Design of a Keyless Coded Home Lock System Using Verilog Hardware Description Language

Kim Ho Yeap[1], Humaira Nisar[1], Kok Hen Chong[2], Zairi Ismael Rizman[3], Chuan Choong Yang[1], Yun Thung Yong[1], Koon Chun Lai[1]

[1] *Faculty of Engineering and Green Technology, Tunku Abdul Rahman University, Jln. Universiti, Bandar Barat, 31900 Kampar, Perak. Malaysia.*

[2] *College of Engineering, Universiti Tenaga Nasional, Km 7, Jln. Kajang-Puchong, 43009 Kajang, Selangor, Malaysia.*

[3] *Faculty of Electrical Engineering, MARA University of Technology, Terengganu Branch, Dungun Campus, 23000 Dungun, Terengganu. Malaysia.*

Email (yeapkh@utar.edu.my)

Abstract: This paper presents the design of a keyless coded home lock system using Verilog HDL. The system allows a house owner to enter a numeric combination code on a push-button keypad. The door of the house will only unlock if the code entered matches the code setting at the setup panel; otherwise the push-button keypad will be disabled after 3 consecutive unsuccessful attempts. The system comprises 5 modules – that is, the keypad and encoder module, setup panel module, sleep circuit module, control unit module, and code checker module. The Verilog code has been successfully written and simulated using Mentor Graphics ModelSim HDL Simulator.

Keywords – Keyless coded home lock system, keypad, encoder, setup panel, sleep circuit, control unit, code checker

I. INTRODUCTION

A keyless coded home lock system is basically a security feature which only allows the house owner to unlock the door if the code entered, via a push-button keypad, is compatible with the code setting at the setup panel.

The objective of the system is to enhance the efficiency of a house security feature. Currently doors installed in houses are unlocked by inserting the appropriate house keys. This feature is deemed unsafe as break-in can easily occur if duplication of the keys is made. The new system introduced here eliminates the possibility of houses being broken in. This can be done by making the code and the code length variable. Thus, one is required to know the exact code and code length at the same time if one wishes to gain access into the house. A further deterrent to break-in cases is to disable the keypad push-buttons after 3 unsuccessful attempts.

The idea of the keyless coded home lock system is inspired from the car lock system discussed in [1]. It is however worthwhile noting that, the system illustrated in [1] is suggested to be built using discrete components. A system built using discrete components could be inconveniently bulky. Furthermore, it is prone to become defective as well, since failures in any of the components or wires will consequently result in malfunction to the entire system. Indeed, troubleshooting might be time consuming in such a system especially when one is uncertain which exact component has broken down. Therefore, there is no other better alternatives but to debug the entire system once failure occurs. Such a debugging process might involve each and every of the component, resulting in laborious and tedious effort to repair the system.

Ever since the advent of the VLSI technology [2] – [4], Application Specific Integrated Circuit (ASIC) devices has been made possible using the simulation of hardware description language (HDL), such as Verilog [5] – [10] and VHDL [11] – [13]. By applying ASIC in home lock system, the entire system can be made so conveniently small that all the user is required to do is to replace the chip with a new one should failures occur. In this paper, we present the design of the home lock system using Verilog HDL. As discussed in the results and discussion section later, the system can be fabricated into one single chip, which consists of 5 modules – i.e. the keypad and encoder module, setup panel module, sleep circuit module, control unit module, and code checker module.

II. PROCESS FLOW

The operational procedure of the keyless coded home lock system is depicted in Fig. 1. In order to unlock the door, the user is required to enter certain code via the keypad module first. The code checker module will then compare the code length entered with the one in the setup panel module. If the length matches, the system will proceed to the next stage; otherwise, it will remain in a pending situation, until the user has entered the correct length.

At the second stage, a comparison between the digits entered with the digits stored in the setup panel is performed. In this case, 2 possibilities may occur – that is, the current digits being compared are compatible or otherwise.

If both digits are compatible, the code checker will then proceed to comparing the subsequent digits. A pointer is implemented to keep track of the digit at the keypad module which is being compared at that moment. The process continues until the last digits are compared.

The door will be unlocked and the entire system will be reset as soon as all the digits entered are found valid.

On the other hand, if the digits are incompatible, a check will then be performed by the code checker to determine the number of unsuccessful attempts. If the number of attempts exceeds 3, the system will be disabled for 3min, during which the keypad will be disabled. The system will be reset (with the number of unsuccessful count returns to 0) after 3min.

