Timers and Clock Dividers

Introduction

Timers use counters to create delays. Clock dividers[1] use counters to generate periodic outputs.

In this lab you will use a clock divider to generate an audible tone by switching the voltage applied to a speaker. The frequency of the tone and the duration of the tone will be determined by timers.

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

Components

You will need:

- your CPLD board, Byte Blaster JTAG interface and mini-USB power connector,
- the matrix keypad
- the speaker from your ELEX 2117 parts kit
- jumpers (or cables with alligator clips on both ends from your ELEX 1117 parts kit[2])

Requirements

You must customize your design using the last three digits of your BCIT ID: n_1, n_2, and n_3. For example, if your BCIT ID is A00123456 then $n_1 = 4$, $n_2 = 5$ and $n_3 = 6$.

Pushing the keypad key $\boxed{n_1}$ should result in a tone at a frequency of $f = 500 + 100 \times n_2$ Hz that lasts for $1 + n_3/3$ seconds.

For the ID given above, pressing $\boxed{4}$ should generate a $500 + 100 \times 5 = 1000$ Hz tone for $1 + 6/3 = 3$ seconds.

The tone should last for the required duration regardless of how long the key is pressed. The tone should be a square wave. The speaker output should be set low when no tone is being generated.

[1] Also called frequency dividers.

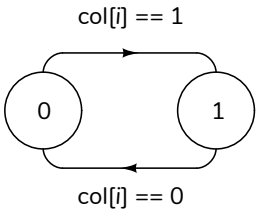
[2] You can put alligator clips over the banana plugs.

Hints

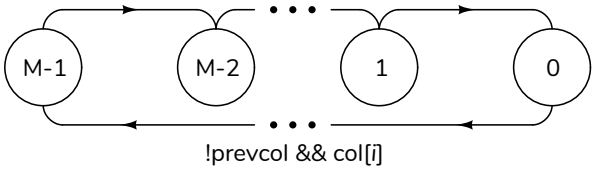
The description hints that three state machines are required: (1) an edge detector to detect the (active-low) button push, (2) a timer to create a tone of the appropriate duration, and (3) a timer to create a tone period equal to $1/f$ where f is the tone's frequency.

To help you get started, below are some state transition diagrams you could use.

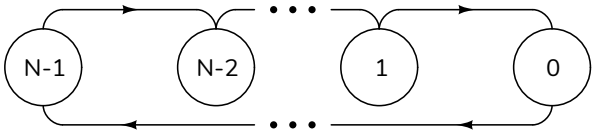
An edge detector can be implemented using a flip-flop (labelled **`prevcol`** here) bit to store the value of the appropriate **`col`** input (labelled **`col[`**i**`]`** here):



The tone duration counter is set to $M - 1$ when a keypress is detected (corresponding to a falling edge on **`col[`**i**`]`**) and counts down to zero:



The tone period counter continuously counts down from $N - 1$ to zero:

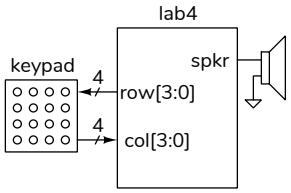


The output is set high[3] while the duration timer is non-zero and tone period counter is less than $\frac{M-1}{2}$.

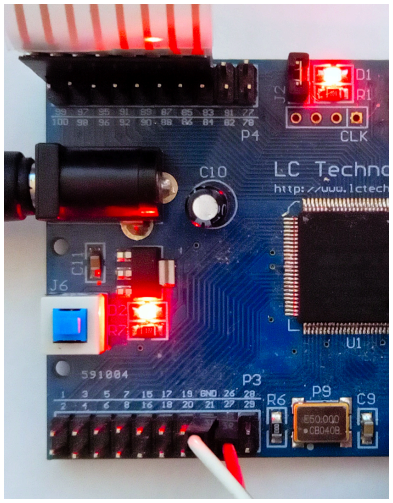
[3] Since the timer runs continuously, the first and last periods of the tone waveform could have durations less than a period. You can fix this if you want.

