Print Exit

You have completed

CS2100 Midterm Quiz

You scored 6/21

Have not received the confirmation Email? Click here to re-send.

SCROLL DOWN TO VIEW
EDITABLE RESULTS

Preparations

1. Did you open the Google Forms for asking questions, as well as the Google Sheet found in your Gradebook: Google Sheet Details item?

   - No (click this because you don't be) (no ingesting CHF FREY rewards)
   - Yes (click this for ONE FREE MARK).

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

MRQ Questions

Each of these questions may have one or more correct answers. Please choose all correct answers and none of the wrong answers. Two marks per question, 1 is partial credit.

2. We wish to implement a nop operation (NOP) instruction in the MIPS datapath that, when executed, effectively does nothing except waste time (this seemsstrange but is actually very useful in applications that must meet strict minimum timing requirements). Choose all the signal combinations that would correctly implement NOP (note: 0 = Low 1 = High). NOP is implemented as an R-format instruction with a function code of 0x20. (Note: Here we only consider the specifications for the signal values, we do not consider actual implementation yet).

   - RegDst=0, RegWrite=1, ALUSrc=0, MemWrite=0, MemRead=0, MemToReg=1, PCSrc=0
   - RegDst=1, RegWrite=1, ALUSrc=0, MemWrite=0, MemRead=0, MemToReg=0, PCSrc=1/0
   - RegDst=1, RegWrite=0, ALUSrc=0, MemWrite=0, MemRead=0, MemToReg=0, PCSrc=0
   - RegDst=0, RegWrite=1, ALUSrc=1, MemWrite=0, MemRead=0, MemToReg=0, PCSrc=0
   - RegDst=0, RegWrite=1, ALUSrc=1, MemWrite=0, MemRead=0, MemToReg=1, PCSrc=0

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

3. Given the hexadecimal value 0x5F2C, whereas all of the possible interpretations of this value below:

   - The C string "_"
   - The decimal value 24172.
   - A 16-bit 2s complementation of -16375.
   - The ASCII characters "_" and ",".
   - A 16-bit 1s complement negation of -16375.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

4. Allen has written a program to multiply a number stored in register $s0 by 3 and store the result in register $s1. Sadly he lost the assembly code and only has the following fragments of MIPS machine code in hexadecimal. Help Allen to find which instructions are from his program. The ordering of the instructions is not important, and assume that the register $s0 contains the value 5.

   - 0x00008820
   - 0x00108824
   - 0x00008820
   - 0x00008820
   - 0x00108824

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

5. Pick all the statements below that are TRUE about the MIPS datapath:

   - In an 5-stage datapath where each stage may take a different amount of time to complete, a multicycle implementation will give an 5 times speedup over a single cycle implementation for instructions that must pass through only one stage.
   - In a multicycle implementation where every stage takes the same amount of time to complete, the instruction that passes through all the stages will have the same execution time as in a single cycle implementation.
   - A multicycle implementation is better than single cycle since not all instructions need to pass through every stage of the datapath.
   - In a multicycle implementation where every stage may take a different amount of time to complete, the instruction that passes through all the stages will NOT have the same execution time as in a single cycle implementation.
   - A single cycle implementation is better than multicycle implementation since every MIPS instruction takes exactly the same amount of time to go through the datapath.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

6. We wish to implement the pseudo instruction BLE $s1, $s2, target, which branches to "target" if $s1<=$s2. Choose all of the sequences below that implement this behaviour correctly (Note: We are only looking for correct behaviour. May not necessarily be the shortest possible implementation.)

   - slt $t1, $s2, $s1
     beq $t1, Zero, target
   - slt $t1, $s1, $s2
     beq $t1, Zero, target
   - slt $t1, $s1, $s2
     bne $t1, Zero, target
   - sub $t1, $s2, $s1
     slt $t1, $t1, Zero
     bne $t1, Zero, target
   - sub $t1, $s1, $s2
     slt $t1, $t1, Zero
     beq $t1, Zero, target

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

Multiple Choice Questions

The questions in this section have only one correct answer, and each question is worth 1 mark.

