

Worcester Polytechnic Institute

Electrical and Computer Engineering Department

EE3801 – Lab3 (Dual Stop Watch)

Copyright: Feb, 2005

Written by Ahmad Hatami
Modified by Samant Kakarla

Please have all individual components created in this lab available to all parties working together to complete the lab. Make sure everyone also has the components available individually from previous labs. It is a good idea to backup your data on another media apart from the network space you have available.

Objective

In this lab you will design a dual stop watch.

Big Picture

Figure 1 shows the overall picture of the system. The system consists of two stopwatches A and B. You are going to use the Spartan3 starter board for this design. The least significant digit displays tenths of a second. The most significant digit displays minutes, and the two middle digits represent tens of seconds and seconds respectively. Here we describe the behavior of the system.

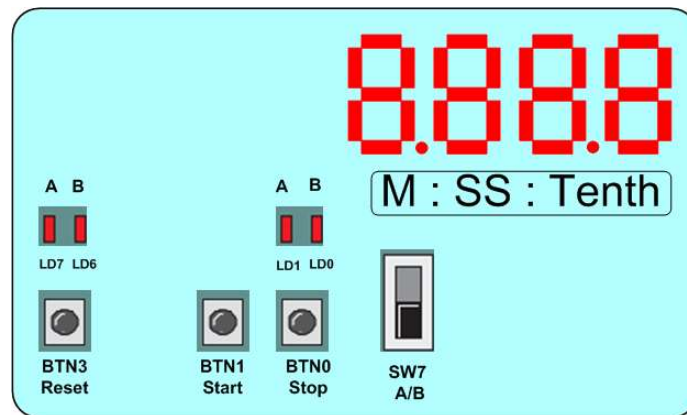


Figure 1

A/B (*timer_a*): This signal selects which one of the stopwatches are selected at the moment. (Zero selects stopwatch 'A' and one selects stopwatch 'B').

Start: Activating this signal has different effects on the selected stopwatch ('A' or 'B'). If the selected stopwatch was originally idle, it starts counting. If the timer was stopped, it resumes counting.

Stop: If the selected timer is idle, it does not have any effect. If the selected timer is counting it stops the counter.

Reset: Resets the selected counter and puts it in idle state.

LD1: This LED blinks when stopwatch A is counting. (Note that it does not matter whether stopwatch A is selected or not)

LD0: This LED blinks when timer B is counting. (Note that it does not matter whether stopwatch B is selected or not)

LD7: This LED is turned on when stopwatch A is selected; otherwise it must be turned off.

LD6: This LED is turned on when stopwatch B is selected, otherwise it must be turned off.

Prelab

Refer to class notes about flip flops, sequencer, counters, and state machine design. Design your Stop Watch Controller state machine (Steps 1 & 2) as well as you mod 10 counter (Steps 9 & 10) *before* going to lab.

Design

It is essential that you break down this system to smaller subsystems and start your design by implementing, and testing the small subcomponents, and at the end put the overall system together. Figure 2 breaks down this system into three subsystems (Stop watch, seven segment driver, and the seven segment displays). You have already designed the seven segment driver and display unit so you can reuse that when needed. Now let's have a closer look at the stopwatch subsystem (Figure 3). The controller unit is a simple state machine. If you consider your timer as a group of counters tied together, their counting state is either idle or counting. So in the first step you will design a counter controller.

