EECS 312: Digital Integrated Circuits

Homework #6 Solutions

1. **Schmitt Triggers**:
   - **a.** From the description of the alpha particles, we can develop some noise constraints upon the trigger to guide the sizing of devices M3 and M4. The alpha particles "add" charge to the input node (when the input is in the '0' state, the voltage can increase), but not remove charge (when the input is in the '1' state, there is no effect). There is a constraint upon V_{M+}, but there is no constraint upon V_{M-}. Since there is no constraint upon VM-, we can minimize the size of M3 (NMOS) to minimize energy. Therefore, **W_{M3} = 0** will meet the noise constraint (none) and minimize energy.

     Now, if alpha particles deliver 40fC of charge to the input, we need to convert this into a voltage and solve for the device size of M4 that will allow the noise voltage to be less than the switching threshold of the trigger.

$$Q = CV$$

$$40fC = \left(18fF + 3um\left(6fF/um^2\right)\left(0.2um\right)\right)V$$

$$V = \frac{40fC}{(21.6fF)} = 1.85V$$

$$I_{PM2} = 30ua(10)\left((0.65-0.4)(0.25) - \frac{0.25^2}{2}\right)(1+0.1(0.65)) \approx 10ua$$

$$I_{PM4} = 30ua\left(\frac{W}{0.2}\right)\left((2.5-0.4)(0.65) - \frac{0.65^2}{2}\right)(1+0.1(0.65))$$

$$I_{NM1} = 115ua(5)\left((1.85-0.43)(0.63) - \frac{0.63^2}{2}\right)(1+0.06(1.85)) \approx 444.7ua$$

$$I_{PM4} = I_{NM1} - I_{PM2} = 434.7ua$$

$$W_{M4} = 2.35um$$

   - **b.** A static CMOS inverter can be sized to obtain significant noise immunity in one transition direction. In this problem, a static inverter could be substituted for the Schmitt trigger. However, in many cases, there will be noise on both input states, and a static inverter will not be able to protect both transitions from noise.

2. **Pipelining:**
   - **a.** Find the pipelining overhead due to registers.

$$400MHz = \frac{1}{T_{cycle}} \Rightarrow T_{cycle} = 2.5ns$$

$$FO4/cycle = \frac{T_{cycle}}{40\ ps} = 62.5\ FO4$$

$$Overhead = \frac{4FO4}{62.5FO4} = 6.4\%$$

   - **b.** Using full scaling, with S = 250/130 = 1.923, we can recalculate the FO4 delay and clock cycle time to find the overhead in the Pentium 4.

$$3GHz \Rightarrow T_{cycle} = 333\,ps$$

$$\frac{FO4}{cycle} = \frac{T_{cycle}}{40/1.923} = 16\,FO4$$

$$Overhead = \frac{4}{16} = 25\%$$

c. Using the result that 16 FO4 delays leads to 25% overhead, we can extrapolate the limit on clock frequency (barring architectural or device breakthroughs).

$$25\% \Rightarrow 16\,\frac{FO4}{Cycle} \Rightarrow 16\left(\frac{40}{250/15}\right) = 38.4\,ps = T_{cycle}$$

$$f \le \frac{1}{T_{cycle}} = \frac{1}{38.4\,ps} = 26\,GHz$$

3. C^2MOS Latch:

a. The redesigned C^2MOS latch functions the same. When CLK=1, OUT=IN' and when CLK=0, OUT is held dynamically by C_L. In the master-slave register context, the clock overlap immunity is provided as the C-Q race condition is still clocked by the same CLK signal (not CLK and CLK_bar).

b. The new latch will have a slightly better T_{DQ}, since the data arrives at the transistor closest to the output. For example, when the clock is setup beforehand, the internal nodes in each stack will be charged to the appropriate voltage, removing those capacitances from an Elmore formulation. T_{CQ} will suffer slightly, compared to the conventional design, as the CLK signal arrival will be forced to discharge the output node and the internal nodes.

T_{DQ} is generally considered a more important latch metric, as the most critical signals to a latch based-design arrive during the transparency period (right before the latch closes). Additionally, latch-based designs are usually setup to have signals arrive during the transparency period rather than before the clock edge. If this is the case, you might be wondering why this circuit is not preferred over the conventional design. The true disadvantage is explained in (c).

c. **Advantage:** Reduced clock feed-through to the output node.
Disadvantage: Potential for charge-sharing at the output node. This is the reason the latch in the text is preferred.

d. Below, in Figure 3.1 is a simple pipelined implementation of the function using only one or two input gates. Each stage has two gate delays and there are 3 stages shown. Notice that the logic in each stage is “non-inverting” to maintain the clock overlap immunity of the C^2MOS design.

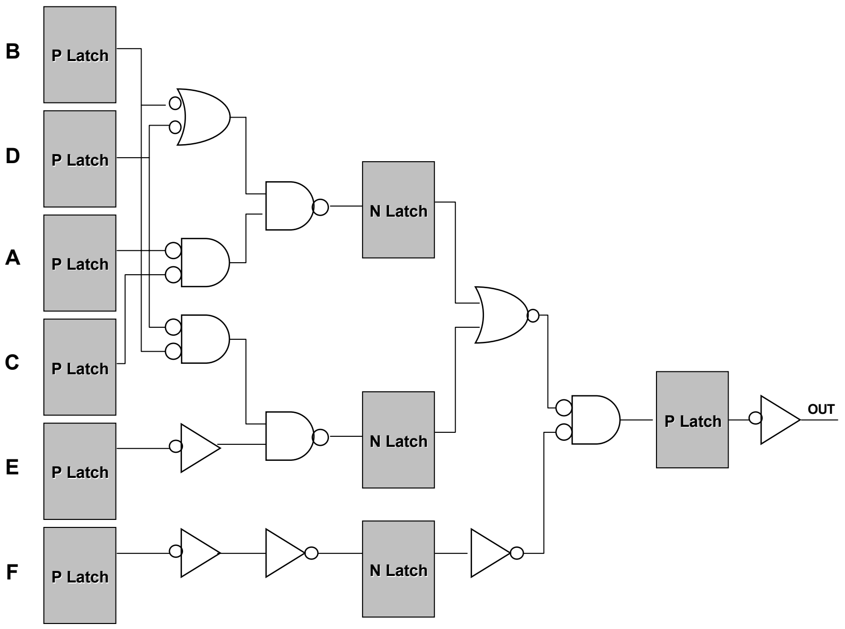


Figure 3.1

4. ROM:

a. The worst-case condition in a ROM, is activating one word line (no more than one will be active at any time). The problem is similar to a pseudo-NMOS with one pull-down device.

$$V_{OL} = 1V$$

$$I_{DP} = I_{DN}$$

$$30ua\left(\frac{W_P}{0.2}\right)\left((2.5-0.4)(1)-\frac{1}{2}\right)(1+0.1(1.5)) = 115ua\left(\frac{0.5}{0.2}\right)\left((2.5-0.43)(0.63)-\frac{0.63^2}{2}\right)(1+0.06(1))$$

$$W_P = 1.29\,um$$

b. The worst-case programming for access time of a NOR ROM, would be a word line with 32 transistors connected (gates) and a bit line with 32 transistors connected (drains). This would greatly increase the capacitive load on both bitline and word line.

$$R_{eqP} = 0.8\left(\frac{2.5}{300ua(2)}\right) = 3333.33\Omega \Rightarrow R_{10-90P} = \frac{0.55}{0.8}R_{eqP} = 2291.66\Omega$$

$$C_{word} = 32\left(6\,{}^{fF}\!/_{um^2}\right)(0.2um)(0.5um) + \left(0.6\,{}^{fF}\!/_{um}\right)(3um) = 21\,fF$$

$$C_{bit} = 32\left(0.6\,{}^{fF}\!/_{um}\right)(0.5um) + W_P\left(0.6\,{}^{fF}\!/_{um}\right) = 10.4\,fF$$

