CS2100 AY2023-24, Sem2
Midterm Questions: (40 marks)

Q1: (1 mark) MCQ
The unsigned number 2132 in base 4 is equivalent to 185 in which base system?

A. Base-7
B. Base-8
C. Base-9
D. Base-11
E. None of the above

Q2: (2 marks, each 0.5 mark) FITB
Convert the decimal number 85.125 into its IEEE 754 single-precision floating-point representation.
Note: For parts (a) to (c), you will enter only the binary bits and for (d), you will answer in hexadecimal format.

(a) Binary representation of the decimal number 85.125 is:_____
(b) Exponent: _____
(c) Mantissa: _____
(d) IEEE 754 single-precision binary representation of 85.125: _____

Q3: (2 marks, each 0.5 mars) FITB
Convert 01000010010101110000000000000000 in the IEEE 754 single-precision floating-point representation to its decimal equivalent. Answer the following MCQs:

a. The sign bit "0" indicates that the number is _____.
b. After converting the exponent part "10000100" to decimal and subtracting the bias, the actual exponent is _____ in decimal.
c. Including the implicit leading one, the full mantissa (fraction part) in binary format is _____.
d. The decimal equivalent of the IEEE 754 representation "0 10000100 10101110000000000000000" is _____ in decimal.

Q4: (1 mark) MCQ
Consider the MIPS code below:

```
main: j loop
loop: addi $t0, $t0, 1
      j main
```

Assuming that the first line of the program is at address 0x00400000, what is the encoded instruction for "j loop"?

A. 0x08000001
B. 0x08000004
C. 0x08100001
D. 0x08100004
E. None of the above.

Q5: (1 mark) MCQ
Consider the MIPS code below:

```
main: j loop
loop: addi $t0, $t0, 1
      j main
```

Assuming **$t0** is initialized to **0**, after how many iterations of the loop will cause an overflow in register **$t0**?

A. Overflow will never occur because the MIPS architecture handles integer overflow gracefully.
B. After 2^{31} iterations, because **$t0** will overflow when it attempts to increment from $2^{31}-1$ to 2^{31}.
C. After 2^{32} iterations, because **$t0** can hold values from 0 to 2^{32} before overflowing.
D. Overflow occurs immediately as **$t0** cannot be incremented.

Q6: (2 marks) FITB

For this question, assume size of **int** is 4 bytes, size of pointer **void*** is 4 bytes, size of **char** is 1 byte.

Program-1

```c
#include <stdio.h>

struct A {
    struct A* ptr;
    char* str;
    int num;
};

int main() {
    //create example struct A
    struct A example = {NULL, "example", 7};
    //print address of example
    printf("%p\n", &example);

    return 0;
}

```

After running the above C program on a 32-bit machine, let's say the output by the printf function is 0x7FFCC8B0

What is the address of the example.num variable?

If it is not possible to tell, answer "Not possible".

Q7: (1 mark) MCQ

For this question, assume size of **int** is 4 bytes, size of pointer **void*** is 4 bytes, size of **char** is 1 byte.

Program 2 uses the same **struct A** defined in program 1. Here, we introduce functions **changeStr** and **changeNum**.

Program-2

```
 9  int main() {
10      // create example struct A
11      struct A example = {NULL, "example", 7};
12      // changeNum & changeStr
13      changeNum(example, 888);
14      changeStr(example, "changed");
15
16      return 0;
17  }
18
19  void changeStr(struct A srctA, char* targetStr) {
20      srctA.str = targetStr;
21  }
22
23  void changeNum(struct A srctA, int targetNum) {
24      srctA.num = targetNum;
25  }
26
```

What is the final value of example after running the above C Program?

A. {NULL, “example”, 7}
B. {NULL, “example”, 888}
C. {NULL, “changed”, 7}
D. {NULL, “changed”, 888}
E. None of the above

Q8: (1 mark) MCQ

For this question, assume size of **`int`** is 4 bytes, size of pointer **`void*`** is 4 bytes, size of **`char`** is 1 byte.

