CSIE ADIC Lab 2023

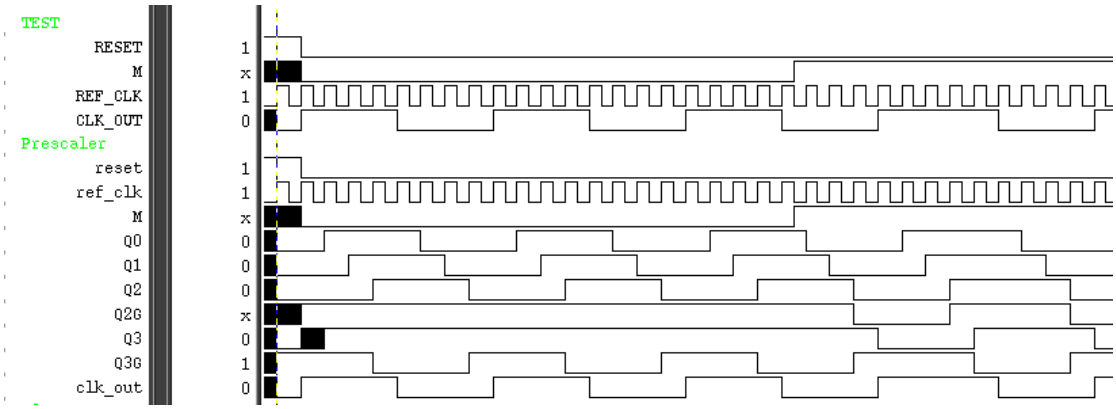
Lab04B. Build-Up DCO model and PFD model with NC-Verilog

1. Data Preparation

1. Extract LAB data from TA's directory:
   % tar xzvf ~adicta/lab04b.tar.gz
2. The extracted LAB directory (**lab04b**) contains:
   a. Demo1/ : Prescaler verilog demo code
   b. Demo2/ : DCO Verilog model demo code
   c. Exercise/ : DCO and PFD Verilog model design

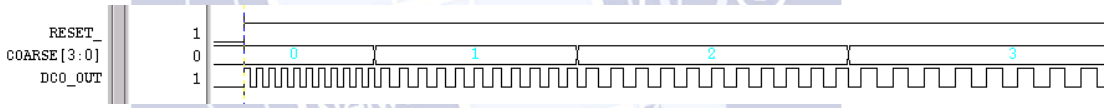
2. Run Prescaler Demo1

1. Change to the directory: **Demo1**
2. Open and read the demo1 Verilog codes.
   % gedit *.v &
   This is the prescaler circuit which is shown in **02_All-Digital_PLL.pdf, page 57**. The input clock (ref_clk) will be divided by 8 or 9 with different M control pin inputs.
3. Execute NC-Verilog to simulate demo1.v
   % ncverilog demo1.v
   A log file named ncverilog.log will be generated. It records error or warning messages during Verilog simulation.
4. Start Verdi to debug the output waveform:
   % nWave &
5. In nWave, open **demo1.fsdb**, and restore the saved signal file **demo1.rc**. Then you can see the simulation results. Note: The output duty cycle is not 50% when input is divided by 9.



3. Run DCO Model Demo2

1. Change to the directory: **Demo2**
2. Open and read the demo2 Verilog codes.
   % gedit *.v &
   This is an example of a DCO that uses a linear delay model. The min. period of DCO output and the step size in DCO control are specified with parameters in **dco_model.v**. In **dco_model.v**, a procedure block that takes negative edge reset_ as event input is used to disable the DCO in reset.
3. Execute NC-Verilog to simulate demo2.v
   % ncverilog demo2.v
   A log file named ncverilog.log will be generated. It records error or warning messages during Verilog simulation. The test module in **demo2.v** will test all possible DCO control codes to verify the correctness of **dco_model.v**.
4. Start Verdi to debug the output waveform:
   % nWave &
5. In nWave, open **demo2.fsdb**, and restore the saved signal file **demo2.rc**.
   Then you can see the simulation results.



4. Build up the DCO Model

1. In this exercise, you must build up the Verilog behavior model of the DCO you designed in **lab05b**. You must use the simulation data from SPICE simulation to build up your DCO behavior model for further all-digital PLL simulation.
2. In this exercise, you also need to test all possible DCO control codes in your DCO to verify the correctness of the DCO model.

5. Build up the PFD Model

1. In this exercise, you need to build up the Verilog behavior model of PFD, which you designed in **lab05d**. You must use the simulation data from UltraSIM simulation to build up your PFD behavior model for further all-digital PLL simulation.
2. In this exercise, the dead zone of PFD should also be modeled. Thus you need to build up the functional model of PFD with a dead zone. In addition, the output delay of flagU and flagD must be modeled (using worst-case

delay).

System Integration
Silicon &
Implementation
Group.
ELECTRONICS
NCTU
Video
Telecom
Network
Graphics