CONCORDIA UNIVERSITY
Department of Computer Science and Software Engineering
COMP 228/4 Section PP
Midterm Exam

Instructor: Tadeusz S. Obuchowicz
Date: Tuesday, February 28, 2012
Time Allowed: 1 hour and 30 minutes

Instructions:

- Closed book/closed notes
- No calculators allowed, no laptops allowed, no cell phones allowed.
- Answer all 5 questions. If you make any assumptions clearly indicate so in your answer booklets.

<u>Question 1:</u> [20 points] (Integer binary numbers)

(a) Represent the decimal numbers **-9** and **-7** using

(i) **5-bit sign magnitude** notation
(ii) **5-bit two's complement** notation.

(b) Perform the operation -(9) + (-7) using the rules of 5-bit sign magnitude addition. Indicate whether overflow occurs.

(c) Perform the multiplication of (-5) x (+3) using the **Booth algorithm** assuming 5-bit two's complement notation is used for the multiplicand and multiplier. Show all your steps.

<u>Question 2:</u> [20 points] (Floating point representation)

Consider a **8-bit** floating point format similar to the IEEE 754 notation, but simplified somewhat such that it is amenable for hand calculations. The **base is 2** and the exponent is stored in **3 bits** using **excess-4** notation, the two **end values** of the stored exponent (namely 0 and 7) are used as "special cases" to represent exact 0.0 and infinity respectively. The most significant bit is used to represent the sign of the number with 0 designating a positive number and 1 representing a negative number. The mantissa is stored using **4 bits** and is normalized in a manner similar to that of the IEEE 754 format, i.e. there **is** a "hidden" or "implied" 1 to the left of the binary point.
It is often said that a picture is worth a thousand words, so Figure 1 illustrates this floating point format.

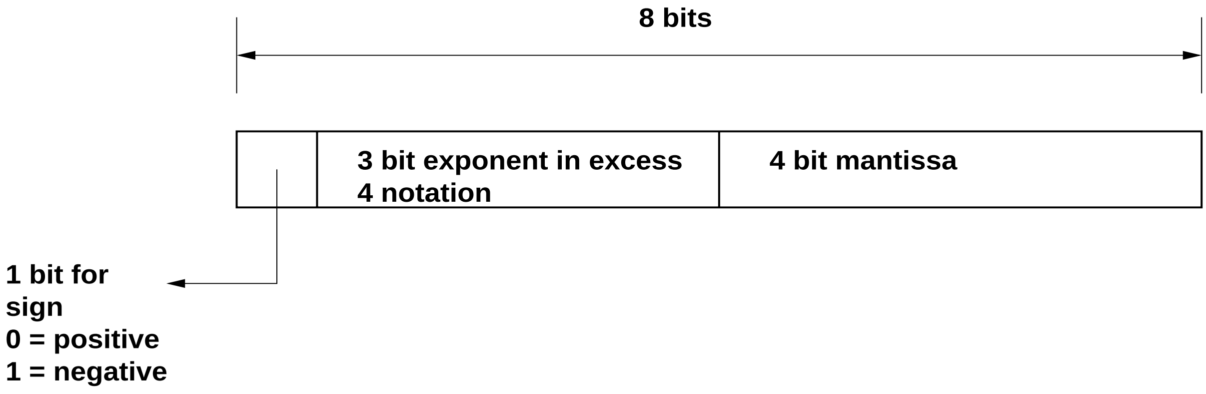

Figure 1: A 8-bit "mini IEEE" floating point format.

(a) Express the decimal number +6.9 using this format. Express your answer as a 8-bit binary number identifying clearly which bits represent the sign, exponent, and mantissa.

(b) What decimal number does the value $6B_{16}$ represent in this floating point format?

<u>Question 3:</u> [20 points] (Karnaugh Maps)

(a) Obtain the **minimal sum-of-products** Boolean expressions for the F_2, F_1, and F_0 outputs of the following truth table which has inputs X,Y, and Z. Use three separate 3-variable K-maps to perform the minimization. Indicate clearly on your K-maps the groupings of "1"s which make up the chosen prime implicants.

Table 1: A truth table

X	Y	Z	F_2	F_1	F_0
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

(b) Draw the logic circuits of the three sum-of-product equations obtained in part (a). You may assume that the true and complemented forms of the input variables are available.

(c) State in a clear English sentence the operation specified by the truth table and implemented by the hardware obtained in part (b). For example, you should phrase your answer similar to:

"The given truth table specifies that the output is the 3-bit Hamming code of the given input values"

Of course, the real answer is something completely different. This example is given merely to illustrate how one should phrase the answer.

