Automation and Optimization in Layout Generator

Submitted By

Niravkumar Ashokkumar sharma

14MCEN22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Automation and Optimization in Layout Generator

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Niravkumar Ashokkumar Sharma

(14MCEN22)

Guided By

Prof. Usha Patel

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Certificate

This is to certify that the major project entitled **"Automation and Optimization in Layout Generator"** submitted by **Niravkumar Ashokumar Sharma (Roll No: 14MCEN22)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering(Networking Technologies) 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.

Mr. Ashu Talwar
Section Manager,
TRnD Department,
ST Microelectronics,Greater Noida

Mr. Rohit Pandey
S/w Application Developer,
TRnD Department,
ST Microelectronics,Greater Noida

Prof. Gaurang Raval
Associate Professor,
Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. Usha Patel
Assistant Professor,
Computer Science Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof.P. N. Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Niravkumar Ashokkumar sharma**, Roll. No. **14MCEN22**, give undertaking that the Major Project entitled ”**Automation and Optimization in Layout Generator**” submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering(Networking Technologies)** 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 gives me tremendous joy in communicating thanks and significant appreciation to my guide **Prof. Usha Patel**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for her significant direction and nonstop support all through this work. The gratefulness and nonstop bolster she has granted has been an extraordinary inspiration to me in coming to a higher objective. Her direction has activated and sustained my scholarly development that I will profit by, for quite a while to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Prof.P. N. Tekwani**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

My Sincere as a result of **Mr.Ashu Talwar** Manager, ST Microelectroniocs, Greater Noida. I exploiting his vast information and great future vision and owe his heaps of appreciation for essentially affecting this report.

I would like to thank my Mentor, **Mr. Rohit Pandey** , ST Microelectroniocs, Greater Noida for her valuable guidance. Throughout the training, he has given me valuable advice on project work. He has given me all kind of support to handle complex situation.

I would like to thank you to my Institution, All faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Niravkumar Ashokkumar Sharma
14MCEN22

Abstract

VLSI technology is used for generation of memory. It has requirement of three crucial parameter : time, cost, effort. So our motive is to optimize this parameter. If we reduce the time causes reduction in effort and also vary the cost.Here we for different technology along with configuration parameter, we develop the compiler to generate the views.If any single configuration parameter change then we need to start from the beginning although belong to same technology. So to reduce the time and effort we make a framework called unibe.It provides the common platform for all technology along with different combination of parameter.We pass configuration parameter as input file to framework and it will generate the views.Before passing input file we are validating it through checker. Moreover we are also trying to optimize the input file so most of the task can be shifted before it passes to unibe. As a result in reduction of time and saves the effort which reduces to execution time as well as validation time of view of the memory.It will take 6 to 8 month to develop the cut and execution time will take 15 to 20 mins.

Abbreviations

CDL	Circuit Design Language.
GDS	Graphical Design Stream.
LVS	Layout Vs Schematic.
DRC	Design Rule Check

–

Contents

List of Figures

Chapter 1

Introduction

1.1 General

Our Information society at the turn of the 21st century bears an immense interest for microelectronic gadgets with definitely improved execution, keeping in mind the end goal to satisfy the general public future necessities for data preparing, exchange and capacity of devices. A standout amongst the most progressively advancing divisions of microelectronics is right now the memory market, which is seen by the buyer in the pervasive memory sticks, memory cards for advanced cameras, and so forth. the memory can be divided int two parts which is volatile and non-volatile memory

As the name suggest, Microelectronics is a subfield of electronics. Microelectronics relates to the study and manufacture (or microfabrication) of very small electronic designs and components. Usually, but not always, this means micrometre-scale or smaller. These devices are typically made from semiconductor materials. Many components of normal electronic design are available in a microelectronic equivalent. These include transistors, capacitors, inductors, resistors, diodes and (naturally) insulators and conductors can all be found in microelectronic devices

The quality of achievement in microelectronics business is basically depends on execution time. It is not simply execute the thing as looked for. It should work for extended period of time ceaselessly. Even the things also depend on design time of circuit.depending of both : Quality time and Design time the validation time is also calculated.

1.2 Objective of Study

As we know that VLSI technology is used to develop memory and it is very crucial. The fail of fabrication of memory causes major loss. So it is require passing through some simulation process which develop the cell of memory and pass through different cases to make robust. The process through which memory cut is developed called compiler. The compiler works on different technologies and different types of memory. The process of developing the memory is called design time of chip. To analyze the memory we need the different view which makes the memory robust and fail-safe which is called the validation phase of memory. To make process reliable and more faster we need to automate and optimize the thing.

1.3 Motivation

A user will provide the specification and based on that we develop the memory. Again for different memory needs, We need to change the compiler configuration which is taking more time. We need something generalized structure for all memory architecture. For that we need to concentrate on layout part of memory to make the process fast and reliable. So our aim is to optimize the code and automate the thing. The input should be simpler so that customer can develop the memory cut and if there is any issue found ,can be solved by the expert person. Thus reduces the load at the expert and provide satisfactory solution to customer.

1.4 Scope of Work

The guideline method of reasoning of the assignment is to give an interface between others instrument which makes the memory cuts and diminish the compiler advancement time. Unibe thing supports all designing. That suggests Unibe will have the ability to make view for every technology, no need to use different compiler to create view.

To automate the process we write one text file or provide GUI interface which has all data about to generate the view of memory. The layout structure will parse the file and generate the product. So our task is to generate the different cut and validate it.Thus it will saves the design time. We just need to validate the cut and found any issue then fix it.

1.5 Activities

- Understanding various processes involved in the Circuit Design flow.
- Understanding Technology that can be useful in project infrastructure.
- Learning the Reflection API, understanding the usage of Reflection API and analyse Java pattern.
- Learning shell script to validate the product.
- Implement Text Parser
- Implement checker

Chapter 2

Literature Survey

2.1 Architecture of Memory

Memory architecture explains the way of implementing electronic computer data storage in a proper manner that is a combination of the most reliable, fastest, most durable and least expensive way to retrieve and store information. Based on the specific application, There are two types of basic architecture introduced :

