IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 6, Issue 1 (May. - Jun. 2013), PP 19-23
www.iosrjournals.org

FPGA Based Implementation of Electronic Safe Lock

Charnjit Singh[1], Abhilasha Sharma[2]
[1](M.Tech Student, Eternal University, H.P. India)
[2](Asst. Professor, Eternal University, H.P. India)

Abstract :*Thispaper is based on design of an "Automatic Security System Using VHDL" providing understandable and adequate operating procedure to the user. The operation is conducted by six different modules. If any of the modules fails, the failed module can be replaced without affecting the activity of others. The safety is ensured to the user by setting a secret code number which is the combination of three numbers, by doing so, only the authorized users can unlock the safe. The paper finds its appositeness in big organizations, military and banking sectors. Simulation through VHDL is quite generous and fiscal due to the reduction in number of components. Important operation consideration is to not give any indication to the user that the combination entered is incorrect until after the user has entered the all three numbers and pressed the OPEN key. Otherwise, it is possible for a user to determine the combination in no more than 96 attempts, as opposed to no more than 32,768 attempts.*

Keywords *–FPGA, FSM, LOCK, VHDL, XILINX*

I. INTRODUCTION

The electronic lock is often introduced to illustrate the possibilities of design of an electronic system using a top-down approach. This is because it is not too complicated but must be designed as a complete system. The "customer" must be able to specify the key data sequence and the sequence must be easily modified when required. As designs become more complex, it becomes more efficient to move away from ad-hoc methods and use tools that allow the design to be carried out at a higher level of abstraction[1].

The design of electronics to control any automated system is generally referred as a CODE LOCK MACHANISM. An application of this is design of electronic safe. The front panel of the safe is kept as simple as possible for understanding of user. A code locking mechanism that transmits information between a key and a locking mechanism secured against interception misuse, but flexible with respect to original equipment, replacement parts, and emergency functions[2] , is obtained if the key code is, in each case, cryptographically encoded through a hardware encoder. When the locking mechanism is first operated, a control element inductively coupled to a locking mechanism read-write unit with a new key once and in a non-over writable manner, transmits an indent number to the object memory [3]. Object-specific identity data are also stored in the control element. With the latter, a key number from a key register of the control element is combined with the key code to read the ladder together with a roll-in random code into the set of first still-neutral keys and into the object memory. Thus, a key is valid only if the object-specific identity data are taken into account in its key code [4].

The lock used in the electronic safe is basically a combinational lock. A combinational lock is a lock that requires the entering of a specific sequence of symbols to dislodge and open. The symbols are typically numerical, letters and other types of symbols may also be used to formulate a sequence. Depending on whether the lock is single-dial, multiple-dial; or discs or keypads may be used to enter sequences. Combinational locks are generally thought to be secure.

II. Overview Of Electronic Safe

The safe`s electronics receives inputs from the front panel & provide signals to the seven segment displays & to a motorized bolt. When the bolt`s motor is actuated to turn in one direction, it unlocks the safe door [5].

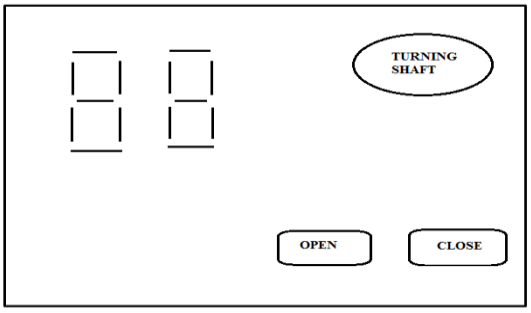

Fig1. Basic View Of Electronic Safe

III. System Design

The block diagram of electronic safe lock and it is consists of six design entities [7].
U1: ose_decoder_fsm U5&U6: bcd_7seg
U2: bcd_2dec
U3: digit_compare
U4: master_fsm

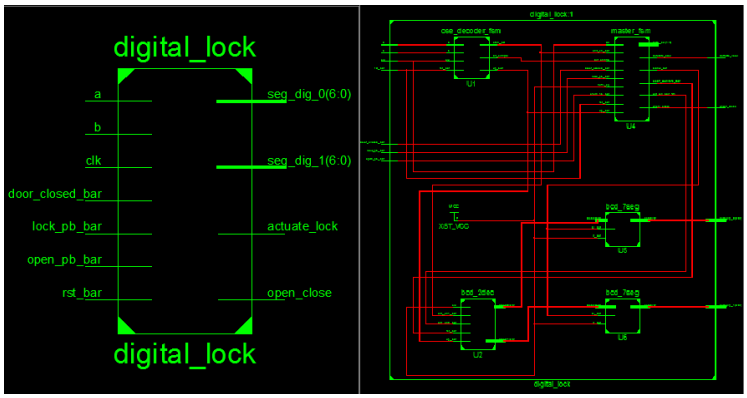

Fig2. Block diagram of electronic safe lock

U1&U4 are both fsms which together control the operation of the system. U2 is two digits modulo BCD counter. U5&U6 convert each BCD digit from U2 to a seven segment code. In theory, the control portion of the electronic safe could be designed as a single fsm. However, it simplifies defining the operation of electronic safe to partition the control function into two cooperating fsms. Ose fsm is component of ose_decoder_fsm. It has limited and very specific functions. The other fsm is master_fsm. This fsm controls the overall operation of the safe.

U1: OSE_DECODER_FSM

We can design a four time decoder that enables the counter to count once for each combination of A&B. this increases the effective resolution of OSE and decoder combination by a factor of 4, compared to simply counting positive edges of A(or B) directly[4].

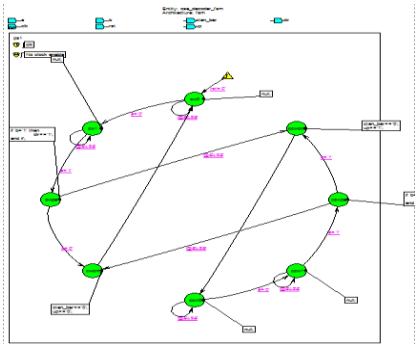

Fig3. State diagram of ose_decoder_fsm

U2: BCD_2DEC

This two-digit BCD counter counts from 00 to 31 and then rolls over to 00. The count direction can be up or down. The counter has two counts enable inputs: both must be asserted for the counter to count. Integer variables are used to store the count.

U3: DIGIT_COMPARE

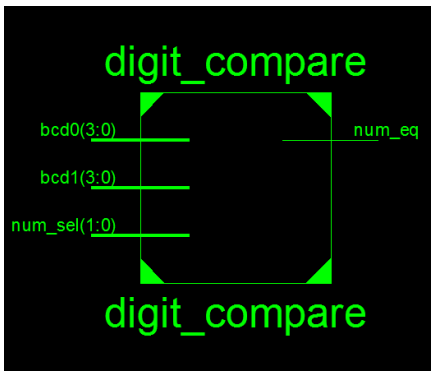

Fig4. Block daigram of digit_compare

This block is used to compare the two digits being entered by the user. The entity digit_compare has bcd0, bcd1 and num_sel as inputs and num_eq as an output. The architecture body compares the three combinations entered by the user. When the entered combination is correct the output signal num_eq becomes equal to '1' otherwise it remains equal to '0'.

U4:MASTER_FSM

The master_fsm is the most important and useful block of electronic safe. This block controls the functioning of all the signals and blocks. In other words we can say that it is the heart of the electronic safe[9].

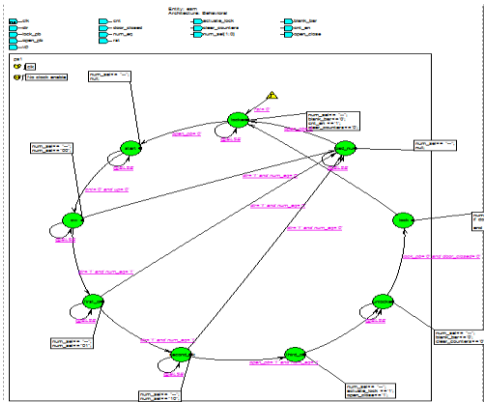

Fig5. State diagram of master_fsm

The overall characteristics of the safe are determined by component master_fsm.

U5&U6:BCD_7SEG

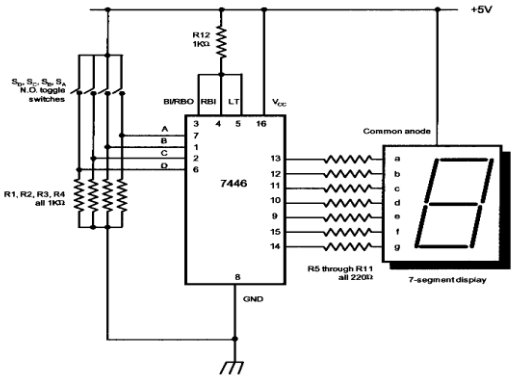

Fig6. Bcd to seven-segment decoder

An often used decoder function is a BCD to seven-segment decoder [8]. This decoder takes as its inputs four bits that represent a binary coded decimal value and generates a 7-bit output to drive a seven-segment LED. If the BCD input has a value from 0 to 9, the output consists of the value needed to display the corresponding decimal digit. If the BCD input is greater than 9, all the outputs are don`t care. Since these input values should not occur.

IV. Rtl Simulation

The RTL simulation step is used to verify the correctness of the RTL VHDL description. The designers use it to describe the clock by clock the behavior of the design. A standard VHDL simulator can be used to read the RTL VHDL description and verify the correctness of the design. The designer can look at the output of the simulator and determine whether or not the design is working properly.

The simulator will generate the output data that can be analyzed. The designer usually has number of ways to analysis the data. The most common are output waveform and text tabular output. The RTL simulation is shown below [10]:

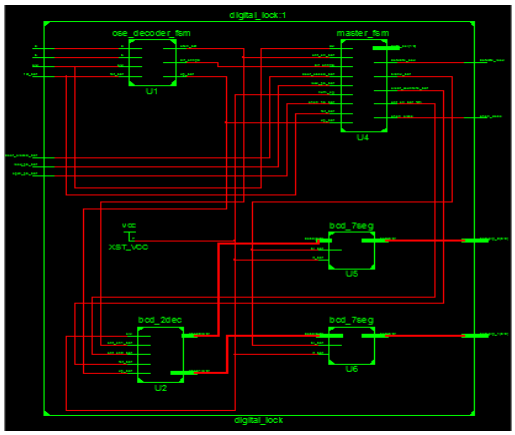

Fig7. RTL view

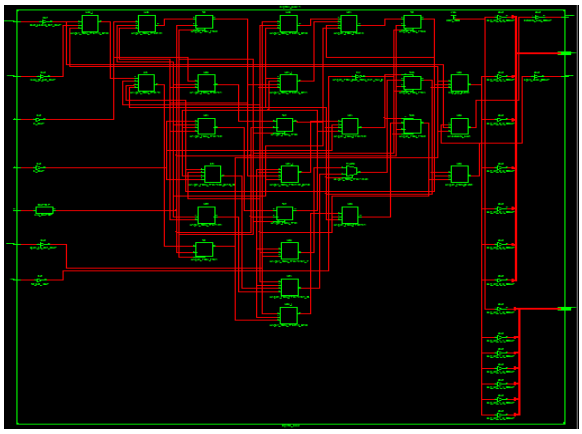

Fig8. Technology view

Fig9. Simulation result

V. Conclusion

In the present scenario of the technology there are two fields in engineering that hold great demand in the industry, namely-communication & VLSI. Therefore, in order to add glimpse to this development we thought of this research.A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device such as CPLD or FPGA, then it is the actual hardware being configured, rather than the VHDL code being.When implementing the circuit in anti-fuse type FPGA, it is an advantage that the configuration file in memory is not necessary. In this case it is already verified that the circuit will be fully functional.

References

[1] JialiangZhang , The Design, Simulation, Verification and Implementation of Vending Machine Based on FPGA , The Ohio StateUniversity 2012 , Advisor: Prof. Y. F. Zheng, 2012.
[2] David Lee, MihalisYannakakis, PRINCIPLES AND METHODS OF TESTING FINITE STATE MACHINES, AT&T BellLaboratories,Murray Hill, New Jersey Proceeding of the IEEE, 1996.
[3] Scott Harper and Peter Athanas, A Security Policy Based upon Hardware Encryption, Proceedings of the 37th Hawaii InternationalConference on System Sciences - 2004
[4] Ana Monga1, Balwinder Singh2, Finite State Machine based Vending Machine Controller with Auto -Billing Features, InternationalJournal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012
[5] Books:
[6] R.P.JAIN, "Modern Digital Electronics", Tata McGraw Hill, Third Edition 2003.
[7] J.Bhaskar, "A VHDL Primer", Pearson Education Asia, Third Edition 1999.
[8] Kenneth L. Short, "VHDL for Engineers", Pearson Education Asia 2009.
[9] John N.Roth, "Digital Design using VHDL".
[10] ACTIVE HDL, "ACTIVE HDL Software Quick Start Guide".
[11] XILINX INC, "XILINX Software sampler Quick Start Guide".
[12] M.morris Mano, "Digital Logic and Computer Design", Prentice Hall of India, Nov 1995.
[13] Volnei A. Pedroni, "Circuit Design with VHDL", Prentice Hall of India 2006.
[14] SudhakarYalamanchili, "Introductory VHDL", Pearson Education Asia 2005.
[15] Kevin Skahill, "VHDL for Programmable Logic", Pearson Education Asia 2004.