Sabile Güzel	Donnerstags-Tutorium
Abgegeben von:	
Naja von Schmude	M-Nr.: 4127652
Norman Warnatsch	M-Nr.: 4123733

TI 1.Übungsblatt

1.Aufgabe

Betriebssystem: Das Betriebssystem (engl. operating system) ist ein in Maschinensprache geschriebenes Programm, welches die Betriebsmittel wie Speicher, Ein- und Ausgabegeräte verwaltet, die Ausführung von Programmen steuert und die Kommunikation (Interaktion) zwischen Mensch und Computer ermöglicht.

Compiler: Unter einem L_i-Compiler versteht man ein Maschinenprogramm, das ein Programm P_i der Sprache L_i in ein Programm P_j der Sprache L_j transformiert, das äquivalent zu P_i ist, d.h. das gleiche Ein-/Ausgabeverhalten wie P_i hat.

Assembler: Ein Assembler ist ein spezieller Compiler, der ein in einer maschinennahen Assemblersprache geschriebenes Programm in Maschinensprache übersetzt.

Hauptspeicher: Im Hauptspeicher werden DRAM-Speicherzellen, oder Erweiterungen von diesen, eingesetzt. Diese sind matrixförmig angeordnet.

Cachespeicher: Der Cachespeicher ist in der Speicherhierarchie zwischen den CPU-internen Registern und dem Hauptspeicher angesiedelt. Sinn und Zweck des Cache-Speichers ist es, die große Performanzlücke, durch recht kleinen Speicher, der sehr schnelle Zugriff erlaubt, zwischen Register und Hauptspeicher bestmöglich zu schließen.

Nichtflüchtiger Speicher: NVRAM (Abk. Non Volatile Random Access Memory) ermöglicht eine Speicherung der Informationen auch ohne Aufrechterhaltung der Energieversorgung.

Flüchtiger Speicher: Die volatilen (flüchtigen) Speicher, wie DRAM oder SRAM, verlieren ihre gespeicherten Informationen bei fehlender Energieversorgung.

2.Aufgabe

a.) 1970 wurde der erste kommerziell erhältliche DRAM-Chip des Typs 1103 von Intel vorgestellt. Es waren 1024 Speicherzellen (1kBit) enthalten. Seither konnte die Kapazität eines DRAM-Chips um den Faktor 1 Mio. gesteigert und die Zugriffszeit auf ein Zehntel verkürzt werden.
Im Jahr 2006 besaßen DRAM-Chips Kapazitäten von bis zu 2 GByte und Zugriffszeiten von 10 ns.

b.) Nicht nur die Prozessorentwicklung , heute soweit fortgeschritten, dass die Zahl der Transistoren auf einem Chip sich um das 20 000 – fache des 4004 erhöhte, sondern auch die Weiterentwicklung der Speichermedien und –Kapazität sind maßgeblich für eine Leistungssteigerung verantwortlich. Der Festplattenspeicher wuchs von anfänglichen 5 MB zu Beginn der 80-er Jahre auf heute bis zu über 250 GB, und das bei gleichzeitiger Reduktion des Platzbedarfes.

c.) 1965 hat Gordon Moore, der später mit Bob Noyce Intel gründete, die Prognose abgegeben, die Kapazität eines Computers werde sich jedes Jahr verdoppeln. Zu diesem Ergebnis kam er, nachdem er das Preis-Leistungs-Verhältnis von Computerchips für die letzten 3 Jahre untersuchte und in die Zukunft projiziert hatte. In Wahrheit glaubte Moore nicht, daß dieses Entwicklungstempo lange anhalten werde. Doch noch zehn Jahre später erwies sich seine Prognose als zutreffend, woraufhin er vorhersagte, die Kapazität werde sich fortan alle zwei Jahre verdoppeln. Bis auf den heutigen Tag haben sich seine Vorhersagen bestätigt und ihr Mittelwert – eine Verdopplung alle anderthalb Jahre – wird unter Informatikern als Mooresches Gesetz bezeichnet. In diesem Zusammenhang sei erwähnt, dass von Intel ein Entwicklungsplan erstellt wurde, der die Weiterentwicklung der Prozessortechnik im Rahmen des Mooreschen Gesetzes bis ins Jahr 2020 gewährleistet.

