

Kurs 01609 Computersysteme II

Einsendearbeit 2

Aufgabe 1 (5 Punkte)

Entscheiden Sie, welche der folgenden Aussagen über Prozessortechniken korrekt sind.

- ☐ a) Skalare Prozessoren erreichen die Ausführung von nahezu einem Befehl pro Takt, superskalare Prozessoren erreichen dagegen die Ausführung von mehreren Befehlen pro Takt.
- ☐ b) Signalprozessoren arbeiten u.a. mit Mehrfachzuweisungstechniken, bei denen die Superskalartechnik, die VLIW- sowie die EPIC-Technik zum Einsatz kommt.
- ☐ c) Ein superskalarer RISC-Prozessor besteht u.a. aus einer Befehlsholeeinheit, einer Dekodiereinheit, einer Zuordnungseinheit, mehreren Ausführungseinheiten, On-board-Caches, Hauptspeicher (HSP), sowie verschiedenen Registern.
- ☐ d) Bei einem VLIW-/EPIC-Prozessor können wie auch bei einem skalaren Prozessor mehrere Befehle pro Takt zugewiesen werden, diese müssen jedoch vorher durch einen Compiler entsprechend gebündelt werden.
- ☐ e) Während die Superskalartechnik eine Mikroarchitekturtechnik ist, handelt es sich bei der VLIW-Technik um eine eigene Architekturtechnik, was zur Folge hat, dass für jeden Prozessor ein spezieller nicht portabler Befehlskode generiert werden muss.

Aufgabe 2 (5 Punkte)

Entscheiden Sie, welche der folgenden Aussagen über Pipelining, Skalar- und Superskalarprozessoren korrekt sind.

- ☐ a) Wesentliches Merkmal eines superskalaren Prozessors ist, dass aus einem sequentiellen Strom von normalen Befehlen mehr als ein Befehl pro Takt durch die Hardware in Form eines komplexen Schedulers den Ausführungseinheiten dynamisch zugewiesen werden kann, wobei die Zuweisungsbreite maximal der Anzahl der verfügbaren Ausführungseinheiten entspricht.
- ☐ b) Befehls-Pipelining als auch Superskalartechnik nutzen die sog. feinkörnige Parallelität, wobei das Befehls-Pipelining diese Parallelität ausschließlich zeitlich ausnutzt und die Superskalartechnik zeitlich und räumlich.
- ☐ c) Eine superskalare Pipeline benötigt gegenüber einer skalaren Pipeline eine Zuordnungsstufe und eine Rückordnungs- sowie Rückschreibstufe.
- ☐ d) Programmcode, der für einen skalaren Mikroprozessor generiert wurde, kann ohne Änderung auch auf einem superskalaren Prozessor mit der gleichen Architektur ablaufen und umgekehrt.
- ☐ e) Eine superskalare Pipeline besteht u.a. aus der In-Order-Section mit der Befehlsholestufe und der Befehlsdekodier- und Registerumbenennungsstufe, in der externe Registerbezeichnungen durch interne physikalische Adressen ersetzt werden.

Aufgabe 3 (7 Punkte)

Entscheiden Sie, welche der folgenden Aussagen über die Abläufe in Skalar- und Superskalarprozessoren korrekt sind.

