Exp# Mini-proj **Smart Parking System** Date:11/12/2024

ID:202242780 Name: Abdulaziz Aldossary

ID:202274280 Name: Yousef Alhadlaq

|  |  |
| --- | --- |
| **1. Objectives:** | |
| * Display parking availability on seven-segment displays and LED indicators for normal and handicapped spaces. * Use a debounce circuit for push buttons and counters to track parking spaces accurately. * Verify the system through simulations to ensure functionality. * Optimize the circuit design for efficient use of resources. | |
|  | |
| **2. Procedure:** | |
| **Task 1: Normal\_parking\_counter**   1. **Open Xilinx ISE Design Suite:** We launched the Xilinx ISE software to start the design process. 2. **Create a New Project:** We sit up a new project in Xilinx ISE and specify the appropriate FPGA model for implementation. 3. **Write the Verilog Module:** We created a Verilog module named Normal\_Parking\_counter with parameters maximum and minimum and input signals clk, reset, entry, and exit. Use registers entry\_prev and exit\_prev for edge detection. 4. **Implement Parking Logic:** We developed logic to increment or decrement the slots register based on the entry and exit signals, ensuring accurate updates. 5. **Test and Simulate:**   We configured clock signal (clk) with a period of 2 ns. We initially forced reset signal to a constant value of 1 and then cancelled after 2 ns. We changed the slots radix to unassigned decimal.   * We activated the clock entry signal starting at 2 ns and continued until 44 ns, with a period of 2 ns. * We activated the clock exit signal starting at 60 ns and continued until 100 ns, also with a period of 2 ns.   **Task 2: Handicapped\_Parking\_counter**   1. **Write the Verilog Module:**  We created a Verilog module named Handicapped\_Parking\_counter with parameters maximum and minimum and input signals clk, reset, entry, and exit. Use registers entry\_prev and exit\_prev for edge detection. 2. **Implement Parking Logic:** We developed logic to increment or decrement the slots register based on the entry and exit signals, ensuring accurate updates. 3. **Test and Simulate:**.   We configured clock signal (clk) with a period of 2 ns. We initially forced reset signal to a constant value of 1 and then cancelled after 2 ns. We changed the slots radix to unassigned decimal.   * We activated the clock entry signal starting at 2 ns and continued until 14 ns, with a period of 2 ns. * We activated the clock exit signal starting at 16 ns and continued until 28 ns, also with a period of 2 ns. | |
| **Task 3: led\_control**   1. **Create a New Verilog Module**: We created a Verilog module named led\_control with inputs clk, slots\_normal, and slots\_handicapped. We defined outputs led\_green\_normal, led\_red\_normal, led\_green\_handicapped, and led\_red\_handicapped to control the LEDs. 2. **Implement LED Logic**: We wrote logic to control the normal parking LEDs. We ensured that led\_green\_normal turns on when slots\_normal > 0 and led\_red\_normal turns off. Conversely, when slots\_normal == 0, led\_red\_normal turns on, and led\_green\_normal turns off. Similar logic was implemented for the handicapped parking LEDs using slots\_handicapped. 3. **Test and Simulate**: we created a test bench module and wrote verilog code to make the slots start at 20 and decrements till the minimum value possible this idea of making a test bench was taken from experiment 5 then we simulate the test bench file and made the slots radix to unassigned decimal.   **Task 4: seven segment display**   1. **seven-segment display was modified by updating the DISP7SEG file, which was provided on the Blackboard . Specifically, we made the following changes:**   **1. The letter H was assigned to segment D7.**  **2. The letter A was assigned to segments D6 and D3.**  **Task 5: debounce**   1. **Create a New Verilog Module:** We created a Verilog module named debounce and downloaded the debounce code provided to us on Blackboard to detect button presses accurately.   **Task 6: Smart Parking system**   1. **Create the Module:** We created a Verilog module names Smart\_Parking\_System that integrates all components of the parking system. The module takes inputs including Sysclk, reset, entry\_normal, exit\_normal, entry\_handicapped, and exit\_handicapped. The outputs include seven-segment display signals seg and an, along with LED indicators for normal and handicapped parking availability. 2. **Debounce Circuits:** We used debounce circuits to clean the input signals for entry\_normal, exit\_normal, entry\_handicapped, and exit\_handicapped. The debounced outputs cleanentry, cleanexit, cleanentry\_ha, and cleanexit\_ha were generated to ensure signal stability. 3. **Integrate Parking Counters:**We instantiated the Normal\_Parking\_counter module to track the available normal parking slots using the cleanentry and cleanexit signals.We instantiated the Handicapped\_Parking\_counter module to manage handicapped parking spaces using cleanentry\_ha and cleanexit\_ha. 4. **Seven-Segment Display:** We connected the outputs of the parking counters to a DISP7SEG module. This module displayed the number of available parking slots for both normal and handicapped spaces on the seven-segment displays. 5. **LED Control:** We added an led\_control module to handle the LED indicators. The module managed the green and red LEDs for normal and handicapped parking, accurately reflecting parking availability based on the slot counters. 6. **Synthesize and Implement:** Synthesize the design and implement it on the FPGA for hardware testing. | |
|  | |
| **3. Problems Faced:**  The counter was skipping intermediate values, jumping directly from 20 to 0 or 0 to 20, because the button press was being detected continuously by the high-speed clock (100 MHz). This caused the counter to increment or decrement too quickly, as it updated on every clock cycle while the button was pressed.  We solved this issue by detecting only the rising edge of the button press. This means the counter only updates once when the button is initially pressed, instead of continuously updating while the button is held. | |
| **4. Work Distribution:** | |
|  | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Names | Task1 | Task2 | Task3 | Task4 | Task5 | Task6 | Lab Report | | Abdulaziz | 0% | 0% | 0% | 100% | 100% | 100% | 50% | | yousef | 100% | 100% | 100% | 0% | 0% | 0% | 50% |   **5. Snapshots:** |  | |  |  | 50% | 45% |
| Figure1:Verilog Model For Noraml\_parking\_counter simulation.      ---------------------------------------------------------------------------------  Figure2: Verilog Model For Handicapped\_parking\_counter simulation.      ---------------------------------------------------------------------------------  Figure3: Verilog Model For led\_control simulation.      ---------------------------------------------------------------------------------  Figure4: Verilog Model For Noraml\_parking\_counter code. |  | |  |  | 50% | 55% |
| ---------------------------------------------------------------------------------  Figure5: Verilog Model For Handicapped\_parking\_counter code.    ---------------------------------------------------------------------------------  Figure6: Verilog Model For led\_control code.    ---------------------------------------------------------------------------------  Figure7: Verilog Model For Debounce code.    ---------------------------------------------------------------------------------  Figure8: Verilog Model For Seven segment display code.    ---------------------------------------------------------------------------------  Figure9: Verilog Model For Smart\_Parking\_System code. |
| **6. Conclusion:** |
| In this project, we successfully implemented a **Smart Parking System** using Verilog on the Nexys-A7 FPGA board. The objectives included tracking parking availability, displaying it on seven-segment displays, controlling LED indicators, and ensuring accurate button detection with debouncing circuits.Through this project, we gained a solid understanding of digital design, Verilog programming, and FPGA resource management. Testing and simulations confirmed the system's functionality, meeting all objectives efficiently. |