DEPARTMENT INFORMATIK
Lehrstuhl für Informatik 3 (Rechnerarchitektur)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3, 91058 Erlangen

27.09.2012

Klausur zu

Grundlagen der Rechnerarchitektur und -organisation

.....	.....	.....	.....
Matrikelnummer	Geb.-Datum	Vorname	Name

- Es sind *keine* Hilfsmittel erlaubt!
- Legen Sie den Ausweis (mit Lichtbild!) griffbereit auf den Platz!
- Dieses Aufgabenheft umfasst 16 Seiten. Überprüfen Sie die Vollständigkeit!
- Gesondert beigelegte Blätter werden nicht bewertet.
- Schreiben Sie deutlich! Unleserliches wird nicht bewertet!
- Es darf nicht mit der Farbe rot geschrieben werden!
- Offensichtlich falsche oder überflüssige Antworten können zu Punktabzug führen!
- Begründen Sie Ihre Antworten!

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen
- die Kenntnisnahme der obigen Informationen.

Erlangen, den 27.09.2012 .....
(Unterschrift)

Ich bin damit einverstanden, dass mein Prüfungsergebnis der Klausur unter Angabe der Matrikel-Nummer veröffentlicht wird.

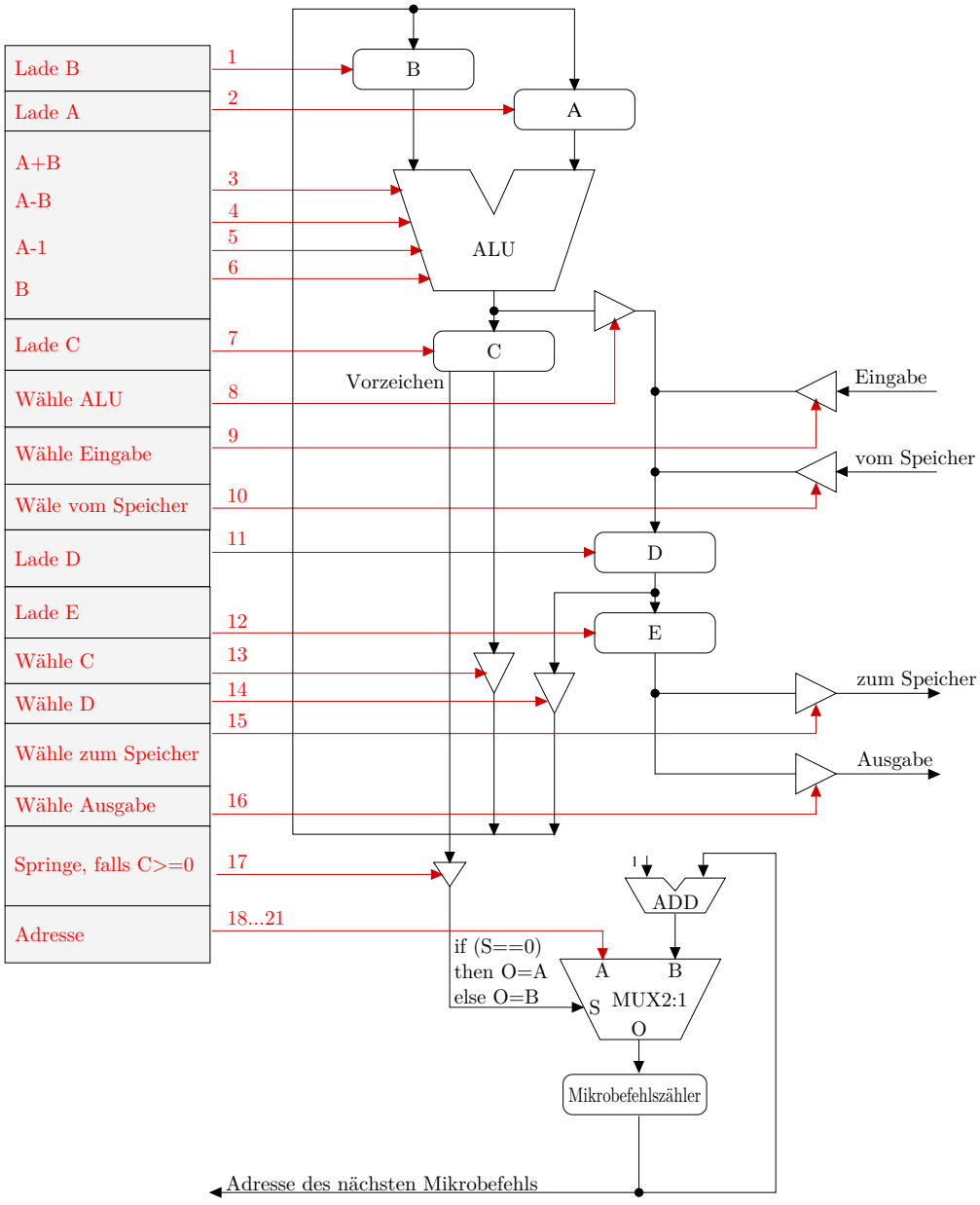
Erlangen, den 27.09.2012 .....
(Unterschrift)

