CSE 260M Fall 2024 Exam 2 Due at end of session

This exam is: **closed-book**, **NO electronic devices allowed**, and **closed-notes**. The exception is the "sage page" of the designated size on which you may have notes to consult during the exam.

If anything is unclear or seems to have an error, write down your assumptions.

Be sure you: **Provide legible answers in designated areas (credit will not be given for work that is difficult to read or not where expected)**, **Ensure you clearly fill in circle/square(s) on multiple choice questions**, **Use indentation of your code to show its structure**, **Leave the exam stapled together in its original order**, **Do *NOT* attach any other pages to the exam**. You are welcome to use the blank space on the exam for any scratch work.

If you need to leave the room for any reason prior to turning in your exam, you must leave your exam and any electronic devices with a proctor.

Question:	1	2	3	4	5	6	7	Total
Points:	14	7	17	31	20	8	3	100

You must complete all the identifying information below correctly. Failure to do so is grounds for a zero on this exam:

1. Name (**print clearly**): _____

2. Student ID (**print clearly; 1 digit per underline**): _____ _____ _____ _____ _____ _____

3. You must sign the pledge below for your exam to count. The penalty for cheating will be decided during academic integrity review, but the instructors will recommend an F in this course as the minimum penalty.

   I have read the instructions on this page and I will neither give nor receive any unauthorized aid on this exam.

   _____

   (Sign above)

⟹ Do not proceed until told to do so! ⟸

⟹ Initial the top right corner of each page before starting ⟸

1. Miscellaneous concepts

   (1) (2 points) Verilog uses the concept of a _____ as a user-defined building block that often represents the entirety of a "chip" and can be connected to other parts.

   (2) (2 points) A(n) _____ is the common name for a HDL file that is used for trying to verify other HDL files behave correctly.

   (3) (2 points) How many "ports" does the RISC-V register file have? _____ .

   (4) (2 points) A(n) _____ style HDL model would be most precise representation of the schematic, gate-level diagrams, like those that were used in JLS.

   (5) (2 points) In order to have a synchronous reset in a Verilog model of a flip-flop, the reset signal must be included in the sensitivity list.

   ◯ False ◯ True

   (6) (2 points) The propagation delay for a ripple carry adder is proportional to the number of bits in the two numbers being added.

   ◯ False ◯ True

   (7) (2 points) The ripple carry adder is the fastest known circuit for adding two numbers.

   ◯ False ◯ True

2. (7 points) Given the following Verilog definition for a full adder:

```
module adder(input  logic     a, b, carry_in,
             output logic     sum, carry_out);
    ...
endmodule
```

   Using the module above, provide the complete code for a module that adds two, 2-bit numbers, named x and y, to produce a 2-bit result, named sum2, and a carry, named carry:

3. Consider the register file defined in Verilog in the Harris & Harris text:

```
module regfile(input  logic         clk,
               input  logic         we3,
               input  logic  [4:0]  a1, a2, a3,
               input  logic  [31:0] wd3,
               output logic  [31:0] rd1, rd2);

  logic [31:0] rf [31:0];

  always_ff @(posedge clk)
    if (we3) rf[a3] <= wd3;
  assign rd1 = (a1 != 0) ? rf[a1] : 0;
  assign rd2 = (a2 != 0) ? rf[a2] : 0;
endmodule
```

(1) (4 points) What logic signals are used when setting the value of 5 into register 6 (t0)?
Check all that apply

☐ a1 ☐ a2 ☐ a3 ☐ wd3 ☐ we3 ☐ rd1 ☐ rd2

(2) (5 points) What should the value on each be in decimal (for part 1: store 5 in register 6)?
Leave any that are not used or not relevant blank

a1's Val: _____ a2's Val: _____ a3's Val: _____

wd3's Val: _____ we3's Val: _____

rd1's Val: _____ rd2's Val: _____

(3) (2 points) Which most accurately describes when the value 5 is stored in register 6 (for part 1)?

◯ rising clock edge ◯ falling clock edge ◯ asynchronously, after input variables are set

(4) (4 points) What logic signals are needed to obtain the value of the second register at rd1?
Check all that apply

☐ a1 ☐ a2 ☐ a3 ☐ wd3 ☐ we3 ☐ rd1 ☐ rd2

(5) (2 points) Which most accurately describes when the value from the second register arrives at rd1 (for part 4)?

◯ rising clock edge ◯ falling clock edge ◯ asynchronously, after input variables are set

4. RISC-V Instructions:

(1) (9 points) Convert the following instruction from hex (shown with the "big end" on the left) to its equivalent RISC-V assembly language instruction: `0x0f f5 c5 13`
Show the value of each field used in binary where provided (leave blank if unused)

$funct7$: _____

$rs2$: _____

$rs1$: _____

$funct3$: _____

rd: _____

op: _____

$imm_{11:0}$: _____

$imm_{4:0}$: _____

$imm_{31:12}$: _____

Now give the assembly language representation of the instruction. Any registers should use their common names, like t0. Constants should be shown with hexadecimal notation.

