COL215P - Hardware Assignment 1

Stopwatch

Submission Deadline: 4 September Aug 2022

1 Introduction

In this assignment, you will construct a stopwatch using the LED displays and push button on the Basys 3 board.

2 Problem Description

Display format of the clock will be (M:SS:T): minutes (M) on one LED display, seconds (SS) on two LED displayS and tenth of second (T) on one LED display. Refer to the format above. The task is to design and implement the following modules:

1. Seven segment decoder for single LED display
2. Timing circuit to drive all LED displays
3. Modulo N counter
4. Debouncing module for push button

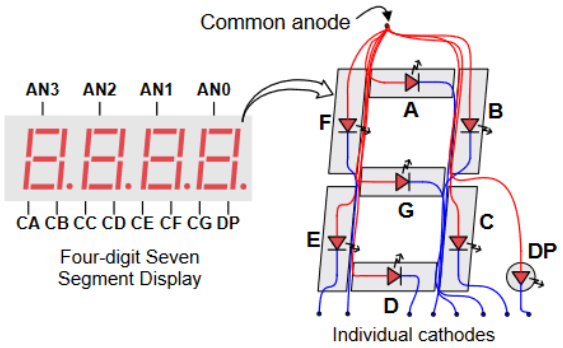

Figure 1: Pin details for 7 seven display on Basys 3 board

2.1 Seven segment decoder

The first module to be implemented is for displaying a 4-bit number on a one 7-segment display. Figure 1 shows the pin-out details for the one display. It has 7 cathode pins, 1 anode pin (connected to all 7 LEDs) and 1 pin for decimal point. To switch an LED on, the anode should be driven HIGH and cathode LOW. In general, LED display is activated via driving anode HIGH and the corresponding digit via varying cathode signals. Now, if the input number is "0000", then all LEDs except G should be switched ON. Similarly, for "0011" LEDs F and E will be OFF, and rest wil be ON.

Note that on basys 3 board anode/cathode are ACTIVE LOW pins (i.e., 0 = ACTIVE, 1 = INACTIVE). Details, refer basys 3 reference manual.

Design a combinational circuit for which the inputs are the four bits of the given decimal number and outputs correspond to the 7 cathode pins. The following need to be completed:

- Create a Truth Table with 4 inputs and 7 outputs. Using this, find out the minimized combinational logic to drive the 7 output signals.
- Implement a decoder module in VHDL, with input as a 4-bit number and output as 7-bit cathode signal, and 4-bit Anode signal. To test your implementation on the board, you can use the slider switch for entering the number.

NOTE: use of VHDL case/if-else for this module is not allowed. Use elementary operations (AND, OR, etc.)

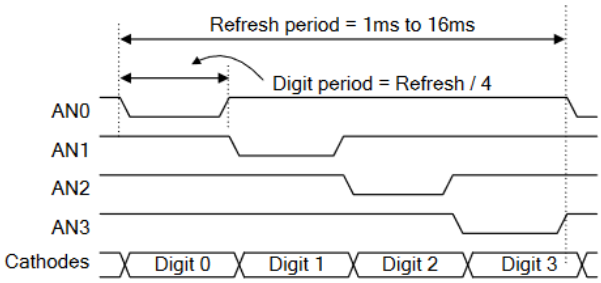

Figure 2: Timing details for 7 seven segment displays

2.2 Driving all four LED displays

As shown in Figure 1, 7 cathode pins are common for each LED display. This means only one display can be activated at a given time. To display a separate digit on each LED display, the corresponding anode signal needs to be activated in cyclic manner. To avoid flickering, refresh rate should vary between 1kHz - 60Hz (1-16 ms). Figure 2 shows the timing details and the signal waveforms. For more details, refer to the Basys 3 reference manual.

