Problem 1: Hidden Caches

The L1-miss penalty is NOT included in addition to any L2 latencies for the L1+L2 configuration. Also there are two accepted answers because this wasn't taught in class but the more correct answer has an asterisk. This is because on average, your access time must always be greater than your hit time - even when you have a very small miss time.

(1) (a) ****L1 only AMAT**: $1 + 0.1 \cdot 5 =$ 1.5ns

(b) **L1 only AMAT**: $0.9 \cdot 1 + 0.1 \cdot 5 =$ 1.4ns

(2) (a) ****L1 + L2 AMAT**: $1 + 0.1 \cdot (3 + 0.05 \cdot 10) =$ 1.35ns

(b) **L1 + L2 AMAT**: $0.9 \cdot 1 + 0.1 \cdot (0.95 \cdot 3 + 0.05 \cdot 10) =$ 1.235ns

(3) Can assume that both caches are clocked on the positive edge, so the valid and tag bits are not available until 1 t_{cq} from the edge. After the clk-edge, the direct-mapped cache will have a critical path equal to:

$$t_{cq\_tag\_sram8x32} + t_{eq27bits} + t_{and}$$

while the 2-way-set associative cache will have a critical path of:

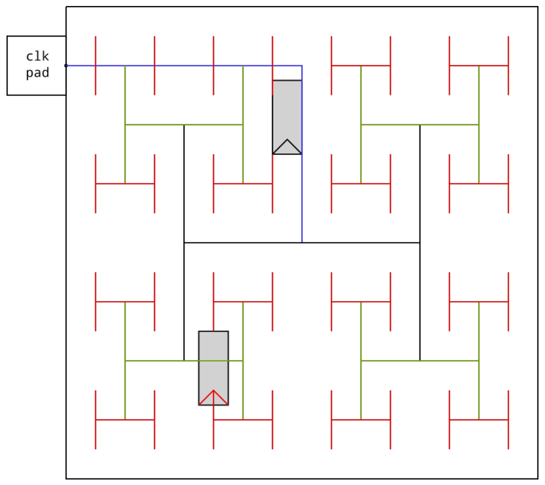
$$t_{cq\_tag\_sram4x32} + t_{eq28bits} + t_{and} + t_{mux}$$

So the direct-mapped cache might have a slightly longer sram t_{cq} but it has less circuitry afterwards, which makes it faster on a hit than a 2-way set-associative cache.

Ultimately, direct mapped caches should have a faster hit time.

Problem 2: Yea, I've got time

(1) The 3-level H-tree is shown below. In order to minimize skew in any sort of clock tree distribution, it's important to take the clock to the center and then branch out from there.



(2) This solution assumes a simple RC wire model. Also, this was listed on Piazza but was not in the original hw pdf. The gate cap can be assumed as 2pF, the clk driver's output resistance can be written as R_{clk} the Wire cap units are really aF/mm . I should also note that due to the numerous errors in the homework problem, Question 2.2,2.3,2.4 are no longer graded. This is a possible solution based off the previously stated assumptions just in case anyone is curious. A question like this WILL NOT be on the final.

- **delay 1:** $R_{clk}(1.25 \cdot 0.02\text{pF/mm}) + (1.25\text{mm} \cdot 0.1\Omega/\text{mm})(2\text{pF} + \cdot 1.25 \cdot 0.02\text{pF/mm})$
- **delay 2:** $R_{clk}(1.25 \cdot 0.02\text{pF/mm}) + (1.26\text{mm} \cdot 0.1\Omega/\text{mm})(2\text{pF} + \cdot 1.26 \cdot 0.02\text{pF/mm})$

The skew is the difference in delay from the first to second flop.

(3) Assuming the resistance and gate-cap of a unit inverter is $R_i = 1\Omega$ and C_i=2pF. Note that C_i and C_{load} and are the same, so C_i will represent the flop input cap. The propagation delay in terms of the total stages, N, where the wire length is l, the wire resistance is R_w=0.1Ω/mm, and the wire cap is C_w=0.02pF/mm (assumin a pi model for the wire), is shown below:

$$t_{pd} = N\left[R_i\left(C_i + C_w l/N + C_i\right) + R_w l/N\left(C_w l/2N + C_i\right)\right]$$

Differentiating and solving for N yields

$$\begin{aligned} N &= l/2\left(\frac{R_w C_w}{R_i C_i}\right) \\ &= 1.25/2\left(\frac{0.1 \cdot 0.02}{1 \cdot 2}\right) \\ &= 0.02 \end{aligned}$$

So, the optimal number of stages, N, is 0 (because the wire parasitics are very good). The skew is the same, since there are zero buffers.

(4) reusing the same equation from above, but plugging in $N = 3$, we solve for both flops. Note, i'm cheating here and changing the inverter input cap to 0.4pF to be able to reuse the simple approximations above:

$$t_{pd1.25} = 3\left[R_i\left(C_i + C_w 1.25/3 + 4 * C_i\right) + R_w/3\left(C_w 1.25/6 + 4 * C_i\right)\right]$$
$$t_{pd1.26} = 3\left[R_i\left(C_i + C_w 1.26/3 + 4 * C_i\right) + R_w/3\left(C_w 1.26/6 + 4 * C_i\right)\right]$$

Plugging in the rest of the numbers results in:

$$t_{pd1.25} = 12.225ps$$
$$t_{pd1.26} = 12.225ps$$

So the skew is reduced to less than 1fs, but its much slower than zero buffers.

