Begründen Sie **alle** Antworten. Für Antworten ohne Begründung werden keine Punkte vergeben. Schreiben Sie Ihre Antworten in die vorgegebenen Lücken. In Aufgaben bei denen nach Assembler Code gefragt wird dürfen Sie nur Instruktionen von der beigelegten MIPS Referenz verwenden. Sie haben **2 Stunden** Zeit die Prüfung zu bearbeiten. Die maximale Punktzahl ist **100**.

Name: _____ Matrikelnummer: _____

Aufgabe	1	2	3	4	5	6	7	8
Total	10	10	17	12	10	16	14	11
Punkte								

1 Multiple-Choice Fragen (10 Punkte)

Korrekte Antwort: +1 Punkt, Falsche Antwort: -1 Punkt, Keine Antwort: 0 Punkte. Eine negative Punktzahl wird als 0 Punkte gerechnet.

(a) In einer MIPS Pipeline-Architektur mit Daten-Forwarding können die Eingangssignale der ALU aus jedem beliebigen Pipelineregister kommen, nicht nur aus ID/EX.
◯ Wahr ◯ Falsch

(b) Die Befehle `add`, `sub` und `or` sind alle vom Typ R.
◯ Wahr ◯ Falsch

(c) Bei der dynamischen Sprungvorhersage ist die 2-Bit Vorhersage immer besser als die 1-Bit Vorhersage.
◯ Wahr ◯ Falsch

(d) In einer MIPS Pipeline-Architektur ist die Ausführungszeit eines Programms genau gleich der Anzahl Befehle im Programm multipliziert mit der Taktzeit.
◯ Wahr ◯ Falsch

(e) CISC Befehle haben üblicherweise eine feste Grösse.
◯ Wahr ◯ Falsch

(f) SRAM muss periodisch aufgefrischt werden, um die Daten zu behalten.
◯ Wahr ◯ Falsch

(g) In einem vollassoziativen Cache können die Daten an einer beliebigen Position platziert werden.
◯ Wahr ◯ Falsch

(h) Prozessoren mit Mehrfachzuordnung (multiple issue) erlauben einen effektiven CPI < 1.
◯ Wahr ◯ Falsch

(i) In einem Prozessor mit Pipeline-Architektur werden Pipelineregister verwendet, um die Stufen der Pipeline zu isolieren.
◯ Wahr ◯ Falsch

(j) Bei der Ermittlung der E/A Bandbreite kann der Prozessor ein Engpass sein.
◯ Wahr ◯ Falsch

2 Speicher & Leistung (10 Punkte)

(a) (2 Punkte) Angenommen wir haben einen Prozessor mit einer Taktfrequenz von 2.4 GHz und einem perfekten CPI von 1.3. Ein Hauptspeicherzugriff benötigt 250 Taktzyklen. Wie viele Nanosekunden dauert jeder Speicherzugriff?

(b) (2 Punkte) 22% aller Befehle sind *store*/*load* Befehle welche separate Level 1 Caches für Daten und Befehle haben. Der Level 1 Datencache hat eine Fehlzugriffsrate von 3%. Der Level 1 Cache für Befehle hat eine Fehlzugriffsrate von 1%. Was ist der Gesamt-CPI?

(c) (3 Punkte) Angenommen wir fügen zwischen Level 1 Cache und Hauptspeicher einen Level 2 *unified* Cache ein mit einer Zugriffszeit von 10ns sowohl für Treffer wie auch Fehlzugriffe. Die Fehlerrate zwischen Level 2 Cache und Hauptspeicher is reduziert auf 0.1%. Was ist der Gesampt-CPI dieses Systems (Beachte: Der perfekte CPI ist 1.3)?

(d) (3 Punkte) Welcher Typ von RAM ist geeigneter für Level 2 Cache: SRAM oder DRAM? Geben Sie einen Vorteil sowie einen Nachteil von DRAM gegenüber SRAM an.

