DEPARTMENT INFORMATIK
Lehrstuhl für Informatik 3 (Rechnerarchitektur)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3, 91058 Erlangen
30.09.2009

Klausur zu

Grundlagen der Rechnerarchitektur und -organisation

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

- Es sind *keine* Hilfsmittel erlaubt (Ausnahme: 1 DIN-A4 Blatt mit Notizen)!
- Legen Sie den Ausweis (mit Lichtbild!) griffbereit auf den Platz!
- Dieses Aufgabenheft umfasst 22 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!

Durch meine Unterschrift bestätige ich

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

Erlangen, den 30.09.2009 .....
(Unterschrift)

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

Erlangen, den 30.09.2009 .....
(Unterschrift)

Aufgabe	1	2	3	4	5	6	7	8	9	Summe
max. Punktzahl	4	9	12	19	5	10	10	9	12	90
erreichte Punktzahl										

Aufgabe 1: Der Universalrechenautomat (4 Punkte)

- Nennen Sie die Phasen des Maschinenbefehlszyklus des klassischen von-Neumann-Rechners und erklären Sie, was in jeder Phase jeweils passiert. (4 Punkte)

Aufgabe 2: Mikroprogrammierung (9 Punkte)

Für das Leitwerk eines von Neumannschen Universalrechenautomaten gibt es unterschiedliche Entwurfsprinzipien.

1. Erläutern Sie kurz, welche Vorteile ein mikroprogrammiertes Leitwerk im Vergleich zu einem festverdrahteten Leitwerk hat.
   (3 Punkte)

2. Warum wird in einigen Hochleistungs-Prozessoren dennoch ein festverdrahtetes Leitwerk eingesetzt?
   (1 Punkt)

Aufgabe 2: Mikroprogrammierung (Fortsetzung)

3. Varianten der Mikroprogrammierung (5 Punkte)

   - Was versteht man unter „Horizontaler Mikroprogrammierung“?

   - Was versteht man unter „Vertikaler Mikroprogrammierung“?

   - Welche Vor- und Nachteile haben diese Varianten jeweils?

Aufgabe 2: Mikroprogrammierung (Fortsetzung)

- Wie bezeichnet man eine Variante der Mikroprogrammierung, bei der versucht wird, die jeweiligen Vorteile von „Horizontaler Mikroprogrammierung“ und „Vertikaler Mikroprogrammierung“ zu vereinen?
  Wie geht man bei dieser Variante vor?

Aufgabe 3: Befehlsatzarchitekturen (12 Punkte)

1. Um Speicheradressen in Befehlen zu spezifizieren, gibt es eine Vielzahl von unterschiedlichen Adressierungsarten.
   Nennen Sie vier verschiedene Adressierungsarten und erläutern Sie kurz, wie diese funktionieren.
   (2 Punkte)

2. Was kennzeichnet einen Prozessor, der nach dem RISC-Architektur-Prinzip aufgebaut ist?
   (3 Punkte)

Aufgabe 3: Befehlsatzarchitekturen (Fortsetzung)

3. Was versteht man unter „Pipeline-Verarbeitung“ in Prozessoren?
   Welche Probleme können dabei grundsätzlich auftreten?
   (4 Punkte)

Aufgabe 3: Befehlsatzarchitekturen (Fortsetzung)

4. Was ist das Kennzeichen einer „superskalaren“ Prozessorarchitektur?
   (1 Punkt)

5. Was kennzeichnet eine „Multi-Core“-Architektur?
   Warum sind aktuelle Prozessoren meist „Multi-Core“-Architekturen?
   (2 Punkte)

Aufgabe 4: Speicherhierarchie (19 Punkte)

Um dem Ideal eines sehr großen, sehr schnellen und dennoch preiswerten Speichers möglichst nahe zu kommen, verwendet man in aktuellen Rechnern eine mehrstufige Speicherhierarchie.

