UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

UCSC

BACHELOR OF SCIENCE IN INFORMATION SYSTEMS

Academic Year 2013/2014 – First Year Examination – Semester I – 2014

IS 1002 – Computer Systems

TWO (2) HOURS

To be completed by the candidate

Examination Index No: ______________

Important Instructions to candidates:

1. The medium of instruction and questions is **English**.
2. If a page or a part of this question paper is not printed, please inform the supervisor immediately.
3. Note that questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
4. Write your index number in each and every page of the question paper.
5. This paper has **4** questions and **14** pages.
6. Answer **ALL** questions. All questions carry equal marks (**25** marks).
7. Any electronic device capable of storing and retrieving text including electronic dictionaries and mobile phones are not allowed.
8. Calculators are allowed.

For Examiner's use only	
Question No	**Marks**
1	
2	
3	
4	
Total	

1. (a) Convert the number **+47.84375** from decimal notation into **16-bit floating point representation.** Assume 8 bits for the mantissa and 7 bits for the exponent.

[6 Marks]

$+47.84375 = 101111.11011$

$= 1.0111111011 \times 2^{+5}$

$K = 2^{N-1} - 1 = 2^{7-1} - 1 = 63$

$\therefore$ Exponent $= 63 + 5 = 68$

$\therefore$ Answer

0	1	0	0	0	1	0	0	0	1	1	1	1	1	1	0

Exponent   Mantissa

(b) What is the loss of accuracy (round-off-error) in (a) above?

[7 Marks]

Round-off-error $= 101111.11011 - 101111.11000$

0.00011

$= 0.09375$

(c) Convert the number **+129.859375** from decimal notation into IEEE standard 32-bit floating point representation.

[6 Marks]

$+129.859375 = +1000\ 0001 . 110\ 111$

$= 1 \cdot 000\ 0001\ 1.1\ 0\ 111 \times 2^{+7}$

$K = 2^{N-1} - 1 = 127$

$\therefore$ Exponent $= 127 + 7 = 134$

$\therefore$ Answer

$$0\ \underbrace{\boxed{1}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{0}}_{\text{Exponent}}\ 000\ 000\ 1110\ 111\ 000 \ldots$$

(d) What is the equivalent decimal number to the IEEE standard 32-bit floating point representation of **1 10001010 11111111111111000000000**

[6 Marks]

Exponent $= 138 - 127 = 11$

$\therefore$ Answer $= -1 \cdot 1111111111111111 \times 2^{11}$

$= -111111111111.111$

$= -4095.875$

2. **(a)** Simplify the following Boolean function.

$$F(x, y) = (\overline{x \cdot y}).(\overline{x} + y).(y + \overline{y})$$

[5 Marks]

$F(x, y) = (\overline{x \cdot y}) \cdot (\overline{x} + y) \cdot (y + \overline{y})$

$= (\overline{x \cdot y}) \cdot (\overline{x} + y) \cdot 1$

$= (\overline{x} + \overline{y}) \cdot (\overline{x} + y)$

$= \overline{x} \cdot \overline{x} + \overline{x} \cdot y + \overline{x} \cdot \overline{y} + y \cdot \overline{y}$

$= \overline{x} + \overline{x} \cdot y + \overline{x} \cdot \overline{y}$

$= \overline{x}(1 + y + \overline{y})$

$= \overline{x}$

(b) Suppose,

$$f(ABCD) = (AB\overline{C}\overline{D}) + (AB\overline{C}D) + (ABCD) + (ABC\overline{D}) + (A\overline{B}CD) + (A\overline{B}C\overline{D}) + (\overline{A}BCD)$$

where A, B, C and D are Boolean variables.

Simplify the above Boolean expression $f(ABCD)$. Clearly write down the steps in the simplification.

[5 Marks]

$$F(A, B, C, D) = AB\overline{C}\overline{D} + AB\overline{C}D + ABCD + ABC\overline{D} + A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}BCD$$

CD \ AB	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

$$F(A, B, C, D) = A \cdot B + A \cdot C + B \cdot C \cdot D$$

(c) A fire alarm signal system consist of three smoke detecting sensors and connected to the living room (S1), master bed room (S2) and the children's play room (S3) respectively. The fire alarm will be activated if at least two of the three smoke detectors are activated.

Assume that logic one for an activated sensor and output should be logic one to activate the fire alarm signal system.

(i) Provide the truth table for the above fire alarm system.

[5 Marks]

S_1	S_2	S_3	$F(S_1, S_2, S_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(ii) Get the most simplified **Sum of Products** (SOP) and **Product of Sums** (POS) expressions that represents the logic function of the above fire alarm system in **(b)(i).**

[5 Marks]

S_3 \ S_1S_2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$F(S_1, S_2, S_3) = S_1S_2 + S_2S_3 + S_1S_3$$

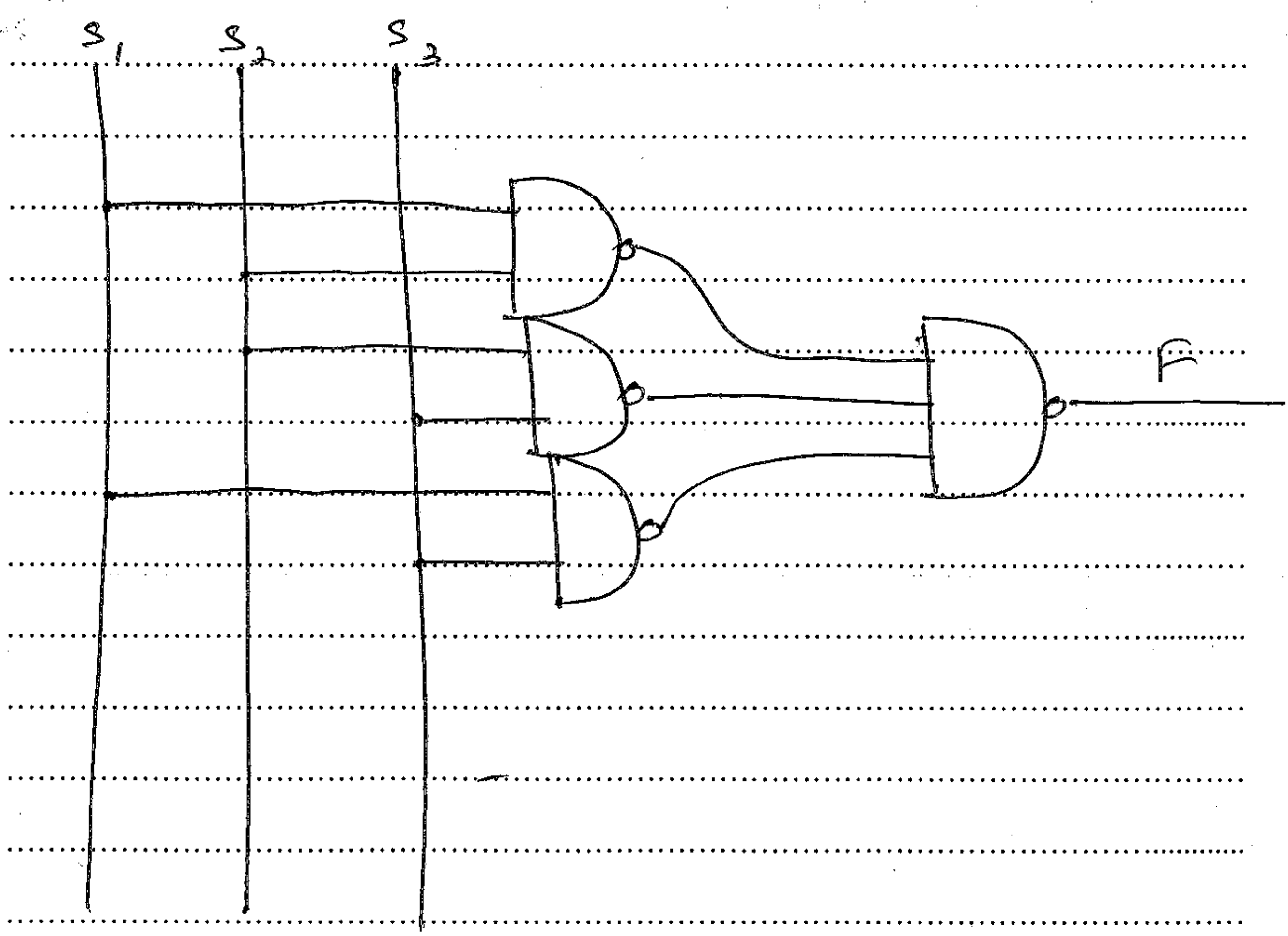
$$F(S_1, S_2, S_3) = (S_1 + S_2) \cdot (S_1 + S_3) \cdot (S_2 + S_3)$$

(iii) Design the logic circuit for the above **SOP** expression in (b)(ii) using only **NAND** gates. Clearly indicate the method and steps.

[5 Marks]

$$F(s_1, s_2, s_3) = \overline{\overline{S_1 S_2 + S_2 S_3 + S_1 S_3}}$$

$$= \overline{\left(\overline{S_1 \cdot S_2}\right) \cdot \left(\overline{S_2 \cdot S_3}\right) \cdot \left(\overline{S_1 \cdot S_3}\right)}$$



3. (a) Consider a machine with instruction format of the form **opcode R M** where **R** is a register address and **M** is a memory address. Instructions are 16 bits long and one of the instruction formats provides 4 bits for the op-code, 4 bits for the register and other 8 bits for the memory address of the operand. Assume that the word size of this machine is 8 bits (byte addressable).

