Lab 3

CS M152A

Aaron Cheng

Aditya Padmakumar

Introduction

The goal of this lab was to use our experience from previous labs in order to go through the complete flow of FPGA design by creating a stopwatch circuit with multiple functionalities. We implemented our stopwatch on the Nexys 3 FPGA board using its seven segment displays, switches, and push buttons. The stopwatch we designed has a basic operation mode, in which it counted minutes and seconds, from 00:00 up to 59:59, where the left two digits represent the minutes and the right two digits represent the seconds.

Other than the basic upward counting at 1 Hz, there are two input switches, SEL and ADJ, that can alter the time stored in the stopwatch. The SEL switch is used to choose between minutes and seconds in the adjusting mode.

SEL	Selected
0	Minutes
1	Seconds

The ADJ switch is used to turn the adjusting mode on and off. When the stopwatch is in adjusting mode, the selected portion of the stopwatch indicated by the SEL switch will blink and increment at a rate twice as fast as during the basic operation, allowing the user to adjust the time more quickly. The portion that is not selected will be paused and not blink.

ADJ	Mode
0	Basic Operation
1	Adjusting Mode

In addition, there are two push buttons, RESET and PAUSE, that can be used in our stopwatch. The RESET button, when pressed, will revert the counters to the initial state of 00:00. The PAUSE button, when pressed, will pause all counters, and when pressed again, will continue then.

Design Description

To begin the implementation of our stopwatch, we identified four states of operation that it would be in and what actions would cause transitions between those states. The four states that we identified were Normal, Paused, SEL Min, and SEL Sec. In the Normal state, the stopwatch increments once every second. In the Paused state, all counters are paused and the displays are frozen. In the SEL Min state, the minute digits blink and increment twice every second while the second digits are frozen. In the SEL Sec state, the second digits blink and increment twice every second while the minute digits are frozen. Figure 1 shows those states and the actions that cause transitions between the states, where posedge(p) stands for positive edge of the PAUSE button.

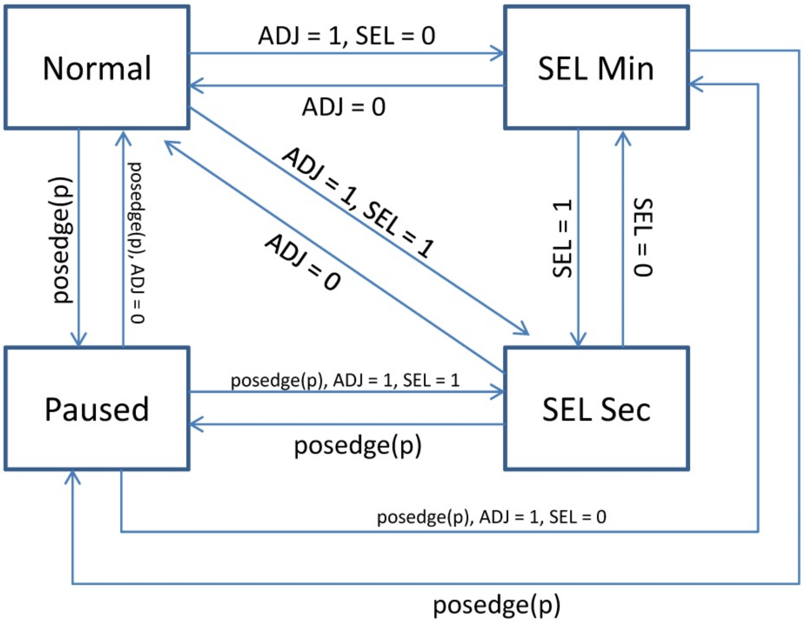

Figure 1: State Diagram and Transitions

Using this state diagram, we decided to divide our stopwatch circuit into several higher-level modules: a clock module, a state controller, a select function module, a counter controller, and a display controller. The stopwatch as a whole has four inputs, clk, reset, pause and sw[1:0], and two outputs, seg[7:0] and an. Figure 2 shows a diagram of the stopwatch's inputs, outputs, modules, and the interactions between modules.

