Purpose

The purpose of this lab is to: (1) familiarize yourself with VHDL behavioral descriptions and especially such descriptions of sequential circuits (i.e., finite state machines or FSMs); (2) Describe a clocked system, i.e., a circuit that operate in synchrony with a clock. You will achieve these by behaviorally describing an FSM that is triggered on the positive edge of a clock.

Design Problem

You need to design a Moore FSM with the following specifications:

1. It has two bit-serial inputs $x$, $y$, and has a single output $f$.

2. It outputs a 1 on $f$ whenever it has received a multiple of 4 number of 1’s on $x$ and a multiple of 3 number of 0’s on $y$.

Provide your design as follows:

- A neatly drawn state-transition diagram with mnemonically labeled states that indicate the meaning of the state (also, be sure to label the reset state as such). Separately provide a description of the purpose/functionality of each state. Finally, provide a concise rationale and description of your FSM design (refer to the state labels to describe your design).

VHDL Description

Describe your designed FSM in VHDL using multiple processes as discussed in class. The FSM should be positively edge triggered. See the lecture notes on “Sequential Circuits (FSMs) and Their VHDL Descriptions” for FSM descriptions in VHDL.
You need to have appropriate signal assignment statements with delays to simulate a state-transition logic delay of 5 ns, output logic delay of 3 ns and a FF delay of 2 ns.

You also need to have a clock process that generates a clock with a period of 10 ns and a duty cycle of 50% (i.e., high period of 5 ns). Such a clock process is given below.

```vhdl
architecture behav of pattern_det_fsm is
signal clk: std_logic := 0;
other declarations

clock: process is
begin
clk <= '0','1' after 5 ns;
wait for 10 ns ;
end process clock;

end architecture behav;
```

**Simulation**

Design a TB that feeds the appropriate input bit strings on the two lines $x, y$ to your FSM; the bit strings will be given by the TA. From the two bit strings, your TB will need to generate a bit pair (for $x, y$) for feeding to your instantiated FSM design every negative edge of the clock.

Note that by feeding the input to the FSM every negative edge, we ensure that the FSM will be able to reliably sample it at the clock’s positive edge.

Provide the timing graphics of the inputs and the outputs corresponding to the input bit strings provided to you.