- ☐ a) Tritt bei einem superskalaren RISC-Prozessor ein Cache-Miss auf, dann kann ohne Blockierung weiterer Speicher-Zugriffe auf den Hauptspeicher (HSP) zugegriffen werden.
- ☐ b) Durch die Verwendung eines Primär-Cache-Speichers nach dem Prinzip einer Harvard-Architektur können Daten- und Strukturkonflikte auf dieser Ebene vollständig vermieden werden.
- ☐ c) Im Falle eines durch einen Cache-Miss verursachten Zugriffs auf den Hauptspeicher (HSP) kann bei einem nicht-blockierenden Cache parallel dazu der nächste Cache-Zugriff erfolgen.
- ☐ d) Durch die Umbenennung von Operanden- und Ergebnisregister in der Dekodierstufe sind die Befehle anschließend frei von Lese-/Schreib- sowie Schreib-/Lese-Konflikten.
- ☐ e) Nach der Behandlung der Struktur- oder Ressourcenkonflikte können echte Datenkonflikte durch Zuweisung der Befehle in die Umordnungspuffer der jeweiligen Ausführungseinheit gelöst werden, wobei natürlich keine neuen Strukturkonflikte entstehen können.
- ☐ f) Logische bzw. virtuelle Speicheradressen, bei denen die entsprechenden Speicherzellen unter Umständen auf einer Festplatte untergebracht sind, werden beim Zugriff von der MMU in physikalische Speicheradressen umgerechnet.
- ☐ g) Als Dispatch bezeichnet man eine Pipeline-Stufe, in welcher der Übergang vom Warten im Umordnungspuffer zur Ausführung vollzogen wird.

Aufgabe 4 (7 Punkte)

- ☐ a) Bei einem Sprungbefehl wird während der Dekodierphase (ID), d.h. sobald dieser als Sprungbefehl identifiziert wurde, auf den BTAC (Branch Target Address Cache), auf die BHT (Branch History Table) und einen Kellerspeicher, in dem die Rücksprungadresse abgelegt wird, zugegriffen.
- ☐ b) Wenn eine Unterbrechung auftritt, werden die Resultate aller Befehle, die in der Programmreihenfolge vor diesem Ereignis stehen, gültig gemacht und diejenigen aller nachfolgenden Befehle verworfen.
- ☐ c) Mit Hilfe einer Multimediaeinheit kann eine Rechenoperation parallel auf mehreren Registern ausgeführt werden.
- ☐ d) Wie lange Befehle in der Pipeline der Gleitkommaeinheit eines superskalaren Prozessors verbleiben, hängt nicht von der Art der Berechnung ab.
- ☐ e) Die Beendigung eines Befehls (completion) ist dadurch definiert, dass die Ausführung abgeschlossen ist und das Resultat unabhängig von der Programmordnung datenabhängigen Operanden zur Verfügung gestellt wird.
- ☐ f) Rückordnung eines Befehls bedeutet das Entfernen des Befehls aus dem Rückordnungspuffer mit oder ohne das Gültigmachen des Resultats.
- ☐ g) Ein Commitment eines Befehls wird durchgeführt, wenn die Resultate aller Befehle, die vor dem betreffenden Befehl stehen, gültig sind oder im gleichen Takt gültig gemacht werden oder wenn eine Unterbrechung vor oder während der Ausführung auftrat.

Aufgabe 5 (10 Punkte)

- ☐ a) Während die Parallelverarbeitung bei superskalaren Prozessoren im wesentlichen durch die Hardware organisiert wird, geschieht dies bei der VLIW-/EPIC-Technik im wesentlichen durch die Software.
- ☐ b) Das Trace-Scheduling zur Organisation der spekulativen Ausführung von Befehlen nach bedingten Sprüngen wird bei VLIW- und EPIC-Maschinen eingespart, da die Anordnung der einzelnen Operationen und insbesondere der Speicherzugriffe bereits durch den Compiler erledigt wird.
- ☐ c) Bei VLIW- sowie EPIC-Maschinen kann in der Speicherhierarchie nur ein einstufiger Cache verwendet werden, da es dem Compiler möglich sein muss, die Speicherzugriffe zu organisieren.
- ☐ d) Die EPIC-Technik versucht mit einem dreifach VLIW erweiterten Dreibefehlsformat die Vorteile der Einfachheit und hohen Taktrate des VLIW-Prozessors mit den Vorteilen des dynamischen Scheduling zu verbinden.
- ☐ e) Durch die Umordnung von Befehlen innerhalb eines Befehlspaketes können bei VLIW-Prozessoren Datenabhängigkeiten entfernt werden. Befehlspaketübergreifend ist dies nicht möglich.
- ☐ f) Der EPIC-Ansatz versucht die Einfachheit und hohen Taktraten von VLIW-Prozessoren mit den Vorteilen des dynamischen Scheduling und der Veränderung der Ausführungsreihenfolge zu verbinden.
- ☐ g) Im Gegensatz zu VLIW-Prozessoren benötigen EPIC-Prozessoren keine Leerbefehle zum Auffüllen von Befehlsbündeln.
- ☐ h) In einem EPIC-Befehlsbündel dürfen Befehle stehen, die voneinander abhängig sind.
- ☐ i) Ein sechsfacher EPIC-Prozessor, wie beispielsweise der Itanium-Prozessor, kann bis zu 18 Befehle gleichzeitig ausführen.
- ☐ j) Ein EPIC-Befehl besteht aus einem Opcode, einem Prädikatfeld, drei Adressfeldern, weiteren Spezialfeldern und Template-Bits.

