Multiplexed Display

Introduction

In many applications we can reduce the number of I/O pins required by time-multiplexing them. Typical examples include displays and keyboards. In this lab you will display the last four digits of your BCIT ID on the 4-digit 7-segment LED display.

A multiplexed display cycles through the digits, turning on the appropriate segments for each digit. If this is done fast enough then all four digits will appear to be on continuously.

The frequency at which you cycle through the digits should be high enough that you cannot see the LEDs turn on and off. A rate of 100 Hz is above most people's flicker fusion threshold. However, the period should be long enough that the waveforms are not affected by the rise and fall times of the circuit.

Components

You will need the same components, wired up in the same way, as in previous labs that used the 4-digit 7-segment LED display. The switch inputs will not be needed for this lab.

Requirements

Your circuit should show the last four digits of your BCIT ID on the display.

Each digit should be displayed at a frequency of $100 + n$ Hz where n is the last two digits of your BCIT ID number. Since there are four digits, your circuit must divide the 50 MHz clock by $^{50\times10^6}/_{(4\cdot(100+n))}$.

The following table shows how the last digit of your BCIT ID determines the order the display digits are enabled:

last digit of your ID	digit sequence
0	3, 0, 1, 2
1	1, 2, 0, 3
2	0, 1, 2, 3
3	0, 1, 3, 2
4	2, 1, 3, 0
5	2, 1, 3, 0
6	1, 2, 0, 3
7	0, 2, 1, 3
8	0, 3, 2, 1
9	2, 0, 3, 1

For example, if your ID was A12345678 then you would assert **en0** (rightmost digit), then **en3** (leftmost), then **en2** and finally **en1**. This sequence would repeat at 178 Hz and each digit would be turned on for $\approx$ 1.4 ms at a time ($^1/_{(4\cdot178)}$).

CPLD I/O

The connections to the LED display are as shown in previous labs.

Design Ideas

Your circuit will need a clock divider that controls a digit counter. The digit counter will change once per clock period (as given above). For the example period given above the divider would count down from $^{50\times10^6}/_{4\cdot178} - 1 \approx 70\,224$ to zero.

Your digit counter can sequence over the enable values (e.g. 4'b0001, 4'b1000, 4'b0100, 4'b0010) or over a value that represents the digit of your ID (e.g. 3, 2, 1, 0).

The digit counter value is then used to enable the correct digit enables (unless you used the enable value as the counter values) and the correct segment enables. This can be done with multiplexers or lookup tables.

Procedure

Draw a block diagram for a design that will meet the requirements above. Write a Verilog module that corresponds to your block diagram. Create a project, compile it, and configure your CPLD. Assign pin 12 to the 50 MHz oscillator input. Test your design and make the measurements described below for inclusion in your report.

Submission

To get credit for completing this lab, submit to the appropriate assignment folder a PDF document containing:

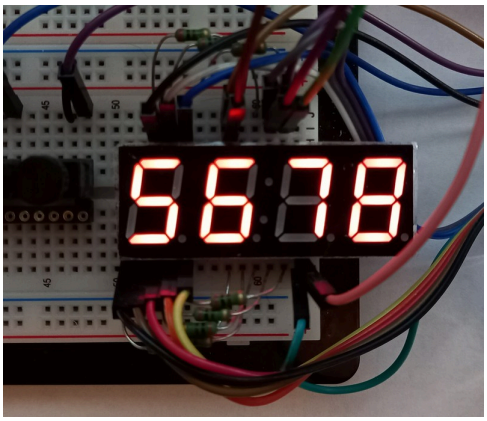
- The calculation of the clock divider value and the enable sequence. Note that these will be different for each student.
- A block diagram corresponding to your design (drawn by you, not generated by Quartus).
- A listing of your Verilog code.
- A screen capture of your compilation report (**Window** > **Compilation Report**) similar to this:

Flow Summary

<<Filter>>

Flow Status	Successful - Mon Feb 01 21:03:29 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	lab4
Top-level Entity Name	lab4
Family	MAX II
Device	EPM240T100C5
Timing Models	Final
Total logic elements	42 / 240 (18 %)
Total pins	13 / 80 (16 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

- A photo of your breadboard distinctly showing the last four digits of your BCIT ID on the four LEDs. For example:



- A screen capture of the 'scope measurement of the **en0** signal including a frequency measurement. The last two digits of the frequency should match the last two digits of your ID. This might look like Figure 1.
- A screen capture of the logic analyzer capture of the **en3** through **en0** signals showing the required digit enable order. This might look like Figure 2.

