Name _____ Student ID _____

CSc 252: Computer Organization

Spring 2025

Test 1: Feb. 6, 2025

Time: 30 min

DO NOT OPEN THE EXAM UNTIL INSTRUCTED TO DO SO

Please read this page and follow the directions before proceeding with the rest of the exam.

- To give all students the same amount of time to do the exam, please DO **NOT** OPEN THIS EXAM UNTIL INSTRUCTED TO DO SO.
- You are not allowed to use any external resources such as cellphones, notes, headphones, watch, neighbors, calculator, etc. If you have not done so yet, turn your cellphone off and place it in your backpack.
- The cellphone cannot be on you during the exam. If your cellphone is in your pocket, it will be considered cheating even if you are not looking at it (same for headphones and watches). We will collect your exam and ask you to leave.
- Place your final answers in the given boxes. You can show your work on any blank spaces.
- We recommend skimming the entire exam before completing any problems.
- Read carefully every question before answering and raise your hand if the question is unclear.
- DO NOT SPEAK TO ANYONE AT YOUR TABLE.

****** Good Luck! ☺ ******

Allowable MIPS Instructions

When writing MIPS assembly, the only instructions that you are allowed to use (so far) are:

- `add, addi, sub`
- `beq, bne, j`
- `slt, slti`
- `and, andi, or, ori, nor, nori, xor, xori`
- `lw, lh, lb, sw, sh, sb`
- `la`
- `syscall`

Name _____ Student ID _____

Write your name and student ID on all the exam pages for one extra credit point.

1. (2 points): Convert each of the powers-of-2 below to its approximate power-of-10 value. (That is, your answer should be something like “32,000,” or “32 thousand”, not “32,768”.)

		Place your final answers here.
2^{10}	≈	1,000
2^{21}	≈	2,000,000
2^{32}	≈	4,000,000,000
2^{53}	≈	8 x 10^{15}

Find the explanation on slide:
https://docs.google.com/presentation/d/1Krya4teGuaYlRy27-Bnc9fCll-Rb1kNSlThSx-imm5Q/edit#slide=id.g32ef4a2facf_0_88

2. (2.5 points): Take the 2’s complement of this 10-bit binary number `1011110101` and write your answer in binary. Show your work.

Find the explanation on slide:
https://docs.google.com/presentation/d/1QjUtlOVqTCrJVA3GI48S2MAY_-DuAQfoKHg30_CIulM/edit#slide=id.g32dff47ed01_0_3632

Place your final answer here. One bit on each box

0	1	0	0	0	0	1	0	1	1

3. (2.5 points): Convert `011101100111010101` from base 2 to 24 bits in hexadecimal.
Hint: you can pad with zeros to fit the 24 bits

Find the explanation on slide:
https://docs.google.com/presentation/d/1QjUtlOVqTCrJVA3GI48S2MAY_-DuAQfoKHg30_CIulM/edit#slide=id.g32dff47ed01_0_2564

Place your final answer here. One nibble on each box

0x

0	3	B	3	D	5

Name _____ Student ID _____

4. (4 points): Give 8-bit signed binary numbers for each of the following:

Write one bit on each box

-1	1	1	1	1	1	1	1	1
Maximum number	0	1	1	1	1	1	1	1
Most negative number	1	0	0	0	0	0	0	0
Zero	0	0	0	0	0	0	0	0

Find the explanation on slide: 02-Numbers Representation in the Computer and Addition

5. (2.5 points): The following five numbers are 16-bit signed integers. Mark your responses by completely darkening the circle in the corresponding column. e.g., ●

	Positive	Negative
`0x0054`	●	○
`0xabcd`	○	●
`0x1fff`	●	○
`0xf000`	○	●
`0x7f0f`	●	○

Find the explanation on slide:
https://docs.google.com/presentation/d/1QjUtlOVqTCrJVA3GI48S2MAY_-DuAQfoKHg30_CIulM/edit#slide=id.g32dff47ed01_0_2818

6. (7.5 points): Convert the 12-bit number `111111000000` to a decimal signed integer.
Hint: remember that signed integers are stored in 2's complement form

Find the explanation on slide:
https://docs.google.com/presentation/d/1QjUtlOVqTCrJVA3GI48S2MAY_-DuAQfoKHg30_CIulM/edit#slide=id.g32dff47ed01_0_4491

Place your final answer here. -64

Name _____ Student ID _____

7. (10 points) Calculate the sum of the following two signed integers. Give your answer in binary including overflow and carryout bits. Place final answers on the boxes, one bit on each box.