Aufgabe 6 (8 Punkte)

- ☐ a) In der IF-Stufe eines Superskalarprozessors wird in jedem Takt ein der Zuordnungsbreite entsprechender Befehlsblock vom Befehlszählerregister adressiert und aus dem Code-Cache-Speicher geholt.
- ☐ b) Zur Vermeidung von Datenkonflikten besteht der Primär-Cache-Speicher aus einem separaten Code-Cache-Speicher und einem Daten-Cache-Speicher.
- ☐ c) Bei einem gemeinsamen Code- und Daten-Cache-Speicher können in der Sekundärstufe Inkonsistenzen zwischen Sekundär-Cache und Hauptspeicher auftreten.
- ☐ d) Um die Effizienz der Befehlsholung zu verbessern, können selbstausrichtende Code-Cache-Speicher die verminderte Zuweisung von Befehlen nach einer Steuerflussänderung ausgleichen, indem sie die an der Zieladresse stehenden aufeinanderfolgenden Befehls- bzw. Cacheblöcke zusammenfassend in einem Takt lesen.
- ☐ e) Stehen Steuerflussbefehle jeweils am Anfang eines Befehlsblockes, dann können die nachfolgenden Pipeline-Stufen problemlos ohne Effizienzverlust durch die Befehlsholestufe mit Befehlen versorgt werden.
- ☐ f) Um die Effizienz der Befehlsholung zu verbessern, kann man einen Cache-Block gegenüber einem Befehls-Block vergrößern und damit einen nicht auf den Beginn eines Cache-Blocks ausgerichteten Zugriff ermöglichen.
- ☐ g) Der Trace-Cache-Speicher bietet die Möglichkeit aus verschiedenen Cache-Blöcken extrahierte und zusammengesetzte Befehlsfolgen in einem Takt zu lesen.
- ☐ h) Soll das Beschreiben und Aktualisieren eines Trace-Cache-Speichers parallel zur Programmausführung durchgeführt werden, dann ist dies bei der Ausführung der Pipeline zu berücksichtigen.

Aufgabe 7 (7 Punkte)

- ☐ a) Die Methode der Ausführung der alternativen Programmsequenzen (Prädikation) nach einem bedingten Sprung ist bei statistisch unabhängigen Sprungausgängen gegenüber der Sprungvorhersage wesentlich effektiver.
- ☐ b) Bei einem 2-Bit-Prädiktor mit Hysteresemethode kann der Vorhersagezustand aus der sicheren Vorhersage eines genommenen Sprungs direkt in die unsichere Vorhersage des nicht genommenen Sprungs wechseln.
- ☐ c) Bei statistisch unabhängigen Sprungausgängen werden bei einer Sprungvorhersage mit zwei Bits gegenüber einer Vorhersage mit nur einem Bit Schleifen-Sprünge besser vorhergesagt.
- ☐ d) Sprünge, bei denen mit einer Wahrscheinlichkeit von 95% verzweigt wird, werden durch einen 2-Bit-Prädiktor besser vorhergesagt als durch einen 1-Bit-Prädiktor.
- ☐ e) Alternierende Sprünge (t-nt-t-nt-...) werden durch einen 2-Bit-Prädiktor besser vorhergesagt als durch einen 1-Bit-Prädiktor.
- ☐ f) Ein Prädiktor, der wegen der begrenzten Adresslänge eine Sprungvorhersage für zwei oder mehr Verzweigungen vornimmt, wird sich an das durchschnittliche Verhalten aller beteiligten Verzweigungen anpassen.
- ☐ g) Ein gemeinsamer Prädiktor, der wegen der begrenzten Adresslänge eine Sprungvorhersage für zwei oder mehr Verzweigungen vornimmt, wird insbesondere bei langen Befehlsfolgen gegenüber getrennten Prädiktoren in der Regel eine geringere Vorhersagegüte aufweisen.

Aufgabe 8 (21 Punkte)

Sprungvorhersage Es sollen statische und dynamische Sprungvorhersageverfahren bei der Behandlung zweier verschachtelter Schleifen miteinander verglichen werden. Zu diesem Zweck wird das folgende Programm betrachtet, das zwei Matrizen A und B der Größe $m \times n$, mit $m \times n$, addiert und das Ergebnis in der Matrix C ablegt. Dabei seien verwendete Register folgendermaßen vorbelegt:

R0	0
R1	1
R2	m
R3	n
R4	Adresse von Matrix A
R5	Adresse von Matrix B
R6	Adresse von Matrix C

1	MADD:	ADD	R11,R0,R0	R11:=0
2	LOOP1:	ADD	R11,R1,R11	R11 inkrementieren
3		ADD	R12,R0,R0	R12:=0
4	LOOP2:	ADD	R12,R1,R12	R12 inkrementieren
5		SUB	R7,R11,R1	R7:=R11-R1
6		MULT	R7,R3,R7	R7:=R3*R7
7		ADD	R7,R12,R7	R7:=R12+R7
8		SUB	R7,R7,R1	R7:=R7-R1
9		ADD	R8,R4,R7	R8:=R4+R7
10		LB	R9,0(R8)	R9 erhält Inhalt der Adresse aus R8
11		ADD	R8,R5,R7	R8:=R5+R7
12		LB	R10,0(R8)	R10 erhält Inhalt der Adresse aus R8
13		ADD	R9,R10,R9	R9:=R10+R9
14		ADD	R8,R6,R7	R8:=R6+R7
15		SB	0(R8),R9	Adresse aus R8 erhält Inhalt von R9
16		SLT	R15,R12,R3	Ist $R12 < R3$?
17		BNEZ	R15,LOOP2	Wenn ja, GOTO LOOP2
18		SLT	R15,R11,R2	Ist $R11 < R2$?
19		BNEZ	R15,LOOP1	Wenn ja, GOTO LOOP1

Das Programm besteht aus zwei Schleifen, die äußere wird m mal durchlaufen, die innere n mal. Für die Lösung der folgenden Aufgaben ist die Berechnung innerhalb der Schleifen nicht von Bedeutung. (Bei der Eingabe der algebraischen Ausdrücke in die folgenden Felder können Multiplikationszeichen wie üblich weggelassen oder durch einen $*$ dargestellt werden, überflüssige Klammern müssen jedoch zwingend weggelassen werden)

a) Wie viele Verzweigungsbefehle werden insgesamt durchlaufen? (2 P.)

Hinweis: Als abhängige Variablen dürfen in den folgenden Feldern nur m und n auftreten.

Bei wie vielen durchlaufenen Verzweigungsbefehlen wird ein Sprung genommen? (1 P.)

b) Es sollen statische Sprungvorhersagetechniken miteinander verglichen werden.

