REVIEW PAPER
International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009

FPGA Based Autonomous Vehicle Locking System- A Smart Door Lock

[1]Dr. S .N. Singh, [2]Jayendra Kumar, [3]Ravi Pratap Singh and [4]Sanjay Kumar
[1,2,3,4]Department of Electronics and Communication Engineering, National Institute of Technology, Jamshedpur, India
Email: snsnitjsr@gmail.com, jaynitjsr@rediffmail.com, singh.ravipratap88@gmail.com, san.elnkumar@gmail.com

***Abstract*–Our paper is termed the Autonomous Vehicle Locking System – A Smart Door Lock. The two main objectives are to unlock a set of vehicle doors when the owner approaches his or her vehicle and to lock the vehicle after the owner leaves. In order to design this system, the establishment of wireless communication between the vehicle owner and their vehicle is required. The wireless transmission must ensure vehicle security and also be reliable in various climates. Overall, the total power dissipated needs to be low to account that it will be on for long periods to time.**

This project was selected because this was a functionality that everybody would like to see implemented. This system is intended to provide added convenience to the vehicle owner.

The object of this paper is to implement a low-power system where the customer will be granted access to his locked car as he approaches the vehicle. With such a system, the advantages of keyless access will be accomplished with only passive involvement from the client.

***Index Terms*—Field Programmable Gate Arrays (FPGAs), Very High Speed Integrated Circuits Hardware Description Language (VHDL), Register, Comparator, Controller, Processor.**

I. Introduction

The Autonomous Vehicle Locking System is designed to improve vehicle security and accessibility. With the use of wireless technology vehicle owners are able to enter as well as protect their automobiles with more passive involvement. Originally, vehicles were accessed and secured manually by inserting a key into a lock. More recently, a keyless entry system was created which allowed the owner to lock and unlock their vehicle with the touch of a button. Communication between the owner and the vehicle was established using wireless technology. With the Autonomous Vehicle Locking System, wireless communication expands upon past technology. It is able to grant access and security without pushing a button. For this new technology to work, the vehicle's internal computer must be utilized in conjunction with sensors that can detect the owner's presence, as well as the status of the locks. Finally, the circuit design needs to turn the locking mechanism in the lock or unlock position. The fields used are RF communication, digital systems software programming, logic circuit design, as well as microelectronic design.

A. Objective

The intent of this project is to implement an autonomous vehicle door locking system that will unlock the vehicle doors when the vehicle owner approaches their automobile and to lock the doors when the vehicle needs to be secured. An autonomous system will keep the vehicle owner from having to lock and unlock their doors manually with a key or a push-button entry system.

Product Benefits:

1) Owner will automatically be able to enter car upon arrival

2) Unique owner-specific identification code will ensure vehicle security

B. Performance Specifications

* Unlock doors upon vehicle owner's arrival
* Lock doors upon vehicle owner's departure
* Unique owner-specific identification code to ensure vehicle security
* Sensing to deactivate the system while owner is in the vehicle
* Sensing to determine when the vehicle doors are unlocked or locked
*Low-power overall system

II. Block Diagram

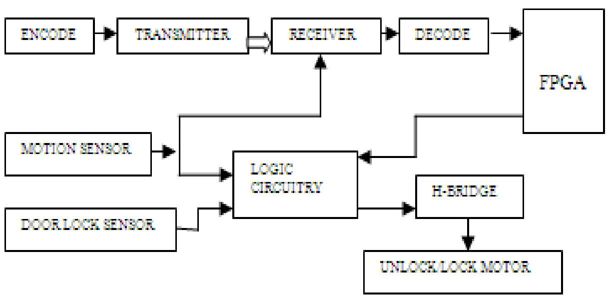

Fig 1. Final block diagram

A. Block Diagram Descriptions:

i) Vehicle Lock Sensor: Sensor to detect whether door is currently locked or unlocked. This information will be provided to the Logic Circuitry to determine whether the Unlock/Lock Motor should be in operation or not. **ii)** Motion Sensor: Another sensor to detect whether the vehicle owner is inside their car. It provides a signal to

Logic Circuitry and also Receiver to power down the system as shown in Fig. [1].

iii) Encoder: Device that feds a data string of logic 1's and 0's to the transmitter.

iv) Transmitter: Transmitter the user would carry that would transmit a code to the Receiver as shown in Fig.[1].

v) Receiver: Receiver that receives the code from the transmitter and sends it to Decoder, shown in Fig. [1].

vi) Decoder: Decodes the data from the Receiver and sends it to the FPGA as shown in Fig. [1]

vii)FPGA: Microcontroller which utilizes data from Decoder to control operation of Unlock/Lock Motor as shown in Fig. [1]. This unit would be a part of the internal computer that most modern vehicles already possess.

viii) H-Bridge: A circuit that controls the operation of the motor depending on the inputs it receives from the Logic Circuitry as shown in Fig. [1].

ix) Unlock/Lock Motor: DC powered motor that can spin in two directions, clockwise and counterclockwise. The motor will spin in one direction to simulate the car doors locking and the other direction to simulate the car doors unlocking.

III. Software Design

Software is needed to compare the master security combination to the receiver combination. . The software will output a high or a low signal depending on whether or not the receiver combination is correct. The assembly programming language VHDL was chosen ..

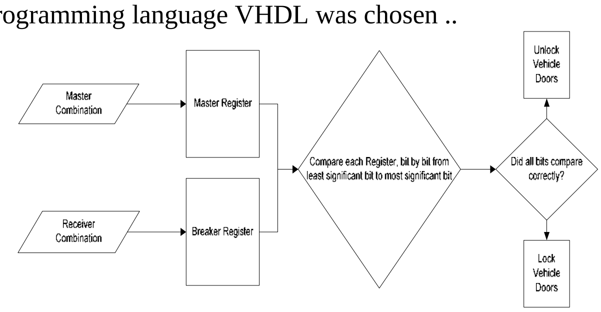

Fig 2. Software Flow Chart

A. Register

First, two registers need to be created. One register takes in 8 bits from the master security combination and the other register takes in 8 bits from the receiver combination. The Master Register takes in the master combination and the Breaker Register takes in the receiver combination. The combinations are loaded into the registers when the Load signal is equal to 1. When Load does not equal 1, then the same information that was loaded the last time Load was equal to 1 will stay into the register. Finally, the 8 bit string of information in the registers will be outputted to designated signals. These signals will be inputs to the comparator.

B. Comparator

The comparator takes in the register signals from the Master Register and the Breaker Register. But it will only take in these signals when the Enable signal is high. This signal will tell the comparator to compare the two registers. If these two registers compare correctly bit by bit from most significant to least significant then a 1 will be assigned to the Unlock signal. If Reset signal is equal to 1 then the Unlock signal will become 0. The comparator code is placed in the attached appendix.

C. Controller

This component of the software assigns 0 or 1 values to the internal signals between the registers and the comparator depending on what state the controller is in. The controller has four inputs and four outputs. The four inputs control what state the machine is in and it also controls values all outputs possess. Two inputs are used to load the Master Register and the Breaker Register. The reset input will put the software in state A. The last input is the clock input. In order to advance states, the internal oscillator clock signal must be on the rising edge. Overall, seven states are used. Each state waits for a certain condition before advancing to the next state. The first two states are used to load the Master Register. These states also reset the comparator. The next four states are used to load the Breaker Register and also enable and reset the comparator. The comparator needs to be reset every time new information is loaded from the Breaker Register. Then the comparator will be enabled to compare the two registers. The last state will be used to make the decision of either returning to the first two states to put new information into the Master Register or to return to the third state to go through the process of loading new Breaker Register information again. This state allows the continuous loop of loading new information from different transmitter information. The state diagram for the controller is shown in Fig. [3].

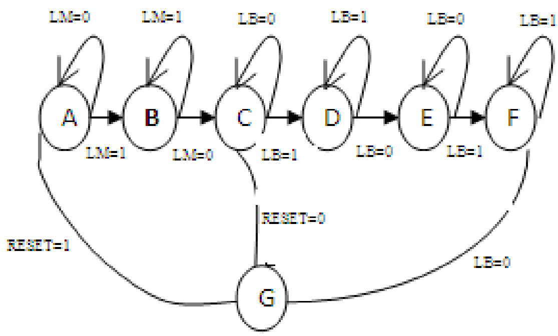

Figure 3: State Machine for Controller

LM = Load_Master LB = Load_Breaker

TABLE 1: CONTROLLER OUTPUT SIGNALS

Signal	State A	State B	State C	State D	State E	State F	State G
Enable_Breaker	0	0	0	1	0	1	0
Enable_Master	0	1	0	0	0	0	0
Reset_Compare	1	0	0	1	0	1	0
Enable_Compar	0	0	0	0	1	0	1

D. Processor

The processor defines and designates all outside and internal input and output signals for the program. All signals were defined according to their operation. The processor assigns the inputs and outputs of the registers. It does the same for the comparator. The controller has four inputs that control four outputs going to the two registers and the comparator with each signal being routed by the processor.

E. Software Used : Xilinx ISE 10.1

Software formed to help designers create successful VHDL code. The software synthesizes code, creates a pin assignment file, constructs a test bench for simulation, and generates a Programming File to upload the code to the FPGA.

