

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

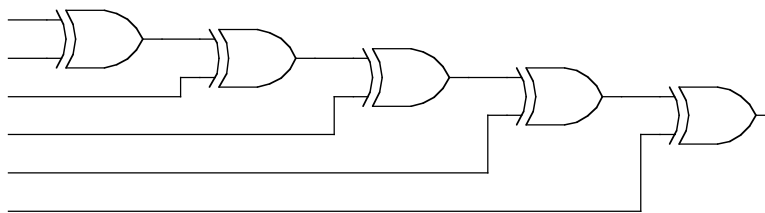
EECS150
Spring 2000

J. Wawrzynek

Homework #2

This homework is due on **Friday February 4th 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. This problem is based on the parallel to serial converter shown in class. For this problem assume the nibble 1101 is loaded and shifted out with the LSB being first.
 - a) Similar to the waveform shown in class for the final output (OUT), draw the waveform for the output of each flip-flop.
 - b) The design in class used *positive-edge triggered* flip-flops. Draw the waveform for the final output (OUT), assuming *negative-edge* triggered flip-flops.
2. Consider the design of a *bit-parallel* parity checker using 2-input XOR gates as building blocks. (Recall that a parity checker generates a 1 output if an odd number of inputs are a 1.) One possible implementation of a 6 input parity generator is shown below.



- a) If the delay through each gate is T seconds, what is the worse case delay from input to output in the parity generator?
 - b) Show another implementation of the same function with the same number of gates. Your design should operate with minimum worse case delay.
 - c) Write down the general time delay equation for your implementation when used to implement an n-input parity generator.
3. Design a **One-Hot Encoded Finite State Machine** that implements a “positive transition detector”. This circuit should have a 1 output if and only if the bit-serial input just changed from a low value to a high value. A sample input and output

stream is shown below (bits are feed into the circuit from left to right, one per clock period). Show the state transition diagram, the implementation (using FF(s) and logic gates), and the input and output waveforms for the input sequence 0110.

Input	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	0
Output	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0