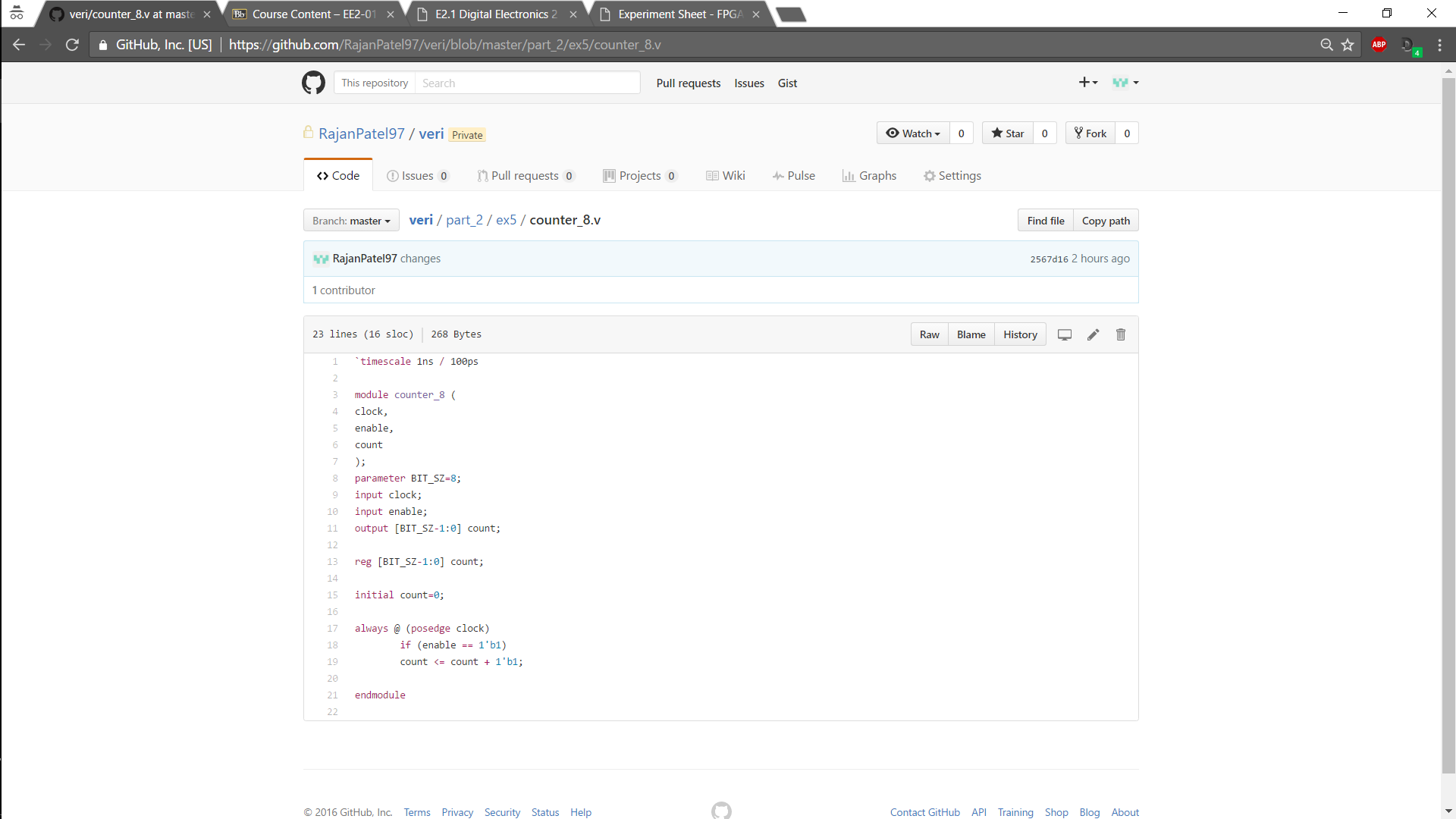
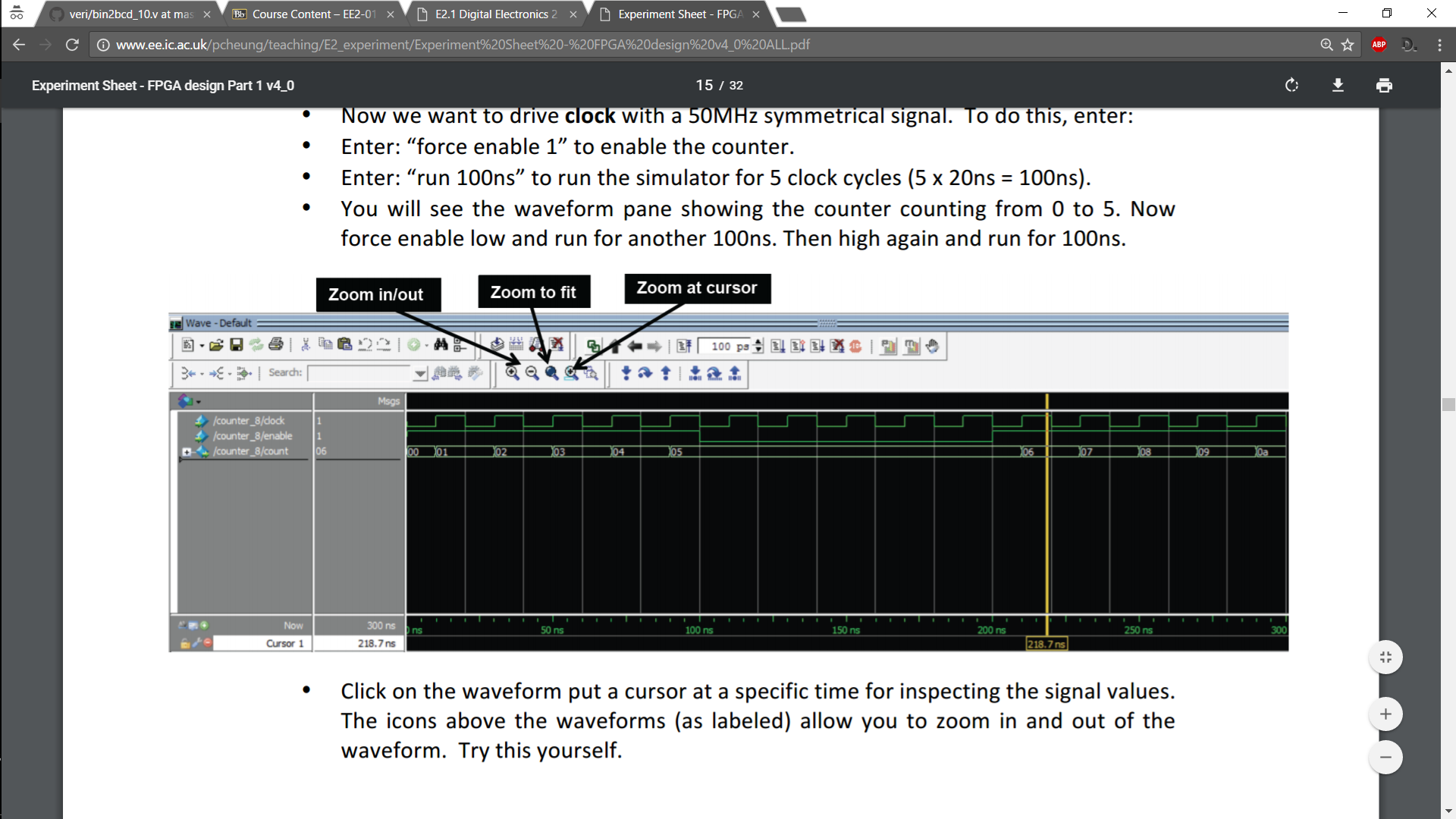