If the number of attempts is less than 3, the system will stay in a pending situation, waiting for the user to press the "reset" button at the keypad. As soon as the "reset" signal is received, the system will be reset (with the number of unsuccessful count incremented by 1).

III. ARCHITECTURE

A block diagram of the keyless coded home lock system is shown in Fig. 2 [1]. As mentioned in section I, the system consists of 5 interdependent modules – they are, the keypad and encoder module, setup panel module, sleep circuit module, control unit module, and the code checker module. Discussion on each module is performed subsequently.

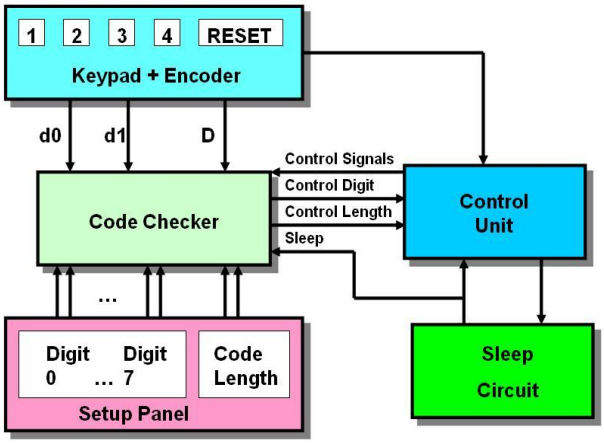

Figure 2. Schematic block diagram of a keyless coded home lock system.

A. *Keypad and Encoder*

The keypad comprises 4 push buttons, which represents digit 1, 2, 3, and 4, and a reset button. The encoder converts any of the digits into a 2-bit binary code. It comprises 3 output pins – d0, d1, and D. When 1 of the 4 digit buttons is pressed, output D will go high and a 2-bit binary code (d0, d1) will be generated to represent the digit. The encoder circuit is shown in Fig. 3 [1]; whereas,

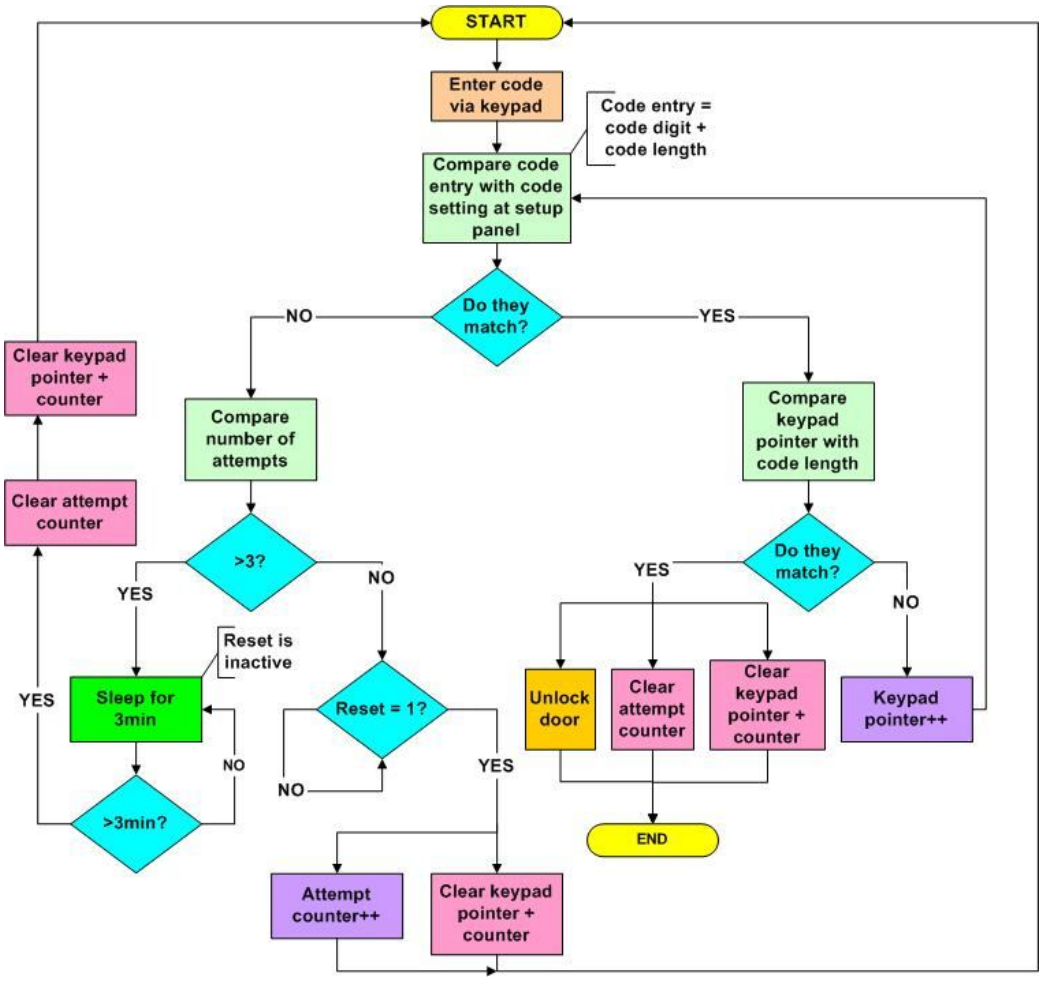

Figure 1. Flowchart of how a keyless coded home lock system operates.

the decimal digit values represented by each push button and their respective 2-bit encoded value is shown in Table 1 [1].

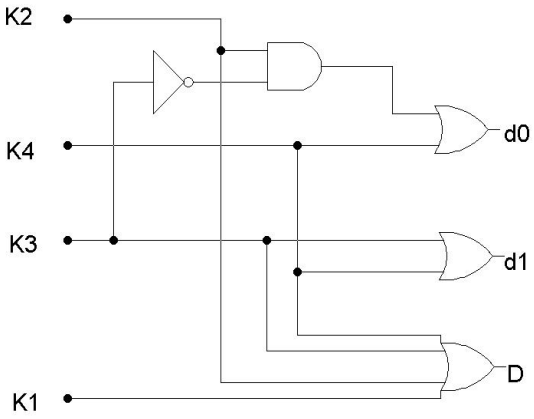

Figure 3. Encoder circuitry.

Table 1. 2-bit code and digit representation of the keypad input

Input	d0	d1	Digit
K1	0	0	4
K2	1	0	2
K3	0	1	3
K4	1	1	1

B. *Setup Panel*

A graphical representation of the setup panel is depicted in Fig. 4 [1]. The setup panel allows the user to specify the code and code length setting. The code can be made up of any of the 1, 2, 3, or 4 digit numbers. The digit numbers are represented by 4 different combinations of 2 switches, as shown in Table 2. The code length can have a minimum of 4 and maximum of 7 digits long. The length is specified in accordance with the different combination of the 2 switches at the setup panel, as shown in Table 3 [1].

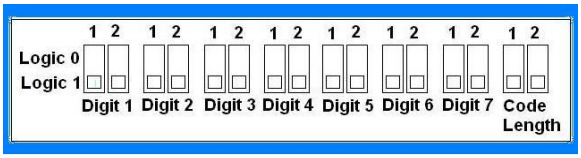

Figure 4. Setup panel.

Table 2. Digit representation using 2 switches

Switch 1	Switch 2	Digit
0	0	4
1	0	2
0	1	3
1	1	1

Table 3. Code length representation using 2 switches

Switch 1	Switch 2	Code Length
1	1	4
1	0	5
0	1	6
0	0	7

C. *Control Unit*

Control unit plays an essential role in the system as it provides timing and control signals for all modules. It dictates which module is to operate at a particular time unit. It also clears all registers when a "reset" signal is received and send an "unlock" signal when the code entered is valid.

D. *Code Checker*

The 2 core functions of a code checker are to – (i) Compare the code length and the digits in the code entered via a keypad with the code setting at the setup panel. (ii) Check if the number of unsuccessful attempts exceeds 3.

E. *Sleep Circuit*

After 3 consecutive unsuccessful attempts made, the sleep circuit will be activated (triggered with an active low signal). The circuit allows the entire system to enter "sleep mode", during which any attempts of pressing the keypad will be waived. The system will be reset and the number of unsuccessful try will be cleared after the 3min "sleep mode" elapsed.

IV. Simulation Results and Discussion

The Verilog codes have been compiled and simulated successfully using Mentor Graphics ModelSim HDL simulator. The simulation results for the keypad and encoder module and the setup panel module will first be discussed. This is then followed by a discussion on the simulation results for the entire keyless coded home lock system.

A. *Keypad and Encoder*

The encoder circuitry shown in Fig. 3 is converted into Verilog code as can be seen in Listing 1.

```
assign  ANDK2K3 = regK2 & INVK3;
assign  INVK3 = ~regK3;
assign d0 = ANDK2K3 | regK4;
assign d1 = regK3 | regK4;
assign D = regK1 | regK2 | regK3 | regK4;
```

Listing 1. Verilog code for an encoder circuitry.

The inputs entered by the user via K1 to K4 keypad buttons are first saved into their respective registers (namely, regK1, regK2, regK3, regK4). Logical operations are performed on each input values and their respective output pins are assigned to d0, d1, and D output registers.

Listing 2 shows the mechanism of storing each output values into a register in correspond with their output sequence.

```
kp_counter = kp_counter + 1;
case(kp_counter)
3'b001   :        #0.5 code1 = {d0,d1};
3'b010   :        #0.5 code2 = {d0,d1};
3'b011   :        #0.5 code3 = {d0,d1};
3'b100   :        #0.5 code4 = {d0,d1};
3'b101   :        #0.5 code5 = {d0,d1};
3'b110   :        #0.5 code6 = {d0,d1};
3'b111   :        #0.5 code7 = {d0,d1};
default  :        $display("INVALID CONTROL
SIGNALS");
endcase
```

Listing 2. Mechanism for storing input values.

Each time the user presses an input button, the kp_counter will be incremented. This allows the encoded values to be saved into code1 to code7 registers in their respective order.

The codes are tested on a testbench and part of the output waveforms are displayed in Fig. 5. As can be seen from the circled portion, when the user presses the K3 input button (logic 1), the encoder output pins (d0, d1) gives a 01 output signal after a 0.5 time unit delay. The output values (01) are being saved into a code2 register to denote the second digit being pressed.

B. *Setup Panel*

The conversion from the switches signals to code length in Table 3 is represented by Verilog codes as shown in Listing 3 below. B7 and B8 represent the input pins of the 2 setup panel switches; whereas, sp_counter register stores the code length.

```
case({B7,B8})
2'b00    :        sp_counter <= 3'b111;
2'b01    :        sp_counter <= 3'b110;
2'b10    :        sp_counter <= 3'b101;
2'b11    :        sp_counter <= 3'b100;
default  :        $display("INVALID SIGNALS");
endcase
```

Listing 3. Conversion from switches signals to code length.

A selected part of the output waveforms from a testbench is displayed in Fig. 6. The sp_counter which is initially set to 6 (that is, 110 in binary representation), is later set to 4 (that is, 100 in binary representation). As can be seen from the circled portion of the waveforms, when the code length is set to 4, only 4 of the internal registers are allowed to store the code determined by the switches in the setup panel; whereas the remaining 2 registers will be assigned unknown values (represented by red lines).

C. *Keyless coded home lock system*

As soon as all modules have been compiled and simulated successfully, a top module, which is the keyless coded home lock system module, is used to instantiate all modules. The pseudocode for the top module is shown in Listing 4.

```
Keyless coded home lock system module;
{
        Code Checker module;
        Control Unit module;
        Sleep circuit module;
        Keypad and Encoder module;
        Setup Panel module
}
```

Listing 4. Pseudocode for the Keyless coded home lock system module.

Three testbenches were written to validate the performance of the entire system in 3 different situations. The first testbench tests the functionality of the system when both the code setting at the setup panel and the code entered by the user match each other. A list of selected output waveforms is displayed in Fig. 7. Notice that the values stored in the keypad registers – code1 to code4 is equivalent with the values stored in the setup panel registers – sp_code1 to sp_code4. That is, the binary digits stored in both modules are 11, 10, 01, and 00 respectively. The code checker performs a check on the code in both modules. A C output pin in the checker module will transmit a pulse when the digit compared matches each other. As shown in Fig. 7, the C output pin sent out 4 consecutive pulses. At the positive edge of the fourth pulse, L, UNLK, CLRT, and CLRD output pin were activated (as shown by the yellow line) and a pulse from each pin were transmitted. L goes high when the code length has been compared and found valid. Logic 1 in the UNLK pin dictated the door to be unlocked;

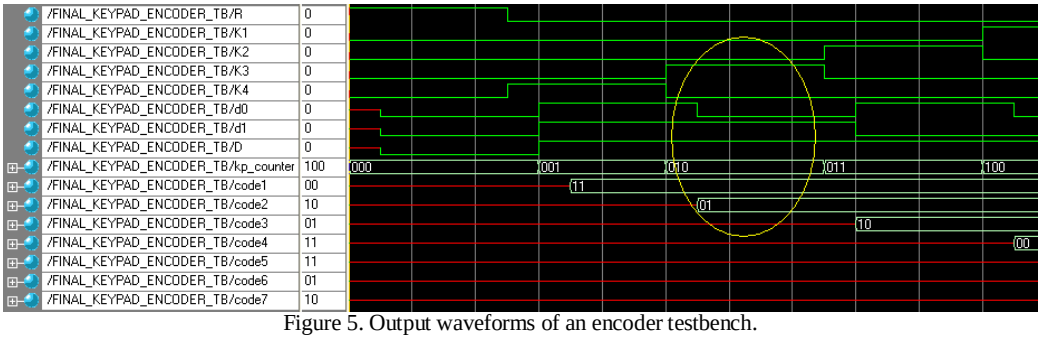

Figure 5. Output waveforms of an encoder testbench.

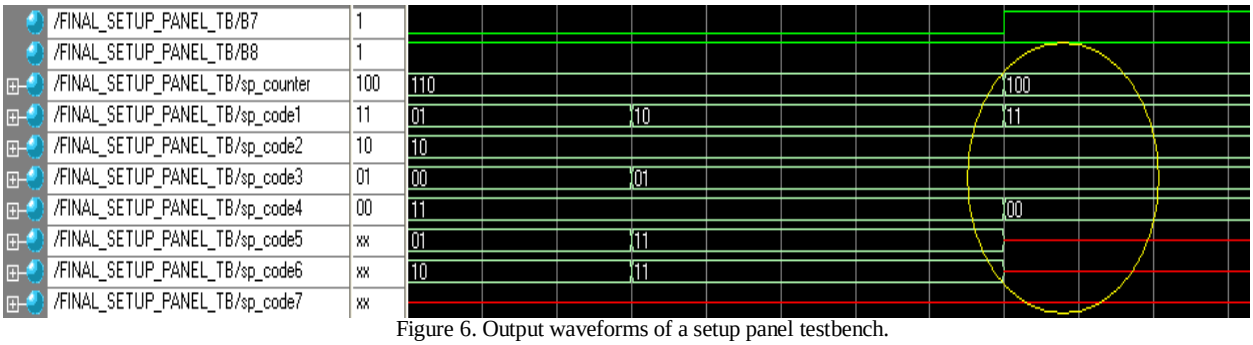

Figure 6. Output waveforms of a setup panel testbench.

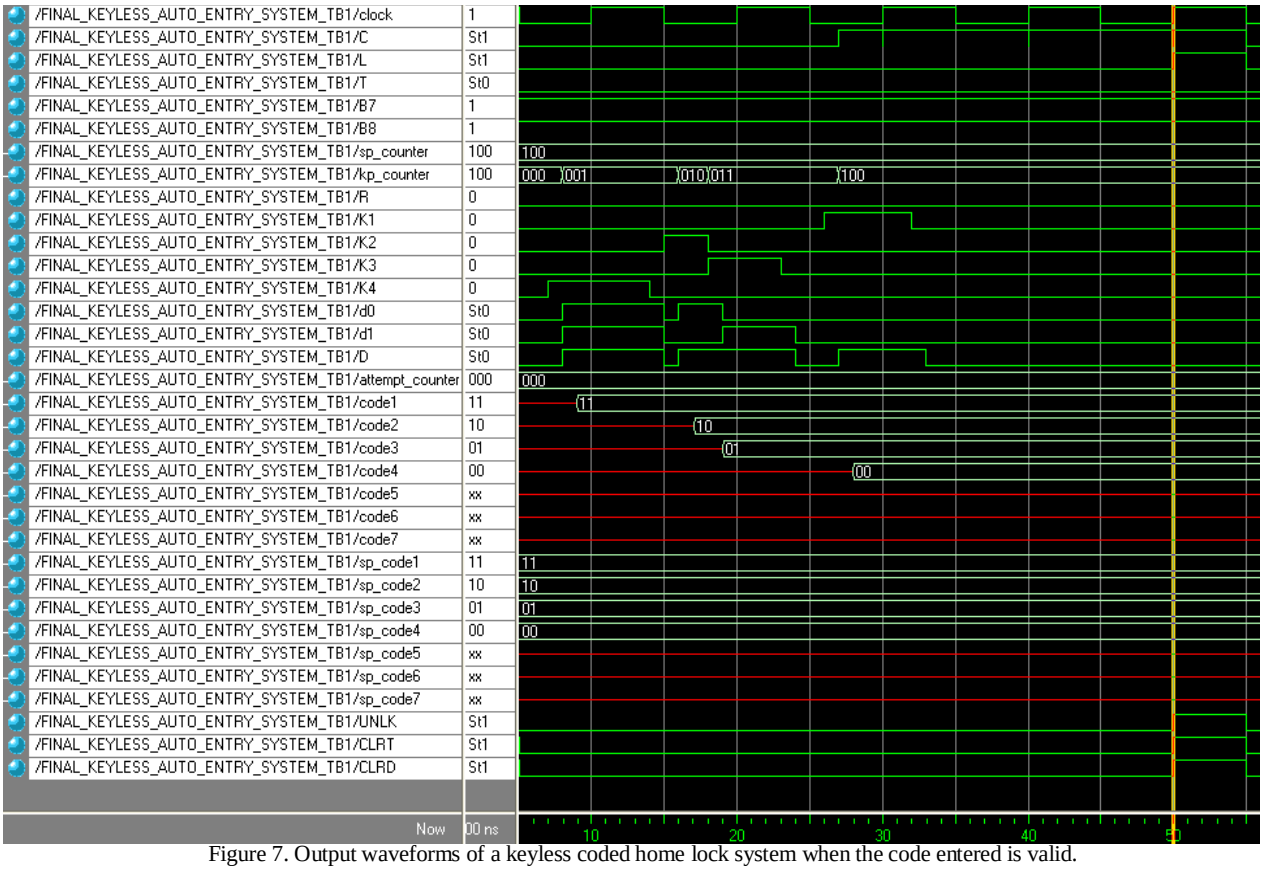

Figure 7. Output waveforms of a keyless coded home lock system when the code entered is valid.

whereas, logic 1 in the CLRT and CLRD pin clear the number of unsuccessful count in the attempt_counter register and reset keypad pointer back to 1 respectively.

The second testbench tests the functionality of the system when the code length entered is invalid. The selected output waveforms are displayed in Fig. 8. As soon as the D output pin from the encoder circuit sent a signal to denote that the user has keyed in the fourth digit – that is, the code length entered is now compatible with the code length specified by the setup panel, the code checker starts to compare the code digit by digit. While comparing the third digit (after C has transmitted 2 pulses), the user keyed in the fifth digit (at the 5ns interval confined by the 2 yellow vertical lines). At this point, the code checker detected a mismatch in the code length and stopped operation. The system stayed in a pending position until a reset signal is received.

The third testbench tests the functionality of the system when the code entered is incorrect and the user was attempting for the fourth times. As can be seen in Fig. 9, the digits stored in code1 to code4 are 11, 10, 01, 01 respectively; However, the digits stored in sp_code1 to sp_code4 are 11, 10, 01, 00 respectively. A mismatch occurs in between the 2 codes. The user attempted to reset the system by pressing the "reset" button for 4 times. R input pins sent out 4 pulses and the attempt_count, which keep track of the number of unsuccessful attempt, incremented to 4. The user then managed to key in the correct code. However, at this point, T transmitted logic 1 to indicate that there has been 3 unsuccessful previous attempts. The sleep circuit is activated (an active low at the

