International Journal of Electrical, Electronics and Computer Engineering **1**(2): 42-50(2012)

ISSN No. (Online) : 2277-2626

Design of A Keyless Digital Security System

E.O. Oyetunji **and** ***J. Asare*****

*Academic Quality Assurance Unit
University for Development Studies, Ghana
**Department of Computer Science
University for Development Studies, Ghana

(Received 30 Augest, 2012, Accepted 29, September, 2012)

ABSTRACT: This paper considers the design of a keyless digital security system using a hardware programming tool. The system uses a 4 x 4 keypad that allows users to enter four keystrokes as security password. These keystrokes are converted to BCD codes by the keypad encoder. The designed system comprises of two basic components fused together as a block (the Memory logic and the Security logic). The memory logic is the storage medium for the 4-keystrokes while the security logic compares entry keystrokes to keystrokes that are stored in the memory logic.

Keywords: Security, Keyless, Alarm, Password, Authentication, Digital logic.

I. INTRODUCTION

Several lock systems have been developed using different programming language such as C programming language [1, 2, 3]. There exists the problem of memorizing security keys when the number of input bits is so numerous. These numerous number of inputs bits also leads to a large logic and memory size. This has, therefore, necessitated the use of keyless digital systems [4, 5, 6]. A keyless entry system is an electronic lock that controls access to a building or vehicle without using a traditional mechanical key [1].

A system is any collection of component elements that work together to perform a task. On the other hand, a sub-system is a system which is part of another system. Typically, a system consists of components (or elements) which are connected together in order to facilitate the flow of information, matter or energy. Digital systems, which are mainly hardware based, use digital signals (0s and 1s) in their operations. They are designed using logic gates in a Hardware Description Language and once designed and implemented, they mostly become impossible to decode [7, 8]. Therefore, the aim of this paper is to design a digital security system using a hardware description language (VHDL). The digital security system designed can be used by banks and other financial institutions to maintain maximum security and reduce the occurrences of theft and robberies across their branches.

II. LITERATURE REVIEW

Security system is an electrical device that sets off an alarm when someone tries to break in [9, 10]. In the past, people relied on simple means to alert others of a breach in security. These includes: little bells (which are attached to a door that rang when it was opened), tin cans (which are tied to a string across a pathway). Sophistication in crimes nowadays has made improved security system imperative. The 25 GTO Digital Keypad is a multipurpose keypad that can work with other applications in addition to GTO gate openers and locks. As a wired keypad it can operate garage door openers and gate openers, which require 24 volts or less and accept normally open contacts. As a wireless keypad it can be used with any gate or garage door opener that receives a 318 MHz radio frequency (RF) signal [11].

Mohd [2] developed a Smart Keyless Entry that used a receiver and a transmitter to send data within a range of two meters. His technology was applied in cars where a solenoid valve driven by 3 circuits are used to move the master actuator that serves as lock. In his application addressing is used to give transmitters and receivers a unique identity so that direction to a receiver can be specific. Another keyless security system was developed by Bell System Telephones Ltd. (BSTL), a C106 Coded Access Keypad including CS106 Door Entry & Coded Entry. It has**;** 10 codes each of 1 to 8 digits, two time zones for Staff/Executive operation, Output for Fail-Safe and Fail-Secure locks, Exit facility, lock timer, secure programming via the keypad, and non-volatile memory. It controls access to a door by means of a keypad and an electric lock release mechanism. The C106 Keypad may be programmed with up to ten unique access codes [12].

ATMEL, developed AVR245. A Code Lock with 4x4 Keypad and I2C™ LCD, Ideal for low pin count AVRs. It uses; I/O pins to read 4x4 keypad, Timer/Counter to control piezoelectric buzzer, USI in TWI mode to communicate with I2C™ LCD, and a firmware written entirely in C language [13].

Logical Security consists of software safeguards for an organization’s systems, including user identification and password access (also known as logins. User names, logons or accounts are unique personal identifiers for agents of a computer program or network that is accessible by more than one agent. These identifiers are based on short strings of alphanumeric characters, and are either assigned or chosen by the users.), authentication (the process used by a computer program, computer, or network to attempt to confirm the identity of a user. Blind credentials (anonymous users) have no identity, but are allowed to enter the system. The confirmation of identities is essential to the concept of access control, which gives access to the authorized and excludes the unauthorized.), access rights and authority levels. These measures are to ensure that only authorized users are able to perform actions or access information in a network or a workstation [3, 14, 15]. The key challenge for most security systems are the numerous numbers of inputs bits that leads to a large logic and memory size. This has necessitated the need for a new security system. This security system is a password authentication program that is aimed at storing four bit security code in memory logic permanently which can only be changed by reprogramming the logic.

III. MATERIALS AND METHODS

The security system was designed to take up four entry key strokes as security codes. It is made up of a keypad encoder, memory storage, memory address decoder, memory array, memory logic, a One- shot, and security logic. Each of these logical gates has sub- circuit gates embedded in them to form a complete block/ unit. The system was designed using the Quartus II environment [16]. At each stage of the development process, components were tested to ensure a successful compilation and final simulation.

IV. THE DESIGNED SYSTEM

The major components of the designed security system are described below.

A. Keypad encoder

The keypad encoder is necessary for converting a key stroke to a BCD code. It consist of; a 12-input NAND gate used as a negative OR gate, a 74147 encoder selected from the software library, and 14 NOT gates. The keypad encoder has input pins In0 through In9 while the five output pins are denoted as D0 - D3, G1Out (Fig. 1). After the keypad encoder has been developed, saved and compiled, it was blocked for simplicity of subsequent logic gates as shown in Fig. 2.

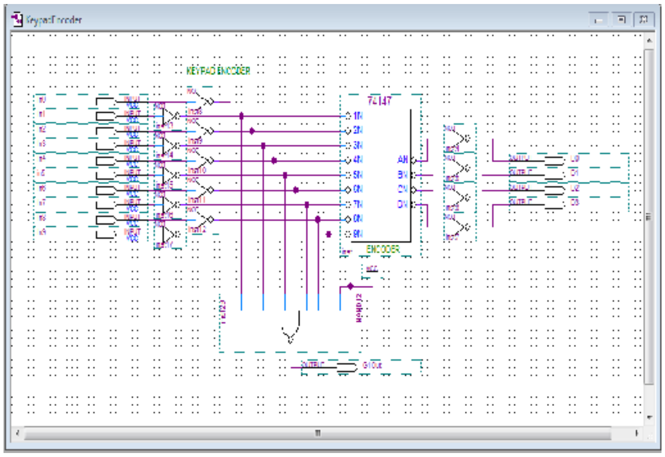

Fig. 1. Schematic diagram of the keypad encoder.

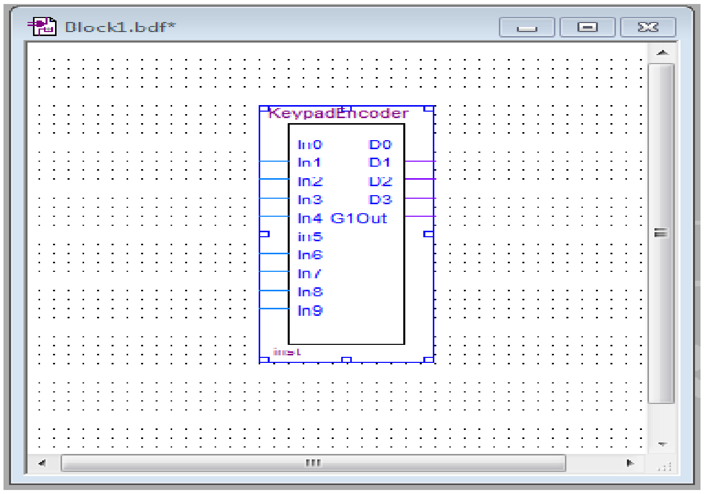

Fig. 2. Block diagram of the keypad encoder.

B. Memory storage

A J-K flip-flop was used as the basic storage element. It can be operated in two modes (read and write). In the write mode, AddSel (address select) is High and the R/W (read/write) input is low. This is basically to enable gates G1 and G2. The J-K flip-flop uses two inputs J and K so that the input D = JQ + KQ. It combines the behavior of SR and T flip-flops in a useful way. It behaves as the SR flip-flop where J = S and K = R for input values except J = K = 1.

The JK flip-flop toggles its state like the T flip-flop. The input bit is applied to the J input whole its compliment is applied to the K input. The input bit is then stored on the positive edge of the clock pulse (from an external source). In the read write mode AddSel is HIGH and R/W is HIGH enabling G3. The stored bit on the Q output of the flip-flop appears on the output G3 (Bit out) (Fig. 3). The memory storage gate was compiled after it has been saved and blocked as seen in Fig. 4. Sixteen (16) blocked diagrams of the memory storage circuits will be required in the development of the memory array.

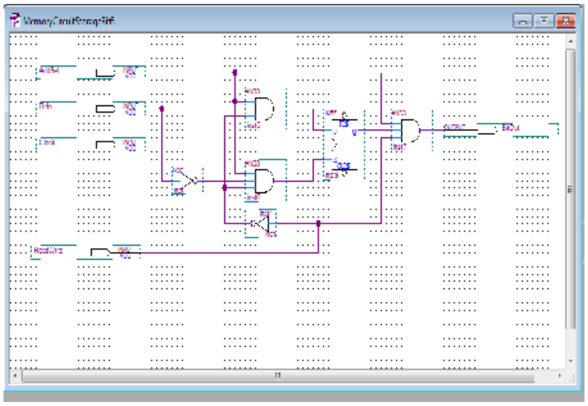

Fig. 3. Schematic diagram of the memory storage gate.

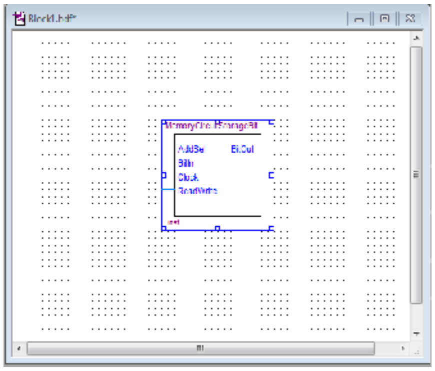

Fig. 4. Block diagram of the memory storage gate.

C. Memory address decoder

A 2-bit binary sequence was applied to the selected input pins (S0, S1) to select each of the four memory addresses using the AddSel lines (Address Select lines). The input to the address decoder was sequenced through each of four states (00, 01, 10, 11) to select each row in the memory (Fig 5).

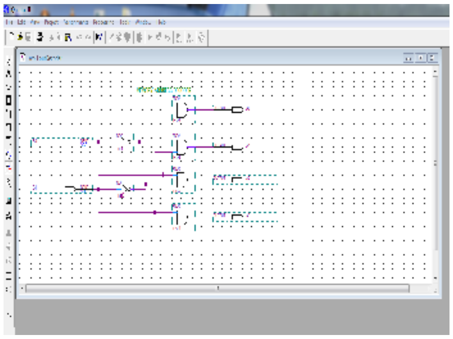

Fig. 5. Schematic diagram of the memory address decoder.

After the memory address decoder has been developed, saved and compiled, it was then blocked for simplicity of subsequent logic gates as shown in Fig. 6.

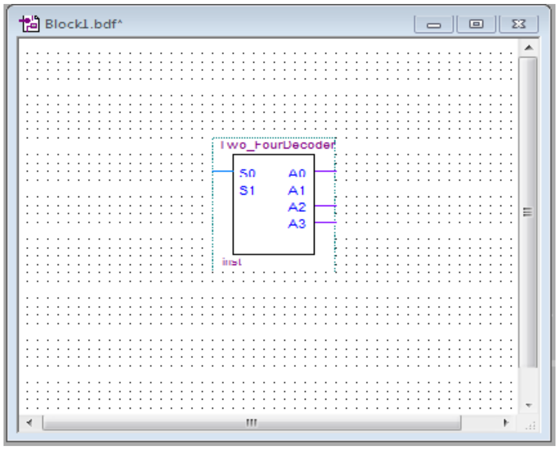

Fig. 6. Block diagram of the memory address decoder.

D. Memory array

The memory array was composed of sixteen memory storage blocked circuits also called cells. These cells were arranged in rows with each row constituting four cells. The Address Decoder was connected to the AddSel ends of the cells to sequentially select each row (Fig. 7). After the memory array has been developed, saved and compiled, it was then blocked for simplicity of subsequent logic gates as shown in Fig. 8.

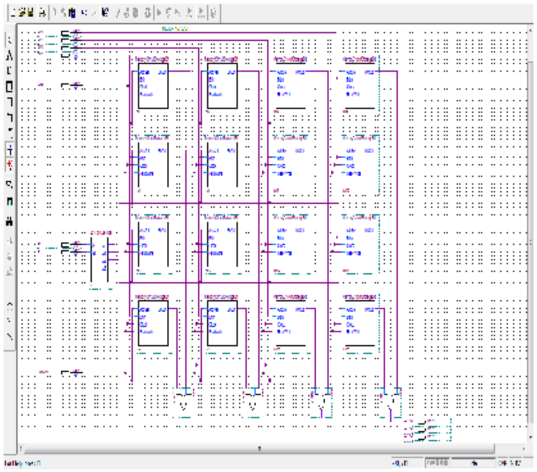

Fig. 7. Schematic diagram of the memory array.

E. Memory logic

The Keypad encoder is necessary for converting a key stroke to a BCD code, and a 2-bit counter was used to produce the sequence for selecting the memory addresses. The counter was reset to the 0 state by a reset input from the code entry logic and was clocked through its sequence after each key entry (Fig. 9). After the memory logic has been developed, saved and compiled, it was then blocked for simplicity of subsequent logic gates as shown in Fig. 10.

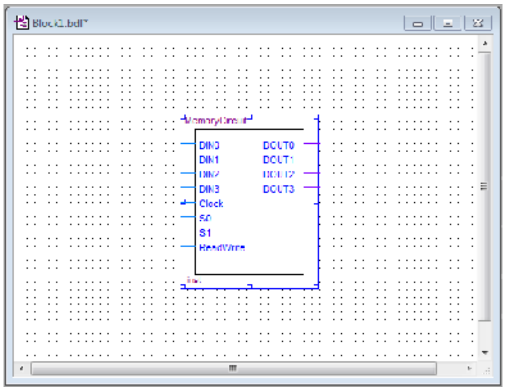

Fig. 8. Block diagram of the memory array.

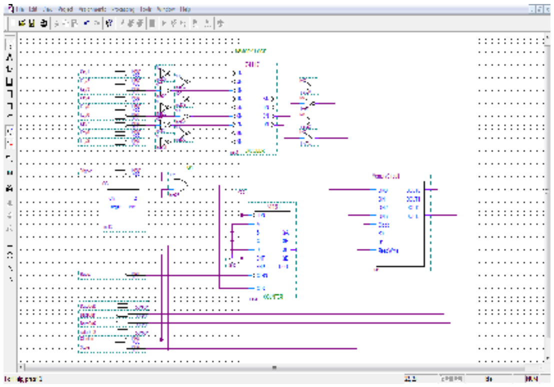

Fig. 9. Schematic diagram of the memory logic.

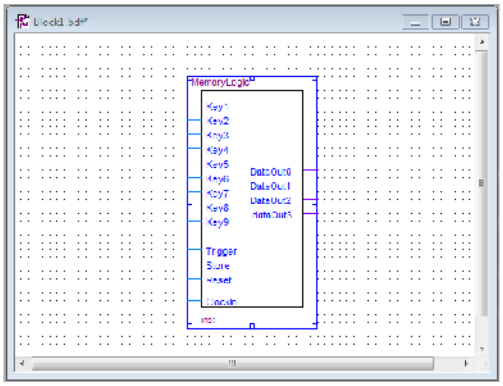

Fig. 10. Block diagram of the memory logic.

F. One-shot

Three One-Shot are needed in the security logic which was programmed in the same way using the VHDL. It was first programmed with codes and then blocked (Fig. 11).

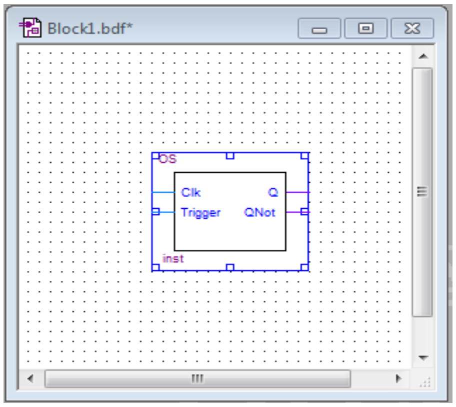

Fig. 11. Block diagram of the one- shot.

G. Security logic

The security code logic controls the arming and disarming of the system. When the system is armed, shift register C is preset to 00010000 producing a LOW on ArmOut to activate the system sensors and alarm circuit and to light the LED. To deactivate the system, the first digit of the security code that is stored in the memory is entered from the keypad. The decimal-BCD encoder produces the BCD code representing the digit that was pressed on the keypad. One-Shot A (OSA) is triggered by gate G1 and produces a pulse that clocks the 4-bit BCD code from the encoder into register A and the correct pre-stored 4-bit BCD code from memory address 0 into register B. Once they are clocked into the registers, these two codes are applied to the inputs of the comparator. If the correct digit has been entered on the keypad, the 4 bits on the A inputs to the comparator and the 4 bits on the B inputs are the same, resulting in a HIGH on the A = B output of the comparator. When a digit is entered One-Shot A also triggers One-Shot B (OSB) which in turn, triggers One-Shot C (OSC). The output pulse from OSB goes to gate G3, and since the Arm/ Disarm input is HIGH, produces a trigger pulse that goes to the memory to advance it to the address of the next BCD code. The output pulse from OSC clocks register C and shifts the preset contents (0001000). Since there is still a 0 on the serial output (ArmOut), the system remains armed. After the second correct digit is entered on the keypad, shift register C contains 00000100, and the system remains armed. After the third correct digit is entered on the keypad, shift register C contains 00000010. After the fourth and final correct digit is entered, shift register C contains 00000001. Since there is now 1 on the serial output (ArmOut), the system is disarmed and one can enter the building. If an incorrect digit is entered at any time, the comparator output will be LOW, causing gate G2 to produce a LOW on the SH/LD input of register C putting it in the parallel load mode. OSC then clocks the register and loads 00010000. At this point, you must start over and reenter the entire 4 digits (Fig 12). After the security logic has been developed, saved and compiled, it is then blocked for simplicity of subsequent logic gates as shown in Fig 13.

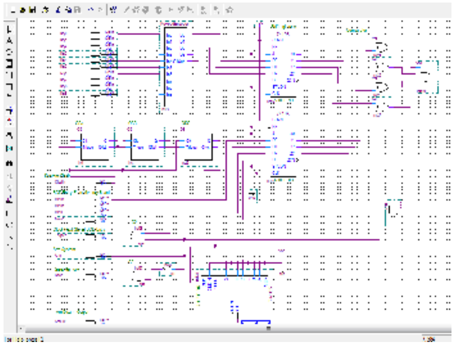

Fig. 12. Schematic diagram of the security logic.

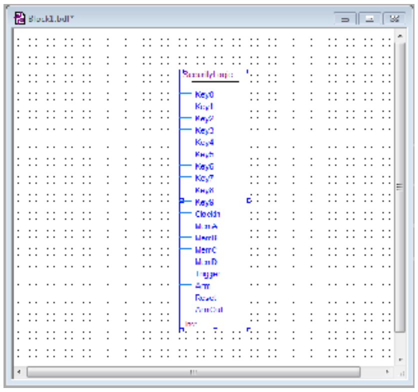

Fig. 13. Block diagram of the security logic.

H. *Complete logic blocked of the security system*

Having fully designed and successfully compiled both the Memory logic and the Security logic, they are then joined together to form the entire security system as shown Fig 14.

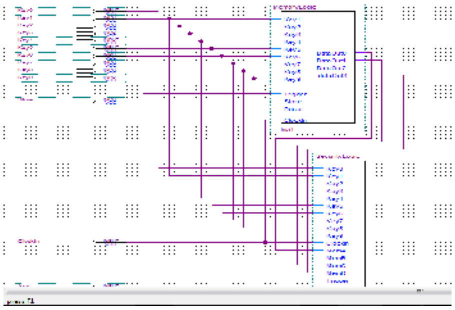

Fig. 14. Schematic diagram of the security system logic.

IV. SIMULATION RESULT

The security logic was then simulated which produced an output/result in the wave form indicating that the logic will work in accordance when it is being uploaded onto a printed circuit board.

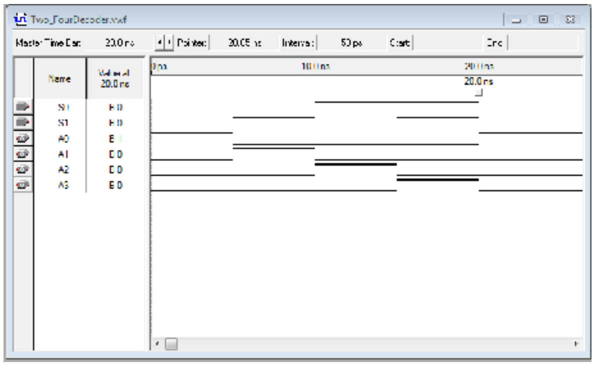

Fig. 15. Wave form diagram after simulation.

CONCLUSION

In this paper, we have designed a digital security system using a hardware description language. The various components of the system was separately designed, tested and compiled for success and then later combined together to form a single digital security system. The digital security system was simulated and the result indicated that the logic will work properly when uploaded onto a printed circuit board. The digital security system developed can be used by banks and other financial institutions to maintain maximum security and reduce the occurrences of theft and robberies across their branches.

REFERENCES

[1] The Fight for Security, From Padlock to Keyless Door Lock Systems. (Http://www.comerciogt.com/jm/technology/, Accessed: May, 2012).

[2] Mold, F. B. R., (2007), Development of an Automated Door Locked (Smart Keyless Entry), Unpublished BSc. Thesis, University Technical of Malysia, April 2007.

[3] Nisar, A.R. and Shekh, M. A., (2010), Digital Security Lock Made by Using AT89S52 Microcontroller, COMSATS Institute of Information Technology, Islamabad, CEPEX 2010, 2nd -3rd APRIL 2010.

[4] Smith, D., (1993), Passive Keyless Entry, Latest From Lectron.

[5] Kagin, R. (2007), Keyless Entry, University of Illinois.

[6] Waraksa, T., Farley, K., Kiefer, R., Douglas, D., and Gilbert, L., (1990), Passive Keyless Entry System.

[7] Enoch, O. H., (2005), Digital Logic and Microprocessor Design With VHDL.

[8] Volnei, A. P. (2008), Digital Electronics and Design with VHDL, Elsevier- Morgan Kaufmann publications, Burlington, MA 01803, USA.

[9] Alrabady, A. and Mahmud, S. (2003), Some attacks against vehicles' passive entry security systems and their solutions. Vehicular Technology, *IEEE Transactions on*, **52**(2): 431 – 439.

[10] Alrabady, A. and Mahmud, S. (2005), Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs. *IEEE Transactions on Vehicular Technology*, **54**(1): 41–50.

[11] 25 Code GTO Digital Keypad. (Http://www.gtoinc.com , Accessed: June, 2012).

[12] http://www.bellsystem.co.uk (accessed: June, 2012).

[13] (http://www.atmel.com ((accessed: June, 2012).

[14] Diem, W. (2001), "Smart Card Opens the Door", Auto Technology, Vieweg Publishing, Wiesbaden, DE, 1(1), Feb. 2001, pp. 32-33.

[15] Http://www.tomrubenoff.hubpages.com/hub/Keyless_Lock. (Accessed: May, 2012).

[16] Malika, S.K. and Poonam, G. (2011); Quartus Usage Manual.