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

E. Alon

Thursday, May 11th, 2017
3:00-6:00pm

EECS 151/251A: SPRING 2017—FINAL

NAME	Last First

GRAD/UNDERGRAD	

Problem 1: _____/ 18

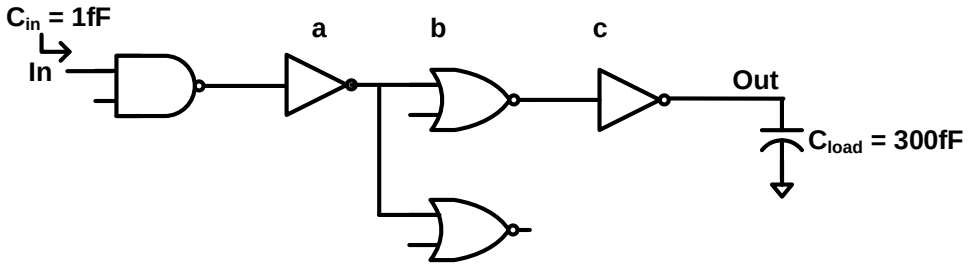
Problem 2: _____/ 12

Problem 3: _____/ 26

Problem 4: _____/ 32

Total: _____/ 94

PROBLEM 1. Logical Effort and Gate Sizing (18 points + BONUS 6 pts)



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

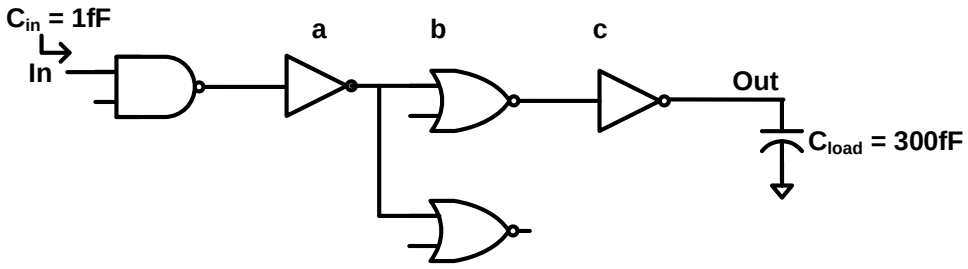
b) **(2 pts)** What EF/stage minimizes the delay of this chain of gates?

c) **(6 pts)** Size the gates to minimize the delay from In to Out.

Size	Value (fF)
a	
b	
c	

d) **(6 pts)** While maintaining the same logical functionality and without changing C_{in}, improve the delay of this chain of gates (repeated below) by adding or removing inverters; note that you don't need to worry about maintaining the logical polarity. Please draw a schematic of the improved chain; note that you don't need to provide gate sizes, but you do need to explain why your design would improve the delay in order to receive full credit.

Original chain:

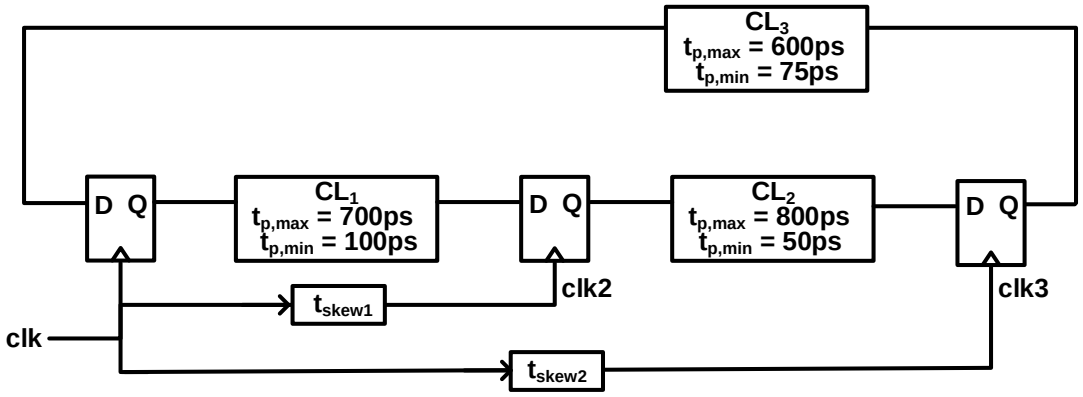


Improved chain:

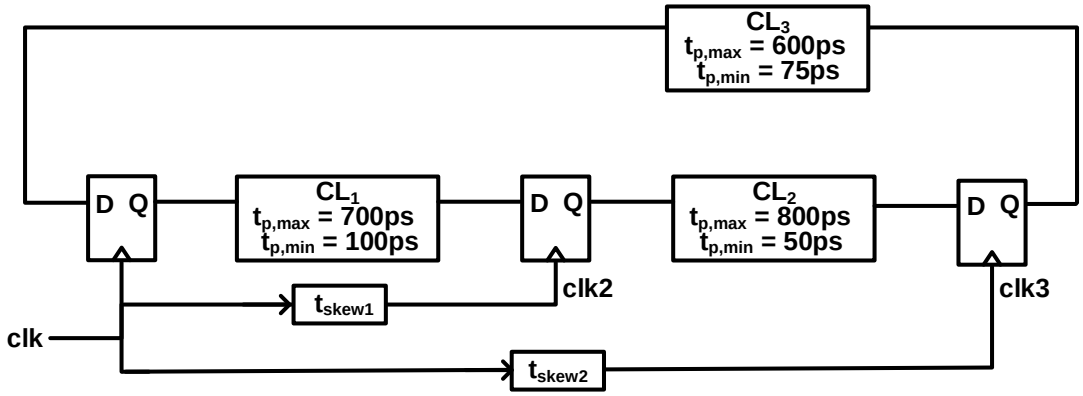
e) **(BONUS: 6 pts)** Now let's imagine that you are working in a new technology where the γ ($= C_D/C_G$) of the transistors is 1000. Would this alter your answer to part d)? If so, explain (qualitatively, but as specifically as possible) how your answer to part d) would be different. If not, explain why changing the γ of the transistors shouldn't affect your answer to part d).

PROBLEM 2: Timing (18 points)

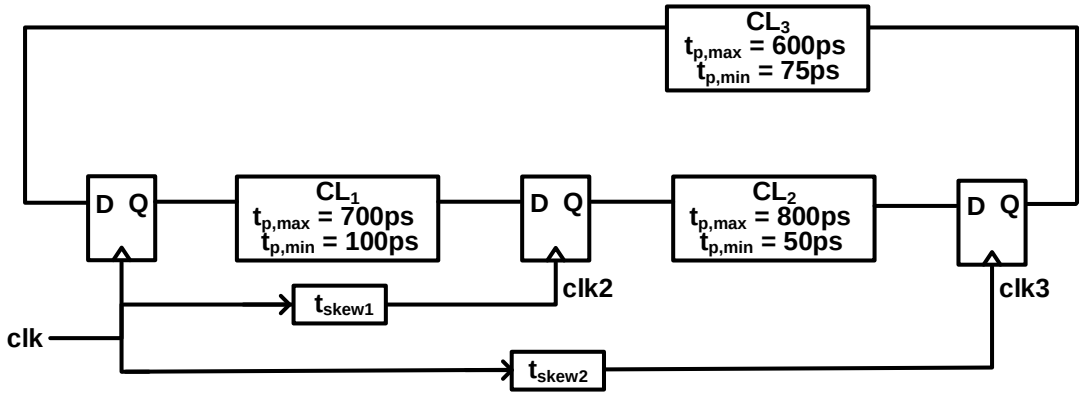
In this problem we will be examining the pipeline shown below. The minimum and maximum delays through the logic are annotated on the figure, and the flip-flops have the following properties: $t_{clk\text{-}q}$ = 50ps, t_{setup} = 25ps, and t_{hold} = 40ps. You can assume that the clock has no jitter, but t_{skew1} and t_{skew2} can be either positive or negative.



a) (4 pts) What is the minimum clock cycle time if t_{skew1} = 100ps and t_{skew2} = 50ps?



b) (6 pts) Assuming t_{skew1} is fixed at 100ps, how positive can t_{skew2} be before this pipeline (repeated above for your convenience) fails a hold-time constraint?



c) (8 pts) If you could intentionally set the values of t_{skew1} and t_{skew2}, what values would you choose in order to minimize the cycle time of this pipeline (again repeated above) without introducing any hold time errors? What would be the cycle time in this case?