3 MIPS Eintaktdatenpfad (17 Punkte)

Betrachten Sie die unten abgebildete Eintakt Implementierung eines MIPS Prozessors. Nehmen Sie an, der Prozessor führt den Befehl `ori $r20,$r23,0xc1` aus. Das Register `$PC` hat den Wert `0x1004'0004`. Die folgende Tabelle beschreibt den Zustand des Registersatzes.

Registersatz	
Register	Inhalt
`$r20`	`0x0000'0035`
`$r21`	`0x0000'1042`
`$r22`	`0x0000'1040`
`$r23`	`0x0000'1018`

Beantworten Sie die untenstehenden Fragen gemäss dieser Implementierung und den gegebenen Registerinhalten. **Beachten Sie:** In den folgenden Aufgaben bezeichnet der Präfix `0x` eine Hexadezimalzahl.

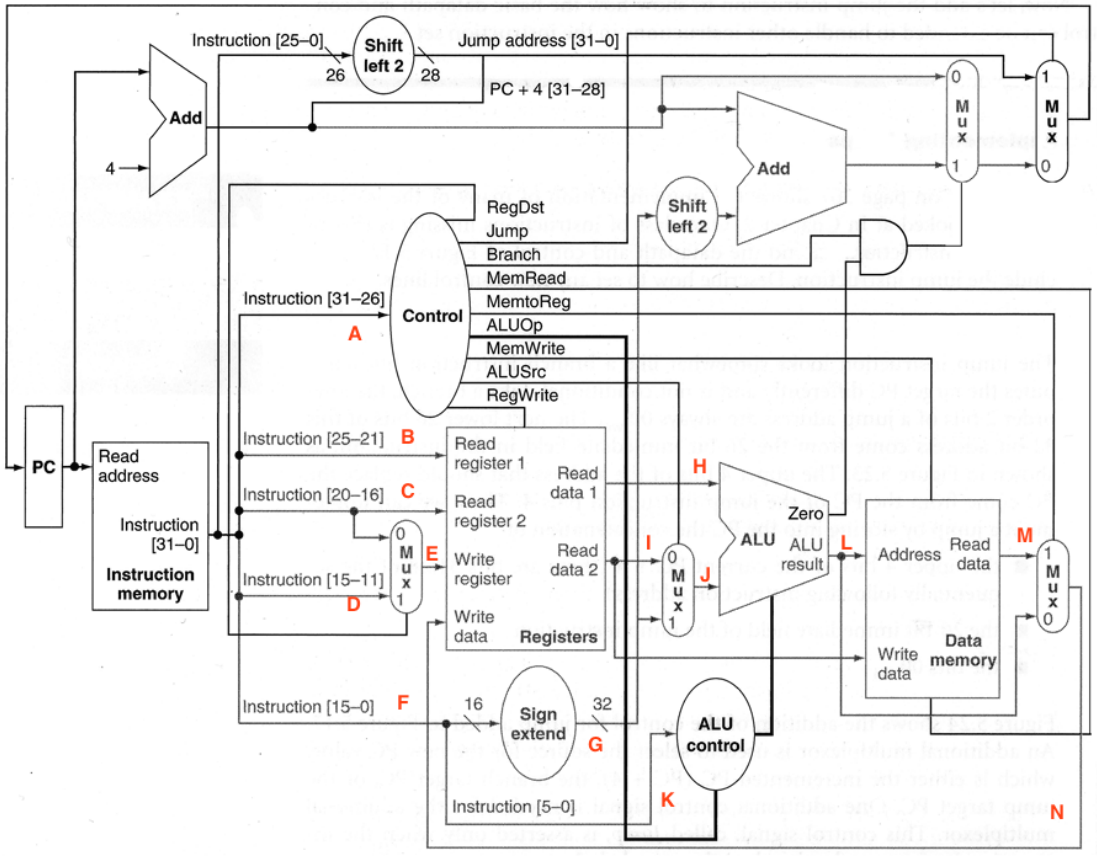
(a) (1 Punkt) Welchen Wert enthält das Register `$PC` nach Ausführung des Befehls.

