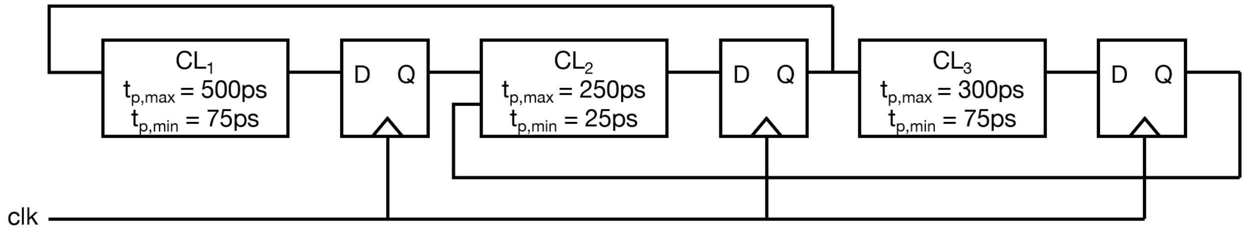
EECS 151/251A Homework 10

Due Friday, Dec 4th, 2020

Problem 1: Timing [16 points]

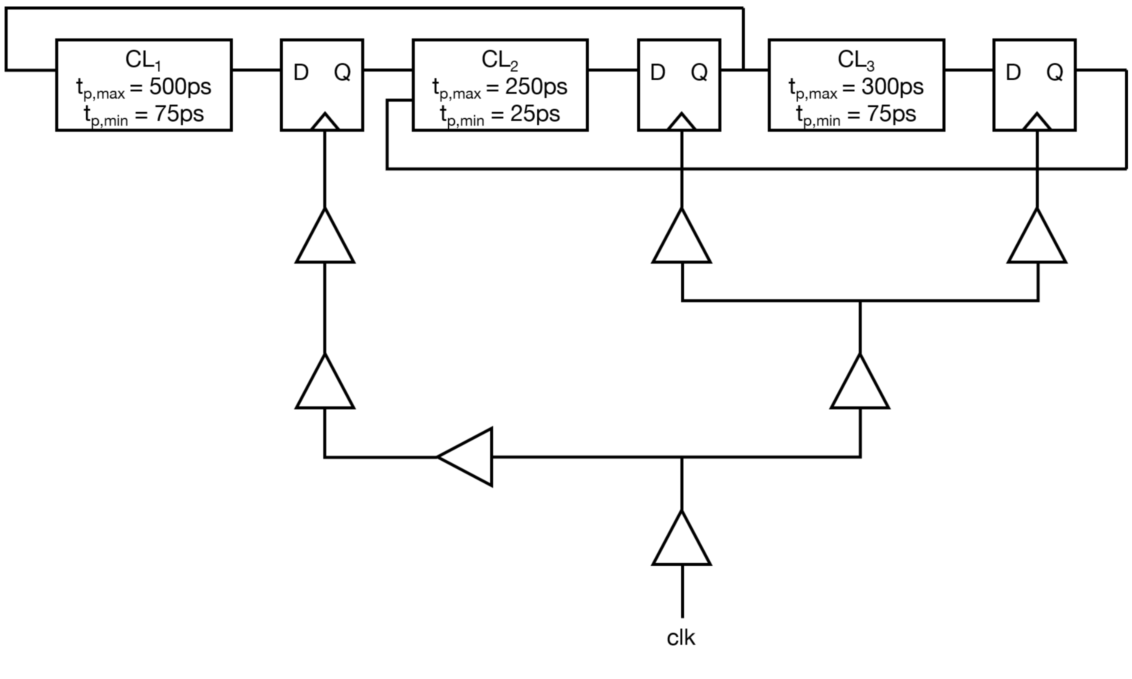
Consider the following sequential circuit. The minimum and maximum logic delays are annotated on the figure. The flip-flops have the following properties: $t_{clk-q} = 50ps$, $t_{setup} = 50ps$, and $t_{hold} = 25ps$.