- Basic architecture
- Advance architecture

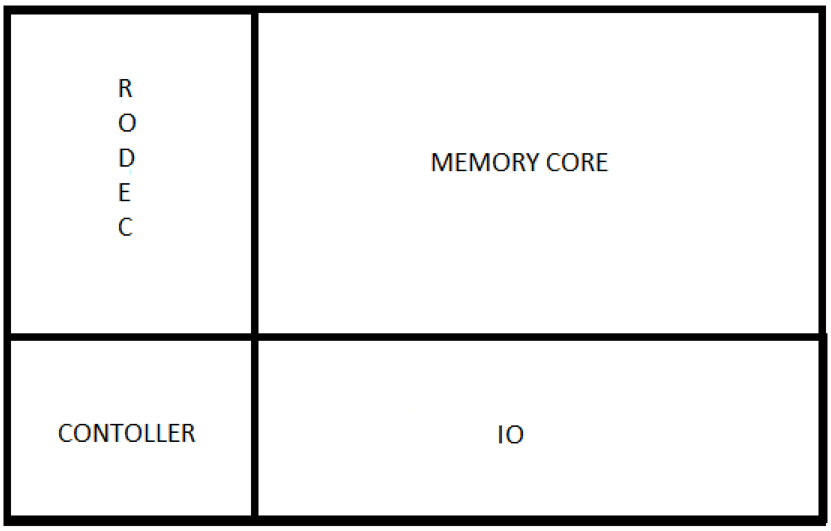


Figure 2.1: Basic Architecture
[1]

2.1.1 Basic architecture

Every one of the cells joined with the word line in the line is active, when they chose lines are high in this manner dispersal increment. In the crucial fundamental building, line access time is the two main considerations add to the read access. When the size of the Memory expands the heap is lessened in light of the fact that the quantity of cells joined with the word line increments. Consequently, The word line deferral expands due to the increment in the word line capacitance. By lessening the bit line capacitance and the word line capacitance are the two elements that can be enhanced, however this is accomplished strictly when utilizing an alternate architecture and the other architecture is spilt core architecture and bank architecture as shown in figure 2.2 and 2.3. [1]

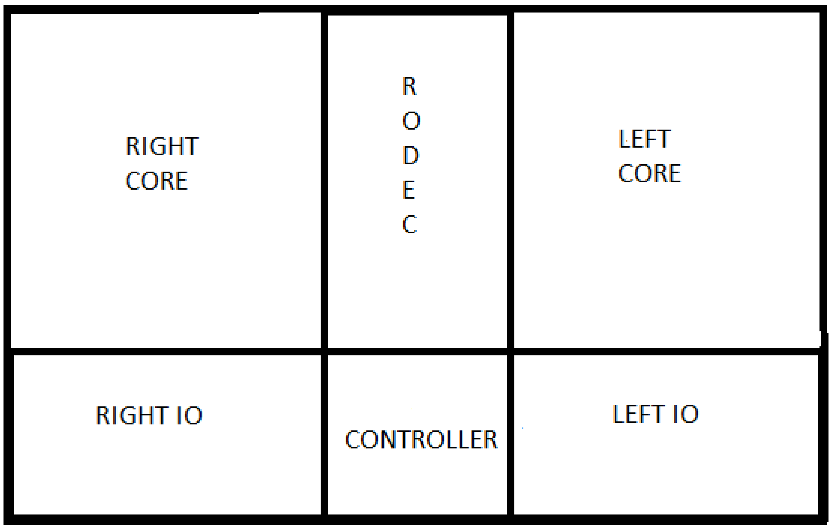


Figure 2.2: Spilt Architecture
[1]

2.1.2 Advance architecture

The advance architecture is spilt core and bank architecture. The basic problem in previous architecture is read and write the data to memory cell. It takes less time in cell which is near to rowdec while take higher time which is far from rowdec. Again there is drop of voltage while writing or reading data to memory cell far to rowdec. To overcome this issue we need the advance architecture which resolve this matter and make the memory architecture more efficient.

In spilt core architecture the row-decore is shifted to middle of memory core to reduce the time of retrieving and but still require the time to store data. The problem is

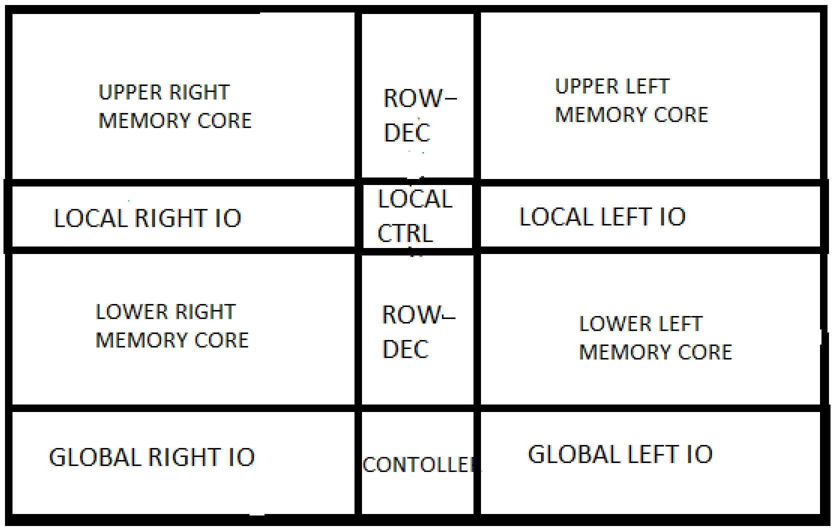


Figure 2.3: Bank Architecture
[1]

overcome by the bank architecture. In this architecture the IO is shifted to memory core horizontally. So it has the two IO systems: Local IO and Global IO. There is also more flavors available based on environmental condition. [1]

2.2 Important Terminology

Memory is constantly given as far as words*bits. In any case, when we create we change over it into rows and column.

Rows= words/mux; cols=bits*mux;

2.2.1 Importance of MUX

Presently assume the structure of memory is not suitable for SoC. We can diminish the structure of memory and expand the width by keeping up the same size as far as words*bits. Memory =rows*cols = ((words/mux)*(bits*mux)) = words*bits. All in spite of the fact that we have changed over words*bits into rows*col still we are looking after same memory size. With mux=2 Rows = 512/2 = 256 Cols =16*2 =32. So now width is twice and height is half. Now in the event that we utilize mux= 4 height will be diminished to 1/4 and widht will be incresed by variable of 4. Along these lines we can change the perspective proportion with the utilization of mux.[2]

2.2.2 Importance of Row-decoder

Rowdec is utilized for selecting specific row. Presently single rowdec can select from predefined set of rows for example: 4 in this way rowdec can choose 4 rows only.

The arrangement of columns from which a rowdec can choose a line is known as rowdecunit. In this way we required numerous rowdec to get to all columns of memory. Why we can't choose all rows with single rowdec? Since the quantity of columns are not fixed it relies on upon words and mux so on the off chance that we make single rowdec it won't work when configuration changes. In this way number of rowdec is computed on premise of columns.

NoRowdec= (lines/rowdecunit) + (columns % rowdecunit==0?0:1); [3]

2.2.3 Importance of IO Cells

Io cell relies on upon number of bits. Number of IO's will be put as commonly as bits. In mono structural planning bits are even or odd. It doesn't make a difference yet if there should be an occurrence of split and bank architecture it relies on upon rationale on which side we need to put one additional IO. As here our lief and right part won't be identical.[2]

2.2.4 Importance of Wrapper Cells

As the name suggest, It is wrapper to the cell which is used to provide additional functionality to particular cell.

