CS220: Lab#8B

1. Implement a 3-to-8 decoder that takes a 3-bit value as input and produces the decoded eight-bit output. Store the following eight indices in the first eight rows of a memory that has 32 rows and each row is eight-bit wide: 3, 4, 7, 2, 3, 5, 0, 2. Use an `initial` block to store these. Maintain a row counter register initialized to zero. On every posedge of clock, you should read the row pointed to by the row counter, decode the value in that row, and increment the row counter. After decoding the eight values, show the bitwise XOR of the eight outputs in the LEDs (LED7 is the most significant bit). Also, show the parity of the XORed value. The parity is defined to be 0 if the number of 1's in the binary representation of the value is even; otherwise the parity is 1. Use LED0 to display the parity. The display should be changed from the XORed value to the parity when a push button is pressed.

Grading policy: No marks for faulty attempts. Correct implementation gets full marks. If you are able to synthesize your hardware correctly, please send a mail to cs220submit2018@gmail.com with subject "Lab#8B submission Group#X" and attach the Verilog files. Replace X by your group number in the subject line. If your design does not synthesize correctly on the FPGA, please do not send the files. The TAs will check only those submissions that they have evaluated for correctness during the lab.