AUTOMATION OF IP REPORTING SUBSYSTEM

Submitted By

MEDHA PARASHAR

13MCEN07

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2015

AUTOMATION OF IP REPORTING SUBSYSTEM

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Networking Technologies)

Submitted By

MEDHA PARASHAR

(13MCEN07)

Guided By

PROF. USHA PATEL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

MAY 2015

Certificate

This is to certify that the major project entitled **”AUTOMATION OF IP REPORTING SUBSYSTEM”** submitted by **MEDHA PARASHAR (Roll No: 13MCEN07)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Networking Technologies (CSE) of Institute of Technology,Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Prof.Gaurang Rawal PG Coordinator - NT, Nirma University, Ahmedabad.	Prof. Usha Patel Internal Guide, Nirma University, Ahmedabad
Mrs. Jyoti Kumar Manager, STMicroelectronics, Greater Noida	Mr. Rishabh Bansal Mentor, STMicroelectronics, Greater Noida
Dr. Sanjay Garg Professor and Head, Nirma University, Ahmedabad	Dr K Kotecha Director, Nirma University, Ahmedabad.

Statement of Originality

I, **MEDHA PARASHAR**, Roll. No. **13MCEN07**, give undertaking that the Major Project entitled "**AUTOMATION OF IP REPORTING SUBSYSTEM**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date:
Place:

Endorsed by
Prof. USHA PATEL
(Signature of Guide)

Acknowledgements

It is giving me pleasure to express thanks and gratitude to Manager, ST Microelectronics, **Mr. RISHABH BANSAL**, Mentor,ST Microelectronics,**Prof. USHA PATEL** Associate Professor, Computer Science Department, Nirma University, for their guidance and encouragement provided by them throughout the period. Due to their constant support and motivation, I have been able to attain my goal. The guidance provided by them has helped in improving my intellectual maturity ,that will help me a lot in the time to come.

I also want to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, for his support and also for providing a good infrastructure and healthy research environment.

A great obligation towards textbfDr K Kotecha, Hon'ble Director, Institute of Technology, Nirma University for his motivation through course of work.

And a thank you also fro the Institution, all faculty members of Computer Engineering Department, Nirma University.

- MEDHA PARASHAR
13MCEN07

Abstract

To build a subsystem which performs the task of data integration i.e. a combination of data from different sources and to give user a output of all these data with the help of report generation. All the manipulation is done on the data , so that the relevant information can be extracted and presented to the user.

The frontend is also designed in such a way ,so that the users can compare the data they are getting after manipulation and further improvement can be done on it. For making comparison more better we also provide results in the form of charts. So that makes data more comparable in perspective of the user. The backend is based on a Pentaho tool and Perl language. Perl is regarded as a very efficient language in terms of data integration.

ETL method of data integration is being followed here, as has been choosen to be the best for this project, due to its certain advantages. And apart from this BIRT tool is used for the generation of reports. This makes it more simpler for the user.

Abbreviations

IP	Intellectual Property.
SOC	System On Chip.
ETL	Extraction Transformation Loading.
KPI	Key Performance Indicator.
VSAM	Virtual Storage Access Method
IMS	Information Management System

–

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Our aim is of building a subsystem which performs the task of data integration i.e. a combination of data from different sources and to give user a output of all these data with the help of report generation. The form of data integration used here is ETL. ETL is the short form for the complete word extract, transform, and load. Definition: It can be defined to be a set of processes for getting data into data warehouse from OLTP systems. Apart from OLTP systems , data is also coming from email databases, websites,spreadsheets,files and personal databases like Access.

1.1.1 What is ETL ?

The main ETL steps can be categorized into:

1.Extract: Extraction can be said to have a connection of bunch of data sources , then extraction of data from these data sources and the last step is to make this data available for all the necessary processing steps.

2.Transform: Transformation can be defined to be the passage between the extraction of source data and the loading of data into target , by applying this on the extracted data. It can be further summarized as: The movement of data contained. Validation that the data is per the quality rules. To modify the structure and content of data. Integrating of the data with and from other sources. Aggregated or derived values are calculated on the basis of processed data.

3.Load: The processings which are required for loading data into target system.[1]

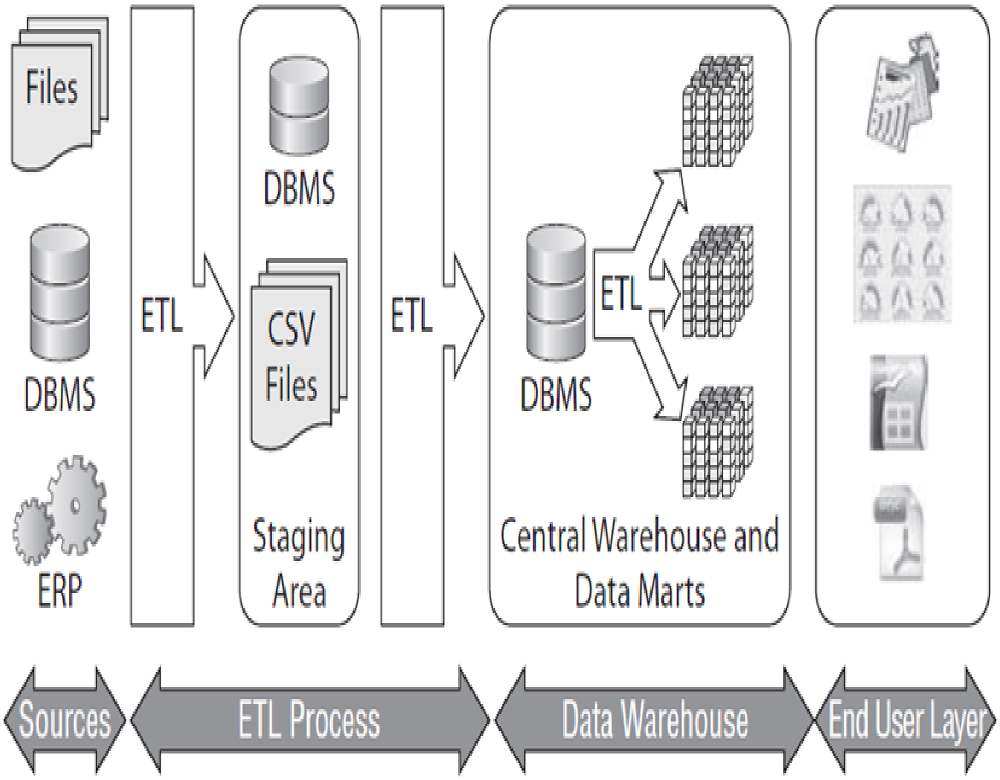

Figure 1.1: Classic Architecture Of ETL

Description

Extraction: First step of ETL process is extracting data from system sources. This can be regarded as a challenging feature of an ETL, because if data is extracted correctly, then all the subsequent processes can be performed accordingly.[2]

Data from different source systems are consolidated at many data warehousing projects. If every system is seen separately then we can say that it uses a different format or a data organization. Relational databases, flat files and XML's are common data source formats.

But it may also include non relational database structures like VSAM or IMS or Indexed Sequential Access Method (ISAM) or getting data from outside sources like screen-scraping or web spidering. When no intermediate storage of data is required then another way of performing ETL is streaming of extracted data from data source and loading to the destination database.

In more generalize way, it can be said that while extracting the data gets converted into single format that is which is necessary for the processing of transformation.

Transform: In the transformation stage functions or some rules are applied to the data that we have got from our source to achieve the data , to load data into target. There are some data for which there is no need of transformations and these are called as direct move or pass through data in technical terms.

Data cleansing which aims on passing only proper data to the target is regarded as an important function of data transformation.Their also comes a challenge in communication with each other which is based on how data are stored.

There are some characters which are available in 1 system but may or may not be present in other systems. These are categorized into special case , that needs to be handled in the right manner or it can cause no of issues related to the data quality. To meet the technical needs of the server and business some of the transformation may be required.

Load: In this phase data is loaded into the target which may be a data warehouse or simple de-limited flat file. Based on the organizations requirements it varies widely. While in some cases data warehouses overwrites with cumulative information , the existing information.

The updation of the extracted data is done on monthly, weekly or daily basis. New data may be added in historical forms from other data warehouse at regular intervals, for example - like hourly. To explain it, we can consider that to maintain last year's sales record we have a data warehouse.

The work of this data warehouse is to overwrite any old data ,i.e a year older with newer data. But entry of data is done in historical manner. A complex system is required to maintain the history and to keep track of all the changes.[3]

1.2 Scope of Work

By developing this, we can make it easier for the user to dump data in database within seconds and also for the end users to review all the data , by seeing their reports easily. So that any enhancement can easily be made, without further digging into depth.

Chapter 2

Literature Survey

In this project , for backend the technology used is Perl and internal ST tool for data integration . And for Frontend BIRT tool has been used in addition to Java.[4]

For the Perl programming language , the DBI is database access module. A consistent database interface is provided through a set of methods, variables and conventions , which is independent of actual database that is being used. DBI takes all SQL commands through Application Programming Interface ,and dispatches them to actual execution's driver. And DBI is also responsible to exchange results between the driver and give it to the calling script.[5]

Sql queries can be generated by using Perl's text-handling capabilities.By using hashes, arrays and automatic memory management , it becomes easy to collect and process returned data. [6] For eg, in Tim Bunce's Perl DBI API, the arguments to the Application Programming Interface can be the text of SQL queries; so it makes it possible of programming at the same time in multiple languages (e.g., for generating a web page using JavaScript, HTML and SQL).[7]

The use of Perl variable interpolation for customizing each Sql queries programmatically and Perl hashes and arrays are present as structures for holding the resultant data set from each queries. This helps majorly in accessing a high amount of data for post-processing.[8] In earlier versions of Perl, interpreter were relinked with a client side database for creation of a database interface. This was regarded as difficult, as it was mostly implemented only on most widely used and most important databases and restricting the perl executable of using one database interface at a time.

An the ST tool used here is basically for data integration. It is like a data integration tool in which data collected from other forms are integrated into one module. Referring to several such tools , this tool has been designed.[9] It is mostly responsible for the Extraction, Transformation and Loading processes.This tool solves various purposes such as: Migration of data between databases and applications. To export data from db to flat files. Massive loading of data into database. Cleansing of data.

Further it has more responsibilities: It is responsible for Transporting data , i.e clustering, merging, duplicating, splitting, partitiioning, joining.

Transforming data , i.e splitting, filtering , databases,files ,selecting, looking up data etc.

Load the data, i.e partitioned load, clustering, datawarehouse population, bulk load etc.[10]

Business Intelligence and Reporting Tool is being used. Report Object Model abbreviated as ROM is created by ERD model as a format and ERE executes it.

A brief intoduction to BIRT:

The elements which are common propeties of all the elements of ROM are called base elements.

The design which consists of report libraries, design report and parameters of report are termed as report design.

The model in which elements and layout rules are present are termed as layout model.

The list and table elements are present for the display of data from a data set

Textual Elements - Elements which include Text,Label,multi line data are termed as textual elements.

Chapter 3

Implementations

The tool we are currently working on is Weblib Row data Population.In this tool the main functionality is about taking data from input files like csv files and do all processing on data and finally generating reports of these data. The flow of project is divided into 2 phases:

3.1 Phase 1:

Firstly from the daily reports and log files that are generated are provided to us and we have to do all the manipulations on them , so that charts are generated. The processing on data is done using internal ST tool and perl languages. And then all data are dumped in database We have 2 kinds of reports as inputs: Daily Reports (Reports which are dumped in the database on daily basis). Log Effort (Reports which are dumped in database , in weekly or monthly basis.) We will have multiple Csv of both daily reports and log efforts. And we have to make this run on a scheduler , so that it keeps running automatically after certain duration of time. We run our tool on a crontab scheduler, by fixing the durations after a certain period of time

There are mainly @ types of reports that are provided to us. They are basically comma separated files. They are as under:

3.1.1 Daily Report :

The reports generated on daily basis. The detailed description of the elements present in it can be said to be a library. That is for a particular library the invocation of reports is being done. Now we can define library as : A library is a consolidated data for use in

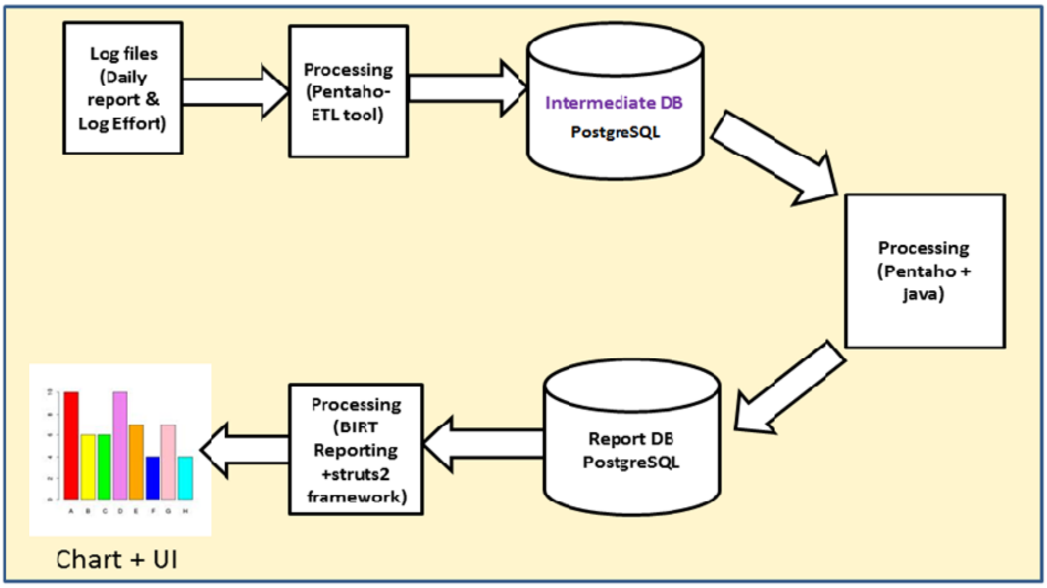
Log files (Daily report & Log Effort)
Processing (Pentaho-ETL tool)
Intermediate DB PostgreSQL
Processing (Pentaho + Java)
Report DB PostgreSQL
Processing (BIRT Reporting +struts2 framework)
Chart + UI

Figure 3.1: Flow Of Project

designing a system on chip. The library is comprised of various views which are useful for designing a chip. Lobrary further contains cells. The definition of cell can be as under: Cell is defined as a component which is performing a basic function. More a digital design concept: Boolean and basic function (AND, OR).

3.1.2 Logeffort :

A logeffort is again divided into 2 parts , based on their functionalities. They are: 1. Logeffort Header 2.Logeffort Footer

Logeffort header consists of following fields such as date , library name, path ,technology etc . Logeffort Footer consists of following fields such as step names

3.1.3 Processing :

Some brief overview about the steps being followed in processing Steps of Processing: The first step is to get all Daily Report CSV names.

And then further manipulation and processing is done on all the data like :

- The last modified date of the file is extracted and saved. And then the comparison of the file with the date is done . If the comparison is true then the files are

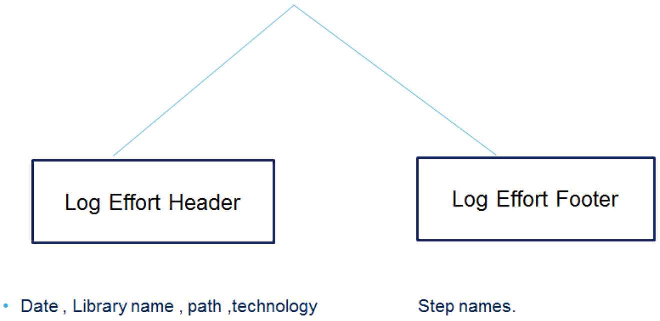

Figure 3.2: LogEffort Footer

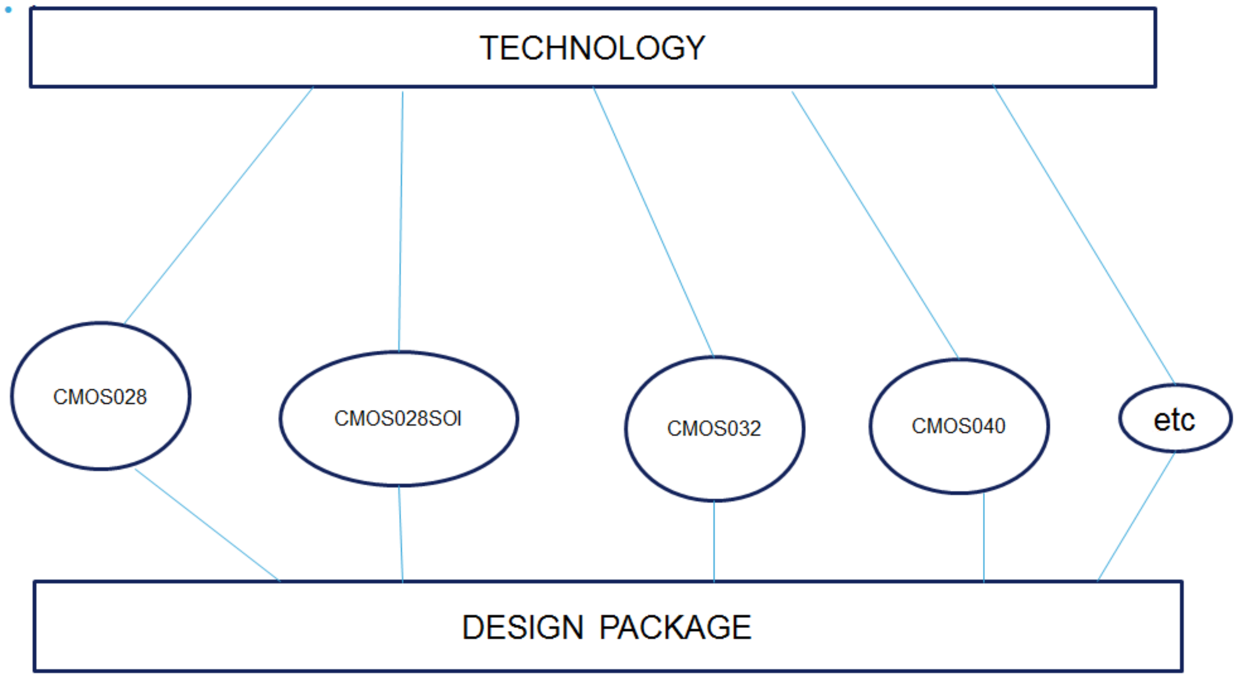

Figure 3.3: Hierachy Of Technologies

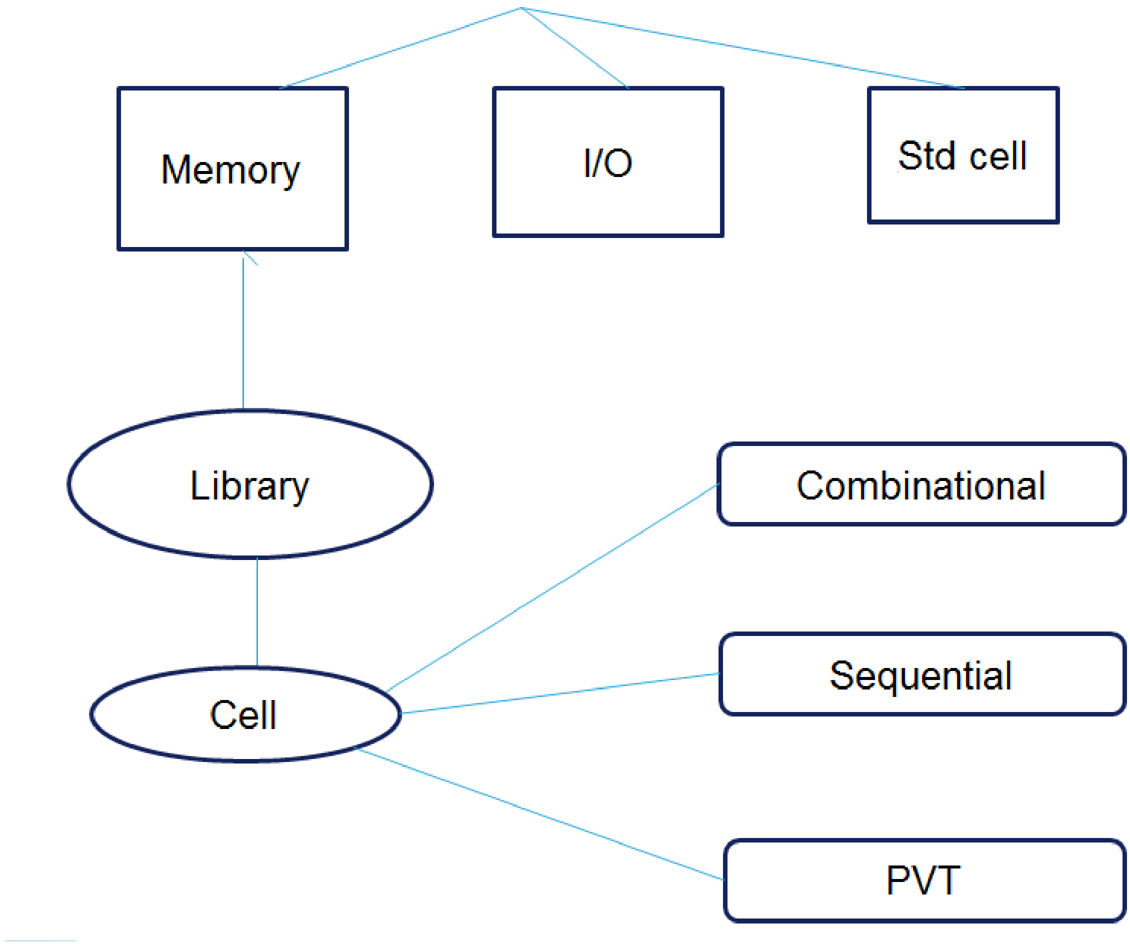

Figure 3.4: Continued

processed.

- A configuration file is made for storing all the directory paths.
- Do lookups in the database for all the technology names , if they exist then proceed further and if not , then send a mail to the administrator of adding the names.
- Do lookups regarding the libraries ,step etc.
- Dump all Daily Report data: All data are dumped in database PostgreSql.
- Get all the Log Effort:
- Similar processing is applied on it also.
- Dumping of Log Effort Data: All data is dumped in database.
- Calculation is performed on the
- Throughput and runtime are calculated based on all the data retrieved.
- Some formulas are applied.

3.2 Phase 2 :

For different System on chip flows , the data are dumped. The flows are mainly divided into 2 parts: One is Front End and the other is backend

3.2.1 Front end flow:

We have libraries , which are basically a set of transistors , or combination of different gates or combination of memory devices, or combination of I/O devices.

So all these are integrated in final chip and these are SOC's. The next step is that they are synthesised which helps us in generating netlist. and then on this kits are used , with the help of which reports are generated. The reports generated can be in the form of csv or can be in any oyther format.

3.2.2 BackEnd flow :

In this just the development of layout views are done . In technical terms, floor planning is done or we can say a blue print of chip is prepared. And after preparing blue print,

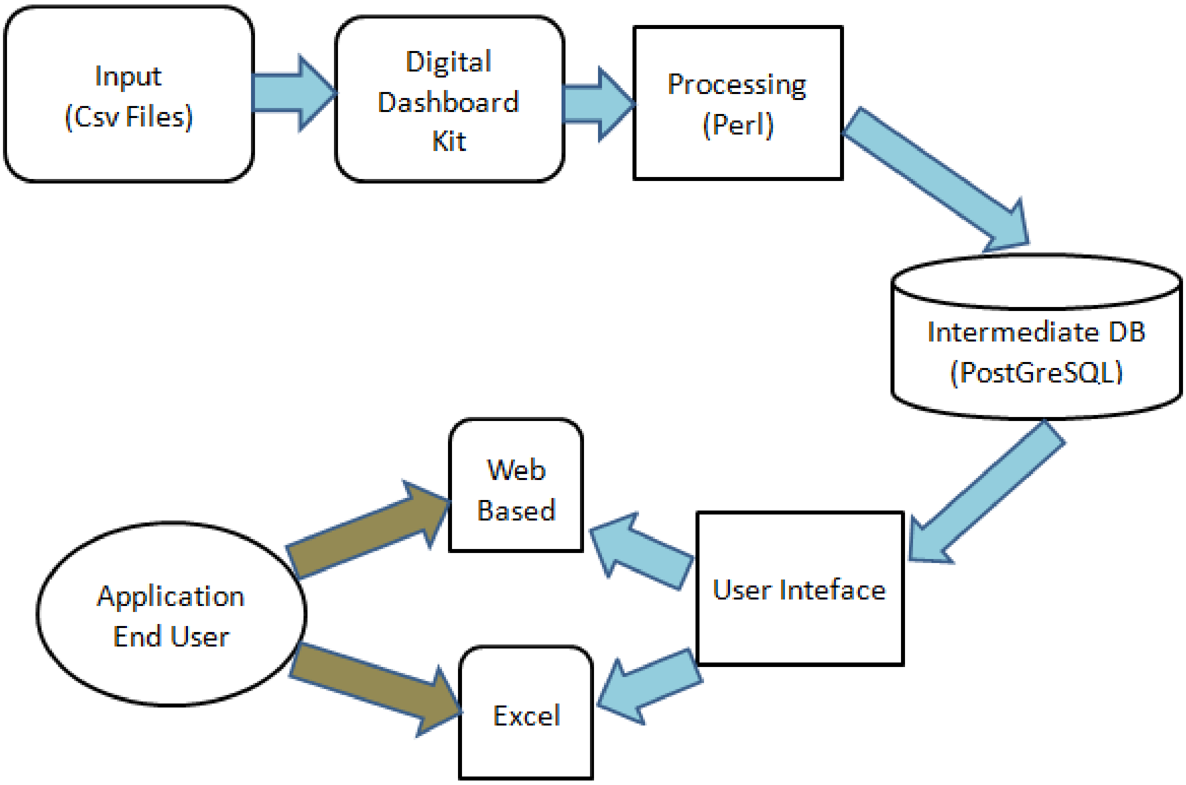

Figure 3.5: Flow Of Project

the placement is finalized. After this step comes the step of routing , which ensures the flow of the kit prepared.

3.2.3 Terminologies:

Libraries:

A library is defined to be a combination of different gates or combination of memory devices, or combination of I/O devices.

SOC:

An IC also known as integrated circuit , is an integration of all the components of an electronic system or a computer into a single chip. All these are integrated then into a final chip and these are termed as SOC's.

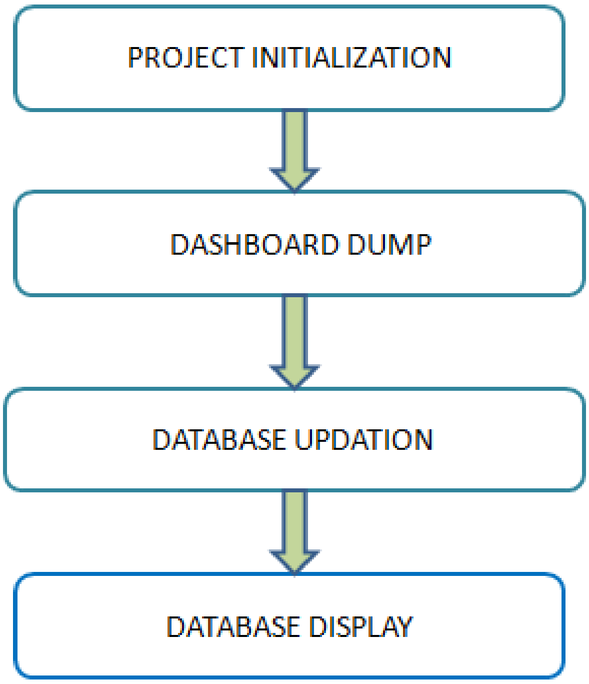

Figure 3.6: Steps Of Execution

Dashboard:

It is defined to be a raw csv dump containing technical key performance indicator (KPI) for Power Performance Area (PPA) Analysis of System On Chip (SOC).

KPI:

A KPI also known as a key performance indicator is used for evaluating factors that are necessary for the growth of the organization.

Dictionary:

It is defined to be a collection of definition of KPIs.

3.2.4 Steps Of Execution:

- Project Initialization: All data corresponding to a particular project is dumped at a time, such as project information flows required users of the project roles of different users etc.

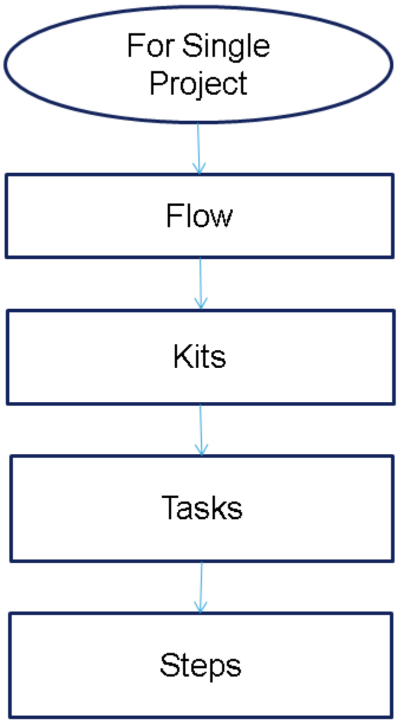

Figure 3.7: Flow of Project

- Dashboard Dump: The dashboard csvs are dumped in the database with respect to the uniqueness of certain parameters.

- Database Updation: Addition deletion of data in the tables initialized from project initialization is done , such as dictionary, ows , kits ,tasks, steps etc. The template of all the les are also provided through the command line arguments. The users can also be added or deleted.

