# Review

We will be doing mostly review problems today. We aren’t going to do all of these today, but they make for a good review, so you should consider doing them…

## Combinational logic:

1. Using only tri-state devices, 2-input AND gates, 2-input OR gates, 2 to 4 decoders and inverters, build a 4 to 1 MUX. Use the same labels as the diagram below. Your design must use *seven* or fewer devices to receive credit. **[8]**

3

2

1

0

A

B

C

D

Out

S[1:0]

1. If you had to build a 4 to 1 MUX using only two-levels of gates (not including inverters) how would you do so?
2. Using only 2-input AND, OR, and XOR gates as well as inverters draw a circuit which takes 6 bits as an input and outputs a 1 if the parity of those bits is odd. Label the inputs as I[0:5] and the output as “Odd\_P”. Your solution must have 8 or less gates (including the inverters).

## Sequential logic

1. Design a D flip-flop using a J-K flip-flop and standard gates. (JK is “hold” if both 0, clear if J=0&K=1, set if J=1&K=0, invert if both 1).
2. Draw the state-transition diagram associated with the circuit below.

A

D Q

C Q’

D Q

C Q’

CLK

B

X

Y

1. (This one is hard)
   1. Explain why you can’t build a FSM that solves the following problem:

There are two inputs X and Y and one output Z. Z is to be a “1” iff X has been a 1 more times than Y has been a 1.

* 1. If you were told by your boss: ”no really, we need this solved”, what would you do?

## Minimization

1. Using a Kmap minimize the following function W=(A⊕B⊕C)+A\*!B\*C.

2. Do Problem 1 again, but using Q-M.



## Misc.



## Setup/hold and clock skew.

**D flip-flop**

D **B** Q

C QB

**D flip-flop**

D **A** Q

C QB

**D flip-flop**

D **C** Q

C QB

1. Consider flip-flops A, B and C, each nominally clocked off of the same clock. Assume

* Each flip-flop has a set-up time of 5ns and a clock-to-Q delay of 2ns to 4ns.
* The AND gate has a delay of 2 to 6ns.
* The NOT gate has a delay of 1 to 2 ns.
* Flip-flops A and B have no clock skew between them.
* Flip-flop C’s rising edge may be as much as 0.5 ns before A and B’s rising edge or as much 2.5ns after.

1. What is the fastest clock period you could safely clock this system at?
2. What is the (non-negative) range of values for the hold time that would be sufficient?
3. Redo part a) assuming the NOT gate always had a delay of 0ns
4. Redo part b) assuming the NOT gate always had a delay of 0ns.

***This problem is quite hard.*** There was at least one semester where a company basically asked this question as their first-level filter on interviews even though they knew we didn’t teach this in 270 (or 312, at least at the time).