Timers and Frequency Dividers

Revision 2: added more details and state transition diagrams to Specifications section.

Introduction

Timer circuits measure time by counting clock cycles. Clock dividers, also called frequency dividers, are timer circuits that generate periodic outputs.

In this lab you will use a frequency divider to generate a tone by switching the voltage applied to a speaker at an audio frequency. The duration of the tone will be determined by a timer.

Your design will set a counter with a value $N-1$ when the keypad button [1] is pushed. This counter will then count down by 1 on each clock until it reaches 0 and then stop.

The frequency divider uses a counter that continuously counts down from $M-1$ to 0. When the counter reaches zero the output is toggled (changes value) if the timer counter is still counting down (i.e. it is non-zero). If the timer counter is zero, then the output is set low.

The tone duration is N times the clock period. The tone period is M times the clock period.

In this lab you will use the 50 MHz clock on the FPGA board. The period of this clock is $1/50\times10^6 = 20$ ns.

Components

You will need:

- your FPGA board, Byte Blaster JTAG interface and mini-USB power connector,
- the matrix keypad
- the speaker from your ELEX 2117 parts kit
- two cables with alligator clips on both ends from your ELEX 1117 parts kit[1]

[1] You can put alligator clips over the banana plugs.

Specifications

Pushing keypad button [1] should result in a tone that lasts for one second. The required tone frequency, f, is determined by your BCIT ID:

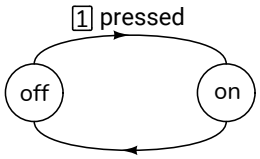
$$f = 500 + 100d_0 \text{ Hz}$$

where d_0 is the rightmost digit of your BCIT ID. For example, if your BCIT ID was A01234567 the tone frequency would be 1200 Hz.

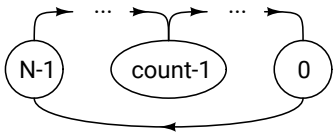
The tone should last for one second regardless of how long [1] is pressed. To do this you'll need a state machine that detects the falling edge of the `col[3]` signal and a timer with a duration of 1 second.

To generate the tone you'll need a frequency divider to divide the 50 MHz clock frequency down to a frequency $f = \frac{1}{T}$ by counting enough 20 ns clock periods to create a delay of $\frac{1}{2} \cdot \frac{1}{T}$ between toggling of the `spkr` output.

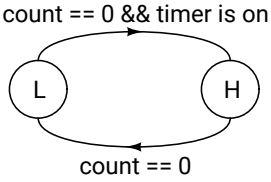
A state transition diagram for the timer state machine could be drawn as:



A state transition diagram for the frequency divider could be:



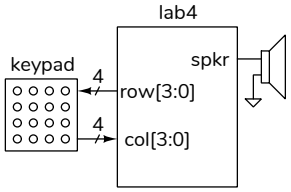
where the state is the value of a counter. The state transition diagram for togging the speaker output might be:



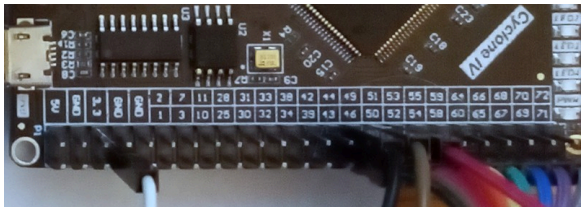
where the state is the speaker output level.

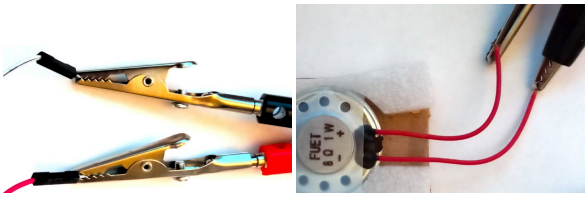
FPGA I/O

The following diagram shows the connections to the FPGA:



Connect the matrix keypad to the FPGA as in previous labs. Connect an FPGA I/O pin to the speaker pin and a ground pin with alligator clip cables (pin 53 was used as the **`spkr`** pin here):





Procedure

Draw a block diagram that implements the circuit described above. Use multiplexers, registers and subtraction blocks.