a) Let's first assume that the clock has no jitter. What is the minimum clock cycle time for this circuit?

b) Under the conditions established so far, does the circuit meet all hold time requirements? Explain.

c) Now let's include a clock distribution network for this circuit, as shown below. Assume the delay of each clock buffer has an delay of 50ps (unaffected by fanout, etc.) and has no variation. Continue to assume no jitter. What is the minimum clock cycle time?



d) Under the same conditions as c), do we have any hold time violation?

e) Now let's introduce 50ps of maximum cycle-to-cycle clock jitter. Do your answers from c) and d) change? Explain.

f) Repeat parts c) and d), but now with the condition where each clock buffer's delay varies randomly by $\pm 20\%$. Analyze the timing first with no clock jitter, followed by with clock jitter (same amount as part e)).

g) (251A students only) As you have (hopefully) analyzed, this clock distribution network was designed to increase performance, but only in the absence of both clock buffer delay variation and clock jitter. In practice, both clock distribution networks and combinational paths are adjusted to meet setup and hold timing constraints. Your task is to improve on the circuit in part c) given the following rules:

- The combinational logic blocks as given cannot be modified (fixed min./max. delay, logic configuration).
- You may **add combinational logic buffers** of any amount of delay. You can assume these buffers have no delay variation.
- The depth of the clock distribution network must be at least 3 buffers deep.
- The maximum fanout of clock buffers is 2. The load can be clock buffers or flip-flop clock pins, which are assumed equivalent for this problem. Fanout doesn't affect clock buffer delay for this problem.
- Clock buffer delay variation and cycle-to-cycle clock jitter is now present, with the same values as above.

Propose a new circuit configuration that will have at least **50ps hold time margin** (we're *really paranoid!*) while **minimizing the cycle time** in the presence of delay variation and jitter. Include a diagram and explain.

Problem 2: SRAM [14 points]

The following 7T SRAM was once proposed and claimed to save power as compared to a standard 6T SRAM. Like the 6T SRAM, there is only 1 pair of bitlines, a read word line (R), and a write word line (WL). There is a new signal, W, that cuts off the feedback connection between the cross-coupled inverters before a write operation. W is logically equivalent to $\overline{WL}$ during a write operation, and high otherwise. R and WL are both high during read, while only WL is high during a write.

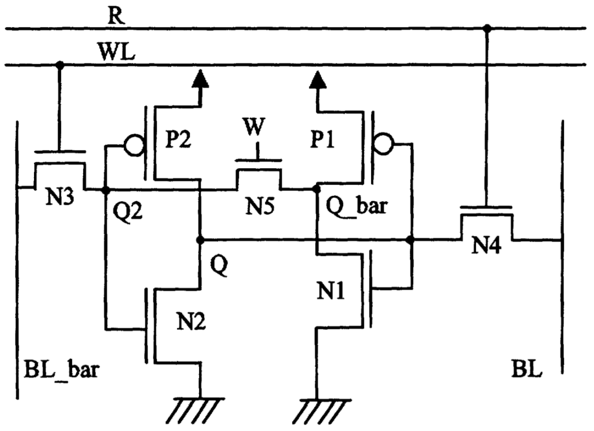


Fig. 1. The proposed 7T SRAM cell.

a) Determine which transistors are involved in a Write operation. How is this different from a 6T SRAM cell? Comment on how sizing can be used to overcome the difference between writing a "0" (BL=0) and a "1" (BL=1). Do we have the same sizing concerns as a 6T SRAM for write?

b) Determine which transistors are involved in a Read operation. How is this different from a 6T SRAM cell? Comment on how sizing can be used to overcome the difference between reading a "0" and a "1". Do we have the same sizing concerns as a 6T SRAM for read?
(251A students only) Are there any additional problems with this cell compared to a 6T SRAM cell?

c) Qualitatively explain how this scheme saves power. What might be the penalty of this? For what applications might this be useful?

d) Qualitatively explain how reducing the cell supply voltage without changing signal voltage levels external to the cell affects the read stability, read access time, writability, and write delay of the cell.
(251A students only) Does this help solve any additional problems from part b)?

