THOMPSON RIVERS UNIVERSITY

CENG 3010 – Digital System Design

SMART SAFE

Ekam Taneja (T00692000)

Harsh Sarvaiya (T00697722)

Toshiro Turner (T00684495)

Dec 7, 2023

Table of Contents

List of Figures

List of Tables

1 Introduction

Due to the recent shortage of security staff, and the rapid population growth, there has been a significant increase in the theft of commercial and personal property. Law enforcement agencies, businesses, and individuals all showed an increased demand for security products to safeguard their cash, jewelry and other valuable assets. Specifically, customers are asking for security safes with timed auto locking and programmable passcodes. Furthermore, customers are asking for a security safe to feel at peace when away from home or work. Since security is an important issue, there is a large and urgent demand for this product that we hope to fill.

In response to these rising demands, our team has come up with an appropriate design that caters to the user's needs while providing enhanced security. To document the design process, and provide an understanding of the decision making process in this project, our team has created this detailed report outlining the problem, requirements, different solutions, our roles in the team and many other aspects of the project. This report also provides a timeline of our project, and the possibilities for product refinement and the expansion for the future. The report ends with a personal reflection from each member of the team, and what they would change about the project in the future. Lastly, a section has been reserved for the necessary references and appendix.

2 Design Problem

2.1 Problem Statement

Law enforcement agencies are facing a surge in theft and robbery incidents due to rapid population growth and a shortage of security personnel. To combat this issue, our locking safe is equipped with a digital controller using Xilinx Basys 3 FPGA Board. The safe must not only prevent theft but also be interactive, accommodating various categories of valuables and providing detailed information about its usage.

2.2 Design Requirements

Our digital safe controller must fulfill a set of requirements to ensure we have addressed our problem effectively.

- Enhanced Security
    - The safe must have a numerical code based lock which prevents unauthorized access.
    - The safe must have some tamper detection which triggers an alarm after a predetermined set of incorrect attempts.

- Interactivity
    - The controller should provide an interactive display with features (such as the keypad and green light) allowing the user to quickly and efficiently access the safe's functionalities with ease.
- Efficiency
    - The design should avoid unnecessary digital elements, such as cosmetic leds, which will ensure an efficient and optimized solution.
    - The design should have a power-saving option where after a predetermined time, the safe's lcd screen will turn off to consume less power and to increase efficiency.
- Sequential Machine
    - At least one sequential machine inorder to have safe operations.
    - The design should have a log system which takes record of when different functions are executed.

2.2.1 Functions

- The device should have user authentication
    - Verify the identity of the user through secure authentication methods, such as PIN codes or biometric recognition
    - In the case where an unauthorized user attempts to access the safe, the safes in-built tamper detection system will activate a buzzer to notify the authorized user of an intruder.
- Locking and Unlocking
    - Safe door must check if it is closed before locking. If the door is not closed after a given time, an alarm will sound.
    - Code must check if the safe door is closed and lock itself if open for an extended period of time.
- Security alarms
    - Safe will ring if there are a series of failed attempts on the lock. This will alert people nearby that there might be people trying to break in.

2.2.2 Objectives

- The safe should be secure
    - Create a secure system that protects valuables effectively and deters theft.
    - Encrypt and decrypt the data being input and output in order to provide another level of boundary between user and system.
- The safe should be useable
    - An intuitive interface that allows users to quickly and effectively program and access the functionalities of the safe to their liking.
    - Multiple compartments to allow users to safely and effectively store differently shaped items within the confines of the safe.
- The safe should be functional
    - Ensure that all the specified functions of the safe are addressed and operate efficiently, providing a safe and reliable solution.
- The safe should be efficiently designed
    - Simplify and optimize the design in order to increase resource efficiency and minimize unnecessary components.

2.2.3 Constraints

- FPGA Compatibility (specifically Xilinx Basys 3)
    - Code must be able to be easily transferred to other FPGA devices for wide scale use
- Safety and compliance
    - Safe must not break under everyday hit, fall, and crush forces

    - Locking mechanism must be secure and must provide adequate protection against theft attempts
- Storage compartments
    - The design must have compartments of multiple sizes to accommodate items of different sizes.
- Efficient Design
    - Design must not have unnecessary digital elements in order for devices to run optimally.
        - This means optimizing code to take minimum space and increase speed using what we learned in this class and in Data Structures and Algorithms
- The code must have at least one sequential machine
    - Software must utilize the process function in VHDL with relevant sensitivity list and sequential code
- Project Budget
    - This safe must not take more than 10$ on top of allotted equipment (Xilinx Basys 3 FPGA Board and accompanying parts) to reproduce.
- Socio-economic constraints
    - Due to the setup and use of this safe, the user of this safe must have basic logic skills, and be able to read and identify english numbers. Also, they must have the capability of remembering or storing safely the combination passcode in case the RFID card is lost.

3 Solution

3.1 Solution 1

For the initial solution, we proposed a 3D printer-based design for the safe housing, door, and locking mechanism. These parts will all be printed using robust and durable material in order to ensure ample protection for the user's belongings. The design will use a simple, yet efficient, door hinge design to ensure a smooth user experience while accessing the items within the safe.

The safe's locking mechanism will be controlled by a servomotor within the structure of the safe. This solution ensures an FPGA compatible system by using a Xilinx Basys 3 FPGA Board which also allows us to easily program our system for the user's ease of use. The design will automatically lock the door after it detects the door being closed for a preset time.

Due to the 3D printer-based design, the safe will be extremely light, durable, water-resistant to tiny spills. In addition, the design allows it to be manufactured for very little cost and time, increasing its manufacturability. Finally even with the extremely cheap and light safe housing, the design still provides sufficient security and an exceptional user experience.

The cons of this design are mainly the lack of durability due to a plastic structure. In addition to this, the design isn't fully waterproof which lacks security.

3.2 Solution 2

This solution involves creating the safe out of sheet metal. The material is versatile and strong, which can be taken advantage of to create a durable, small-scale safe. Downsides to using sheet metal are its sharp edges and high tooling costs (welding, cutting, obtaining). The design will include a hinged door, keypad lock, LCD screen, and a servo-controlled locking mechanism. Any and all programming will be done through the Xilinx Basys 3 FPGA Board, coded in VHDL.

When the correct permutation of numbers is entered into the keypad, which is displayed on the LCD screen, the servo motor will rotate to disable the locking mechanism on the inside of the safe.

3.3 Final Solution

This solution involves creating the safe out of cardboard, and having the Xilinx Basys 3 FPGA Board along with features such as lever action switches, a numpad, and a locking mechanism. The switches would be the primary method of accessing the safe. By flipping the correct switches, a green led, depicting the locked state, will dim. Shortly after, the door locked by the servo motor would unlock to allow access to the compartments of the safe.

While the safe is locked, the safe will turn on a green light to depict that it is primed and locked. Once it is unlocked, all the user has to do is change the inputted sequence. The safe will lock again and require the correct input to unlock again.

This solution was chosen instead of the others since it scored highest on our decision matrix, while it was not the most durable and easily producible solution. It scored highest in safety and security. These categories are especially important to our design, as it aligns with our vision to provide a safe and secure solution for protection. Furthermore, this solution is also user-friendly and provides an intuitive user experience for people of all abilities.

These features will yield an ambitious safe that ensures proper safety and adequate security while providing a smooth user experience.

Table I Decision matrix chart for the considered alternatives

		Solutions					
		Solution 1		**Solution 2**		**Final Solution**	
Criteria	**Weight**	**Score**	**Partial Score**	**Score**	**Partial Score**	**Score**	**Partial Score**
Cost	0.40	8/10	0.320	6/10	0.240	9/10	0.360
Safety	0.25	5/5	0.25	3/5	0.150	5/5	0.250
Durability	0.20	9/15	0.120	14/15	0.187	9/15	0.120
Manufacturability	0.15	9/10	0.135	6/10	0.090	8/10	0.120
Sum	1.00		0.825		0.667		0.85

3.3.1 Features

- Numerical keypad entry - keypadEntry()
    - Accepts a numerical input and verifies it for safe access.
- LED indicator - controlLED()
    - Controls the red and green LEDs to signify the locked and unlocked states.
- LCD display - displayLCD()
    - Displays status messages such as “Locked”, “Unlocked”, “Incorrect Input” as well as display the live user input.

Test Cases:

User Authentication: Test for valid username and password.

PIN Code Entry: Test by setting and entering correct PIN code.

Alarm System: Test by triggering the alarm, then resetting it.

LCD screen: Test by checking all possible statuses.

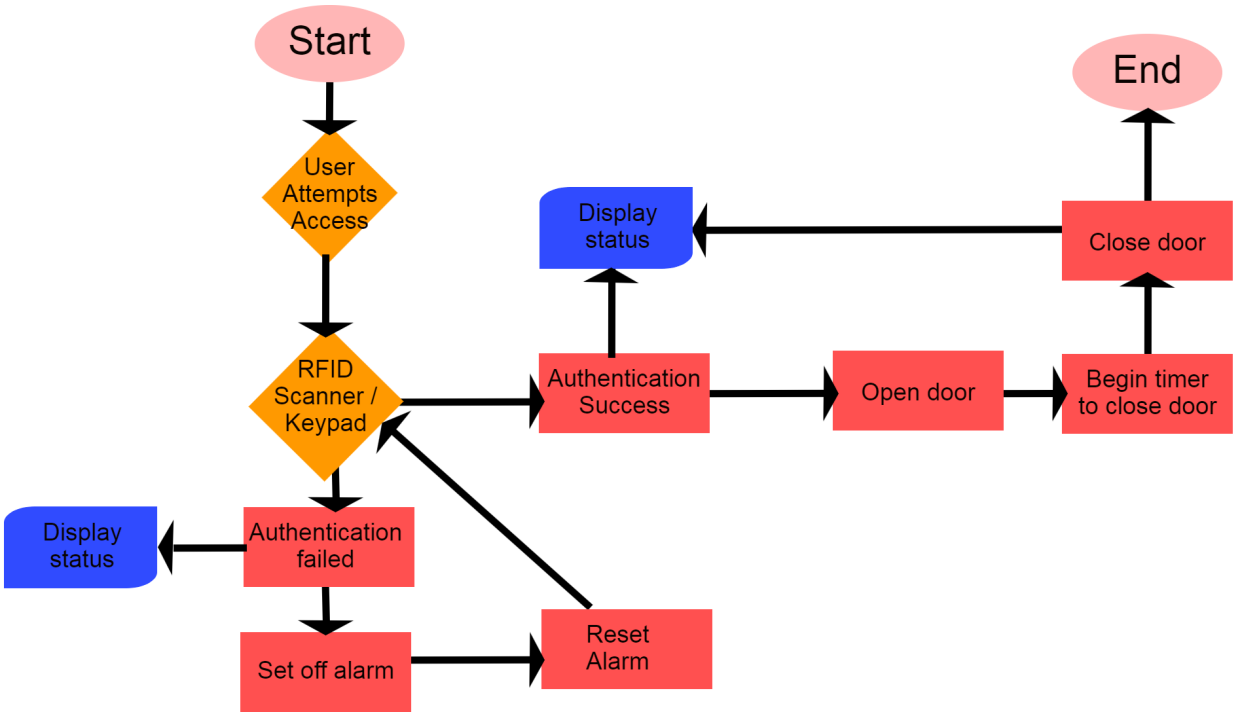

Figure 1: Flow chart process of using safe.

Give an account of all the features your prototype has and which functions or methods will enable those features. These features may be tabulated (with a title) for improved comprehension. For example, the structure of the program developed, including

- The specifications of the classes defined, and the public/private member functions/variables inside - explain as far as possible why your team makes such choices
- The flow of execution. (It is good to include a flow chart to illustrate it.)
- Testing of your program, which shows
    - How you validate your program, i.e. confirm that the solution is correct.
    - Write all test cases.

3.3.2 Environmental, Societal, Safety, and Economic Considerations

- Environment Considerations

The SMART SAFE design is an environment friendly design, as it uses very resource friendly components. The safe is made with recycled and reused cardboard for its structure. The accompanying xilinx board is an energy efficient processor, that can be reused multiple times due to its reprogrammable flexibility. The safe also does not use unnecessary cosmetic or aesthetic lighting or sound, and is dedicated to providing the necessary features with minimum overhead.

- Economic Considerations

The SMART SAFE design is a cost-effective design, as it uses components that are easily accessible at a cheap price. These materials are also sustainable and do not require a lot of electricity. The cardboard needed for this design will be acquired for free from supermarkets like walmart. The xilinx Basys 3 board retails for $150 CAD but can be used for many other projects and can be reprogrammed multiple times.

- Societal Considerations

The SMART SAFE design has been created to be accessible to people of all abilities. The design features an RFID card for those with eyesight problems, along with automatic opening and closing doors for those requiring cognitive and motor assistance. Furthermore, red and green indicator lights have been added to signal whether the safe is locked or unlocked.

- Safety and Health Considerations

The SMART SAFE design prioritizes safe use of our product. Our safe will include rounded corners to prevent any sharp or harsh edges. Furthermore, the assembly of the safe will be done in a clean and sanitary environment to avoid any biohazard/bacterias/contagions. Finally, a flashing red light will alert the user when the door is about to close, this has been implemented to avoid any pinching hazards.

3.3.3 Limitations

- Structural Limitations

The assembly of this prototype will be done in cardboard. While multiple layers of good quality cardboard will be used. The safe would be limited in its defense against physical and elemental attacks. The cardboard design would be susceptible to water damage, along with blunts and sharp attacks.

- Economic / Technological Limitations

Due to the budget and scope of this project, the project will not be able to have a large budget. While resourcefulness will be prioritized, some of the components such as the servo motor may occasionally have minor bugs.

4 Team Work

4.1 Meeting 1

Time: 6:30 PM

Wednesday Sept 27th

Agenda: Deliverable #1

Team Member	Previous Task	Completion State	Next Task
Harsh Sarvaiya	Introduction	100%	Solution 3 / fix feedback errors
Toshi Turner	Design problem	100%	Solution 2 / Decision Matrix
Ekam Taneja	Design problem	100%	Solution 1 / Features

4.2 Meeting 2

Time: 6:30 PM
Monday Oct 30

Agenda: Distribution of Tasks / Working on Deliverable #2

Team Member	Previous Task	Completion State	Next Task
Harsh Sarvaiya	Solution 3 / fix feedback errors	100%	Limitations/ Economic, Environmental, Societal and Safety Considerations
Toshi Turner	Solution 2 / Decision Matrix	100%	Feature flow chart
Ekam Taneja	Solution 1 / Features	100%	Features

4.3 Meeting 3

Time: 6:30 PM
Monday Oct 30

Agenda: Deliverable #2

Team Member	Previous Task	Completion State	Next Task
Harsh Sarvaiya	Limitations/ Economic, and Environmental Considerations	100%	VHDL Simulation /FPGA Implementation

Toshi Turner	Solution 2 / Decision Matrix	100%	VHDL Simulation /FPGA Implementation
Ekam Taneja	Solution 1 / Features	100%	VHDL Simulation /FPGA Implementation

4.4 Meeting 4

Time:8:00 PM - 8:30 PM

Agenda: Finishing prototype

Team Member	Previous Task	Completion State	Next Task
Harsh Sarvaiya	Create Hardware, Assemble final Prototype	100%	Finish prototype
Toshi Turner	Program Lever Action, and servo motor	100%	Finish prototype
Ekam Taneja	Create circuit, Create Hardware	100%	Finish prototype

5 Project Management

Programmable Lock Safe

Harsh Sarvaiya , Toshiro Turner, Ekam Taneja

Sept 25 2023 - Nov 10 2023

TASK	Start	Days	End
Project Research			
Brainstorming	Sep 25	8	Oct 3
Materials	Oct 4	1	Oct 5
Design Drawing/Planning	Oct 6	3	Oct 9
Standards/Requirements	Oct 10	1	Oct 11
Make Changes	Oct 12	2	Oct 14
Solidworks			
First Draft Design	Oct 15	2	Oct 17
Draft Improvement Brainstorming	Oct 18	3	Oct 21
Apply Improvements	Oct 22	2	Oct 24
Prototype Assembly			
Purchase Items needed	Oct 25	1	Oct 26
Discuss Building Procedure	Oct 27	2	Oct 29
Assemble Prototype	Oct 30	10	Nov 9
Group Report			
Introduce Project	Nov 10	Dec 31	Nov 11
Problems with Project	Nov 12	1	Nov 13
Solutions to Problems	Nov 14	3	Nov 17
Filling out Specifics	Nov 18	1	Nov 19
SlideShow	Nov 20	1	Nov 21
Final Edit	Nov 22	1	Nov 23
Video			
Final Polishes on project	Nov 24	1	Nov 25
Record with Project	Nov 26	1	Nov 27
Edit Video/Cool Transitions	Nov 28	1	Nov 29
Present in Class	Nov 30	1	Dec 1

6 Conclusion and Future Work

Future Work:

- Material Optimization: Investigate alternative materials or material reinforcements that can enhance the durability and security of the safe while maintaining cost-effectiveness.

- User Experience Enhancement: Gather feedback from potential users through prototypes and beta testing to identify areas for improvement in user experience. Consider incorporating additional features or refining existing ones based on user input.

- Security Upgrades: Explore advanced security features or technologies that could further strengthen the safe's protection against theft or unauthorized access. This may include biometric authentication, tamper detection, or advanced encryption methods.

Conclusion:

1. Problem Identification: Developed skills in identifying and analyzing real-world problems related to security and responded proactively to the increased demand for secure solutions.

2. Requirements Analysis: Gained experience in conducting thorough requirements analysis to understand the needs of law enforcement agencies, businesses, and individuals in the context of security safes.

3. Innovative Design Thinking: Applied innovative design thinking by integrating advanced technologies like the Xilinx Basys 3 FPGA Board, lever action switches, and a servo motor for enhanced security features.

4. Decision-Making Process: Developed and utilized a decision matrix to objectively evaluate different solutions and selected the design based on factors such as safety, security, and user-friendliness.

5. Interdisciplinary Collaboration: Worked collaboratively within a team with diverse skills and roles, including engineering, design, and project management, fostering interdisciplinary collaboration.

7 Self Reflection and Life-Long Learning

Harsh Sarvaiya: I was responsible for the hardware design and assembly of this safe, I was also the group leader and spearheaded the scheduling and the main design components of this project. I really enjoyed spending time designing the safe prototype to be sturdy and robust while also being intuitive and easy to use. Initially, I found the syntax of VHDL to be difficult to work with, but I was able to adapt and contribute to the programming after. One thing that worked really well was the depiction of the material choices in the final safe. We decided to use aluminium foil and a plastic bag to depict a metal outer and a plaster inner layer, respectively. This was meant to show the safe was sturdy and waterproof. Since we were not able to implement an alarm system in our first prototype, I would like to implement that in the next iteration.

Ekam Taneja: For the smart safe project, I mainly focused my efforts towards software design aspects. I mainly focused on trying to get different components working and spent my time trying to figure things out. I really enjoyed the trial and error aspect of learning to use Vivado. It would feel great when we would get a component working. In contrast, I found the programming in VHDL to be quite troublesome when first starting off. However once we learned functions in class, my previous knowledge of OracleSQL started to make connections with this new learning. What really worked well was splitting up. By splitting up our work load, we were able to get a lot more done than if we had worked together on one section. Next time we should probably spend more time planning as we were still making design changes very late into the development of the project.

Toshi Turner:
What I did?
I did most of the programming in VHDL using Vivado. I utilized pulse width modulator, clock slower, PMOD keypad, and Basys 3 lever entities and combined them as components in the main entity.
What I enjoyed:
I enjoyed programming in VHDL, especially when it functions properly after the first run.
What I found difficult:
Using the PMOD keypad was difficult, as there were multiple issues with the clock and timings to make it work to its full potential.
What really worked:
The lever combination worked in relatively few attempts and was programmable by the user with LED indicators.
Next time:
I would not use VHDL with the Basys 3 board, if I had the option not to.

8 References

"IEEE Standard for VHDL Language Reference Manual," in IEEE Std 1076-2019 , vol., no., pp.1-673, 23 Dec. 2019, doi: 10.1109/IEEESTD.2019.8938196.

Thingiverse.com, “Digilent Basys 3 Xilinx Artix-7 FPGA Trainer Board Case by NotSinaRoughani,” www.thingiverse.com. https://www.thingiverse.com/thing:3094814 (accessed Dec. 08, 2023).

9 Appendix

Safe Presentation

```
--leverLock.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity LLock is
        Port ( sw : in STD_LOGIC_vector(15 downto 0);
        led: out std_logic;
        pos: out std_logic_vector(6 downto 0));
end LLock;

architecture LLock_arch of LLock is
signal rightcode: std_logic_vector(15 downto 0) := "1000000000000001";
begin
        process(sw)
        begin
        if(sw = rightcode) then
        led <= '0'; --unlocked
        pos <= "1111111";
        else
        led <= '1'; --locked
        pos <= "0000000";
        end if;
        end process;

end LLock_arch;

--clk64kHz.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity clk64kHz is
    Port (
        clk    : in  STD_LOGIC;
        reset  : in  STD_LOGIC;
        clk_out: out STD_LOGIC
    );
end clk64kHz;
```

```
architecture Behavioral of clk64kHz is
   signal temporal: STD_LOGIC;
   signal counter : integer range 0 to 780 := 0;
begin
   freq_divider: process (reset, clk) begin
      if (reset = '1') then
         temporal <= '0';
         counter  <= 0;
      elsif rising_edge(clk) then
         if (counter = 780) then
            temporal <= NOT(temporal);
            counter  <= 0;
         else
            counter <= counter + 1;
         end if;
      end if;
   end process;

   clk_out <= temporal;
end Behavioral;

--mainServo.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity servo_pwm_clk64kHz is
        PORT(
        clk  : IN  STD_LOGIC;
        reset: IN  STD_LOGIC;
        sw: in std_logic_vector(15 downto 0);
        led: out std_logic;
        servo: OUT STD_LOGIC
        );
end servo_pwm_clk64kHz;

architecture Behavioral of servo_pwm_clk64kHz is
        COMPONENT clk64kHz
        PORT(
        clk      : in  STD_LOGIC;
        reset  : in  STD_LOGIC;
        clk_out: out STD_LOGIC
        );
```

```
        END COMPONENT;

        COMPONENT servo_pwm
        PORT (
        clk   : IN  STD_LOGIC;
        reset : IN  STD_LOGIC;
        pos   : IN  STD_LOGIC_VECTOR(6 downto 0);
        servo : OUT STD_LOGIC
        );
        END COMPONENT;

        component Llock
        Port ( sw : in STD_LOGIC_vector(15 downto 0);
        led: out std_logic;
        pos: out std_logic_vector(6 downto 0));
        end component;

        signal pos : std_logic_vector(6 downto 0) := "0000000";
        signal clk_out : STD_LOGIC := '0';
begin
        clk64kHz_map: clk64kHz PORT MAP(
        clk, reset, clk_out
        );

        servo_pwm_map: servo_pwm PORT MAP(
        clk_out, reset, pos, servo
        );

        lever_lock_map: Llock port map(
        sw, led, pos
        );
end Behavioral;

--pmod.vhd:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity PmodKYPD is
    Port (
                        clk : in  STD_LOGIC;
```

```
                        JA : inout  STD_LOGIC_VECTOR (7 downto 0); -- PmodKYPD is
designed to be connected to JA
        an : out  STD_LOGIC_VECTOR (3 downto 0));   -- Controls which position of the
seven segment display to display
end PmodKYPD;

architecture Behavioral of PmodKYPD is

component Decoder is
        Port (
                        clk : in  STD_LOGIC;
        Row : in  STD_LOGIC_VECTOR (3 downto 0);
                        Col : out  STD_LOGIC_VECTOR (3 downto 0);
        DecodeOut : out  STD_LOGIC_VECTOR (3 downto 0));
end component;

signal Decode: STD_LOGIC_VECTOR (3 downto 0);

begin

        C0: Decoder port map (clk=>clk, Row =>JA(7 downto 4), Col=>JA(3 downto 0),
DecodeOut=> Decode);

end Behavioral;


--servo.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity servo_pwm is
        PORT (
        clk   : IN  STD_LOGIC;
        reset : IN  STD_LOGIC;
        pos   : IN  STD_LOGIC_VECTOR(6 downto 0);
        servo : OUT STD_LOGIC
        );
end servo_pwm;

architecture Behavioral of servo_pwm is
        -- Counter, from 0 to 1279.
        signal cnt : unsigned(10 downto 0);
```

```
        -- Temporal signal used to generate the PWM pulse.
        signal pwmi: unsigned(7 downto 0);
begin
        -- Minimum value should be 0.5ms.
        pwmi <= unsigned('0' & pos) + 32;
        -- Counter process, from 0 to 1279.
        counter: process (reset, clk) begin
        if (reset = '1') then
        cnt <= (others => '0');
        elsif rising_edge(clk) then
        if (cnt = 1279) then
                cnt <= (others => '0');
        else
                cnt <= cnt + 1;
        end if;
        end if;
        end process;
        -- Output signal for the servomotor.
        servo <= '1' when (cnt < pwmi) else '0';
end Behavioral;

-- constraints.xdc
set_property PACKAGE_PIN W5 [get_ports clk]

        set_property IOSTANDARD LVCMOS33 [get_ports clk]
        create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports
clk]

set_property PACKAGE_PIN U18 [get_ports reset]
        set_property IOSTANDARD LVCMOS33 [get_ports reset]

set_property PACKAGE_PIN K17 [get_ports {servo}]
        set_property IOSTANDARD LVCMOS33 [get_ports {servo}]

set_property PACKAGE_PIN M18 [get_ports {led}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led}]

set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
```

```
set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
```