(b) (5 Punkte) Wie wird der Befehl `ori $r20,$r23,0xc1` im Befehlsspeicher kodiert? Geben Sie die Binärdarstellung an und zeigen Sie dabei alle Schritte im Lösungsweg. Der Opcode von `ori` ist `0xd`.
Hinweis: `ori` ist ein Befehl vom Typ I. Die Syntax ist `ori $rt,$rs,imm` und die dabei ausgeführte Operation ist `R[rt] = R[rs] | imm`.

(c) (4 Punkte) Welchen Wert haben die Steuersignale `RegDst`, `Jump`, `Branch`, `MemRead`, `MemtoReg`, `MemWrite`, `ALUsrc` und `RegWrite`? Geben Sie falls nötig auch die "don't care" Terme an, und markieren Sie diese mit "X".

RegDst	**Jump**	**Branch**	**MemRead**
MemtoReg	**MemWrite**	**ALUsrc**	**RegWrite**

(d) (7 Punkte) Bestimmen Sie die Werte der mit Buchstaben markierten Leitungen in der Abbildung und tragen Sie diese in die Tabelle ein. Sie können Binär-, Hexadezimal-, oder Dezimalzahlen angeben. Falls vorhanden, markieren Sie undefinierte Signale mit "X".



A	B	C	D
E	**F**	**G**	**H**
I	**J**	**K**	**L**
M	**N**		

4 C Programmierung (12 Punkte)

(a) (8 Punkte) Schreiben Sie eine C Funktion, die die grösste Primzahl in einem Array zurückgibt. Wählen Sie einen angemessenen Wert, falls das Array keine Primzahlen enthält. Die Deklaration der Funktion sollte wie folgt sein:

```
int largestPrime(int array[], int size)
```

Sie können zudem annehmen, dass Ihnen die Implementation einer Funktion `int isPrime(int x) { ... }` zur Verfügung steht, welche 1 (true) zurückgibt wenn `x` prim ist, und sonst 0 (false).

(b) (2 Punkte) Warum müssen wir der oben beschriebenen Funktion die Grösse des Arrays als Argument übergeben?

(c) (2 Punkte) Welche Änderung würden Sie in der Deklaration vornehmen, damit die Funktion auf einem Integer-Pointer operiert?

5 MIPS (10 Punkte)

Das folgende MIPS Assemblerprogramm liest aus den Integer Arrays `a` und `b` und schreibt in das Integer Array `a`. Nehmen Sie an, die Basisadresse von `a` and `b` sei in den Registern `$t0` bzw. `$t1` gespeichert, und dass `sizeof(int) = 4`.

```
1  addi $t2, $zero, 4              // i = 4
2  addi $t3, $zero, 2
3  sw $t3, 0($t0)                  // a[0] =  _________________
4
5  loop:   lw $t3, 0($t0)          // lese a[0]
6
7          beq $t2, 16, end
8
9          addi $t2, $t2, 4        // i++
10
11         add $t4, $t2, $t1       // $t4 = i + Basisadresse von b
12         lw $t5, 0($t4)          // $t5 = b[i]
13         add $t3, $t5, $t3       // $t3 = b[i] + a[0]
14
15         add $t5, $t2, $t0
16         sw $t3, 0($t5)          // a[i] = _________________
17
18         j loop
19
20 end:    sw $zero, 0($t0)        // a[0] = _________________
```

(a) (3 Punkte) Füllen Sie die drei Lücken in den obigen Kommentarzeilen 3, 16, und 20 mit passendem C Pseudocode.

(b) (1 Punkt) Wie viele Iterationen durchläuft die Schlaufe?

(c) (6 Punkte) Übersetzen Sie das oben beschriebene MIPS Programm in C Code.

