ECEN 684 INTERNSHIP REPORT

<u>Last name</u>: Sathyanarayana

First name: Ganesh Prabhu

UIN: 127004441

Date: 7/25/2018

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 - 326

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: Development of System Validation Tester (SVT)

Objective: Parse clinical IEGM (Intracardiac ElectroGraM) data files and send them to inputs of the IC under test. The IEGM data is test data that is similar to the ECG signals received by the pacemaker under normal operation. Create a user-friendly way to communicate with, and program the processor of the IC under test. Any command or program to the processor is handled by the UART interface of the IC.

The IC under test operates on 1.5MHz and 32KHz clocks, and the IEGM data is processed by the sense path of the IC. Serial interface of the sense path has three inputs, Trig_in, SER1_in, and SER2_in. The Trig_in is turned on every 64 clock cycles (1.95ms) of the IC. The trigger signal is sampled at the falling edge of the 32KHz clock. The 16-bit clinical data are fed to 8 channels of the time-division-multiplexed slots through SER1_in and SER2_in, which ultimately mapped to 8 DMA channels. Data of channels 0 to 3 are through SER1_in and channels 4 to 7 are through SER2_in. The timing diagram is as shown below.

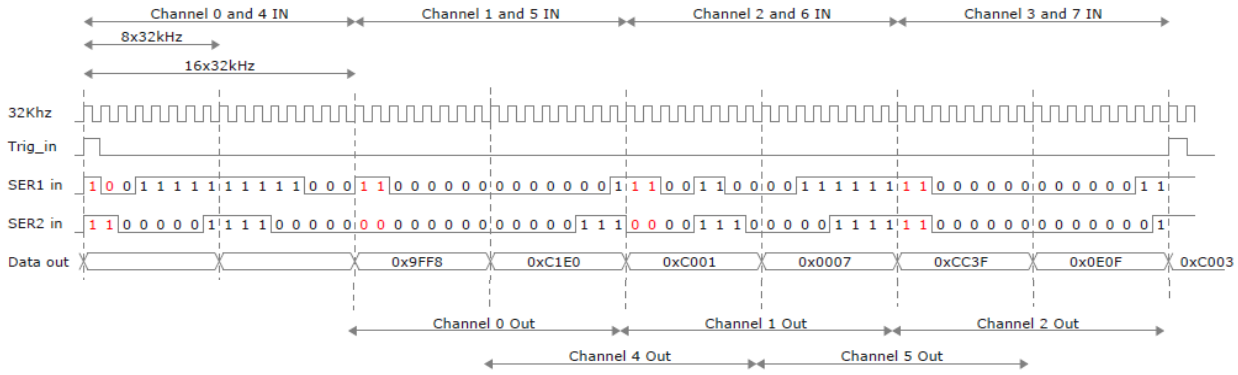

Figure 1: Timing diagram of serial IEGM interface

Features of SVT:

- Can download and run RAM Application.
- Can access any memory location (register, sram, rom) within IC.
- Can inject data sample into the IC's sense path, all 8 channels are supports.
- Can parse user file and issue all of the above operations.

Solution: This was achieved by developing a command line interface (CLI) on the PC to enable users perform different tests on the IC and communicate with the processor by sending system specific commands using UART protocols. A FPGA design was implemented to receive and differentiate between commands and data sent by the PC and route it respectively to the processor and serial IEGM inputs of the IC under test.

Figure 2: System level diagram for SVT

The CLI was programmed in C#, it provides users options for different operations that can be performed and based on the operation sends commands and data to the processor and informs the user of a response from the processor. Figure 3 shows the interface.

The FPGA operates on 12MHz clock and this is divided into 1.5MHz and 32KHz and feeds into the IC. Figure 4 shows the FPGA design for the system.

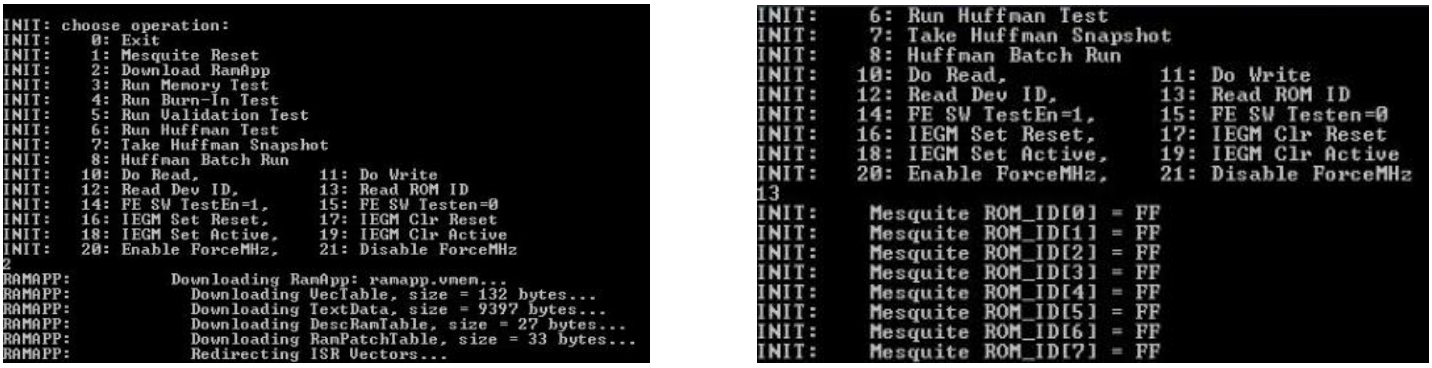

Figure 3: Command line interface

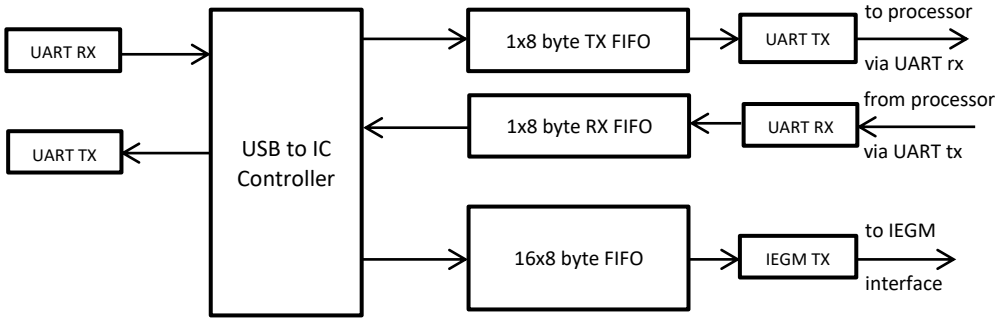

Figure 4: FPGA design

UART receivers and transmitters handle the UART protocol. They receive inputs from CLI and drive inputs to UART interface of IC and send responses from FPGA and the IC to the CLI. *USB to IC controller* receive commands from the CLI referring to the different operations that can be performed on the IC, based on which it performs the required procedure. Any data or command to or response from the processor is routed through an *8 byte-deep TX and RX FIFOs*. In case of test IEGM interface command the data is stored in *16 byte-wide FIFO* where each byte corresponds to each channel of the DMA. In the event that the FIFO is full the controller requests the data to be sent again. The FIFOs act as buffers between the PC and IC which operate at different speeds. The *IEGM transmitter* reads the data from the FIFO and converts it to the form as shown in the timing diagram in Figure 1.

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 Firmware • Understanding and modifying the firmware as required	Week 4: • Reading simulation test cases • Designing IEGM Transmitter
Week 5: • Designing the IEGM transmitter • Integrating FIFO design	Week 6: • Designing the USB to IC controller • Running simulations and verification
Week 7: • Designing the USB to IC controller • Running simulations and verification • High level design	Week 8: • Programming FPGA with the design • CLI programming
Week 9: • Adding features to CLI • Run the SVT and perform certain tests	Week 10: • Modifications to CLI • Run the SVT and perform certain tests

Project Evaluation (by Martin Li, Member of Technical Staff):

Ganesh completed this project professionally. The scope of the project involved multiple engineering disciplines including FPGA design/verification in Verilog, firmware development in C/assembly, software application development in C#. All of these tasks were not trivial even for a seasoned developer, but Ganesh was able to pick up these skills, completed all the assignments in a timely fashion, and handled them extremely well.

Most of the mistakes throughout the project he made were all due to the fact that he was not familiar with the full IC functionality, and this was fully understandable. Given his short 3-month of intern time, he was not given enough time to review and understand the full IC specification (which total span thousands of pages), and this was the only area that he required guidance and supervision heavily.

In general, Ganesh is a talented engineer, full of potential, and destined for great success in career.