

University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

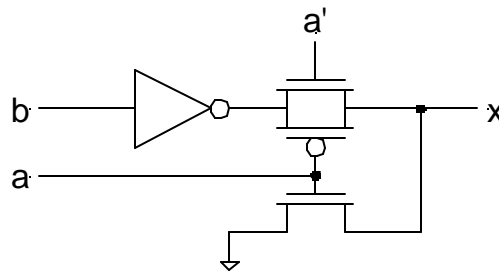
EECS150
 Spring 2000

J. Wawrzynek

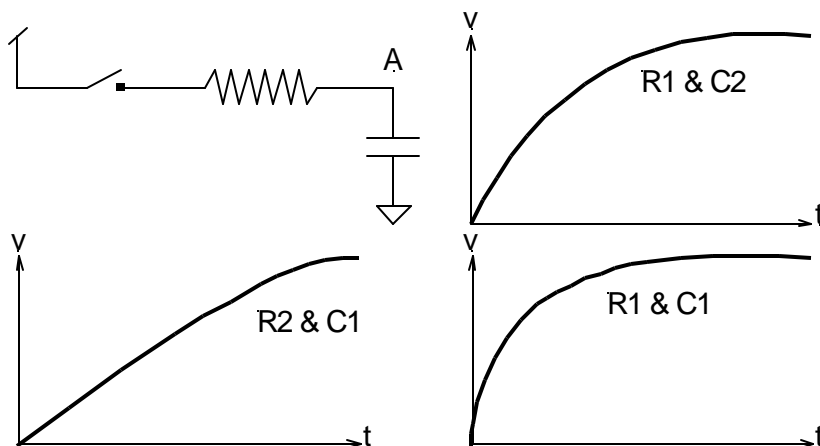
Homework #5

This homework is due on **Friday February 25th by 11am**. Homework will be accepted in the EECS150 box on the door to room 218 Cory Hall. Late homework will be penalized by 50%. No late homework will be accepted after the solution is handed out.

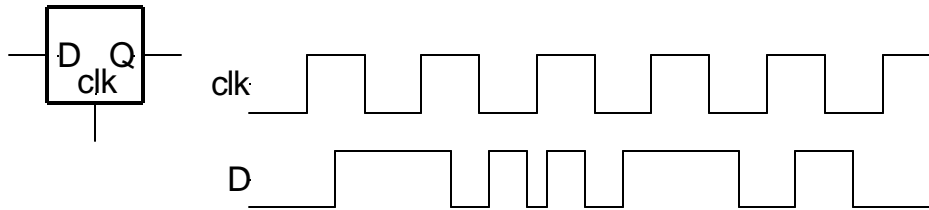
1. Using transistors in a CMOS technology, implement the following: a) 3-input NAND gate, b) 2-input OR, and c) 2-1 multiplexor using only 6 transistors. Remember that pffets can only successfully pass logic 1s and nfets logic 0s. Also, for all parts assume that complemented inputs are not available. Try to minimize the number of transistors used for each circuit.
2. What is the truth table for this circuit? What is its function?



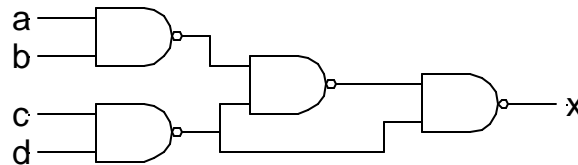
3. There are two resistors, R_1 and R_2 , and 2 capacitors, C_1 and C_2 . When one resistor and capacitor are arranged in the way presented, the voltage at A is 0 before the switch is closed. For different combinations of the resistors and capacitors, the graphs are shown for voltage at A as a function of time. All graphs are drawn with the same time and voltage scales. Which is the larger capacitor? The larger resistor?



4. Draw the waveform the output signal Q of a level-sensitive latch with the clock and D input waveforms shown below (remember that we assume that latches are transparent when the clock is high and “latching” when the clock is low):



5. For the logic circuit shown below, find the critical path from input to output. Estimate the delay of that path in units of NAND gate delays.



6. Draw the **transistor level circuit diagram** for a 2-LUT based only on level-sensitive latches and tri-state buffers. Assume that the inputs are available in both complemented and uncomplemented form. Also assume that the LUT contents are programmable, but do not draw any additional circuitry needed for programming.