**Automated Car Parking System**

**Using Verilog**

**Abhinav C (35, 170907274) and Adithyaa Sunder (46, 170907358)**

Email id:-abhinavcnambiar@gmail.com Email id:- adithyaa98@gmail.com

V Semester, Section-A, Electronics and Communication Department,

Manipal Institute of Technology, Manipal

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Abstract- In this era of modernization and urbanization, there has been a considerable rise in the number of vehicles. Particularly, there has been a significant rise in the usage of four wheelers. Due to this, the problem of parking has turned out to be a pressing issue. This is even more noticeable in public places like malls, theatres, educational institutes, hospitals etc. Limited parking spaces and the ever-increasing count of vehicles has become an issue that needs attention right away. There is a dire need for an effective and efficient system in place to combat this issue. We designed “Automated Car Parking System” as an attempt to tackle this matter.  Keywords- entry, exit, number of cars, password, parking, Verilog, automatic system, FPGA.   1. INTRODUCTION   As the population continues to rise at a rapid pace, so do the number of cars on the road. Vehicle parking has become a real issue and needs to be looked at and solved right away.  According to a recent analysis, an exponential rise in the sale of vehicles has been noted over the last 2 decades.  The number of cars sold worldwide increased tremendously from 39.2 million in 1999 to 77 million in 2019.In the last 20 years, the sales have noted an increase of as high as 40 million units.   1. TOOLS USED 2. Vivado Design Suite by Xilinx:   Vivado enables developers to synthesize a design and simulate the design’s reaction to various stimuli, examine RTL diagrams, perform timing analysis and configure the target device with the programmer. It is mainly a design environment for FPGA products from Xilinx, and cannot be used with FPGA products from other vendors.It was introduced after its predecessor, Xilinx ISE was discontinued.  C:\Users\Admin\Desktop\MC\300px-XilinxVivado_2014.jpg  It is an IDE which includes a common debug environment and electronic system level design tools for synthesizing and verifying C-based algorithmic IP.  We made use of Vivado in order to write our Verilog code and run the simulation.   1. METHODOLOGY   The proposed state diagram of the automated car parking system is shown below.  The system is initially in the “idle” state. It consists of a sensor which detects an incoming vehicle.  Once the sensor is triggered, it moves to the “Wait\_Password” state where a password is then requested in order to open the gate.  If the password is correctly entered, the system moves to the “Right\_Pass” state the gate automatically opens, allowing the vehicle to enter the parking lot.  If the entered password turns out to be incorrect, the system moves to the “Wrong\_Pass” state, where it will stay until the correct password is entered.  If another car happens to be detected by the exit sensor at the same time that a car is being detected by the entry sensor, the system moves to the “Stop” state and the gate is automatically closed. |  | The direct impact of this increase in the production of vehicles has resulted in the congestion on roads. These congestions sometimes even cause accidents. The below figure shows congestion problems in urban areas.  Related image.   1. Artix 7 FPGA kit by Xilinx:   The Xilinx Artix®-7 family of FPGAs has redefined cost-sensitive solutions by cutting power consumption in half from the previous generation while providing advanced functionality for high-performance applications. Designers can leverage twice the logic density for the same power budget. These low-end devices are built on the 28 nm high-performance, low-power (HPL) process to deliver best-in-class performance-per-watt for products like portable medical equipment, military radios, and compact wireless infrastructure. Artix-7 FPGAs meet the needs of size, weight, power, and cost (SWaP-C) sensitive markets like avionics and communications  C:\Users\Admin\Desktop\MC\images (1).jpg  We made use of the Artix-7 FPGA in order to implement our design.    Once the other car has exited, the password is requested again.  To achieve this, a Verilog code is implemented which is further synthesized on a FPGA kit  C:\Users\Admin\Desktop\MC\images (3).jpg   1. CODE DEVELOPMENT   We have defined 4 inputs:   * Sensor\_entrance * Sensor\_exit * Clock * Reset   Following this, the state variables are assigned accordingly.   |  |  | | --- | --- | | States | Code | | Idle | 3’b000 | | Wait\_Password | 3’b001 | | Right\_Pass | 3’b010 | | Wrong\_Pass | 3’b011 | | Stop | 3’b100 |   Apart from these, a few register variables have been defined.   * Current\_State * Next\_State * Counter\_Wait   The values for these registers are defined based on the values of the inputs “Sensor\_Entrance” and “Sensor\_Exit”. |
| These values are compared at the positive edge of clock, and negative reset.  Lights of different colors are used to indicate if the password has been accepted or rejected.   |  |  | | --- | --- | | Colors | Password | | Green | **Accepted** | | Red | **Rejected** |   We used Vivado Design Suite to develop our program for the Carry Look Ahead Adder module.  Module Inputs:   * X (32-Bit) * Y (32-Bit) * Cin   Module Outputs:   * Sum (32-Bit) * Cout  1. SIMULATION   Initial test input-  reset\_n = 0;  sensor\_entrance **=** 0;  sensor\_exit = 0;  password\_1 = 0;  password\_2 = 0;  After a delay of 100 units to wait for reset to finish ;  reset\_n = 1;  After a delay of 20 units;  sensor\_entrance = 1;  After a delay of 100 units;  sensor\_entrance = 0;  password\_1 = 1;  password\_2 = 2;  After a delay of 2000 units**;**  sensor\_exit =1; |  | verilog code for car parking system   1. CONCLUSION   We were successful in implementing an automated car parking system which combats the issue of parking, an issue that has been prevailing for quite some time and an issue that has become quite the menace owing to population growth and limited parking spots.  We hope to keep working on this and improving the system’s efficiency. With advances in technology, the system can be further updated to include a plethora of features in the future.   1. REFERENCES   <https://www.fpga4student.com/2016/11/verilog-code-for-parking-system-using.html>  <https://www.xilinx.com/products/design-tools/vivado.html>  <https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html>  <https://www.irjet.net/archives/V5/i7/IRJET-V5I7257.pdf> |