6 Pipelining & Konflikte (16 Punkte)

Gegeben sei folgendes MIPS Codestück.

```
1         addi $t0, $zero, 0
2         addi $t3, $zero, 24
3         lw $t1, 0($t2)
4   Loop:
5         beq $t0, $t3, Load
6         addi $t0, $t0, 8
7         sw $t1, 0($t4)
8         add $t4, $t4, $t3
9         j Loop
10  Load:
11        lw $t1, 4($t2)
```

(a) (2 Punkte) Wie oft wird der Befehl `addi` in Zeile 6 in die Pipeline geladen, wenn wir keine Sprungvorhersage machen (nur Leerläufe)?

(b) (4 Punkte) Zeichnen Sie die Zeittafel für die Zeilen 1-8 (eingeschlossen) im ersten Durchlauf der Schleife (einschliesslich aller Leerläufe). Das Codefragment wird in einer einfachen Pipeline ausgeführt, die kein Forwarding macht, simultane Schreib-/Leseoperationen (im gleichen Taktzyklus) unterstützt, den Sprungbefehl in der ALU berechnet und die Hardware zum Sprungentscheid in der DM Stufe hat (keine Sprungvorhersage). Wie viele Taktzyklen werden für diese Zeilen gebraucht?

instruction	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

(c) (2 Punkte) Wie oft wird der Befehl `addi` in Zeile 6 in die Pipeline geladen, wenn wir eine statische Sprungvorhersage mit "Sprung wird nicht ausgeführt" hätten?

(d) (4 Punkte) Treten im oben beschriebenen MIPS Code Steuerkonflikte auf? Falls ja, wo treten diese auf und was ist die Ursache?

(e) (4 Punkte) Welche Art von dynamischer Sprungvorhersage wäre in diesem Codesegment effizienter: 1-Bit oder 2-Bit Vorhersage, angenommen die Vorhersage sei initialisiert auf "Sprung wird ausgeführt"? Begründen Sie Ihre Antwort.

7 Cache (14 Punkte)

(a) (6 Punkte) Zählen Sie drei Möglichkeiten auf, den Cache zu verbessern.

(b) (4 Punkte) Was sind die Vorteile einer Speicherhierarchie in Bezug auf *temporale Lokalität* und *räumliche Lokalität*?

(c) (4 Punkte) Was geschieht, wenn die Blockgrösse in einem Cache zu gross ist relativ zur Cachegrösse?

8 Sprungvorhersage (11 Punkte)

Angenommen Sie haben ein kleines Programm A, welches einen einzigen Sprungbefehl beinhaltet. Sie führen das Programm aus und beobachten, dass die Sprungbefehle wie folgt ausgewertet werden (T steht für "Sprung ausgeführt" und N steht für "Sprung nicht ausgeführt"):

T N T N T T T T T N

(a) (1 Punkt) Angenommen eine dynamische 1-Bit Sprungvorhersage wird verwendet. Was ist die beste Wahl für den Startwert des Vorhersage-Bits in Programm A?

(b) (6 Punkte) Sie bekommen jetzt zusätzlich zwei Programme B und C, die beide äquivalent zu Programm A sind aber ein anderes Sprungverhalten aufweisen. Sie müssen bestimmen, welches Programm am besten für eine (dynamische) 2-Bit Vorhersage geeignet ist, wenn diese auf den Startwert "Sprung wird sicher nicht ausgeführt" (00) eingestellt ist. Berechnen Sie die Genauigkeit der Vorhersage für jedes Programm, indem Sie die Tabellen unten ausfüllen.

Programm A										
Sprung:	T	N	T	N	T	T	T	T	T	N
Vorhersage:										
Programm B										
Sprung:	N	T	T	T	T	T	N	T	N	T
Vorhersage:										
Programm C										
Sprung:	N	T	N	T	N	N	N	N	N	T
Vorhersage:										

Genauigkeit		
A	B	C

(c) (4 Punkte) Welchen Zweck erfüllt der Sprungzielpuffer (BTB) in der dynamischen Sprungvorhersage und warum braucht es diesen, wenn es ja schon den Sprungvorhersagepuffer gibt?