Assignment 12

Finite State Machines and FPGA I/O

Embedded Logic Design

November 7, 2015

1 Description

In this assignment you will have to use the Basys2 board driving the 7 segment LEDs (`AN3...0`) located on the board.

1. In Verilog HDL describe a hardware that is able to generate a clock frequency f_0 of approximately 3Hz. Display this clock by connecting it to LED `LD7` to verify your approach.
2. Based on your experience gained in 1, implement two more modules whose output is $f_1 = \frac{f_0}{2}$ and $f_2 = \frac{f_0}{4}$. Show both clocks by toggling LED `LD6` and `LD5` accordingly. Test and verify your approaches thoroughly since you are going to need your knowledge at the problems below.
3. In Verilog HDL describe a module that is able to illuminate one LED of the left most 7 segment display (`AN3`). After a specific time t has passed, the next LED is illuminated while the previous one is switched off. Refer to figure 1 to see the required pattern. "Rotate" the LED in an anti clock wise pattern and set $t = \frac{1}{f_0}$.
4. Create an FSM which takes as input the clock of the FPGA, a reset signal, f_0, f_1, f_2 and the state of buttons `BTN3` and `BTN2`. The output of the FSM is the desired clock frequency for the 7 segment LEDs. The following functionality is to be implemented:
   - If button `BTN3` is activated, the output frequency of the FSM changes to $f_n = f_{n+1}$ with $n = \{0, 1, 2\}$.
   - If button `BTN2` is activated, the output frequency of the FSM changes to $f_n = f_{n-1}$.

Always think about bottom up and start with small modules. Club together small modules to bigger ones to obtain modules with a more sophisticated functionality. This approach also allows you to reuse already implemented functionalities.

In any case optimize you FSM to minimize hardware utilization. Draw a graph and a state transition table of your FSM. Mark the required outputs and state transitions. How many states do you need?

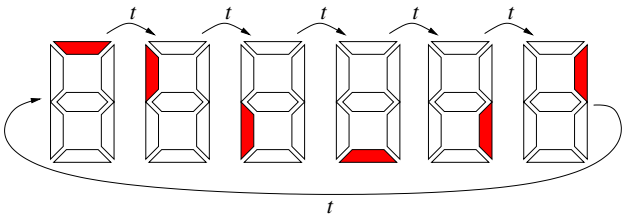

Figure 1: The pattern changes every t seconds.

2 Deliverables

All files must be submitted to `nanu.iiitd.edu.in` via `git` or `subversion`. Late submissions are not evaluated nor will be submissions through `https://www.usebackpack.com` or mail. Your repository has to contain:

- Verilog code of all required modules
- Verilog code for the test bench(es)
- Documentation including the answers about your FSM.

2.1 Remarks

If you encounter a problem, ask Google, DuckDuckGo, Bing, etc. first. The TAs will not type the question that you have, into the mask in the search engine for you. Required resources, textbooks, etc. are available on the ELD course website of `https://www.usebackpack.com` or in the Internet (datasheets, AVR library documentation, etc.)