Datum: 23.12.2010 **Name:** _____ **Vorname:** _____

Arbeitszeit: 60 Minuten **Matr.-Nr.:** _____

Hilfsmittel: alle eigenen **Unterschrift:** _____

wird vom Prüfer ausgefüllt

1	2	3	4	5	6	7	8	9			Σ

Diese Probeklausur hat **reduzierten Umfang** (60 statt 90 Minuten in der Prüfung). Sie deckt auch nicht alle möglichen Themen ab, die in der richtigen Klausur auftauchen können.

1. Von-Neumann-Architektur (7/66 Punkte)

a) Der Universalrechner (von-Neumann-Rechner) besteht aus fünf Komponenten. Nennen Sie diese fünf Komponenten. (1 P)

b) Für ein Programm müssen der Programmcode und die Programmdaten im Speicher vorgehalten werden. Verwenden von-Neumann-Rechner gemeinsame oder getrennte Speichersysteme für Code und Daten? (2 P)

Nennen Sie jeweils einen Vor- und einen Nachteil von getrennten Speichersystemen. (2 P)

c) Was ist die „Pseudo-Harvard-Architektur“? (2 P)

2. RISC vs. CISC (6/66 Punkte)

Im Folgenden finden Sie sechs fiktive Befehle, deren jeweilige Funktion neben dem Kommando beschrieben ist.

1) INCR2 Reg — erhöht den Wert im angegebenen Register um 2

2) INCR2 Mem — erhöht den Wert im Hauptspeicher (an der angeg. Adresse) um 2

3) SWAP Mem1,Mem2 — vertauscht die Inhalte der Speicherzellen Mem1, Mem2

4) CHECK Reg,o1,o2 — Prüft Inhalt des angegebenen Registers.
– Falls Inhalt < 0: relativer Sprung um o1 Byte vorwärts
– Falls Inhalt > 0: relativer Sprung um o2 Byte vorwärts
– Falls Inhalt = 0: fortfahren an nächster Adresse

5) SAVEALL Mem — schreibt den Inhalt sämtlicher Register in den Hauptspeicher (ab Adresse Mem)

6) LDBLK Reg,Mem — liest einen durch Reg festgelegten 1 KB großen Block von der Festplatte und speichert ihn im Speicher ab Adresse Adr

Kreuzen Sie in der folgenden Tabelle an, welche dieser Befehle es auf einer RISC- bzw. CISC-Architektur geben kann. (Es muss keine konkrete CPU geben, die den jeweiligen Befehl bietet.)

	RISC	CISC
1) INCR2 Reg		
2) INCR2 Mem		
3) SWAP Mem1,Mem2		
4) CHECK Reg,o1,o2		
5) SAVEALL Mem		
6) LDBLK Reg,Mem		

3. Abhängigkeit (7/66 Punkte)

Zeichnen Sie für folgendes Programm den Abhängigkeitsgraph (ohne transitive Pfeile, ohne RAR-Abhängigkeiten). (5 P)

```
1     a     IS    $0
2     b     IS    $1
3     t     IS    $2
4     y     IS    $3
5           MUL   t,a,a
6           SUB   t,t,b
7           SUB   t,t,b
8           MUL   y,b,b
9           SUB   y,y,t
```

Geben Sie außerdem an, welche Funktion das Programm berechnet, wenn a und b Argumente, y das Ergebnis und t ein Zwischenwert sind. (2 P)

$y = f(a,b) =$

4. Superskalarität (8/66 Punkte)

a) Was bedeutet „superskalar“ im Vergleich zu „skalar“? (2 P)

b) Eine CPU-Pipeline bestehe aus 100 Stufen. Fast alle Stufen benötigen zum Erledigen ihrer jeweiligen Aufgabe ca. 1 ns, aber drei der Stufen benötigen ca. 4 ns.