Here, we investigate a mysterious function in the below program below:

Program 3:

```
1  #include <stdio.h>
2
3  struct A {
4      struct A* ptr;
5      char* str;
6      int num;
7  };
8
9  int main() {
10     struct A arrA[3] = {{&arrA[1], "string", 6},
11                         {&arrA[2], "AAA", 3},
12                         {arrA, "hello", 5}};
13     mysteriousFn(arrA, 3);
14     //HERE
15     return 0;
16 }
17
18 void mysteriousFn(struct A arrA[], int size) {
19     for(int i=0; i<size; i++) {
20         *(arrA[i].ptr) = arrA[i];
21     }
22 }
23
```

At the point marked 'HERE' in main(), immediately following the call to **mysteriousFn** with **arrA**, what are the values of the **str** and **num** members for each struct A in **arrA**? Disregard the **ptr** member values of struct A, which are denoted by '_'. Choose the option that accurately reflects the state of **arrA**:

A. {{_, "string", 6}, {_, "string", 6}, {_, "hello", 5}}
B. {{_, "AAA", 3}, {_, "hello", 5}, {_, "string", 6}}
C. {{_, "hello", 5}, {_, "string", 6}, {_, "AAA", 3}}
D. {{_, "hello", 5}, {_, "string", 6}, {_, "hello", 5}}
E. {{_, "string", 6}, {_, "string", 6}, {_, "string", 6}}

Q9: (1 mark) MCQ

Consider the subtraction: $(1011101)_{1s}$ - $(1001100)_{1s}$.
What is the answer in 1's complement?

A. $(0010000)_{1s}$
B. $(0010001)_{1s}$
C. $(1101110)_{1s}$
D. $(1101111)_{1s}$
E. None of the above.

Q10: (1 mark) FITB

Consider the decimal value **31.434**. What is its value in quaternary (base 4), correct to 3 quaternary digits?

Q11: (3x1 = 3 marks) FITB

Consider the following MIPS program and assume that the program is correct and can run to completion. Answer the following questions:

```
      xor  $1, $1, $1          # instr1
      addi $1, $1, 0x0A        # instr2
      addi $2, $1, 0x01        # instr3
Loop: andi $3, $2, 0x01        # instr4
      beq  $3, $zero, Exit     # instr5
      srl  $2, $2, 0x01        # instr6
      j    Loop                # instr7
Exit:
```

(a) What is the value of $1 after executing the first instruction?
(b) How many times is the code segment from instr4 to instr6 (inclusive) executed?
(c) What is the value of register $2 after executing the above code? Write your answer in base 10 without any leading zero.

Q12: (1 mark) MCQ

In a typical MIPS processor's Datapath, which includes elements such as a fetch unit, decoder, register file, an Arithmetic Logic Unit (ALU), and control units. What is TRUE with respect to the function of the ALU?

A. The ALU is responsible for determining the processor's branch decisions and directing the flow of data within the datapath.
B. The ALU performs arithmetic and logical operations only on the data provided by the processor's register file.
C. The ALU serves as the main storage area for instructions that are queued for execution.
D. The ALU accelerates the connection between the processor and external devices, managing data transfers to and from memory.
E. None of the above.

Q13: (1 mark) MCQ

Given an "ADDI" instruction in MIPS, which requires adding an immediate value to a register's content, select the sequence that correctly outlines the steps the datapath performs to execute this instruction:

A. The instruction is fetched from memory and placed into the Instruction Register (IR). The specified register's content is read and sent to one input of the ALU. The immediate value is decoded from the instruction and supplied to the other input of the ALU. The ALU executes the addition, and the resulting sum is written back into the designated register in the register file.

B. The instruction is fetched from memory into the Register File, the immediate operand is moved into the ALU, the register is selected and its content is sent to the ALU, the ALU performs the addition, and the result is written back into the register file.

C. After fetching the instruction from memory, the immediate value is temporarily stored in the Register File. The content of the specified register is then fetched into the ALU where the addition takes place using the stored immediate value.

D. The instruction is retrieved from memory, causing the Program Counter (PC) to increment by 8, pointing to the subsequent instruction. The fetched instruction is decoded, and its operands are processed by the ALU.

Q14: (2x2 = 4 marks) FITB

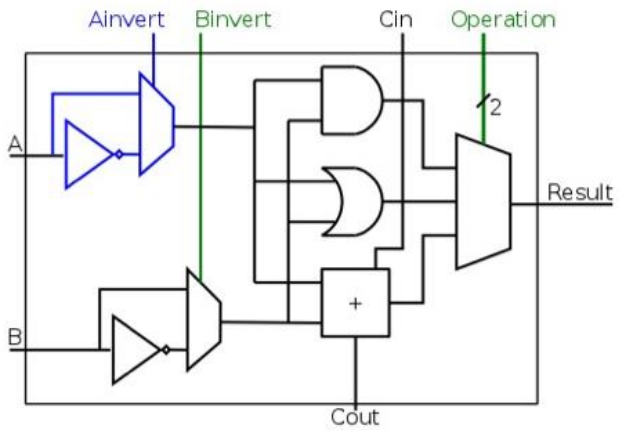
Given MIPS instructions, find out the signals/values for the control and data along the specified paths. All provided values are in decimal. All answers should also be expressed in decimal.

