text{CoreID} \in [0..c-1]$ ,  $A = \text{Array mit Zahlen}$ ,  $s$  Anzahl der Zahlen

**Ensure:** None

```
1:  $Sum[P_n] = 0$ ;  
2: for ( $i = c * P_n$ ;  $i < s/c * (P_n + 1)$ ;  $i++$ ) do  
3:    $Sum[P_n] = Sum[P_n] + A[i]$ ;  
4: end for  
5:  $half \leftarrow c$   
6: do  
7:    $synchron()$ ;  
8:   if ( $mod2(half) \neq 0 \wedge (P_n == 0)$ ) then  
9:      $Sum[0] += Sum[half - 1]$ ;  
10:  end if  
11:   $half = half/2$ ;  
12:  if ( $P_n < half$ ) then  
13:     $Sum[P_n] += Sum[P_n + half]$ ;  
14:  end if  
15: while ( $half > 1$ )
```

### **Aufgabe 5: Multi-core Prozessoren - Amdahl'sches Gesetz**

Nach dem Amdahl'schen Gesetz wird der Geschwindigkeitszuwachs bei paralleler Ausführung von Programmen vor allem durch den sequentiellen Anteil des Problems beschränkt, d.h. durch den Anteil, der sich nicht parallelisieren lässt.

Sei  $p$  der parallelisierbare Anteil eines Problems und  $c$  die Anzahl der Kerne eines Prozessors, so ist das Amdahl'sche Gesetz zur Abschätzung des Geschwindigkeitszuwachses  $s$  (*speedup*) gegenüber einem Kern:

$$s = \frac{1}{1 - p + \frac{p}{c}}$$

Betrachten Sie folgende Probleme mit ihrem parallelisierbarem Anteil  $p$  und vergleichen Sie den *speedup* für diese Probleme zwischen folgender Anzahl an Prozessorkernen  $c \in \{1, 4, 8, 32, 64\}$ .

a)  $p = 0.1$

b)  $p = 0.5$

c)  $p = 0.9$

d) Welche Tendenzen fallen Ihnen auf, wenn Sie die Ergebnisse aus den Unteraufgaben a), b) und c) interpretieren? Vergleichen Sie insbesondere den *speedup* bei 8 und bei 32 Kernen für  $p = 0.1$  und für  $p = 0.9$ . Wie können Sie den unterschiedlichen Geschwindigkeitszuwachs erklären? Was schließen Sie aus dem Vergleich des *speedups* für  $p = 0.1$  mit 32 Kernen und für  $p = 0.1$  mit 64 Kernen?

### **Aufgabe 6: Multi-core Prozessoren - Ausnutzung**

Sie wollen ein bereits existierendes Programm verbessern und möchten dieses teilweise parallel ausführen. Lösen Sie folgenden Aufgaben unter der Annahme, dass Sie die Synchronisierung und die Kommunikation zwischen den Kernen vernachlässigen können.

- a) Wie groß muss der parallelisierbare Anteil Ihres Programmes  $p$  für eine Performanceverdoppelung sein, wenn Sie  $c$  Prozessoren zur Verfügung haben? Wie hoch ist jeweils die Ausnutzung?

(i)  $c = 8$

(ii)  $c = 4$

(iii)  $c = 2$

- b) Wie groß muss der parallelisierbare Anteil Ihres Programmes  $p$  für eine dreifache Performancesteigerung sein, wenn Sie  $c$  Prozessoren zur Verfügung haben? Wie hoch ist jeweils die Ausnutzung?

(i)  $c = 8$

(ii)  $c = 4$

(iii)  $c = 2$

## Aufgabe 7: Netzwerke – Datenkapselung

In einem Netzwerk sollen mittels FTP 8200 Byte Nutzdaten übertragen werden. Sie verwenden folgende Protokolle:

$$UDP - IPv4 - Ethernet - FTP$$

Es gelten folgende Header-Größen: IPv4 24 Byte, UDP 16 Byte, Ethernet 14 Byte, FTP 20 Byte. Die Transportschicht beschränkt die Größe eines übertragenen Pakets (Nutzdaten + Header) auf maximal 1024 Byte. Alle darunterliegenden Schichten unterliegen keiner weiteren Beschränkung.