Problem 3: Decoders

(1) Assuming NMOS/PMOS have equal drive strength the logical efforts of all three gates are:

- $C_{in} : \frac{5}{2}$
- $C_1 : 1$
- $C_2 : \frac{3}{2}$

All stages have a branching effort of 1 and the total fanout, F is 100. There are 3 stages (N = 3) and $C_W = 0$. Thus we can can derive C_1 and C_2 with the method of logical effort outlined below.

$$FGB = 100 * \frac{5}{2} * 1 * \frac{3}{2} = 375$$
$$h = FGB^{1/N} = 375^{1/3} = 7.211$$
$$C_2 = \frac{3}{2} * \frac{100C_{in}}{h} = 20.8C_{in}$$
$$C_1 = \frac{20.8C_{in}}{h} = 2.884C_{Cin}$$

(2) Assuming t_{inv} is the delay of a zero-fanout inverter. The units of t_d is number of zero-fanout inverter delays.

$$\begin{aligned} t_d &= t_{inv}\left[d_1 + d_2 + d_3\right] \\ &= t_{inv}\left[(4 + \frac{5}{2}\frac{C_1}{C_{in}}) + (1 + \frac{C_2 + C_W}{C1}) + (2 + \frac{3}{2}\frac{C_L}{C_2})\right] \end{aligned}$$

Remember that d = p + hf, where p is the parasitic delay of the gate, h is the logical effort, and f is the fan-out. As an example, for d_1, the parasitic delay is 4, $h = \frac{5}{2}$, and $f = \frac{C_1}{C_{in}}$. Remember that the parasitic delay, p, is defined by $\frac{C_{out-gate}}{C_{out-inverter}}$.

(3) This can't be solved using the typical logical effort, since the wire load, C_W, is not a fixed multiple of C_2! So we will take the partial derivative of the delay equation with respect to C_1 and again to C_2 and then solve the resulting system of linear equations.

$$\frac{\partial t_{inv}}{\partial C_1} = \frac{5}{2}\frac{1}{C_{in}} - \frac{C_2 + C_W}{C_1^2} \tag{1}$$

$$\frac{\partial t_{inv}}{\partial C_2} = \frac{1}{C_1} - \frac{3}{2}\frac{C_L}{C_2^2} \tag{2}$$

After solving with wolfram alpha and plugging in 1fF (1e-15 F) you should get:

$$C_1 = 5.21fF \tag{3}$$
$$C_2 = 27.9fF \tag{4}$$

(4) SO we will fix $C_2 = 20.8C_{in}$ and use it in addition to C_W to set the 'load' of C_1.

$$H = FGB \tag{5}$$
$$H = \frac{20.8C_{in} + C_W}{C_{in}}\frac{5}{2} \tag{6}$$
$$H = 152 \tag{7}$$
$$h = 12.33 \tag{8}$$
$$\tag{9}$$

We can use this to calculate the size of C_1 now!

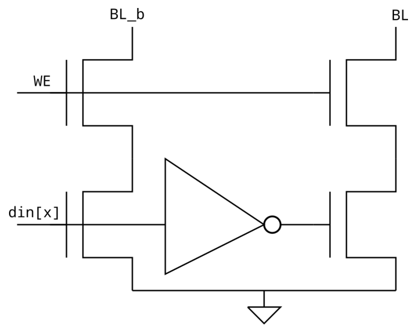
$$C_1 = \frac{20.8 + 40}{12.33}C_{in} \tag{10}$$
$$C_1 = 4.93C_{in} \tag{11}$$

I just realized that each line got an equation number ... that was a mistake, and I'm not going to bother removing them now so we'll have to deal with it together. even if it does look silly.

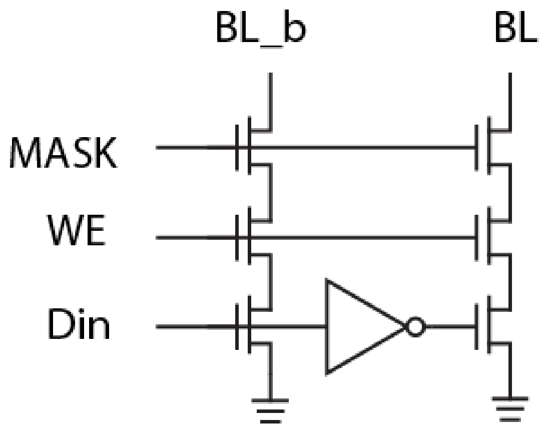
(5) The first method (part 3) provides a more optimized version of the logic chain but requires significantly more calculation and slightly more area. Part 4's heuristic is significantly easier to calculate, provides a slightly less optimum version because now the stages have different effective stage efforts, but does use slightly less area.

Problem 4: I Think I remember

(1) The write-footer is shown below. We should not drive the bitlines from `din[x]` directly, which is why the extra 2 NMOSs are used.



(2) The mask being "active low" means that when the mask bit = 0, you want to mask the cell from being written.



Problem 5: I Think I Remember Something Else

(1) You have to refresh each bit at least once every 64ms.

(2) Each row will needs to be refreshed at least once every 64ms. This means at least every 8μs, there will be a refresh, which will require a read and a write-back, making the DRAM unavailable. This is equivalent to over 128,000 refreshes per second.