University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS151/251A J. Wawrzynek
Spring 2021 3/11/21

Exam 1

Name: _____

Student ID number: _____

Class (EECS151 or EECS251A): _____

You have three hours to take the exam. This exam comprises a set of questions with 1 point per expected minute of completion with a total of approximately 135 points. As with homework problems, submit your solutions using Gradescope. At the end of the exam time, you will have extra time to scan and submit your answers.

You are allowed to refer to your notes, the class lecture notes, and any other reference materials that you have available. You are not allowed to speak or communicate with anyone on any topic related to the exam during the exam period. After completing the exam, sign the following statement attesting that you did not discuss the exam problems with anyone else. You may either scan this page or copy the statement word-for-word.

I hereby declare that I have not spoken with nor otherwise communicated with anybody regarding the content of this exam while taking the exam (except for course staff):

(sign here): _____

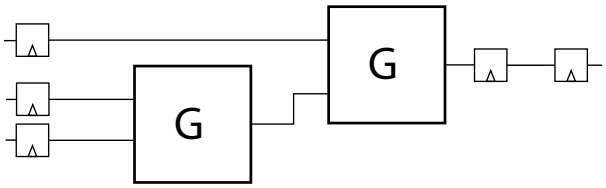
For each problem if you find yourself taking excessive time to work out a solution consider skipping the problem or a fresh approach. Also, start by answering the easier questions and then move on to the more difficult ones.

Neatness counts. We will deduct points if we need to work hard to understand your answer.

Before you turn in your exam, write your student ID number on all pages.

1. Tradeoffs [12 pts]

   You're working for an ASIC design company and responsible for a part of the company's latest chip design. You come up with a design shown in block diagram below.



   The blocks labeled G are combinational logic blocks and both implement the same Boolean function. For your process, flip-flops have the following area and delay:

$$area_{FF} = 5\,\mu m^2$$
$$\tau_{clk-q} = \tau_{setup} = 10\,ps$$

   The area and delay of your G block is:

$$area_G = 40\,\mu m^2$$
$$\tau_G = 30\,ps$$

   Your boss tells you, thanks for the design, and asks you to make sure your implementation meets the following specification for total area and critical path delay:

$$T_{max} = 80\,ps$$
$$area_{max} = 80\,\mu m^2$$

   Your implementation of G comes from a circuit generator. You had tried a range of parameters with the generator and discover that it can produce a family of designs trading-off area for delay with the following values:

area (μm^2)	10	20	30	40	50
delay (ps)	120	60	40	30	24

   (a) Does your original design meet the specifications?

   (b) If not, can you come up with a new design that does? *You are allowed to pick different G generator parameters and rearrange the circuit, as long as you maintain the same input/output functionality.*

2. FPGAs [15 pts]

(a) John is a theoretician at heart but works on FPGA architecture research. He comes up with the following two formulas for expressing LUT area and LUT delay as a function of N, the number of LUT inputs, for N ≥ 2. In the formulas k_a and k_t are layout and process related constants.

$$Area = k_a \cdot 2^N$$

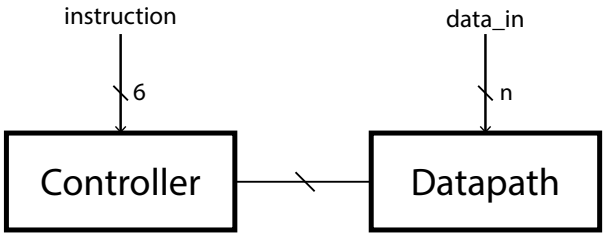
$$Delay = k_t \cdot N$$

Are John's formulas accurate? Explain. What are the constants, k_a and k_t, meant to represent?