<u>Note</u>: Answers of the form:

"The truth table specifies that when the input is 000, the output should be 000, when the input is 001, the output should be 111, when the input is 010 the output should be 110, etc".

will receive a grade of 0.

Hint: The hardware shown in Figure 1 gives the same results as the given truth table:

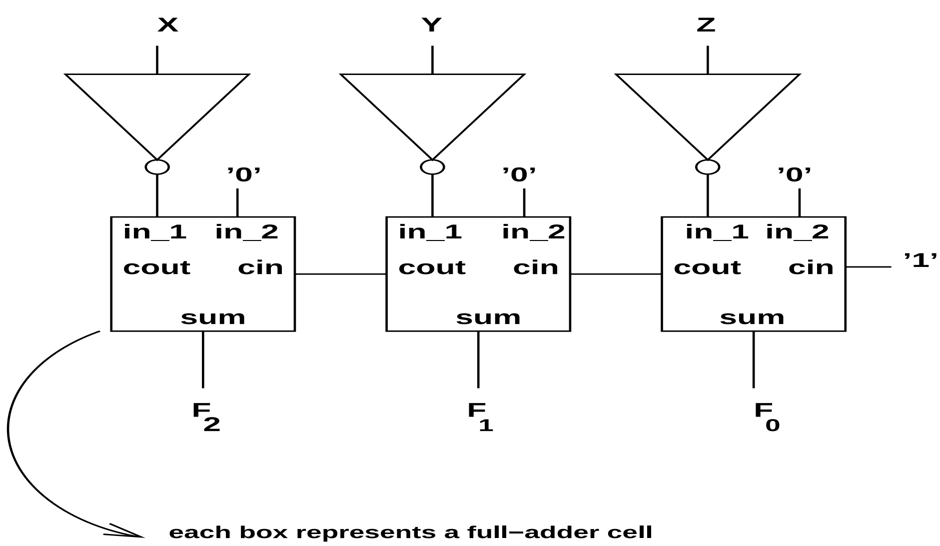

Figure 1: Hardware which implements the functionality of the Table 1.

<u>Question 4:</u> [20 points] (Sequential Logic: flip-flops and timing diagrams)

Consider the circuit illustrated in Figure 2 consisting of two positive edge-triggered D-type flip-flops with selective load capability, an input bus, an output bus and two tri-state buffers.

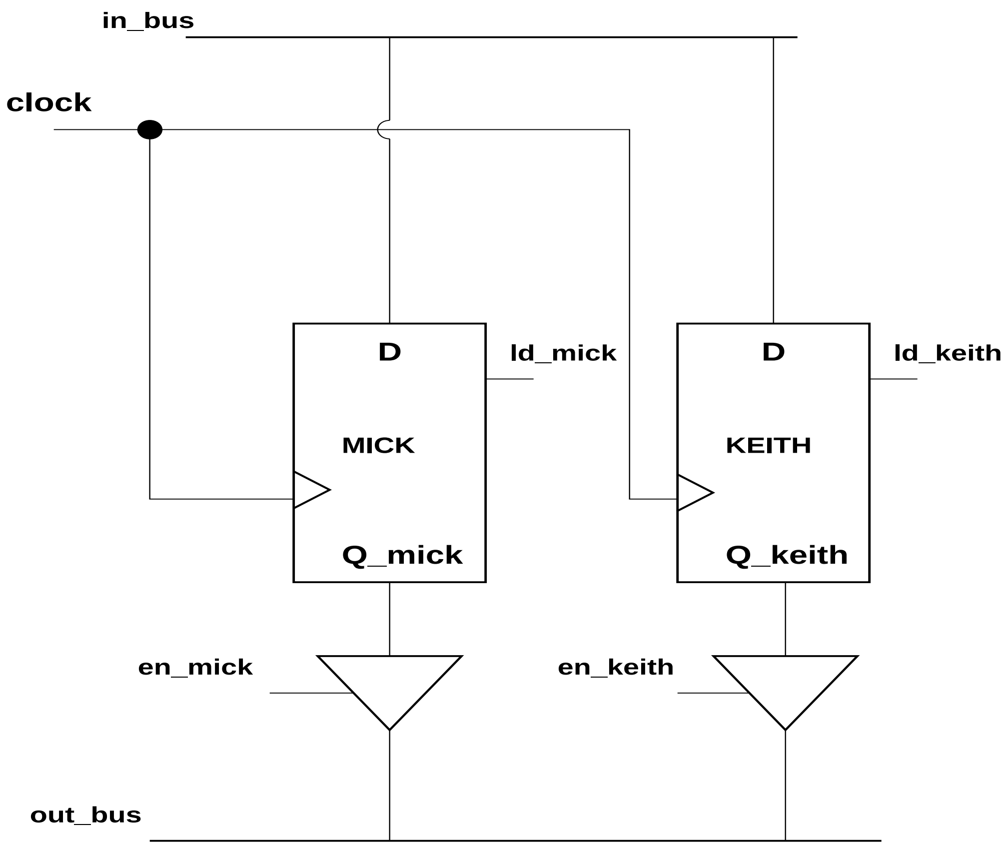

Figure 2: A simple bus circuit.

