Prüfungsklausur 31231 Teil 20046 – SS 2013

Prof. Dr. J. Keller

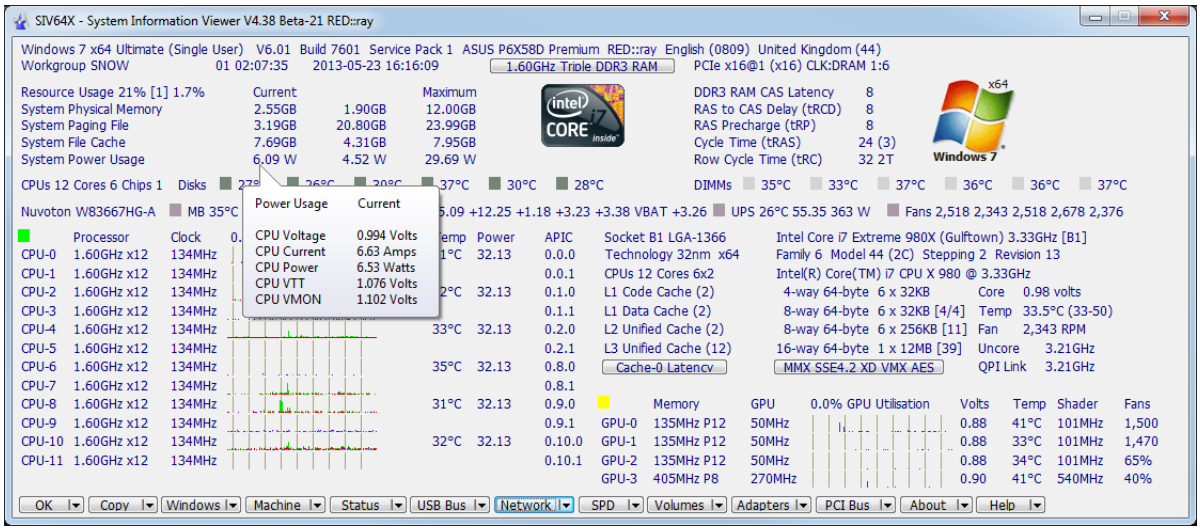
24.08.2013

Inhaltsverzeichnis

Bewertungsschema

Aufgabe	a	b	c	d	e	f	g	h	total
1	2	2	2	2					8
2									6
3									4
4									6
5	4	4							8
6	7	(4)							7 (plus 4 Zusatzpunkte)
7	2	4							6
8									5

1 Aufbau und Funktion eines Personal Computers



a) Welche Hauptplatine ist im PC installiert?

b) Unter welchem Betriebssystem wird der PC betrieben?

c) Welcher Prozessor welches Herstellers ist in dem PC eingesetzt?

d) Welche Caches mit welchen Größen und welchen Blockgrößen werden in dem PC eingesetzt?

Lösungsvorschläge

Zu a) ASUS P6X58D Premium RED

Zu b) Windows 7 x64 Ultimate Service Pack 1

Zu c) Intel Core i7 Extreme 980X (Gulftown) 3.33GHz [B1]

Zu d)
L1 Code-Cache: 32KB
L1 Daten-Cache: 32KB
L2 Code/Daten-Cache: 256KB
L3 Code/Daten-Cache: 12 MB

2 Speichermedien und Peripheriegeräte

Welche der folgenden Aussagen treffen zu?

