Wintersemester 2020/2021

Klausur zur Vorlesung
Maschinenmodelle
Teil 1 - Informationsverarbeitung

26. Februar 2021

Name, Vorname:

Matrikelnr.:

Aufgabe	1	2	3	4	5	6	Gesamt
Punkte							
Maximalpunktzahl	5	13	3	15	11	4	51

Aufgabe 1 Verständnisfragen (5 Punkte)

Kennzeichnen Sie Ihre Antwort eindeutig! Es ist immer nur eine Antwort richtig.

a) Welche Prozessoreinheit gibt vor, welcher Befehl als nächster ausgeführt wird?

i) der Adressbus
ii) die Systemuhr
iii) der Programmzähler
iv) das Statusregister

b) Welche Sprache erlaubt dem Programmierer, den Prozessor auf der Architekturebene zu programmieren?

i) Assembler
ii) Java
iii) Fortran
iv) C++

c) Ein Demultiplexer hat im Allgemeinen

i) einen Dateneingang, mehrere Datenausgänge, verschiedene Steuereingänge
ii) einen Dateneingang, einen Datenausgang, einen Steuereingang
iii) mehrere Dateneingänge und Datenausgänge, verschiedene Steuereingänge
iv) mehrere Dateneingänge, einen Datenausgang, verschiedene Steuereingänge

d) Welche Operation kopiert Daten **vom** Hauptspeicher in ein Register?

i) load
ii) store
iii) move
iv) add

e) Wie heißt eine der Konventionen zur Angabe der Kodierung eines Zeichens als Bitfolge?

i) DOS
ii) ASCII
iii) HTML
iv) ANSI

f) Es soll von der Speicheradresse 0x0040001A ein Datum in das Register \$5 geladen werden (Befehl *lw*). Im Register \$10 steht 0x00400000. Wählen Sie den richtigen Befehl!

i) $lw\ \$10, 0x1A(\$10)$

ii) $lw\ \$10, 0x1A(\$5)$

iii) $lw\ \$5, 0x1A(\$10)$

iv) $lw\ \$5, 0x1A(400000)$

g) Welche zwei Charakteristika des Speicherzugriffs werden durch Caches in Programmen ausgenutzt, sowohl auf Daten als auch auf Befehle?

h) Welche strukturelle Maßnahme kann die Trefferwahrscheinlichkeit für Caches mit direkter Abbildung erhöhen? Gilt dies auch für satzassoziative Caches?

Aufgabe 2 Einfacher Prozessor (9 + 4 Punkte)

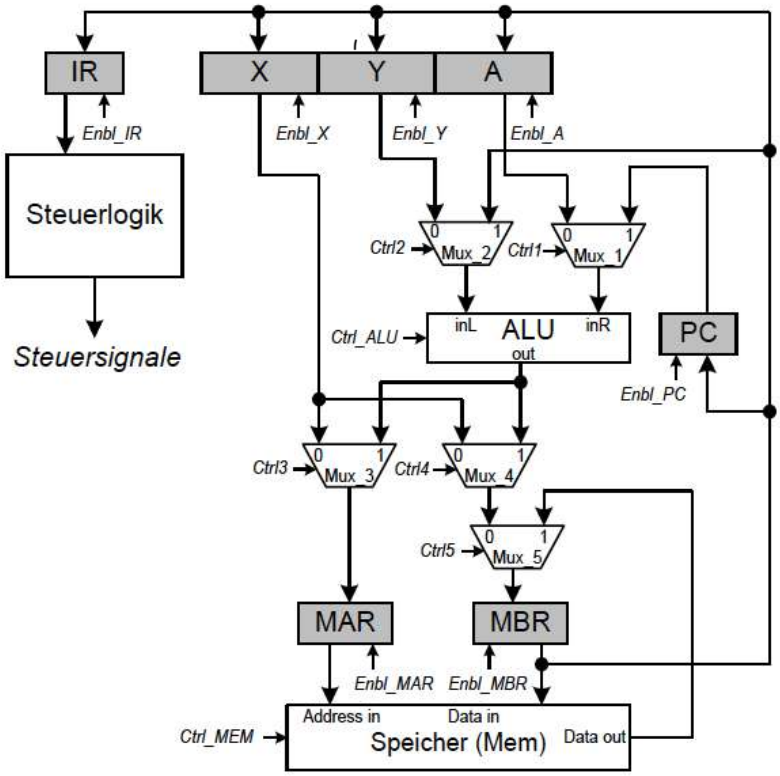

Abbildung 1: Daten- und Kontrollpfad eines fiktiven Prozessors. Sich berührende Leitungen haben nur dann eine Verbindung, wenn dies durch einen schwarzen Punkt am Berührungspunkt markiert ist.

Gegeben sind das Steuer- und Rechenwerk des Prozessors in Abbildung 1. Die ausgehenden Signale der Steuerlogik sind in der Abbildung nicht mit den Steuereingängen der Komponenten des Datenpfades verbunden, um die Abbildung übersichtlicher zu halten. Mit dem Präfix `Enbl` sind Ladesignale für Register bezeichnet, die, auf 1 gesetzt, dafür sorgen, dass das Register den anliegenden Wert bei einer steigenden Taktflanke speichert. Mit dem Präfix `Ctrl` sind die Steuersignale der ALU, des Speichers und der Multiplexer bezeichnet. Der Wert am Dateneingang i eines Multiplexers wird durch Anlegen des Binärwertes i am Steuereingang auf den Multiplexerausgang durchgeschaltet. Das Steuersignal für den Speicher kann die Werte r (aus dem Speicher lesen), w (in den Speicher schreiben), und n (keine Operation) annehmen. Die ALU mit ihren Eingängen inL, inR und dem Ausgang out unterstützt die Operationen:

- out := inR + 1, falls ctrl = 3,
- out := inL + inR, falls ctrl = 2,
- out := inL - inR, falls ctrl = 1,
- out := inR, falls ctrl = 0.

Für einen Assemblerprogrammierer sichtbare Register sind PC (Program Counter), A (Akkumulator), X und Y (Indexregister). Alle Register sind 8-Bit breit.

a) Welche der folgenden Mikrobefehle sind für diesen Datenpfad zulässig (Abarbeitung in einem Takt) und welche nicht? Tragen Sie für die gültigen Mikrobefehle die Werte der an den Datenpfad zu sendenden Steuersignale in der folgenden Tabelle ein, so dass dadurch die entsprechende Registertransferoperation durchgeführt wird. Lassen Sie für ungültige Mikrobefehle die Zeile leer.

Mikrobefehl	Steuersignale													
	Enbl-Signale für							Ctrl-Signale für						
	IR	X	Y	A	PC	MAR	MBR	Mux1	Mux2	Mux3	Mux4	Mux5	ALU	Mem
MAR <= X; MBR <= A														
MAR <= X; MBR <= Y														
X <= MBR; MBR <= X														
MAR <= Y; MBR <= A														
X <= MBR; Y <= MBR														
MBR <= MAR														
PC <= PC+1														

b) Geben Sie für obigen Datenpfad die Sequenz der Mikrooperationen für die Befehlsholphase an. D.h., der Befehlscode wird aus dem Speicher von der Adresse geholt, auf die der PC verweist, in das Register IR geladen und der PC wird um 1 inkrementiert.

Aufgabe 3 MIPS Assembler (3 Punkte)

Gegeben ist die folgende kurze Befehlssatzreferenz für einen 32-Bit MIPS Prozessor.

Befehl	Beschreibung
la *rs*, *label*	Lädt die durch *label* bezeichnete Adresse in das Register *rs*.
lbu *rs*,(*rt*)	Lädt ein Byte aus dem Speicher ohne Vorzeichenerweiterung in das Register *rs*.
addiu *rs*,*rt*,*imm*	Addition: $rs := rt + imm$.
subu *rd*,*rs*,*rt*	$rd := rs - rt$
sltu *rd*,*rs*,*rt*	if $rs < rt$ then $rd := 1$ else $rd := 0$
bnez *rs*,*label*	Springt zur Adresse *label*, falls $rs \neq 0$.
beqz *rs*,*label*	Springt zur Adresse *label*, falls $rs = 0$.
bgeu *rs*, *rt*, *label*	Springt zur Adresse *label*, falls $rs \geq rt$.
bltu *rs*, *rt*, *label*	Springt zur Adresse *label*, falls $rs < rt$.
bne *rs*, *rt*, *label*	Springt zur Adresse *label*, falls $rs \neq rt$.
beq *rs*, *rt*, *label*	Springt zur Adresse *label*, falls $rs = rt$.
b *label*	Unbedingter Sprung an *label*.

Abbildung 2: Befehlssatzreferenz

Implementieren Sie unter ausschließlicher Verwendung von Befehlen aus obiger Liste den Vergleich zweier vorzeichenloser 64-Bit Zahlen a und b. Der Wert von a befindet sich in den Registern \$$t0$ und \$$t1$, wobei \$$t0$ die niederwertigeren 32-Bit von a enthält. Der Wert von b befindet sich in den Registern \$$t2$ und \$$t3$, wobei \$$t2$ die niederwertigeren 32-Bit von b enthält. Ihr Programm soll zum Label *a_kleiner_b* verzweigen, falls $a < b$ ist und zum Label *a_nicht_kleiner_b*, sonst.

