International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

RTL Synthesis and Analysis of Digital Code Lock System

Sabyasachi Mukhopadhyay[1], Sandeep Singh[2]

[1]*Assistant Professor, Dept. of ECE, SET, Sharda University, Greater Noida, UP, India.*
[2]*Assistant Professor, Dept. of ECE, SET, Sharda University, Greater Noida, UP, India.*

Abstract - *Security is a prime concern in our day-to-day life. An access control for locks forms a vital link in a security chain. The work which has been carried out in this paper presents the design of a keyless coded lock system. The operation is conducted by entering a combination of binary code to access the lock. If the correct sequence is detected, then it will unlock the lock otherwise it will remain locked. RTL synthesis is carried out using in ISE Design suit 14.1. ISim simulator is used for the functional verification.*

Key Words: **FSM, Sequence detector, HDL, RTL synthesis, Behavioral model.**

1. INTRODUCTION

In data communication network, any digital data is transmitted in the form of bit at a very high speed. Such movement of data is commonly called bit stream. One bit in any bit stream looks alike to many other bits. It is of utmost importance that any receiver can identify important information in a bit stream [1]. Digital code based lock system is basically a security system which allows any user to unlock the lock by entering a correct binary code to unlock the lock. The objective of the system is to provide enhanced security features. It also eliminates the possibility of the lock being broken. It is a key less security system where the owner only has to know the proper code to unlock the system [4]. The lock which is generally used in electronic safe is actually a code lock based system. The code which is being used may be numeric or alpha numeric [3]. In this work the code being used is binary in nature.

Finite State Machines (FSM) are generally sequential logic circuits. Such models are of high importance to realize certain types of systems, particularly those whose tasks form a well-defined sequence. The major applications of FSM are to implement operations that are performed in a sequence of steps [5]. FSMs are broadly classified as Mealy and Moore machines. The system has been presented in this paper is of Mealy type as the output depends on the state of the memory unit as well as on the present combination of the input. The system has been coded with HDL. The RTL synthesis and functional verification has been carried out using ISim simulator. The rest of the paper is organized as follows. In section 2, the design flow has been discussed. In section 3, RTL synthesis has been carried out. After that, in section 4, the simulation results have been shown and in section 5, the conclusion has been made.

2. DESIGN FLOW

The operation procedure of the code based digital lock system is based on the detection of a specific pattern given in binary number. In order to unlock the system one has to enter the correct pattern via the keypad module. This kind of lock can avoid the problems caused by the copying of keys. On the premise that its safety is high, it's another feature without key is getting more and more favor of people. There are great varieties of digital code locks. Generally speaking, they can be classified into score of type, such as electronic lock, fingerprint lock, card lock biological lock, etc.

The code which has been used in this design is "10110" to unlock the system. The inputs to the system are considered as clock, reset and x. The corresponding output is y which indicates the locking and unlocking status. Five states of the memory element have been considered. As the system is consists of five states hence three flip-flops of D-type is utilized for the implementation of the system. Each of the states is tested and a state transition is considered whenever there is a correct sequence of bit is received. The unlocking is considered by making the output bit as "1" as the correct pattern is fed in the input. Since overlapping is permitted hence the design consideration looks after the last two bits of the sequence to get another desired code. The state transition of the system is tabulated in Table 1.

After the system realization it has been found that the system output is not only depending on the present state of the memory unit, but also depends on the present input combination, resulting in a class A type machines. Class A type machines are also called Mealy type machines, where the output to the external world depends both on present state of the memory and the present input.

Present State			Input	Next State			Excitations			Output
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0	1	0	0
0	1	1	1	1	0	0	1	0	0	0
1	0	0	0	0	1	0	0	1	0	1
1	0	0	1	0	0	1	0	0	1	0

Table 1- State transition table of the system.

3. RTL SYNTHESIS

HDL is used by most of the hardware designers to describe designs at different level of abstraction. HDL is a high level programming language, with programming constructs such as assignments, conditions, iterations and extensions for timing specification, concurrency and data structure proper for modeling different aspects of hardware. Verilog and VHDL both are programming language which are being designed and optimized to describe the behavior of any digital system. The development, verification, synthesis and testing of hardware designs are supported by HDL [5].

In this paper Verilog has been chosen to describe the behavioral aspect of the system. Type of state encoding to be used is the most important decision to be made when describing any FSM. To encode the state of a state machine, one can select from several styles, where the binary style is the default encoding style.

Synthesis is the process of automatic hardware generation from a design description that has an unambiguous hardware correspondence. A Verilog description for synthesis cannot include signal and gate level timing specifications, file handling, and other language constructs that do not translate to sequential or combinational logic equations. Furthermore, Verilog descriptions for synthesis must follow certain styles of coding for combinational and sequential circuits. These styles are their corresponding Verilog constructs are defined under Verilog for RTL synthesis.