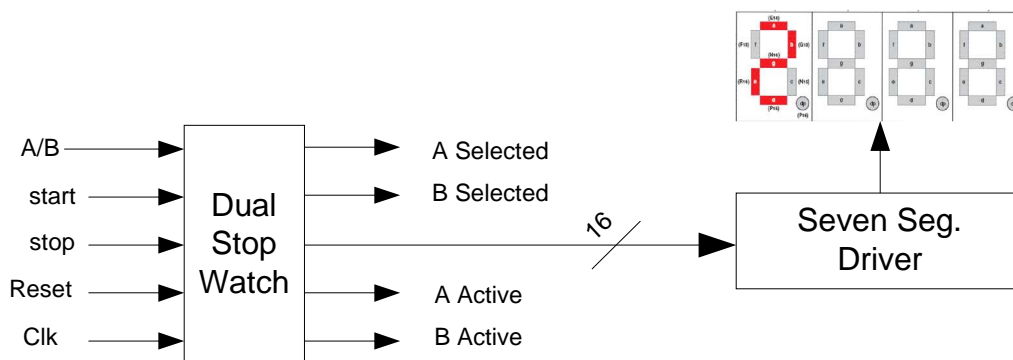


Figure 2

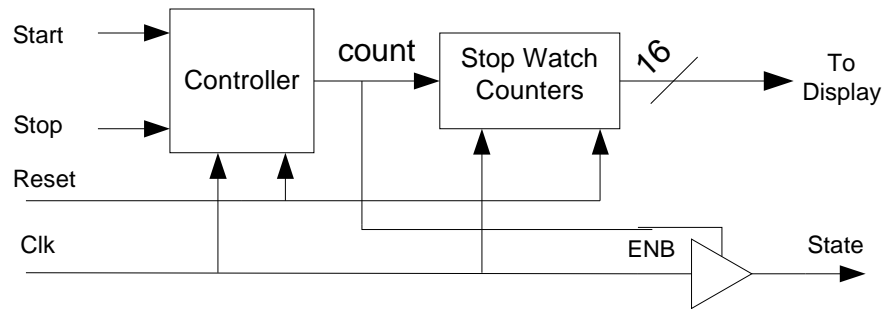


Figure 3

Stop Watch Controller

Controller is a state machine with two inputs (start, and stop) one asynchronous reset and a *clk*. The count output is one when the controller is in state counting; otherwise it is equal to zero. The *state* output is equal to *clk* if the state machine is in counting state, otherwise it is deactivated. Note that the *state* output is used to drive the LED display. We rarely pass a *clk* signal through a get. Why?

1. Draw a state transition diagram for the controller unit.
2. Determine the number of flip flops needed, and find the (minimized) next state and output logic functions.
3. Create a new schematic file in your project for your controller.
4. Enter your design into the file.
5. Check your schematic and save the file.
6. Create a test bench wave form to test the behavior of the controller
7. Demonstrate your design and test bench waveform to the TA.
8. Create a symbol for your design.

Stop Watch Counters

You can break down the stop watch counter subsystem into four individual counters as depicted in Figure 4.

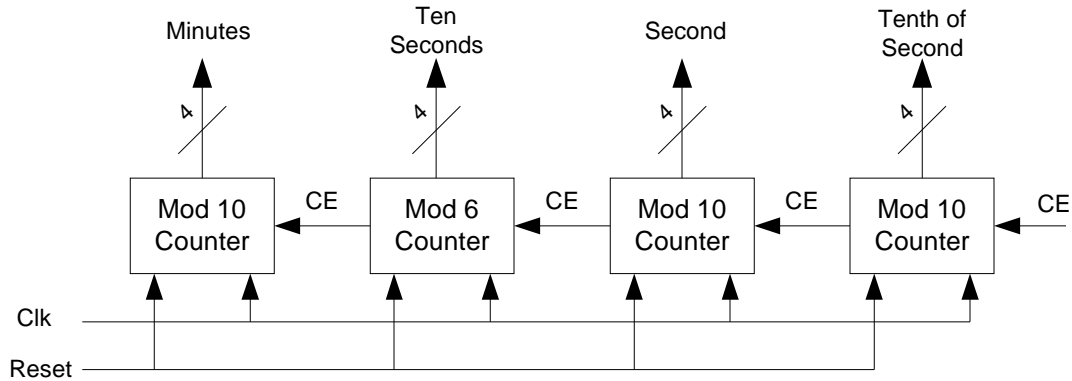


Figure 4

Here we describe the generic mod m counter input output ports (Figure 5):

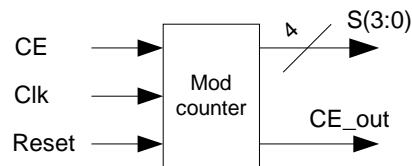


Figure 5

CE (Clock Enable): When activated the counter counts up, otherwise the counter remains in its previous state.

Reset: When activated resets the counter asynchronously.

S(3:0): The output of the counter. Note that for consistency we use four bits even for mod six counter.

CE_out: This output should be activated when the counter reaches its largest value. This can be used for cascading counters.

Mod m counter behavior: Starting from zero this counter counts up from 0 to $m-1$ consecutively and generates a pulse on **CE-Out** upon reaching $m-1$.

9. Design a mod ten counter (you'll need 4 variables). Draw a state transition diagram, find the (minimized) next state and output logic functions.
10. Create a new schematic for your mod ten counter.
11. Enter your design for mod ten counter. You can either by use individual flip flops in your design or you can use Xilinx 4-bit counters and add some combinational logics to make it a decade counter.
12. Check your schematic, and save your design.

13. Create a test bench waveform and test your mod ten counter.
14. Create a symbol for mod ten counter.
15. Create a new schematic, and design a mod six counter similar to the previous steps.
16. Save and test your mod six counter by using a test bench waveform.
17. Create a symbol for mod6 counter.
18. Create a new schematic and enter a design similar to Figure 4.
19. Check your schematic, and save the file.
20. Create a test bench waveform for testing this design.
21. Verify your design and make sure that it behaves as expected.

Putting the Stopwatch Subsystem Together

22. Create a new schematic file for a single stopwatch unit.
23. Enter your design using your newly generated symbols. (Figure 3)
24. Check your schematic and save the file.
25. Create a test bench waveform and verify the circuit with different input combinations.
26. Create a symbol for your design.

Clock

In order to have an accurate timer we need to use an accurate clock. You are going to use the *Clk_convrt* unit that you used in previous labs and use its 10 Hz output to drive the clock input of your stop watch.

27. Create a new schematic in your project similar to Figure 6. This represents a single stop watch.
28. Check your schematic file and save it.
29. Create a users constraint file for this design (Table 1) and save it.
30. Create a bit stream file and download that into the board.
31. Make sure that your system works properly.
32. Show your system to the TA.

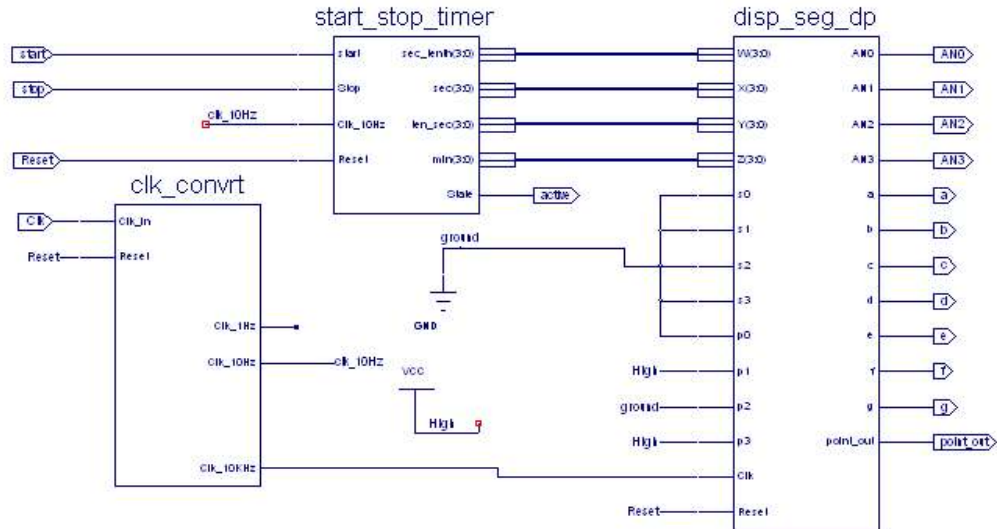


Figure 6

<pre> NET "a" LOC = "E14"; NET "b" LOC = "G13"; NET "c" LOC = "N15"; NET "d" LOC = "P15"; NET "e" LOC = "R16"; NET "f" LOC = "F13"; NET "g" LOC = "N16"; NET "Point_out" LOC = "P16"; NET "AN0" LOC = "D14"; NET "AN1" LOC = "G14"; NET "AN2" LOC = "F14"; NET "AN3" LOC = "E13"; NET "Clk" LOC = "T9"; NET "Reset" LOC = "L14"; NET "Start" LOC = "M14"; NET "Stop" LOC = "M13"; </pre>
--

Table 1

Dual Stopwatch

Now you can complete your dual stopwatch using the subsystems you have already designed. You need to use two single stopwatches, demultiplex the inputs to the system, and multiplex the outputs of your stopwatches to implement a dual stopwatch. You can use Figure 1, Figure 6, Figure 7, and Figure 8 as guidelines.