Some of the op-codes of the above (a) processor is given below:

0001 – L R, A	**LOAD** the register **R** with the content of memory cell **A**
0010 – LI R, I	**LOAD** the register **R** with the value **I**
0011 – ST R, A	**STORE** the content of the register **R** to the memory cell whose address is **A**
0101 – ADD R0, R1, R2	**ADD** the numbers in registers **R1** and **R2** and place the result in register **R0**
1001 - XOR R0, R1, R2	**XOR** the bit patterns in **R1** and **R2** and place the result in **R0**
1000 - AND R0, R1, R2	**AND** the bit patterns in **R1** and **R2** and place the result in **R0**
1110 – JMP R, A	**JUMP t**o the instruction located in the memory cell **A** if the bit pattern in **R** is equal to the one in **R0**
1111 - HALT	**HALT** the execution

Write down the machine code instructions sequence to execute the following program statements.

```
void main( )
{
        int Count = 10 ;
        int Sum =  0 ;
        do
        {
                Sum =  Sum + Count  ;
                Count = Count  - 1 ;
        }
        while ( Count  >  0 ) ;
}
```

Assume that **Count** and **Sum** are variables refer the memory addresses **80**, **81** and the initial program counter (PC) as hexadecimal **30**.

[20 Marks]

```
30  Li   R1  0A      } Count = 10
32  ST   R1  80      }

34  Li   R2  00      } Sum = 0
36  ST   R2  81      }
                       do {
38  ADD  R2  R1  R2  } Sum = Sum + Count

3A  Li   R3  FF      } Count = Count - 1
3C  ADD  R1  R1  R3  }

3E  Li   R0  00      }
40  JMP  R1  [38]    } while (Count > 0).
42  HALT             }
```

(b) Suppose a machine with instruction format of the form ***opcode A B R*** where ***A*** and ***B*** are main memory addresses and ***R*** is a register. If there are 20 registers in the machine, how many addresses are possible to have in the machine's main memory? An instruction occupies 32 bits and the instruction set has 32 opcodes.

[5 Marks]

32 bits

opcode	Memory Address	Memory Addr.	Register.
5 bits			5 bits

$\therefore$ No. of bits for 2 Memory Addresses $= 32 - 10$
$= 22$ bits

$\therefore$ No. of bits for a single Memory Address $= 22/2$
$= 11$ bits

$\therefore$ No. of Memory Address in Main Memory $= 2^{11}$
$= 2048$

4. (a) A computer has 16 pages of virtual address space, but only 4 page frames. Initially, memory is empty. A program references the virtual pages in the following order: **0, 1, 2, 2, 1, 3, 4, 5, 0, 6, 7, 8, 8, 7, 1, 2, 7, 3, 4, 3**.

For each memory reference, write the virtual page stored in each page frame under the **Most Recently Used (MRU)** and **First-In-First-Out (FIFO)** page replacement policies (clearly show the page frames in memory for each reference).

[10 Marks]

MRU - Most Recently Used

0	1	2	2	1	3	4	5	0	6
0	0	0	0	0	0	0	0	0	6
	1	1	1	1	1	1	1	1	1
		2	2	2	2	2	2	2	2
					3	4	5	5	5

7	8	8	7	1	2	7	3	4	3
7	8	8	7	7	7	7	3	4	3
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
5	5	5	5	5	5	5	5	5	5

FIFO – First – In – First – Out

0	1	2	2	1	3	4	5	0	6
0	0	0	0	0	0	4	4	4	4
	1	1	1	1	1	1	5	5	5
		2	2	2	2	2	2	0	0
					3	3	3	3	6

7	8	8	7	1	2	7	3	4	3
7	7	7	7	7	7	7	3	3	3
5	8	8	8	8	8	8	8	8	8
0	0	0	0	1	1	1	1	4	4
6	6	6	6	6	2	2	2	2	2

(b) Suppose you have **256MB** of virtual memory, **32MB** of physical memory, and the page size of **64KB** (2^{16}) bytes.

(i). How many pages are there in virtual memory and in physical memory?

[5 Marks]

No. of Pages in VM = ~~64KB~~ 256 MB / 64KB

= 2^{12} pages

No. of frams in PM = 32MB / 64KB

= 2^{9} page frames.

(ii). Assume that the relevant portion of your Page Table is as follows:

Page	Frame #	Valid Bit
0	2	1
1	3	0
4	8	0
8	0	1
11	6	1
....	....	....
15	7	1
....	....	....
17	4	1
....	....	....
21	13	1
....	....	....

Calculate the referenced address in main memory for the following virtual addresses: **0x0011D910** and **0x00157808**.

[10 Marks]

$0x\,\boxed{0011}\boxed{D910}$

page No offset

Page No 17 is in frame No 4.

∴ Physical Address

0x 0004D910

$0x\,\boxed{0015}\boxed{7808}$

Page No. Offset

Page No 21 is in frame No 13.

∴ Physical Address

0x 000D7808

****