3.Aufgabe

Hätte sich die Geschwindigkeit der Kraftfahrzeuge in ähnlicher Art und Weise erhöht, wie dies bei der Taktfrequenz der Mikroprozessoren erfolgt ist, so könnte man heute in gerade mal 13 Sekunden mit dem Auto von San Fransisco nach New York gelangen.

Auto des Jahres 1977: v= 150 km/h
Verbrauch = 15 l / 100 km

Durchschnittliche Leistungssteigerung: 35% pro Jahr bis `87
50% pro Jahr bis `87

S_1 -> 1,35 bei (Jahr $\leq$1987)
-> 1,5 bei (Jahr > 1987)
n = Jahr - 1977

$v_{Jahr} = v*(S_1)^n$

$verbauch_{Jahr} = 15\ l /[100\ km * (S1)^n]$

$v_{1987}= 150\ km/h * (1,35)^{10} = 3015,83\ km/h$

$verbrauch_{1987} = 15\ l / [\ 100\ km * (1,35)^{10}] = 15\ l / 2010,655\ km = 0,00746\ l/km$

$v_{2007} = 150\ km/h*(1,35)^{10}*(1,5)^{20} = 10028919,033\ km/h$

$verbrauch_{2007} = 15\ l / [100\ km * (1,35)^{10}*(1,5)^{20}\ ] = 0,000002244\ l/km$

Aufgabe 4

1.)

$12_{10} => 00001100_2$

$+ 43_{10} => 00101011_2$

└→ $- 43_{10} => 11010100_2 + 00000001_2 => 11010101_2$

$125_{10} => 01111101_2$

$+ 77_{10} => 01001101_2$

└→ $- 77_{10} => 10110010_2 + 00000001_2 = 10110011_2$

$+ 80_{10} => 0101000_2$

└→ $- 80_{10} => 10101111_2 + 00000001_2 => 10110000_2$

2.)

001 1 1 00 1 1 1 0 1 1 1 1 0 1 $_2$

$= 2^{14}+2^{13}+2^{12}+2^9+2^8+2^7+2^5+2^4+2^3+2^2+2^0 = 29629_{10}$

29629: 4 = 7407	Rest 1
7407 : 4 = 1851	Rest 3
1851 : 4 = 462	Rest 3
462 : 4 = 115	Rest 2
115 : 4 = 28	Rest 3
28 : 4 = 7	Rest 0
7 : 4 = 1	Rest 3
1 : 4 = 0	Rest 1

$13032331_4 = 29629_{10}$???

111	001	110	111	101_2	
7	1	6	7	5	→ 71675_8

$7*8^4 + 1*8^3 + 6*8^2+7*8^1+5*8^0$

$28762+ 512 + 384 + 56 + 5 = 29629_{10}$, daher 71675_8 korrekt

A = 10; B=11;

C= 12; D= 13;

0111	0011	1011	1101	
|	|	11	13	
7	3	B	D	→ $73BD_{16}$

<u>Aufgabe 4</u>

3.) **→ $254{,}137_{10}$**

254 : 16 = 15	Rest 14	=	E
15 : 16 = 0	Rest 15	=	F

A = 10	B = 11
C = 12	D = 13
E = 14	F = 15

0,137 * 1 = 0	Rest 0,137
0,137 * 16 = 2	Rest 0,192
0,192 * 16 = 3	Rest 0,072
0,072 * 16 = 1	Rest 0,152
0,152 * 16 = 2	Rest 0,432

<u>$254{,}137_{10}$ = FE + 0,2312 = FE, 2312_{16}</u>

→ $11466{,}8765_{10}$

11466 : 16 = 716	Rest 10 = A
716 : 16 = 44	Rest 12 = C
44 : 16 = 2	Rest 12 = C
2 : 16 = 0	Rest 2

0,8765 * 1 = 0	Rest 0,8765
0,8765 * 16 = 14 → E	Rest 0,024
0,024 * 16 = 0	Rest 0,384
0,384 * 16 = 6	Rest 0,144
0,144 * 16 = 2	Rest 0,304

<u>$11466{,}8765_{10}$ = 2CCA + 0,E062 = 2CCA, $\mathrm{E062}_{16}$</u>

→ $1234{,}4321_{10}$

1234 : 16 = 77	Rest 2
77 : 16 = 4	Rest 13 = D
4 : 16 = 0	Rest 4

0,4321 * 1 = 6	Rest 0,4321
0,4321 * 16 = 6	Rest 0,9136
0,9136 * 16 = 14 → E	Rest 0,6176
0,6176 * 16 = 9	Rest 0,8816
0,8816 * 16 = 14 → E	Rest 0,1056

<u>$1234{,}4321_{10}$ = 4D2 + 0,6E9E = 4D2, $\mathrm{6E9E}_{16}$</u>

Aufgabe 4

4.)