Aufgabe	1	2	3	4	5	6	7
max. Punktzahl	10	17	10	14	13	12	14
erreichte Punktzahl							

Summe	/90		
Bonus	/15		
Gesamt	/90	Note	

Aufgabe 1: Mikroprogrammierung — 10 Punkte

Gegeben sei der folgende Teil einer CPU:



Hinweis:
Am Ende der Klausur finden Sie eine Kopie der Angabe, die Sie zur Bearbeitung heraustrennen dürfen.

Das folgende Mikroprogramm liest einen Parameter von der Eingabe und gibt das Doppelte des Wertes auf der Ausgabe zurück. (Das Kommentarfeld wird nicht bewertet!)

Adr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Kommentar
0	0	0	0	0	0	0	0	0	**1**	0	**1**	0	0	0	0	0	0	0	0	0	0	
1	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	
2	0	0	**1**	0	0	0	0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	

1. Wieviel Platz im Mikroprogrammspeicher wird für das Programm benötigt? 1 Punkt

2. Wie lässt sich der Platz minimieren, ohne die Anzahl der Befehle zu ändern? Geben Sie ein Beispiel! 2 Punkte

3. Wie nennt sich diese komprimierte Art der Mikroprogrammierung? 1 Punkt

4. Schreiben Sie ein Mikroprogramm für obige CPU, das zeichenweise eine Zeichenkette von der Eingabe liest und eine Kopie in die Ausgabe schreibt. Das Programm soll beendet werden, sobald das Zeichen „0” gelesen wurde.

   Verwenden Sie die folgende Tabelle. Leer gelassene Steuerleitungen entsprechen dem Wert „0”.

   Geben Sie bei Sprüngen die Zieladresse explizit an! Das Sprungziel ist in der Tabelle von links nach rechts kodiert. Das niederwertigste Bit des Sprungziels entspricht also Steuerleitung 21.

   (Die Tabellenlänge entspricht nicht der erwarteten Mikroprogrammlänge!) 6 Punkte

Adr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Kommentar
0																						
1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						
14																						
15																						

Aufgabe 2: Speicherverwaltung — 17 Punkte

1. Eine CPU biete eine Memory-Management-Unit mit folgenden Eigenschaften:

   - zweistufige Adresstabellen
   - je 1024 Einträge zu je 4 Byte in den Tabellen
     - Bit 31-12: höherwertige Bits der physikalischen Adresse
     - Bit 11-3: unbenutzt
     - Bit 2: Cache-Disabled-Bit
     - Bit 1: Write-Enable-Bit
     - Bit 0: Present-Bit
   - Pages zu je 4 KiB Größe

   Folgendes soll für ein Programm erfüllt sein:

   - im virtuellen Adressbereich `0x00 00 70 00` bis `0x00 00 7F FF` soll schreibgeschützter Programmcode stehen,
   - ab der virtuellen Adresse `0x00 40 00 00` soll der Video Memory der VGA-Karte eingeblendet sein,
   - im virtuellen Adressbereich `0x00 AC 00 00` bis `0x00 AC FF FF` sollen die Programmdaten gespeichert sein,
   - alle anderen Bereiche sollen Zugriffsfehler auslösen.

   Physikalisch befinde sich der Video-Memory an den Adressen `0x00 0B 80 00` bis `0x00 0B FF FF`, freier Speicher sei ab Adresse `0x00 00 10 00` ausreichend vorhanden.