2.2.5 Importance of Environment Cells

Environment cells are utilized in between core and IO cells. There are two sorts of Environment cells. Introductory one is Vertical Environment strap which is used in core and IO while other is Horizontal Environment strap which are used in core and Rowdec.

Environment Cells are used for to absorb electromagnetic impressiveness made by the group of cells. So after each predefined number of cells strap cells are used so that effect of electromagnetic is diminished which does not affect.[3]

2.3 Process Designing Memory

The basic stage in IC organizing is the making of the design utilizing software called virtuoso. This configuration is checked using DRC to affirm that the outline does not release any standards chose in the DRM which is required for the best working of the chip. By then once the design is DRC clean it is separated and the flavor model of plan to guarantee that the course of action addresses the same circuit as it is in the flair model. following are data about the checks :

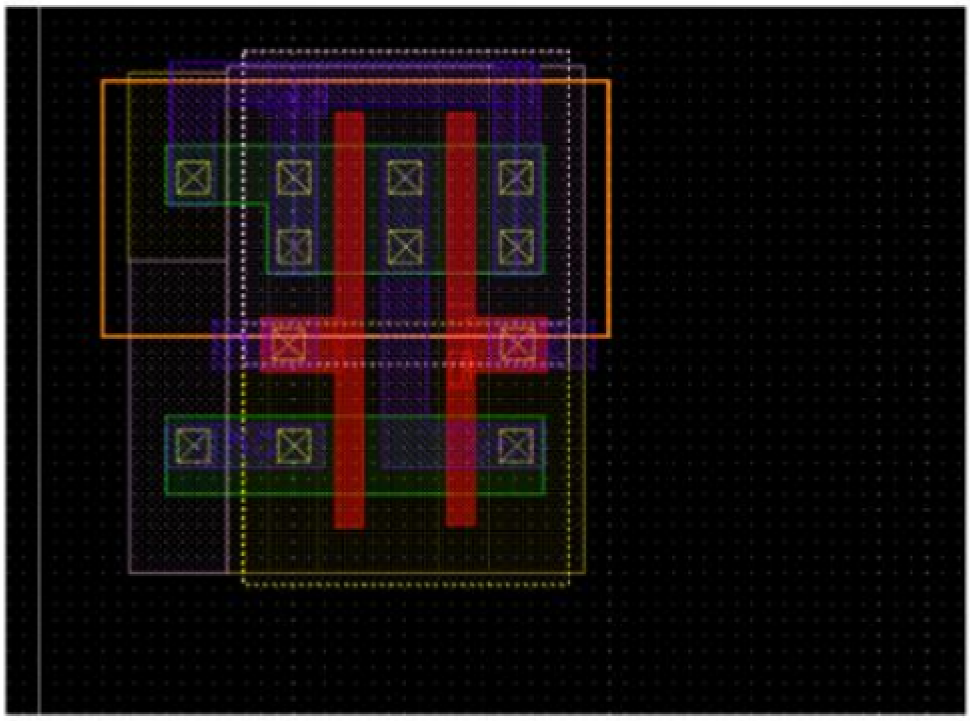


Figure 2.4: Memory Layout

- A diagram block set focuses certain geometric and structure impediments to ensure attractive edges to record for variability in semiconductor passing on star cesses, with a specific choosing target to ensure that a liberal bit of the parts work precisely. While diagram statute checks don't reinforce that the organization will work precisely, They are made to watch that the structure meets the framework obligations regarding a given layout sort likewise handle building. DRC programming by and large takes as enter a course of action in the GDSII standard connection, and produces a report of plan decide encroachment that the maker may change. Purposely "extending" or waiving certain blueprint benchmarks is routinely used to widen execution and bit thickness to the disservice of yields.

- LVS check whether the outline and the schematic are undefined or not. Affiliation showed in both outline and schematic must be same. Inputs for doing LVS may be GDS file or CDL file. The GDS file contains the configuration information for circuit. CDL file contains the schematic information required for circuit. Stream of LVS is showed up in figure.

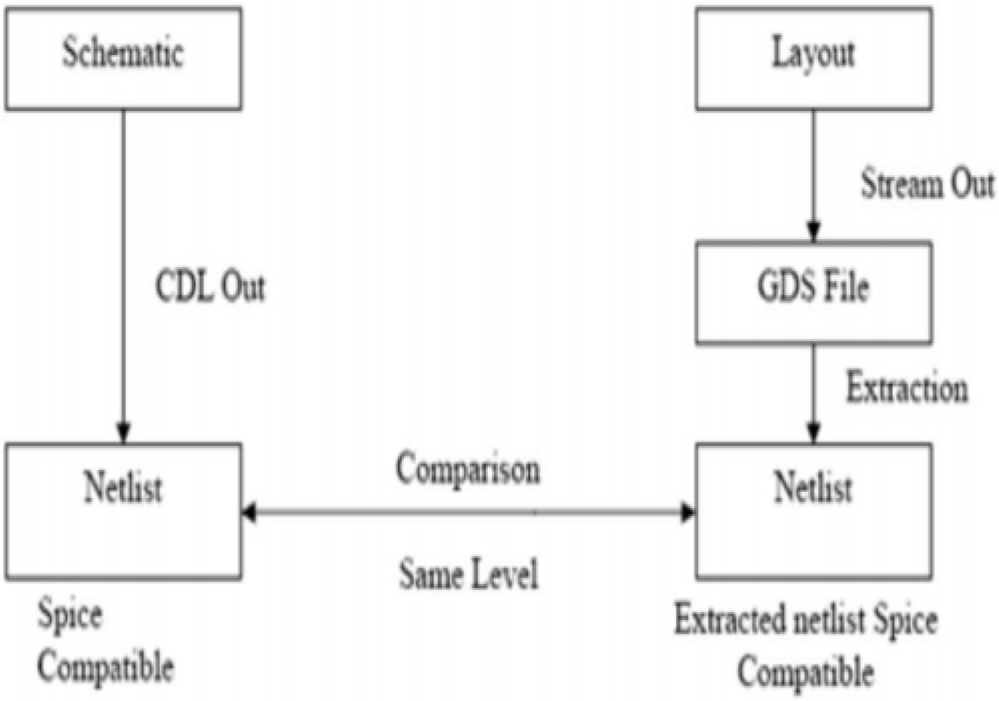


Figure 2.5: flow of LVS

Chapter 3

Project Overview

Essential use of BE Compiler is to generate Layout and netList as per data specification. For Different technology particular BE compiler is come into picture. We need each technology to go ahead single platform.As per the examination, A lot of commonly piece of code used in developing the memory cut for different technology. Following are the motivation behind UNIBE:

- Give single interface to client inputs information.
- Memory designer just fill the template according to as their necessities.
- Support different technology, architecture.

3.1 GDS(Layout) & CDL(NetList)

A standard cell is a mix of transistor and it interface inward structures. The basic cells includes NAND, NOR, and XOR Boolean limit. Using this basic cells the memory cells are developed and group of this cells are called the **BELib** which is the most precious input to develop the SoC. Below Figure shows Schematic view of circuit. The outline of a standard cell is begin at the transistor level, portrayed in the type of a transistor netlist or schematic perspective. The netlist is a nodal depiction of transistors and their interconnections to each other.

Memory Designers proclaims the info parameter to simulate the electronic conduct of the netlist and after that computing the circuit's time domain response. The reproductions checks whether the netlist executes the desired result and foresee others parameters, for example, power or utilization. Since the perspectives are useful for abstract simulation. Layout view is the least level of configuration deliberation. Design view portray the complete data of interconnection between the terminal components.

As the name suggest, GDS is the graphical view of memory which elaborate the number of layers in SoC, How is the connection established and what is the area covered by all the components of memory, What is the size of signal pins how the power is driven from in circuitry.

while designing the SoC, It is mandatory to take a note for DRC value which rule check. As the current flows in the circuitry, Its create the magnetic field. So we need to minimize it. Moreover it is also happened that there might be voltage drop between to cells while we to fetch or retrieve the data.[2]

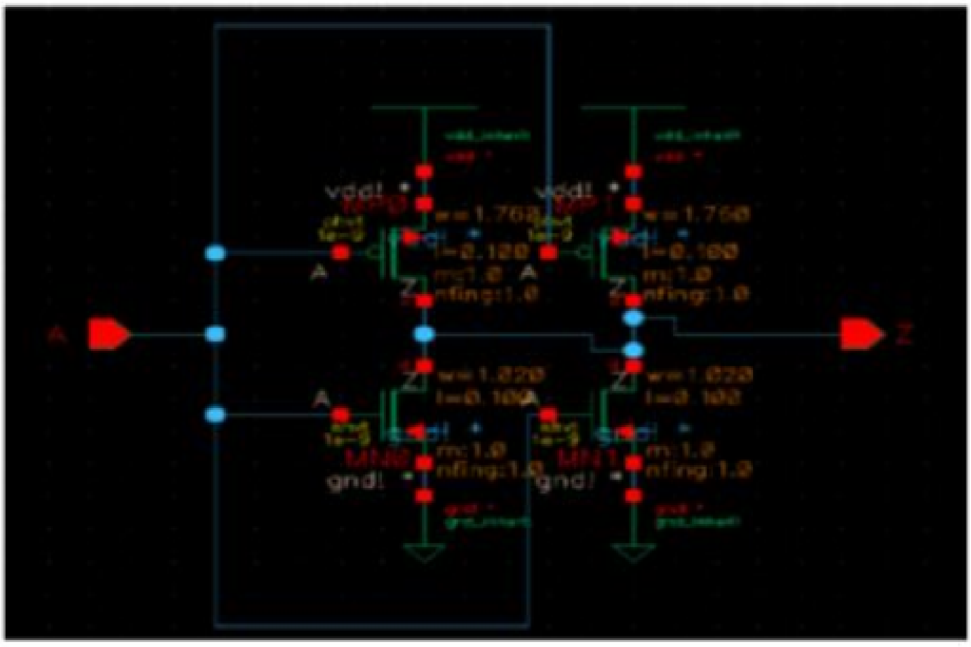


Figure 3.1: Schematic view of circuit

3.2 UNIBE Process Flow

UNIBE process flow is as shown in below figure:

- In the first place Memory Designer requesting to create the memory cut.
- Web gen send demand with information to the unibe.
- Unibe Module working flow

  - Take command
  - View Call
  - Unibe solicitation to make object of top structure from the parser.
  - Parser return object of structure
  - The generated object is converted into the specific object using .jar file.

- Generate views
- (GDS or CDL)View is returned

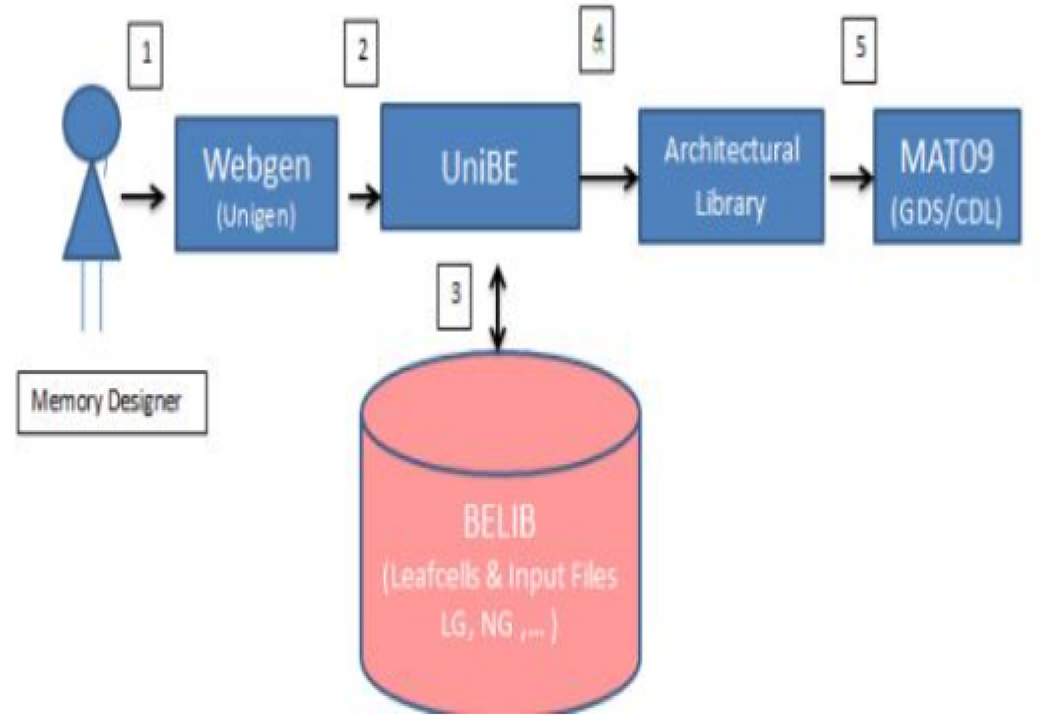


Figure 3.2: Process Flow
[1]

3.3 Object Generation stream for Architectural Library

- Demand for building a Structure.
- Demand for building Alias by Structure is sent.