Wie viele richtige und wie viele falsche Vorhersagen gibt es bei folgenden drei statischen Verfahren?

Geben Sie die Anzahlen in die folgenden Felder ein. Als abhängige Variablen dürfen dabei nur m und n auftreten.

b1) predict always not taken (2 P.)

	Zahl der Vorhersagen
richtig	<input type="text"/>
falsch	<input type="text"/>

b2) predict always taken (2 P.)

	Zahl der Vorhersagen
richtig	<input type="text"/>
falsch	<input type="text"/>

b3) predict backward taken/forward not taken (2 P.)

	Zahl der Vorhersagen
richtig	<input type="text"/>
falsch	<input type="text"/>

c) Nun sollen die Vorhersagen von dynamischen Verfahren verglichen werden. Wie viele richtige und wie viele falsche Vorhersagen gibt es bei folgenden drei dynamischen Verfahren? Geben Sie die Anzahl richtiger Vorhersagen der inneren und äußeren Schleife sowie die Gesamtzahl in die folgenden Felder ein. Als abhängige Variablen dürfen dabei nur m und n auftreten.

c1) Ein-Bit-Prädiktor (Initialzustand: predict taken) (4 P.)

Vorhersagen	Innere Schleife	Äußere Schleife	Insgesamt
richtig	<input type="text"/>	<input type="text"/>	<input type="text"/>
falsch	<input type="text"/>	<input type="text"/>	<input type="text"/>

c2) 2-Bit-Prädiktor mit Sättigungszähler (Initialzustand: predict weakly taken) (4 P.)

Vorhersagen	Innere Schleife	Äußere Schleife	Insgesamt
richtig	<input type="text"/>	<input type="text"/>	<input type="text"/>
falsch	<input type="text"/>	<input type="text"/>	<input type="text"/>

c3) 2-Bit-Prädiktor mit Hysteresezähler (Initialzustand: predict weakly taken) (4 P.)

Vorhersagen	Innere Schleife	Äußere Schleife	Insgesamt
richtig			
falsch			

Aufgabe 9 (18 Punkte)

Ein-Bit-Prädiktor bei Adressüberschneidung

Gegeben sei ein Programm, bei dem zwei Verzweigungsbefehle $B1$ und $B2$ alternierend ausgeführt werden. Dabei werde angenommen, dass die Verzweigungsadressen von $B1$ und $B2$ in den niederwertigen Bits übereinstimmen, so dass diese in der Sprungverlaufstabelle auf die gleiche Speicherzelle abgebildet werden. Die Verzweigungsbefehle $B1$ und $B2$ verwenden für die Vorhersage den gleichen (gemeinsamen) Prädiktor.

a) Bestimmen Sie die Vorhersagegüte, wenn das Programm folgenden Verlauf aufweist und der Ein-Bit-Prädiktor mit NT initialisiert wird (t=branch taken; nt=branch not taken) (9P):

	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$
Sprungverlauf	t	nt	nt	t	t	nt	nt	t

Tragen Sie für jeden Durchlauf die Vorhersage des Prädiktors ("T" für einen genommenen und "NT" für einen nicht genommenen Sprung) in die folgende Tabelle ein und geben Sie an, ob die Vorhersage korrekt (R) oder falsch (F) ist. (8P)

Durchlauf	1		2		3		4	
	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$
Sprungverlauf	t	nt	nt	t	t	nt	nt	t
Vorhersage 1-Bit-Prädiktor								
Vorhersagekorrektheit F/R								

Welchen Wert hat damit die Vorhersagegüte? (1P)

b) Welche Vorhersagegüte ergäbe sich, wenn die Verzweigungsadressen nicht in den niederwertigen Bits übereinstimmen würden? (5P)

In diesem Fall gibt es zwei voneinander unabhängige Ein-Bit-Prädiktoren, wobei die beiden Prädiktoren jeweils mit NT initialisiert werden.