Addition

$1,1001*2^3 + 1,1101*2^4$

$d_1 = 1,1101*2^4 = M_1*2^4$

$d_2 = 1,1001*2^3 = (1,1001*2^{3-4})*2^4$

$= 0,11001*2^4 = M_2*2^4$

$s = (M_1+M_2)2^4$

$s = (1,1101 + 0,11001)*2^4$

$$
\begin{array}{r}
1,11010 \\
\underline{+\ {}_1 0_1,1_1 1001} \\
10,10011
\end{array}
$$

$\underline{s = (10,10011)*2^4 = 1,010011 * 2^5}$

Multiplikation

$1,1001*2^3*1,1101*2^4$

$\langle p\rangle = \langle d_1\rangle * \langle d_2\rangle$

$d_1 = 1,1001 * 2^3$

$d_2 = 1,1101 * 2^4$

$p = (M_1*M_2)*s^{E1+E2}$

$p = (1,1001 * 1,1101) *2^{3+4}$

$$
\begin{array}{r}
\underline{1,1001 * 1,1101} \\
11001 \\
11001 \\
11001 \\
\underline{{}_1 1_1 1_1 0_1 0_1 1_1} \\
\underline{10,11010101}
\end{array}
$$

$\underline{p = 10,11010101*2^7 = 1,011010101*2^8}$

Aufgabe 4

5.)

Anmerkung: Diese Aufgabe wurde nach dem Schaubild im Tutorium erarbeitet und auch nur nach dessen Richtlinien errechnet, darunter wären:
Normalisierung hat laut Tutorium somit die Form $0{,}1100*2^E$, und nicht, wie im Buch „Technische Informatik-Einführung" von Bernd Becker, Rolf Drechsler und Paul Molitor, in dem die Normalisierte Form mit $1{,}100*2^E$ gekennzeichnet wird, und das Vorkomma-Bit als hidden bit später weggelassen wird. Nachteil der Unterschiede zwischen Buch und Tutorium ist, dass der Exponent sich verändert, wodurch automatisch die Charakteristik verändert wird. Mantisse bleibt gleich, da auch beim Tutorium die erste 1 entfernt wird. Dadurch entsprechen sich die Mantissen wieder.
„Die Einschränkung besteht darin, dass für die Mantisse M einer normalisierten Gleitkommazahl stets $1 \leq M < 2$ gelten muss. Daraus folgt direkt die Eindeutigkeit der Zahlendarstellung. Außerdem ergibt sich dadurch aber noch ein weiterer Vorteil. Bedenkt man, dass die Mantisse als binäre Festkommazahl dargestellt in jedem Fall die Form 1, … hat, so ist offensichtlich, dass die führende Eins vor dem Komma nicht eigens repräsentiert werden muss. Es genügt also, in der Bitfolgen-Darstellung die Nachkommastellen als Binärzahl abzulegen. Die Bits $m_{22} \ldots m_0$ werden als Nachkommastellen einer Festkommazahl aufgefasst, wobei m_{22} die höchste Wertigkeit, also 2^{-1}, besitzt. Diese Festkommazahl wird in eine dezimale Darstellung gebracht, und die Addition von eins sorgt für die führende eins vor dem Komma, die wegen der impliziten Repräsentation allgemein auch als hidden bit bezeichnet wird."

Und so beginnen wir die Aufgabe:

→ $-45{,}5625_{10}$ 1 Bit Vorzeichen-> 1 weil negativ.

