

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

EECS151/251A
Spring 2019

J. Wawrzynek
3/14/19

Midterm Exam

Name: _____

Student ID number: _____

Class (EECS151 or EECS251A): _____

This is a *closed-book* exam, but you are allowed a single sheet of notes. No calculators, phones, pads, or laptops. Each question is marked with its number of points (one point per expected minute of time). No question is worth more than 6 points, so 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. You can tear off the spare pages at the end of the booklet and/or use the backs of the pages to work out your answers. Neatly copy your answer to the allocated places. **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. Pareto Optimality [3pts]:

John did a design space exploration for his design of a digital widget and came up with the following table of results for maximum frequency in GHz, energy efficiency in nanoJoules per operation, and cost as chip area in mm^2 . Circle those rows that represent design points that lie on the Pareto optimal frontier.

f_{max}	Energy	Cost
2.0	20	1.5
1.5	10	1.5
1.5	20	1.5
1.5	20	1.0
1.0	10	1.5
1.0	20	1.0
1.0	10	1.5
1.0	20	1.0

2. Gate Implementation [5pts]

John comes across a new technology that allows him to build two types of circuit elements:

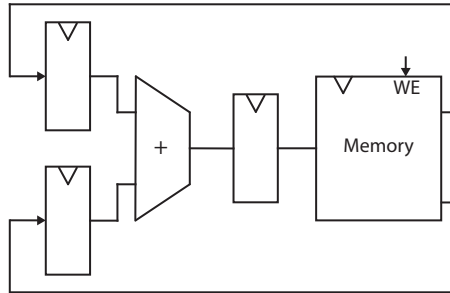
- A “multiply gate” has two voltage inputs, A and B , and at its output generates the *product*, $A \times B$, of the input voltages.
 - A “subtract gate” has two voltage inputs, A and B , and at its output generates, $A - B$, the difference of the voltages.
- (a) Explain how these gates, and only these gates, could be used to implement logic circuits.

- (b) Would this be a *practical* implementation technology? (Does it have characteristics that could create problems?) Explain.

3. Circuit Clocking [4pts]

Consider the circuit shown below. The memory block has synchronous write (a value can be written on every positive clock edge) and two *asynchronous* read ports (two values can be read at the same time and without using the clock). Assume that a new value written to a memory location can be read during the same clock cycle. A controller (not shown) generates all necessary memory addresses and write enable control.

The memory is preloaded with a sequence of numbers $\{X_0, X_1, \dots, X_7\}$. What is the minimum number of clock cycles needed to compute $\sum_{i=0}^7 X_i$, leaving the result in the memory?



4. Design Implementation Alternatives [3pts]

You work at a company designing a product with computation needs that can be met by using a single FPGA or a single ASIC. Your total expected volume for the product is 1 Million units. For your product the per chip cost estimates for the FPGA and ASIC are \$50 and \$10, respectively. The total NRE costs would be \$250 thousand and \$10 million for the FPGA and ASIC, respectively. Which would you recommend to use in your product. Why?

5. Hardware Description Languages [3pts]

John has the idea that he is going to implement his RISC processor using a behavioral description at the top-level such as below:

```
case (opcode)
  add: // ADD instruction
    begin
      regfile[rd] <= regfile[rs1] + regfile[rs2];
    end
  add: // ADD immediate instruction
    begin
      regfile[rd] <= regfile[rs1] + sign_extend(immediate);
      pc <= pc + 4;
    end
  sub: // SUB instruction
    begin
      regfile[rd] <= regfile[rs1] - regfile[rs2];
      pc <= pc + 4;
    end
  and: // Bitwise AND instruction
    begin
      regfile[rd] <= regfile[r1] & regfile[rs2];
      pc <= pc + 4;
    end
  beq: // Branch if Equal instruction
    begin
      if (regfile[rs1] == regfile[rs2])
        pc <= pc + sign_extend(immediate);
      else pc <= pc + 4;
    end

    // And so on, for all the defined instructions.
```

Briefly explain why this is not a good way to implement a processor and most likely will not lead to a good result.

6. Sequential Verilog [3pts]

Using the template below, write the Verilog for a module that implements a *negative* edge-triggered flip-flop with synchronous reset.

```
module FF_RST(q, d, clk, rst)
```

```
endmodule
```

7. Verilog Generator[3pts]

Assume we have previously defined a Verilog module DFF that describes a flip-flop. We instantiate that module to implement a 2-bit wide register below:

```
module Reg (Q, D, clk)
  input [1:0] D, clk;
  output [1:0] Q;
  DFF ff1( .q(Q[1]), .d(D[1]), .clk(clk)),
          ff0( .q(Q[0]), .d(D[0]), .clk(clk));
endmodule
```

Write a new version of this module that is a generator for an N-bit wide register using instances of DFF.

8. FPGA Logic Block [5pts]

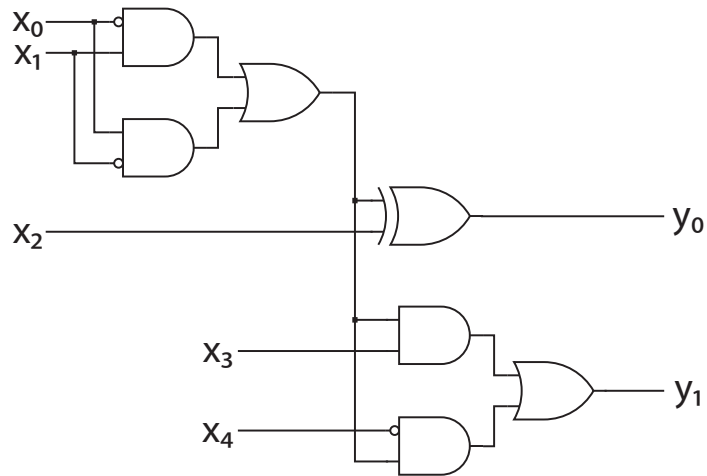
- (a) How many unique logic functions can be implemented by an n-input LUT?

- (b) As a function of n, how does the chip area consumed by an n-input LUT scale with n?

- (c) As a function of n, how does the delay through an n-input LUT scale with n?

9. FPGA Mapping [6pts]

For the logic circuit shown below, fill in the table with the minimum number of each type of LUT need for implementation. Each implementation can use only one type of LUT.



2-LUT	
3-LUT	
4-LUT	

10. Boolean Algebra [4pts]

Using the laws of Boolean algebra, derive a simplified expression for \overline{F} given $F = ab + bc$.

11. Canonical Forms [3pts]

Express the following function in its *canonical* sum-of-products form: $F = ab + cd$.

12. K-Maps [3pts]

Use a K-map to simplify the following expression and leave in products-of-sums form:

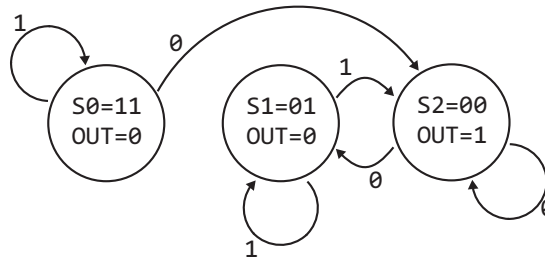
$$F = (\overline{a} + b)(\overline{a} + \overline{b} + d)(a + b + \overline{d})$$

13. Finite State Machine STD [3pts]

Draw the state-transition diagram for a Moore style finite state machine with a single bit input that outputs a one iff it sees a string of at least 2 zeros followed by exactly 2 ones. After outputting a one it starts over and outputs a zero.

14. Finite State Machine Logic [4pts]

Based on the state-transition diagram for a Moore style finite state machine shown below and the following state encoding, $S_0 = 11, S_1 = 01, S_2 = 00$, write the Boolean algebraic expressions for the next state, NS_1, NS_0 , and the output, OUT , based on the present state PS_1, PS_0 , and input, IN .



15. Finite State Machine One-hot Encoding [4pts]

Based on the state-transition diagram from the previous problem, draw the circuit for a *one-encoded* version of the state machine. You may make use flip-flops with synchronous reset and set inputs.

16. IC Technology [3pts]

What is the primary advantage of FinFET transistors over traditional planar MOS transistors?

What is their primary disadvantage?

17. IC Processing Trends [3pts]

What is the current state-of-the-art IC processing node for commercial chips (expressed in nanometers)?

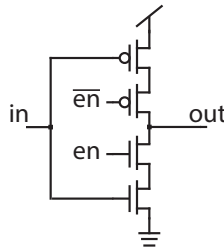
What IC processing node do you expect will be the end of Moore's Law? (Make a guess and justify it.)

