Assignment 2

Digital Clock with PyRTL

In this assignment we will create a Digital Clock using PyRTL. First let’s start with visualizing what we are going to create. A digital clock usually has:

2 digits to count from 01 to 12 in case of 12 hour clocks and from 00 to 23 in case of 24 hour clocks.

2 digits for minutes which count the minutes from 00 to 59.

2 digits for seconds which count the seconds from 00 to 59.

How will our clock show the time

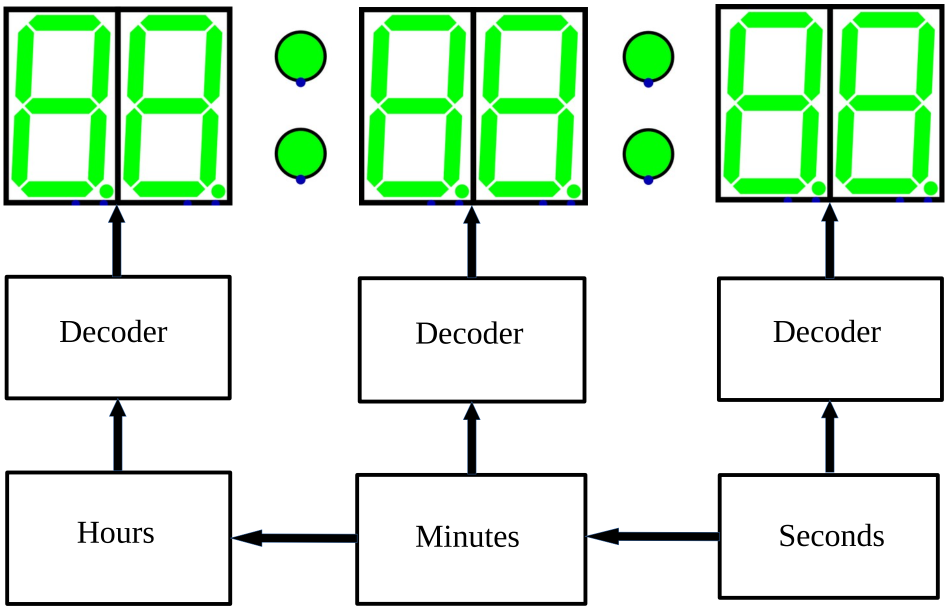
We will develop the 4 digit clock which counts the hours/minutes in 12 hour standard, meaning that the clock needs 2 digits for hours counting from 01 to 12 and 2 digits counting from 00 to 59. This clock counts from 01:00 to 12:59 and back to 01:00. It can also have a blinking separator (:).

01:00
→
12:59
→
01:00
→
...

As an extra credit you can add seconds to the clock counting from 00 to 59, that way as soon as the clock starts, counting seconds is visible with out wait one whole minute for 00 to changes 01 in the minutes part. Also you can add adjust circuitry to adjust the clock to the desired hour(s) and minute(s) and then let the clock go from there. Also adding AM/PM would be another improvement to this project.

Assignment 2

Below is a diagram of our digital clock.



Use the skeleton file (digital_clock.py) and build up your project on that.

Think about a module which counts numbers in stages and each stage updates the next stage (digits to its left). Don’t need the reset. Your circuit will start from a stable state. (PyRTL takes care of that)

Once the code is completed

If you add your own Binary to 7-segment decoder you can map the pins to 7-segment PMOD on IceBreaker board to have a working digital clock. :) Synthesize it using Yosys for iCE40HX-1k and upload the bit stream to the IceBreaker board.

Your Verilog file should generate the .vcd file for waveform viewer (GTKWave). Please insert a screenshot of your wave form in your lab report cropped to show the minute change in the wave form.

Assignment 2

Your report should include

For Part 1:

The completed skeleton file (digital_clock.py).

Explain how you implemented your circuit and how would you tell if your circuit would work as a digital clock would without a need for a test bench.

For Part 2:

Perform an analysis on your circuit using PyRTL tools for:

Pre- and Post-optimization report on:

- Estimated Area
- Maximum Frequency[1]

Generate a graph representing your circuit using PyRTL's built-in tools

Generate the Verilog output file and its test bench, then use iverilog to generate the .vcd file and include the .vcd file with the rest of the assignment files.

A screenshot of your waveform in GTKWave. Please only display the critical parts of the part of waveform.

Extra Credit:

Add seconds counter to the clock counting from 00 to 59, that way as soon as the clock starts, counting the seconds is visible through change of digits without wait one whole minute for 00 to changes 01 in the minutes part.

Add adjust circuitry to adjust the clock to the desired hour(s) and minute(s) and then let the clock go from there.

Adding AM/PM would be another improvement to this project.

Your bit stream generated by Yosys to be uploaded to the IceBreaker board with 7-segment PMODs.

[1] The clock frequency will be 1 second but for practice purposes we want to see how fast this circuit can count numbers.