AUTOMATED REGRESSION ENVIRONMENT FOR IMPROVING DESIGN VERIFICATION PRODUCTIVITY

By

Keshav Rath

(Roll No. 2013104)

SUPERVISOR(S):

EXTERNAL:

Mrs. RAMA KOWSALYA NAMBURI
Senior Project Manager
Canon ISDC, Bangalore

INTERNAL:

Prof. P. N. KONDEKAR
Professor
PDPM IIIT DM Jabalpur

Electronics and Communication Engineering (B. Tech 2013)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR

Report

(16th May 2016 – 22nd August 2016)

1. INTRODUCTION

The main purpose of this project is to enhance the design and verification productivity using PERL for automation. It will enhance the speed for the analysis of the reports which further help in the design and verification productivity.

Previously there used to be various scripts to run various ongoing projects

PERL is the most well-known language for the practical extraction of the files and reporting language. It has borrowed features from other programming languages as well. It provide a powerful text processing facilities, facilitating easy manipulation of text files.

For this project prior knowledge of PERL as well as VHDL has to be acquired. Some knowledge about verification methodology (such as OVM, UVM) is also required.

2. PERL VHDL AND UVM

2.1 PERL

PERL stands for Practical Extraction and Report Language. It is a scripting language that has been developed by Larry Wall in 1987. This language was initially developed as a general purpose Unix Scripting Language but later it became a different independent Language.

For this language all the basics were learnt, which include the basic syntax and semantics to using external CPAN modules for better productivity.

2.1.1 CPAN

Perl has mechanisms to use external libraries of code, making one file contain common routines used by several programs. Perl calls these modules. The Comprehensive Perl Archive Network (CPAN) is a repository of over 150,000 software modules and accompanying documentation for 33,000 distributions, written in the Perl programming language by over 12,000 contributors. The CPAN's main purpose is to help programmers locate modules and programs not included in the Perl standard distribution. Its structure is decentralized. Files on the CPAN are referred to as distributions. A distribution may consist of one or more modules, documentation files, or programs packaged in a common archiving format, such as a gzipped tar archive or a Zip file.

For the understanding of this and to practice these modules available, following modules I have downloaded and practiced upon them.

CPAN Class: Spreadsheet

- It helps to access an excel file.
- It helps to read or write an excel file through the PERL code.
- Spreadsheet::WriteExcel – To write a string, number to an excel file (Excel 95-2003).
- Spreadsheet::XLSX – To read the content of the excel file (Excel 2007).
- Spreadsheet::ParseExcel – Also to read the content of the excel file (Excel 95-2003).
- Spreadsheet::WriteExcel::Chart – To draw a pie chart using the data in the excel sheet in the sheet itself.

Spreadsheet is a broad class in CPAN and it has the many modules in it. Some of them I have described above. All the above modules are not independent, they need some of the other modules for their implementation.

2.2 VHDL

VHDL stands for VHSIC Hardware Description Language (VHSIC – Very High Speed Integrated Circuit). VHDL is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. VHDL can also be used as a general purpose parallel programming language.

For making RTL designs we mainly use Hardware Description Language (HDL).The mainly used HDL these days is SystemVerilog and VHDL language. The HDL that is used for this project is SystemVerilog.

Hierarchical Modeling Concepts in SystemVerilog tells about writing modules in SystemVerilog language. Different Methodologies of design hierarchy like top-down design methodology, bottom-up design methodology are followed in SystemVerilog language. These methodologies are used to write code in SystemVerilog for the digital circuits that are available. In bottom-up methodology, we first identify the building blocks that are available to us. We build

bigger cells using these building blocks until we build the top level block. Bottom up design methodology follows the vice-versa. Different levels of coding that are present in SystemVerilog language are Behavioral or Algorithm level, Data Flow Level, Gate Level, Switch Level. In Algorithm level designing is done very similar to C language. At this Data Flow level the module is designed by specifying the data flow. At the Gate Level the module is implemented in terms of logic gates. At Switch level the module is implemented in terms of switches, storage nodes, and the interconnections between them.

Different blocks that are available in SystemVerilog language are design block and simulation block. In design block we will write the code for the particular design and the simulation block is used for test bench purpose.

Different lexical conventions such as comments, operators, number specification, strings, different data types like value set, Nets, Registers, Vectors, integer, real and time register data types are used in SystemVerilog. This SystemVerilog language is quite similar to C language in using different functions for example if-else, case statement, for loop etc.

2.3 UVM

The Universal Verification Methodology (UVM) is a standardized methodology for verifying integrated circuit designs. UVM is derived mainly from the OVM (Open Verification Methodology). The UVM class library brings much automation to the SystemVerilog language such as sequences and data automation features (packing, copy, compare) etc.

3. RESULTS AND DISCUSSIONS

- Initially PERL and its basics were studied.
- The study of CPAN provides an efficient way for processing the text document and display the output in the ordered manner.
- The study of VHDL and UVM are going to be useful in the next part of the project.

4. CONCLUSION

- I got a hands on experience on PERL language and its external libraries storage like CPAN which is a collection of external libraries and Template toolkit which has inbuilt many libraries which I have mentioned in the beginning.