(a) sw $17, 16($8) where $17 holds the value 1 and $8 holds the value 48, what are the control signals and data value at:
   (i) RD2 = ____
   (ii) RD1 = ____
   (iii) ALUSrc = ____
   (iv) MemWrite = ____

(b) 000100 01000 00000 111111111111110 where $8 holds the value of 0 and $16 holds the value 1, what are the control signals and data values at:
   (i) RR2 = ____
   (ii) ALUSrc = ____
   (iii) is0? = ____
   (iv) PCSrc = ____

Q15: (3x2 = 6 marks) FITB

You are implementing a 4-bit ALU based on the single-bit ALU taught during the lecture, which is also shown in the below figure.



(a) To implement the 2's complement subtraction function, what signal should you set for the Ainvert, Binvert and Cin? Consider an example where A=3 and B=2 (both in decimal), and we want to calculate A-B, what are the control signals and data values at:
   (i) Ainvert = ____
   (ii) Binvert = ____
   (iii) Cin = ____
   (iv) The Least Significant Bit of Result = ____

(b) To implement the overflow function (for the 2's complement addition/subtraction), you will need to add an additional logic gate and an additional output called Overflow. For each of the following, determine whether it is T (for True) or F (for False).
   (i) Overflow = Cin XOR Cout at the Most Significant Bit.
   (ii) Overflow or not, the Result output should remain the same.

(c) Suppose the latency of all logic gates and adder (including NOT, AND, OR, and ADDER) is 1ps (pico-second), and latency of all MUXs is 2ps, what is the maximum latency of this 4-bit CPU (excluding the implementation of overflow function in part (b))? More specifically, once the input and the control signals are ready, in the worst case, how many ps is required to obtain the complete 4-bit Result for all operations? Express your result in decimal value only, for example, if the latency is 10ps for signed addition and 11ps for signed subtraction, just answer 11.

Q16. (1 mark) MCQ

MIPS Jump Instruction

For a MIPS jump instruction that is located at the address 0xCFFFFFFC, which of the following addresses can it jump to?

A. 0xBF000000
B. 0xC0008888
C. 0xCFFFFFFF
D. 0xD4448888
E. None of the above.

Q17: (11 marks) FITB

ISA: Suppose that we are making an Instruction Set Architecture (ISA) with THREE classes of instructions:

Class X: contains 2 registers, and 8-bit immediate value (imm).
Class Y: contains a 12-bit jump address (addr).
Class Z: contains 2 registers and a shift amount (shamt).

The ISA contains at least the following instructions:

```
LDH $RD $RA imm
STR $RD $RA imm
JMP addr
JAL addr
SRA $RD $RA shamt
```

The following information about the architecture are given:

- There are 6 registers which can be used.
- Each register holds a 16-bit value.
- The size of each instruction is one word, which is 16 bits for this architecture.
- The address space is 16 bits, and jumps are only done to word-aligned addresses.

a. How many bits are needed to represent the maximum shift amount in this architecture? You are to follow the same design reasoning used in MIPS. (1 mark)

Note: From this point onwards, assume that **`shamt`** is a 3-bit value.

b. Suppose that the encoding space is fully utilized and we are minimizing the number of instructions in the ISA. How many instructions would be under (i) Class X, (ii) Class Y, and (iii) Class Z? (3 marks)

c. Suppose that the encoding space is fully utilized and we are maximizing the number of instructions in the ISA. How many instructions would be under (i) Class X, (ii) Class Y, and (iii) Class Z? (3 marks)

d. Given that JMP is the jump instruction, what would be maximum jump range available for the JMP instruction? (1 mark)

A. 4096 bytes
B. 8192 bytes
C. 16384 bytes
D. 32768 bytes
E. 65536 bytes

e. By changing the number of bits of address in Class Y instructions, maximize the jump range available for the JMP instruction. Make sure that all modifications would still allow the implementation of Class X and class Z instructions in the ISA. Write your answer as a single decimal number without leading zero. (3 marks)

I. What would be the maximum jump range (in number of bytes) available after your modification?

II. What is the maximum number of instructions possible in the ISA after your modification, assuming that the encoding space is fully utilized?

III. What is the minimum number of instructions possible in the ISA after your modification, assuming that the encoding space is fully utilized?