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

E. Alon

Thursday, Mar 2nd, 2017
7:00-8:30pm

EECS 151/251A: SRPING 2017—MIDTERM 1

NAME	Last First

GRAD/UNDERGRAD	

Problem 1: _____/ 12

Problem 2: _____/ 12

Problem 3: _____/ 16

Problem 4: _____/ 26

Total: _____/ 66

PROBLEM 1. (12 pts) Digital Design, Timing, and FPGAs

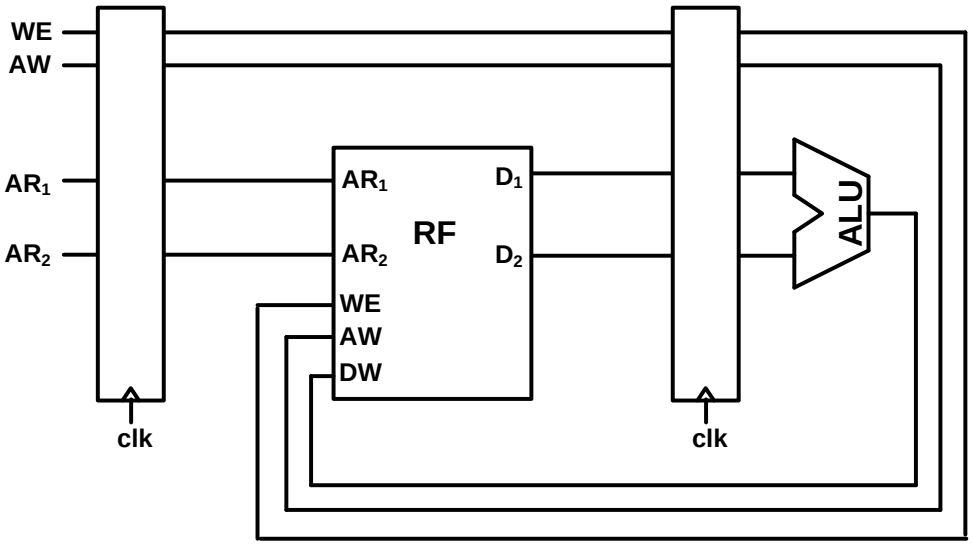
a) (**6 pts**) Draw a gate-level circuit diagram that would implement the behavior described by the Verilog code below.

```
module alpha(A, B, sel, clk, en, out)
    input A, B, sel, clk, en;
    output out;
    reg C, out;
    wire gclk;

    always @(A or B or sel) begin
            C = A;
            if (sel == 1'b1) C = B;
    end

    assign gclk = clk & en;
    always @(posedge gclk) begin
            out <= C;
    end
endmodule
```

b) **(6 pts)** For this problem we will be analyzing the timing constraints of the portion of a CPU implementation shown below. Note that the details of how this portion of the CPU actually operates are not relevant for this specific problem, but for the sake of clarity, the "RF" represents a Register File, and the "ALU" is an Arithmetic Logic Unit. For all of the flip-flops, assume that t_{clk_q} = 20ps, t_{setup} = 5ps, and t_{hold} = 25ps. Furthermore, you can assume that the delay through either the AR1 or AR2 inputs to the D1 or D2 outputs of the RF is 200ps, that the delay from the WE, AW, and DW inputs to the D1 or D2 outputs of the RF is 75ps, and that the delay from any of the ALU's inputs to the its output is 80ps.



What is the minimum clock period that the design above can support? Does the design suffer from any hold-time constraint violations? (Note that to receive credit for the answer to the second question, you must include any inequalities you used to check to hold-time constraint violations.)

PROBLEM 2. (12 pts) Finite State Machine Design

In this problem we will explore the design of an FSM that can assist with converting a single serial input bitstream into multi-bit output data. The inputs to the specific FSM we'll be designing are a serial bitstream "**s_in**" and a reset signal "**rst**" (as well as of course a clock), and the FSM has a single output "**valid"**.