18. CMOS Gates [3pts]

Draw a single complementary static CMOS gate for $F = \bar{a}(b + c)$ assuming inputs are available in both complemented and uncomplemented forms.

19. CMOS Gates [3pts]

You are given an inverting tri-state buffer (as presented in lecture and shown below). You plan to use two instances of this circuit to implement a 2x1 noninverting multiplexor with inputs in_0 , in_1 , sel , \overline{sel} , and output, y . Show your multiplexor circuit and label all inputs and output.

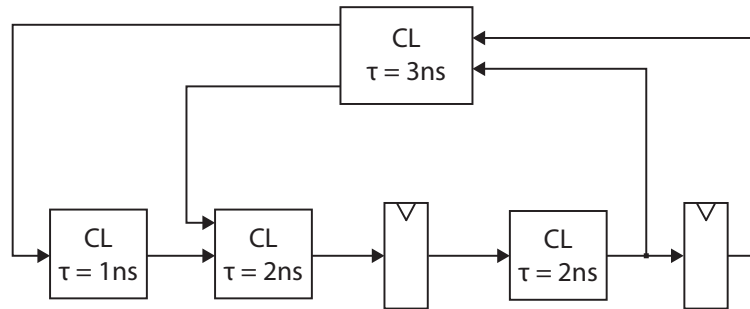


20. Gate Delay [4pts]

You would like to compose a 2-input AND-gate made up of a 2-input NAND followed by an inverter. You would also like to have increased output drive of your AND gate, so decide to use an inverter with 1/2 drive resistance of a normal inverter. The propagation delay for a normal inverter is given by $\tau_{p(INV)} = \tau_{p0}(1 + f/\gamma)$, and for the NAND (from either input to output) is given by $\tau_{p(NAND)} = \tau_{p0}(2 + 4f/3\gamma)$.

Write a formula for $\tau_{p(AND)}$, the propagation delay of the AND-gate.

21. Circuit Timing [3pts] For the circuit below what is the maximum clock frequency for correct operation?



Flip-Flop/Reg Timing:

$$\tau_{\text{clk} \rightarrow \text{q}} = \tau_{\text{setup}} = \tau_{\text{hold}} = 1 \text{ ns}$$

22. Circuit Delay [3pts]

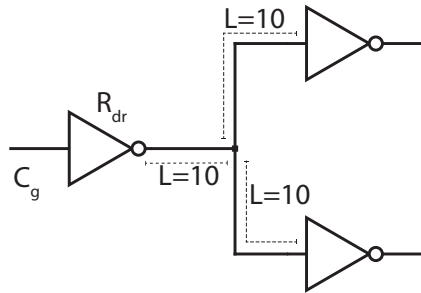
Consider a chain of three inverters with the input capacitance of the first inverter being C_g and the load capacitance of the third inverter as $125C_g$. What sizes would you make the second and third inverters to minimize the delay?

23. Flip-flop Circuit [3pts]

Consider the operation of the positive edge-triggered flip-flop presented in lecture. We usually design the internal circuits to minimize the clock setup time. With respect to the clock signal, what is the maximum value the setup time requirement will ever be?

24. Logic and Wire Delay [3pts]

Consider the circuit shown below. All inverters are the same size. The wires are short, so we will ignore the wire resistance. Assume the inverter input capacitance is C_g , and its drive transistor has resistance, R_{dr} . Also assume that the wire capacitance per unit length is c_w and is equal to $C_g/10$. Write an expression for the delay of the left most inverter.



25. **251A Only:** Wire Delay [3pts]

John understands that wire capacitance is an important part of logic delay, and so decides to reduce the width of all the wires (without reducing their thickness) in his chip design, with the hopes of reducing the delay. Will this have the desired effect? Explain.

26. **251A Only:** Dennard Scaling [3pts]:

With the application of Dennard scaling by a factor of κ , supply voltage, V_{dd} , all circuit dimensions, and doping concentrations are scaled down by κ . One consequence is that currents scales by $1/\kappa$. How does gate delay scale with κ ? Explain. *Ignore wires.*

Student ID number:

Spare page. *Will not be graded.*

Student ID number:

Spare page. *Will not be graded.*

Student ID number:

Spare page. *Will not be graded.*

Student ID number:

Spare page. *Will not be graded.*

Spare page. *Will not be graded.*