

## COL215 Assignment 2

**Title:** Digital Clock for displaying time of the day

### **Introduction:**

This exercise involves use of VHDL for describing design of a digital clock. The design is to be done keeping in view the inputs and outputs provided by a specific type of FPGA board that is commonly used in our Lab.

### **Specification:**

Design a digital clock that displays the time of the day in hours, minutes and seconds. Make a provision for user to set the time to a desired value. The design has to work with the following inputs and outputs provided by the BASYS3 FPGA board.

- 4 seven-segment displays, each one can display a digit, optionally with a decimal point
- 5 push-button switches are available, but try to use as few as possible, keeping in mind convenience of the user
- A 10 MHz clock signal

Assume a 24 hour format for the time, that is, time would range from 00:00:00 hr:min:sec to 23:59:59 hr:min:sec.

Your submission needs to include the following.

- Overview of your design, including all your major design decisions
- Design description in VHDL
- Explanation of VHDL code, including meaning of each signal used

### **Design Considerations:**

#### **A. Display**

Clearly, with the limit of 4-digit display, you cannot simultaneously display hours, minutes as well as seconds. Therefore, you need to define multiple display modes. Different parts of the information can be displayed in different modes. You may utilize the decimal points available with each digit of display in some useful manner. For example, in a display mode in which seconds are not displayed, a dot flashing at the rate of 1 Hz could be useful. Refer

to Lecture 07 (13<sup>th</sup> Oct, 2020) for details of the circuit required to drive the displays of BASYS3 board.

## **B. Time setting**

Note that there is no key-board available to specify the hour, minute or second values to be set. With push-buttons, it makes sense to increment/decrement hours, minutes or seconds as the user pushes them. It is convenient for the user if hours, minutes and seconds can be set independently and even more convenient if each digit can be set independently. You may also consider allowing slow increment/decrement (change by one when the button is pressed and released once) for small adjustments and fast increment/decrement (keep changing while the button is kept pressed) for large adjustments. Define multiple time setting modes as per your design choices. Mode changing will also require use of push button (s).

## **C. Timing**

In this circuit, there are many periodic activities with different repetition rates, as listed below.

- Time is updated every second.
- Display requires a refresh period of 1 to 16 ms.
- Since there are 4 digits to be displayed, the digits need to be scanned at 4 times the refresh rate.
- If a dot flashing every second is to be displayed, a signal toggling at the rate of 2 Hz will be required.
- If a digit is to be incremented/decremented in a fast mode (see section B above), it should happen at the rate of around 4-5 per second.

You can either work with multiple clock signals of appropriate frequency or have a single master clock that triggers various activities after appropriate counts of its pulses. The second approach leads to a design that is easier to test and debug. 100 MHz clock available on BASYS 3 board needs to be divided by a suitable number to get the master clock. Note that a modulo N counter divides frequency by N.

## **D. VHDL style**

Do not consider VHDL as just another programming language. As you design, do not think

in terms of VHDL constructs, rather think in terms of circuit structures and describe these in VHDL. This will help you in not losing the sight of the underlying hardware. Restrict yourself to the limited subset of VHDL we have discussed in the class. Avoid temptation to use other VHDL features like variables, loops, wait statements etc.

Break-up your circuit into a few major sub-circuits and describe each as an entity-architecture pair. Put these together using instantiation statements. Clearly identify combinational and sequential parts in each entity. Make all sequential parts as synchronous. Asynchronous inputs may be used for initialization, but not for main activity.

Describe combinational parts using concurrent signal assignments, conditional signal assignments, selected signal assignments or processes. In order to ensure that no latches get formed inadvertently, ensure that the following conditions are met.

- There is no cyclic dependence among the statements describing combinational circuits.
- Every signal that is a destination of some assignment, gets assigned under all conditions.
- Each process includes all its input signals in its sensitivity list.

Describe the sequential parts as processes triggered by the master clock (except for the clock divider process that generates the master clock). These processes will usually be sensitive only to the clock input. Other signals may be included in the sensitivity list if required for initialization as shown below.

```
PROCESS (clock, init)
BEGIN
  IF init = '1' THEN
    . . . do the initialization here . . .
  ELSIF clk = '1' AND clk'EVENT THEN
    . . . do all the clock triggered assignments here . . .
  END IF;
END PROCESS;
```

Remember that all sequencing is to be achieved by state transitions, not by putting statements in a sequence inside a process.

Also ensure that each signal has at most one driver. That is, each signal is assigned in not more than one concurrent statement. If that statement is a process, there can be multiple assignments to that signal within it.