Durchlauf	1		2		3		4	
	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$
Sprungverlauf	t	nt	nt	t	t	nt	nt	t
Vorhersage 1-Bit-Prädiktor								
Vorhersagekorrektheit F/R								

Welchen Wert hat damit die Vorhersagegüte? (1P)

c) Konstruieren Sie ein Sprungverlaufsmuster, das mit einem gemeinsamen Prädiktor stets falsche Vorhersagen liefert! (4P)

Durchlauf	1		2		3		4	
	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$	$B1$	$B2$
Sprungverlauf								
Vorhersage 1-Bit-Prädiktor								
Vorhersagekorrektheit F/R								

Aufgabe 10 (5 Punkte)

- ☐ a) Die Vorhersagegenauigkeit von n -Bit-Prädiktoren ist durch die Berücksichtigung der Ausgänge anderer benachbarter im Programmablauf vorhergegangener Sprünge nicht zu verbessern, da diese den aktuellen Sprungausgang nicht beeinflussen.
- ☐ b) Während n -Bit-Prädiktoren für eine Vorhersage immer nur den Verlauf des zugeordneten Sprungs berücksichtigen, berücksichtigen Korrelationsprädiktoren für eine Vorhersage auch die Verläufe benachbarter im Programmablauf vorhergegangener Sprünge.
- ☐ c) Ein (3,3)-Korrelationsprädiktor berücksichtigt für die aktuelle Vorhersage die Ausgänge der letzten drei Sprünge, unabhängig davon durch welche Befehle diese drei Sprünge ausgeführt wurden.
- ☐ d) Die Auswirkungen einer falschen Sprungvorhersage können alternativ auch durch Mehrpfadausführung oder Kontextwechsel in mehrfädigen Prozessoren gemindert werden.
- ☐ e) Die Anwendung der Prädiktion, d.h. die parallele Ausführung mehrerer Programmpfade nach einer Verzweigung ist dann besonders effizient wenn Sprünge mit großer Zuverlässigkeit richtig vorhergesagt werden können.

Aufgabe 11 (20 Punkte)

Korrelationsprädiktor

Ein (m,n)-Korrelationsprädiktor nutzt das Verhalten der letzten m Sprünge für die Vorhersage des aktuellen Sprungverhaltens. Für diese Vorhersage trifft er anhand des BHR, das die Vergangenheit der letzten m Sprünge speichert, eine Auswahl aus 2^m n -Bit-Prädiktoren, welcher dann den aktuellen Vorhersagewert liefert. Im Folgenden soll das Verhalten eines (2,2)-Korrelationsprädiktors mit Sättigungszähler zur Sprungvorhersage an folgendem in Maschinensprache dargestellten IF-THEN-Konstrukt untersucht werden.

```

      :
      :
      BNEZ  R1, LOOP1   Sprung nach LOOP1 falls R1 ≠ 0
      ADDI  R1, #1      R1=0+1=1
LOOP1: SUBI  R3, R1, #1  R3=R1-1
      BNEZ  R3, LOOP2   Sprung nach LOOP2 falls R3 ≠ 0
      :
      :
LOOP2: :
      :
      :

```

Es sei angenommen, dass der Wert der Variablen d , die im Register R1 abgelegt ist, beim Eintritt in diese Programmsequenz zwischen 0 und 2 alterniert. Somit ergibt sich eine Sprungfolge nt-t-nt-t-nt-t-nt-t... für den Sprungbefehl S1 als auch für den Sprungbefehl S2. Der zweite Sprung wird genommen wenn der erste genommen wurde und wird nicht genommen wenn der erste auch nicht genommen wurde.

Die folgende Tabelle zeigt einen für die zwei Sprünge S1 und S2 relevanten Ausschnitt aus der initialen Sprungverlaufstabelle PHT des (2,2)-Korrelationsprädiktors, in der alle Elemente auf den Initialzustand "predict weakly taken" (WT) gesetzt sind. Das zwei Bit breite Sprungverlaufsregister (BHR) ist auf den Initialzustand

1	1
---	---

 gesetzt.

Anfangszustand von BHR und PHT

		00	01	10	11
BHR	11	S1	WT	WT	WT
		S2	WT	WT	WT

a) Geben Sie die Verläufe von BHR und PHT für insgesamt 10 Sprünge an, wobei initial $d=0$ gelte.

Dabei kann sich jeder der vier 2-Bit-Prädiktoren in den Zuständen SNT,WNT,WT,ST bewegen. Geben Sie in die folgenden Felder jeweils den Inhalt des BHR und der einzelnen Elemente der PHT ein.

BHR und PHT nach dem 1. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 2. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 3. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 4. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 5. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 6. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 7. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 8. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 9. Sprung

		00	01	10	11
BHR		S1			
		S2			

BHR und PHT nach dem 10. Sprung

		00	01	10	11
BHR		S1			
		S2			

b) Bestimmen Sie die stationäre Wahrscheinlichkeit für eine korrekte Sprungvorhersage. (2P)

Aufgabe 12 (7 Punkte)

Entscheiden Sie, welche der folgenden Aussagen zur Registerumbenennung (Register Renaming) korrekt sind.

- ☐ a) Die Registerumbenennung kann auf statische oder auf dynamische Weise erfolgen, wobei beide Methoden als Mikroarchitekturtechnik implementiert werden können.
- ☐ b) Aufgrund der Registerumbenennung bei einem RISC-Prozessor stehen dem Programmierer effektiv eine größere Anzahl von Registern zur Verfügung.
- ☐ c) Mit Hilfe der Registerumbenennung können alle Abhängigkeiten unter den Registeroperanden aufgelöst werden.
- ☐ d) Datenabhängigkeiten zwischen Registeroperanden können nach der dynamischen Umbenennung ohne Betrachtung der Befehlsanordnung im Programm einfach durch Vergleich der Registerbezeichner ermittelt werden.
- ☐ e) Durch das Register Renaming wird bei einem RISC-Prozessor die Sicherung von Registerwerten für noch nicht beendete Maschinenbefehle erreicht.
- ☐ f) Durch das Register Renaming wird insbesondere auch eine Erhöhung der Taktrate erreicht.
- ☐ g) Das Befehlsfenster entkoppelt den Befehlsbereitstellungs- und Dekodiertteil vom Ausführungsteil des Prozessors, dabei sind alle Befehle innerhalb des Befehlsfensters frei von Gegen-Ausgabe- und Steuerflussabhängigkeiten.

Aufgabe 13 (9 Punkte)

Entscheiden Sie, welche der folgenden Aussagen zur Befehlszuordnung und den Ausführungsstufen einer Pipeline korrekt sind.

