2021 EXAMINATIONS

Lancaster University

Part I

COMPUTING AND COMMUNICATIONS – On-line Assessment

Available Time **[23 Hours]**

Recommended Completion Time **[3 Hours]**

SCC.150 Digital Systems

*Candidates are asked to answer **THREE** questions from **FOUR**; each question is worth a total of 25 marks.*

Question 1

1.a By using your knowledge of binary, hexadecimal and decimal numeral systems, answer the following:

i. Convert $(0.339)_{10}$ to binary. (1 mark)

ii. Convert the binary number from your answer for question 1.a.i back to decimal. (1 mark)

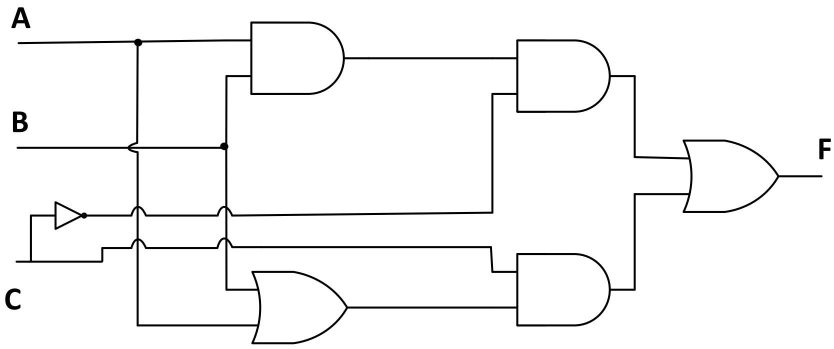
iii. What is the amount of accuracy loss (base-10) in question 1.a.ii? (1 mark)

iv. Assuming 8-bit two's complement numbers, write down the result of the subtraction $(11011000)_2 - (10011111)_2$. Express your answer in 8-bit 2's complement binary. (2 marks)

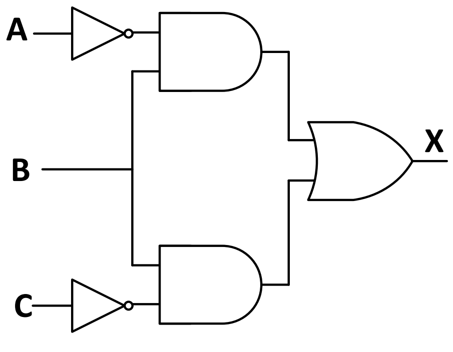
[**5 marks**]

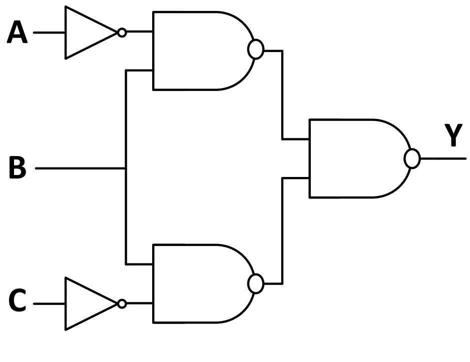
1.b

i. The logical circuit below implements a logical expression. Can you express the logical operation of the circuit as a sum of products? (2 marks)
Hint*: You might need to simplify the equation to express it as sum-of-*products



ii. The logical circuits below represent two function X and Y. Express the logical circuits as Boolean expressions. Are the two expressions equivalent? Justify your answer and show steps clearly. (6 marks)





iii. Briefly describe how sequential logic differs from combinational logic. (2 marks)

[**10 marks**]

Question 1 continues on next page…

1.c

i. Use Boolean algebra identities to solve the following equation. Show clearly all your steps. (3 marks)

G = BC (C' + D) + CD + C + A'

ii. The truth table of a logic circuit that has two inputs (A,B) and two outputs (Y.Z) is shown below. Derive the Boolean expressions of the logic circuit outputs Y and Z and draw the logic circuit diagram for the logic circuit. (3 marks)

INPUT		OUTPUT	
A	B	Y	Z
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

iii. Consider the following Boolean expression:

F = ABCD + A'BCD + ABC'D + A'BC'D + ABC'D' + A'BC'D' + A'B'C'D' + AB'C'D' + AB'CD'

Minimise it using a Karnaugh map and demonstrate clearly the Karnaugh map table, the identified groups extracted from the table as well as the simplified version of the Boolean expression using that approach. (4 marks)

[**10 marks**]

Total 25 marks

Question 2

2.a The MIPS ISA provide the syscall instruction.

i. Describe what is a syscall. Discuss an example use of the instruction. (2 marks)

ii. Write a MIPS program that prints a string on the standard output. You program should assume that the string is in the memory address 0x1001000 and is NULL-terminated. (3 marks)

[**5 marks**]

2.b Below you are given a 32-bit hex value that encodes a MIPS instruction.