33. Create a new schematic file for your dual stopwatch.
34. Enter your design, check the schematic and save your file.
35. Create a user constraints file for your design and save it (Table 2).
36. Generate the programming file and download it into the board.
37. Make sure that the system works as described before.
38. Demonstrate your work to the TA for sign off.

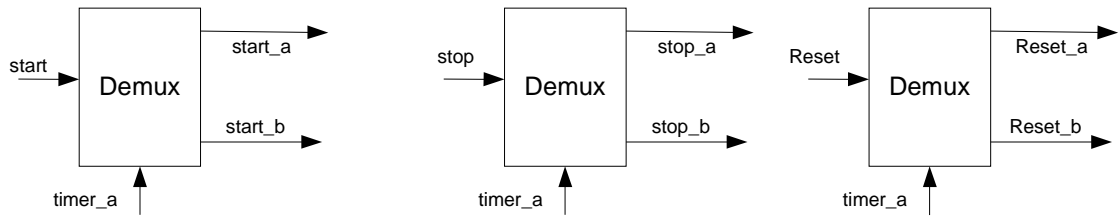


Figure 7: Demultiplexing at inputs

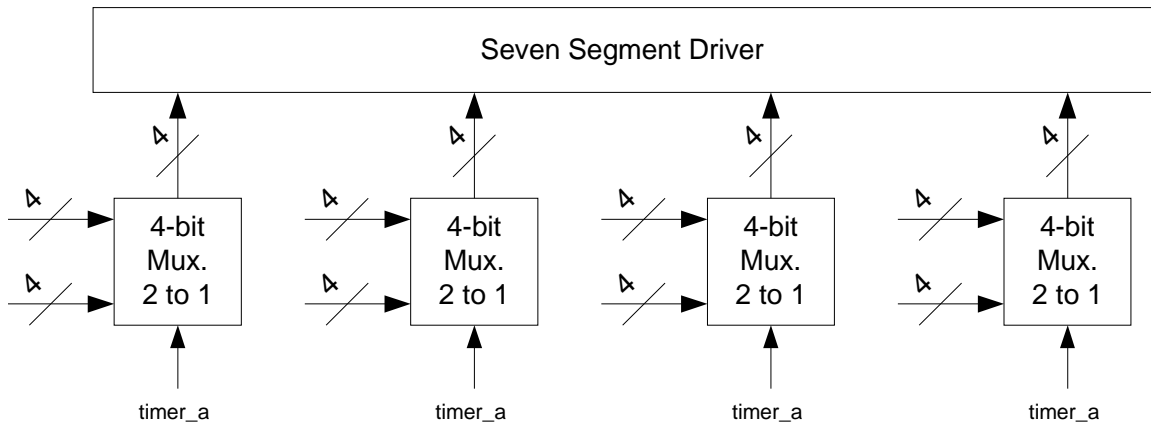


Figure 8: Multiplexing of stopwatches outputs

```
NET "a" LOC = "E14";
NET "b" LOC = "G13";
NET "c" LOC = "N15";
NET "d" LOC = "P15";
NET "e" LOC = "R16";
NET "f" LOC = "F13";
NET "g" LOC = "N16";
NET "Point_out" LOC = "P16";
NET "AN0" LOC = "D14";
NET "AN1" LOC = "G14";
NET "AN2" LOC = "F14";
NET "AN3" LOC = "E13";
NET "Clk" LOC = "T9";
NET "Reset" LOC = "L14";
NET "Start" LOC = "M14";
NET "Stop" LOC = "M13";
NET "state_a" LOC = "K12";
NET "state_b" LOC = "P14";
NET "timer_a" LOC = "K13";
NET "a_active" LOC = "P11";
```

Start/Stop Watch Timer Sign-Off Sheet

Make sure lab instructors initialize each part. Attach this page to the end of your report.

Your Name:

Lab Partner:

Date Performed:

Demonstrated that the circuit works correctly:

- Stop Watch Controller (state machine) works _____
- Single Stop Watch works: _____
- Dual Stop Watch works: _____