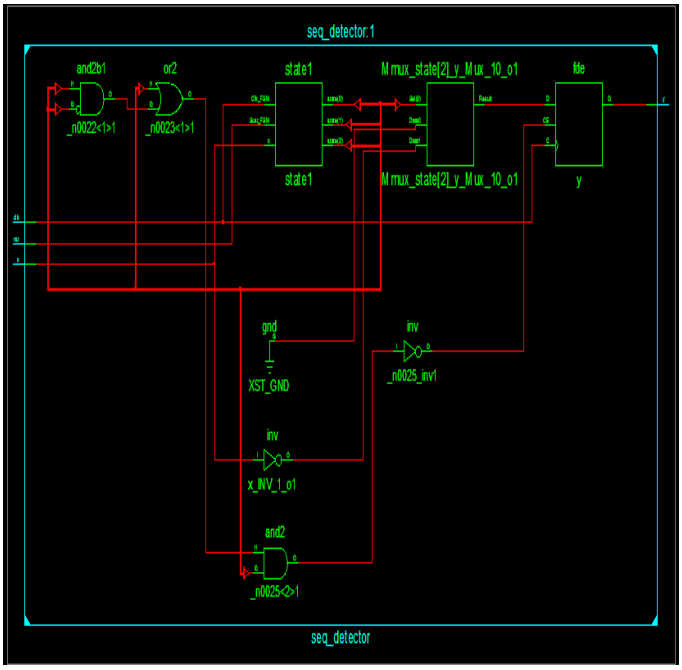

Fig 1- RTL Schematic of the system.

High-level design is less prone to human error because designs are described at a higher level of abstraction. High-level design is done without significant concern about design constraints. Conversion from high-level design to gates is done by synthesis tools, using various algorithms to optimize the design as a whole. This removes the problem with varied designer styles for the different blocks in the design and suboptimal designs. Logic synthesis tools allow technology independent design. Design reuse is possible for technology-independent descriptions.

Because high level Verilog designs are usually described at the level that specifies system registers and transfer of data between registers through busses, this level of system description is referred to as register transfer level (RTL). A complete design described as such has a clear hardware correspondence. Verilog constructs used in an RT level design are procedural statements, continuous statements and instantiation statements. The design summary is presented in Fig.2.

```
Top Level Output File Name         : seq_detector.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 5
#      LUT2                        : 1
#      LUT3                        : 2
#      LUT4                        : 1
#      LUT5                        : 1
# FlipFlops/Latches                : 4
#      FD                          : 1
#      FDR                         : 3
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 3
#      IBUF                        : 2
#      OBUF                        : 1
```

Fig 2- Design summary.

4. RESULT ANALYSIS

Simulation for design validation is done before a design is synthesized. This simulation pass is also called as behavioral simulation. A complete design that is described in Verilog may consist of behavioral Verilog, bus and interconnection specifications, and wiring of other Verilog components. Simulating a design requires generation of test data. This process can be achieved by use of a module called testbench.

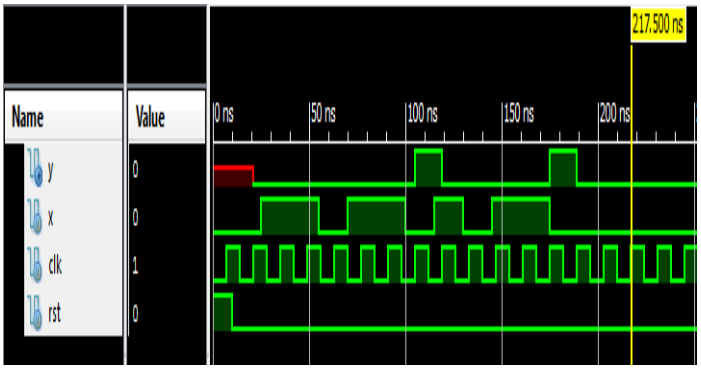

Fig 3- Simulated result.

Inside the testbench, the design that is being simulated is instantiated. The testbench together with the design forms a simulation model used by a simulation engine. Verilog codes have been compiled and simulated successfully using Xilinx ISim HDL simulator. The simulated result is furnished in Fig.3.

From the simulated result it is very much clear that, at the positive rising edge of the clock stream when the system identify the correct code at the eighth clock input, the output goes 1. Here, the output being high indicates that, the lock is unlocked as soon as it receives the correct pattern of code. Otherwise for rest of the cases it remains off if the correct code is not detected.

5. CONCLUSION

Looking in to the security concern in our day to day life, electronic locking system is becoming very popular today. In this paper we have tried to describe the working principle of digital code based locking system with its HDL implementation. After that, RTL synthesis of the system has been done and explained with design summary. Finally the functional verification has been carried out with proper analysis of the output data. The simulated output signifies the proper functioning of the system. In future, changing the code of the lock can be considered as open research issue.

ACKNOWLEDGEMENT

This work is supported by the electronics and communication engineering department of Sharda University, Greater noida,

REFERENCES

[1] Sasanka.P.V, Rao.Y.V.R, Siridhara.A.L, "A Verilog Model of Universal Scalable Binary Sequence Detector", International Journal of Scientific and Research Publications, Vol.3, Issue 4, April 2013, pp. 1-4.

[2] Oommen.A.P, et al., "Design and Implementation of a Digital Code Lock", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 2, February 2014, pp.7604-7607.

[3] Singh. C., "FPGA Based Implementation of Electronic Safe Lock", IOSR Journal of Electronics and Communication Engineering, Vol. 6, Issue 1, 2013, pp. 19-23.

[4] Yeap. K. H, et al., "Design of a Keyless Coded Home Lock System Using Verilog Hardware Description Language", IJECCT, 2012, Vol. 3, Issue 1, pp. 10-16.

[5] Chiuchisan. I, Potorac. A. D, Graur. A, "Finite State Machine Design and VHDL Coding Techniques", 10th International Conferrence on Development and Application Systems, Suceava, Romania, May 2010, pp. 27-29.

[6] Navabi. Z, "Verilog Digital System Design", Second Edition, Mc Graw Hill Education.

Yoeli. M, "Binary Ring Sequences", American Mathematical Monthly, Vol. 69, November 1962, PP. 852-855.