Starting of with LED 0, we had to compare two different number of switches. Since there are 5 switches total (4:0), the LED is lightened once 3/5 switches are high. For LED 1, we used a NOR gate because we needed all the switches to be LOW. Looking at LED 2, this LED is lightened when 4/5 switches are low. I checked all possible combinations where at least 4/5 inputs are low. for LED 3, we used multiple XOR gates which causes the input to be 1 if an odd number of switches are on. We added an XNOR gate at the last combination to make it look for even numbers. For led 4, we used an AND gate to make sure that all the switches are high. For LED 5, I used the same method as LED 2. However, I only needed to check for when it was 2/5 inputs are LOW. Looking at LED 6, I used the same method as LED 3 but this time I didn't have an XNOR gate as the XOR gates showed me the odd number of switches. for LED 7, I used an AND gate as LED was lightened up only when all 5 switches are HIGH. Lastly, for LED 8, I used an OR gate as I needed only 1/5 switches to be HIGH for the LED to be lightened.

To implement push-button 3 to change mode of operation, we used a flag variable that toggles each time it is pressed, starting at wraparound (1'b0). This flag is checked whenever the system is in state 00 and 99 and recognizes whether the flag is wraparound (1'b0) or rebound (1'b1). If the flag triggers a rebound operation at state 99, it will rebound to 98 and continue in the downwards direction towards state 00. However, if the state is at 00, the next state will be 01 and remain there until the direction is manually changed. To assure that all push buttons are ignored when in state 00 or state 99, there is an if statement that allows the push button functionality to be used if the conditional is met. As the flag is set on push button 3, it can still be toggled while the counter is stopped since it is independent from it unless its at state 00 and state 99. Finally, the power-up state of the system is in state 00, while operating in the wraparound mode.