American University of Beirut
Maroun Semaan Faculty of Engineering and Architecture
Department of Electrical and Computer Engineering

EECE 320: Digital Systems Design
VHDL Project

As a new engineer at State Machines, Inc., your first assignment is to design a multiple-of-three sequence detector. The detector should have three inputs: `i_clk`, `i_reset_n`, and `i_serial`; and one output: `o_mul3`. Upon reset, the detector should monitor its serial input port (`i_serial`) and assert its output each time the bits of the input represent a binary number (of arbitrary length) that is a multiple of three. Assume the number is entered from most to least significant bit.

The `i_reset_n` signal should be asynchronous; it can be asserted at any time. On the other hand, the `i_serial` signal should be synchronous. Although it can be toggled at any time, it should not produce an output until the next active clock edge.

Detector Design

Please respond to the following questions. Provide your answers in a typed report and include listings of your VHDL models and testbench.

1. Provide a state diagram for your controller.
2. Derive the corresponding state-assigned table. Do not use a one-hot state encoding.
3. Check if you have any redundant states. Show your work, and update your state diagram and state-assigned table accordingly.
4. Implement the controller using D-FFs. For each flip-flop in your controller, derive the simplified logic equations for its D input. Also derive the simplified logic equations for the controller output. Show your work.
5. Draw a diagram of your controller circuit.

6. Develop a **structural** VHDL model of your controller. Use the following top-level entity and the D-FF VHDL model posted on Moodle as a component.

```
entity mul3Detector is
  port (
    i_clk : in std_logic;
    i_reset_n : in std_logic;
    i_serial : in std_logic;
    o_mul3 : out std_logic
  );
end mul3Detector;
```

7. Develop a suitable testbench (`mul3DetectorTB`) to test your model. Your testbench should verify that the controller will be reset as soon as the `i_reset_n` signal is asserted and that it will assert its output for multiple-of-three inputs of arbitrary length.

   Use the following process to generate 20 cycles of the clock signal. You can modify the code to generate more clock pulses if need be:

```
process
begin
  for i in 1 to 20 loop
    tb_clk <= '0';
    wait for 10 ns;
    tb_clk <= '1';
    wait for 10 ns;
  end loop;
  wait;
end process;
```

   Note that `tb_clk` should be declared in the testbench architecture and mapped to the controller's `i_clk` port.

8. Now develop a **behavioral** VHDL model of your controller. Use the same entity name (`mul3Detector`) for your behavioral model, and use the same testbench to validate it.

Deliverables

Please include the following in one folder, compress it, and submit it to Moodle:

1. A typed report with your answers to questions 1-8. Please include listings of your structural VHDL model, behavioral VHDL model, and testbench in your report.
2. Source code models of your structural VHDL model, behavioral VHDL model, and testbench.
3. This project is due by **Sunday, December 4, 2022 at 11:59 PM**. Late submissions will not be accepted.