- a) Ordnen Sie die angegebenen Protokolle den Layern im TCP/IP-Referenzmodell zu. In Klammer sind die jeweils entsprechenden Schichten des OSI-Modells angegeben.

Layer	Protokoll
Application (5-7)	
Transport (4)	
Internet (3)	
Network (1-2)	

- b) In wie viele Einzelpakete muss die Transportschicht das ursprüngliche FTP-Paket aufteilen?

- c) Wie viel Prozent Overhead  $\left( = \sum \frac{\text{Größe Header}}{\text{übermittelte Daten}} \right)$  fallen bei der Übertragung an?

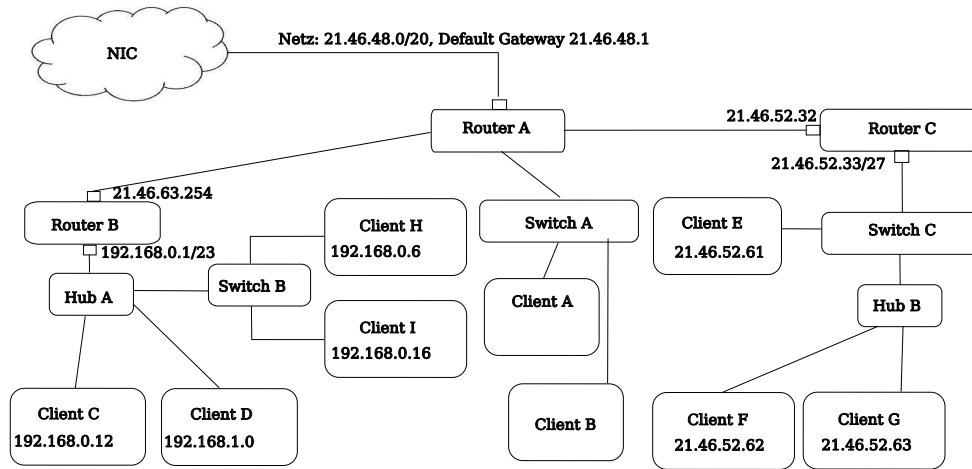
- d) Nehmen Sie an, Sie können die Pakete im Netzwerk mit 64 kb/s übertragen. Wie lange dauert die Übertragung? Wie hoch ist die Nutzdatenrate?

*Hinweis:* Die *Nutzdatenrate* gibt den Teil der gesamten Datenrate an, der nach Abzug des Overheads noch an tatsächlich nutzbarer Kapazität des Mediums für die eigentlichen Nutzdaten übrig bleibt.

- e) Sie verbinden sich über eine mittels VPN (*Virtual Private Network*) gesicherte WLAN-Verbindung zum Netzwerk und können die Pakete mit 64 kb/s übertragen. Welche Konsequenzen ergeben sich durch Ihre Verbindung in Bezug auf die Übertragungszeit?

## Aufgabe 8: Netzwerke – Analyse

Sie bekommen das folgende Netzwerk vom *Network Information Center* (NIC) zugewiesen:



Beantworten Sie folgende Aufgabenstellungen unter Zuhilfenahme obiger Grafik:

- Geben Sie die Netzadresse und die Broadcast-Adresse Ihres Netzwerks an.
- Geben Sie die Netzadresse und die Broadcast-Adresse des von Router B verwalteten Subnetzes an.
- Von Client F wird ein Paket zu Client B geschickt. An welche Netzwerk-Elemente wird dieses Paket übermittelt?
- Von Client E wird ein Paket zu Client I geschickt. An welche Netzwerk-Elemente wird dieses Paket übermittelt?
- Client H schickt ein Paket an die IP-Adresse 212.164.65.5. An welche Netzwerk-Elemente wird dieses Paket übermittelt?
- Wie viele IP-Adressen stehen noch für weitere Hosts im Teilnetz von Router C zur Verfügung?
- Finden Sie den/die Fehler in der Netzwerkkonfiguration.