Freie Universität Berlin
Sommersemester 2003
Vorlesung Rechnerorganisation
Probeklausur

1. Aufgabe: Verbindungsstrukturen (4 Punkte)

Welche der folgenden Aussagen treffen zu?

- O Auf Grund der kurzen Entfernungen innerhalb eines PCs ist die Laufzeit der Signale irrelevant.
- O USB unterstützt keine garantierten Bandbreiten.
- Ọ Diverse Busse gibt es auch auf Grund der unterschiedlichen elektrischen Eigenschaften der Busse (z.B. Leitungslängen, Signalstärken).
- O Die Leistungsfähigkeit eines Systems ist i. Allg. proportional zu den Kosten. Überall die schnellsten Busse einzubauen wäre daher zu teuer.
- O Zur Überbrückung größerer Entfernungen kann nicht die gleiche Technik wie zur leistungsfähigen Überbrückung kurzer Entfernungen genutzt werden.
- O USB ist eine hochleistungsfähige Parallelschnittstelle.
- O Langsame Busse sind immer von der BVG
- O Zeitmultiplex wird oft zum Einsparen von Leitungen eingesetzt.
- O Langsame Busse benötigen keine Buszugriffssteuerung.

2. Aufgabe: DMA/PIO (3 Punkte)

Welche der folgenden Aussagen treffen bei einem Vergleich von Direct Memory Access und Programmed I/O zu?

- O DMA ist immer effizienter als PIO.
- O PIO wird stärker durch Unterbrechungen ausgebremst als DMA.
- O Bei DMA müssen Cache-Inkonsistenzen nicht extra berücksichtigt werden.
- O Sehr viel schneller als DMA und PIO ist das neue LMAA.
- O DMA hat den Nachteil, dass der Hauptprozessor keine Datenübertragung starten kann.
- O DMA hat den Vorteil, dass die dazu benötigte Hardware (der DMA-Controller) vielfach auf unterschiedlichen Peripheriegeräten vorkommen kann. Dadurch ergibt sich eine größere möglich Parallelität im Vergleich zu PIO, das ein Programm auf dem Hauptprozessor darstellt.
- O DMA entlastet den Hauptprozessor deutlich im Vergleich zu PIO.

3. Aufgabe: RISC/CISC (4 Punkte)

Welche Eigenschaften weist ein purer RISC-Prozessor typischerweise auf?

- O Alle Mikrobefehle haben die gleiche Länge und einen einheitlichen Aufbau.
- O Load/Store-Architektur kombiniert mit Harvard-Architektur für Caches.
- O Die richtige Anordnung der Befehle für eine effiziente Ausführung in der Pipeline bleibt dem Compiler überlassen.
- O Viele Adressierungsarten.
- O RISC heißt voll ausgeschrieben “Risc Is Smart Computing”
- O Alle Befehle können in einem Takt abgearbeitet werden.
- O relativ tiefe Pipeline (mehr als 5 Stufen).
- O Ein RISC Prozessor besitzt keine ISA Ebene, sondern nur eine Mikrocode Ebene

4. Aufgabe: Assemblerprogrammierung (20 Punkte)

Der Funktion procedure wir ein char-Zeiger *pointa übergeben.
Im Speicher stehen beginnend mit der Stelle auf die der Zeiger *pointa zeigt folgende 32Bit Werte : 7,1,3,4,6,2,9,0,8,5

Aufruf im C++ Code : int x = procedure(*pointa,5);

```
0:    int procedure (char* a, int b)
1:    {
2:          __asm
3:          {
4:                mov eax, a;
5:                mov ebx, b;
6:                dec ebx;
7:    start:      mov ecx, 0;
8:    tt:         mov edx, [eax + 4 * ecx];
9:                inc ecx;
10:               cmp edx, [eax + 4 * ecx];
11:               jge weiter;
12:               xchg edx, [eax + 4 * ecx];
13:               dec ecx;
14:               xchg edx, [eax + 4 * ecx];
15:               jmp start;
16:   weiter:     cmp ecx, ebx;
17:               jl tt;
18:               mov eax, [eax + 4 * ebx];
19:               mov ebx,0;
20:         }
21:   }
```

Welche Aussagen treffen zu ?

O Nach jedem Befehl mit indirekter Adressierung (wie in Zeile 8) muss eine Befehl ohne indirekte Adressierung folgen.
O Labels dienen nicht nur der besseren Lesbarkeit.
O Ein heutiger x86 Prozessor kann den Befehl „xchg eax , [ebx];“ in einem Takt abarbeiten.
O Der Wert des ersten Elements der Liste verändert sich nicht.
O Der Algorithmus berechnet das kleinste Element der Liste
O Der Algorithmus sortiert die Liste absteigend
O Der Algorithmus sortiert die Liste aufsteigend
O In der Variable x steht nach dem Aufruf der Wert 0
O Beim erfolgreichen Aufruf der Funktion ist der Rückgabewert immer Null.
O Die Zeilen 12 bis 14 des Algorithmus werden im schlechtesten Fall n^2 mal durchlaufen.
O Der Wert im Register ECX ist nie größer als der im Register EBX.
O Assembler wurde 1971 vom Harvard Professor Edward Assembler entwickelt.

5. Aufgabe: Betriebssystem, Assembler (6 Punkte)

Welche Aussagen zum Thema Betriebssysteme und Assembler treffen zu?