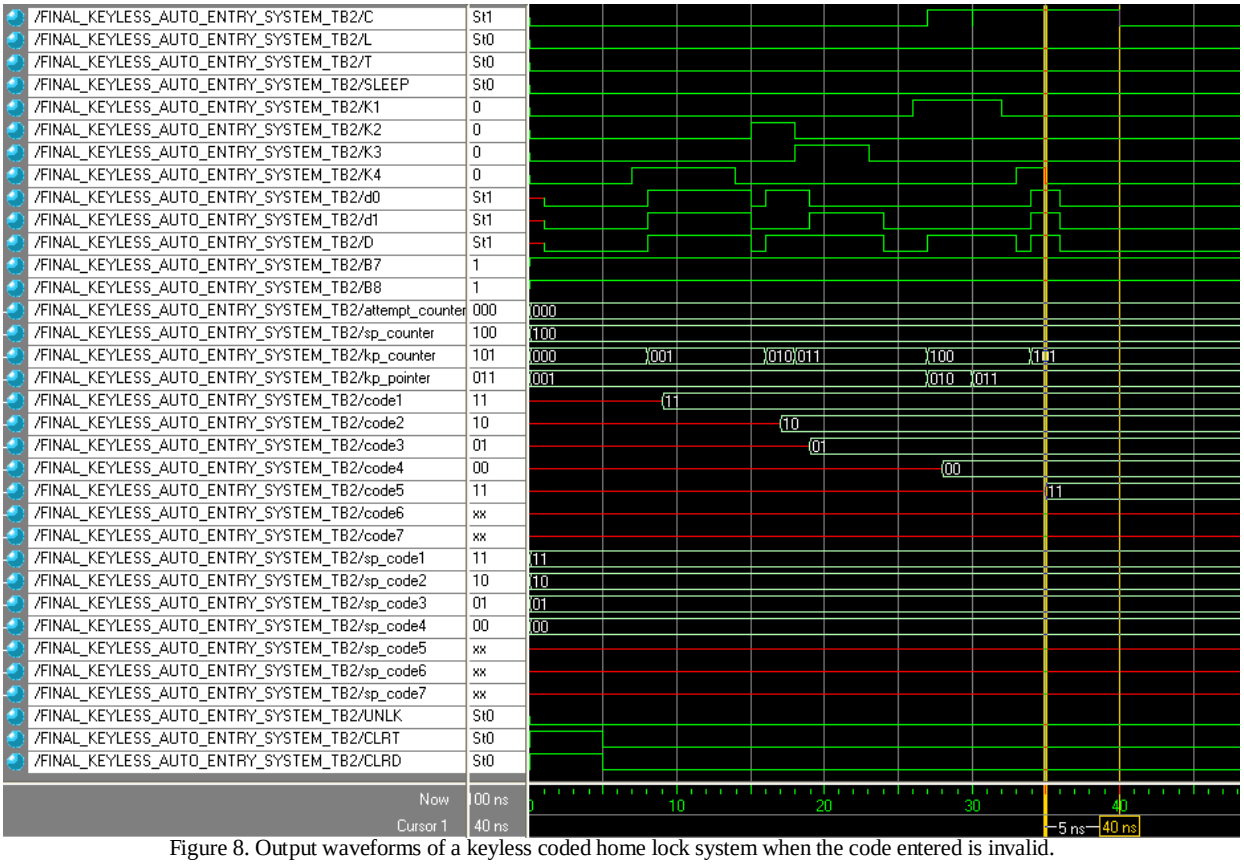

Figure 8. Output waveforms of a keyless coded home lock system when the code entered is invalid.

INVSLP pin), and the SLEEP output pin sent out a positive signal for a certain duration of time (as shown by the yellow line). During this interval, any attempt from the keypad will be ignored.

V. CONCLUSIONS

All Verilog output waveforms simulated and displayed by the ModelSim tool show that the keyless coded home lock system is operating accordingly. Hence, we can conclude that the design of a keyless coded home lock system using Verilog HDL is successful and is feasible to be implemented.

ACKNOWLEDGEMENT

We thank V. V. Yap and Kenneth Wong for their discussion and suggestions.

REFERENCES

[1] V. P. Nelson, H. T. Nagle, B. D. Carrol, and J. D. Irwin, Digital Logic Circuit Analysis and Design, 1st Edition, Prentice Hall, 1995.

[2] I. Ahmad, Y. K. Ho, and B. Y. Majlis, "Fabrication and characterization of a 0.14 µm CMOS device using ATHENA and ATLAS simulators," Int. Scientific J. of Semiconductor Physics, Quantum Electronics, and Optoelectronics, vol. 9, pp. 40 – 44, 2006.

[3] M. S. Sulong, I. Ahmad, L. T. Foo, and Y. K. Ho, "Characterization of a 130 nm CMOS device using CVIV and focused ion beam," Jurnal Sains dan Teknologi, vol. 4, pp. 37 – 46, 2007.

[4] K. H. Yeap, I. Ahmad, Z. I. Rizman, K. Chew, K. H. Chong, and Y. T. Yong, "Characterization of a submicron PMOS in mixer circuits," Proc. of the IEEE Conf. on Sustainable Utilization and Development in Engineering and Technology, 20 – 21 November, 2010, Petaling Jaya, Malaysia, pp. 129 – 132.

[5] C. C. McAndrew, Z. Huszka, and G. J. Coram, "Bipolar transistor excess phase modeling in Verilog-A," IEEE J. of Solid-State Circuits, vol. 44, pp. 2267 – 2275, 2009.

[6] G. M. Blair, "Verilog: Accelerating digital design," Electronics and Communication Engineering J., vol. 9, pp. 69 – 72, 1997.

[7] S. Palnitkar, Verilog HDL – A Guide to Digital Design and Synthesis, 2nd Edition, Prentice Hall, 2003.

[8] Y. Ye and S. Cheng, "Implementation of 2D-DCT Based on FPGA with Verilog HDL," Lecture Notes in Electrical Engineering, vol. 97, pp. 633 – 639, 2011.

