MINI-PROJECT

Goals: Practice and understanding:

- ***Edge trigger and Level trigger***
- ***Blocking and non-Blocking assignment.***
- ***Experiments on real FPGA technology – DE2I150 board***

Requiremnents:

- **Submit all source files (.v, .bdf) and a report file (.pdf). The report file should have these chapters:**
    - **Chapter 1 : Introduction – Introduce your topic, goals, software/hardware used and its specification. Instructions on how to use your digital clock (Which SWs, LEDs, BUTTONs are used and its purposes).**
    - **Chapter 2: Design and Implement – Include your block diagrams, RTL viewer, flow-charts, state machines.**
    - **Chapter 3: Experiments – Simulation/waveform results along with testcases and explanations. Images and videos which demonstrate your system on FPGA board.**
    - **Chapter 4: Conclusion and Future Work – State your conclusion, future improments, strong/weak aspects of your implementation. Table of duty roster.**
- **Compress as *.zip file and submit onto e-learning system.**
- **Deadline : 19/06/2020. You should print your report and hand it to your instructor on that day!**

Overview

We are going to implement a basic digital clock based on 50Mhz clock pulses of DE2I-150 FPGA board.

Below are the requirements and also instructions which are useful for your project.

You can use clock divider module to generate 1Hz clock from 50MHz clock. This 1Hz then can be used for second counter. Use buttons to configure/adjust hour, minute, and second values.

Display results (hhmmss) on 7-seg LEDs. Here is an example block diagram, if you want to use this block diagram in your report, please re-draw it.

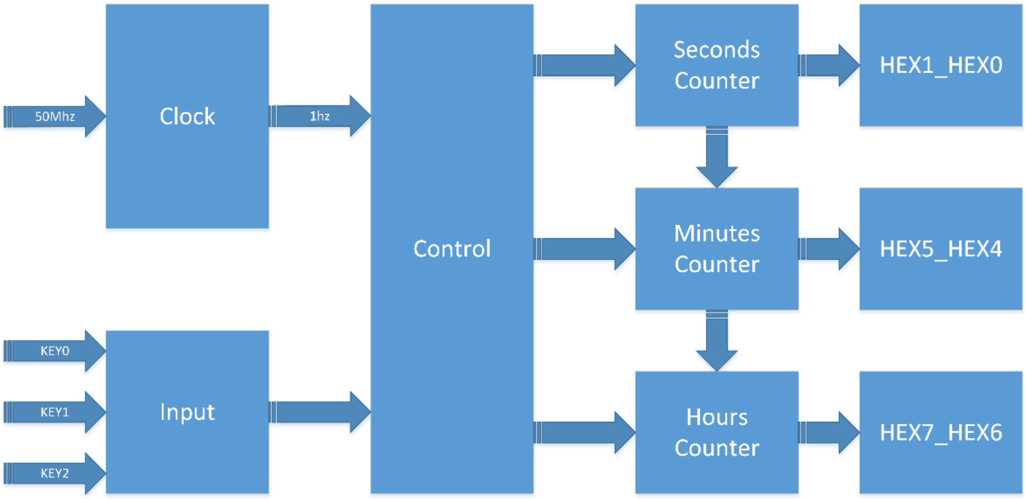

Figure 1 Digital clock design

Problem

1.

Design and implement your clock divider module which generate 1Hz pulse from 50MHz pulse of DE2I-150 board. Write testbench to check if your module function correctly, then install into DE2I-150 board (LEDR blinking test).

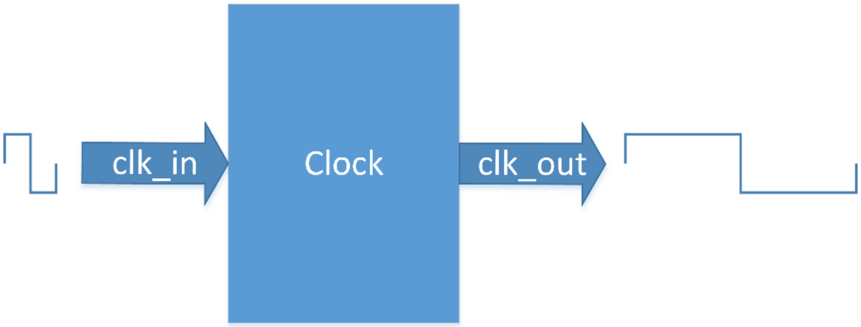

Figure 2 Clock divider module

Hint: You can use clock divider module implemented in Lab3 and set hz_in = 1.