0x363000ff

i. Convert the instruction to binary. (2 marks)

ii. Is this an R-type, I-type or J-type instruction? Explain your answer. (3 marks)

iii. Depending your answer in question ii, compute the value of the opcode, funct, rs, rt, rd, shamt and imm fields, where applicable. (5 marks)

You can find a detailed list of opcode and funct values for all significant MIPS instructions and a register mapping table at the end of this question.

[**10 marks**]

2.c You are given the following MIPS program:

```
addi $t1, $zero, 0x10010000
addi $t2, $zero, 10
sw $t2, 12($t1)
```

This program compiles and runs as expected on the MARS MIPS emulator but fails to compile when *deselecting* the option: "Permit extended (pseudo) instructions and formats" in the emulator configuration menu. Propose a solution to this compilation error and explain your code fix.

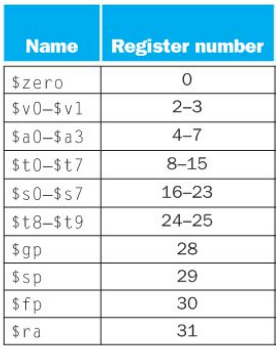
[**5 marks**]

2.d Registers $t1 and $t2 contain each a 16-bit integer. Write a MIPS program that will load the value of register $t1 in the upper 16-bit of register $t0 and register $t2 in the lower 16-bit of register $t0.

[**5 marks**]

Question 2 continues on next page…

Question 2 continued…



Name	Register number
$zero	0
$v0–$v1	2–3
$a0–$a3	4–7
$t0–$t7	8–15
$s0–$s7	16–23
$t8–$t9	24–25
$gp	28
$sp	29
$fp	30
$ra	31

Mnemonic	Meaning	Type	Opcode	Funct
Add	Add	R	0x00	0x20
Addi	Add Immediate	I	0x08	NA
addiu	Add Unsigned Immediate	I	0x09	NA
addu	Add Unsigned	R	0x00	0x21
and	Bitwise AND	R	0x00	0x24
andi	Bitwise AND Immediate	I	0x0C	NA
beq	Branch if Equal	I	0x04	NA
blez	Branch if Less Than or Equal to Zero	I	0x06	NA
bne	Branch if Not Equal	I	0x05	NA
bgtz	Branch on Greater Than Zero	I	0x07	NA
div	Divide	R	0x00	0x1A
divu	Unsigned Divide	R	0x00	0x1B
j	Jump to Address	J	0x02	NA
jal	Jump and Link	J	0x03	NA
jalr	Jump and Link Register	J	0x00	0x09
jr	Jump to Address in Register	R	0x00	0x08
lb	Load Byte	I	0x20	NA
lbu	Load Byte Unsigned	I	0x24	NA
lhu	Load Halfword Unsigned	I	0x25	NA
lui	Load Upper Immediate	I	0x0F	NA
lw	Load Word	I	0x23	NA
mfhi	Move from HI Register	R	0x00	0x10
mthi	Move to HI Register	R	0x00	0x11
mflo	Move from LO Register	R	0x00	0x12
mtlo	Move to LO Register	R	0x00	0x13

Mnemonic	Meaning	Type	Opcode	Funct
mfc0	Move from Coprocessor 0	R	0x10	NA
mult	Multiply	R	0x00	0x18
multu	Unsigned Multiply	R	0x00	0x19
nor	Bitwise NOR (NOT-OR)	R	0x00	0x27
xor	Bitwise XOR (Exclusive-OR)	R	0x00	0x26
or	Bitwise OR	R	0x00	0x25
ori	Bitwise OR Immediate	I	0x0D	NA
sb	Store Byte	I	0x28	NA
sh	Store Halfword	I	0x29	NA
slt	Set to 1 if Less Than	R	0x00	0x2A
slti	Set to 1 if Less Than Immediate	I	0x0A	NA
sltiu	Set to 1 if Less Than Unsigned Immediate	I	0x0B	NA
sltu	Set to 1 if Less Than Unsigned	R	0x00	0x2B
sll	Logical Shift Left	R	0x00	0x00
srl	Logical Shift Right (0-extended)	R	0x00	0x02
sra	Arithmetic Shift Right (sign-extended)	R	0x00	0x03
sub	Subtract	R	0x00	0x22
subu	Unsigned Subtract	R	0x00	0x23
sw	Store Word	I	0x2B	NA