   Beschreiben Sie eine mögliche Page-Tabelle, die obige Bedingungen erfüllt! Achten Sie auf die korrekte Vergabe der Bits für die Flags! — 12 Punkte

Fortsetzung Aufgabe 2

2. Was passiert, wenn die CPU auf die virtuelle Adresse `0x00 00 78 88` schreibend zugreift? Begründen Sie Ihre Antwort! 1 Punkt

3. Was ist der maßgebliche Unterschied zwischen Segmentierung und Paging? 2 Punkte

4. Wie könnte bei Segmentierung ein gemeinsamer Arbeitsspeicherbereich zwischen mehreren Prozessen realisiert werden? Ist dies auch bei Paging möglich? 2 Punkte

Aufgabe 3: Umformung — 10 Punkte

Gegeben sei folgendes Hochsprachenprogramm:

```
1 do {
2   c = getchar();
3   if (skip_magic == 1) continue;
4   do_magic(c);
5 } while ( (!(c == EOF) && !(c == 'q' && skip == 0)) && quit == 0 )
```

1. Warum kann obiges Programm nicht direkt auf einer herkömlichen CPU (z.B. Intel x86) ausgeführt werden? 2 Punkte

2. Formen Sie das Programm so in if-goto-Darstellung um, dass es sich möglichst leicht in Assembler übersetzen lässt.

   Verändern Sie dabei die eigentlichen Operationen nicht und führen Sie keine Optimierungen durch. 8 Punkte

Aufgabe 4: Arbeitsspeicher — 14 Punkte

1. Beschreiben Sie grob, wie ein DRAM-Chip organisiert ist. 1 Punkt

2. Ein 512 MiB großes Speichermodul soll aus 1 Byte breiten Speicherbausteinen der Größe 32 MiB aufgebaut werden. Das Modul sei 64 Bit breit. Es werden 32 Bit breite Adressen verwendet. Sequentielle Zugriffe sollen durch Interleaving performant durchführbar sein.

   Auf welches Byte in welcher Zeile welcher Hauptspeicherbank wird zugegriffen, wenn von der Adresse `0xCAFEBABE` gelesen werden soll? 4 Punkte

3. Was unterscheidet DDR(1)-SDRAM von einfachem SDRAM? 1 Punkt

4. Wodurch wird die Beschleunigung von DDR2– und DDR3-SDRAM gegenüber DDR-SDRAM erreicht? Was muss dazu an den Speicherzellen selbst geändert werden? 2 Punkte

5. Auf einem Speichermodul steht die Typenbezeichnung PC3-6400.

   Wie hoch ist die maximale Transferrate, der nötige Bustakt und der Speichertakt? Die Busbreite beträgt 64 Bit. 6 Punkte

Aufgabe 5: Parallelverarbeitung — 13 Punkte

Gegeben sei folgendes Programm in Pseudocode:

```
1  Loop: load R1, a[x+2]    ;Lädt Arrayelement in Register
2        load R2, a[x+1]
3        load R3, a[x-2]
4        load R4, a[x-1]
5        add R1, R1, R2     ;R1+R2->R1
6        add R3, R3, R4
7        add R1, R1, R3
8        mul R1, R1, 0.25
9        store R1, b[x]     ;Speichert Registerinhalt im Array
10       inc x              ;Inkrementiert x
11       bl x, 256, Loop    ;Springt zu Loop falls x < 256
```

1. Worin unterscheiden sich Pipelining, Superskalarität, Thread-Parallelität und echte Parallelität im Hinblick auf den Instruktionsstrom? 4 Punkte

2. Warum wird der reale Speedup auf einer CPU mit Fließbandverarbeitung bei obigem Programm niedriger sein als der maximale? Verwenden Sie die entsprechenden Fachbegriffe! 2 Punkte

3. Kann eine superskalare CPU die Ausführung des Programms beschleunigen? 2 Punkte

4. Welche Varianten von Parallelität auf Thread-Ebene wurden in der Vorlesung vorgestellt? Erläutern Sie kurz die jeweilige Vorgehensweise. 3 Punkte

5. Erläutern Sie kurz, wie obiges Programm verändert werden müsste, um auf einer Multicore-CPU performant ausgeführt werden zu können! 2 Punkte

Aufgabe 6: Stack — 12 Punkte

Gegeben sei das folgende Assemblerprogramm:

```
1  start:
2          push 4
3          call fib
4          add esp, 4
5          jmp end
6  fib:
7          cmp [esp+4], 2
8          jg .L2
9          mov eax, 1
10         jmp .L3
11 .L2:
12         push ebx
13         mov eax, [esp+8]
14         sub eax, 1
15         push eax
16         call fib
17         add esp, 4
18
19         mov ebx, eax
20         mov eax,[esp+8]
21         sub eax, 2
22         push eax
23         call fib
24         add esp, 4
25         add eax, ebx
26         pop ebx
27 .L3:
28         ret
29 end:
```

Hinweis:
Am Ende der Klausur finden Sie eine Kopie der Angabe, die Sie zur Bearbeitung heraustrennen dürfen.

