Prüfungsklausur 31231 – SS 14

Prof. Dr. J. Keller

23.08.2014

Inhaltsverzeichnis

Bewertungsschema

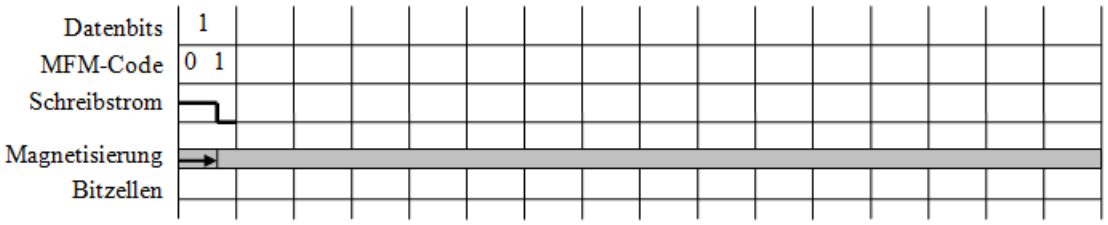
Aufgabe	a	b	c	d	e	f	g	h	total
1									6
2									6
3									4
4									6
5	3	2	3	3	3				14
6	2	4	4	4					14

1 Codierungsverfahren

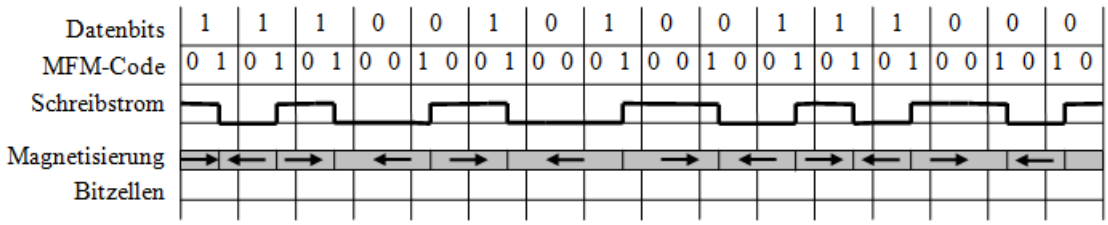
Die Datenbit-Folge D = 1110 0101 0011 1000 soll auf einer Festplatte mit dem MFM-Verfahren aufgezeichnet werden. Tragen Sie dazu in das folgende Bild die aufgezeichneten Datenbits, die resultierende MFM-Codierung, den Verlauf des Schreibstroms sowie die Magnetisierungsrichtung der Festplatte ein. Als Rastergröße in waagerechter Richtung ist darin die Länge einer Bitzelle vorgegeben, d.h. der kleinste Abschnitt gleichgerichteter Magnetisierung. Ordnen Sie die Daten- und Code-Bits jeweils über den Bitzellen an, in denen sie abgespeichert werden. (6 P.)

Hinweis: Die folgende Tabelle zeigt die Zuordnung der Datenbits bei der MFM-Codierung:

Datenbit		Speichercode
D_{n-1}	D_n	
0	0	10
1	0	00
0	1	01
1	1	01



Lösungsvorschläge



2 Speichermedien und Peripheriegeräte

Welche der folgenden Aussagen treffen zu? (6 P.)

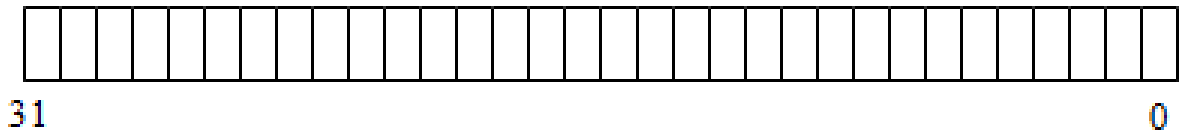
trifft zu:	trifft nicht zu:	
☐	☐	Auf einer CD-ROM werden die Daten in einer einzigen spiralförmigen Spur geschrieben.
☐	☐	Beschriebene CD-Rs werden auch als Rohlinge bezeichnet.
☐	☐	Der Trackball ist ein kleiner, in die Tastatur von Notebooks integrierter Stift zum Bewegen des Mauszeigers.
☐	☐	Der Scancode setzt sich aus dem Dualcode der Spalten- und Zeilennummern der Tasten einer Tastatur zusammen.
☐	☐	Linux verwaltet die Festplattenadressen durch Verkettung einzelner Speicherblöcke. Diese Speicherblöcke werden auch Inodes genannt.
☐	☐	LCDs basieren auf den optischen Eigenschaften von Flüssigkeitskristallen, die aus durchsichtigen organischen Molekülen bestehen.

Lösungsvorschläge

Die erste, vierte und sechste Aussage treffen zu.

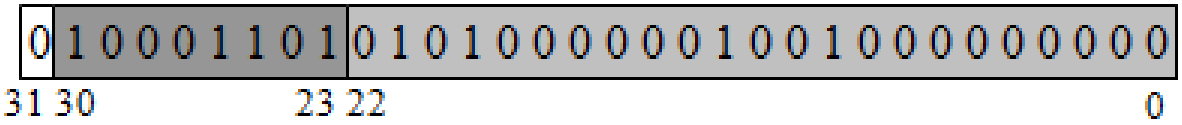
3 Datenformate

Stellen Sie die Dezimalzahl $Z_1 = 22513$ im 32-bit-Format des IEEE-754-Standards in binärer Form dar. Kennzeichnen Sie die unterscheidbaren Bitfelder (Vorzeichen, biased Exponent, Mantisse). (4 P.)



Lösungsvorschläge

$Z_1 = 22513 = 2^{14}+2^{12}+2^{10}+2^3+2^0 = 22513 = (-1)^0 \cdot 2^{141-127} \cdot (1.010100000010010\cdots 0)_2$
Daraus ergibt sich:

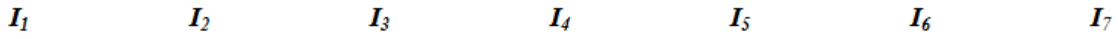


4 Pipeline-Konflikte

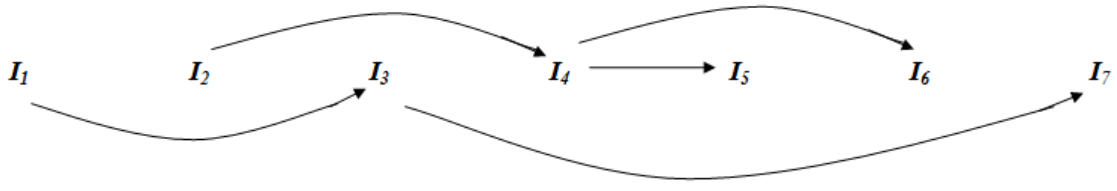
Wir betrachten die folgende Adress-Befehlsfolge:

	Befehl	Bedeutung
I_1	ADD R1,R2,R3	R1=R2+R3
I_2	ADDI R4,R5,#1	R4=R5+1
I_3	MUL R3,R1,#2	R3=R1*2
I_4	ADD R5,R4,R6	R5=R4+R6
I_5	ADD R6,R5,R2	R6=R5+R2
I_6	MUL R6,R5,#3	R6=R5*3
I_7	ADDI R4,R3,#1	R4=R3+1

Welche echten Datenabhängigkeiten bestehen in dieser Befehlsfolge? Stellen Sie die echten Datenabhängigkeiten in einem Präzedenzgraphen dar. (6 P.) 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

a) Gegeben sei ein Cache C1 mit voller Zuordnungsfreiheit (vollassoziativer Cache). Der Cache umfasst 1024 Blöcke von je 32 Bytes. Der Speicher ist byte-weise adressierbar, Adressen sind 24 Bit lang (Bit 23 bis Bit 0).

Geben Sie an, wieviele Bits der Adresse jeweils zu Tag, Index, und Wortadresse gehören.(3 P.)

Tag:

Index:

Wortadresse:

b) Ändert sich die Anzahl der Bits des Index aus Teil a), wenn es sich um einen Cache mit direkter Zuordnung statt voller Zuordnungsfreiheit handelt? Wenn ja, auf welchen Wert, wenn nein, warum nicht? (2 P.)

c) Gegeben ist ein Cache C2 mit direkter Zuordnung (direct-mapped Cache).

Es gilt: Wortadresse=4 Bit, Index=8 Bit, Tag=4 Bit.

C2 soll zu Beginn leer sein, dann sollen Lesezugriffe auf die folgenden hexadezimalen Adressen durchgeführt werden. Bestimmen Sie die Anzahl der Misses.

0112 0121 0110 1110 0110 0123 012F 1110 (3 P.)

d) Bestimmen Sie die mittlere Zugriffszeit t_{eff} eines Caches bei einer Hit-Rate von 80%. Der Cache habe eine Zugriffszeit von 10 ns, der Speicher eine Zugriffszeit von 70 ns, bei einem Miss soll der Zugriff sequentiell erfolgen. (3 P.)

e) In einem physikalischen Speicher der Größe 10^5 (Adressen 0 bis 99.999) gibt es drei Segmente (ohne Seitenverwaltung), die sich an den Adressbereichen (dezimal)

0	bis	5.999
10.000	bis	69.999
80.000	bis	94.999

befinden, so dass sich im Adressraum drei Lücken befinden.

Es soll ein weiteres Segment der Größe 5.000 (dezimal) platziert werden. Geben Sie an, in welche Lücke das Segment bei Benutzung von first fit, best fit, und worst fit jeweils platziert wird. Begründen Sie jeweils Ihre Antwort. (3 P.)

Lösungsvorschläge

Zu a) Wortadresse: 5 Bit, Index: 0 Bit, da volle Zuordnungsfreiheit, Tag: $24-5 = 19$ Bit.

Zu b) Ja, sie ändert sich auf 10 Bit.

Zu c) Die Anzahl der Misses ist 5, nur beim 3., 6. und 7. Zugriff gibt es einen Hit.

Zu d) Es gilt $t_{eff} = 0,8 \cdot 10 + 0,2 \cdot (10 + 70) = 24\ ns$.

Zu e) Bei first fit wird das Segment an die Stelle 70.000 platziert, da die Lücke ab 6.000 zu klein ist.
Bei best fit wird das Segment an die Stelle 95.000 platziert, da diese Lücke so großwie das Segment ist.
Bei worst fit wird das Segment an die Stelle 70.000 platziert, da diese Lücke die größte ist.

6 Endliche Automaten

a) Gegeben ist das Alphabet $\{a, b\}$. Geben Sie einen regulären Ausdruck an, dessen zugeordnete Sprache $ba, bab, baba, babab, bababa, \ldots$ ist. (2 P.)

b) Gegeben ist das Alphabet $\{a, b, c\}$. Geben Sie einen deterministischen endlichen Automaten ohne ε-Regeln an, der die Sprache akzeptiert, die dem Ausdruck $a(cb)^*$ zugeordnet ist. (4 P.)

c) Gegeben ist ein Problem aus $\mathcal{NP}$ sowie ein Algorithmus, der für eine Eingabe der Größe n eine Lösung mit $365 \cdot 10^n$ Operationen berechnet. Berechnen Sie, bis zu welcher Größe n_0 dieser Algorithmus Lösungen innerhalb von 10 Jahren findet. Wie lange braucht der Algorithmus, um eine Lösung für eine Eingabe der

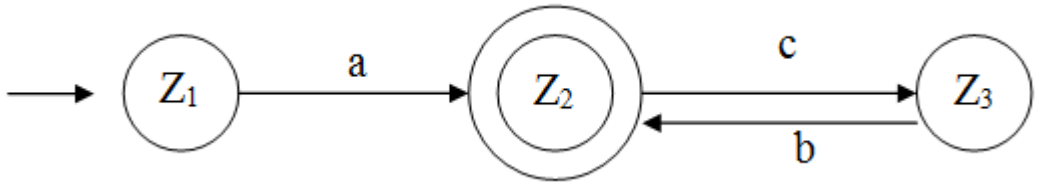
Größe $n_0 + 2$ zu finden? Gehen Sie hierbei davon aus, dass an einem Tag 10^{13} Operationen durchgeführt werden können, und dass ein Jahr stets 365 Tage hat. (4 P.)

d) Gegeben seien 90 Gegenstände der Größe jeweils $7m^3$, und 90 Gegenstände der Größe jeweils $3m^3$. Diese Gegenstände sollen in Container der Größe $10m^3$ gepackt werden. Die Gegenstände sollen verformbar sein, so dass für die Frage ob Gegenstände in einen Container passen nur das Volumen, nicht aber die konkrete Form berücksichtigt werden muss. Wieviele Container benötigt man mindestens? Wieviele Container benötigt der First-Fit-Decreasing Algorithmus? (4 P.)

Lösungsvorschläge

Zu a) $ba(ba)^* + bab(ab)^*$

Zu b)



Zu c) In 10 Jahren können $10 \cdot 365 \cdot 10^{13} = 365 \cdot 10^{14}$ Operationen durchgeführt werden. Damit kann eine Lösung bis zur Größe $n_0 = 14$ berechnet werden. Für eine Eingabe der Größe $n_0 + 2$ braucht der Algorithmus $365 \cdot 10^{n_0+2} = 100 \cdot 365 \cdot 10^{n_0}$ Operationen. Da $365 \cdot 10^{n_0}$ Operation gemäß der Aufgabenstellung gerade 10 Jahre dauern, braucht der Algorithmus 1000 Jahre für eine Eingabe der Größe $n_0 + 2$.

Zu d) Wenn man einen großen Gegenstand und einen kleinen Gegenstand in einen Container packt, dann ist dieser komplett gefüllt. Man braucht also 90 Container. Der First-Fit-Decreasing Algorithmus packt die 90 großen Gegenstände in 90 Container, und packt jeweils 3 kleine Gegenstände in einen Container, d.h. er braucht 30 weitere Container, also insgesamt 120 Container.