- Data Display: The data display of any desired table can also be seen on the terminal , such as dashboard,users,roles, blocks, ows,kpis etc. For all these command line arguments are made , and using them data are dumped in database within fraction of seconds. SQL queries are also highly optimized.

3.2.5 Data Dumping :

So keeping this all in mind, we develop an efficient database , so that processing of data becomes simpler.

In database , we have for a particular flow , we have multiple kits , and within it we have multiple tasks, and inside it we have multiple steps. So this becomes the hierarchy.

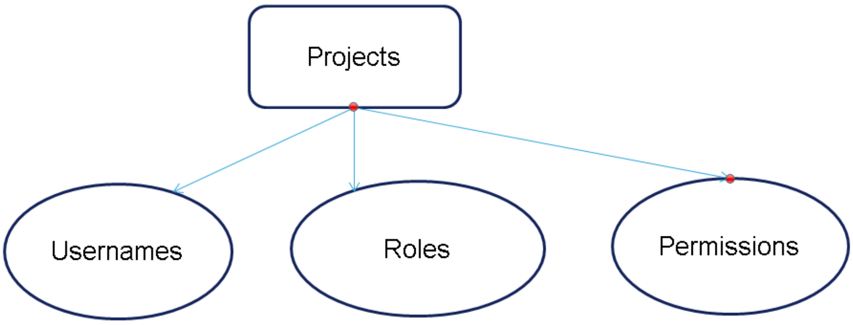

Figure 3.8: Continued

Then accordingly multiple projects are defined for which this will be used. Their usernames are given and their roles are aslo defined. Plus the permissions which they have.

And then at the running stage, the complete run information is tracked , and the refined data are dumped in a particular table.

Chapter 4

Tool For Verification Of IP

4.1 The Need For Tool:

With the expanding utilization of IP in the business of semiconductor, the interest to confirm IP for quality is high.So for this reason an instrument was outlined which can approve IP. To implement the tool ,the coding has been done in tcl. Better concurrency is allowed by using spiral development model, which provides convergence which is faster.

The same thinking holds for IP verication in light of the fact that the affirmation group sits tight for the outline group to finish the IP acceptance before starting to ensure it.

An IP inside STMicroelectronics is organized with a list document (vc.bbview record), which characterizes the arrangement of the IP deliverables.

Various ways exist to confirm the structure of a deliverable (i.e., a standard perspective), or consistency.

4.1.1 Inputs:

The IP acceptance or certificate begins with a coupling, which is an arrangement of all the data bits of data together. While perusing in every information includes syntactic checks,the binding step performs semantic checks to check the consistency of the inputs.

4.1.2 Syntactic and consistency checks:

1. The tool gives both a graphical client interface and a batch mode. The GUI centers of the ease of use of the tool, that is a simple to-utilize interface and the likelihood of propelling checks independently and simultaneously. While the cluster mode

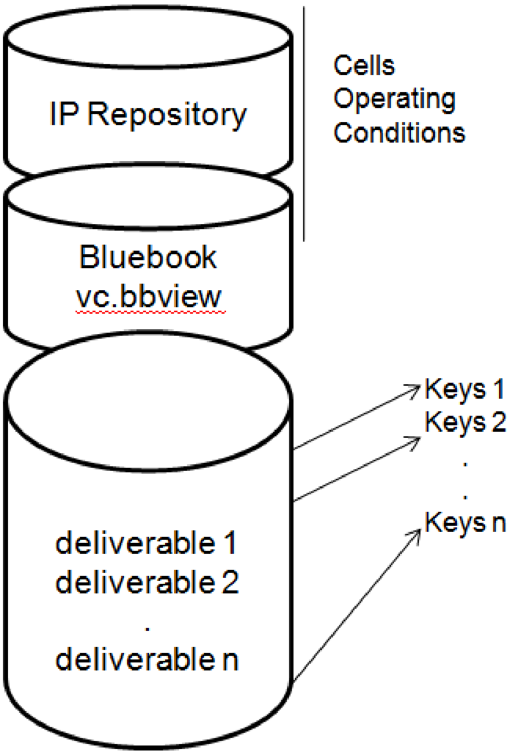

Figure 4.1: Structure of IP Repository

permits the certificate to be run with no connection but with programmed report era when all the checks complete .

2. It is advantageous to note that the decision of the tool may prompt diverse certificate status: a few tool print note, while others print cautioning, and others raise a blunder! As a result, the report needs to be translated by the certificate group.

3. Whichever check we want to choose can be run as stand alone. The activities can be said to be dependent among themselves, so we may face difficulty in running them all at one time.

4. Activities have been created for consistency and syntactic checks to facilitate the use of doing the job. So by doing this , nothing has to be done by themselves, just have to wait till the checks get completed and the dependencies of inter check gets resolved automatically.

4.1.3 Outputs:

The status of the IP affirmation depends on the execution of the obliged arrangement ofcommercial tools.

The tool gives two sorts of yield :

1. Log files which are created provides us with the message which the ipscreen gives as output plus the complete expected result of execution of tool. It facilitates the designer so that it can check the executions of check in detail.
2. One summary file is also present which gives the output by comparing the actual results and the expected results. So by this, the failure or success of check can be validated.
3. The content of the summary file is shorter and helps in providing steps which are important in the execution of checks. SO it helps in the increasing the knowledge of user about the IP issues that need to be diagonized. SO that it becomes much simpler for the designer to get the deliverables fixed..
4. Both these files play an important role , for example: If we have already seen the summary report then it becomes easier for the designer to analyze the log files afterwards. And according to his analysis he can decide to change status or if he wants to add some more information before launching the checks again.
5. A GUI option is available so that the layout of the module gets updated with the fixed colors. They are : If the result is success , then color is green If the result is failure , then color is red If the result is with warnings, then color is orange If the result is in progress, then color is yellow If the result is in the pipe, then color is pink If the result is idle , then color is grey If the result is missing , then color is white.
6. Using both the shell option and GUI , HTML report is being made when the session ends. A tabular form is displayed with the status certification and gives the log,scripts and the summary files which where used for performing verification.
7. The report can also be sent to the design team . And it may also be a part of catalogue IP documentation.