2.

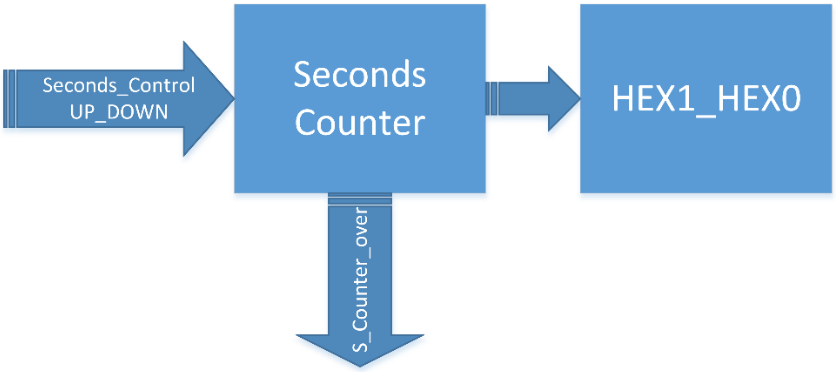

Figure 3 second counter module

Design second counter module which can count from 0 to 59 and control the 7-seg LEDs. The input signal can control this module to count up or count down. Write testbench for this module, then install to DE2I-150 board for testing.

KEY0 and KEY1 are UP and DOWN input signals. S_Counter_over is output signal which should be showed in LEDR. With every active pulse of inputs, counter will increase/decrease by one unit.

When counter value exceed 59, the counter then reset its value to 0 and the S_Counter_over will active as to notice that one minute is passed (this signal will be used as input for minute counter module).

Hint: This module can count up/down depends on input control. For 7-seg LED, you can reuse module implemented in Laboratory.

3.

Implement the three modules which are hour, minute, second counter modules and display results on 7-seg LEDs. Picture below as an example

Write testbench to test your circuit, you can use the below interfaces:

KEY0 -> UP Seconds.

KEY1 -> UP Minutes.

KEY2 -> UP Hours.

KEY3 -> DOWN Seconds.

KEY3 -> DOWN Minutes.

KEY3 -> DOWN Hours.

(When KEY3 is pressed, all 3 values are decreased by 1)

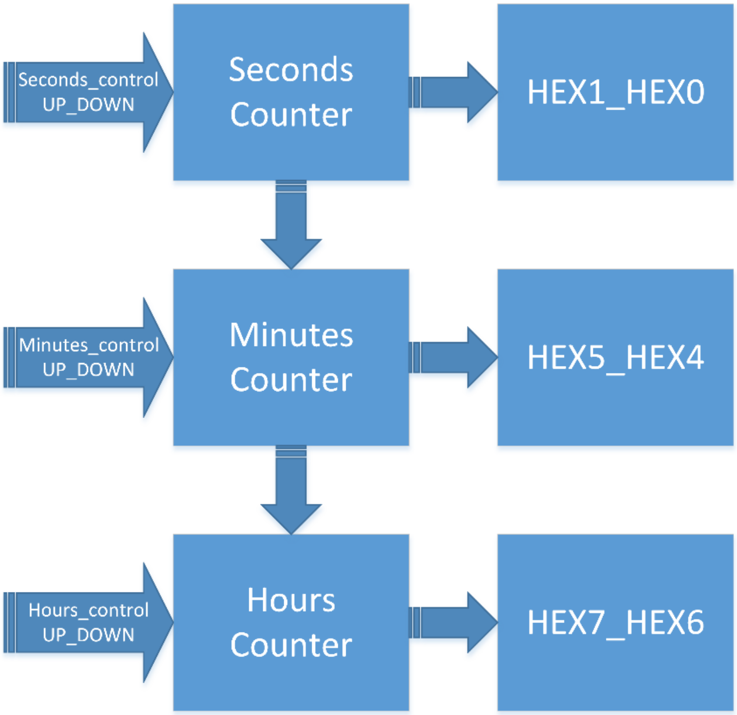

Figure 4 time counter

Hint: The minute counter module is similar with second counter module. This module receive trigger signal which is S_Counter_over from second counter module to increase minute values. Use the same strategy for hour counter module (but only count to 23).