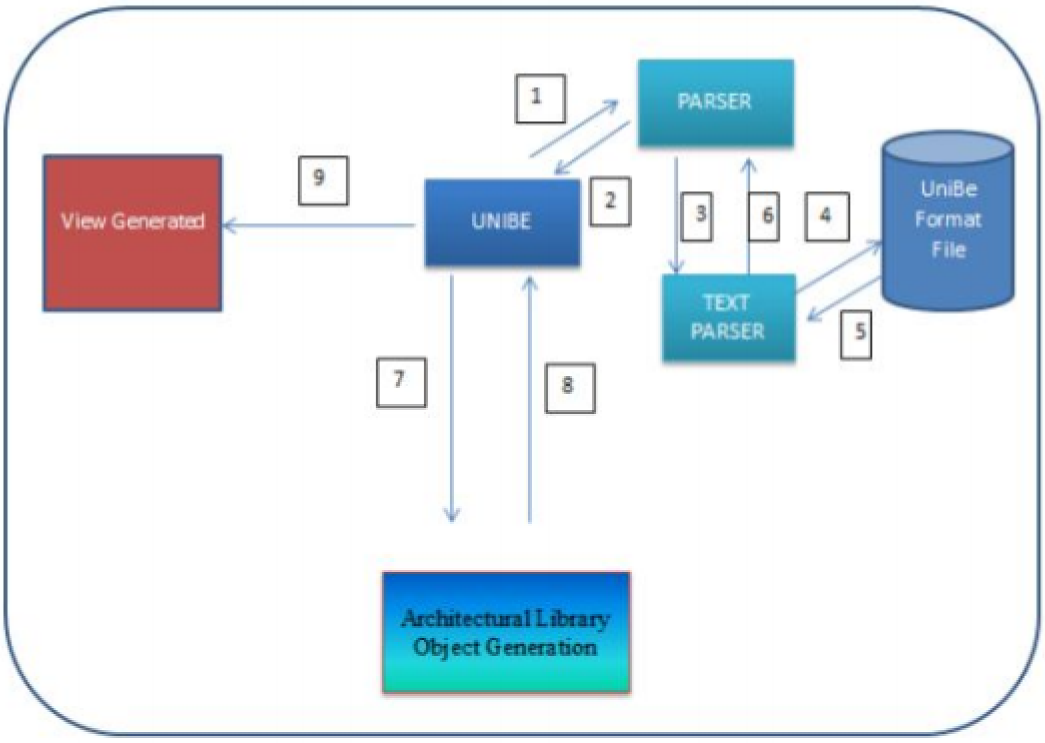


Figure 3.3: Flow Diagram of UNIBE
[1]

- Request to assemble Block particular square Object.
- Request is sent to generation of Sub Blocks.
- Sub Block Object is returned which is to be set on top of BLOCK.
- Object of Block is Returned.
- STRUCTURE Object is Return.

Chapter 4

Implementation and Result

4.1 Existing Approach

In order to develop the memory Compiler, The user will provide **INPUT PARAMETERS** Configuration file along with needed other things like FE product, BE-Lib, Modules, Memory-Cell. Using this six product including BE product generated by our team will provide to **COMPILER** and develop the view of memory cut as **PRODUCT**.

To generate one mature memory cut will take 10-12 moths and then it is deliver for fabrication.

This thing we have done for one particular architecture. Now parameter change then we need to repeat the whole process to generate memory cut although it is belong to same technology. The existing approach as shown in figure

This approach is trust-worthy but taking long time to generate the memory cut. We need one another approach which maintain the flexibility of process and take less time compare to conventional approach. So it is demand to develop new approach should be feasible, efficient terms of time and flexible.

4.2 Proposed Approach

To overcome all the problem faced by the existing approach, We need unique and common framework which support all combination of configuration parameter along with different technology [3]

UNIBE is framework which is product of ST Microelectronics supports different technology of memory along with configuration parameter.The input to this framework

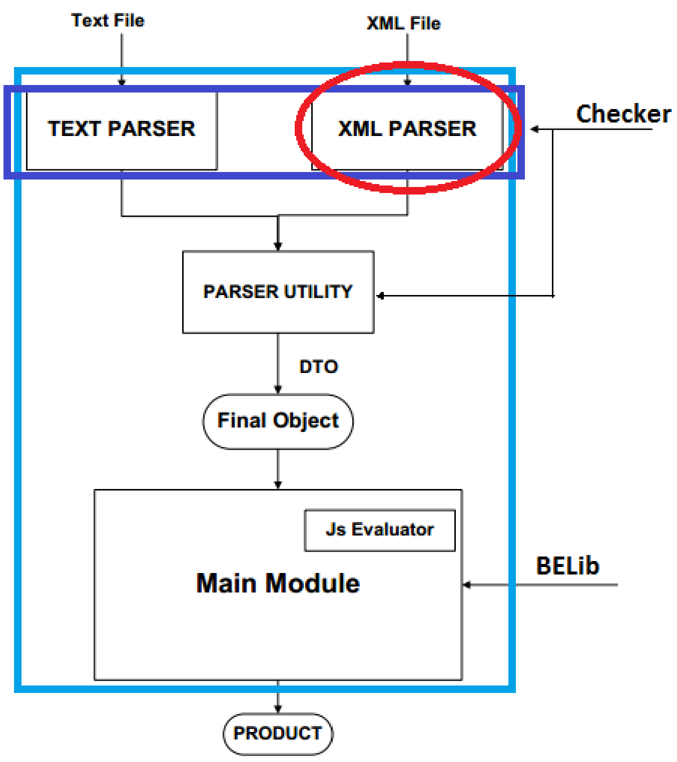
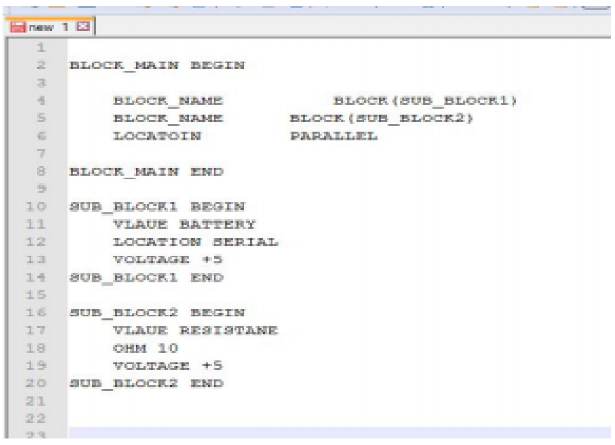


Figure 4.1: Proposed Approach
[3]

is the Text file of configuration parameter and what kind of view need to generate. This would work as black-box module to the end user this framework is taking care of everything for which manually done by the team. Now the term used in proposed architecture is explain as below:



```

BLOCK_MAIN BEGIN

     BLOCK_NAME           BLOCK(SUB_BLOCK1)
     BLOCK_NAME       BLOCK(SUB_BLOCK2)
     LOCATOIN         PARALLEL

BLOCK_MAIN END

SUB_BLOCK1 BEGIN
     VLAUE BATTERY
     LOCATION SERIAL
     VOLTAGE +5
SUB_BLOCK1 END

SUB_BLOCK2 BEGIN
     VLAUE RESISTANE
     OHM 10
     VOLTAGE +5
SUB_BLOCK2 END
```

Figure 4.2: Text File

- TextFile : This is a input file which contain the configuration data regarding developing memory cut which is shown in figure 3.3 .