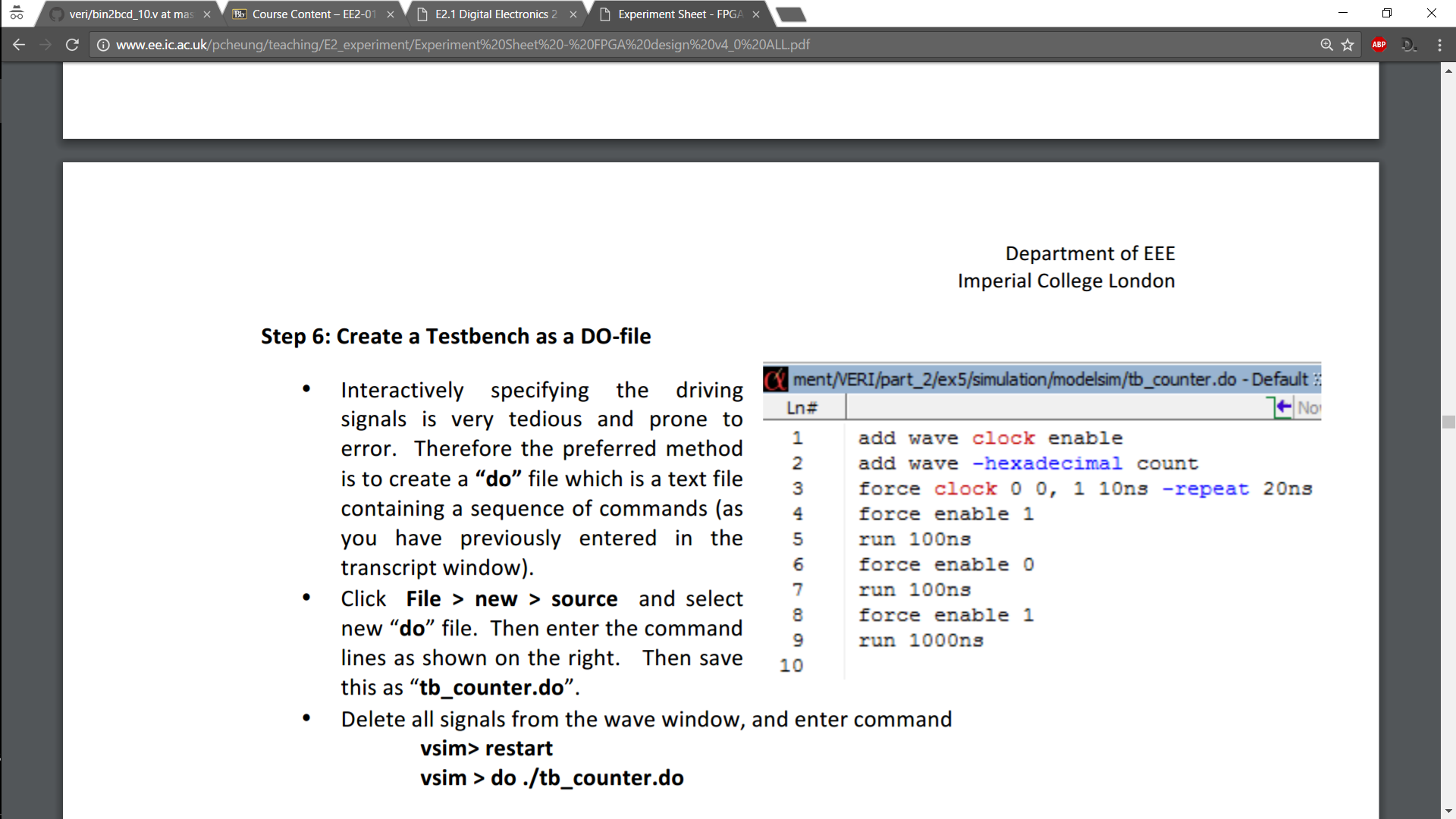
Verilog Experiment - Part 2

