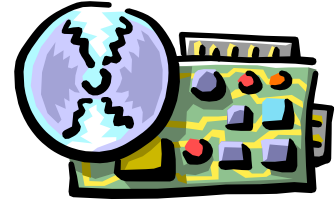




TI II:
Rechnerarchitektur
SS 2006
Klausur 31.07.06



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 Arrayelementes steht im Register \$4. Das Array hat 17 Elemente.

2. Aufgabe: Verbindungsstrukturen (6 Punkte)

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

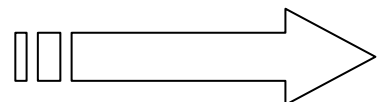
- a) Der Vorteil von PCI ist, dass angeschlossene Geräte jederzeit auf den Bus zugreifen dürfen.
- b) USB arbeitet nach dem Daisy-Chaining-Prinzip.
- c) Der Vorteil von USB ist, dass angeschlossene Geräte jederzeit auf den Bus zugreifen dürfen.
- d) Daisy-Chaining spart Leitungen.
- e) Die Signallaufzeit auf einem Bus hängt von der Taktfrequenz ab.
- f) Multiplexing spart Leitungen.
- g) Eine zentrale Bus-Zuteilung spart Leitungen.
- h) Die Länge eines Busses spielt eine wesentliche Rolle hinsichtlich der Übertragungsrate.

3. Aufgabe: Speicherverwaltung (6 Punkte)

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

- a) Das Betriebssystem ist im Allgemeinen alleine für die Umsetzung von virtuellen in physikalische Adressen zuständig.
- b) L2-Caches können nur bei Segmentierung eingesetzt werden.
- c) Seitengrößen sind programmabhängig.
- d) Aus Leistungsgründen muss die Umsetzung von virtuellen in physikalische Adressen durch spezielle Hardware unterstützt werden.
- e) Segmentierung kann mit Paging kombiniert werden.
- f) Interne Fragmentierung ist ein typisches Problem der Segmentierung.
- g) Seitenwechsel müssen bei der Programmierung nicht berücksichtigt werden.
- h) Seiten spiegeln die logische Programmstruktur wider.

Bitte wenden!



4. Aufgabe: Pipelining (15 Punkte)

Gehen Sie im Folgenden von einer einfachen 6-stufigen Pipeline aus: Befehl holen (IF), Befehl dekodieren (ID), Operanden holen (OF), Ausführung (EX), Befehlsbestätigung (CO), Rückspeichern (WB). Weiterhin liegt eine reine Load/Store-Architektur ohne architekturelle Beschleunigungsmaßnahmen (z.B. forwarding, reordering etc.) oder Hardware zur Erkennung von Hemmnissen vor. Operanden können erst dann aus Registern geholt werden, nachdem sie zurück gespeichert wurden.

- Wie lange (in Takten) dauert die vollständige Abarbeitung von $Z:=X*Y$; $A:=Z*B$; $D:=Z+B$? (5 Punkte)
- Wie groß ist im Idealfall die Beschleunigung der Programmabarbeitung mit dieser Pipeline? (2 Punkte)
- Welche Pipelinekonflikte können auftreten (nennen Sie 2 Konflikte mit Kurzbeschreibung)? (4 Punkte)
- Wie können die Pipelinekonflikte aufgelöst werden? (Nennung von 2 Möglichkeiten)? (4 Punkte)
- Nennen Sie die Hauptgemeinsamkeit und den Hauptunterschied zwischen Vektorpipelining in einem Vektorrechner und Befehlspipelining in einem Mikroprozessor. (2 Punkte)
- Ist jeder Prozessor mit einer Pipeline ein RISC-Prozessor? (1 Punkt)

5. Aufgabe: CPUs von heute (20 Punkte)

- Skizzieren Sie den Aufbau einer CPU und erläutern Sie kurz die wesentlichen Komponenten. (8 Punkte)
- Welche Voraussetzungen im Prozessor müssen gegeben sein, damit Pipelining sinnvoll genutzt werden kann? Geben Sie 2 Beispiele in Ihrer CPU-Skizze. (6 Punkte)
- Vergleichen Sie die Ansätze VLIW und superskalar hinsichtlich der Art der Parallelisierung. Wer steuert diese? Welchen Vorteil erhofft man sich von VLIW? (6 Punkte)

6. Aufgabe: Virtueller Speicher (10 Punkte)

Ein Computer habe einen virtuellen Adressraum mit 128 Seiten, aber lediglich 4 Seitenrahmen. Anfangs sei der Speicher leer. Ein Programm referenziere die virtuellen Seiten in folgender Reihenfolge:

13, 17, 23, 39, 17, 41, 75, 23, 39, 23, 41, 17

- Welche Referenzen verursachen einen Seitenfehler bei LRU als Ersetzungsstrategie? (5 Punkte)
- Welche Referenzen verursachen einen Seitenfehler bei FIFO als Ersetzungsstrategie? (5 Punkte)

7. Aufgabe: Multiprozessorsysteme (18 Punkte)

- Was versteht man unter Cache-Kohärenz? Warum strebt man nicht Konsistenz an? (4 Punkte)
- Welche Voraussetzung wird an die Verbindungsstruktur gestellt, damit MESI funktioniert? Wie verhält es sich mit der Skalierbarkeit des Ansatzes? (4 Punkte)
- Warum kauft man Multiprozessormaschinen für Millionen, wenn doch auch billigere Cluster oder Grids verfügbar sind? (4 Punkte)
- Angenommen ein Einprozessorsystem benötigt für 2000 Operationen 2000 Schritte (klassischer RISC), ein 8-Prozessor-System benötigt 2500 Operationen, aber nur 1100 Schritte auf Grund der möglichen Parallelisierung. Berechnen Sie die Effizienz E und die Beschleunigung S . (6 Punkte)

8. Aufgabe: Abhängigkeiten (5 Punkte)

Befehl	Bedeutung
ADD R_i, R_j, R_k	$R_k = R_i + R_j$
MUL R_i, R_j, R_k	$R_k = R_i * R_j$

- Warum lassen sich Datenabhängigkeiten zwischen Befehlen nicht durch Umsortieren auflösen? (2 Punkte)
- Beschreiben und benennen Sie alle Abhängigkeiten bei der folgenden Befehlsfolge, die auf einem superskalaren Prozessor ausgeführt werden soll (Angabe der Art der Abhängigkeit, Benennung der beteiligten Befehle und Register). (3 Punkte)

```
S1: ADD R2, 2, R1
S2: ADD R1, R3, R4
S3: MUL R5, 3, R3
S4: MUL R6, 3, R3
```