[9] C. Sharma, "Designing of Four Port Controlled Switch Using Verilog," J. of Global Research in Computer Science, vol. 2, pp. 34 – 36, 2011.

[10] K. Aditya, M. Sivakumar, F. Noorbasha, T. P. Blessington, "Design and Functional Verification of A SPI Master Slave Core Using System Verilog," Int. J. of Soft Computing and Engineering, vol. 2, pp. 558 – 563, 2012.

[11] S. G. Shiva and J. U. Jones, "VHDL-based expert system for hardware synthesis – An overview," Computing and Control Engineering J., vol. 4, pp. 85 – 91, 1993.

[12] J. M. Chang and S. K. Agun, "Design-for-reusability in VHDL," Computing and Control Engineering J., vol. 12, pp. 231 – 239, 2001.

[13] A. Depari, P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "A VHDL model of a IEEE1451.2 smart sensor: Characterizations and applications," IEEE Sensors J., vol. 7, pp. 619 – 626, 2007.

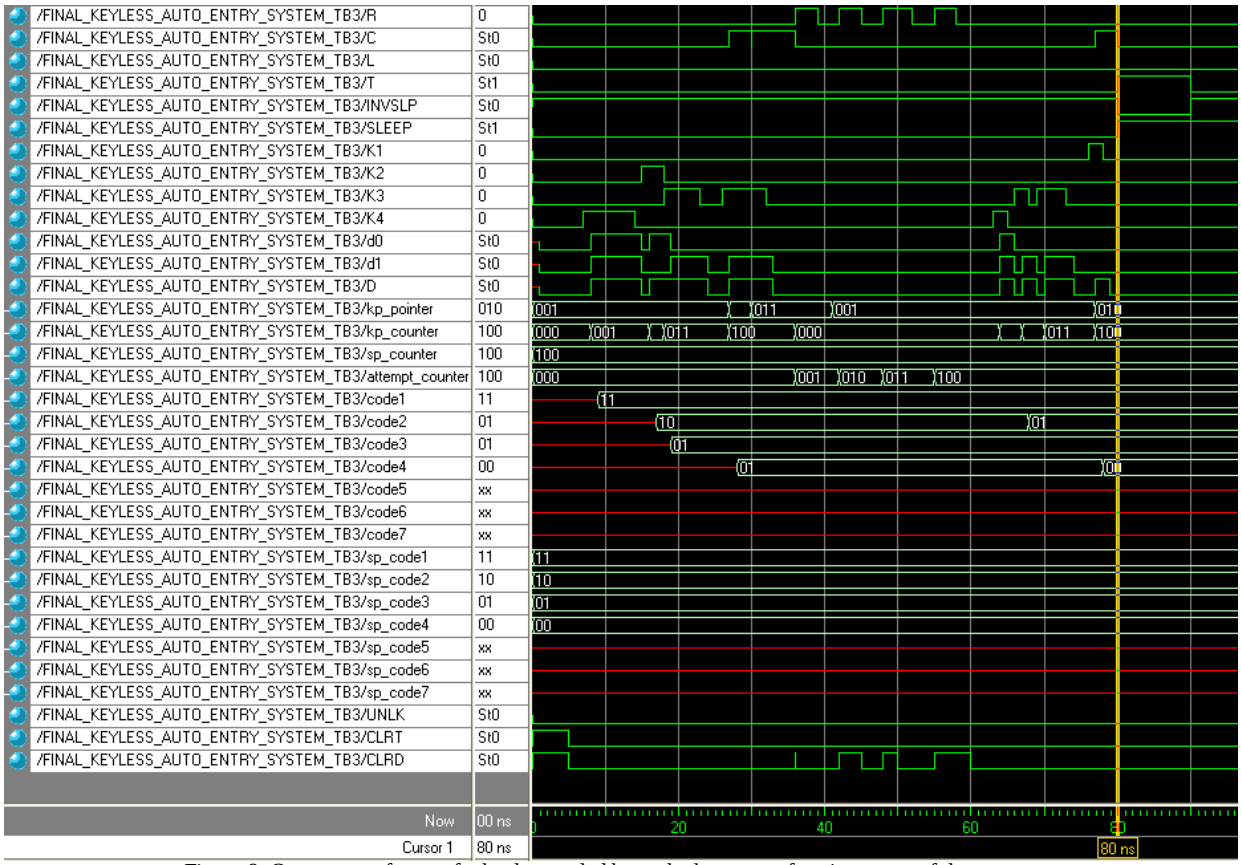

Figure 9. Output waveforms of a keyless coded home lock system after 4 unsuccessful attempts.