The serial bitstream **s_in** is conceptually divided in to packets that vary in length from 3 bits to 6 bits; the values of first two bits in the packet determine the remaining number of bits in the packet. On the clock cycle after the final bit in the packet is received (at which point the input **s_in** will now have the value of the first bit in the next packet), the **valid** signal should be set high for the entire cycle. The mapping between the values of the first 2 bits in the packet and the total number of bits in the packet is as follows:

s_in	Packet Length
00	3
01	4
10	5
11	6

Note that in the table above, the left most bit is the first one to be received in time (i.e., 01 means that the first bit on **s_in** in the packet was 0, and the second bit in the packet was 1). Note that you should assume that the packets on **s_in** are received continuously (i.e., one new bit each clock cycle), and the reset signal **rst** was asserted once at the beginning of the FSM's operation to indicate the beginning of the first packet.

a) **(2 pts)** Based on the description above, should this FSM be implemented as a Moore or a Mealy type state machine? You will only receive credit for this part if your answer explains why the FSM must be one type or the other.

b) **(10 pts)** Ignoring any transitions associated with the initial reset, draw a state-transition diagram for an FSM that operates as specified on the previous page.

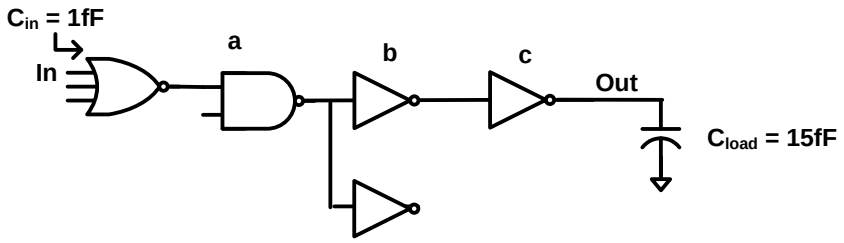
PROBLEM 3. (16 pts) Gate Design

a) **(8 pts)** Implement the function $F = \overline{A+B+C\cdot D}$ with a complex static CMOS gate. Assuming that for this process $R_P = 0.75*R_N$ (i.e., for the same width, a PMOS has 0.75 times the resistance of an NMOS), size your gate so that the worst-case pull up resistance is equal to the worst-case pull-down resistance.

b) **(8 pts)** What is the logical effort of this gate from the A and C inputs?

PROBLEM 4. Logical Effort and Wires (26 points)

Throughout this problem, you can assume that the channel length L for all transistors is 20nm, C_G = 2fF/μm, C_D = 0fF/μm, R_N = 20kΩ/□ (i.e., the resistance of an NMOS transistor with L = 20nm and W = 40nm would be 10kΩ), and that R_P = 40kΩ/□.

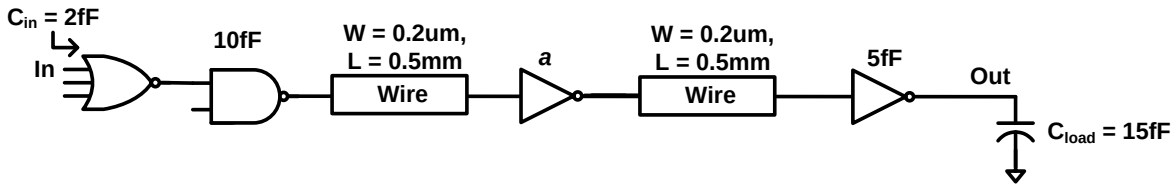


a) **(4 pts)** What is the path effort from In to Out?

b) **(2 pts)** What EF/stage minimizes the delay of this chain?

c) **(6 pts)** Size the gates (in terms of their input capacitance) to minimize the delay from In to Out.

Size	Value (fF)
a	
b	
c	



d) (**10 pts**) Assuming that the wire capacitance per unit length C_w = 0.2fF/μm and that the sheet resistance of the wire is R_w = 0.2Ω/□, as a function of *a* (the input capacitance of the inverter in the middle of the chain), what is the delay from In to Out for the circuit shown above?

e) (**4 pts**) What value of *a* (in fF) would minimize the delay of the circuit from part e)?

EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK