**PERL** **SCRIPT** **FOR** **AUTOMATION OF** **SINGLE** **AND** **REGRESSION** **TEST**

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

(12th July 2016 - 26th July 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.

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 C, AWK, sed (stream editor), sh (shell scripting), and BASIC (Beginners’ All-purpose Symbolic Instruction Code) as well. It provides 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 basic 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. Perl is a stable, cross platform programming language. It is also called 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. This language is used for application such as graphics programming, system administration, network programming, finance, bioinformatics, and other applications.

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](https://en.wikipedia.org/wiki/Software_repository) of over 150,000 [software modules](https://en.wikipedia.org/wiki/Modular_programming) and accompanying documentation for 33,000 distributions, written in the [Perl](https://en.wikipedia.org/wiki/Perl) [programming language](https://en.wikipedia.org/wiki/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 gzip 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. Some of those dependant modules are:

* Crypt::RC4 – To use the RC4 encryption algorithms
* Digest::MD5 – Used for RSA data encryption.
* OLE::Storage\_Lite – For Document Interfacing
* ExtUtils::MakeMaker – For creating Makefile module