**DSD Project Report**

**Smart Packaging System**

**Team 86**

Mohamed ayman mohamed T18 58-22010

Omar Khaled T25 58-1891

Omar Ahmed T19 58-16880

Hamza T18 58-25367

Mohamed Kareem T16 58-1483

Khalid Wael T22 55-22068

Project Idea:

The project Idea is packaging system where you have a conveyor belt the carries and moves the products we intend to package, A buzzer for alerts (goes of when the sensor detects that no worker is operating the belt) , Two IR sensors detecting two physical parameters respectively (one detecting if a worker is operating the belt, the other for detecting the magnetic products to be packaged), FPGA that is programmed to the project logic having a 7 Segment display counting the products that got detected by the IR sensor.

The 7 segment display counts the products being passed when 5 products are passed by the conveyor belt when that happens the FPGA stops the movement of the conveyor belt motor furthermore when the other IR sensor detects that the worker isn’t around the buzzer is activated to alert the worker assigned.The worker then proceeds to package the 5 products then reset the system in order to package more products and allow the belt to move once more and the cycle repeats.

Inputs:

Clk which is used to represent the clock.

Reset\_n which is used by the worker in order to reset the system.

ProxSensor which represents the input from the IR proximity sensor detecting the products passing on the belt.

Humansens used to detect whether a worker is close to the conveyor belt or not.

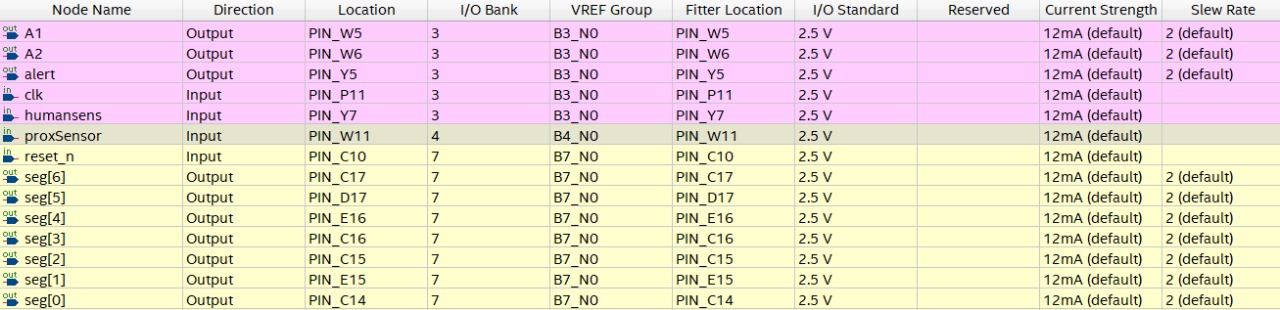
Outputs:

Seg which is a 7 bit number used to control the output to the 7-segment display.

A1 and A2 which is the output to control the A1 and A2 terminals on the H-bridge to control the motor.

Alert to output signal to the alert in order to turn it on and notify the worker that the package finished.

Pin Assignment:



library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity SmartPackagingSystem is

port (

clk : in std\_logic; **Clock input**

reset\_n : in std\_logic; **Active-low reset**

proxSensor : in std\_logic; **Proximity sensor input**

seg : out std\_logic\_vector(6 downto 0); **7-segment display output**

A1 : out std\_logic; **H-bridge input A1**

A2 : out std\_logic;

humansens : in std\_logic ;

alert :out std\_logic

);

end SmartPackagingSystem;

architecture Behavioral of SmartPackagingSystem is

signal counter : integer range 0 to 5 := 0; **Counter value**

signal seg\_data : std\_logic\_vector(6 downto 0);

signal clk\_divider : integer := 0; **Clock divider counter**

signal tick\_500ms : std\_logic := '0'; **0.5-second tick signal**

constant CLOCK\_FREQ : integer := 50000000; **FPGA clock** frequency (50 MHz)

constant TICK\_LIMIT : integer := CLOCK\_FREQ / 2; **0.5-second interval (50M / 2)**

signal motor\_active : std\_logic := '0';

begin

**Process to generate 0.5-second tick**

process(clk, reset\_n)

begin

if reset\_n = '0' then

clk\_divider <= 0;

tick\_500ms <= '0';

elsif rising\_edge(clk) then

if clk\_divider = TICK\_LIMIT - 1 then

clk\_divider <= 0;

tick\_500ms <= '1'; **Generate a tick**

else

clk\_divider <= clk\_divider + 1;

tick\_500ms <= '0';

end if;

end if;

end process;

**Process to handle proximity sensor, counter updates, and motor control**

process(clk, reset\_n)

begin

if reset\_n = '0' then

**Reset the counter, motor, and 7-segment display**

counter <= 0;

seg\_data <= "1000000";  **Display 0**

motor\_active <= '1'; **Start the motor**

alert<='0'; **Stop the buzzer**

elsif rising\_edge(clk) then

if tick\_500ms = '1' then **Only update every 0.5 seconds**

if proxSensor = '0' then **Assuming logic '1' indicates object detection**

if counter < 5 then

counter <= counter + 1; **Increment counter**

else

counter <= 0; **Reset counter after reaching 5**

end if;

end if; **This is added to slow down cycle to allow sensors to enough time to read**

**Motor control logic**

if counter = 5 then

motor\_active <= '0';

if humansens = '1' then

alert <= '1'; **Set alert if no human is detected**

elsif humansens = '0' then

alert <= '0'; **Clear alert if human is detected**

end if;

**Start the motor**

else

motor\_active <= '1';

alert<='0';

**Stop the motor and alert**

end if;

**Update the 7-segment display based on the counter value**

case counter is

when 0 => seg\_data <= "1000000"; -- Display 0

when 1 => seg\_data <= "1111001"; -- Display 1

when 2 => seg\_data <= "0100100"; -- Display 2

when 3 => seg\_data <= "0110000"; -- Display 3

when 4 => seg\_data <= "0011001"; -- Display 4

when 5 => seg\_data <= "0010010"; -- Display 5

when others => seg\_data <= "1000000"; -- Blank

end case;

end if;

end if;

end process;

**Assign motor control signals for H-bridge**

A1 <= not motor\_active; **High when the motor is active**

A2 <= motor\_active; **Low when the motor is active**

**Assign the 7-segment data to the output**

seg <= seg\_data;

end Behavioral;

For further explanation of our code and results kindly refer to the modelsim simulation video in out submission files

**Notes:**

Given that the purpose of the project is to have a smart packaging system for magnetic products it would have been more intuitive to use a magnetic sensor .. We have attempted this approach but unfortunately the magnetic sensor/switch seemed to be overcharged upon detection of our magnets therefore caused issues within our circuits meaning that it would overcount the number of products when detecting a single magnets thus messing with our counter and by extension our functionality.