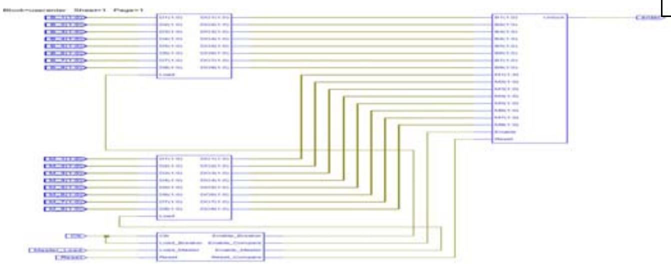

Fig 4. RTL Schematics

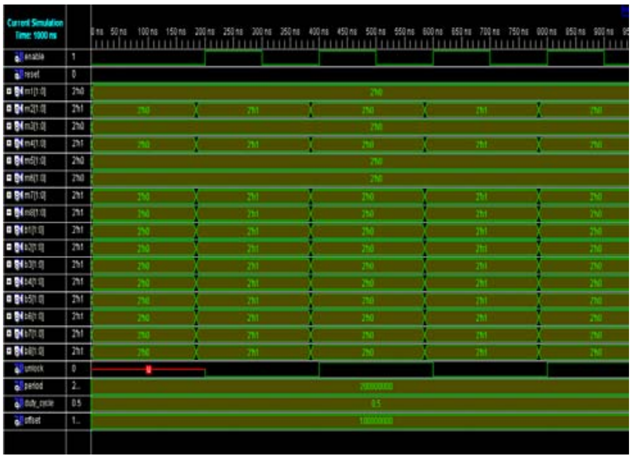

Figure 5 : Simulation of software program

IV. Logic Circuitry

The logic circuitry was designed knowing how the H-Bridge controlled the functionality of the motor as shown in Table [2]. One terminal of the motor is connected to the drains of the MOSFET's labeled A1 and B1. The other terminal is connected to the drains of A2 and B2.The Motion Sensor determines when the vehicle owner is sitting in the driver's seat. The Motion Sensor is 0 when the owner is out of the vehicle and 1 when the owner is in the vehicle. The Vehicle Lock Sensor senses when the lock/unlock motor is in a certain position. The Vehicle Lock Sensor is 0 when the vehicle is unlocked and 1 when it is locked. In final design, though, sensors will be needed in order for the system to be autonomous. The FPGA output will be 1 when the master combination is compared correctly to the receiver combination and will be 0 when they do not. The truth table was formed and was analyzed. From the truth table, it was determined that B1 is the complement of A2. Also, B2 is the complement of A1. Thus, using Boolean algebra, Eqs. 1 – 4 were established.

B1= (Mot.Sen.)'(Veh.LockSen.)(FPGA) , (1)

B2= (Mot.Sen.)'(Veh.LockSen.)'(FPGA)' , (2)

A1= (B1)' , (3)

A2= (B2)' . (4)

TABLE 2: TRUTH TABLE FOR THE FUNCTIONALITY OF UNLOCK/LOCK MOTOR

Mot. Sen.	Veh.Lock Sen.	FPGA	A1	B1	A2	B2	Functionality of motor
0	0	0	0	0	1	1	Spin CW/Lock
0	0	1	1	0	1	0	Do Not Spin
0	1	0	1	0	1	0	Do Not Spin
0	1	1	1	1	0	0	Spin CCW/Unlock
1	0	0	1	0	1	0	Do Not Spin

V. Testing

The software code is to be compiled ,debugged and tested.The VHDL code written was to be synthesized.The software Xilinix ISE 10.1 was used and the implementation was done on SPARTAN - 3E FPGA starter kit board[product by Texas Instruments].

Tolerance analysis : A critical consideration in this system is the temperature and humidity in the car environment. Although cool, dry conditions should not exhibit any negative effects on the received signal, high temperatures and humidity would decrease the SNR of the received signal necessitating a higher power signal from the transmitter. SNR measurements will be performed at increasingly higher humidity and temperature levels to determine an appropriate power for the generated signal.

VI. Conclusions And Future Work

An FPGA system implementation was presented.The VHDL code was simulated and synthesized successfully. There are two key successes of the design which accomplish commercial viability. The swift and reliable transmission of the code from the user to vehicle ensures proper timing and dependability. Consistent transmission is necessary because the vehicle doors must unlock rapidly as the owner approaches the vehicle. Ideally, this project could be made more convenient and secure.The use of sensors rather than switches is very important in making the system autonomous. Using more bits in the security and receiver combination would make the design more secure.

References

[1] A. S. Sedra and K. C. Smith, "Microelectronic Circuits". New York: Oxford University Press, 5th Edition,2006

[2] Volnei A. Pedroni,"Circuit Design with VHDL" ,New Delhi :Prentice-Hall of India,2005.

[3] Nazeih M. Botros, "HDL PROGRAMMING VHDL and Verilog ". New Delhi: Dreamtech Press, Edition 2006

[4] http://www.freepatentsonline.com/5647234.html

[5] Michael Laurence Peterson, Carlos Pacheco, "Autonomous Vehicle Locking System" ECE 345, Senior Design Project, Illinois University, Fall-2004.

[6] "Remote Keyless Entry and Convenience Center Reference Design with LIN Bus Interface" www.jimfranklin.info/microchipdatasheets/00728a.pdf

[7] "Xilinx UG230 *Spartan-3E* FPGA Starter Kit Board *User Guide*"www.xilinx.com/support/documentation/ boards...**kit**s/uug230.pd