Create a project, compile it, and configure the FPGA.

If you use the same keypad pins as in the previous lab and Pin 53 for the speaker output, you should end up with the following pin assignments:

To	Assignment Name	Value
clk50	Location	PIN_23
spkr	Location	PIN_53
row[3]	Location	PIN_103
row[2]	Location	PIN_100
row[1]	Location	PIN_98
row[0]	Location	PIN_86
col[3]	Location	PIN_84
col[2]	Location	PIN_80
col[1]	Location	PIN_76
col[0]	Location	PIN_74
col*	Weak Pull-Up Resistor	On

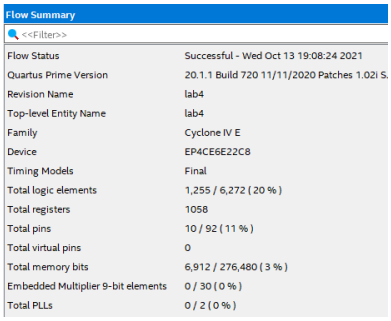
You can download the **`lab4.sdc`** file from the course website and add it to your project (**Project** > **Add/Remove Files in Project**... > **....**) to allow the timing analysis to run without warnings.

Test your design.

Submission

To get credit for completing this lab, submit the following to the appropriate Assignment folder on the course website:

1. A PDF document containing:
   (a) A block diagram of your design (not the one generated by Quartus).
   (b) A listing of your Verilog code.
   (c) A screen capture of your compilation report (**Window** > **Compilation Report**) similar to this:



Flow Summary	
Flow Status	Successful - Wed Oct 13 19:08:24 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 Patches 1.02l S.
Revision Name	lab4
Top-level Entity Name	lab4
Family	Cyclone IV E
Device	EP4CE6E22C8
Timing Models	Final
Total logic elements	1,255 / 6,272 (20 %)
Total registers	1058
Total pins	10 / 92 (11 %)
Total virtual pins	0
Total memory bits	6,912 / 276,480 (3 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	0 / 2 (0 %)

   (d) The schematic created by **Tools > Netlist Viewers > RTL Viewer** and then **File > Export**.... The schematic might look like Figure 1.

2. If you do not demonstrate your completed lab in person, submit a short video, including audio, showing a brief button press and the tone

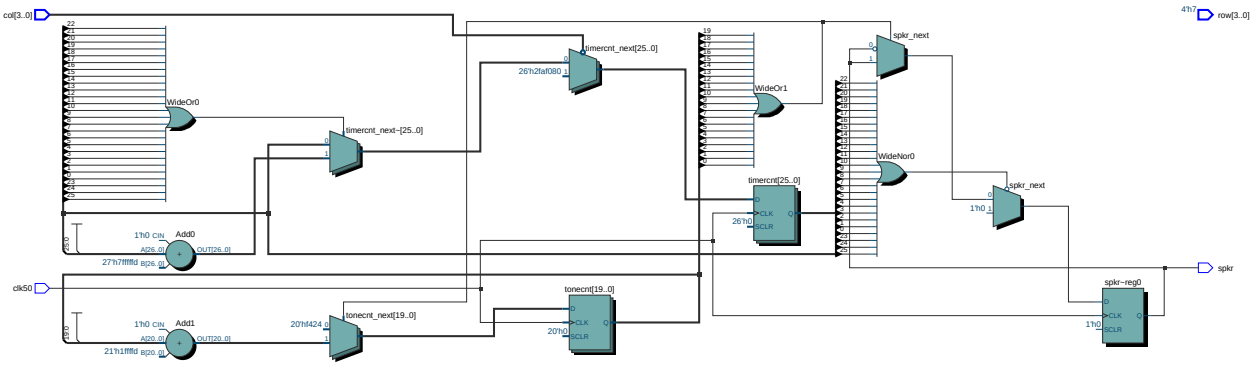


Figure 1: Example RTL Schematic for Lab 4.

generated. An example video is available on the course website.