Complete the timing diagram which is included at the end of this exam. **Write your name and ID number in the space provided and remove the last page of this exam and insert it into your answer booklets.** Indicate in the provided timing diagram the behaviour of the two flop outputs (q_mick and q_keith) as well as the behaviour of out_bus between the indicated "start" and "end" times.

Use the symbol "Z" to denote when the bus is in the high impedance (tri-state value) state. In the provided timing diagram, it is assumed that the **initial value** of q_mick is logic '0' and that the **initial value** of q_keith is logic '1'. Note also that the timing diagram **intentionally** contains a fatal design error. You are to explain in words (in your answer booklet) the nature of this design error and to indicate (on the provided timing diagram) when this error occurs. HINT: use the word **"FIRE"** to indicate the state of the out_bus at that point in time when the design error occurs.

<u>Question 5:</u> [20 points] (Intel Assembly Language Programming)

Consider the following Intel assembly language listing file:

```
 1                                       ; Ted Obuchowicz
 2                                       ; Thu Feb 23 21:00:27 EST 2012
 3                                       ; PROGRAM 1
 4                                       section .data
 5
 6 00000000 0102030405                   number: db 1,2,3,4,5
 7 00000005 00                           sum:    db  0
 8
 9
10                                       section .text
11                                               global _start
12
13                                       _start:
14 00000000 B905000000                   keith:  mov ecx, 5
15 00000005 B800000000                   ron:    mov eax, 0
16 0000000A BB[00000000]                         mov ebx, number
17 0000000F 0203                         again:  add al,  [ebx]
18 00000011 81C301000000                         add ebx,1
19 00000017 81E901000000                         sub ecx,1
20 0000001D 75F0                                 jnz again
21 0000001F A2[05000000]                         mov [sum], al
22
23 00000024 B801000000                           mov eax,1
24 00000029 BB00000000                           mov ebx,0
25 0000002E CD80                                 int 80h
```

(a) When this program is run, what value will be stored in memory location `sum`? Give your answer in hexadecimal notation.

(b) What is the significance of the value F0 in line 20 (indicated in bold font) ?

Now, consider the following Intel assembly language listing file:

```
 1                                       ; Ted Obuchowicz
 2                                       ; Thu Feb 23 21:06:24 EST 2012
 3                                       ; PROGRAM 2
 4                                       section .data
 5
 6 00000000 0102030405                   number: db 1,2,3,4,5
 7 00000005 00                           sum:    db  0
 8
 9
10                                       section .text
11                                               global _start
12
13                                       _start:
14 00000000 B905000000                   keith:  mov ecx, 5
15 00000005 B800000000                   ron:    mov eax, 0
```

```
16 0000000A BB[00000000]                    mov ebx, number
17 0000000F 81C304000000                    add ebx, 4
18 00000015 0203                    again:  add al,  [ebx]
19 00000017 4B                              dec ebx  ;decrement ebx
20 00000018 49                              dec ecx  ;decrement ecx
21 00000019 75FA                            jnz again
22 0000001B A2[05000000]                    mov [sum], al
23
24 00000020 B801000000                      mov eax,1
25 00000025 BB00000000                      mov ebx,0
26 0000002A CD80                            int 80h
```

(c) Does this program perform the same task as the first program (PROGRAM 1) ? What value will be stored by PROGRAM 2 in memory location `sum`? Are there any differences between the two programs?

(d) What is the significance of the value FA in line 21 of PROGRAM 2 ?

(e) If these programs are stored in a memory which is byte-organized (i.e a memory in which every memory read access results in one byte of information being read from the memory) which program will execute faster? Justify your answer.

WRITE YOUR NAME AND ID IN THE SPACE PROVIDED.
REMOVE THIS PAGE AND INSERT IT INTO YOUR ANSWER BOOKLET.

NAME:________________________
ID: ___________________________

TIMING DIAGRAM FOR QUESTION 4.