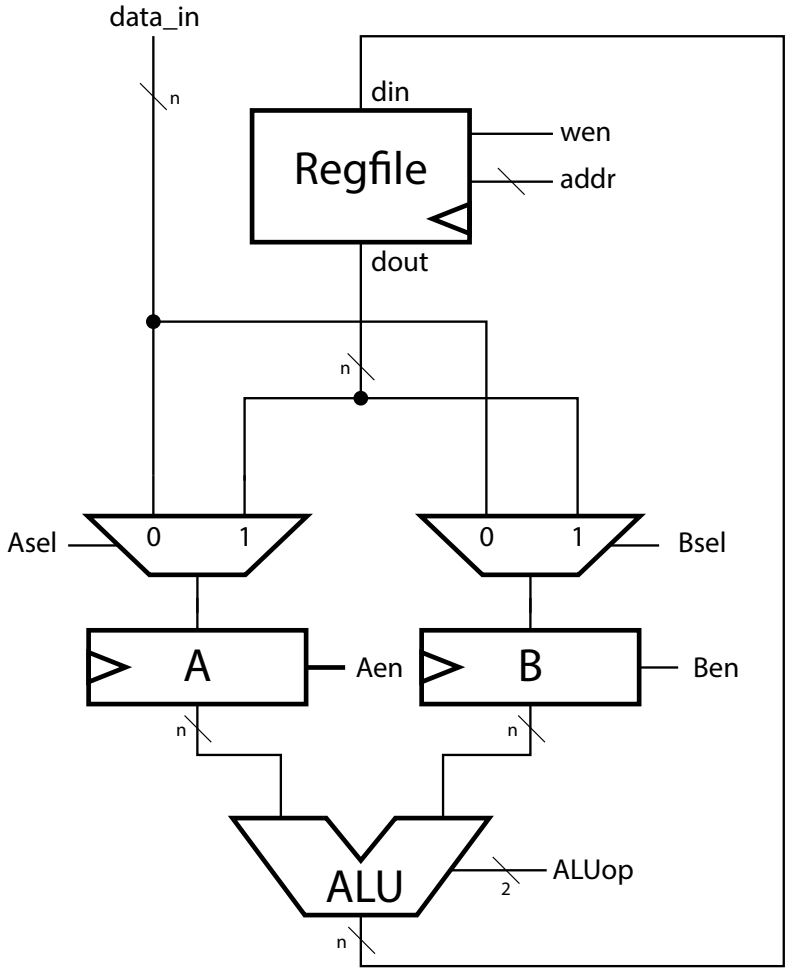
(b) Consider the problem of mapping an arbitrary function of P inputs using only 3-LUTs. Without knowing details of the function, write a formula that expresses the least number of 3-LUTS needed to implemement any arbitrary function. Assume $P \geq 3$. Explain your approach and show your work.

3. Verilog [25 pts]

Consider the design of a simple n-bit wide computation engine that accepts and executes exactly one instruction per clock cycle. At the top level the engine comprises a controller and a datapath, as shown below.



Details of the datapath are shown below. The register file (regfile) has asynchronous read and synchronous write.



The engine has an instruction set with only 6 instructions, described below. All instructions use the same format shown here.

```
Instruction Format:

| opcode |  reg   |
 5      3 2      0
    <- bits ->
```

```
Instruction Definitions:

opcode | action
0        regfile[reg] <- A + B
1        regfile[reg] <- A - B
2        regfile[reg] <- A XOR B
3        regfile[reg] <- A NAND B
4        B <- regfile[reg],  A <- datain
5        A <- regfile[reg],  B <- datain
```

Assume that an external circuit presents one instruction per cycle to the engine and holds it constant throughout the cycle.

Your job is to write the Verilog specification for four modules, the ALU, the datapath, the controller, and the engine. Remember, *no register inference*—use register instantiation. Also, you have available to you a predefined register file generator with the following definition:

```
module regfile (
 clk,    // system clock
 din,    // data input for writes
 dout,   // data output for read
 wen,    // write enable for synchronous write
 a       // address in register for write
 );
 parameter N;    // register width
 parameter M;    // number of registers
 parameter AW;   // width of address in bits
```

4. CMOS Gates [20 pts]

   In this problem you are asked to design and analyze a 4-input static CMOS gate that implements:

   $$f = ab + cd$$

   Your approach will be to use a composition of a single gate that implements $\overline{f}$ followed by an inverter.

   (a) Show your circuit diagram for f.

   (b) Size the transistors in your $\overline{f}$ gate so that each input has the same capacitance as a unit sized inverter, and derive the equation for worst case delay. Assume that the resistance per unit width for pFETs is twice that of nFETs. Also size the transistors so that the rise and fall times are equivalent.

   (c) Now, assuming that your output inverter is unit sized, derive a delay equation for the composite gate.

   (d) You realize that you might be able to scale up the size of the output inverter for better performance—a 4X inverter seems like a good choice. Derive a delay equation for this new composite gate.

   251A only — *Optional* Challenge Question for 151

   (e) An alternative way to implement this function is with NAND gates. Draw the circuit diagram for this approach and derive the delay equation with input capacitance the same as a unit inverter. Now consider scaling up the output stage. What would be its optimal size, assuming our composite gate has fanout of 4?