- ☐ a) Bei der Zuordnung der Befehle zu den Ausführungseinheiten unterscheidet man statische und dynamische Zuordnung, wobei Skalarprozessoren in der Regel eine statische Zuordnung durch die Hardware, VLIW-Prozessoren eine dynamische Zuordnung durch die Software realisieren.
- ☐ b) Die Befehlszuordnung bei Superskalarprozessoren kann sowohl mit einem statischen als auch mit einem dynamischen Scheduling, bei dem die Hardware flexibel über die Zuordnungsreihenfolge der vom Compiler festgelegten Programmordnung entscheidet, ausgeführt sein.
- ☐ c) Für einen effizienten Ablauf müssen in jedem Takt die ausführbereiten Befehle im Befehlsfenster durch Prüfen der Verfügbarkeit der Operandenwerte und der Ressourcen ermittelt und zugeordnet werden.
- ☐ d) In den spezialisierten Ausführungseinheiten innerhalb der Pipeline-Ausführungsstufen werden die durch die Befehle spezifizierten Operationen in einem oder mehreren Takten ausgeführt und die Ergebnisse in temporären Registern zwischengespeichert.
- ☐ e) Lade-/Speichereinheiten organisieren die Schreib-/Leseoperationen auf den Code-Cache-Speicher innerhalb des Prozessors in zwei oder drei Takten.
- ☐ f) Da Speicherbefehle häufig auf den zu speichernden Datenwert, der von vorangegangenen arithmetischen Operationen erst geliefert wird und in der Rückordnungsstufe als gültig markiert werden muss, warten, werden nachfolgende Ladebefehle grundsätzlich den Speicherbefehlen während der Wartezeit vorgezogen.
- ☐ g) Unter SIMD-Parallelität versteht man die Ausführung verschiedener Operationen auf Teilen von Registersätzen durch Multimediaeinheiten.
- ☐ h) Das Gültigmachen von Resultatwerten erfolgt durch das Kopieren aus den Umbenennungspufferregistern in die entsprechenden Architekturregister. Falls keine von den separaten Umbenennungspufferregistern existieren, wird einfach die Abbildung des entsprechenden Architekturregisters auf das physikalische Register gültig gemacht.
- ☐ i) Eine präzise Unterbrechung kann nur sicher gestellt werden, wenn alle Befehle, die in der Programmordnung vor dem Unterbrechungsereignis stehen, vollständig ausgeführt sowie die Resultate gültig gemacht wurden, und wenn die Resultate aller Befehle, die in der Programmordnung nach dem Unterbrechungsereignis stehen, verworfen wurden.

Aufgabe 14 (23 Punkte)

Scheduling und Renaming

Gegeben sei ein Maschinenprogramm und ein superskalärer Prozessor mit zwei Integer-ALUs (IU) und zwei Load/Store-Einheiten (LS). Bis auf die Multiplikation benötigen alle Operationen nur einen Taktzyklus. Die Multiplikation erfordere zwei Taktzyklen.

```

1  LOAD  R1,[A]
2  LOAD  R2,[B]
3  MUL   R3,R1,R2
4  LOAD  R1,[C]
5  LOAD  R2,[D]
6  MUL   R4,R1,R2
7  ADD   R1,R3,R4
8  STORE [E],R1
9  SUB   R2,R3,R4
10 STORE [F],R2

```

a) Wie viele Taktzyklen würden zur Ausführung des Maschinenprogramms auf einem skalaren Prozessor benötigt, der über nur je eine der oben genannten Ausführungseinheiten verfügt und der diese nur nacheinander benutzen kann?

Berücksichtigen Sie nur die Zeiten, die von den Ausführungseinheiten benötigt werden! (2P)

Antwort: Bei rein sequentieller Ausführung werden Taktzyklen benötigt.

b) Die Befehle sollen nun den Ausführungseinheiten so zugeordnet werden, dass die Zeit zur Ausführung minimiert wird. Dabei darf zwar die vorgegebene Befehlsreihenfolge, nicht aber die Registerzuordnung verändert werden. (9P)

Geben Sie jeweils die Befehle, die in dem entsprechend Takt ausgeführt werden, syntaktisch exakt in die entsprechenden Felder ein.

Takt	IU1	IU2	LS1	LS2
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

c) Die Ausführungszeit des Maschinenprogramms kann durch die Umbenennung einiger Register (register renaming) weiter verringert werden. Dazu soll eine statische Registerumbenennung, wie sie durch einen Compiler erfolgt, durchgeführt werden. (9P)

Geben Sie jeweils die Befehle, die im jeweiligen Takt ausgeführt werden, syntaktisch exakt in die entsprechenden Felder ein.

Die Umbenennung von Registern soll durch eine Verdopplung des Index erfolgen, aus R1 wird R11, usw.

Takt	IU1	IU2	LS1	LS2
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

d) Wie groß ist die unter c) erreichte Beschleunigung (Speedup) in Bezug auf die sequentielle Lösung? (2P)

Wie groß ist der theoretische Optimalwert für den Speedup? (1P)