Experiment 5

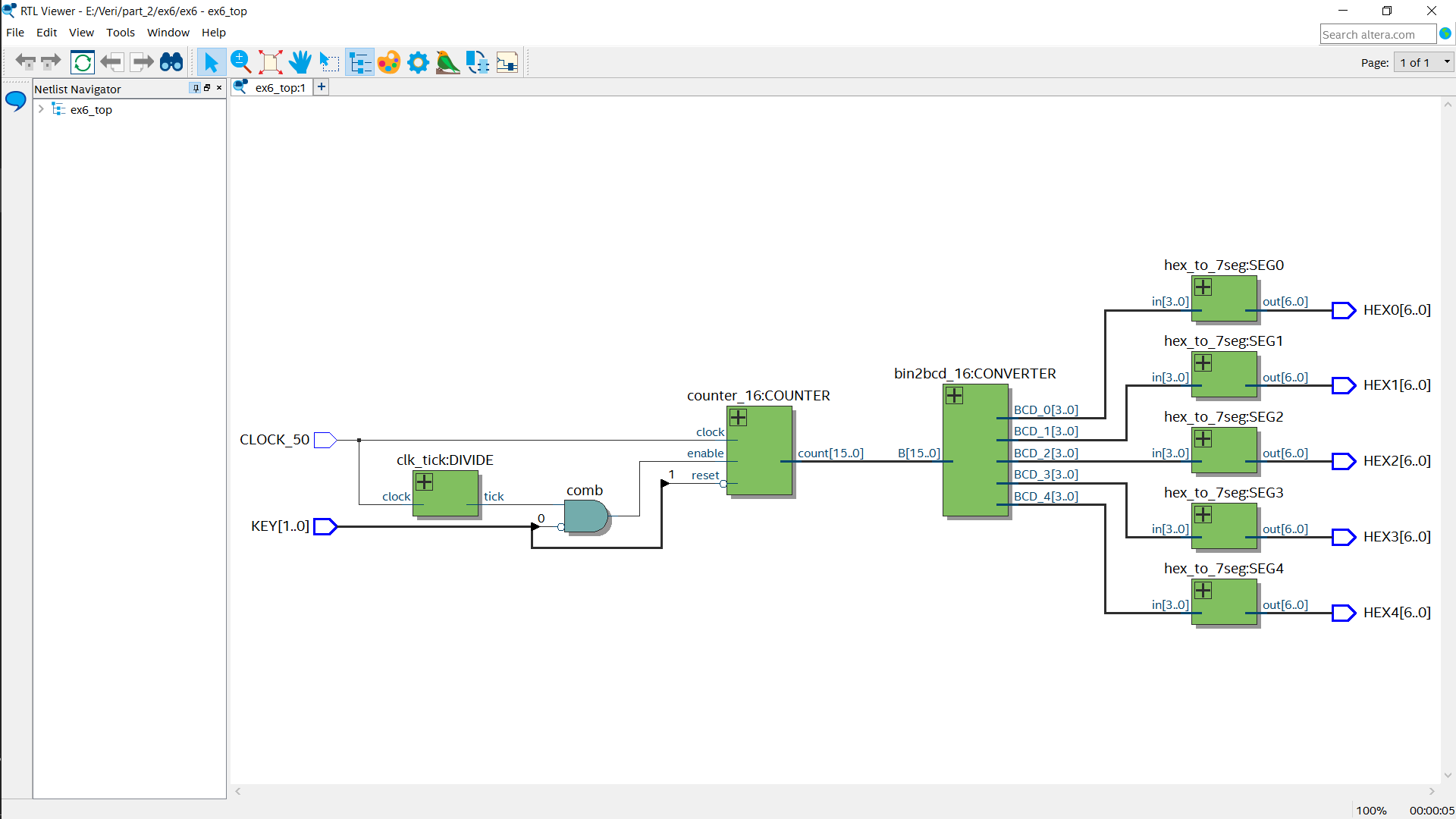
In this experiment we made an 8 bit counter. The first line indicates the unit time is 1ns with a 100ps resolution. The output is registered as its value is changing and we are using non- blocking assignment inside an always block which is positively edge triggered with clock. Parameters are used to easily change the bit size and thus the count length.

Below is the timing diagram of the counter using modelsim. As can be seen when enable is low the counter holds its value until enable is high again. It is also counting the correct sequence and can count up to 128.

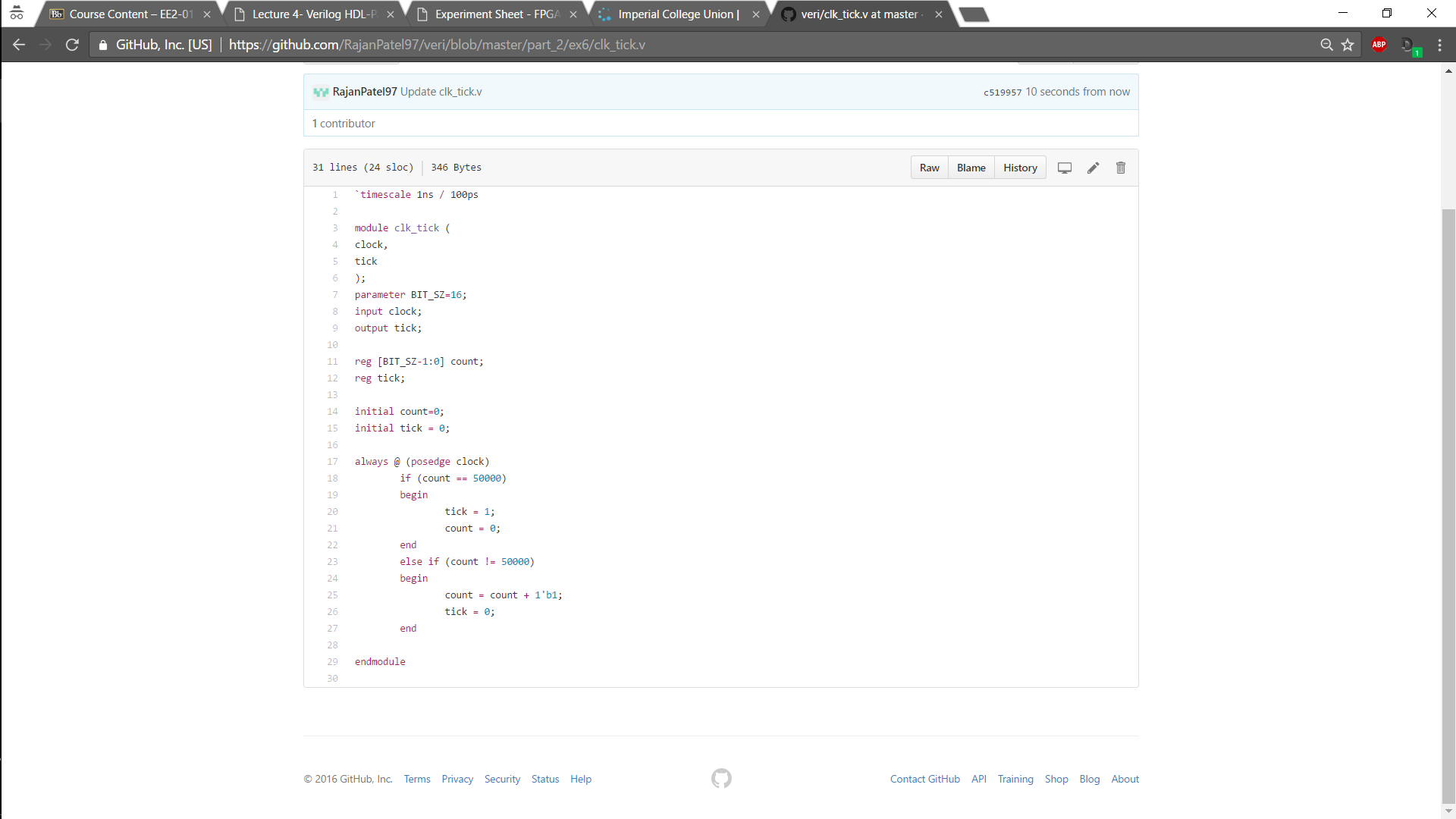


On the left is a do file which essentially allows us to test bench the counter without having to continuously enter commands and easily lets us change our designs. We can also single step through our Verilog code and see how it affects output.

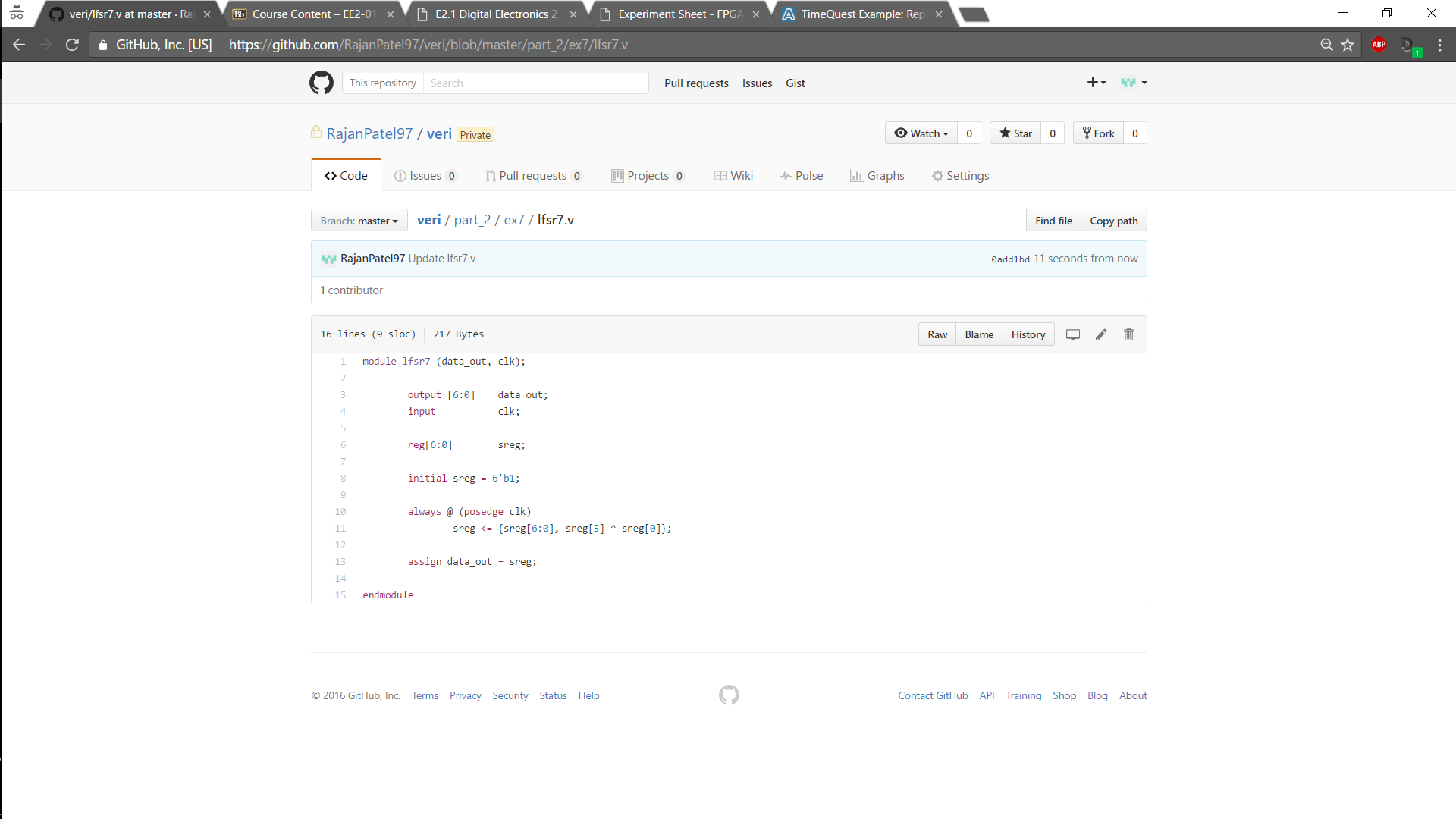
Experiment 6

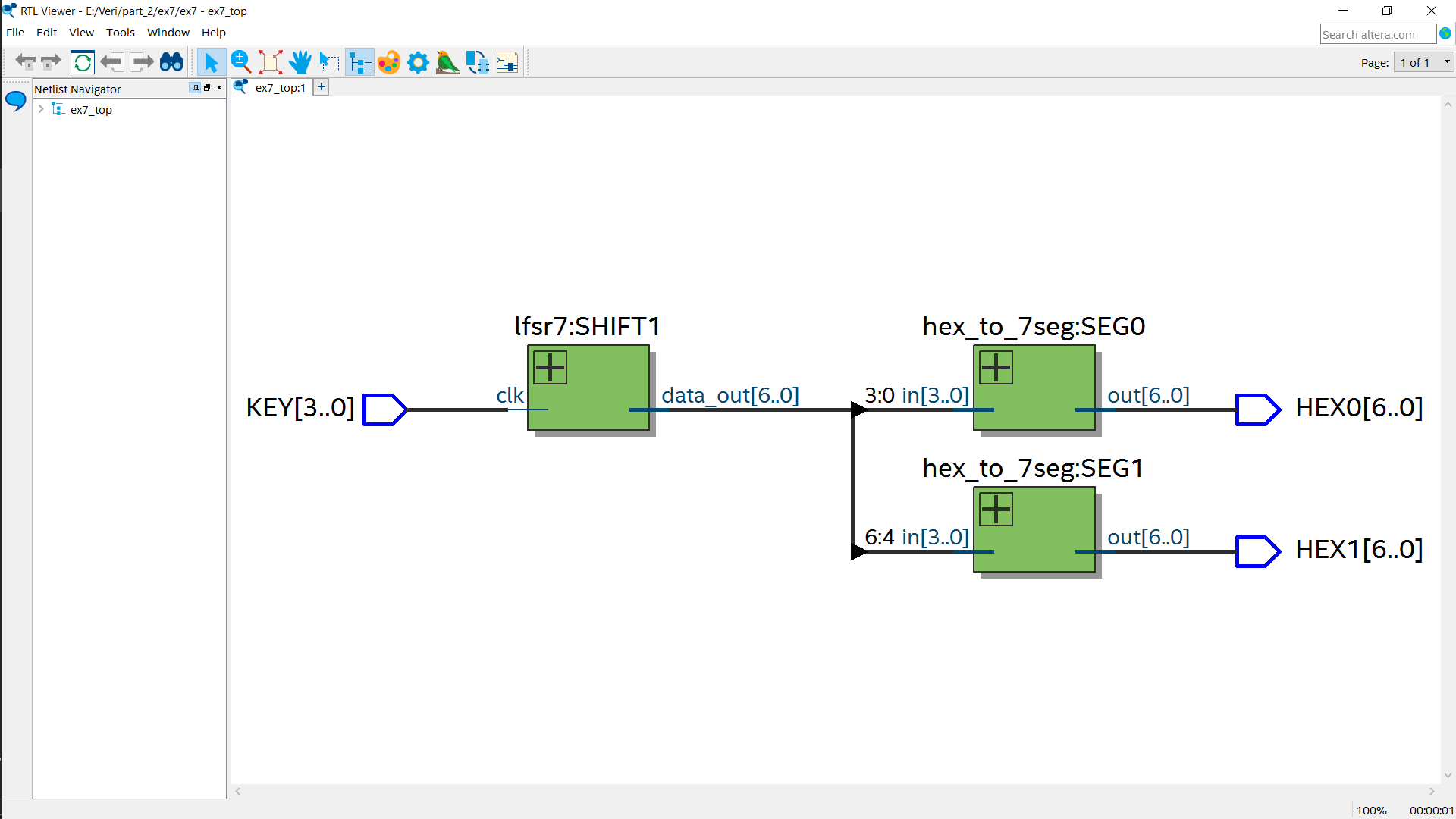
The 16 -bit converter worked fine and displayed the correct count sequence on the segment screens and reset when reset input was active low as specified. The max Frequency at 85C was 445.04MHz and at 0C was 422.48MHz. The TimeQuest entry was red due to unconstrained ports which are a type of unconstrained path, that are paths without any timing constraints specified to them. The report details the type of unconstrained paths: clocks, input ports, outputs ports. **“Altera recommends that all paths and ports be constrained to achieve optimal placement and fitting results.”**

Above is the top level RTL block diagram for the cascaded 16-bit counter, which uses a slower clock of 1KHz so that we can see the counter changing on the display. To do this I created a module called clk\_tick shown below, which produces an output every 50000 cycles resembling a clock that is x50000 slower than 50MHz and is fed into the enable input of the counter.



