Practical Lab Exam

Problem Set 2

Problem 1: Two-Bit Sequence Detection

1. Write VHDL code that implements the following functions:

- The entity operates with a 50 MHz system clock (**CLK**) suitable for synchronous FPGA design. The clock signal is provided by an onboard clock generator.
- The design includes an asynchronous, active-low reset input (**RESET_N**). Use an active-low push button for this signal.
- The FSM receives a two-bit input signal, **SW[1:0]**, driven by two slide switches on the FPGA board.
- The debounced and registered SW signals should be used as inputs to the FSM. A slower clock can be used to capture the SW input signal.
- The onboard LEDs, **LEDS[4:0]**, are used for the output signals. Each LED corresponds to a state of the FSM, indicating whether the FSM is active or inactive in that state.
- State transitions occur according to the state transition diagram or the table given below. S0 is the initial state and S4 is the end or final state.
- When the FSM reaches the final (end) state, the LED output should be HIGH; otherwise, it should remain LOW.
- Use two VHDL processes to implement the FSM: one process for the next-state logic (combinational), and another process for the state update (sequential).

2. Create a project in Quartus Prime Lite, implement the design, upload the bitstream to the **Terasic DE0-CV board**. Add the **Signal Tap Logic Analyzer** to your design and use it for on-chip debugging.

3. Test your design using the FPGA board. Demonstrate the correct operation of your design to the lab instructor.

Current State	**SW[1:0]**	**Next State**
S0	00	S1
S0	others	S0
S1	01	S2
S1	others	S1
S2	11	S3
S2	others	S2
S3	10	S4
S3	others	S3
S4	--	S4

VHDL entity template

```
ENTITY top IS
  PORT (
    CLK     : IN  STD_LOGIC;
    RESET_N : IN  STD_LOGIC;
    SW      : IN  STD_LOGIC_VECTOR(1 DOWNTO 0);
    LEDS    : OUT STD_LOGIC_VECTOR(4 DOWNTO 0)
  );
END top;
```

Problem 2: PWM-based Brightness Adjustment of an LED Array

1. Write VHDL code that demonstrates the following behavior:

   - The entity operates with a 50 MHz system clock (**CLK**) suitable for synchronous FPGA design. The clock signal is provided by the onboard clock generator.
   - The design includes an asynchronous, active-low reset input (**RESET_N**).
   - Use an active-low push button for this signal.
   - The design features an N-bit LED output, **LEDS[N-1:0]**, utilizing the onboard LEDs.
   - A PWM signal is used to control the brightness of the active LED.
   - Initially, all LEDs are OFF. After reset, the rightmost LED gradually increases its brightness from 0% to 100%. Once it reaches full brightness, the next LED begins to change brightness in the same way, and this process continues sequentially until the last LED and all LEDs are fully turned ON.

2. Create a project in Quartus Prime Lite, implement the design, upload the bitstream to the **Terasic DE0-CV board**.

3. Test your design on the FPGA board. Demonstrate the correct operation of your design to the lab instructor.

VHDL entity template

```
ENTITY top IS
    GENERIC (
        N : INTEGER := 10 -- number of onboard LEDs
    );
    PORT (
        CLK     : IN STD_LOGIC;
        RESET_N : IN STD_LOGIC;
        LEDS    : OUT STD_LOGIC_VECTOR(N-1 DOWNTO 0)
    );
END top;
```

Problem 3: Sending Bytes Using the Serial Protocol

1. Write VHDL code that shows the following behavior:

- The entity operates with a 50 MHz system clock (**CLK**), suitable for synchronous FPGA design. The clock signal is provided by the onboard clock generator.
- The design includes an asynchronous, active-low reset input(**RESET_N**). Use an active-low push button for this signal.
- The design accepts 8-bit slide switch inputs, denoted as **SW[7:0]**.
- Every **100 msec**, the current value of **SW[7:0]** is captured and transmitted through the **TX output pin** using the serial protocol (**8N1**, **9600 baud**).

2. Create a project in Quartus Prime Lite, implement the design, upload the bitstream to the **Terasic DE0-CV board**.

3. Test your design on the FPGA board. Use a **digital oscilloscope** to capture the TX waveform and verify the correctness of your design. Demonstrate the correct operation of your design to the lab instructor.

VHDL entity template

```
ENTITY top IS
    PORT (
        CLK     : IN STD_LOGIC;
        RESET_N : IN STD_LOGIC;
        SW      : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        TX      : OUT STD_LOGIC
    );
END top;
```

Problem 4: VGA Display

1. Write VHDL code that implements the following functions:

- The entity uses a 50 MHz system clock (**CLK**) for a synchronous FPGA design. The clock signal is provided by an onboard clock generator.
- It has an asynchronous active-low reset input (**RESET_N**) to reset the design. Use an active-low push button for this signal.
- The entity provides **4-bit per channel R, G, B outputs** for the VGA display, plus HSYNC (**VGA_HS**) and VSYNC (**VGA_VS**) signals required by VGA.
- It has a push button input (active-low), **BTN**, that toggles the display. The display is initially blank.
- On a push-button click (pressed and released, after debouncing), **toggle** between two modes:
  - Blank screen mode
  - Display mode showing the **Thai national flag in landscape orientation**.
- The push button must be debounced and sampled synchronously; implement edge detection so one click = one toggle.

2. Create a project in Quartus Prime Lite, implement the design, upload the bitstream to the **Terasic DE0-CV board**.

3. Test your design by connecting the VGA connector of the FPGA board to an LCD screen that supports VGA in the lab. Demonstrate the correct operation of your design to the lab instructor.

VHDL entity template

```
ENTITY top IS
  PORT (
    CLK     : IN  STD_LOGIC;
    RESET_N : IN  STD_LOGIC;
    BTN     : IN  STD_LOGIC;
    VGA_HS, VGA_VS : OUT STD_LOGIC;
    VGA_R, VGA_G, VGA_B : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
  );
END top;
```