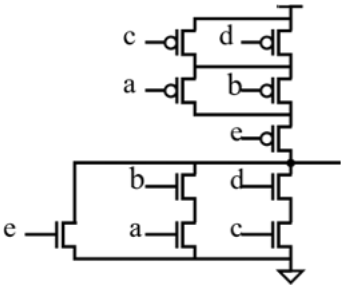
1) What are the two most important optimization criteria in digital design today? Please give reasons for your answer.

   The two most important criteria are: designer effort and energy. Designer effort is important because modern chips are incredibly complex, and the market is incredibly competitive. This means if you can't meet your time to market, you'll either go bankrupt, or by the time you reach market your design will already be outdated and no one will buy it. Energy is number 2 since every application is now energy/power limited: laptops/tablets/phones need long battery life and can't burn users or tables, desktops need to be air-coolable, and data centers biggest costs are cooling and power so energy efficiency yields huge savings.

Use this circuit for the following questions:



2) Size each of the transistors so that delay for the output rising is half the that of the output falling (excluding parasitic capacitance).

There are many ways to approach this problem, but the correct solution will have worst-case pmos paths have half the drive resistance as worst-case nmos paths.

To start, I'll size NMOS "e" as W. NMOS transistors a, b, c, and d each are in series with another transistor, so they'll be 2W each. For PMOS, in each worst-case, a, b, c, d, and e are all in series with 2 other transistors (each transistor is in 3 transistor chain), so they need to be 3 times as wide as they'd otherwise be to achieve unit resistance. They need to be twice as wide as a single NMOS due to PMOS's limited carrier mobility, and for PMOS to achieve half the rise time, they need to be doubled again. So $PMOS = W * (2_{carrier}) * (3_{series}) * \left(2_{fastrise}\right) = 12W$.

Summarized:

	NMOS	PMOS
a	2W	12W
b	2W	12W
c	2W	12W
d	2W	12W
e	W	12W

3) Now I resize the above circuit so that inputs a, b, c, d, and e to have the same rising and falling delay. If, after this, I go back and resize input e's NMOS so that input e has the same input capacitance as the other inputs, what would the logical effort for the output falling transition for input e be?

   To resize for equal rise and fall delay, I just half the PMOS width, so PMOS a-e are all 6W. To resize e to have the same input capacitance, NMOS e becomes 2W. For my logical effort calculations, I'm going to use a reference inverter with the same drive strength as my 2W input e: PMOS inv = 4W, NMOS inv = 2W. My input capacitance for e is equal to $NMOS_e + PMOS_e = 2W + 6W = 8W$. My inverter has input capacitance of 6W. Since I sized my inverter to have the same drive resistance as the falling output for input e, I just take the ratio of my input capacitance for e and the inverter capacitance:

$$LE_e = \frac{R_e C_e}{R_{inv} C_{inv}} = \frac{8W}{6W} = \frac{4}{3}$$

4) Which of the following are true about static timing analysis problems? Please select all that apply.

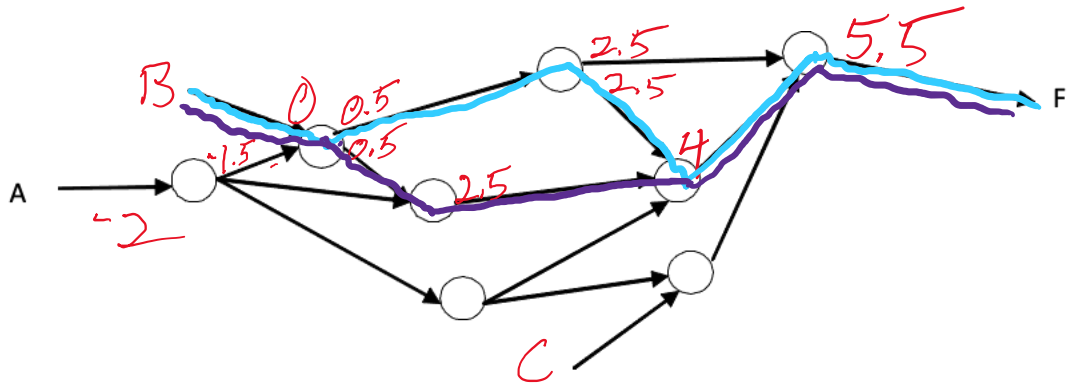
   Correct descriptions:

   - Static timing analysis only works on acyclic combinational logic
     Explanation: If you have feedback (cycles), the longest path in the tools eyes could be infinite as you just loop through the cycle forever.
   - The quality of static timing analysis results depend in part on estimates for gate delay
     Explanation: It would be too slow to run SPICE on all of our gates, so we have to estimate gate delays. If our estimates are too conservative, we put in a lot of unnecessary effort to get to our target clock speed. If our delay estimates are too loose, our final circuit can't operate at the target clock speed. Either way, it's important to have good estimates.

   Incorrect:

   - Static timing analysis enumerates all paths through the circuit
     Explanation: This would make STA take exponential time, rendering it infeasible for complex circuits
   - Static timing analysis does not account for interconnect delay
     Explanation: It does account for this, or at least it tries to. While interconnect may not matter much for compact circuits, if we don't account for interconnect delay on longer wires, STA wouldn't be of much use
   - The path with the most slack is the critical path
     Explanation: The path with the *least* slack is critical

5) Assume the "wire" delay along each edge is 1, and the "gate" delay at each circuit is 0.5. Also, assume input A arrives at -2. Please draw the critical path(s) on the graph, and indicate its/their delay.



6) Which of the following are true about CMOS and modern digital design?

Correct descriptions:

- CMOS logic cells are generally inverting (inverters instead of buffers)
  Explanation: PMOS are bad at pull-down and NMOS are bad at pull-up, so we have inverting logic.
- In modern design, area for logic is essentially free
  Explanation: Every few years we get double the transistors, but not double the ability to specify logic. Metal routing layers don't double either, so we find ourselves with cheap-as-free gates. In modern designs, gates and local routing are cheap-to-free, but design/verification time and global routing are expensive

Incorrect:

- Components are cheap because transistors are printed in series
  Explanation: Transistors are cheap because they are printed in parallel (all at once)
- The optimal number of gate stages in a circuit does not depend on parasitic capacitance
  Explanation: The optimal number of stages trades off the fixed delay of gates (sum of parasitics) with the effort/load driving delay of gates (sum of effort delays) to achieve overall minimum delay
- Static power has surpassed dynamic power as the key source of energy dissipation in most digital designs
  Explanation: Static power is a design parameter with can tune by changing threshold voltage. Given how important energy dissipation is, it wouldn't make sense to spend most of our power on idle circuits, so we ensure that our design's dynamic power dominates. Some very special structures, like caches, though, are dominated by static power dissipation, but the entire chip's power will be dominated by dynamic.