45 : 2 = 22	Rest 1	0,5625 * 2 = 1	Rest 0,125
22 : 2 = 11	Rest 0	0,125 * 2 = 0	Rest 0,25
11 : 2 = 5	Rest 1	0,25 * 2 = 0	Rest 0,5
5 : 2 = 2	Rest 1	0,5 * 2 = 1	Rest 0
2 : 2 = 1	Rest 0		
1 ; 2 = 0	Rest 1		

$+\ 45,\ 5625_{10} = 101101{,}1001_2*2^0$
$= 0{,}1011011001_2*2^6$

$-\ 45{,}562510 = (-1)^1 * (0{,}1011011001_2)*2^6$
$6 = \text{char} - 2^{(15-1)-11}$
$6 = \text{char} - 2^3$
$6 = \text{char} - 8$
$14 = \text{char}$

14 : 2 = 7	Rest 0
7 : 2 = 3	Rest 1
3 : 2 = 1	Rest 1

$1 : 2 = 0$ Rest 1

Charakteristik = 1110_2

→ VZ Char Mantisse

1	1110	0101101100

→ $125,1253_{10}$

125 : 2 = 62	Rest 1	0,1253*2 = 0	Rest 0,2506
62 : 2 = 31	Rest 0	0,2506*2 = 0	Rest 0,5012
31 : 2 = 15	Rest 1	0,5012*2 = 1	Rest 0,0024
15 : 2 = 7	Rest 1	0,0024*2 = 0	Rest 0,0048
7 : 2 = 3	Rest 1	0,0048*2 = 0	Rest 0,0096
3 : 2 = 1	Rest 1		
1 : 2 = 0	Rest 1		

$125,1253_{10} = 1111101,00100_2*2^0$
$= 0,111110100100_2*2^7$
$= (-1)^0*(0,11111010010)*2^7$

$7 = \text{Char} - 2^{(15-1)-11}$
$7 = \text{Char} - 2^3$
$15 = \text{Char}$

15 : 2 = 7	Rest 1
7 : 2 = 3	Rest 1
3 : 2 = 1	Rest 1
1 : 2 = 0	Rest 1

Charakteristik = 1111_2

→ VZ char Mantisse

0	1111	11110100100

Aufgabe 5.1

Die eigentliche Darstellung unserer Zahl ist in der Mantisse zu finden. Je mehr Bit für die Mantisse zur Verfügung stehen, desto genauer kann die Zahl dargestellt werden. Man muss jedoch bedenken, dass die Anzahl der reservierten Bits für die Gleitkommazahl begrenzt ist. Logische Schlussfolgerung daraus ist, dass je mehr Bits für die Mantisse verwendet werden, desto weniger Bits hat die Charakteristik zu Verfügung. Da die Charakteristik entscheidend für den Exponenten ist, wäre damit auch die Größe des Exponenten begrenzt. Schlussendlich bedeutet dies nichts anderes, dass die darstellbare Zahl mit größerer Mantisse genauer dargestellt werden kann, aber sie dafür in einem kleineren Zahlenbereich vorhanden ist. Je mehr Bits für die Charakteristik reserviert werden, desto weniger bleiben für die Mantisse. Die Zahl kann zwar in einem großen Zahlenbereich vorhanden sein, wird aber ungenauer.

Zur Veranschaulichung nehmen wir eine Reservierung von 10 Bits, 1 Bit ist reserviert für das Vorzeichen, bleiben noch 9 übrig. Wir entscheiden uns für zwei Extremfälle:

I.) 8 Bits sind für die Charakteristik reserviert, eines für die Mantisse.
Das heißt, die Charakteristik kann im extremsten fall aus lauter 1 bestehen: Char = 11111111_2 , was $2^7+2^6+2^5+2^4+2^3+4+2+1$ bedeutet.
Char = $2^7+2^6+2^5+2^4+2^3+4+2+1 = 128 + 64 + 32 + 16+8+4+2+1= 255$
Exponent = $255 – 2^{(9-1)-1} = 127$
Dezimal = $(-1)^{VZ}*(0,\text{Mantisse})*2^{127}$
Dezimal = $(-1)^{VZ}*1{,}7014*10^{37}$, wobei die 1-Bit-Mantisse 1 ist.
Das wäre die maximal darstellbare Zahl bei den gewählten Parametern, aber wie sieht es mit der kleinsten Zahl bei dieser Bitverteilung zwischen Mantisse und Charakteristik aus?
Annahme Char = 00000000_2
Exponent =Char – $2^{(9-1)-1}$
Exponent = $- 2^7 = - 128$
Dezimal = $(-1)^{VZ}*(0,\text{Mantisse})*2^{-128}$,wobei 1-Bit-Mantisse
Dezimal = $(-1)^{VZ}*2{,}938735*10^{-40}$
Dass ist der Zahlenbereich, in dem die Mantisse darstellbar ist. Ein großer Bereich, bei näherer Betrachtung kann man jedoch feststellen, dass die Genauigkeit stark gelitten hat. Es können gerade mal [2*(128+127)+1]= 511 Zahlen in diesem riesigen Bereich dargestellt werden. dargestellt werden. Dass rührt daher, weil der Nachfolger einer dieser Gleitkommazahlen das doppelte seines Vorgängers ist. Sprich: Je größer die Zahl, desto größer ist der Abstand zu dem Vorgänger so wie dem Nachfolger der darstellbaren Zahlen geworden (um den Faktor 2). 128 für die Anzahl der Zahlen, die mit positiven Exponenten berechnet werden, 127 für diejenigen, die mit negativen Exponenten berechnet werden, diese werden mit 2 multipliziert, da diese ja auch noch im negativen Zahlenbereich dargestellt werden können, und 1 für die Mantisse, wenn sie 0 ist.

II.) nun betrachten wir den genau anderen Fall, 8 Bits für Mantisse und 1 Bit Charakteristik.

Char = 1 oder Char = 0

Exponent = $1- 2^{(9-1)-1}$

Exponent = -127

Dezimal = $(-1)^{VZ}*(0,11111111)*2^{-127}$

Ist die größte Zahl in dem positiven Wertebereich. Und die kleinste ist???

Char = 0

Exponent = $0 – 2^{(9-1)-1}$

Exponent = -128

Dezimal = $(-1)^{VZ}*(0,00000000)*2^{-128}$

Dies ist die kleinste im positiven Bereich. Man erkennt sofort, dass dieser Zahlenbereich um einiges kleiner ist als der in I.) mit einer 8-Bit-Charakteristik und einer 1-Bit-Mantisse.

Um nun festzustellen, ob in diesem Zahlenbereich, wie in meiner Annahme behauptet, eine größere Genauigkeit festzustellen, reicht es schon, die Anzahl der Elemente zu zeigen: Die Mantisse kann 2^8 unterschiedliche Zahlen angeben, hinzu kommt, dass wir dies mal 2 multiplizieren, wegen der Vorzeichen , und dann mal 2 wegen 2 möglichen unterschiedlichen Exponenten. So kommen wir auf eine Anzahl von 2^8*4= 1024 Zahlen, während in I.) gerade mal 511 Zahlen darstellbar sind, und dass trotz eines größeren Zahlenbereichs.

Damit dürfte gezeigt sein, dass je größer die Anzahl der Bits für die Charakteristik, desto geringer die durch die Mantisse darstellbare Genauigkeit. Anders rum gilt jedoch auch: Je größer die Anzahl der Bits für die Mantisse, desto höher ist die Genauigkeit, aber desto kleiner ist der durch die Charakteristik darstellbare Zahlenbereich.

Aufgabe 5.2

- Die darstellbaren Zahlen liegen im Bereich des Nullpunktes enger beieinander als bei Festkommadarstellungen, jedoch wächst der Abstand zwischen den darstellbaren Zahlen mit steigendem Betrag.
- Bei genauerer Betrachtung der Gleitkommadarstellung fällt auf, dass der Dezimalwert einer normalisierten Zahl einem Produkt dreier Faktoren entspricht, von denen keiner den Wert 0 annehmen kann. Damit kann der Zahlenwert nicht null werden, weshalb die Null nicht durch normalisierte Zahlen darstellbar ist.
- Bestimmte Bereiche der Zahlen sind mit den Gleitkommazahlen nicht darstellbar, egal wie genau die zahl ist. Gerade im Bereich um die Null wird gezeigt, dass diese nie erreicht werden kann, eine Nährung an Null ist möglich, dass erreichen dieser jedoch nicht.
- Die Zahlendarstellung bei Gleitkommazahlen ist nicht „von der Natur aus“ eindeutig. Dies liegt daran, dass eine Vervielfachung der Mantisse durch entsprechendes Anpassen des Exponenten neutralisiert werden kann. Mehrdeutigkeiten bei der Zahlendarstellung sind jedoch nicht wünschenswert, weshalb durch eine zusätzliche Einschränkung eine Normalform erzwungen wird.