5. Sequential Circuit Design [15 pts]

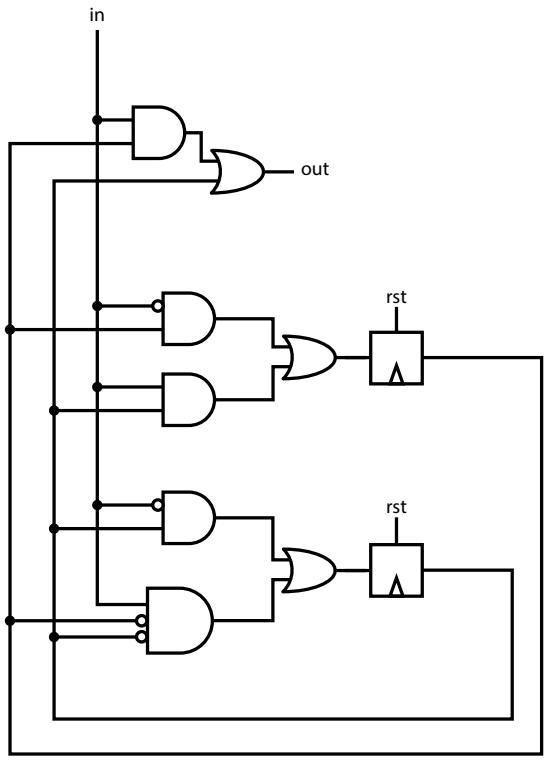
   Some particular sequential circuit has a 3-bit output labeled $[x_2, x_1, x_0]$. It outputs a new value on each clock cycle in the following repeating sequence:

   `3, 2, 5, 7, 6, 1, 4, 3, 2, ...`

   Using flip-flops and 2-input ANDs and ORs, and, if needed, inverters. Derive a circuit with this behavior. Optimize for cost by trying to minimize the number of logic gates. Show your work. *Hint: think about this as a FSM.*

6. Finite State Machine [12 pts]

   Draw the state transition diagram representing the behavior of the circuit shown below. The flip-flops reset to 0.



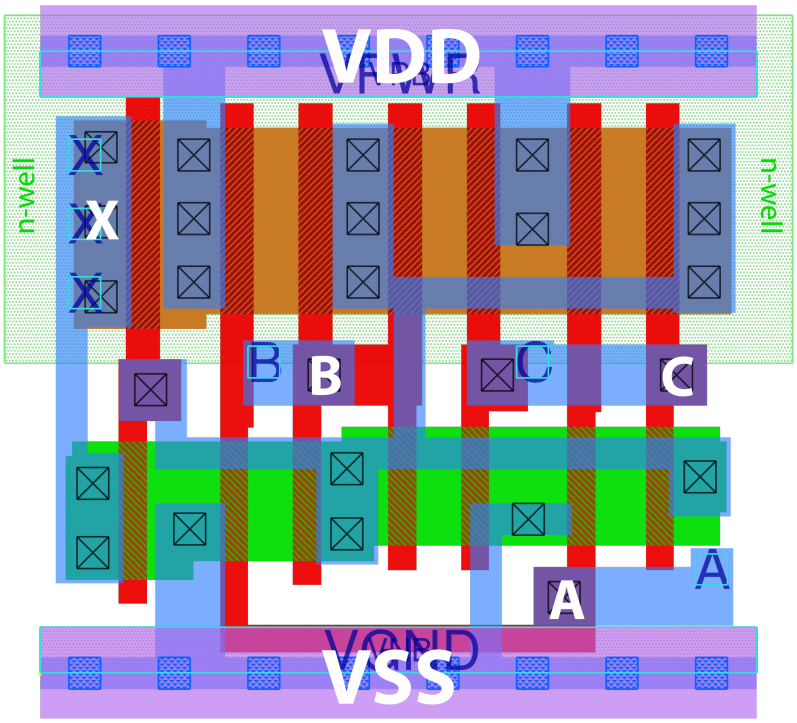
7. Layout [15 pts]

   Consider the layout shown below.

   (a) Extract the transistor level circuit diagram and sketch it.

   (b) Write an Boolean expression for its function.

   (c) Is there a common name for this function?



8. Boolean Algebra [12 pts]

   Consider the Boolean expression for some function:

$$y = abcd'e' + f$$

   (a) The inputs to a circuit for this function are the signals, $a, b, c, d, e, \& f$, none of them are available in inverted form. Draw a circuit for y using only 2-input AND and OR gates and inverters that minimizes the worst case path delay. For this problem, assume that all gates and inverters have the same delay.

   (b) Convert the circuit to one with the same function, made up of only 2-input NAND and NOR gates.

   (c) Write a Boolean expression for the NAND/NOR solution.

   (d) Using Boolean algebra show that your NAND/NOR algebraic expression is equivalent to y above.

Student ID number:

Thursday 18th March, 2021 18:40