1. Für welche Klasse von Befehlssatzarchitekturen wurde das Programm wahrscheinlich geschrieben? Begründen Sie Ihre Entscheidung anhand des Programms! 2 Punkte

2. Geben Sie die Signatur der Funktion `fib` in C-(ähnlicher) Notation an. 2 Punkte

3. Wieviel Speicherplatz braucht das Programm zur Laufzeit?

   Hinweis:
   Überlegen Sie sich dazu, wieviele Bytes bis zum ersten Aufruf der Funktion `fib` nötig sind, wie tief der rekursive Abstieg maximal ist und wieviel Daten jeweils auf dem Stack abgelegt werden müssen.

   Bedenken Sie, dass Speicher auf dem Stack auch wieder freigegeben wird. 6 Punkte

4. Warum werden auf RISC-Architekturen oft Register zur Parameterübergabe genutzt, auf CISC-Architekturen hingegen der Stack? 1 Punkt

5. Wann braucht ein Programm besonders viel Platz auf dem Stack? 1 Punkt

Aufgabe 7: Cache — 14 Punkte

1. Ein Cache umfasse 64 KiB Nutzdaten. Die Blockgröße sei 64 Byte. Die Adressbreite betrage 32 Bit.

   Vergleichen Sie die Organisationsformen *Vollassoziativ, 8-fach Assoziativ* und *Direkt-Abbildend* hinsichtlich des Speicherplatzes für Verwaltungsinformationen und der Menge der Adressvergleicher, die jeweils für die Realisierung des Caches insgesamt nötig sind! Ignorieren Sie dabei die Ersetzungsstrategie!

   Hinweis:
   Denken Sie daran, dass ein gültiger Eintrag von einem ungültigen unterscheidbar sein muss.

   7 Punkte

2. Eine 32-Bit CPU verwende einen 2-fach assoziativen Cache mit insgesamt 4 Blöcken. Jeder Block umfasst 8 Bytes. Der Cache sei zu Beginn leer. Die Ersetzung erfolgt nach dem LRU-Prinzip.

   Ein Programm liest nacheinander jeweils 4 Byte von den folgenden Adressen:

   `0x00 CD 10 50`, `0x01 00 00 00`, `0x01 00 00 08`, `0x00 CD 10 54`.

   Der relevante Speicherbereich hat folgenden Inhalt:

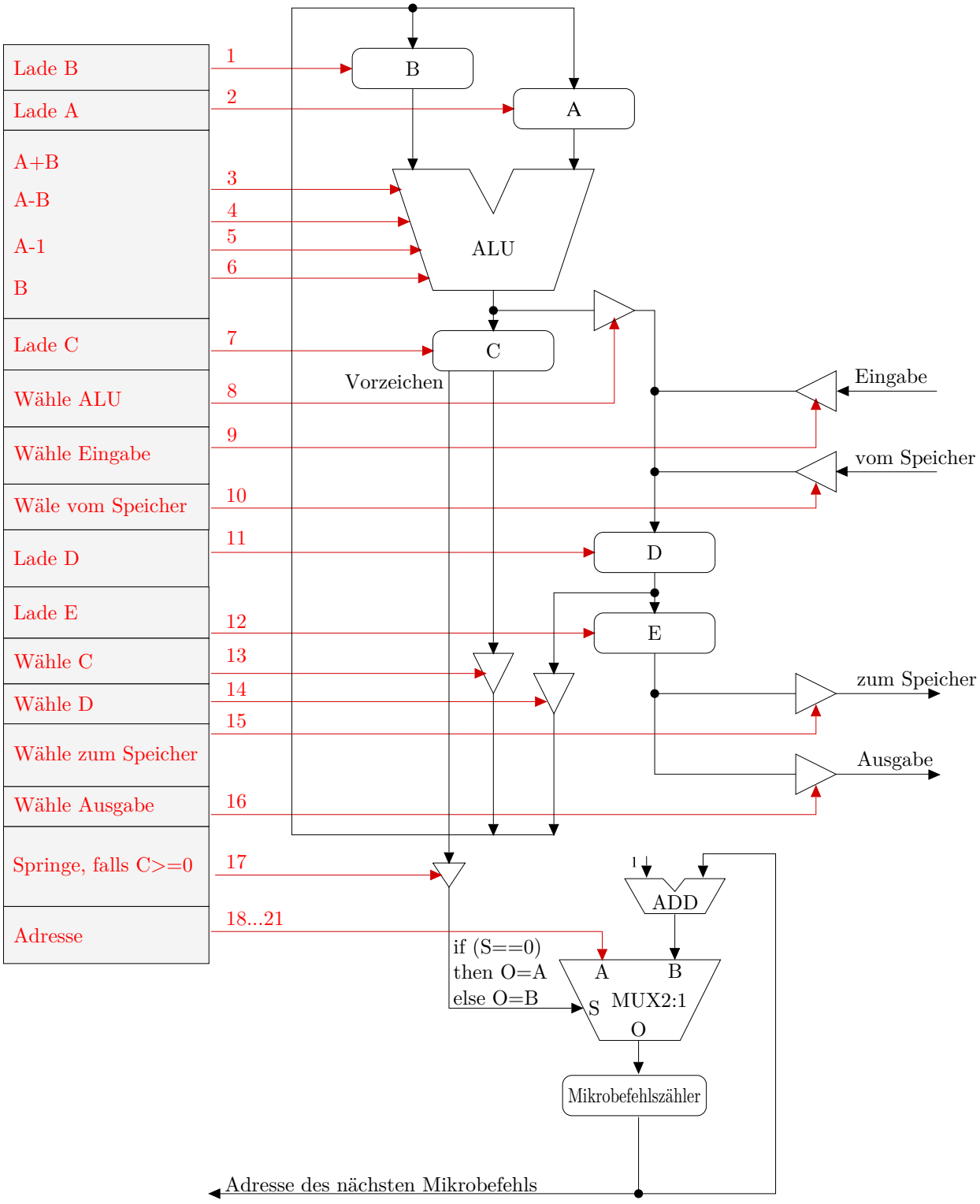
```
0x00 CD 10 50: CA 01 6B 7F FF 00 10 BC  BF 41 45 E8 00 A7 56 00
0x00 CD 10 60: E8 00 A4 22 FF FF 50 BF  40 B0 C7 00 A3 05 21 21
0x00 CD 10 70: 57 41 56 41 55 41 54 41  89 55 53 FD 89 48 48 F3
...
0x01 00 00 00: A8 64 FF FF 89 48 E8 DF  A9 1C FF FF 8B 48 ED 3D
0x01 00 00 10: 32 4B E1 75 FB C1 11 09  7B 80 2E 01 94 0F FF 10
...
```

   Welchen Inhalt hat der Cache nach Ausführung des Programms?

5 Punkte

Menge	Tag	Valid	Daten
0			
0			
1			
1			

3. Was passiert beim Zugriff auf die Adresse `0x00 CD 10 74`? 2 Punkte


Lade B
Lade A
A+B
A-B
A-1
B
Lade C
Wähle ALU
Wähle Eingabe
Wäle vom Speicher
Lade D
Lade E
Wähle C
Wähle D
Wähle zum Speicher
Wähle Ausgabe
Springe, falls C>=0
Adresse
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18...21
B
A
ALU
C
Vorzeichen
Eingabe
vom Speicher
D
E
zum Speicher
Ausgabe
1
ADD
if (S==0)
then O=A
else O=B
A
B
S
MUX2:1
O
Mikrobefehlszähler
Adresse des nächsten Mikrobefehls

```
1  start:
2          push 4
3          call fib
4          add esp, 4
5          jmp end
6  fib:
7          cmp [esp+4], 2
8          jg .L2
9          mov eax, 1
10         jmp .L3
11 .L2:
12         push ebx
13         mov eax, [esp+8]
14         sub eax, 1
15         push eax
16         call fib
17         add esp, 4
18
19         mov ebx, eax
20         mov eax,[esp+8]
21         sub eax, 2
22         push eax
23         call fib
24         add esp, 4
25         add eax, ebx
26         pop ebx
27 .L3:
28         ret
29 end:
```