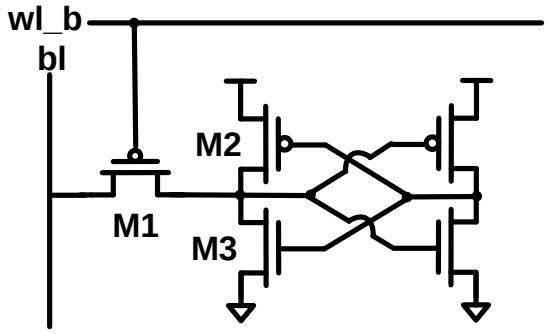
PROBLEM 3. (26 pts) Miscellaneous

a) **(8 pts)** Given *x[n]* and *x[n-1]* as inputs and using only adders, multipliers, and registers, sketch a design that uses unrolling to produce new values of *y[n]* and *y[n-1]* on each clock cycle, where *y[n]* is set by the following equation:

*y[n] = x[n] – k*y[n-2]*

where *k* is some fixed constant.

b) **(8 pts)** One of your colleagues approaches you and suggests switching all of your memories from using the classic 6 transistor SRAM cell to the 5 transistor (5T) cell shown below. Assuming that you are not allowed to use sense-amps or mode-dependent (read vs. write) voltage levels to read out of the memory, explain why it is not possible to ensure both robust read and write operations from a memory built out of these 5T cells.



c) **(10 pts)** Now assuming that you can use sense-amps (but still no mode-dependent voltages) and that the PMOS transistor M1 is sized so that its resistance is much lower than that of both M2 and M3, explain how you would read out of a memory built out of such 5T cells. In particular, you should sketch a waveform of what should happen on the bitline during a read, indicate when the sense-amp should be triggered and what the reference voltage for the sense-amp should be, and what should happen to the bitline after the sense-amp has been triggered.

PROBLEM 4. (32 points) Datapaths and Arithmetic

You recently joined a start-up named ML4eva that has decided to implement a custom processor to accelerate neural network computations, and you have been put in charge of developing the datapath for ML4eva's processor design. Fortunately, your datapath (for now) only ever needs to handle one specific instruction named MLinst.

Calling MLinst($c1, $c2, $c3, $r1, $r2, $r3, $r4) tells the processor to perform the following computation:

r4 = f (c1*r1 + c2*r2 + c3*r3)

where f () is some non-linear function, c1, c2, and c3 are the values stored in registers in a "coefficient file" (at locations indicated by $c1, $c2, and $c3), and r1, r2, r3, and r4 are the values stored in registers in a register file (at locations indicated by $r1, $r2, $r3, and $r4). In other words, register $r4 stores the result computed by applying f () to the weighted sum (with weights set by c1, c2, and c3) of r1, r2, and r3.

a) **(4 pts)** Assuming you wanted to implement a datapath that could complete an MLinst instruction in a single cycle, how many independent read ports would the coefficient file require? How many independent read and write ports would the register file require?

b) **(12 pts)** Assuming that both the coefficients and the registers each contain 2-bit unsigned values (e.g., the value stored in $c1 is c1[1:0], where c1[1] is the MSB) and using only AND gates, full adders, and half-adders, sketch an implementation of a hardware block that would compute c1*r1+c2*r2+c3*r3. In order to receive full credit, the critical path of your design should be no more than 6 gates long (where you can assume all gates have equal delays; in this context you can treat both full adders and half adders as gates). Note that you will receive full credit for sketching your design using a "dot diagram" as long as you indicate how the initial partial products are formed.

c) **(16 pts)** Assuming you are given coefficient file and register file blocks as well as a block that produces as its output *f* (c1*r1 + c2*r2 + c3*r3) (given c1, c2, c3, r1, r2, and r3 as inputs), and ignoring the hardware associated with fetching the instruction, sketch a 3-stage pipelined implementation of this datapath that would consist of Register/Coefficient Fetch, Execute, and Write Back. You can assume that the coefficient file and register file have asynchronous reads but synchronous writes. For full credit, be sure to include any forwarding hardware necessary to avoid data hazards.

EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK

EXTRA PAGE FOR ADDITIONAL/SCRATCH WORK