1. Was versteht man in diesem Zusammenhang unter „zeitlicher Lokalität“ und was unter „räumlicher Lokalität“?
   (2 Punkte)

2. Was ist ein Cache und welche Funktion hat er?
   (2 Punkte)

Aufgabe 4: Speicherhierarchie (Fortsetzung)

3. Was versteht man bei einem Cache unter dem „Platzierungsproblem“ und was unter dem „Identifikationsproblem“?
   (2 Punkte)

4. Nennen und erklären Sie die drei in der Vorlesung vorgestellten Organisationsformen für Caches!
   (6 Punkte)

Aufgabe 4: Speicherhierarchie (Fortsetzung)

5. Wie heißt die Aktualisierungsstrategie, die bei einem Primär-Cache bevorzugt eingesetzt wird und wie heißt die Aktualisierungsstrategie, die bei einem Sekundär-Cache bevorzugt eingesetzt wird?
   Wie geht man dabei jeweils vor?
   (1 Punkt)

6. Welche prinzipielle Funktion erfüllt ein „Translation Lookaside Buffer“ bei der virtuellen Speicherverwaltung?
   Wo befindet sich der „Translation Lookaside Buffer“?
   (2 Punkte)

Aufgabe 4: Speicherhierarchie (Fortsetzung)

7. Beim „Paging“-Verfahren ist die Größe einer Seite ein wichtiger Punkt. Erörtern Sie die jeweiligen Vor- bzw. Nachteile einer kleinen bzw. großen Seitengröße!
   (4 Punkte)

Aufgabe 5: Busse (5 Punkte)

1. Erläutern Sie die Unterschiede zwischen dem PCI-Bus und dem PCI-Express-Bus. Welche der beiden Varianten kommt der im Universalrechenautomaten enthaltenen Lösung am nähesten?
   (2 Punkte)

2. Warum ist PCI-Express heute das Mittel der Wahl?
   (3 Punkte)

Aufgabe 6: Arithmetik (10 Punkte)

Schreiben Sie eine schnelle(!) C-, C++-, Pascal- oder Java-Funktion, die zwei (unsigned short) Integer-Zahlen multipliziert, jedoch keinen Multiplikationsoperator verwendet (Shift-Operationen bzw. Multiplikationen mit 2 sind erlaubt)

Aufgabe 7: Unterprogramme (10 Punkte)

Vereinfachen Sie das folgende Code-Stück! Benutzen Sie soweit wie möglich nur `if-then-goto`-Anweisungen mit einfachen Bedingungen sowie `goto`-Anweisungen!

```
if ((! (a < 0)) || ((a < b) && (b != 0)))
    anw1();
else
    anw2();
```

Aufgabe 8: Caching (9 Punkte)

Gegeben sei folgendes Assemblerprogramm:

```
    movl $0, %eax
    movl $0, %ebx
    movl $1000, %ecx
loop:
    addl (%ecx), %ebx
    addl $4, %ecx
    addl $1, %eax
    cmpl $16, %eax
    jne loop
```

Ein Befehl soll 2 Nanosekunden zur Bearbeitung benötigen. Ein Hauptspeicherzugriff dauert 60 Nanosekunden. Jeder Befehl soll mit einem Speicherzugriff geladen werden können.

1. Kommentieren Sie die einzelnen Assembler-Befehle! (1 Punkt)

2. Auf welche Daten-Bytes im Speicher greift das Programm zur Laufzeit zu? (2 Punkte)

Aufgabe 8: Caching (Fortsetzung)

3. Wie lange benötigt das Programm zur Abarbeitung, wenn keine Caches verwendet werden? (3 Punkte)

4. Der Prozessor soll nun über einen ausreichend großen Primär-Cache verfügen, um sämtliche Befehle und Daten des Programms zwischenzuspeichern. Ein Zugriff auf diesen Cache benötigt 4 Nanosekunden. Ursprünglich sei der Cache leer. Wie lange benötigt das Programm nun? (3 Punkte)

Aufgabe 9: Protection (12 Punkte)

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 11-2: unbenutzt
  - Bit 1: Write-enable-Bit
  - Bit 0: Present-Bit
- Pages zu je 4 kByte Größe

Folgendes soll für ein Programm erfüllt sein:

- im virtuellen Adressbereich `0x00000000` bis `0x00001fff` soll auf einen ROM-Baustein zugegriffen werden (Code),
- im virtuellen Adressbereich `0x00010000` bis `0x00010fff` soll auf Hauptspeicher zugegriffen werden (Daten),
- im virtuellen Adressbereich `0xfffff000` bis `0xffffffff` soll auf Hauptspeicher zugegriffen werden (Stack),
- alle anderen Bereiche sollen Zugriffsfehler auslösen.

Physikalisch sind der ROM-Baustein an Adresse `0xffffe000` und der Hauptspeicher von Adresse `0x00000000` bis `0x80000000` am Bus.

Beschreiben Sie eine mögliche Page-Tabelle, die obige Bedingungen erfüllt!

Aufgabe 9: Protection (Fortsetzung)

Zusätzlicher Platz

Zusätzlicher Platz

Zusätzlicher Platz