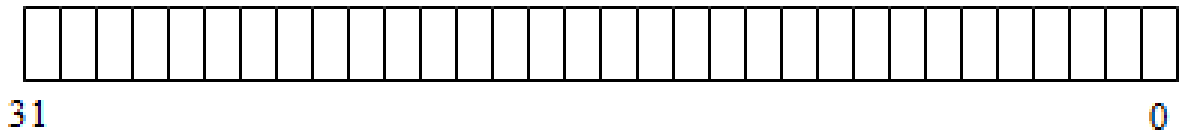
trifft zu:	trifft nicht zu:	
☐	☐	Bei einer CD-ROM bilden die Spuren konzentrische Kreise auf der Oberfläche.
☐	☐	In einer Festplatte wird der zu schreibende Datenstrom vor der Aufzeichnung in einen Speichercode umgeformt, der (beim Lesen) eine Rückgewinnung des Taktsignals erlaubt.
☐	☐	Im Linux-Dateisystem kann eine Datei, die nur 12 Blöcke groß ist, mit einem Inode verwaltet werden, so dass ein wahlfreier Zugriff auf die Blöcke möglich ist.
☐	☐	Ein Makecode ist ein Scancode, der beim Loslassen einer Taste der Tastatur erzeugt wird.
☐	☐	Wird an einen Flüssigkristall in einem LCD-Bildschirm eine Spannung angelegt, dann verhindert seine horizontale Ausrichtung, dass die Polarisationsebene des durchgeleiteten Lichts um 90 Grad gedreht wird.
☐	☐	Ein Trackball ersetzt in einer elektromechanischen Maus die Bildsensoren einer optischen Maus.

Lösungsvorschläge

Die zweite, dritte und fünfte Aussage treffen zu.

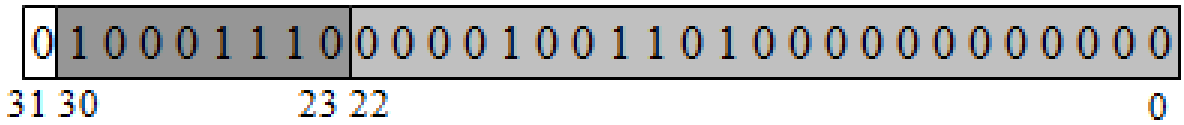
3 Datenformate

Stellen Sie die Dezimalzahl $Z_1 = 34000$ im 32-bit-Format des IEEE-754-Standards in binärer Form dar. Kennzeichnen Sie die unterscheidbaren Bitfelder (Vorzeichen, biased Exponent, Mantisse).



Lösungsvorschläge

$Z_1 = 34000 = 2^{15}+2^{10}+2^7+2^6+2^4 = 34000 = (-1)^0 \cdot 2^{142-127} \cdot (1.0000100110100\cdots 0)_2$
Daraus ergibt sich:

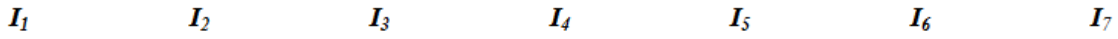


4 Pipeline-Konflikte

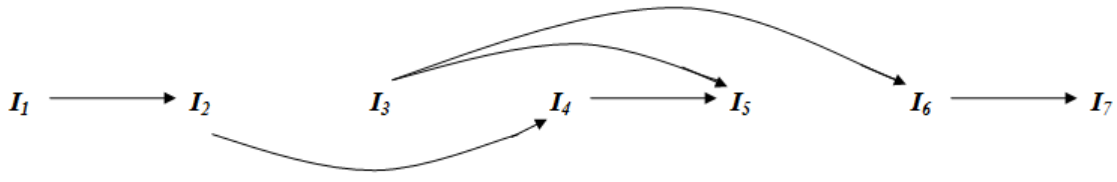
Wir betrachten die folgende Adress-Befehlsfolge:

	Befehl	Bedeutung
I_1	ADD R1,R2,R3	R1=R2+R3
I_2	ADDI R4,R1,#1	R4=R1+1
I_3	MUL R3,R2,#2	R3=R2*2
I_4	ADD R7,R4,R8	R7=R4+R8
I_5	ADD R7,R7,R3	R7=R7+R3
I_6	MUL R2,R3,#3	R2=R3*3
I_7	ADDI R6,R2,#1	R6=R2+1

Welche echten Datenabhängigkeiten bestehen in dieser Befehlsfolge? Stellen Sie die echten Datenabhängigkeiten in einem Präzedenzgraphen dar. Hinweis: Außer den echten Datenabhängigkeiten sollen keine anderen Abhängigkeiten in den Graphen eingetragen werden.
Benutzen Sie die folgende Darstellungshilfe für den Graphen!