Aufgabe 4 Pipelining (7 + 4 + 4 = 15)

a) Gegeben ist folgendes MIPS-Programm-Fragment. (*Die Zeilennummerierungen sind nur als Hilfsmittel zu verstehen.* Wir gehen von einer 5 stufigen Pipeline (IF, ID, EX, MEM, WB) aus.

```
1    lw $1, 40($6)

2    add $2, $3, $1

3    add $1, $6, $4

4    sw $2, 20($4)

5    and $1, $1, $4
```

i) Es gibt kein Forwarding und keine Hazard-Detection. Markieren Sie in obigem Listing die Datenabhängigkeiten zwischen Befehlen, die zu Konflikten führen. Fügen Sie dann *nop*s ein, um die korrekte Pipeline-Abarbeitung zu sichern.

ii) Der Prozessor habe nun Forwarding. Es wurde aber vergessen, die Hazard-Detection-Unit zu implementieren. Was passiert, wenn obiger Code abgearbeitet wird?

b) Geben Sie Rechenzeit, Speed-Up und Effizienz einer Pipeline mit 10 homogen verteilten Stufen für 991 Instruktionen an! Die Dauer für die sequentielle Abarbeitung eines Befehls betrage 0,1 ms. Wir gehen davon aus, dass keinerlei Konflikte die Abarbeitung behindern!

Rechenzeit sequentiell:

Rechenzeit Pipeline:

Speed-Up:

Effizienz:

c) Eine Möglichkeit, Wartezyklen in der Pipeline zu vermeiden ist eine geschicktere Anordnung der Befehle. Geben Sie für die angegebene Befehlssequenz eine Anordnung an, mit der dieselbe Berechnung durchgeführt wird, aber für keinen Befehl seine Ausführung in der Pipeline verzögert werden muss.

```
mul $7, $10, $11 # Anweisung 1:  $7 :=  $10* $11
add $5, $3, $12  # Anweisung 2:  $5 :=  $3* $12
not $2, $10      # Anweisung 3:  $2 :=  neg $10
add $1, $5, $5   # Anweisung 4:  $1 :=  $5 + $5
or $3, $1, $2    # Anweisung 5:  $3 :=  $1 OR $2
and $3, $1, $7   # Anweisung 6:  $3 :=  $1 AND $0
```

Aufgabe _

Aufgabe 5 Cache (1 + 1 + 1 + 1 + 1 + 3 + 3 Punkte)

Folgende Tabelle zeigt einen (sehr kleinen) einfachen Cache (direct mapped) mit seinen Einträgen. *v* bezeichnet das Gültigkeitsbit (valid-Bit).

Cache-Adresse/Index	Cache–Inhalt		
	v	Tag-Bits	Datum
000	0	00011	0000
001	1	00011	0001
010	1	01001	0010
011	1	00011	0011
100	1	00011	1000
101	1	00011	1001
110	0	00010	1010
111	1	00010	1011

a) Wie groß ist der Cache in Bit? Wieviel Bits davon sind für die Verwaltung erforderlich?

Antwort:

b) Wie groß ist der Hauptspeicher in Bit?

Antwort:

c) Wie groß ist der größte zusammenhängende Hauptspeicherbereich in Byte, der vollständig im Cache gehalten werden kann?

Antwort:

e) Wie viele Blöcke des Hauptspeichers werden durch den Cache auf dieselbe Zeile abgebildet?

Antwort:

d) Tragen Sie in der Tabelle ein, was unter den gegebenen Adressen gegenwärtig im Hauptspeicher steht, oder schreiben Sie “unbekannt” in das Feld, wenn das aus den Cache–Angaben nicht hervorgeht. (Die Einträge im Cache seien korrekt.)

Hauptspeicher-adresse	Datum
00010 111	
00011 011	
00010 110	

e) Der Prozessor benötigt die Daten zu den folgenden Hauptspeicheradressen. Tragen Sie ein, ob die entsprechende Cache–Anfrage ein Treffer (hit) oder ein Fehlschlag (miss) ist.

Hauptspeicheradresse	hit oder miss
00011 000	
00011 111	
00011 100	

Aufgabe 6 Leistungsbewertung (2 + 2 = 4 Punkte)

Nehmen Sie an, dass mit **P1** und **P2** zwei Implementierungen eines Instruktionssatzes gegeben sind. Der Instruktionssatz setzt sich aus insgesamt fünf Klassen von Befehlen (**A**, ..., **E**) zusammen. Die Taktfrequenz von **P1** bzw. **P2** sowie die CPI-Werte der einzelnen Befehlsklassen sind wie folgt gegeben:

		CPI-Werte				
Implementierung	**Taktfrequenz**	**A**	**B**	**C**	**D**	**E**
P1	1,0 GHz	1	2	3	4	3
P2	1,5 GHz	2	2	2	4	5

a) Sei die *Peak Performance* eines Computers als die maximale Anzahl an Befehlen definiert, die der Computer bei einem entsprechenden Programm pro Sekunde abarbeiten kann. Geben Sie für **P1** und **P2** die Peak Performance an und charakterisieren Sie das jeweilige Programm verbal, bei dem die Peak Performance erreicht wird.

b) Seien in einem gegebenen Programm alle Befehlsklassen mit Ausnahme von Klasse **A** gleichverteilt. Befehle der Klasse **A** sind im Vergleich zu jeder einzelnen der anderen Klassen doppelt so oft vertreten. Welche der beiden Implementierungen ist schneller und um welchen Faktor?