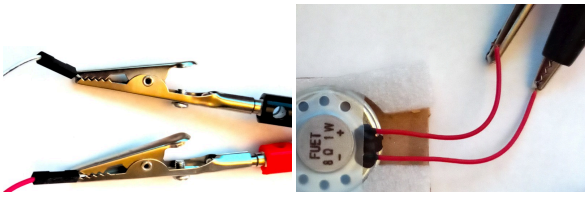
CPLD I/O

The following diagram shows the connections to the CPLD:



Connect the matrix keypad to the CPLD as in previous labs. Connect a CPLD I/O pin to the speaker pin and a ground pin with alligator clip cables (pin 26 was used as the **`spkr`** pin here, a ground pin is next to it):





Procedure

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

If you use the same keypad pins as in the previous lab and Pin 26 for the speaker output, you should end up with the following pin assignments:

From	To	Assignment Name	Value	Enal
	row[3]	Location	PIN_99	Yes
	row[2]	Location	PIN_97	Yes
	row[1]	Location	PIN_95	Yes
	row[0]	Location	PIN_91	Yes
	col[3]	Location	PIN_89	Yes
	col[2]	Location	PIN_87	Yes
	col[1]	Location	PIN_85	Yes
	col[0]	Location	PIN_83	Yes
	clk50	Location	PIN_12	Yes
	spkr	Location	PIN_26	Yes
	led	Location	PIN_77	Yes
	col	Weak Pull-Up Resistor	On	Yes
<...>	<<new>>	<<new>>		

You can import these pin assignments above from the **`lab4.qsf`** file on the course website.

For troubleshooting you can assign internal signals to the **`led`** output on pin 77 . A high level turns on this on-board LED. You can also view this signal with an oscilloscope.

Test your design.

Troubleshooting

Troubleshoot any electronic device in the following order:

- check power and ground voltages (here: check the power switch and LED)
- check for a clock (here: assume OK)
- check the inputs (here: assign to on-board LED)
- check the outputs (here: assign to pin 77 and use a 'scope)

If you use the same pins as above you can program the **`.pof`** file from the course website to check your hardware.

Submission

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

- A PDF document containing:
  1. (a) The values of n_1, n_2, and n_3 corresponding to your BCIT ID.
     (b) The corresponding button to be pushed, tone frequency and duration.

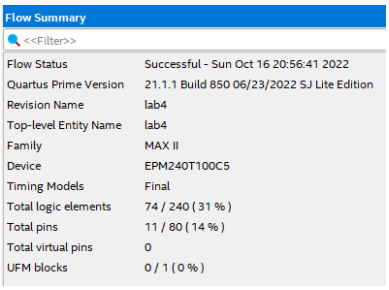
(c) The value of `row` for your value of n_1.

(d) The values of N and M for your values of n_2, and n_3

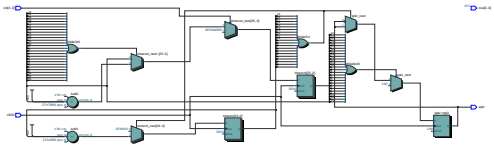
(e) A block diagram of your solution. Follow the instructions in the report and video guidelines document.

2. A listing of your Verilog code.

3. A screen capture of your compilation report (**Window > Compilation Report**) similar to:

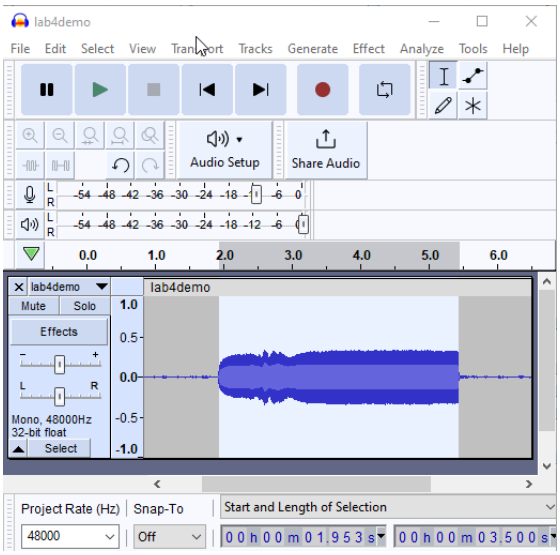


4. The schematic created by **Tools > Netlist Viewers > RTL Viewer** and then **File > Export...**. For example:



- If you do not demonstrate your completed lab in person, submit a short video, including audio, showing: (1) the tone generated by a button press shorter than the tone duration, (2) a button press longer than the tone duration, and (3) that no tone is generated when pressing a button on a different row and a different column. The audio should be clearly audible so that the instructor can check the duration and frequency of your tone. A sample video is available on the course website.

  Optionally, you can check the tone duration and frequency using an audio editor such as Audacity. For example, here is a display of a recorded audio waveform showing that the selected portion has a duration of 3.5 seconds:



and a plot of the spectrum (Analyze/Plot Spectrum...) showing the fundamental frequency at 1000 Hz and the odd harmonics (at 3, 5, ... kHz):