ECEN 684 INTERNSHIP REPORT

Last name: Sathyanarayana

First name: Ganesh Prabhu

UIN: 127004441

Date: 10/25/2018

Semester & year of internship: Summer 2018

Semester & year of expected graduation: Spring 2019

Degree program (M.Eng/MS/PhD): M.Eng

TAMU Advisor: Dr. Scott Miller

ECEN 684 section number, found on Howdy:

ECEN 684 - 526

TAMU Advisor's email: smiller@tamu.edu

About the Company:

Biotronik Inc. is a medical device company that provides solutions for cardiac rhythm management, electrophysiology and vascular intervention. Some of the devices that are developed include pacemakers, implantable defibrillators, remote-monitoring systems for patients, bio-monitors, and neuro-stimulator.

Internship Project: Verification of the digital control unit of a neuro-stimulator chip

Objective:

The objective here was to verify the complete functionality of the digital control unit and its interactions with the analog sensing and actuating unit. This was achieved by implementing test cases at the chip level that cover all functionality specified in the design specification at the inception of the design and also aiming for code coverage and toggle coverage of the design. In order to aid the speed of this process a test generating script was developed.

Background:

Neuro-stimulator provides therapy for chronic pain by delivering low voltage electricity to the peripheral nevres. It helps supress the pain by preventing the signals from reaching the brain. The function of the neuro stimulator can be broadly calssified into the analog section that senses and stimulates the peripheral nerves, and the digital section that controls the stimulating electrodes. The figure below depicts the digital control block under verification. This block controls the type of therapy that can be provided based on the pulse train that drives the stimulation electrode.

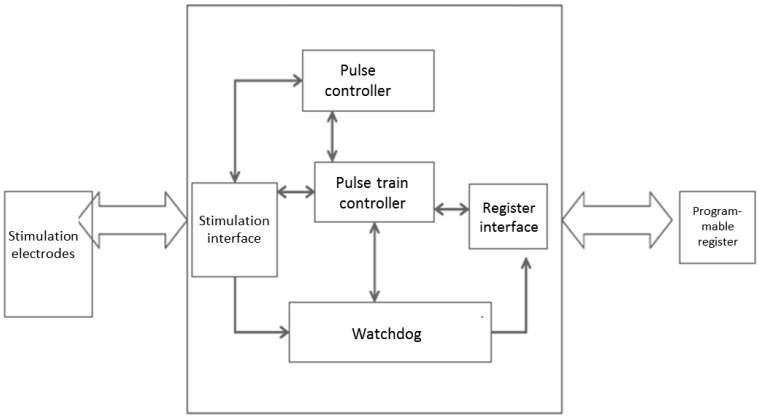

Figure 1: Stimulation control block

Tasks:

The verification process that was tasked follows four steps,

- Development of basic testcases
- Automating testcase generation
- Running regression analysis to identify coverage holes
- Generate testcases to improve coverage

Step 1:

The development of testcases at the chip level was realized by simulating the digital block by loading the programmable registers with typical and corner values and running assertion checks and examining the waveforms. The state machines of the pulse and pulse train controllers were evaluated the various possible inputs and modes of operation. The inputs and modes were randomized in certain cases and evaluations were made. Around 45-50 testcases were run and analysed before the regression.

Step 2:

The test case generation was automated by developing a python script that read values from the CSV file and manipulates them to write to the programmable registers in the SystemVerilog test code. The test case is divided into sections that need to be/can be modified for a particular testcase. These sections are stored in separate text files.

The input parameters of the testcase are given through a csv file which has the following format; the first column has the parameter names; the following columns represent the values of the parameters for each train. This function defined in the script reads the value from the csv file and performs bit manipulation to program the register. The script allows the specification of the testname, path where the test is to be stored, number of pulse trains, the input csv file.

Step 3:

Regression analysis was performed using vManager tool by Cadence Inc. The focus here was on code coverage, particularly toggle coverage of all the important signals. The metrics generated by the tool provide insight into how thoroughly the code of a design is exercised by the tests, based on which testcases covering the sections that were not exercised were developed. The automation script was used to aid the testcase generation. The goal was to get near perfect coverage before tapeout.

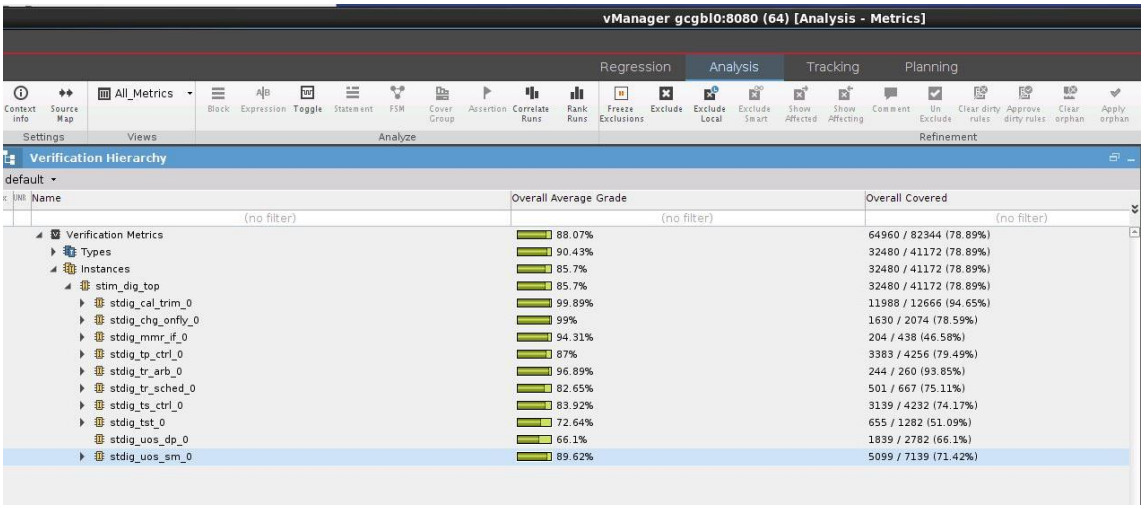

Figure 2: vManager regression analysis

Step 4:

This is a feedback step where, step 1 is repeated again with the help of the script generated in step 2. Multiple runs for different randomizations are performed and the coverage results of those runs are analyzed to implement tests that cover those holes.

Project Schedule:

Week 1: • Setting up workspace. • Reading training documents	Week 2: • Getting acquainted with the design and verification environment • Reading training documents
Week 3: • Training on Running testcases • Understanding and modifying the testcases as required	Week 4: • Reading simulation test cases • Verifying the correctness by examining waveforms
Week 5: • Implementing new testcases • Running new testcases	Week 6: • Implementing new testcases • Running simulations of the new testcases
Week 7: • Developing test generation script • Testing the validity of the script	Week 8: • Developing test generation script • Testing the validity of the script
Week 9: • Running regressions on vmanager • Analyzing coverage results	Week 10: • Developing new testcases • Running simulations of the new testcases • Re-running regressions

Project Evaluation (by Linh Nguyen):

Ganesh is a talented engineer. He quickly picked up the design concept, tools manipulation and jumped into help with test cases development in a short time with a minimum help. He also came up with the idea to automate the test case generation process using python script with minimum guidance. It helps to speed up the verification effort for the team. It proves that he has a great programming skills and knowledge of computer architecture. These skills sets will be a good start for him in the future career.

For future improvement, Ganesh should pay more attention to the details and needs to check the results carefully before submitting work. I can see he is getting better days by days based on my feedback.

Overall, he did an excellent job, beyond my expectation as an intern. Thank you for helping out with neuro project and good luck for your future career.