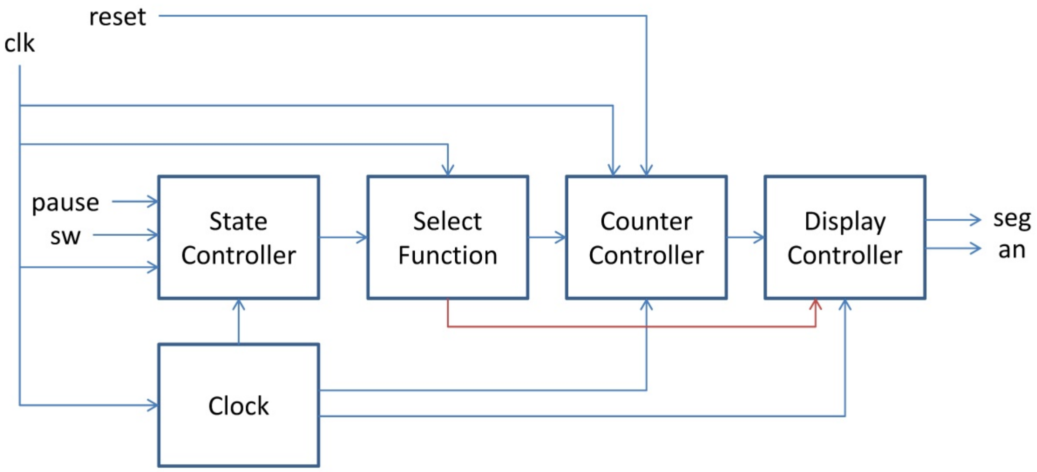

Figure 2: Inputs, Outputs, and Modules

The clock module takes as input the 100 MHz clock of the FPGA board and outputs four different modified clock signals: a 1 Hz clock to be used in the normal mode, a 2 Hz clock to be used during adjusting mode, a 4 Hz to be used for blinking, and a very fast clock (~ 714 Hz) to be used for the refreshing of the seven segment displays as well as for debouncing the push buttons.

The state controller module’s purpose is to determine the state of the stopwatch for the next clock cycle based on the current inputs and the current state. It takes as input the board’s clock, the very fast clock, and the SEL, ADJ, and PAUSE input signals. It then outputs a two-bit signal that signifies which of the four states that the stopwatch will be in during the next clock cycle.

The select function module takes as input the two-bit next state signal from the state controller as well as the board’s clock. Depending on which state the stopwatch will be in, the select function module will determine the values of various control signals that will be sent out to other modules in order to for them to perform the correct tasks during those states. The select function module’s outputs are as follows: min_count and sec_count to signify whether the minute and second digits should increment, min_clock_rate and sec_clock_rate to signify the rate at which the minute and second digits should increment, and min_blink and sec_blink to signify whether the minute and second digits should blink.

The counter controller’s function is to store and control the values of the stopwatch’s digits. It takes as input the board’s clock, the RESET input signal, the min_count, sec_count, min_clock_rate, and sec_clock_rate signals from the select function module, and the 1 Hz and 2 Hz clock signals from the clock module. Within the counter controller are actually four different counters, two mod-6 counters for the high values of the minute and second, and two mod-10 counters for the low values of the minute and second. The counter controller’s four outputs are min1, min0, sec1, and sec0, which signify the values that the seven segment displays should show.

Finally, the display controller’s function is to show the correct representation of the counters’ values on the seven segment displays. It takes as input the very fast clock and 4 Hz clock signals from the clock module, the min_blink and sec_blink signals from the select function module, and the min1, min0, sec1, and sec0 signals from the counter controller. The display controller also actually has four copies of a seven-segment-display module, each corresponding to the four values of the minute or second digits given by the counter controller. During each cycle of the very fast clock, the display controller outputs seg and an, together which determine which of the four seven segment displays and which of its segments to the light up.

Simulation Documentation

We conducted basic tests on each module mentioned in the Design Descriptions to ensure that the module was functioning as required.

Clock Module

Using the simulation waveform provided by ISIM, we observed the waveform produced by our 1Hz, 2Hz, 4Hz and fast clocks simultaneously.
Expectation: A direct relationship between the 1Hz, 2Hz and 4Hz waveforms.
Result: As expected. **Module Successful**.
Note: While testing our implementation of the clock module, we used higher frequency clocks, with the same ratios, to reduce the time needed for the waveform to form.