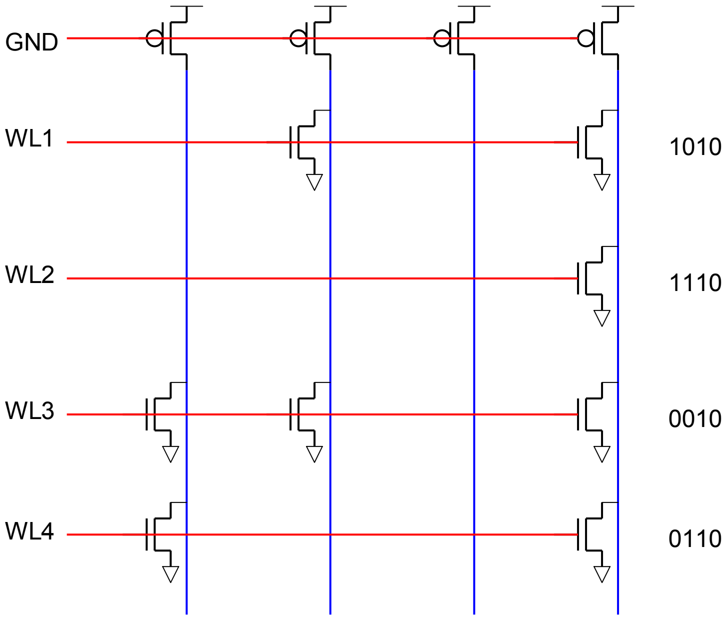
$$t_{pLH}(word) = 0.693(3333.33)(21\,fF) = 48.5\,ps$$

$$t_{rise}(word) = 2.2(2291.66)(21\,fF) = 105.9\,ps$$

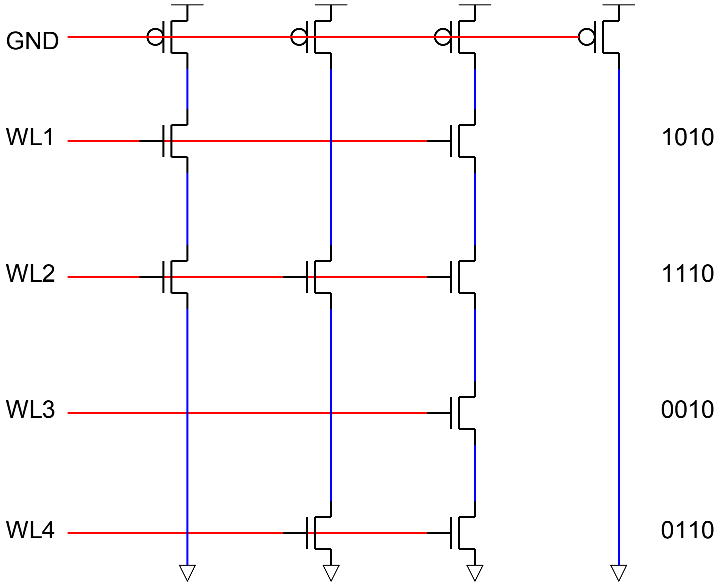
$$t_{pHL}(bit) = 0.693\left(0.8\frac{2.5}{731ua(0.5)}\right)(10.4\,fF) + 0.15(105.9\,ps) = 55.32\,ps$$

$$t_{access} = t_{pLH}(word) + t_{pHL}(bit) = 48.5 + 55.32 = 103.82\,ps$$

c. Sketch a 4 x 4 NOR ROM that stores 1010, 1110, 0010, and 0110.



d. Sketch a 4 x 4 NAND ROM that stores 1010, 1110, 0010, and 0110.


GND
WL1
WL2
WL3
WL4
1010
1110
0010
0110