Postanschrift: FernUniversität, D-5808 Name, Vorname	4 Hage	Matr Adre sons mõg	hier unb ikelnumn sse eintr st keine E ikch.	ner und ragen,	ing		FERNUNIVERSITÄT in Hagen EINGANG
Straße, Nr. PLZ, Wohnort							FERNUNIVERSITÄT in Hagen D-58084 Hagen
						Fa	achbereich Informa
Kurs: 01709 " Klausur am 31.07.20					Höre □ Vo □ Te	rstatus: ollzeitstudent eilzeitstudent veithörer	Klausurort: ☐ Berlin ☐ Bochum ☐ Frankfurt
Zutreffen unbeding		reuze	en 1		□ Ga □ Ba □ Le	veitnorer asthörer achelor ehramt	☐ Hamburg ☐ Karlsruhe ☐ Düsseldorf ☐ München ☐ Bregenz ☐ Wien
- Auf oa be	1	2	3	4	5	Summe	
					Programa		
erreichbare Punktzahl	4	11	40	25	20	100	
bearbeitet							
erreichte Punktzahl							
Note: Hagen, den			_		Re	etreuer:	

©2004 FemUniversität in Hagen

Aufgabe 1 (Grundlagen) 4 Punkte

Bitte geben Sie durch "Ja" oder "Nein" an, ob die folgenden Behauptungen zutreffen.

- a) Vektorrechner gehören zur Klasse der MIMD-Rechner.
- b) Bei einer weichen Echtzeitanforderung kann die Zeitschranke gelegentlich verpasst werden.
- c) Fehlertoleranz bedeutet, dass das Rechnersystem eine defekte Komponente selbst repariert.
- d) Neben der Verarbeitungsgeschwindigkeit gibt es weitere Ziele beim Entwurf eines Rechners.

Aufgabe 2 (Spekulative Ausführung) 11 Punkte

Gegeben sei folgendes Pseudo-C Programm:

```
#include <stdio.h>
#include <string.h>
int i=0;
void main() {
   printf("Kleiner Test\n");
   do {
      printf("%x\n",i);
      i++;
   while (I<10);
   printf("Wert von i ist nun %d\n");
}</pre>
```

Als Sprungvorhersage sei ein bimodaler Prädiktor in der Variante eines Hysteresezählers implementiert, eine Speicherung der Subroutinenaufrufadressen geschieht über einen Return-Address-Stack Nach dem Rücksetzen ist der Return Address-Stack leer, der Prädiktor mit "strongly not taken" initialisiert.

Aufgabe 2 (Spekulative Ausführung, Forts.)

3P

a) Welche Probleme treten in Bezug auf Rücksprungadressen in einem Prozessor mit nicht-spekulativer Ausführung auf?

8P

b) Welche zusätzlichen Probleme in Bezug auf Rücksprungadressen treten dann bei spekulativer Ausführung auf?

Aufgabe 3 (Pipelining) 40 Punkte

ാ	ת
1	$\boldsymbol{\nu}$

a) Bitte geben Sie die im Kurs genannten Pipeline-Konfliktarten an.

6P

b) Man betrachte zwei aufeinander folgende Befehle *Inst*1 und *Inst*2, wobei *Inst*1 vor *Inst*2 ausgeführt wird. Zwischen diesen Befehlen können verschiedene Arten von Datenabhängigkeiten bestehen. Nennen und erläutern Sie diese.

3P

c) Welche Datenkonflikte werden welchen Datenabhängigkeiten zugeordnet?

Datenkonflikte	Datenabhängigkeit

Aufgabe 3 (Pipelining Forts.)

2P

d) Welche Datenkonflikte können beim DLX nicht auftreten wenn Fliesskommaeinheiten nicht benutzt werden?

4P

e) Wann benötigt man Forwarding, was ist Forwarding?

Aufgabe 3 (Pipelining Forts.)

22P

f) Gegeben sei folgendes DLX-Programm:

, Discli	oo-Vonflikt	Toothroaramm	
; Pipelii	ie-kontiikt	Testprogramm	
main:			
S1:	add	r1,r1,8	
S2:	beqz	rl,Finish	
S3:	add	r1, r1, 3	
S4:	multf	f2,f1,f4	
S5:	multf	f1,f8,f4	
S6:	multf	f2,f1,f2	
S7:	multf	f4, f11, f4	
S8:	multf	f5,f12,f4	
S9:	addf	F1,F2,F3	
S10:	movf	F1,F4	
Finish:			
S11:	trap	0	