Assembly Instruction:_____

(2) (9 points) Consider the following code:

```
    beq  a0, a1 next1
    add  a2, a0, a1
next1:
    blt  a0, a1 next2
    mv   a2, a1
    j end
next2:
    mv   a2, a0
end:
```

If a0 = 5 and a1 = 5 initially, what is stored in a2 at the conclusion of the code?

◯ -3 ◯ 0 ◯ 3 ◯ 5 ◯ 8 ◯ 13
◯ None of the above.

If a0 = 5 and a1 = 8 initially, what is stored in a2 at the conclusion of the code?

◯ -3 ◯ 0 ◯ 3 ◯ 5 ◯ 8 ◯ 13
◯ None of the above.

If a0 = 8 and a1 = 5 initially, what is stored in a2 at the conclusion of the code?

◯ -3 ◯ 0 ◯ 3 ◯ 5 ◯ 8 ◯ 13
◯ None of the above.

(3) (3 points) Assume that initially t0=5, t1=5 and a0=3. What will the contents of each be in decimal following the instruction: sll t0, t1, a0?

t0: _____ t1: _____ a0: _____

Answer the following questions given the code below and the contents of memory starting at location 100 (0x64) (note the order of memory addresses, which matches the order used in the RISC-V emulator used in class).

```
li a0, 0x64
li a1, 4
li t0, 2
li a2, 0

lw  s0, 0(a0)
lw  s1, 4(a0)
add t1, t0, t0
add t1, t1, t1
add t1, t1, a0
lw  s2, 4(t1)

stuff:
    addi a0, a0, 4
    lw s3, 4(a0)
    add a2,a2,s3
    addi t0, t0, -1
    bne t0, zero, stuff
end:
    j end
```

address	value
0x84	0xffffff02
0x80	0x00000010
0x7c	0x00000003
0x78	0xfffffffe
0x74	0x00000013
0x70	0x00000009
0x6c	0x00000011
0x68	0xfffffffd
0x64	0x0000000a
0x60	0x00000005

(4) (10 points) When the code completes (that is, the end is reached), what is stored in show the value in hexadecimal:

s0: _____

s1: _____

s2: _____

s3: _____

a2: _____

5. RISC-V Architecture

Consider the single-cycle RISC-V CPU model studied in class (and given in the provided crib sheet) when answering the questions below.

(1) (6 points) Indicate the order in which each part of the CPU is involed in the execution of an addi instruction, starting with the 1st to be used. Items that operate on data at the same time (no strong order) should be listed with the same number. Select "N/A" if the part is not applicable. Items may be used more than once. Ignore the impacts of expicit multiplexors (assume they take zero time and don't impact order of the other listed parts).

ALU:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A
Control Unit:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A
Data Memory:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A
Extend:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A
Instruction Memory:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A
Register File:	☐ 1st	☐ 2nd	☐ 3rd	☐ 4th	☐ 5th	☐ 6th	☐ N/A

(2) (2 points) The propagation delay required for xor is less than the propagation delay required lw:

◯ False ◯ True

(3) (2 points) The propagation delay required for bne is less than the propagation delay required lw:

◯ False ◯ True

(4) (2 points) Which of the instructions below is the most likely to determine the clock rate?

◯ add ◯ addi ◯ beq ◯ lw ◯ sw

(5) (8 points) What values should the control lines have for the instruction `bne` if the branch will be taken. Provide the value in binary or "X" if it's not relevant.

PCSrc: _____

ResultSrc: _____

MemWrite: _____

ALUSrc: _____

ImmSrc: _____

RegWrite: _____

6. Consider the RISC-V micro architecture implementations covered in Chapter 7 on a program with 1000 instructions that is a mix of simple R-type instructions, loads, and stores (no branches):

(1) (2 points) Which implementation will take the fewest clock cycles to complete the program:

◯ Cannot be determined ◯ Multi-cycle ◯ Pipelined ◯ Single-cycle

(2) (2 points) Which implementation will take the most clock cycles to complete the program:

◯ Cannot be determined ◯ Multi-cycle ◯ Pipelined ◯ Single-cycle

In general (not with regard to the example above)

(1) (2 points) Which implementation will take the fewest clock cycles for a single instruction:

◯ Cannot be determined ◯ Multi-cycle ◯ Pipelined ◯ Single-cycle

(2) (2 points) Assuming they are designed with the same basic components, which implementation will take the least actual time for a sinlge add:

◯ Cannot be determined ◯ Multi-cycle ◯ Pipelined ◯ Single-cycle

7. (3 points) Briefly summarize what you've learned about digital logic and its role in computer design: