Volume 4, Issue 1
February 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Implementation of Security solutions using VHDL Design Flow

[1]Awais Ahmed, [2]Asim Shahzad, [3] Muhammad Saeed
[1, 2, 3] School of Electrical Engineering, University of Faisalabad, Pakistan

Email: [1]awais.cheema@tuf.edu.pk

Abstract:

We proposed a design of a digital security system in this paper using VHDL. A security lock is introduced which allows an authorized bit pattern to get unlocked or ignite an alarm signal; a buzzer or a blinking LED. The main objective of this paper is to provide a mature security solution to the authorized persons only.

I. Introduction:

Considering security a prior issue of this age, mankind is always keen to explore techniques which could keep them or their belongings as much safe as possible. Designing a digital system using VHDL is always handy, easy to alter and secure. The advantage of using VHDL is that this language has constructs to handle parallelism inherent in hardware designs, file input and output capability, can be mapped onto a programmable logic device and it can be used for several FPGA implementations. A VHDL project is portable, multipurpose and it allows the description of a concurrent system [1].

II. Methodology:

In this paper, the design of a digital lock is introduced using VHDL and it uses a specific 4 bits combination to get unlocked or an alarm signal is ignited i.e. buzzer or blinking LED. Three inputs A, B and C are introduced which will define authorized input bit stream that will be used to open up the lock. Keystrokes are considered as asynchronous inputs, so in this report different circuitries are introduced in order to generate a single pulse for each pressing as well as to make whole system synchronized. The push buttons pressings responsible of defining input bit stream are fed to a single pulse circuitry that will enable the system to generate a single pulse in response to each pressing of the push button as one may hold the push button for a long interval. The output of single pulse circuit is then fed to a synchronizer circuit that is basically used to lock up the input bit with the system clock rising edge as in any security application e.g. a security lock, the security is the first priority so we have to develop a fool proof security lock for digital applications. The synchronizer circuit will lock up the clock signal with the push button input signal and assertion of a wrong bit stream will be prevented. The output from these synchronizer circuits will be then fed to a finite state machine that will judge the input and act according to the coding in it and finite state machine's states will be displayed on a seven segment display (SSD). The structural and data flow designs can be combined using a variety of VHDL statements and structures [4]. VHDL is always considered suitable for high data encryption rate [5]. A typical VHDL design can be shown as follows:

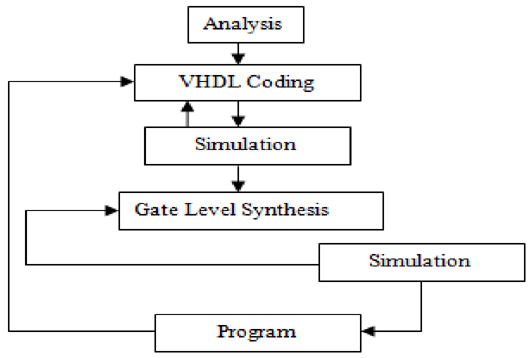

Figure1: Typical Design flow

- **<u>Proposed structure and design of the system</u>:**

In this paper I proposed the design of a digital security lock which allows authorized input bit streams to generate a signal "UNLOCK" or an alarm signal is initiated. This operation takes place in four stages.

- Stage1: Keeping in view that keystrokes are asynchronous inputs, in this stage input bit stream is first fed to the single pulse circuit which will enable the system to generate a single pulse in response to each pressing of the push button as one may hold the push button for a long interval.

- Stage2: The single pulse output is then fed to the synchronizer circuit that is used to lock up the input bit with the system clock rising edge as in any security application e.g. a security lock.
- Stage3: The output from these synchronizer circuits will be then fed to a finite state machine that will judge the input and act according to the coding in it and finite state machine's states will be displayed on a seven segment display (SSD).
- Stage4: This security lock asserts an 'UNLOCK' signal when a combination B-C-A-C is asserted to the system. In response to a wrong code, the system waits for all four inputs assertion and then generates a signal 'Z' that controls the 'ALARM' signal which runs the alarming. In order to reset the alarming state, a combination of A-A is asserted and system is pushed back to the reset state S9. As a safety precaution, from any state a combination C-A is asserted that leads the system to the initial state S9. Figure2 demonstrates suggested block diagram of the system.

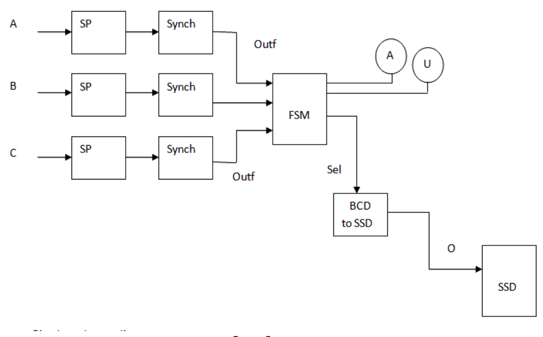

Figure2: Block Diagram

▪ Single Pulse Circuit:

This circuit is essential keeping in view that when a push button is pressed for a short interval of time, the time that switch will be closed is longer than one clock period which may generate an ambiguity for state machine to consider this single bit as a bit stream. So this circuit is necessary to add after each push button in order to prevent this happening. After addition of the circuit, push button switch will generate only one pulse every time one presses the push button and this pulse generation would be time independent. Figure3 demonstrates the circuit diagram for single pulse circuit. Some extra control circuitry and NAND gate can be used to generate a single pulse output [2].

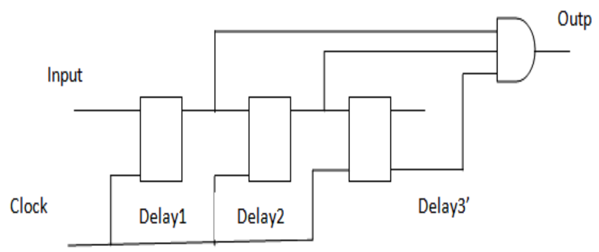

Figure3: Single Pulse Circuit

The AND gate here is used to generate the single pulse as after each delay, one input in gate out of three is asserted enabling the AND gate to generate the output "outp" after the third delay as the Boolean expression for AND gate is "Delay1 and Delay2 and (Not Delay3)". Implementation of single pulse circuit in VHDL lead to the following simulation in Figure4 which clearly shows that the output is generated at the third rising edge of the clock and output is single pulsed.

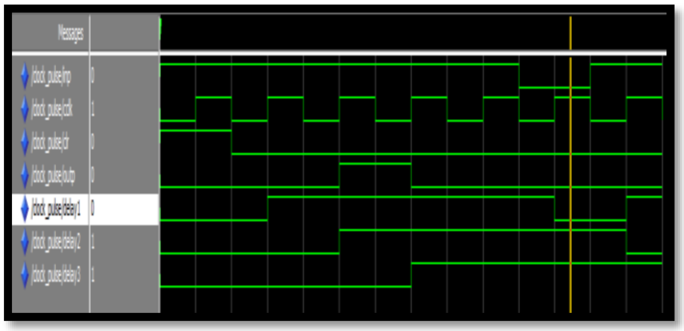

Figure4: Single pulse circuit simulation results

▪ Synchronizer circuit:

Timing is a prime issue while dealing with digital locking systems. If the duration of pressing the push button does not lock up with the clock edge, right combination cannot be asserted and alarm signal will be asserted after every fourth try. In order to prevent this sort of happenings, push buttons need to be synchronized with the clock edge in order to prevent the wrong bit streams. For a synchronizer circuit, one have to lock up the input bit stream or a single bit with the clock pulse and this was implemented by adding flip flop and introducing global clocking. Synchronizer circuits are used to increase reliability of data transfer especially in higher frequencies [3].

Volume 4, Issue 1
February 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Figure5 demonstrates the inclusion of synchronizer circuit for this paper.

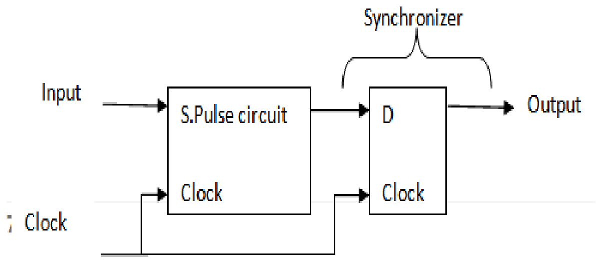

Figure5. Inclusion of synchronizer circuit

The input is first passed through three stages, and then we get the output (outp) after three clock cycles (on rising edge of third clock cycle) and after adding the fourth flip flop that basically acts like a synchronizer here, the final output "outf" is obtained after four clock cycles (on the rising edge of FOURTH clock cycle). Implementation of synchronizer circuit lead to the simulation results in figure6.

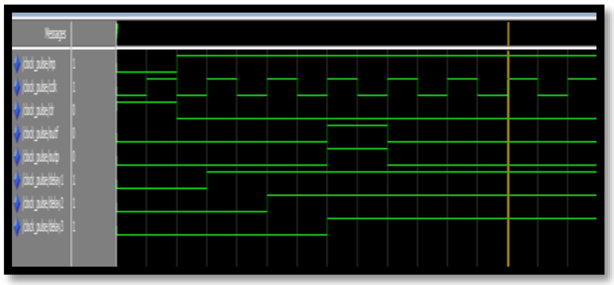

Figure6: synchronizer circuit simulation results

- **BCD to SSD converter:**

The BCD to SSD converter is basically used in order to change a 4 bit input (from FSM) to a 7 bit output. A converter design was implemented in VHDL and simulation results are shown in figure7.

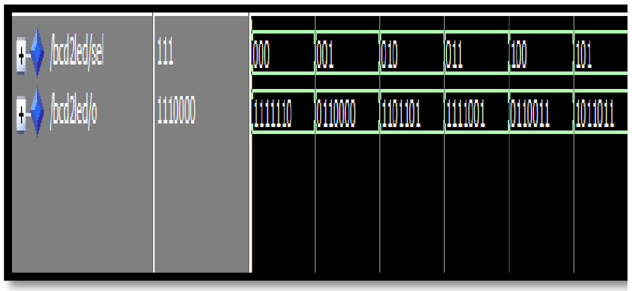

Figure7: BCD to SSD simulation results

- **Finite state machine (FSM):**

The state machine is designed in order to fulfill all the system behavior and system needs. Prior to describing the state machine, the system response should be understood. In this system, only one combination is accepted to generate a signal "UNLOCK" and the combination is "B-C-A-C". If this combination is pressed in a right order, an "UNLOCK" signal will be asserted enabling the lock to be unlocked safely. In case of pressing any other four bit combination except the one described, an alarm signal will be asserted enabling the LEDs or buzzer to switch on. In case of alarm, a combination "A-A" will lead the system to its start/ initial position and in case of unlock, any bit will lead the system to the initial state. Furthermore, keeping human mistake in mind if at any state a combination C-A is pressed, the system will again access to its initial state. This state machine is designed in ASM notation with nine states. Figure8 shows the design flow of finite state machine.

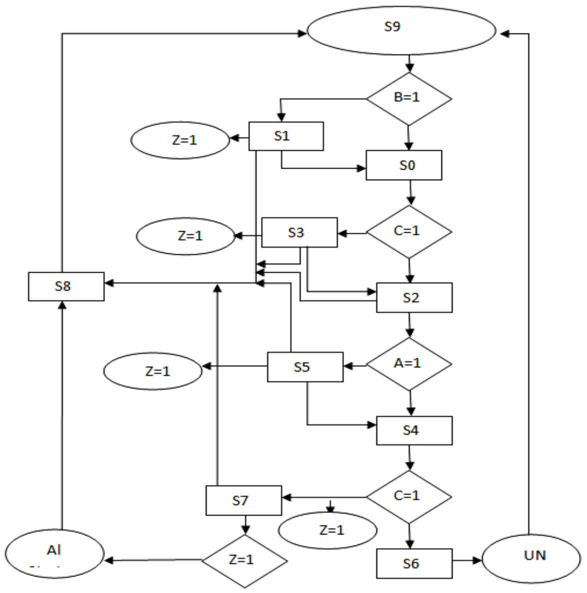

Figure8: Design flow of finite state machine

- **State machine description:**

This state machine is designed in ASM notation with NINE STATES and states are designed to fulfill the security protocols of the system. States descriptions are as follows:

<u>Initial state:</u> At the initial state, the system will be its RESET state and all the inputs and outputs will be

Volume 4, Issue 1
February 2015

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

zero. At this state, a condition check will examine the B input and if B is 1 here, the condition will be TRUE and next state will be S0. In other case, a signal Z would be asserted leading the system to state S1 and after approaching the state S1, system will be allowed to enter in state S0 but with high polarity of Z signal.

State S0: At state S0, a condition check will check the high polarity of C input and in case of TRUE condition, the state S2 will be approached otherwise signal Z will automatically be asserted and system will approach state S3. In state S3, system will be allowed to approach state S2 but with a high polarity of signal Z.

State S2: At state S2, a condition check will check the high polarity of A input and in case of TRUE condition, the state S4 will be approached otherwise signal Z will automatically be asserted and system will approach state S5. In state S5, system will be allowed to approach state S4 but with a high polarity of signal Z.

State S4: At state S4, a condition check will check the high polarity of C input and in case of TRUE condition, the state S6 will be approached and UNLOCK signal will be asserted otherwise signal Z will automatically be asserted and system will approach state S7.

State S7: At state S7, a condition check will be applied to Z signal and in case of high polarity of Z; ALARM signal will be activated enabling the LEDs or buzzer to be switched on. In this state if input A is asserted, the system will approach to state S8.

State S8: At state S8, if input A is asserted, the system will get back to its initial state having all the signals and inputs zero.

Effect of input C at any state: While designing the state machine, it is taken into account that at any state (except states S0 and S4 because at both states if C is asserted, system will approach to next state safely keeping the Z signal deactivated), if input C is pressed, the system approaches state S8 and once getting in this state, input A will lead the system to its initial position. It is clear from the state machine diagram that C input at any state, leads the system to state S8.

III. Results and Discussions:

Simulation results for this security lock can be seen in figure 9.

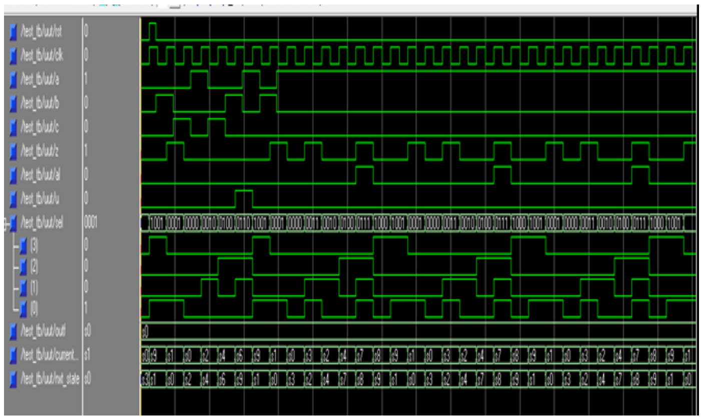

Figure9: Simulation results

It is clear from the waveform that when reset input is asserted, all the outputs are termed as zero and the current state is termed as “S9” while next state is “S0”. When combination of all four inputs goes wrong, a signal ‘Z’ is asserted that enables the alarm to be switched on and it can be seen in the waveform that whenever ‘Z’ signal is active high, the alarm signal also goes active high. ‘UNLOCK’ signal is only asserted by the system when state “S6” is accessed and at that moment, the current state is shown as state S6’s binary equivalent. Whenever a combination “C-A” is asserted, the system goes to its reset state “S9”. This is to be remembered that this combination does not work for states “S2” and “S4” because at both of the states, the “C” option leads the system to the next state as at these two states, according to the locking combination, the next state is achieved by pressing ‘C’. The place of ‘C’ in locking combination is also 2nd and 4th as our combination is “B-C-A-C”. When “Alarm” signal goes active high, the system approaches to its initial state by pressing a combination “A-A”. The pressing of first “A” leads the system to state “S8”. When “Unlock” signal is asserted, the system is lead to its initial state after a time of 10ns. This time can be increased anyway.

An alarm signal can be shown as blinking light emitted diode by the following two ways that are listed as follows:

- Divide the clocking period of the alarm signal. Anyhow this process is preferred.
- Generate another "Alarm2" signal that will be controlled by the original "Alarm" signal and the decision will be made as "WHEN (Alarm ='1' then Alarm2<= '1')" and then another clock can be defined which would have a small time period and that will only be used for the second alarm signal "Alarm2".
- By doubling the frequency, the clocking period can be reduced that makes a signal able to blink.

IV. Conclusions:

While dealing with complicated digital systems, it is a better technique to split the system in some blocks and write the code for each block individually as I did in this report. The simplicity in any complicated digital system can be engaged by the famous electronic work bench rule "Divide and conquer". Once we get the individual blocks for the system, we are able to locate any troubleshoot anywhere in the complete system as our overall system is divided in some easy blocks.

V .References:

1. http://en.wikipedia.org/wiki/VHDL
2. www.wjec.co.uk/uploads/publications/9297.do
3. asyncsymposium.org/async2009/slides/jones-async2009.pdf
4. Adnan shaout, Nevruskaja, Mikhail borovikov "security solution for cloud computing using a hardware implementation of AES" The university of Michigan, Dearborn.
5. Camel Tanougust, Abbas dandache, Mohamed Salah azzaz, said sadoudi "Hardware design of embedded systems for security applications", University of Lorraine of Metz, France.

Authors profile:

1. Awais Ahmed received his Masters degree in Electronics and information Technology in 2012 from USW, England. He has attained a 5 years research experience and working as a lecturer in school of electrical engineering, The University of Faisalabad, Pakistan.
2. Asim Shahzad received his Doctorate from DCU in 2009 in communication. He has attained a 10 years research experience and working as an Assistant professor in school of electrical engineering, The University of Faisalabad, Pakistan.