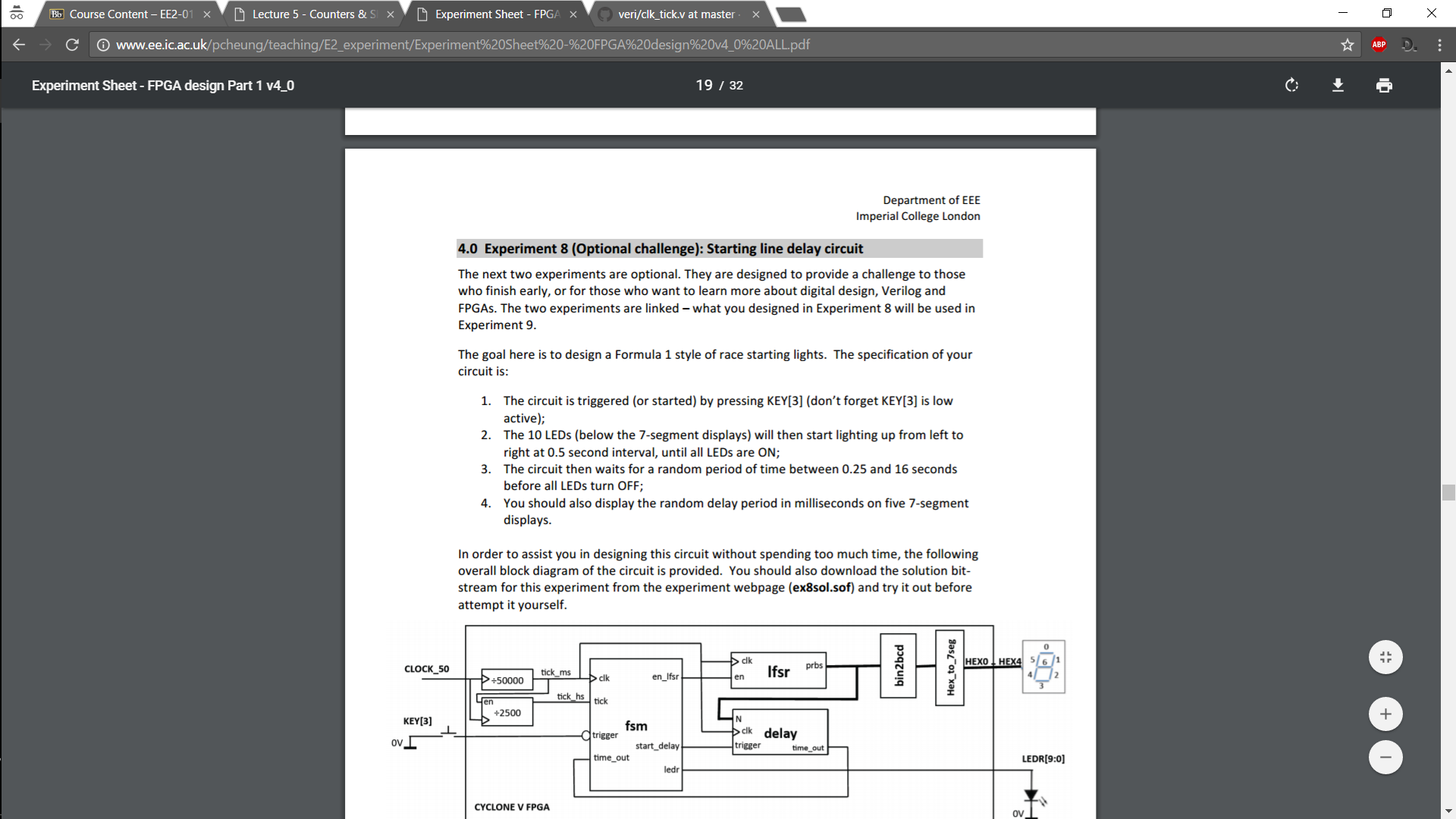
Experiment 7

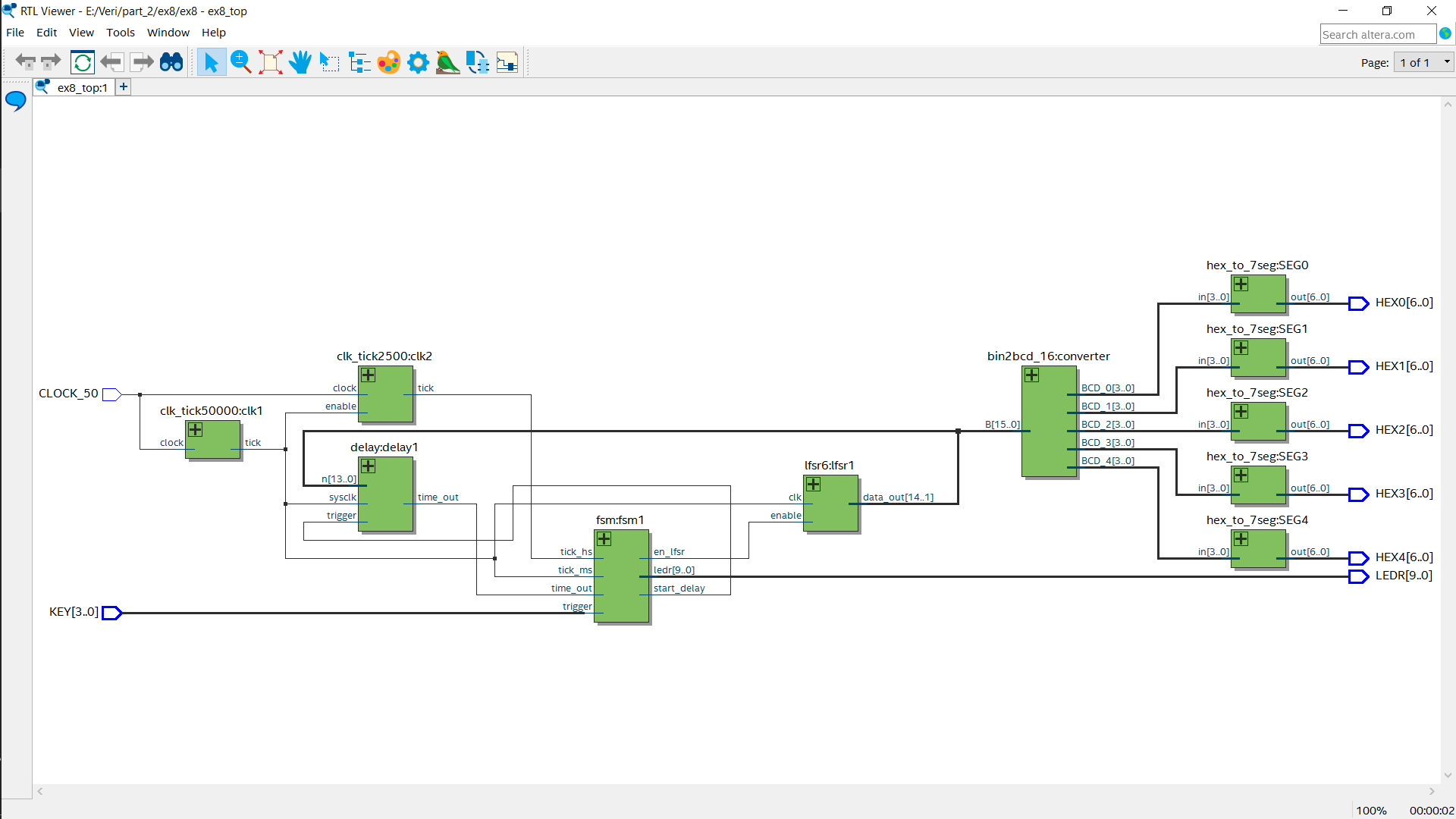


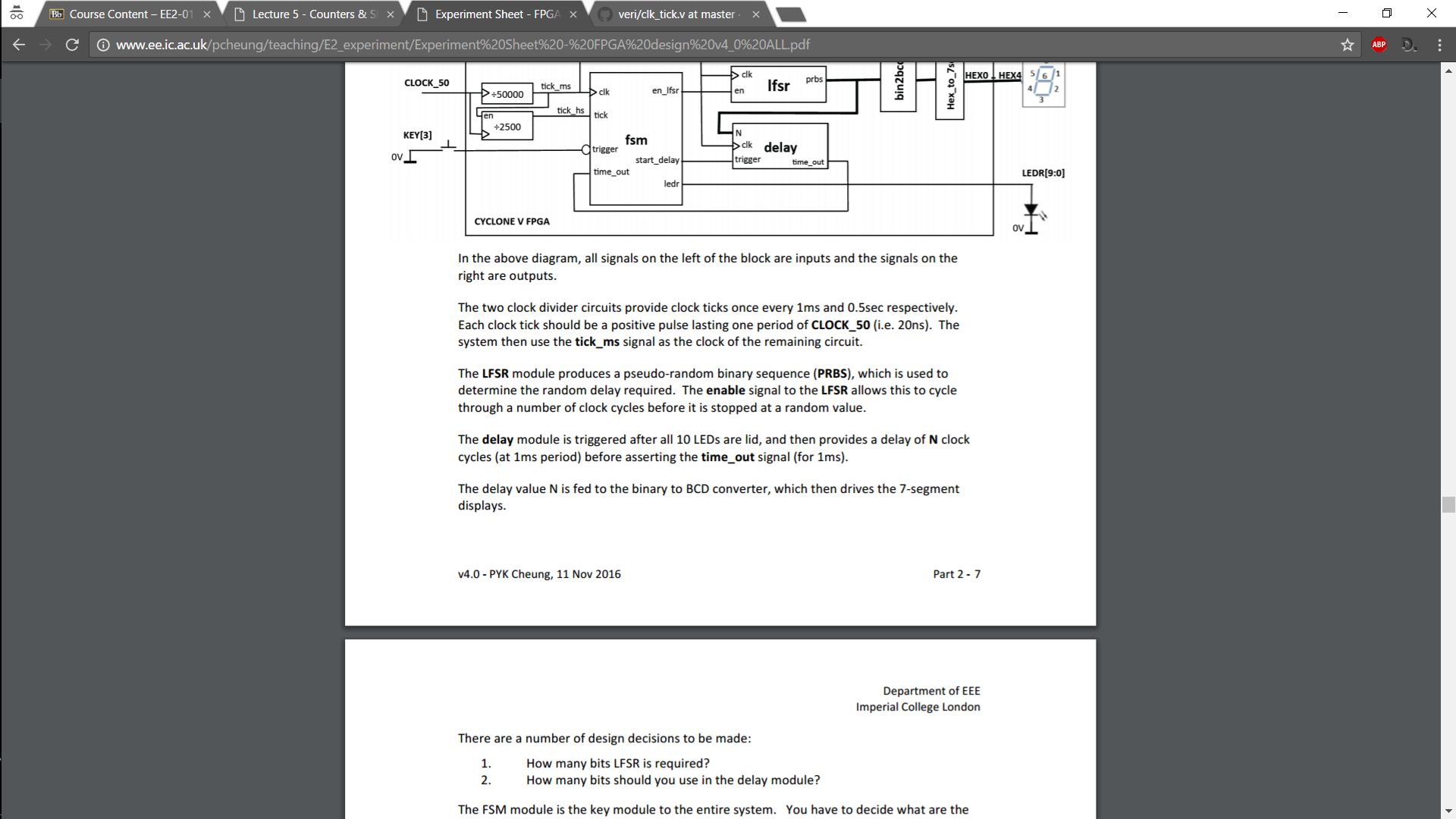
The LFSR outputs a PRBS sequence which is random. In this case it is 7 bit, so the max length sequence is a 127 cycles. In implements the primitive polynomial 1 + x + x^7. KEY[3] is used as the clock to cycle through the values. x and x^7 are tapped off at registers 1 and 7 and then used as inputs to an XOR gate.