7. What does the following C function do?

```
unsigned char mystery2(unsigned char x, unsigned char y) {
    unsigned char a = x;
    unsigned char b = y;
    unsigned char c = 1;
    unsigned char d = 0;

    while(c) {
        if((a & 1) != (b & 1)) {
            d = d + c;
        }

        a = a >> 1;
        b = b >> 1;
        c = c << 1;
    }

    return d;
}
```

   - The function will never exit.
   - Checks whether both x and y are equal.
   - Does a bitwise XOR of x and y.
   - Does a bitwise division of x and y.
   - Shifts x and b both by 1 bit and a right by 1 bit, then adds c to d.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

8. What does the following C function do? Assume that a long is twice the size of an int.

```
unsigned long mystery(unsigned int x, unsigned int y) {
    unsigned long a = 0, b = x;

    while(y) {
                if(y & 0b1) {
                            a = a + b;
                }

                b = b << 1;
                y = y >> 1;
    }
    return a;
}
```

   - Multiplies x and y.
   - Produces an even parity for x and y.
   - Adds x and y.
   - Divides x by y.
   - Shifts b to the right by 1 bit and y to the left by 1 bit and adds them together.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

9. In a 6-bit 1's complement number system, doing -5 - 0 (negative 5 minus zero) will give us:

   - -5
   - 6
   - 0
   - -6
   - -9

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

10. Select the 1's complement of an N-digit decimal number Y we do 10^N - Y. Which ONE of the following statements is TRUE?

   - None of the choices in this question (except this one) are true.
   - In an N-digit 1's complement number system, the leftmost digit has a weight of -10^{N-1} (negative 10^{N})
   - In an N-digit 1's complement number system, the leftmost digit has a weight of 10^{N-1}.
   - In an N-digit 1's complement number system, the leftmost digit has a weight of -10^N (negative 10^N).
   - In an N-digit 1's complement number system, the leftmost digit has a weight of 10^N.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

Multiple Choice Questions - Assembly Language

Guest wrote the following code in MIPS assembly to process some elements of an array whose base address is in $s0. The first element stores the line numbers. Use this program to answer both questions.

1		add $t0, Zero, 0
2		add $s1, Zero, 0
3		add $t1, $s0, 0
4		add $t1, $s0, 4|0
5	sw	sll $s1, $s0, $s1
6		lw $s0, $s0, $s1
7		beq $s0, Zero, x
8		lw $t0, 0($t0)
9		add $t0, $t0, 4
10		add $s0, $s0, 1
11		beq $s0, Zero, y
12		add $s0, $s0, 1
13		j x
14	y:	add $s1, $s1, 1
15		j x
16	x:	

11. What does this code do?

   - It counts the number of array elements with even parity and odd parity and stores the results in $s0 and $s1 respectively.
   - It counts the number of odd and even array elements and stores the results in $s0 and $s1 respectively.
   - It counts the number of array elements divisible by 2 and number of items divisible by 1 and stores the results in $s0 and $s1 respectively.
   - It counts the number of array elements with even parity and odd parity and stores the results in $s1 and $s0 respectively.
   - It counts the number of array elements that are zero and non-zero and stores the results in $s0 and $s1 respectively.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

12. Guest's software license for her assembler has expired, and she decided to "hand-assemble" the code above by consulting the datasheet and translating the assembly code into machine code. However when she ran her code she found that her code always terminates with $s0=0 regardless of the contents of the array, though $s0 is correct. What is the Golden problem?

   - She calculated the offset for the j jump statement at line 16 using the address at line 15 instead of the address at line 16.
   - She calculated the offset for the branch at line 10 based in bytes rather than words.
   - She encoded the and immediate at line 9 in the "and $s1, $t0, 1" instead of "and $s1, $t0, 4".
   - The j statement at line 13 was encoded to jump to line 2 instead of line 5.
   - She calculated the offset for the branch instruction at line 10 based on the instruction at line 10 instead of line 11.

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

Fill In The Blanks (Expanding Opcodes)

In this section we assume a processor with a fixed 16-bit instruction length, 16 registers, and the following 3 instruction classes:

Class0: Two registers.

Class8: One register.

Class1: One register, one 8-bit immediate value.

In all cases we assume that the remaining space for opcodes is fully utilised.

13. Fill in the blanks

   The maximum number of opcodes that can be encoded on this machine is: 1 480

   Enter the correct answer below.

   - 480

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

14. Fill in the blanks

   The minimum number of opcodes that can be encoded on this machine is: 1 (0b16)

   Enter the correct answer below.

   - 2599

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

Fill In The Blanks (Floating Point Numbers)

We have a 16-bit floating point number system with the following format:

1 sign bit for the mantissa	5 bits exponent in 2's complement. The maximum bit	10 bits normalized mantissa, no implicit bit. Indicate bits in sign and exponent with special magnitude representation.	

The leftmost bit and the rightmost 5 bits form a sign-mag/mode machine. Answer the following questions, filling in the **exact** numeric answer each time.

15. Fill in the blanks

   Within the representation range of this number system (i.e. between the most negative and most positive numbers that can be represented), the number 1 120 cannot be represented.

   Enter the correct answer below.

   - 0.1

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.

16. Fill in the blanks

   The most negative number that can be represented in this number system is: -1.4230579688

   Enter the correct answer below.

   - -4.2305795688

   Response Editor: yes
   Points: 1 of 1 (Max: 1) | Auto: 1 of 1 (Max: 1)

   No rubric sub-provided.