

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

EECS151/251A
Spring 2019

J. Wawrzynek
5/17/19

Final 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. Implementation Alternatives [5pts]

For a given application, fill in the rows of the table ordering each implementation alternative from lowest (1) to highest (4). Ties are allowed.

	full-custom	gate-array	standard-cell	FPGA
non-recurring engineering costs				
IC die area				
in-field flexibility				
attainable clock frequency				
attainable energy efficiency				

2. Design Flow [2pts]

In FPGA and ASIC design tool flows for timing verification, we have the option of using either “static” timing analysis or simulation. Explain the difference between the two and the strengths and weaknesses of each.

3. Verilog [6pts]

```
module myblock(a, b, f, r);
  input [7:0] a, b;
  input [1:0] f;
  output [7:0] r;
  always @ *
    case (f)
      3'b00: r = a + b;
      3'b01: r = a + 8'b1;
      3'b10: r = a -{ b;
      3'b11: r = a -{ 8'b1;
    endcase
endmodule
```

- (a) Draw a block diagram for the unoptimized circuit generated from the above Verilog code.

(b) Draw a block diagram for an optimized version.

4. Verilog [6pts]

Using a Verilog module for register generation of width N (*module declaration shown below*):

```
module Reg #(parameter N=1)
    (input [N-1:0] D, input CLK, input RST, output [N-1:0] Q);
```

Devise three different ways to specify a 4-bit shift-register in Verilog, that shifts on every clock cycle. It has input Din and output Dout. Write that Verilog code below:

(a) Version 1:

(b) Version 2:

(c) Version 3:

5. FPGAs [4pts]

Suppose we implement our FPGA 5-LUTs using only flip-flops and 2-to-1 multiplexors.

(a) Our flip-flops have a $\tau_{setup} = \tau_{clk-q} = 10ps$. The multiplexor delay from any input to output is 20ps. Ignoring wires, what is the value of the delay from LUT input to output assuming the configuration is stable and will not change?

(b) Our flip-flops have a chip area of $1\mu m^2$ and the multiplexors have an area of $0.5\mu m^2$. Ignoring wires, what is the total area of the LUT?

6. Logic Design [5pts]

Design a 2-bit gray-code counter with the following sequence: 00, 01, 11, 10. Draw the circuit using flip-flops and simple logic gates.

7. Combinational Logic [7pts]

Consider the following Boolean function: $\bar{a} \bar{b} d + \bar{b} c \bar{d} + a \bar{b} d$.

(a) Use a K-map to simplify.

(b) Use Boolean algebra to simplify. Show your work.

8. Finite State Machine [7pts]

Consider a Moore-style FSM with the following Verilog description for its combinational logic, where the next state signals are represented by “ns” and the present state signals by “ps”.

```
always @*
begin
  out = 1'b0;
  case (ps)
    S0    : if (in == 1'b1) ns = S1;
           else ns = S0;
    S1    : if (in == 1'b1) ns = S2;
```

```
        else ns = S1;
S2      : if (in == 1'b1) ns = S3;
        else ns = S2;
S3      : begin
        out = 1'b1;
        if (in == 1'b1) ns = S3;
        else ns = S0;
        end
    endcase
end
```

The FSM resets into the “S0” state.

- (a) Neatly draw the circuit diagram for the complete FSM circuit using *one-hot encoding*. You may assume that flip-flops have both “reset” and “set” inputs.

- (b) Now assume the states are encoded as 00, 01, 10, and 11 for S0–S3, respectively. Derive simplified equations for the next state (ns) and output (out) signals.

9. CMOS gates [3pts] Neatly sketch a CMOS gate with the following function (inverted inputs are not available):

$$y = \bar{a} \bar{b} + \bar{c}$$

10. Power [2pts]: Write down a formula to explain the *switching energy* of a CMOS circuit and briefly explain what each term represents.

$$E_s =$$

11. Power [1pt]: Write down a formula for the *dynamic* power consumption of a CMOS circuit in terms of switching energy.

$$P =$$

12. Power [2pts]: A design technique for improving energy efficiency in a digital IC is to use high V_t transistors for logic gates on the circuit paths with high timing slack. Briefly explain why this technique is effective.
13. Power [6pts]: A processor core runs at 1GHz with a V_{dd} of 1 volt, and consumes an average of 2 Watts while running some benchmark suite that takes 10 seconds. 50% of the power consumption is in static and 50% in dynamic power.
- (a) How much dynamic energy is consumed running the benchmark?

 - (b) Suppose we lower the clock frequency to 900MHz and run the benchmark. How much dynamic energy now is consumed?

 - (c) Now suppose in addition to lowering the clock frequency, you want to lower V_{dd} .
 - i. What is the approximate minimum V_{dd} that you could achieve at 900MHz.

 - ii. What is the effect on the dynamic energy.

14. CPU Design [4pts]: The RISC-V ISA (instruction set architecture) includes, in addition to BEQ and BNE (branch if equal and branch if not equal), BLT and BGE (branch if less than and branch if greater than or equal). The MIPS processor has a very similar ISA as RISC-V, but does not include BLT and BGE, nor any equivalent instructions. Instead the software must first use a SLT (set on less than) instruction followed by a BEQ or BNE instruction.

Explain the difference in the **single-cycle** datapaths for the RISC-V and MIPS with regard to these instructions.

15. Memory [4pts]: Some memory block is organized as 512x8-bits. The bits are arranged in a cell array that has the same number of rows as columns. The address bits are defined as, a_{N-1}, \dots, a_1, a_0 , for the appropriate value of N, are divided between the row decoder and the column decoder. Find the value for N and specify which bits are used with each decoder.

(a) row decoder address bits:

(b) column decoder address bits:

16. Memory [2pts]: Which has less latency for a read operation, an SRAM array or a DRAM array? Explain.

- (b) Now ignoring resistance assume the wires each have an inductance of 0.1nH . How many wires are needed to keep the voltage drop less than 10% of V_{dd} ?

20. Clock Distribution [2pts]: A good on-chip clock distribution wiring will balance the physical and electrical length of wires from the clock source to all flip-flops and memories. Pairs of inverters are often used along the way to buffer the clock signal, even though they can add skew and jitter to the clock signal. Briefly explain the purpose of using buffers in a clock distribution network.

21. FIFO Design [3pts]: Suppose you are given memory blocks with a single port that on each cycle can either read or write, but not both. You wish to use these memory blocks to implement a typical FIFO that allows simultaneous read and write operations with one clock cycle. Without showing all the details, explain how you would use this block, along with additional logic gates and flip-flops to implement the FIFO.

22. Adder Design [5pts]: The carry-select technique presented in lecture can be applied hierarchically. Imagine applying the technique in a binary way—a larger adder is divided in half using the carry-select and then the same technique is applied to the sub-adders, etc. The process would stop at 4-bit ripple adders.

(a) Assuming a FA and a MUX both have unit delay (delay = 1), what would be the delay (from data in to carry out) of a 32-bit adder?

(b) How would the delay scale with the size of the adder?

23. Adder Design [6pts]: A carry-lookahead adder used to add A to B is organized into groups of 4-bits. A group has carry into the group, c_{i-1} and carry out of the group, c_{i+3} .

(a) In terms of the propagate and generate signals, p_i and g_i , of the individual bit stages, write an expression for the *group propagate signal*, P and *group generate signal*, G .

(b) Write an expression for c_{i+3} based on the group P and G .

- (c) Write an expression for the bit stage sum output, s_i .
24. Multiplier Design [6pts]: The “shift-and-add” multiplier circuit from lecture completes an $N \times N$ multiplication in N clock cycles. We can modify this multiplier to use the Booth encoding technique.
- (a) Explain the changes to the multiplier circuit to enable this change to Booth.
- (b) How many cycles would be needed for a $N \times N$ multiplication?
25. Multiplier Design [2pts]: For constant multiplication using CSD representation, what is the maximum number of adder/subtractor circuits needed for multiplying an N -bit number by an N -bit constant?
26. Shifters [3pts]: In a *Barrel Shifter/Rotator*, how does the delay from data input to data output scale with the number of inputs, N ? Justify your answer.
27. Counters [3pts]: A synchronous binary counter of N bits wide has flip-flops labeled $0, 1, 2, \dots, N-1$. The i -th flip-flop's input and output signals are labeled ns_i and ps_i , respectively. Write a Boolean expression that represents the function of the combinational logic circuit for the input signal to the k -th flip-flop.

28. Counters [3pts]: Ring counters (one-hot counters) are normally designed using a chain of flip-flops. However, for large ring-counters and when flip-flops are expensive, they can be built based on a binary counter and some additional logic gates. Assuming you have a binary counter of any number of outputs, demonstrate this scheme for a 8-bit ring counter.
29. Error Detection [4pts]: Suppose we have an 4-bit word (3-data bits and 1 parity bit). An error is said to occur when one or more bits flip in the word. Assuming that bit-flips occur independently on different bits of the word and that the probability of k bit-flips is $P(k)$. Derive an expression for the probability that the single parity bit will detect an error if one occurs.

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.*

Student ID number:

Spare page. *Will not be graded.*

Thursday 4th May, 2023 10:59