(i) Warum ist das kein verfünftiges Design? (2 P)

(ii) Wie lässt sich relativ leicht ein vernünftiges Design aus der bestehenden 100-Stufen-Pipeline entwickeln? (2 P)

c) In der 5-stufigen RISC-Pipeline kann es zu Kollisionen kommen. Was ist eine Kollision und welche Kollisionen sind hier möglich? (2 P)

5. Floating-Point-Zahlen (9/66 Punkte)

Beim Umrechnen von Brüchen in Floating-Point-Zahlen kommt es oft zu Rundungsfehlern. Welche der folgenden Zahlen kann eine CPU exakt als FP-Zahl speichern? Geben Sie zu jeder Zahl eine **kurze Begründung** an!

	exakt	nicht ex.	Begründung
a) 1,0			
b) 0,1			
c) 1/3			
d) 0,25			
e) 1/12			
f) 0,35			

6. Stack-Maschine (5/66 Punkte)

Schreiben Sie ein Programm für die Stack-Maschine, das die folgende Funktion f berechnet:

$f(a,b,c) = (a^2+b^2+c^2) / (a+b+c)$

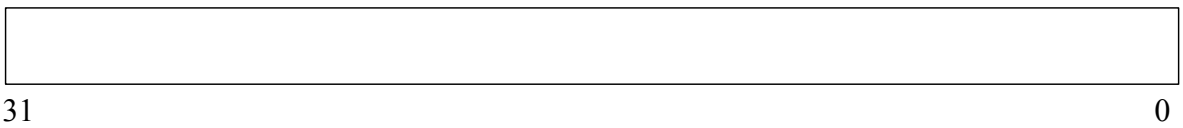
Dabei können Sie davon ausgehen, dass a, b, c bereits auf dem Stack liegen (der erste POP-Aufruf holt c vom Stack). Sie können Speicheradressen ab 0x1000 frei zum Speichern von Zwischenergebnissen verwenden. Das Endergebnis soll nach Ende Ihres Codes auf dem Stack liegen.

7. Virtuelle Adressen / Paging (10/66 Punkte)

Eine CPU arbeitet mit folgenden Werten:

- Seitengröße: 4 KByte
- 32 Bit lange virtuelle Adressen
- Seitentabelleneinträge sind 4 Byte lang

a) Wie ist eine virtuelle Adresse aufgebaut (welche Bits der Adresse haben welche Bedeutung)? (2)



Zeichnen Sie die die Unterteilung hier ein und beschriften Sie die Abschnitte geeignet.

b) Wie groß ist die gesamte Seitentabelle? (5 P)

c) Welche Aufgabe hat ein **Translation Look-Aside Buffer** (TLB)? Dank welchen Prinzips ist er in der Regel auch dann hilfreich, wenn er sehr klein ist? Erklären Sie Ihre Antwort. (Es reicht nicht aus, den Namen des Prinzips aufzuschreiben.) (3 P)

8. Seitenersetzung (5/66 Punkte)

a) Was ist ein Page Fault? (1 P)

b) Warum will man Page Faults vermeiden? (2 P)

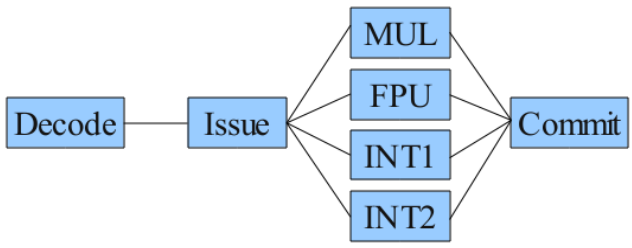
c) Welchen Vorteil bietet bei der Seitenersetzung die LRU-Strategie gegenüber der FIFO-Strategie? (2 P)

9. Reorder Buffer (9/66 Punkte)

Betrachten Sie die folgenden vier Befehle:

```
1    FADD $5,$1,$2
2    FSUB $6,$1,$2
3    FMUL $1,$5,$6
4    FADD $1,$1,1
```

Die Befehle FMUL, FADD und FSUB benötigen jeweils vier Zyklen auf der MUL- bzw. FPU-Ausführungseinheit.



Zeigen Sie, wie diese Befehle auf einem System mit Reorder Buffer (zwei Zuteilungen pro Takt) ausgeführt werden; zeichnen Sie dazu die Inhalte des Reorder Buffers zu den 17 Zeitpunkten t_0 bis t_{16} ein.

0

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

1

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

2

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

3

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

4

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

5

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

6

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

7

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

8

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

9

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

10

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

11

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

12

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

13

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

14

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

15

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

16

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis