Digital Circuit Lab — Lab 1

Roll Call Machine

Date: 03/14/2017

Team 05 Member: 郭子生 b03901032、曾耕森 b03901154、楊其昇 b03901101

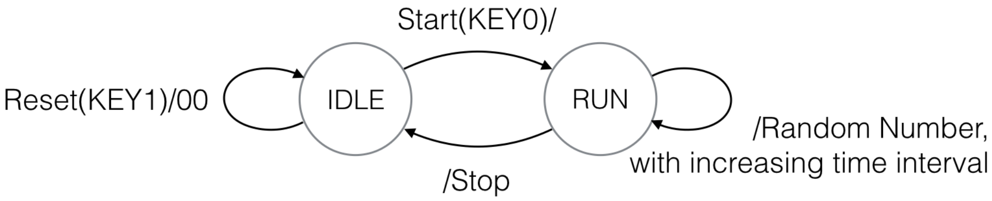
Github: https://github.com/awinder0230/Digital-Circuit-Design-Lab

- **Introduction**

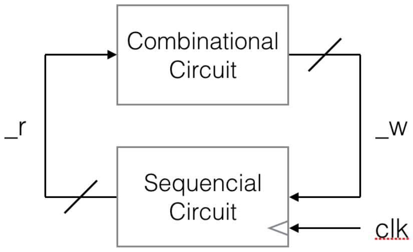
  A roll call machine which generates a random number range between 0 to 15 every time after pressing the start button.

- **Tools**: Altera DE2-115 FPGA board, System Verilog, Quartus II
- **Usage**
  - Clone from Github
  - Connect DE2-115 FPGA board to your computer by USB
  - Install Quartus II on either Windows or Linux
    1. Open Quartus II，choose project file “DE2_115.qpf”
    2. Compile the project by pressing Ctrl + L
    3. Open Tools → Programmer
    4. Click on “Hardware Setup”, choose the FPGA board which is connected to your computer by USB
    5. Click on “Add File”, select “DE2_115.sof” under directory “output_files”
    6. To upload codes to FPGA board temporally, select “JTAG” mode, switch FPGA board to “run” mode, and click “Start” button in Programmer to start uploading
    7. To upload codes to FPGA board permanently, select “Active Serial Programming” mode, switch FPGA board to “Program” mode, and select file “DE2_115.pof” to upload
    8. To convert .sof files into .pof files, simply click on File → Convert Programming Files
  - FPGA board
    1. Press reset button KEY1 for initialization purpose
    2. Press start button KEY0 so as to get a random number every time
- **Simulation**
    1. Premise: to run simulation, you should have the permission to access NTU DCLab server
    2. Login to DCLab sever
    3. Upload codes to server by “scp” command in terminal, or use MobaXterm if your are a Windows user
    4. Enable ncsim for simulation purpose and enable nWave to checkout output waveform
    5. Change directory to sim/
    6. Type "make TEST=LAB1 TOPLEVEL=DE2_115 SV=1" to run GUI simulation
- **Modules**
  - **SevenHexDecoder.sv**: Given a four bit input signal, decodes binary input into 2-digit decimal integer, and shows the number on seven-segment displays.
  - **Debounce.sv**: Due to the instability of buttons on FPGA board, debounce module receives a press-down signal from buttons on FPGA board, and provides a much more stable signal as output for the module DE2_115.
  - **Top.sv**: Given a start signal, generates several random numbers range between 0 to 15 with increasing output time intervals. Also, given a reset signal, output zero.
  - **DE2_115.sv**: Combines modules “SevenHexDecoder”, “Debounce”, and “Top” together, and acts as an interface between Verilog codes and FPGA board. As an user presses down the start button, Debounce module receives the signal and sends a more stable signal to Top module, Top module then generates several random numbers and finally stops at one range between 0 to 15. All the outputs generated by Top module are sent to SevenHexDecoder and being shown on seven-segment displays.

- **Design**
  - **Finite State Machine**: Use enum in SystemVerilog to record your state.



  - **Circuit Implementation**



  - **Random Generator:** We apply linear congruential generator algorithm to create pseudo-randomized numbers. The advantages of this algorithm are fast and minimal memory requirement. Since the situation does not require high-quality randomness, the algorithm works well. Also, we only use bits 23 ~ 6 to increase randomness instead of full bits because the latter has shorter repeating cycle. The random generator was implemented in "Top" module.