Using the module designed in section 2.1, create a 4:1 multiplexer module and timing circuit to drive the LED displays. The following need to be completed:

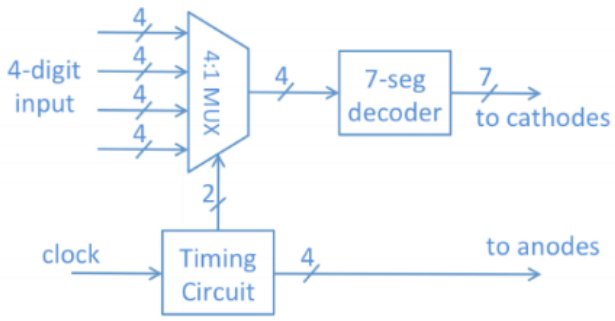

Figure 3: Block diagram for driving four 7 seven segment displays

- Multiplexer module with four 4-bit inputs from slider switches and output going to 7-segment module.
- Timing circuit to drive Anode signals and select signal of multiplexer.

Refer to Figure 3. You need to divide the onboard 100 MHz clock to the suitable display frequency. Note that a modulo N counter divides the frequency by N. It is not necessary to use a modulo N counter in this part.

2.3 stopwatch

Based on the display format for the stopwatch, following modules will be needed:

- modulo 10 counter for minutes
- modulo 6 counter for tens of seconds
- modulo 10 counter for units of seconds
- modulo 10 counter for tenths of seconds

The output from these counters will drive the 7-segment display module designed in Section 2.2. Inputs will be *enable_watch* (for counting the time) and *reset_watch* (reset watch to 0:00.0). You need to design a circuit to divide clock 100 Mhz to 10 Hz, this can be done using modulo N counter where N = 10^7. In addition to this, the watch will be controlled via three push buttons:

- start/continue
- pause
- reset

When start button is pressed then *enable_watch* should be set to HIGH and when pause is pressed then *enable_watch* set to LOW. On pressing reset, *reset_watch* should be set to HIGH. **For using push button on Basys 3 board, you need to take care of debouncing of the button.** When a button is pressed, ideally output signal changes from LOW to HIGH state but due to metallic contacts output signal transition contains glitches as shown in Figure 4.

Your task is to design a module to remove the glitches and output only the final stable state. There are various ways to design a debouncing module, one way is to remove the glitches using a counter (small delay), once the counter overflows then assign the input signal to the output port.

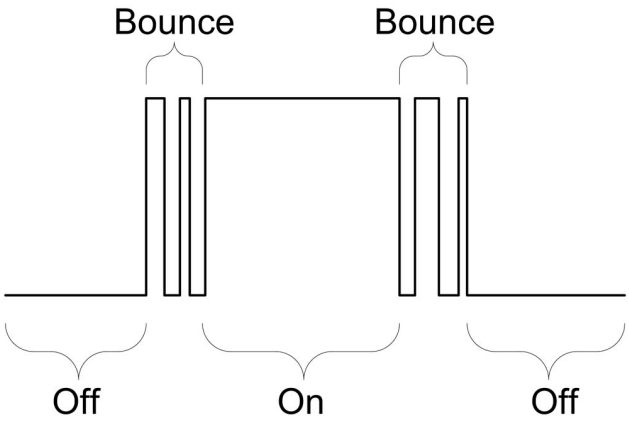

Figure 4: Glitched when operating push button

3 Submission and Demo Instructions

1. Demo should be given in the assigned lab slot.
2. You are required to submit zip file containing the following on Gradescope:
   - VHDL files for all the designed modules.
   - A short report (1-2 pages) explaining your approach. Include block diagram depicting the modules in the stopwatch.
3. Form groups of 2 students for the assignment.
   - Groups will remain the same for subsequent assignments; no changes allowed.
   - Only one submission is needed per group via Gradescope. Gradescope help link: `https://help.gradescope.com/article/m5qz2xsnjy-student-add-group-members`
4. Post your doubts in Assignment 1 thread on Piazza.
5. It is advisable that you should be ready with your design before the lab session, and during the session, perform validation of the design by downloading it into the FPGA board

4 Resources references

- IEEE document: `https://ieeexplore.ieee.org/document/8938196`
- Basys 3 board reference manual: `https://digilent.com/reference/_media/basys3:basys3_rm.pdf`
- Online VHDL simulator: `https://www.edaplayground.com/x/A4`