For the rest of the modules, we conducted tests in an incremental fashion. Starting from the Display Controller module, we went backwards and added on each module. This way, we were able to test our modules directly by observing the output on the seven segment display present on the FPGA board.

Display Controller Module

All inputs other than the clocks were hard coded to test this module alone. Setting these hard coded values allowed us to control the functions of this module.

Test Cases:

Test	Expected	Result (Success?)

Values for min1, min0, sec1, and sec0 set to output 0000.
Min_blink =1.
Sec_blink=0.
	Only the two digits representing the minutes will blink with value 00, while the other digits do not blink.	Y

Values for min1, min0, sec1, and sec0 set to output 0000.
Min_blink =0.
Sec_blink=1.
	Only the two digits representing the seconds will blink with value 00, while the other digits do not blink.	Y

Values for min1, min0, sec1, and sec0 set to output 0000.
Min_blink =0.
Sec_blink=0.
	All digits do not blink and display 0000.	Y

Counter Controller Module

All inputs other than the clocks were hard coded to test this module. All outputs were sent to the Display Controller module to test the overall functionality between these modules.

Test Cases:

Test	**Expected**	**Result (Success?)**
reset is 0. min_count, sec_count is 1. min_clock_rate set to use 1Hz. sec_clock_rate set to use 1Hz.	The 7 segment display will count from 00:00 to 59:59 and then restart.	Y
reset is 0. min_count is 0. sec_count is 1. min_clock_rate set to use 1Hz. sec_clock_rate set to use 2Hz.	The second digits now count at a faster rate than the 1st test case. The minute digits stay constant.	Y
reset is 0. min_count is 1. sec_count is 0. min_clock_rate set to use 2Hz. sec_clock_rate set to use 1Hz.	The minute digits now count at a faster rate than the 1st test case. The second digits stay constant.	Y

Select Function Module

The next state signal was hard coded to test these modules. All the modules after this module were also connected to test the overall functionality between these modules thus far.

Test Cases:

Test	**Expected**	**Result (Success?)**
PAUSE	The display will stay constant.	Y
NORMAL	The 7 segment display will count from 00:00 to 59:59 and then restart.	Y
SEL_Min	The minute digits count at a faster rate and blink, while the second digits stay constant.	Y
SEL_Sec	The second digits count at a faster rate and blink, while the minute digits stay constant.	Y

State Controller Module

This module was tested by connecting all the other modules together and implementing our design as a complete project. All input values were taken from the FGPA board's buttons and switches.

Test Cases:

Test	Expected	Result (Success?)
(no action)	The stopwatch counts from 00:00 to 59:59 and then resets automatically.	Y
Press PAUSE button	The stopwatch value will stay constant at the value it was currently at.	Y
Press PAUSE button, and again after 5 seconds.	Stopwatch will stop, and then continue to count again from the paused value.	Y
SEL switch turned ON and ADJ switch turned ON.	The second digits count up at a faster rate while blinking. The minute digits are disabled.	Y
SEL switch turned ON and ADJ switch turned ON, then turn OFF ADJ after 5 seconds.	The second digits count up at a faster rate while blinking. After 5 seconds, the second digits stop blinking and counts at the usual rate. The minute digits are also enabled now.	Y
SEL switch turned OFF and ADJ switch turned ON.	The minute digits count up at a faster rate while blinking. The second digits are disabled.	Y
SEL switch turned ON and ADJ switch turned ON, then turn OFF ADJ after 5 seconds.	The minute digits count up at a faster rate while blinking. After 5 seconds, the minute digits stop blinking and counts at the usual rate. The second digits are also enabled now.	Y
Press RESET button	The display turns to 00:00 and starts counting.	Y

Conclusion

Based on the tests we designed for this project and the demo we passed, we can confidently say that our design successfully implements the functionality of a stop watch on a FPGA board.

During the lab we dealt with the following problems:

- Debouncing the PAUSE button signal was challenging as we found that the button only worked with 75% efficiency. After consulting our TA about the concept of debouncing and understanding how it was implemented in Lab 1, we were able to solve this problem completely.