4.1.4 System GUI

The GUI displays three main parts:

- Firstly, there is a menu bar which allows us to edit, read, close ,modify the inputs certification, for performing certification, for performing setups, and for performing and parameterizing the binding.

- In the setup, we have availability of tool checking , reading of command files and parameter, check of violation of time , and configure display, color and layout, etc.

- Secondly,there is a layout which displays the module currently under the certification.Once found the flagging of it is done with the colors which are mentioned above. Designer has an option of clicking on any button which is idle, which is corresponding to a consistency or syntactic check.

- In each check there are mainly three buttons available: For installing the check directory: setup button For checking button of executing a tool: check button For extracting summary and performing diagnosis : res button.

- Third, there appears a log window whose work is of printing all the warnings, errors and notes from the commands which designer runs. The command history as well as the log file, are printed in separate files and can be used whenever required.

- Plus the tool also has an online document which can also be invoked from the GUI's main window.

  Documentation gives a deep knowledge of how to use the tool, description of tools EDA and also the grammar used by designers.

  Designer has the choice to move the documentation by hyperlinks.

4.1.5 Qualitative and quantitative results:

There have been recommendations which were given below, on the basis of soft IP recently certified:

Structure of IP repository

1. The user has to separate directories for cell which are different and stand alone , similarly for activities and operating conditions.

2. It is also useful for copy of repository and packaging. But we have to keep in check that in terms of deliverables, the repository is stand alone.

3. It is also useful for consistency check and packaging.The task is also of moving output from design validation that are previous. (netlist , reports).

4. It is mainly useful for diagnosis,copy of repository. We have to make sure that the scripts are configurable at phase of installation in respect to environment variables , paths to access, libraries or are stand alone.

Completeness of the IP deliverables:

1. File extension used should be significant.

2. It is also useful for syntactic checks,packaging. We have to make sure that the scripts are configurable at phase of installation in respect to environment variables , paths to access, libraries or are stand alone.

3. The scripts which are technology independent are used. And this is also useful for consistency checks.

4. The scripts should be self checking enabled and avoidance of personal reports or redirection of hidden output should be there.

5. It is also useful in diagnosis automatically.

Chapter 5

Plugin Structure:

A plugin (named ¡plugin¿) is a standard deliverable made of:

5.1 Mandatory directories

1. plugin.info : Unicad short description of the product. ¡plugin¿.ptbl : Unicad product table.

2. plugin.csh : script to set up the plugin from ipscreen framework. This file is sourced (in the task envirement) before running each plugin task.

3. doc: Directory containing the documentation of the plugin. Its structure is Unicad compliant.

5.2 Other documents can be or not provided with the product:

1. Kps: is for Known Problems and Solutions.

2. Onlinedoc: is used by the Framework to access the documentation on line thanks to the file

3. Index.html. - Contains the changes with respect to previous versions.

4. Spec: is for Specification. Contains the specifications of the plugin.

5. UM: is for User Manual.

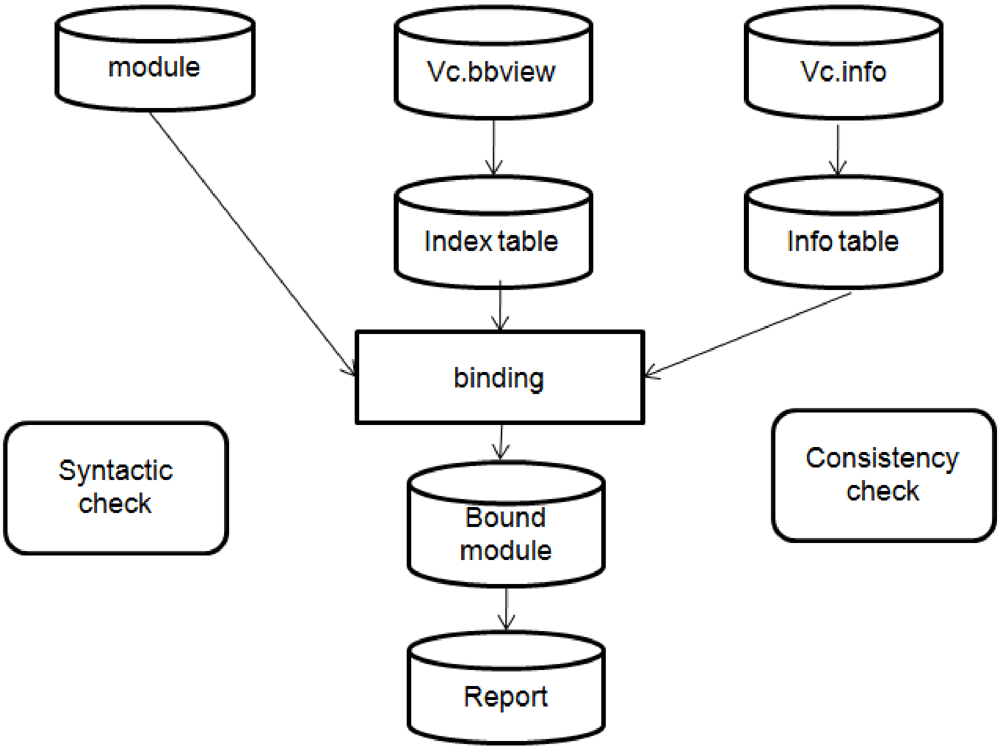

Figure 5.1: Validation an Certification Flow

6. DB: Directory containing plugin-specific database tables.
7. bin, lib: directories which contains all necessary binaries and libraries for the plugin (OS independent) .
8. arch: directory which contains all necessary binaries libraries for the plug-in (OS dependent).
9. misc: directory which contains any files that doesnt belong to previous categories.

Scripts

- Dedicated scripts (as listed in tasks table) must be located in directory scripts.
- Scripts can be written using any shell, tcl or other languages.
- The imperative is that the script is executable.