Lösungsvorschläge



5 Organisation von Cache-Speichern

Gegeben sei ein Cache mit direkter Zuordnung. Der Cache umfasst 8 Blöcke, wobei jeder Block 32 Byte umfasst. Der Speicher soll byteadressierbar sein. Adressen sind 24 Bit lang (Bit 23 bis Bit 0). Der Cache habe eine Zugriffszeit von 1 ns, der Speicher eine Zugriffszeit von 19 ns, bei einem Miss soll der Zugriff sequentiell erfolgen.

a) Geben Sie an, wieviele Bits und welche Bits der Adresse zu Tag, Index, Wortadresse gehören.

b) Bestimmen Sie für eine Miss-Rate von 20% die mittlere Zugriffszeit t_{eff}.

Lösungsvorschläge

Zu a) Wortadresse: 5 Bit (Bit 4 bis Bit 0)
Index: 3 Bit (Bit 7 bis Bit 5)
Tag: 16 Bit (Bit 23 bis Bit 8)

Zu b) Bei einer Miss-Rate von 20% gilt $h = 0,8$. Es gilt weiterhin wegen des sequentiellen Zugriffs beim Miss: $t_{eff} = h \cdot 1 + (1 - h) \cdot (1 + 19)$.
Damit folgt $t_{eff} = 0,8 + 0,2 \cdot 20 = 4,8\ ns$.

6 Cachezugriffe

a) Gegeben sei ein Cache mit fester Zuordnung, der 16 Blöcke mit je 16 Bytes enthält. Der Speicher ist byteweise adressierbar. Gegeben ist eine Folge von Lese-Zugriffen auf die Adressen (alle hexadezimal 3-stellig):

D3F 4B6 E52 D39 4CC 4C8 E5E 211 4BF F2A 216

Bestimmen Sie die Anzahl der Cache Misses, wenn der Cache zu Beginn leer ist.

b) Zum Knobeln für Zusatzpunkte:
Gegeben sei ein Cache mit freier Zuordnung, der unendlich viele Blöcke enthält, die jeweils nur aus einem Byte bestehen. Der Speicher ist byteadressierbar. Ein Programm führt bei anfänglich leerem Cache 1.000.000 Zugriffe aus und greift dabei auf 10.000 verschiedene Bytes zu. Wie hoch ist die Hit-Rate? Begründen Sie Ihre Antwort.

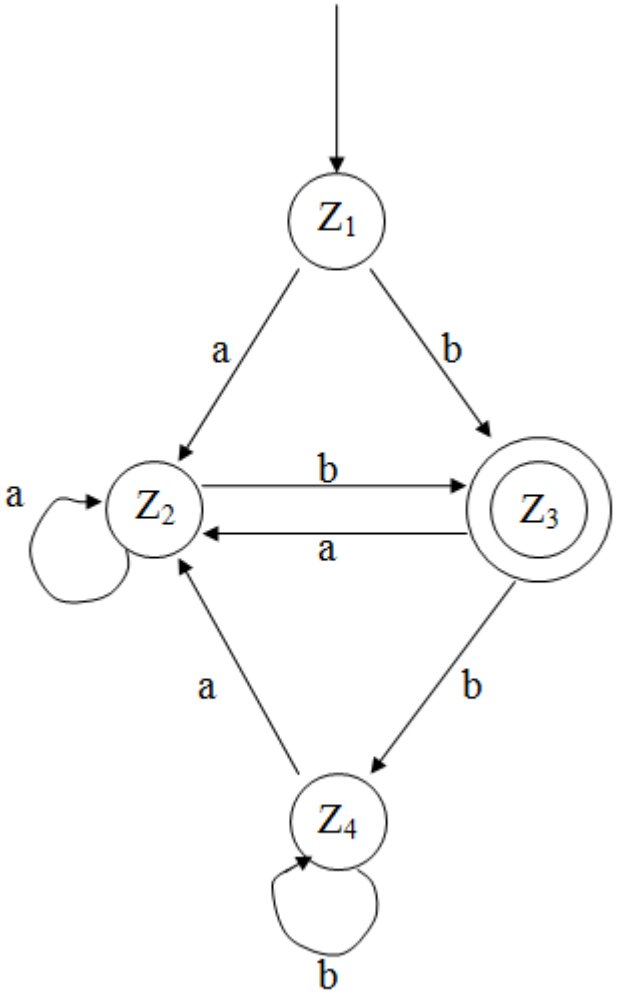
Lösungsvorschläge

Zu a) Die Wortadresse entspricht gerade der untersten Hexadezimalstelle, der Index entspricht der mittleren Hexadezimalstelle, die oberste Hexadezimalstelle entspricht dem Tag. Man muss also für jede Adresse lediglich die mittlere Hex-Ziffer betrachten, und prüfen ob diese Hex-Ziffer vorher bereits als mittlere Ziffer in einer Adresse vorhanden war, und falls ja, ob die linken Ziffern, d.h. die Tags, übereinstimmen. Dann liegt ein Hit vor, ansonsten ein Miss. Es ergeben sich 6 Misses.

Zu b) Bei einem Cache mit freier Zuordnung und unendlicher Größe gibt es keine Verdrängung, weder durch einen Konflikt bei der Zuordnung noch beim Einlagern eines Blocks, wenn alle Blöcke bereits belegt sind. Ein Miss kommt daher nur beim ersten Zugriff auf ein Byte (=Block) vor. Damit kommt es zu einer Miss-Rate von $10^4/10^6 = 1\%$. Die Hit-Rate ist folglich 99%

7 Endliche Automaten

Gegeben sei ein endlicher Automat ohne Ausgabe mit dem Eingabe-Alphabet $\{a, b\}$ und folgendem Zustandsdiagramm:



a) Finden Sie das kürzeste Wort, das der Automat akzeptiert.

b) Gibt es Wortlängen, so dass der Automat kein Wort dieser Länge akzeptiert? Begründen Sie Ihre Antwort.

Lösungsvorschläge

Zu a) Das kürzeste Wort das der Automat akzeptiert ist b.

Zu b) Die einzige Wortlänge, bei der kein Wort akzeptiert wird, ist die Wortlänge 0, denn das leere Wort wird nicht akzeptiert. Bei Wortlängen ≥ 2 akzeptiert der Automat die Worte, die als letztes Zeichen ein b und als vorletztes Zeichen ein a enthalten. Solche Worte gibt es für jede der genannten Wortlängen.

8 Berechenbarkeit und Komplexität

Das statische Job Scheduling Problem mit Schranke erhält als Eingabe eine Liste von Aufgaben T_1 bis T_n, jeweils mit Laufzeit r_i, sowie eine Schranke M. Zu entscheiden ist, ob die Aufgaben so auf $p < n$ vorgegebene Prozessoren verteilbar sind, dass keiner der Prozessoren länger als M braucht, um die ihm zugewiesenen Aufgaben zu bearbeiten.

Zeigen Sie, dass dieses Problem in NP liegt, d.h. geben Sie textuell ein Verfahren an, das entscheidet, ob ein Schedule (d.h. eine gegebene Verteilung der Aufgaben auf Prozessoren) die Schranke M erfüllt oder nicht. Das Verfahren muss eine Laufzeit haben, die polynomiell in n ist.

Lösungsvorschläge

Man durchläuft p-mal die Liste der Aufgaben. Im Durchlauf i addiert man die Laufzeiten aller Aufgaben, die Prozessor i zugewiesen wurden, und prüft am Ende des Durchlaufs, ob diese Summe höchstens M ist. Ist die Bedingung für alle Durchläufe erfüllt, gibt man JA aus, sonst NEIN. Das Verfahren hat eine Laufzeit von $O(n \cdot p) = O(n^2)$.