Geben Sie die Datenabhängigkeiten in der unteren Tabelle an. Kennzeichnen Sie unterschiedliche Arten dabei durch T,A und O entsprechend der Bezeichnung δ^t , δ^a , δ^o im Kurstext. S11, bzw. Trap #0 soll dabei nicht in die Untersuchung mit eingehen.

	s1	S2	S3	S4	S 5	S6	S7	S8	89	S10
s1	-									
S2		-								
S 3			-							
S4				_						
S5					-					
S6						-				
S7							-			
S8			-					-		
S9										
S10										-

Aufgabe 4 (Cache) 25 Punkte

Gegeben seien drei Cache-Speicher DM, A4 und AV, die jeweils acht Cache-Blöcke besitzen, wobei jeder Cache-Block acht Bytes umfasst. Der Cache DM ist als direkt-abgebildeter Cache (direct-mapped) organisiert, Cache A4 als 4-fach assoziativer Cache (4-way-setassociative), Cache AV ist vollassoziativ (fully-associative). Bei den Cache-Speichern A4 und AV soll die least recently used-Ersetzungsstrategie (LRU) angewendet werden. Nehmen Sie an, die Cache-Speicher seien zu Beginn leer, und es soll eine Serie von einzelnen Bytes (byteadressierbar) mit den folgenden 32-Bit-Adressen gelesen werden. Bei den folgenden Zahlen seien die führenden Stellen mit null belegt:

8C 4B 45 B7 53 4A E6 12 42 55

5P

a) Geben Sie unter Berücksichtigung der jeweiligen Cache-Organisationsform die Anzahlen der Sätze und Blöcke pro Satz an. Wie viele Adressbits werden für den Tag bzw. den Index-Teil benötigt? Wie breit ist die Wortadresse?

Cache	Sätze	Blöcke/Satz	#Tag	#Index	#Wort
direkt abgebildet					
4-fach assoziativ					
vollassoziativ					

10P

b) Stellen Sie für jede Cache-Organisationsform dar, welcher Speicherzugriff einen Cache-Hit zur Folge hat, indem Sie für jeden Cache-Hit ein "H" in das entsprechende Feld schreiben.

Cache	8C	4B	45	B7	53	4A	E6	12	42	55
direkt abgebildet										
4-fach assoziativ										
vollassoziativ										

Aufgabe 4 (Cache – Forts.)

10P

c) Stellen Sie den Zustand der drei Caches nach dem letzten Speicherzugriff dar, d.h. für jeden Cache-Block den Cache-Index und den Cache-Tag.

DM		A	A4			.v
Index	Tag	Index	Tag		Index	Tag

Aufgabe 5 (Multiprozessorsysteme) 20 Punkte

4P

a) Wie hoch ist der maximal mögliche Speedup nach Amdahls Gesetz, ausgedrückt als Funktion der Prozessorzahl n, wenn der nicht-parallelisierbare Anteil a=1/(n-1) beträgt.

6P

b) Gegeben sei ein nachrichtengekoppeltes Mehrprozessor-System mit 1 Master und n Slaves. Auf dem System soll ein Problem gelöst werden, das sich in k*n Teilaufgaben zerteilen lässt, wobei k eine Ganzzahl ≥ 2 ist. Jede Teilaufgabe lässt sich in Zeit tb von einem einzelnen Slave-Prozessor lösen. Vor dem Start einer Teilaufgabe und nach ihrer Beendigung ist jeweils eine Kommunikation der Dauer tc zwischen Master und Slave notwendig. Der Master-Prozessor soll keine eigene Arbeit verrichten. Bestimmen Sie den Speedup und die Effizienz als Funktion von n und r, wobei r=tc/tb ist. Für welche Werte von r ist der Speedup mindestens 1, für welche Werte von r ist die Effizienz mindestens 0,5?

10P

c) Gegeben Sei das folgende Programm auf einer priority-CRCW-PRAM mit n^2 Prozessoren. Was tut das Programm? Ändert sich das Programmverhalten, wenn man eine andere Variante (common, arbitrary) benutzt?

```
/* Die n^2 Prozessoren haben die Nummern (u,v), wobei 0<=u,v<=n-1 */
/* Es gibt ein Array int feld[n]. Dieses ist im gemeinsamen Speicher
angelegt. Es soll mit n unterschiedlichen Ganzzahlen bereits belegt
sein. */
/* Es gibt weiterhin ein Array int pt[n] und eine Variable int mindex
im gemeinsamen Speicher. Alle array Elemente und die Variable sind
mit 1 initialisiert. */
if(feld[u]<feld[v]) pt[u]=0;
barrier; /* Hier müssen alle Prozessoren warten, bis auch der letzte
die vorangehende Anweisung bearbeitet hat. */
if(feld[u]==1) mindex=u;</pre>
```

- ENDE -