Problem 3: Cache [10 points]

a) Let's review cache associativity. For a direct-mapped cache, a 4-way set associative cache, and a fully-associative cache, rank them (1st, 2nd, or 3rd) on the following metrics. Explain your ordering for each metric.

Metric	**Direct-mapped**	**4-way set associative**	**Fully-associative**
Hardware simplicity of tag checking			
Cache hit rate			
Cache hit time (for a given size)			
Cache placement policy flexibility			
Cache replacement policy flexibility			

b) For the LRU replacement policy, we need to keep track of all the relative ages of each block within a set in order to determine which one is the least recently used. For a 4-way set-associative cache, what is the minimum number of age bits per set needed to implement LRU?

c) There is an approximation of LRU that uses a binary decision tree for a 4-way set-associative cache. In this algorithm, each node of the tree denotes which half of the lines in the set are older (or newer). Draw a diagram of what this binary tree looks like and then two truth tables: one of how the decision tree bits translate to which line to replace, and the other of what the next decision tree bits would be given a reference to each line (you may need to use X's and notation for unchanged). How would this scale with larger set-associativity, and how does it compare with exact LRU?

d) Nowadays, processors use multi-level caches to improve the average memory access time (AMAT). The AMAT is defined as: $AMAT = hit\ time + miss\ rate * miss\ penalty$. Given a 3-level cache with the following specs, where a miss in a higher-level (i.e. lower number) cache goes to the next lower-level cache, calculate the overall AMAT.

Level	**Access Time**	**Miss Rate**
$L1	1ns	10%
$L2	5ns	2%
$L3	10ns	1%
Main memory	50ns	N/A

Table 1: Memory hierarchy parameters

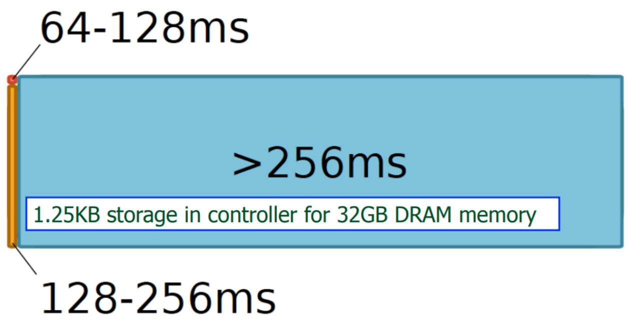
Problem 4: DRAM [Ungraded! Just for your own fun :)]

As you have learned, DRAM is significantly more dense than SRAM. Specifically, it has a footprint of 1 transistor/cell and has a trench capacitor under one terminal (look it up!). However, DRAM requires refreshing due to those leaky trench capacitors. A commonly cited characterization for DDR3 retention time is 64 ms @ 85°C. The retention time is highly inversely proportional to temperature and has cell-to-cell variation.

a) The refreshing is normally accomplished row-by-row. The average time between refreshes is specified at 7.8μs to minimally impact access latency (remember, the array is inaccessible during a refresh!). To the nearest power of 2, what is the maximum number of rows we can have in our SRAM array to ensure we never lose data at 85°C?

b) If we're limited by how many rows we can have in the array, we can make larger DRAM capacity by growing the number of columns (essentially the word size). However, the DRAM interface is only 64bits for DDR3. How should we partition the DRAM so that we don't unnecessarily destructively read and then refresh cells?

c) Liu et al. [ISCA 2012] found that 64 ms is highly pessimistic, due to it being at the tail end of a manufacturing process variation distribution. It was found retention times are more like the following:



Propose as many techniques as you wish to reduce on average how often we need to refresh our DRAM, given the retention time profile and the temperature dependence described earlier. Briefly explain how they would work and what overhead they would have. The thought process matters more than accuracy for this problem!