- O Speicherschutzmechanismen eines Betriebssystems müssen durch den Prozessor unterstützt werden.
- O Assembler-Befehle können direkt von einem Prozessor ausgeführt werden.
- O Zu den Aufgaben eines Betriebssystems gehört unter anderem die Betriebsmittelzuteilung und die Speicherverwaltung.
- O Der Betriebssystemkern ist der Teil eines Betriebssystems, der bei allen Betriebssystemen (z.B. Linux, Windows) gleich sein muss.
- O Windows ist dank DLLs immer schneller als Linux
- O Betriebssysteme sollen die semantische Lücke zwischen Anwendungen und Hardware verkleinern.
- O Programmierer müssen den Ort eines Programms im Speicher stets schon bei der Programmierung festlegen.
- O Betriebssystemfunktionen müssen regelmäßig aufgerufen werden, um zwischen Prozessen umzuschalten.
- O Auch Betriebssysteme können bei einem Speicherengpass (zu wenig RAM) vollständig auf die Festplatte ausgelagert werden.
- O DLLs müssen an die von den benutzenden Programmen referenzierten Speicherbereiche geschrieben werden

6. Aufgabe: Speicherverwaltung (6 Punkte)

Welche der folgenden Aussagen trifft zu?

- O Bei LRU als Ersetzungsverfahren ist die Wahrscheinlichkeit von Seitenflattern i. Allg. geringer als bei LIFO.
- O Interne Fragmentierung ist ein typisches Problem der Segmentierung.
- O Seitenwechselverfahren und Segmentierungsverfahren schließen sich gegenseitig aus.
- O Segmente spiegeln die logische Programmstruktur wieder.
- O L2-Caches können nur bei Segmentierung eingesetzt werden.
- O Bei billigen Computern wird die Festplatte als L3 Cache fuer den Hauptspeicher verwendet.
- O Segmentregister müssen bei Prozesswechseln berücksichtigt werden.
- O Segmentierung bleibt vor einem Assembler-Programmierer verborgen.
- O Aus Leistungsgründen muss die Umsetzung von virtuellen in physikalische Adressen durch spezielle Hardware unterstützt werden.
- O Paging wurde erfunden, um besser mit wachsenden Datenstrukturen, wie Stacks, umgehen zu können.

7. Aufgabe: Pipelining (8 Punkte)

Gehen Sie im Folgenden von einer einfachen 5-stufigen Pipeline aus: Befehl holen (IF), Befehl dekodieren (ID),

Operanden holen (OF), Ausführung (EX), 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. ADD X, Y, Z steht für Z:= X+Y. Welche Aussagen lassen sich dann treffen?

- O Die Abarbeitung von n Befehlen in einer k-stufigen Pipeline benötigt im Idealfall n+k-1 Taktzyklen.
- O Auf einer RISC Architektur muss der Compiler mögliche Pipeline-Konflikte auflösen.
- O Eine 12 Stufige Pipeline kann 96Bit Speichern.
- O Zusätzliche Systembusse begünstigen Parallelität
- O Im schlimmsten Fall benötigt die Ausführung von n Befehlen auf dieser Pipeline 5*n-1 Takte
- O Die Befehlsfolge ADD R2, R3, R1; ADD R1, R5, R4 ist nicht ausführbar.
- O Die vollständige Abarbeitung von Z:=X+Y; A:=Z+B dauert insgesamt 10 Takte.

8. Aufgabe: Unterbrechungen bei einem Standard-PC (6 Punkte)

Welche der folgenden Aussagen treffen hinsichtlich Unterbrechungen zu?

- O Wird gerade eine Unterbrechungsbehandlung durchgeführt, so können keine weiteren Unterbrechungen angenommen werden.
- O Maskierbare Unterbrechungen können im Gegensatz zu nicht maskierbaren ausgeblendet werden, um eine aktuelle Unterbrechungsbehandlung nicht zu unterbrechen.
- O Unterbrechungen können nur prozessorexterne Ursachen haben.
- O Unterbrechungen können nur prozessorinterne Ursachen haben.
- O Unterbrechungsbehandlungsroutinen werden nicht auf dem Prozessor ausgeführt sondern im Interrupt-Controller.
- O Unterbrechungsnachrichten vom Modem können die Telefonleitung blockieren.
- O Unterbrechungsbehandlungsroutinen müssen im Allgemeinen zunächst alle Register sichern, damit der Zustand (PSW) eines gerade laufenden Programms nicht verloren geht.
- O Unterbrechungen sind recht selten in einem Rechner und deuten auf Programmierfehler hin.
- O Unterbrechungsbehandlungsroutinen sind im Prinzip ganz normale Unterprogramme.
- O Nach Abarbeitung eines Interrupts muss das zuvor unterbrochene Programm von vorne gestartet werden.

9. Aufgabe: Cache (6 Punkte)

Welche der folgenden Aussagen treffen auf Caches zu?

- O Durch sinnvolle Programmierung kann ein Cache besser ausgenutzt werden.
- O Caches stehen außerhalb der Speicherhierarchie, da sie transparent für Programme sind.
- O Caches profitieren sowohl von räumlicher als auch zeitlicher Lokalität von Daten bzw. Befehlen.
- O Harvard-Architekturen zeichnen sich durch einen gemeinsamen Cache für Daten und Befehle aus.
- O Caches speichern nur physikalische, nie virtuelle Adressen.
- O Bei Durchschreibeverfahren müssen keine Zusatzvorkehrungen hinsichtlich der Datenkonsistenz getroffen werden.
- O Durchschreibeverfahren sind schneller als Rückschreibeverfahren.
- O Abgesehen von Leistungsvorteilen bringen Caches keine neue Funktion für Programme.
- O Wer weniger als 40 Punkte in diesem Test bekommt hat seine Klausurzulassung verspielt.