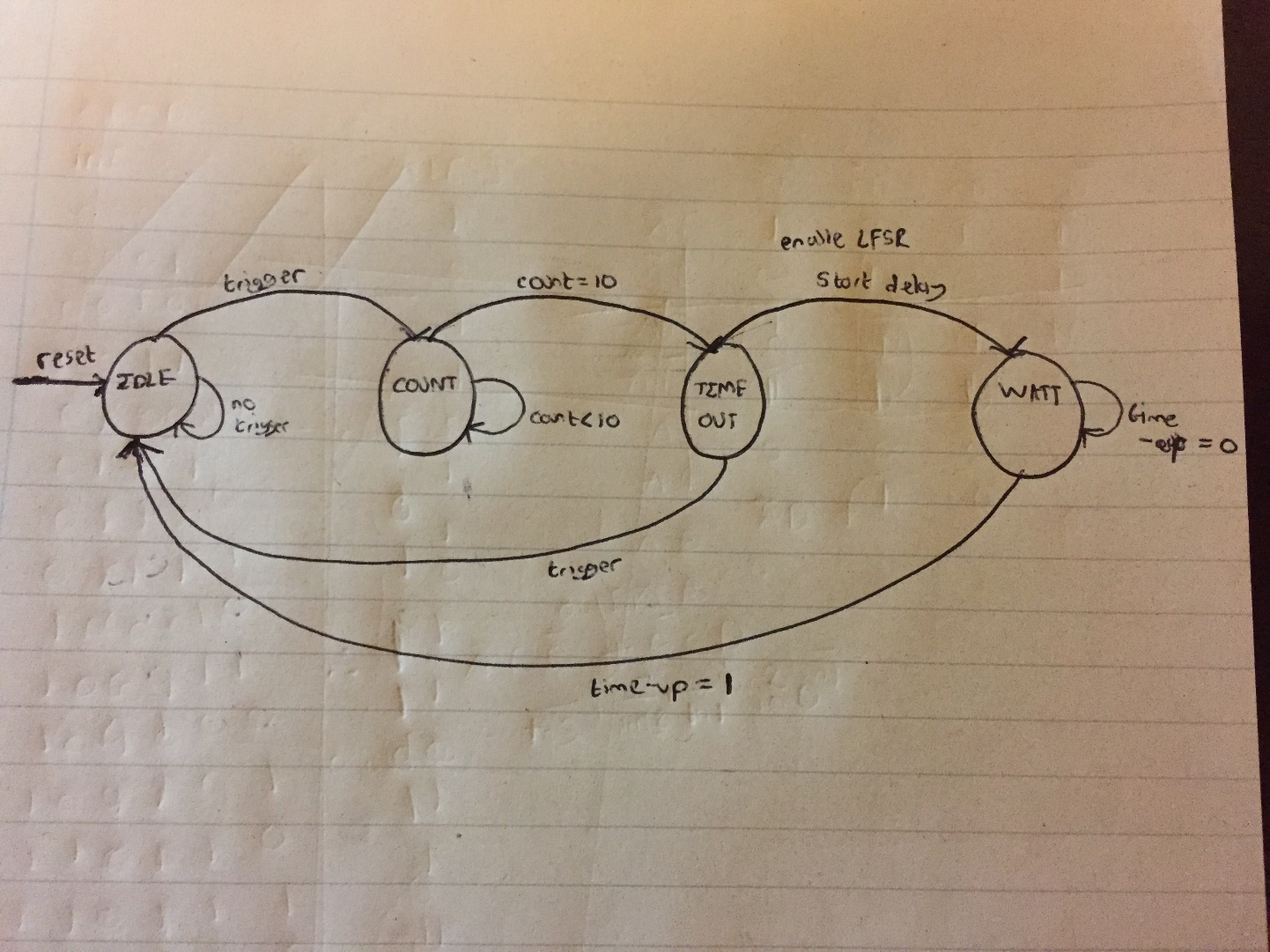
Above is the block diagram for the top – level. As it can be seen that at each clock the lfsr gives one output to 2 segs.

Experiment 8 & 9 (Optional)

The circuit below for experiment 8 has the following behaviour:



Furthermore, more specifically:

In order for the circuit to work correctly the LFSR must be 6 bits \* 250 so truncate to using 14 bits and the value of N in the delay module must also be 14 bits as it counts in milliseconds, so max delay is 16s which is required. Below is the FSM which has been simplified slightly in its representation than from the Verilog code that it describes.

Below is the RTL top-level block diagram from experiment 9. It counts the time from when the LEDs go off and when you press down KEY[0] and then displays your reaction time on the seg displays. All I had to change was to add another output to the FSM which goes high when the count is finished and use that as an enable to a counter that holds its value and outputs it when KEY[0] is held down.

