

2. (____ / 11 Punkte) Interleavingsgraph

Der folgende Code der Threads T1 und T2 wurde erstellt, um zu erreichen, dass die Variable V entweder den Wert 1 oder den Wert 2 (und sonst keinen anderen Wert) erhält.

T1:

a: F1 := 1;

c: if F2 = 0 then V := 1;

T2:

b: F2 := 1;

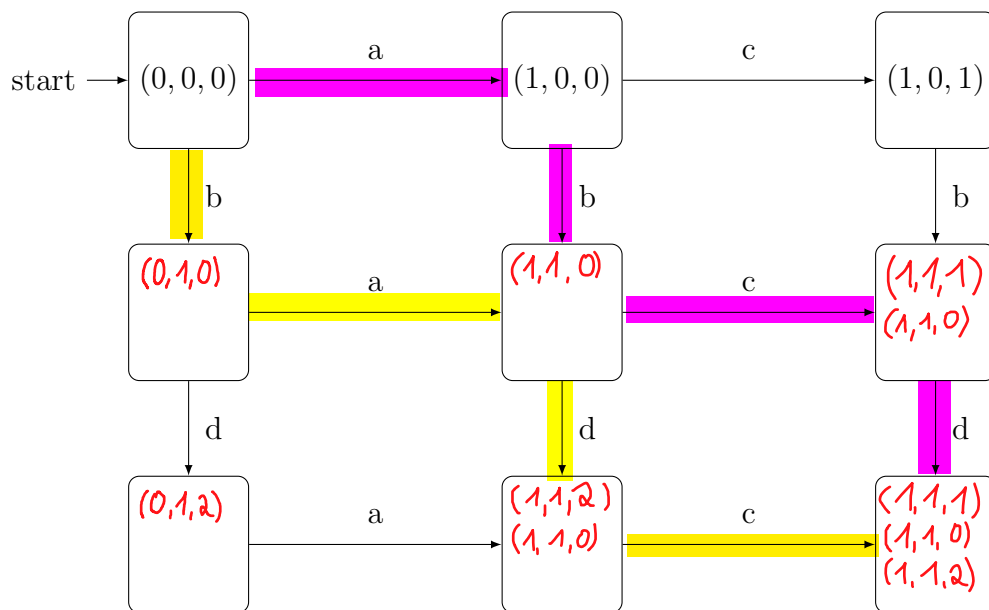
d: if F1 = 0 then V := 2;

Dabei sind F1 und F2 atomar, V ist nicht atomar. Alle drei Variablen sind mit 0 initialisiert.

- (a) Zeigen Sie, dass V auch andere als die gewollten Werte annehmen kann, indem Sie den unten stehenden Interleavingsgraphen vervollständigen (F1, F2, V).

Hinweis: Sie können von einem sequentially consistent Speichermodell ausgehen.

Atomar bedeutet unteilbar



In der Angabe steht, dass V nur de Wert 0 oder 1 haben darf. Wie man im rechten unteren Block gibt es zwei Interleavings, die diese Bedingung verletzen. In b muss man nun die Wege angeben, wie man zu den invaliden Werten kommt.

- (b) Wann tritt das ungewollte Verhalten auf?

■ a -> b -> c -> d
 ■ b -> a -> d -> c

3. (____ / 8 Punkte) Relaxation

Gegeben seien die Threads T1 und T2 in einem Release-Acquire-Speichermodell.

Hinweise: Der Einfachheit halber sollen alle Variablen, die mit A beginnen, atomar sein und alle anderen Variablen nicht-atomar. Es sollen jeweils vor und hinter den angegebenen Anweisungen keine weiteren Anweisungen vorhanden sein, die umgeordnet werden können. Alle Variablen sind mit 0 initialisiert.

davor	T_1
	a1: atomic_store(A1,1,Release);
	x: X := 21;
dahinter	a: atomic_store(A2,1,Release);
	T_2
	if1: if atomic_load(A2,Acquire)≠1 then goto if1;
	y: Y := X;
	if2: if atomic_load(A1,Acquire)≠1 then goto if2;
	z: Z := X;

(a) Welche der Store- und Load-Operationen können relaxed werden?

a1, if2

(b) Darf man in diesem Beispiel atomare Variablen in nicht-atomare umwandeln?
Wenn ja, welche?

A1 muss nicht atomic sein,
weil durch A2 gewährleistet wird,
dass auf A1 nicht gleichzeitig
gelesen & geschrieben wird

2. (_____ / 14 Punkte) Theorie

- (a) Wie heißt die Computer-Architektur, in der Programmspeicher und Datenspeicher voneinander getrennt sind?

Havard-Architektur

- (b) Wie nennt man die Lokalität ("Locality"), bei der gilt: Wird auf einen Speicherinhalt gerade zugegriffen, so ist es relativ wahrscheinlich, dass der nächste Zugriff in dessen Nachbarschaft erfolgen wird.

Spatial Locality

- (c) Welche der Strategien Write-Through, Move-Back, Copy-Back, Write-Around und Fetch-on-Write kann verwendet werden, wenn während eines Schreibzugriffes ein Miss auftritt?

Write-Around, *Write Around*

- (d) Wie nennt man Hazards, bei denen Nachfolgebefehl vom Ausgang des Sprunges abhängt? Welche Maßnahmen sind dagegen effektiv?

Massnahmen: Stall, Predication, Delayed Branching

Control Hazards

- (e) Welche der Abkürzungen *LIFO*, *LILO*, *FIFO*, *SUDO* und *FILO* beschreibt die Funktionsweise eines Stacks und wofür steht die Abkürzung?

LIFO (Last In First Out), *FILO*

- (f) Gegeben sind nachfolgende Instruktionen. Schreiben Sie jeweils rechts neben der Instruktion die verwendete Adressierungsart hin.

Hinweis: In der Vorlesung sind folgende Adressierungsarten betrachtet worden: Direct-Addressing Mode, Immediate Mode, Indirect-Addressing Mode, Register-Indirect Mode und Register Mode.

$R1 \leftarrow R2$	Register Mode
$R7 \leftarrow \text{memory}[\text{memory}[0x666]]$	Indirect-Addressing Mode
$R0 \leftarrow \text{memory}[0x500]$	Direct-Addressing Mode
$R3 \leftarrow -1$	Immediate Mode
$R3 \leftarrow \text{memory}[R4]$	Register Indirect Mode

9. (____ / 10 Punkte) Sie wollen ein bereits existierendes Programm P beschleunigen, indem Sie dieses teilweise parallel ausführen. Das Programm P hat folgende Eigenschaften:

- 10% von P sind nicht parallelisierbar.
- 20% von P können auf maximal zwei Cores (Prozessorkerne) parallel ausgeführt werden.
- Der restliche Teil kann ohne Einschränkungen auf beliebig vielen Cores ausgeführt werden.

(a) Wie viele Cores (Prozessorkerne) sind mindestens notwendig, um einen Speed-Up (Geschwindigkeitszuwachs) von 3 zu erreichen? Die Synchronisierung und Kommunikation zwischen den Cores soll vernachlässigt werden.

$$\frac{1}{\frac{0,1}{1} + \frac{0,2}{2} + \frac{0,7}{C}} = 3 \quad | \quad C = 5,25 \text{ (aufrunden)}$$

$$C = 6 \text{ Cores}$$

(b) Wie hoch liegt der theoretisch mögliche Speed-Up?

$$S = \frac{1}{\frac{0,1}{1} + \frac{0,2}{2} + \left(\frac{0,7}{\infty}\right)} = 0 \quad | \quad S = 5$$

(c) Wie hoch ist die Ausnutzung in Prozent? Runden Sie auf eine ganze Zahl.

$$\frac{\frac{1}{6}}{\frac{0,1}{1} + \frac{0,2}{2} + \frac{0,7}{6}} = 0,526 \approx 53 \%$$

(d) Wie können Sie eine Ausnutzung von 100% erreichen?

Wenn das ganze Programm komplett parallel ausgeführt

Technische Grundlagen der Informatik			Test 3 23.06.2017 90 Minuten Gruppe A
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben. Verwenden Sie keinen Tipp-Ex oder dergleichen. Unleserliche Antworten werden nicht gewertet!

Geben Sie bei Rechenaufgaben den **Lösungsweg** an!

Es sind keine Hilfsmittel zugelassen. Dies inkludiert Bücher, Mitschriften, Ausdrucke von Folien, Smartphones, Taschenrechner etc.

Zusatzblätter werden nicht akzeptiert!

Bei **Ankreuzfragen** werden Minuspunkte auf Teilaufgaben übernommen. Das Minimum je Gesamtaufgabe beträgt 0 Punkte.

1	[10]	[]
2	[12]	[]
3	[10]	[]
4	[8]	[]
5	[15]	[]
6	[10]	[]
7	[8]	[]
8	[7]	[]
9	[10]	[]
10	[10]	[]
Summe		[100] []

1. (10 Punkte) Beantworten Sie folgende Fragen zu Speicher, Caches und Pipelining.

(a) Im Zusammenhang mit Pipelining unterteilt man Hazards in *Data Hazards*, *Control Hazards* und *Structural Hazards*. Ordnen Sie jeder der folgenden Maßnahme zur Reduktion von Hazards jene Art von Hazards zu, die damit reduziert werden kann.

- Erweiterung der Hardware: [Structural Hazards](#)
- Data Forwarding: [Data Hazards](#)
- Branch Prediction: [Control Hazards](#)

(b) Ordnen Sie folgende Speichertechnologien absteigend (langsamster Speicher zuerst) nach ihren Zugriffszeiten: *Solid State Drive (SSD)*, *Dynamic RAM (DRAM)*, *Static RAM (SRAM)*, *Hard Disk Drive (HDD)*

1. HDD
2. SSD
3. DRAM
4. SRAM

- (c) Wodurch unterscheidet sich ein Computer mit einer *Von-Neumann-Architektur* von einem Computer mit einer *Harvard-Architektur*?

Von-Neumann-Architektur:

Der Speicher enthaelt sowohl Programme als auch Daten.

Harvard-Architektur:

Getrennter Programm- und Datenspeicher

Befehle und Daten koennen gleichzeitig geladen bzw. geschrieben werden

- (d) Gegeben sei ein Prozessor der Caching verwendet. Die Speicheradressen sind wie folgt strukturiert: 0xTTIIIOO. Jedes T, I und O ist jeweils Platzhalter für eine volle hexadezimale Ziffer. T steht für *Tag*, I für *Index* und O für *Offset*. Die Adressierung erfolgt Byte-weise und die Größe des Caches beträgt 64 KiB.

i. Wieviele Sets hat dieser Cache?

ii. Mit welcher Blockgröße arbeitet der Cache?

iii. Um welche Art von Cache handelt es sich (direct-mapped, N-way-set-associative, fully-associative)? Im Falle eines N-way-set-associative Caches geben Sie ein gültiges N an.

Technische Grundlagen der Informatik			Test 3 17.06.2016 90 Minuten Gruppe A
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben.
Unleserliche Antworten werden nicht gewertet!

Geben Sie bei Rechenaufgaben den **Lösungsweg** an!

Buch, Mitschriften, Ausdrücke von Folien, Handys,
Taschenrechner etc. sind nicht zugelassen!

Zusatzblätter werden nicht akzeptiert!

Bei **Ankreuzfragen** werden Minuspunkte auf Teilaufgaben
übernommen. Das Minimum je Gesamtaufgabe beträgt 0
Punkte.

1	[10]	[]
2	[8]	[]
3	[12]	[]
4	[12]	[]
5	[10]	[]
6	[10]	[]
7	[10]	[]
8	[8]	[]
9	[10]	[]
10	[10]	[]
Summe	[100]	[]

1. (10 Punkte) Überprüfen Sie folgende Aussagen auf Richtigkeit und kreuzen Sie entsprechend an!
(korrekte Antwort: +2 Punkte, inkorrekte Antwort: -2 Punkte, keine Antwort: 0 Punkte)

richtig falsch

- ☐ ☒ *um den Faktor k* Eine k -stufige Pipeline verbessert den Durchsatz im Idealfall um den Faktor $k/2$.
- ☒ ☐ Eine RISC-Architektur zeichnet sich durch einfache Befehle und wenige Adressierungsarten aus. *Reduced Instruction Set Computer*
- ☒ ☐ Control Hazards können durch Forwarding und Branch Prediction vermieden werden.
- ☒ ☐ Bei einem write-after-read (WAR) Data Hazard wird ein Register beschrieben, bevor der vorherige Befehl es gelesen hat.
- ☒ ☐ Interrupt bezeichnet eine asynchrone Unterbrechung des Programmablaufs, die zum Beispiel durch das Drücken einer Keyboard-Taste ausgelöst werden kann.