4.

Implement a controller to control the counter. This module allow to setup and adjust hour, minute, and second values. This module could have four states: IDLE, Seconds_setup, Minutes_setup, Hours_setup.

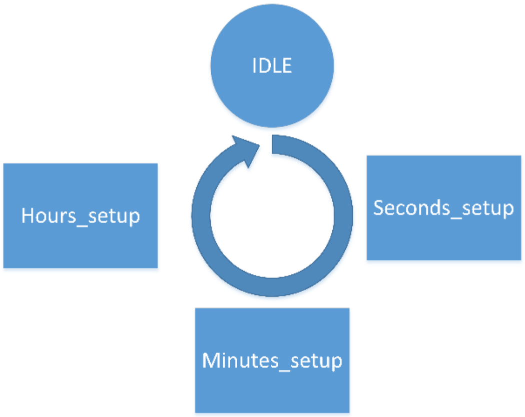

Figure 5 State transition diagram

There are 4 major input control signals: CLK_1, mode, up, down.

CLK_1: clock 1 hz.

mode: input for selecting/changing between states.

up: increase counter value by 1.

down: decrease counter value by 1.

If “mode” is trigger, this module will change state to the next state as showed in the diagram.

In IDLE state, second counter module will count up every second using 1Hz pulse as normal clock does.

In Seconds_setup state, second counter will paused because its value is now depended on “up” / “down” input signal. Similarly, apply these concepts for Minutes_setup and Hours_setup states.

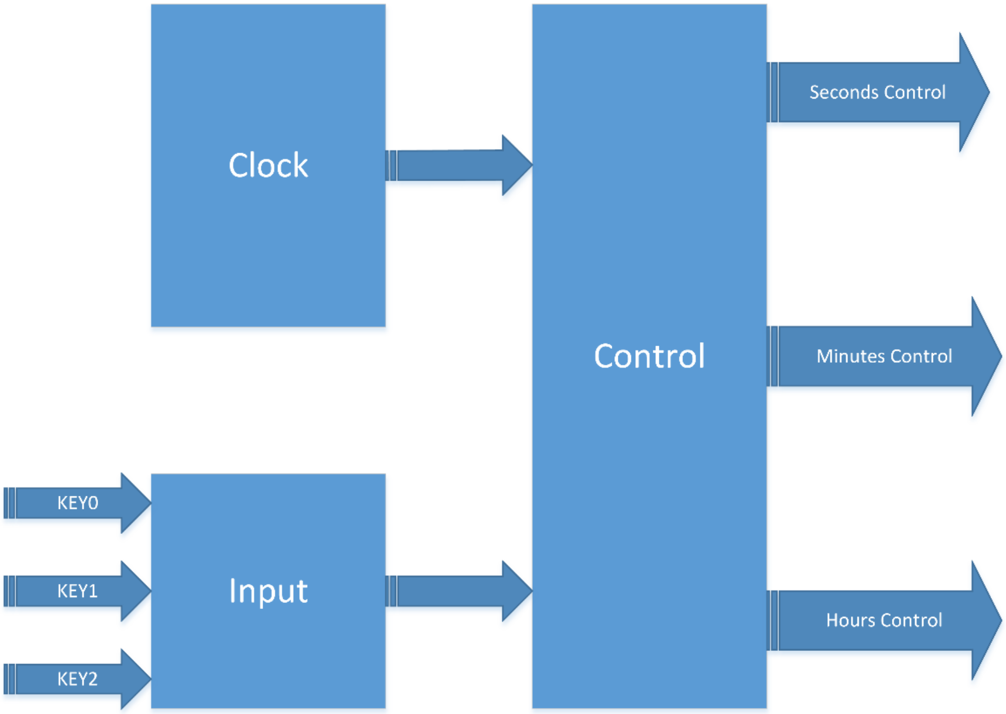

Figure 6 Control block diagram connections

Hint: You could use Finite-state machines or multiplexer on this module.

5.

Connect all above modules to have a complete Digital Clock implementation. Install to DE2I-150 board for evaluation.

Besides the requirements described in this file. You can add more functionalities to your system. For instance: LEDs blinking when setup, alarm clock, sport clock, etc.

Please note that, If you have better idea of how to design and implement this digital clock, you are free to use it. However, your changes should be carefully explain in your report file.