	1	0	1	1	1	1	1	0	0	1	1	1	1	0	0	
	0	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1
+	0	1	0	0	0	1	0	1	0	0	0	1	1	1	1	0
	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	1

Carryout bit [0] Arithmetic Overflow bit [1]

Find the explanation on slide:
https://docs.google.com/presentation/d/1lu4IULTBQdJaqzo6HV3RMYGlztXupZiSob6-jfshBdk/edit#slide=id.g32dff47ed01_0_4210

8. (2.5 points): Consider a signed integer `var1`. Write the code or pseudocode to determine if `var1` is negative (without using the less-than operator)? *Hint: bitwise operators*

Place your final answer here.

```
boolean isNegative (int var1){
      //solution in Java
      if ( (var1>>31 & 0x1) == 0 ) {
            return false;
      } else {
            return true;
      }
}
```

Place your final answer here.

```
bool isNegative (int var1){
      //solution in C
      return (var1>>31 & 0x1);
}
```

Find the explanation on slides:
https://docs.google.com/presentation/d/1MZnk0Lj5BlU3K1G_oVNe5h4HTIrCr4cChfILOtRuFkw/edit#slide=id.g329ea2609dc_0_576

https://docs.google.com/presentation/d/1MZnk0Lj5BlU3K1G_oVNe5h4HTIrCr4cChfILOtRuFkw/edit#slide=id.g329ea2609dc_0_1256

Name _____ Student ID _____

Name _____ Student ID _____

9. (2.5 points): What does the following C program print?

```
int main() {
    int var1 = 0x6;
    int var2 = 0xc;
    int var3 = var1 & var2;
    printf("%d", var3);          //"%d" : prints var3 as an int
    return 0;
}
```

Place your final answer here. 4

Find the explanation on slides:

https://docs.google.com/presentation/d/1MZnk0Lj5BlU3K1G_oVNe5h4HTIrCr4cChfILOtRuFkw/edit#slide=id.g329ea2609dc_0_224

10. (2 points): Write one MIPS instruction to compute `$s0 = $s1 - 5`. *Hint: see page 1*

Place your final answer here. addi $s0, $s1, -5

Find the explanation on slides:

https://docs.google.com/presentation/d/1Krya4teGuaYlRy27-Bnc9fCll-Rb1kNSlThSx-imm5Q/edit#slide=id.g32ee22fb04a_0_337

11. (12 points): Complete the following Assembly code by translating this if-statement to MIPS

If `rand - perrin - lan != egwene`, then put the value 1 into `$s0`

Otherwise, put the value -1 into `$s0`

- You are not required to comment your code, but we recommend it.
- See the first page for the list of allowable instructions.
- You can use as many labels as you need.
- You may need to read from memory - but do not write to memory.

Place final answer in the next page:

Name _____ Student ID _____

```
# values are hidden so that you can't hardcode the answers!
.data
rand:       .word xxx
egwene:     .word xxx
perrin:     .word xxx
mat:        .word xxx
lan:        .word xxx

.text
main:
      # set $s3 = rand
      la $s3, rand
      lw $s3, 0($s3)
      # set $s4 = lan
      la $s4, lan
      lw $s4, 0($s4)
      # set $s5 = egwene
      la $s5, egwene
      lw $s5, 0($s5)
      # set $s6 = perrin
      la $s6, perrin
      lw $s6, 0($s6)

      # rand - perrin - lan without changing vars
      sub $t0, s3, $s6
      sub $t0, $t0, $s4

      # if $t0 == egwene jump to ELSE
      beq $t0, $s5, ELSE

      # put the value 1 into $s0
      addi $s0, $zero, 1
      j END #(1 point)

ELSE:
      # put the value -1 into $s0
      addi $s0, $zero, -1


END:
```

Find the explanation on slides:

- https://docs.google.com/presentation/d/1MZnk0Lj5BlU3K1G_oVNe5h4HTIrCr4cChfILOtRuFkw/edit#slide=id.g32a2292281f_0_91
- https://docs.google.com/presentation/d/1Krya4teGuaYlRy27-Bnc9fCll-Rb1kNSlThSx-imm5Q/edit#slide=id.g32ee22fb04a_0_217
- https://docs.google.com/presentation/d/1Krya4teGuaYlRy27-Bnc9fCll-Rb1kNSlThSx-imm5Q/edit#slide=id.g32af97efe63_0_67