- Parser : This module parse any kind of record that is supported by UNIBE and adherent it to DTO. DTO is specific sort of game plan to store data. We are using Objects to store data. Finally DTO are traded to challenge of particular class.

4.2.1 Checker

The user provide input file in form of Text file or XML file. The Text file is shown in figure 3.3. The defined structure of file is written by user which may contain error by mistake. So we need to validate the file by parsing.There is two level of checker :Checker Level 1 and Checker Level 2 that is static and dynamic checker respectively.

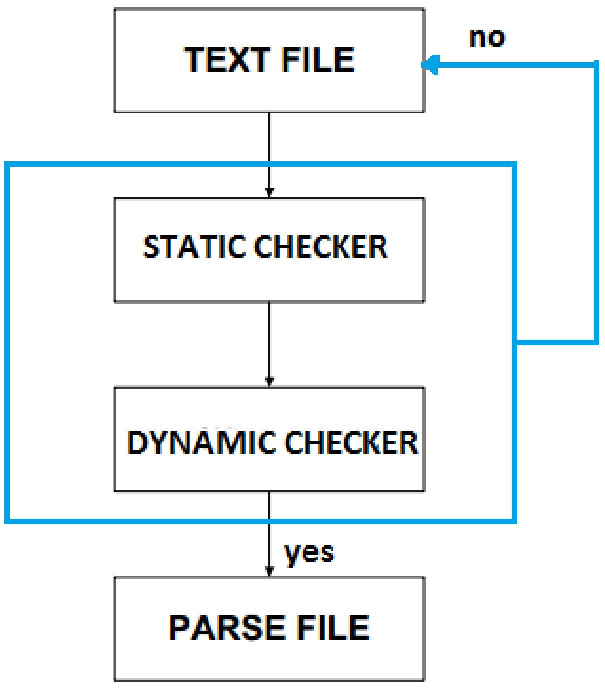


Figure 4.3: Checker Flowchart

- Static Checker : Checking syntax of defined structure. For example it should start with BEGIN and END prefix followed by name of the parameter.

- Dynamic Checker : Checking defined parameter and evaluate it. For example, It is checking parameter inside the block like NAME, REF etc parameter defined properly or not. It is checking the parameter which has variable value and it should be as per domain of the data.

4.2.2 js Evaluator

This is javascript evaluator which is used for evaluation of variable value. One of the variable is “itr_structure“ indicated in figure 3.3. The need for this evaluation engine is explain here : For instance above mention variable value is used in let’s say on ten place and the value of this variable is different. We can define this variable as constant in ten different place but again we need to do manually which is not desirable.to automate the value of the value of this variable is generic so it can be done through java evaluator. We are using “Rhino“ javascript Engine. It is open-source and time taken by it is small.

4.2.3 Some Challenging Problem

Is is easy to parse the file based on defined syntax inside file but it is challenging task to evaluate the parameter defined inside it for example “itr_structBank“ which has variable value and it can be change based on the other parameter dependency. We are parsing the file through java code but it is not evaluating the value. So to evaluate the value we are using java script.

For evaluation of value what kind of script engine to be used. If we choose particular script engine it should not affect the other modules gets updated. Again there are many kind of script engine available in market but we need open source best script engine. For that we find two script engine. [4]

- Rhino
- Nashorn

Both are developed in java but the difference is lying for evaluation of script. Oracle has shown the graph for comparing both script-engine as below :

For evaluation of java script we need script engine which evaluate the variable and use the value ahead. Again the value of same variable can be change for another dependency so we need to evaluate again and again. So as solution we use Rhino script engine. Again for evaluation we are using threading which reduce the time of execution.

As per analysis of compiler to make the framework generic, It has about 146 total variable which is used in almost all compiler. Again it has domain of value can be different.so it need to run in worst case. This situation can be handle by thread but for normal machine it may through out of memory exception. So we have limited number

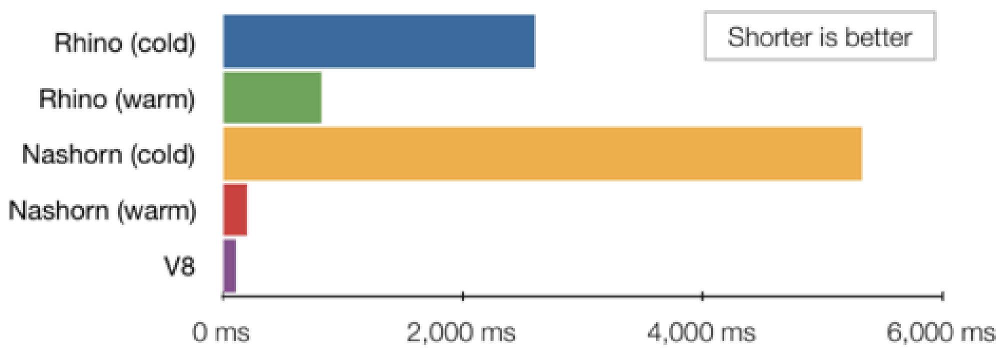


Figure 4.4: Comparison Between two Script Engine
courtesy: developer.mozilla.org

of thread. Again we need to validate the value at run time as it is matching the domain value. Otherwise it should throw an error. So we need to maintain the variable value for all cases and it is handle by counter and parent_counter value which is worse case.

4.3 DEF view

As we know that the main product which is used for fabrication of chip is GDS view. But along with this main view others views are also generated like CDL view. One more view is introduce which is DEF view.

The cells which contains the DEF view cell needs to abut in memory compiler. DEF view method generation needs to be call after abutment of the other memory component.The reason behind it calling after abutment of the memory is that because we need to link the DEF view cells with signal pins.

The DEF view is required as the pin properties contains only small information but could not tell other things like it's orientation, its length.

The DEF view doesn't impact the other component of memory, But the purpose behind the introduction of this view is to validate the position and properties of Signal pins.The signal pins can be Address pins or Data pins.It also contain the information about orientation.