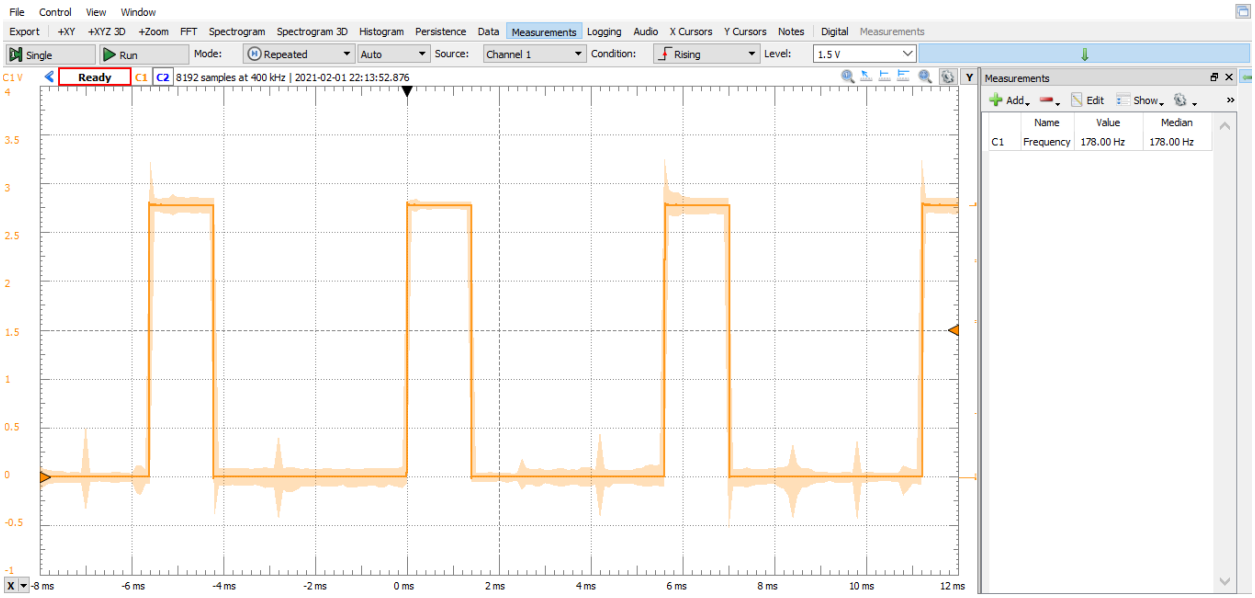


Figure 1: 'Scope Capture.

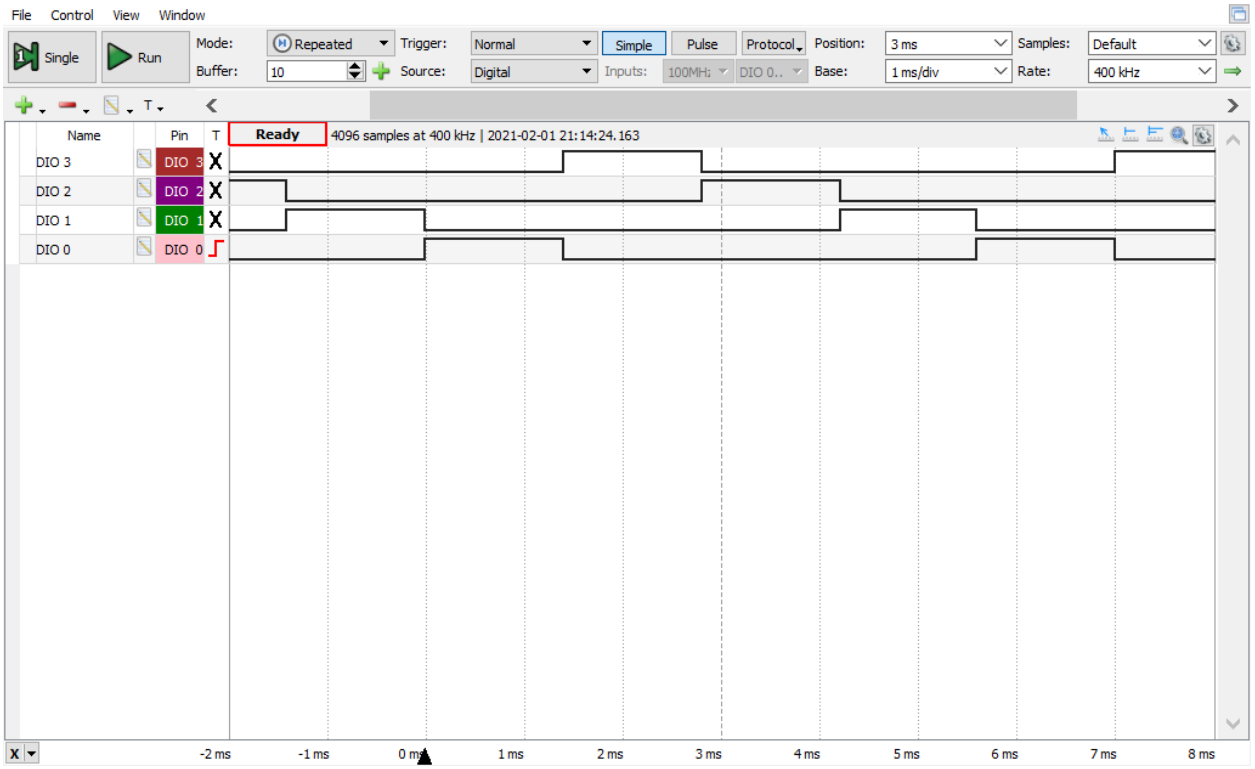


Figure 2: Logic Analyzer Capture.