



TI II:
Rechnerarchitektur
SS 2006
Klausur 13.10.06



Prof. Dr.-Ing. Jochen Schiller, AG Technische Informatik, Freie Universität Berlin

Bearbeitungszeit 90 Minuten – Ohne Hilfsmittel!

Bitte schreiben Sie auf alle Blätter Ihren Namen und Ihre Matrikelnummer!
Maximale Punktzahl: 90, zum Bestehen sind 45 Punkte nötig (50% der max. Punkte).

Vorname: _____

Name: _____

Matrikelnummer: _____

1. Aufgabe: Einfaches Assembler-Programm (10 Punkte)

Schreiben Sie ein einfaches Assemblerprogramm, das die Elemente eines Arrays im Hauptspeicher aufsummiert und das Ergebnis am Ende in Register \$1 schreibt. Die Elemente des Arrays sind 32-bit-Integer-Worte und die Adresse des ersten Arrayelements steht im Register \$4. Das Ende des Arrays wird durch ein Element mit dem Wert 0 markiert.

2. Aufgabe: Virtueller Speicher (12 Punkte)

Ein Computer habe einen virtuellen Speicher mit einer Seitengröße von 1024 Wörtern (ein Wort ist die kleinste adressierbare Einheit, die Größe spielt hier keine Rolle). Es seien weiterhin 8 virtuelle Seiten und 4 physikalische Seitenrahmen vorhanden. Die Seitentabelle in der MMU sehe wie folgt aus (alle Werte zur Basis 10, Punktabzug für falsche Antworten):

Virtuelle Seite	Seitenrahmen
0	3
1	1
2	Nicht im Arbeitsspeicher
3	Nicht im Arbeitsspeicher
4	2
5	Nicht im Arbeitsspeicher
6	0
7	Nicht im Arbeitsspeicher

Welche virtuellen Adressen führen zu einem Seitenfehler?

- A. 1000
- B. 3000
- C. 5000
- D. 8000

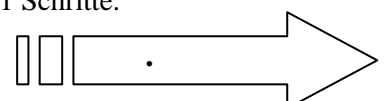
Welche Abbildungen zwischen virtueller und physikalischer Adresse stimmen?

- E. Virtuelle Adresse 1024 auf physikalische Adresse 1024.
- F. Virtuelle Adresse 2048 auf physikalische Adresse 2048.
- G. Virtuelle Adresse 1023 auf physikalische Adresse 4095.
- H. Virtuelle Adresse 0 führt zu einem Fehler.

3. Aufgabe: Prozessorleistung (6 Punkte)

Welche der folgenden Aussagen treffen zu? (Punktabzug für falsche Antworten)

- A. Ein gemeinsamer Cache für Daten und Befehle auf dem Prozessor erhöht die Leistung im Vergleich zur Harvard-Architektur.
- B. Spekulatives Laden kann ein Blockieren der Pipeline verhindern.
- C. Eine k-stufige Pipeline benötigt zum Ausführen von n Befehlen im Idealfall nur n+k Schritte.
- D. Spekulatives Laden kann die Cache-Effizienz erhöhen, aber auch verringern.
- E. Lange Pipelines (viele Stufen) weisen ohne Hemmnisse eine höhere Leistungsfähigkeit als kurze Pipelines auf.
- F. Pipeline-Architekturen kommen mit einem prozessorinternen Bus aus.
- G. Kurze Pipelines verkraften Unterbrechungen besser als lange Pipelines.
- H. Eine k-stufige Pipeline benötigt zum Ausführen von n Befehlen im Idealfall nur n+k+1 Schritte.



4. Aufgabe: Grundlegende Rechnerarchitekturen (10 Punkte)

- a) Erläutern Sie den wesentlichen Unterschied zwischen von-Neumann- und Harvard-Architekturen. (5 Punkte)
- b) Was ist der von-Neumann-Flaschenhals? (2 Punkte)
- c) Warum ergibt sich bei der Harvard-Architektur eine Leistungssteigerung? (3 Punkte)

5. Aufgabe: Cachelstrategien (19 Punkte)

Beantworten Sie jede der folgenden Fragen in Bezug auf Cache-Speicher unter den Gesichtspunkten „vollassoziativ“, „satzassoziativ“ und „direkt abgebildet“.

- a) Wo kann ein Block platziert werden? (3 Punkte)
- b) Wie findet man einen Block? (3 Punkte)
- c) Welcher Block wird evtl. ersetzt? (3 Punkte)

Beantworten Sie jede der folgenden Fragen in Bezug auf Cache-Speicher unter den Gesichtspunkten „Durchschreibeverfahren“ und „Rückschreibeverfahren“.

- d) Wie werden Schreiboperationen verarbeitet? (2 Punkte)
- e) Welche Konsequenz hat das Verfahren im Multiprozessorfall? (4 Punkte)

Vergleichen sie „write invalidate“ und „write update“ hinsichtlich Belastung des Verbindungsnetzes und der Verzögerung beim Datenzugriff in Multiprozessorsystemen. (4 Punkte)

6. Aufgabe: Pipelines (4 Punkte)

Prozessoren, wie z.B. ein Pentium IV setzen auf relativ lange Pipelines, ein Itanium 2 dagegen auf recht kurze. Welche Vor- bzw. Nachteile hat die jeweilige Strategie? Nennen Sie je zwei.

7. Aufgabe: Unterbrechungen (8 Punkte)

- a) Was sind Unterbrechungen und warum werden diese benötigt? (1 Punkt)
- b) Wann können diese zum Beispiel auftreten? Sind Unterbrechungen häufig auf einem normalen Rechner? (2 Punkte)
- c) Was muss vor und nach der Unterbrechungsbehandlung gemacht werden? (2 Punkte)
- d) Woher weiß ein Prozessor, welche Unterbrechungsbehandlungsroutine ausgeführt werden soll? (1 Punkt)
- e) Kann eine Unterbrechungsbehandlung wiederum unterbrochen werden? Warum? (2 Punkte)

8. Fehlerhaftes Programm (10 Punkte)

Das folgende Programm kopiert Wörter von der Adresse in Register \$a0 in die Adresse in Register \$a1 und zählt dabei die übertragenen Wörter in Register \$v0. Das Programm beendet den Kopiervorgang, wenn es ein Wort mit dem Wert 0 erkennt. Der Inhalt der Register \$v1, \$a0 und \$a1 muss nicht erhalten bleiben. Das abschließende Wort muss kopiert, darf aber nicht gezählt werden.

```
        addi $v0, $zero, 0          # initialisiere Zähler
loop: lw  $v1, 0($a0)              # lese nächstes Wort ein
        sw  $v1, 0($a1)            # schreibe Wort an Zieladresse
        addi $a0, $a0, 4           # setze Zeiger auf nächstes Wort
        addi $a1, $a1, 4           # erhöhe Zieladresse um ein Wort
        beq  $v1, $zero, loop      # verzweige, wenn nächstes Wort = 0
```

In diesem Programm befinden sich mehrere Fehler. Beheben Sie diese!

9. Aufgabe: Pipeline (11 Punkte)

Skizzieren Sie den grundlegenden Aufbau einer superskalaren Pipeline, benennen Sie die Einheiten und erläutern Sie kurz deren Funktion.