Total 25 marks

Question 3

3.a A function pointer is a pointer type that points to executable code in memory.

i. Define a function pointer variable that points to a function that accepts an int and a float argument and the function returns a float value. (3 marks)

ii. Assume your code contains the following function definition. Write a line of C code that set the variable in question 3.a.i to point to the following function. (1 mark)

```
float sign (int sign, float val) {
     if (sign) {
          return -val;
     else
          return val;
}
```

iii. Write a line of C code to invoke the function using the variable defined in question 3.a.i, by passing as arguments an int variable x and a float variable y. (1 mark)

[**5 marks**]

3.b Unions and structs are two common data types in C. Describe how the two data types differ with respect to the way they represent their data in memory. Below, we provide the definitions for two structs (s1 and s2) and one union (u1). How large (in bytes) are the s1, s2 and u1 on a 32bit processor architecture? Explain in detail how you calculate the size.

```
 struct s1 {
     char *a;
     char b;
     int   c;
 };

 struct s2 {
    char a[16];
    char   b;
    int c;
}__attribute__((packed));

union u1 {
    char a[16];
    char   b;
    int c;
 };
```

[**6 marks**]

Question 3 continues on next page…

Question 3 continued…

3.c Implement a string manipulation function called left_trim. You function should accept a string input src parameter and it must return a pointer to a copy of the original string without any whitespaces before the first non-whitespace character of the original string. Your function can manipulate the original input string data, or you can use the malloc/strdup functions to allocate memory or create a string replica.

```
char *left_trim(char *src) {
//Insert code here.
}
```

To better describe the required functionality for the function we provide below a sample main method and the expected output.

```
int main() {
     printf(“%s\n”, left_trim(“nonce”));
     printf(“%s\n”, left_trim(“   nonce”));
     printf(“%s\n”, left_trim(“no nonce”));
}
```

Output:

```
nonce
nonce
no nonce
```

[**9 marks**]

3.d A program is used to monitor 8 doors. The state of the doors (open/closed) is represented by the variable doors and each bit corresponds to a different door. The management program invokes the following function to realize a specific standard operation every time the state of a light changes.

```
int process_doors(int doors){

if ((doors & (1 << 3)) &&
   ((doors & (1 << 2)) == 0)) {
      doors = doors | (1<<4);
      doors = doors ^ (1 << 5);
   } else {
      doors = doors & ~(1 << 3);
      doors = doors & ~(1 << 2);
      doors = doors | (1 << 1);
   }
   return doors;
}
```

Question 3 continues on next page…

i. Describe the door test performed in lines 3 and 4? Explain your answer. (2 marks)

ii. What is the result in the door state by the operations performed in lines 5 and 6? Explain your answer. (2 marks)

iii. What is the result of the operation in line 8? Explain your answer (1 mark)

[**5 marks**]

Total 25 marks

Question 4

4.a

i. Specify the decimal number -128 in an 8-bit binary sign and magnitude representation. (2 marks)

ii. Convert the decimal number -14.25_{10} to the 32-bit IEEE 754 floating point and express your answer in hexadecimal. (Reminder: The 32 bits are used as follows: Bit 1: sign of mantissa, bits 2-9: 8-bits of exponent in excess 127, bits 10-32: 23 bits for magnitude of mantissa.) (4 marks)

iii. Describe what pipelining is and state the three different types of pipeline hazards. (2 marks)

[8 marks]

4.b The preprocessor is an essential step in the compilation of C code into a binary executable.

i. Describe the four primary capabilities of the C preprocessor (2 marks).

ii. The two following programs are processed by the C preprocessor. What is the output C code from the preprocessor when processing the two source file? Explain your answer. (7 marks)

```
test1.c:
#define LINUX 0

int check_linux_os() {
   #if LINUX
   return 1;
   #else
   return 0;
   #endif // LINUX
}
```

```
test2.c:
#define LINUX 0

int check_linux_os() {
   #ifdef LINUX
   return 1;
   #else
   return 0;
   #endif // LINUX
}
```

[9 marks]

4.c Your program is servicing an interrupt in the kernel text area. The state of the registers of co-processor 0 is the following:

```
$8  (vaddr)     = 0x10000005
$12 (status)    = 0x0000ff13
$13 (cause)     = 0x00000014
$14 (epc)       = 0x00400010
```

i. Can you identify the cause of this interrupt? Explain your answer based on the content of the co-processor registers. (6 marks)

Question 4 continues on next page…

ii. What is the address of the instruction in the text area that raised this interrupt? Explain your answer (2 marks)

[**8 marks**]

Total 25 marks

--- End of Paper ---