Documentation

Any plugin documentation must be stored in doc directory. If there is specific template to create task report in ipscreen, TaskReport.xsl file can be added in doc directory of correspondent plugin. If this file is missing, ipscreen uses the default template found in IPSCREENROOT/XSL directory. Any path referring to a documentation file must be relative path to doc directory.

5.3 Plugin Database

1. Table format Most of the information handled by ipscreen is stored using a unique format. This format is a text file, called table. Information is stored by line: First line: Descriptor names Next lines: data values

2. Database content A plugin must be delivered with mandatory description tables in its database.

3. Setup: overall attributes of the plugin.

4. Tasks: available tasks.

5. Id mapping: bbview keys versus ipscreen internal identifiers mapping.

6. Tools: necessary tools required by the plugin.

7. Categories: defines groups of tasks, with order, to be viewed in the basic window of the GUI.

5.3.1 Optional Tables:

- Sequences: defines sequences of tasks, to be viewed in sequence window of the GUI.

- Collection: defines collection of tasks, to be viewed in collection window of the GUI.

- Custreport : This table is useful for plugin developers who want to customize the plugin report generated by ipscreen. This table declares all custom reports (normally generated by the plugin) to be inserted in the plugin report.

- NeededFilesForOwnerDebug: This table is useful for plugin developers in order to specify needed files to be joined to mail sent while encountering a bug. (notify plugin owner command in tasks contextual menu).

5.4 Plugin API:

This paragraph intends to describe the interface with the framework. It deals with the mechanisms of input/output, the report and the task status. Each task script will manage these elements.

5.4.1 Data flow :

- Most of the tasks need input(s) to work and create output(s), either to be then used as input(s) by other tasks, or just as a final result.

- The inputs can be of 2 types: bbview or inout.

- The outputs can only have one type that is inout.

- The bbview type corresponds to the data coming from the IP under test or the auxiliary library. It is taken from the vc.bbview file. So, it is some files that are not created by a task.

- The inout type corresponds to all the data created by the tasks or eventually given by the user.

  The idea is now to be able to share the inputs and outputs with all the tasks of the same plugin and, in some conditions to other plugins. To do that a table is used: data store.

Chapter 6

Plugin Check:

This plugin checks the conformance of libraries vs Library Specification,in term of mandatory views are present, and aligned with library structure.

6.1 Check consists of:

1. Setup Plugin: To setup the plugin, some inputs are required. If user doesnt provide inputs through the configuration, then all inputs will be picked from SPEC.

2. Task Setup : This task initializes the plugin db with the corresponding ipstyle for each library under test. This task checks the consistence of the given IPstyle.

3. Bbview Checks : This collection includes three tasks :

4. BBview sections are correct : Checks that bbview is syntactically correct. It checks the header section, cell section and mapping section.

5. Data indexation and coherence : Checks that data are correctly indexed or not. Also checks for the symbolic link, locked and hidden files. It checks that Library should be aligned with structure; if any other directory or file is present in that package then it will report for the same.

6. BBview corporate checks: It checks that bbview keys are corporate or not. For this it checks with id mapping file which present in the plugin. All the keys should be a part of id mapping file. Id mapping file contains all the keys which are available till yet.

   Comparison checks for bbview :

7. BBview vs Library Structure: In this task it compares the bbview with library structure (given by user). BBview keys should be aligned with library structure. If any mismatch occurs, it will report an error message.

8. BBview vs Addendum: In this task it compares the bbview with addendum. If any key which is mandatory for an IP (recommended by Addendum) is not a part of bbview , it will report an error message.

   Comparison checks wrt to Spec:

9. Validation Of LRM Info: The purpose of this check is to validate the information related to the head section of vc.bbview mostly in context to the technology used.lrm info.csv is used as an input from Spec.

10. Validation Of Versions of Format: The purpose of this check is to validate the versions of formats of some files in Package.format.csv is used as an input from Spec.

11. Validation Of Tools: The purpose of this check is to validate the information related to the versions of tools used in validation of an IP Package.tools.csv is used as an input from Spec.

Report : A report is available for each task containing a brief summary reporting all done checks, warnings and errors.

Batch Mode : The plugin can be used in batch mode by writing a command file.

Chapter 7

Conclusion

The conclusion here is that the tool is very useful both for the developer and user. As Perl language is being used for data dumping and manipulation , so it becomes very fast. and the data is getting dumped within seconds. So this can be counted as an advantage if compared to other tools. And in the reporting part ST tool is used , which is also very efficient. And it becomes very feasible for the developer to do all the processing and for the user to track all the changes.

A plan to make the validation tool more robust is being carried on. A thorough regression testing is to be designed which will further check the consistency of the tool.

The reporting mechanism of the tool has to be improved for better results.

Depending on the problems faced by user, the issues arise and accordingly the problem is resolved.

Currently , the tool is internally used by many teams in ST. And the future scope is that , with the constant review from our clients, we can make it more better. And make it more robust , to be used anywhere and by everyone.

Our aim is of building a subsystem which performs the task of data integration i.e. a combination of data from different sources and to give user a output of all these data with the help of report generation.

More enhancement will be done in the future , based on the increasing user requirements.

The validation tool is a great help to easily validate IP's.

In future , more enhancement is to be done on it , based on demand.

References

[1] X. B. Li Jian, “Etl tool research and implementation based on drilling data warehouse,” IEEE.

[2] S. Thakkar, “A data integration approach to dynamically fusing geospatial sources,”

[3] R. Radhakrishna, SravanKiran, “Automating etl process with scripting technology,” IEEE.

[4] “Database technology research group department of informatics,” University of Zurich.

[5] B. Jones, “The semantic model editor: Ecient data modeling and integration based on owl ontologies,”

[6] M. Dominus, “Perl: not just for web programming,” IEEE.

[7] D. Weild, “An adaptive query execution system for data integration,” University of Washington.

[8] W. Moise, D.L, “Extracting facts from perl code,” IEEE.

[9] R. Tian Mi, Aseltine, “Data integration on multiple data sets,” IEEE.

[10] Z. T. Jianhua Wang, Yuefan Liu, “Research on data integration of bioinformatics database based on web services,” IEEE.