1. Find the following code examples from the textbook and copy them into your pre-lab.

Counter - Example 5.5

HDL Example 5.5 COUNTER

SystemVerilog

```
module counter #(parameter N = 8)
               (input  logic clk,
                input  logic reset,
                output logic [N-1:0] q);

  always_ff @(posedge clk, posedge reset)
    if (reset) q <= 0;
    else       q <= q + 1;
endmodule
```

Comparator - Example 5.3

HDL Example 5.3 COMPARATORS

SystemVerilog

```
module comparator #(parameter N = 8)
                   (input  logic [N-1:0] a, b,
                    output logic eq, neq, lt, lte, gt, gte);

  assign eq  = (a == b);
  assign neq = (a != b);
  assign lt  = (a < b);
  assign lte = (a <= b);
  assign gt  = (a > b);
  assign gte = (a >= b);
endmodule
```

Synchronizer - Example 4.20

HDL Example 4.20 SYNCHRONIZER

SystemVerilog

```
module sync(input  logic clk,
            input  logic d,
            output logic q);

  logic n1;

  always_ff @(posedge clk)
    begin
      n1 <= d; // nonblocking
      q <= n1; // nonblocking
    end
endmodule
```

Display Driver - Example 4.24

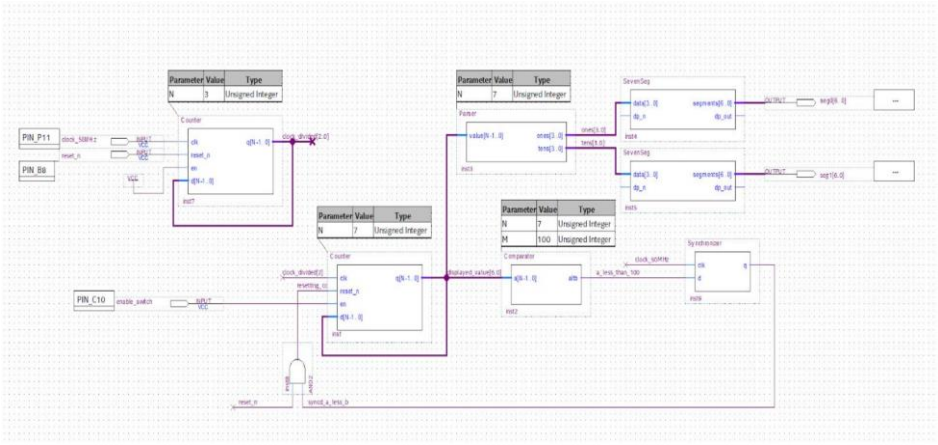
SystemVerilog

```
module sevenseg(input  logic [3:0] data,
                output logic [6:0] segments);
  always_comb
    case(data)
      //                    abc_defg
      0:       segments = 7'b111_1110;
      1:       segments = 7'b011_0000;
      2:       segments = 7'b110_1101;
      3:       segments = 7'b111_1001;
      4:       segments = 7'b011_0011;
      5:       segments = 7'b101_1011;
      6:       segments = 7'b101_1111;
      7:       segments = 7'b111_0000;
      8:       segments = 7'b111_1111;
      9:       segments = 7'b111_0011;
      default: segments = 7'b000_0000;
    endcase
endmodule
```

Which of these blocks are combinational logic and which are sequential?

Display driver and Comparators are combination logic, and Counter and Synchronizer are sequential logic.

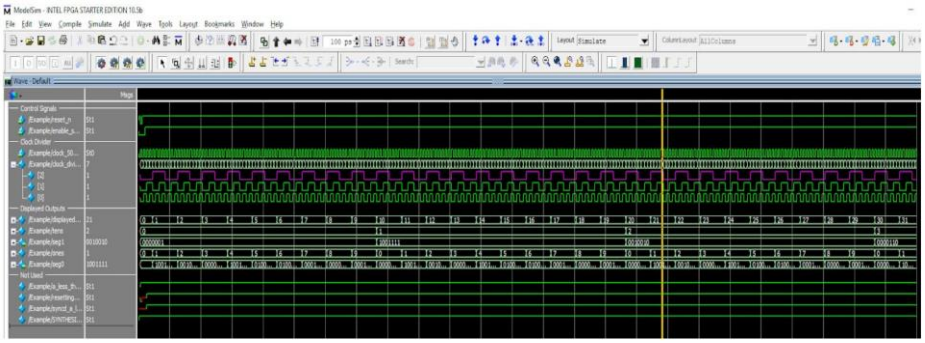
2. Write a paragraph describing the operation of ExampleCounter.png



ExampleCounter.png

First of all, 50MHz clock is connected to 3-bit counter. This 3-bit counter will act as clock divider. Then the output of 3-bit counter will be connected to the clock input of 7-bit counter. Similar to 3-bit counter, this counter will act as clock divider. After 7-bit counter, the output will be connected to parser and comparator. The parser will split the binary inputs to represent it in decimals, and comparator will make an output zero when the input value is over zero. The synchronizer will harmonize the output of comparator with 50MHz clock. Then the output of synchronizer is connected to AND gate with reset input of 7-bit to determine reset the 7-bit counter or not because the reset of 7-bit counter is active low so that if the output of synchronizer is zero when the output of 7-bit counter reaches 100. By two counter which are 3-bit counter and 4-bit counter and 50MHz clock, the input value is incremented, and then because of comparator and synchronizer, the counting is controlled not to reach 100. Hence, because of these combinations: adders, comparator, and synchronizer, the input by 50MHz clock is represented in decimals by parser.

3. Review the ExampleSimulation.png What does the Parser do?



ExampleSimulation.png

The parser splits the binary input value by ones-place and tens-place in decimal, and then sends those value to seven segment displays to each display driver in order to represent the input value in decimals.