4.3.1 DEF view algorithm

The algorithm is as shown in the figure. The input is provided as cells which is place in the memory while streaming the GDS file. Now the one cell name is provided through which we can associate the information regarding the cell. The cells are find from the GDS file and make them sort so that we can ensure that no cell can be left and then it is provided to the label selector which extract the cell and make them differentiate . Now if we find the desire cell then it rename the cell and insert the other information regarding the pin otherwise itâĂŹs simply put the information regarding the cell and generate the DEF file as output.[5] The algorithm is shown in below:

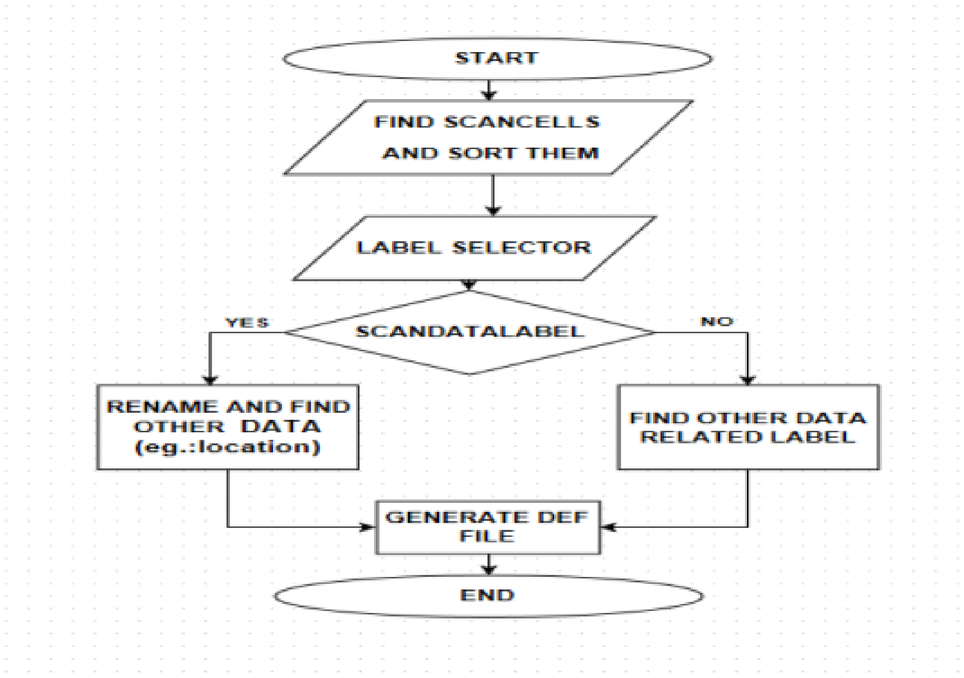
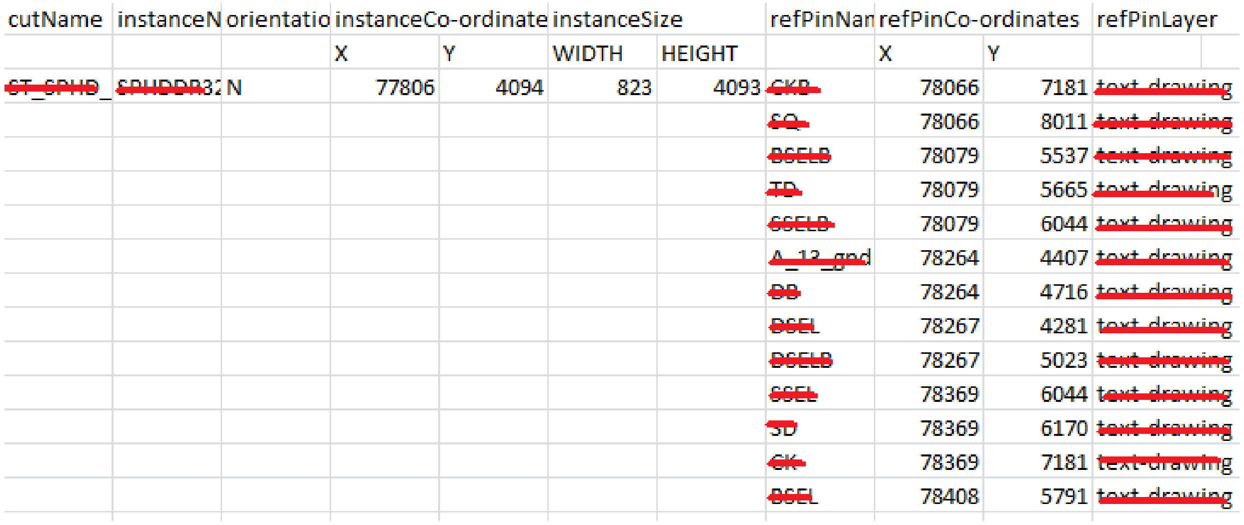

START
FIND SCANCELLS AND SORT THEM
LABEL SELECTOR
SCANDATALABEL
YES
NO
RENAME AND FIND OTHER DATA (eg.:location)
FIND OTHER DATA RELATED LABEL
GENERATE DEF FILE
END

Figure 4.5: DEF view Generation Algorithm



cutName	instanceN	orientatio	instanceCo-ordinate		instanceSize		refPinNan	refPinCo-ordinates		refPinLayer
			X	Y	WIDTH	HEIGHT		X	Y	
[redacted]	[redacted]	N	77806	4094	823	4093	[redacted]	78066	7181	[redacted]
							[redacted]	78066	8011	[redacted]
							[redacted]	78079	5537	[redacted]
							[redacted]	78079	5665	[redacted]
							[redacted]	78079	6044	[redacted]
							[redacted]	78264	4407	[redacted]
							[redacted]	78264	4716	[redacted]
							[redacted]	78267	4281	[redacted]
							[redacted]	78267	5023	[redacted]
							[redacted]	78369	6044	[redacted]
							[redacted]	78369	6170	[redacted]
							[redacted]	78369	7181	[redacted]
							[redacted]	78408	5791	[redacted]

Figure 4.6: DEF view Generation Output

4.3.2 Graphical view of DEF view

As we know that memory in the form of Graphical view is complex. So for better understanding of DEF view I have closed the other layer of memory. Following is the Graphical view of DEF view:

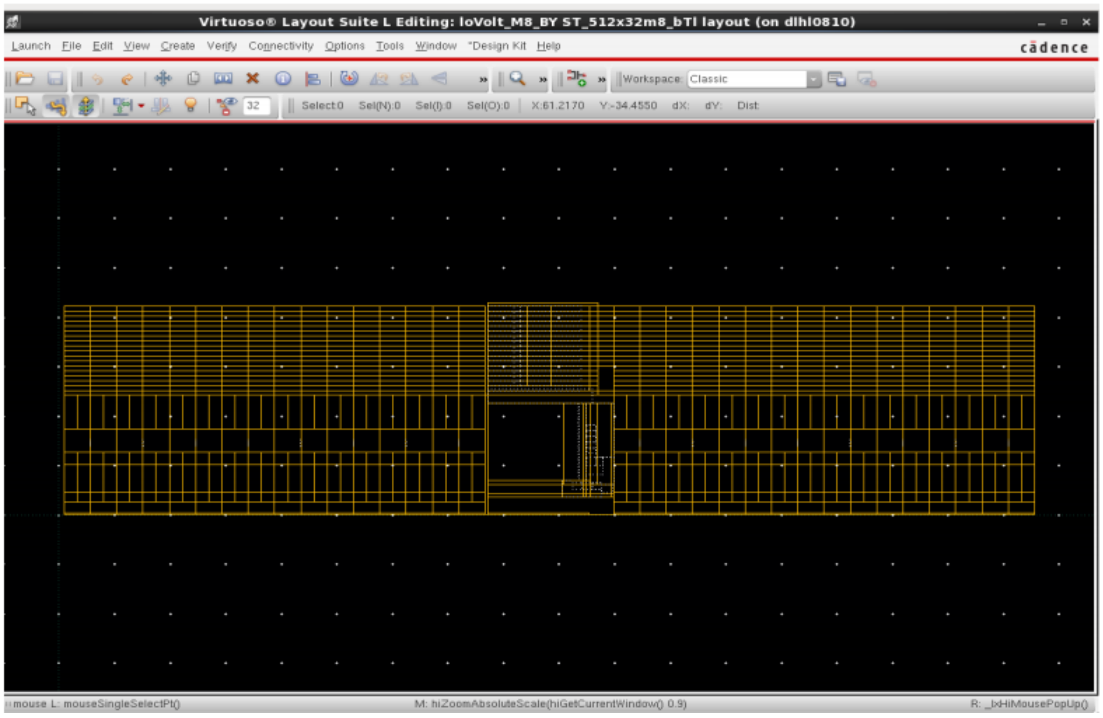

Virtuoso® Layout Suite L Editing: loVolt_M8_BY ST_512x32m8_bTI layout (on dlhl0810)
Launch File Edit View Create Verify Connectivity Options Tools Window "Design Kit Help
cadence
Workspace: Classic
Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X:61.2170 Y:-34.4550 dX: dY: Dist:
mouse L: mouseSingleSelectPt()
M: hiZoomAbsoluteScale(hiGetCurrentWindow() 0.9)
R: _lxHiMousePopUp()

Figure 4.7: TEXT layer of Memory

The first figure shows the text layer on which the cell are placed for the DEF view. The second figure shows the def view cells which is connected with the signal pins which contain the properties of signal pins. Third figure shows label promoted on the top level and one of the label represent the propertied of the signal pin.

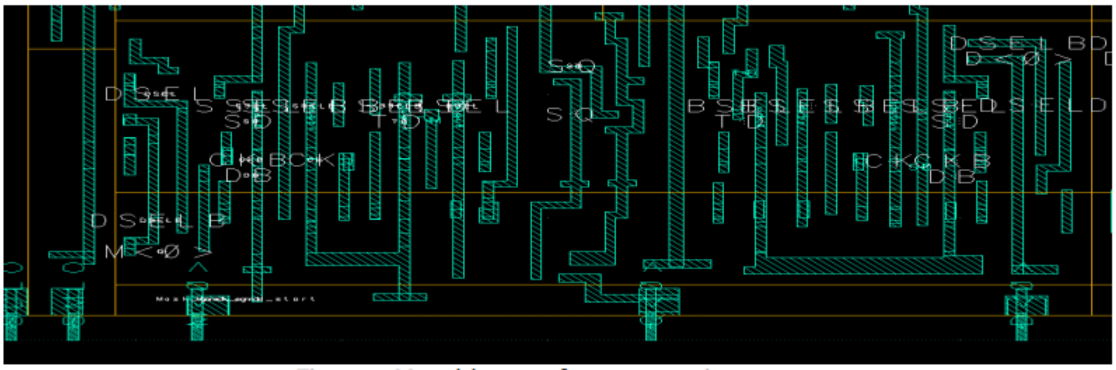


Figure 4.8: METAL layer of Memory

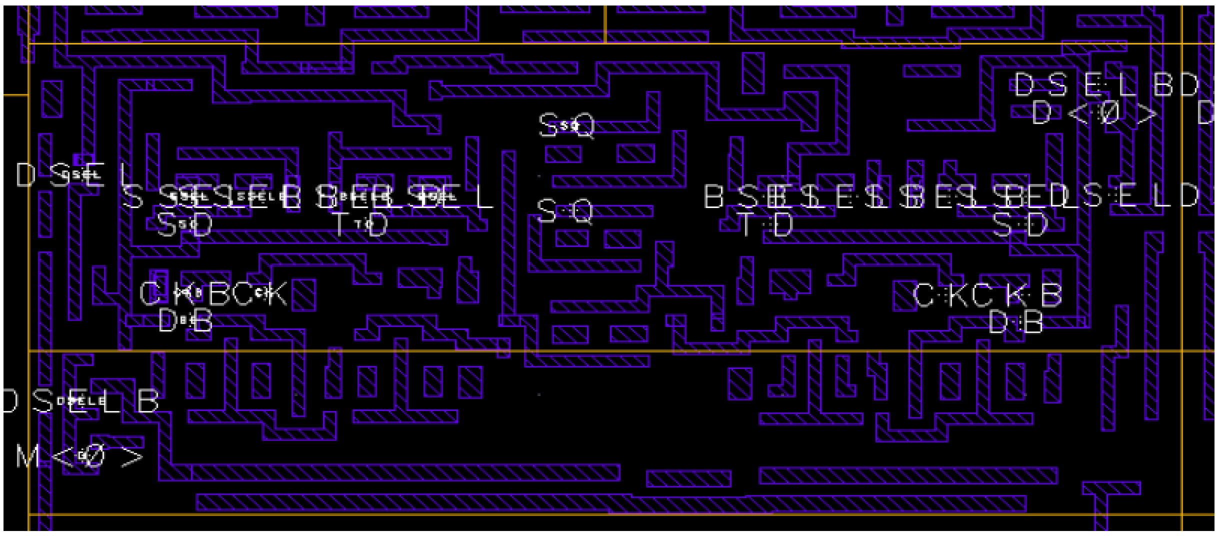
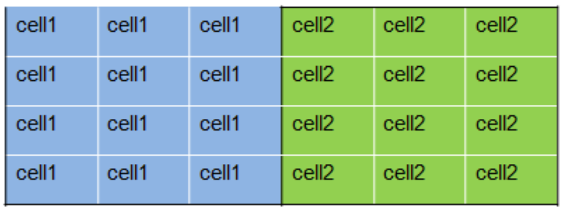


Figure 4.9: Top view label promotion

4.4 Optimization

Memory is composition of cells. By making particular pattern, repeating the pattern we can built the core of the memory. Given below figure through demonstrate the optimization in core. This time difference between shifting the code doe



cell1	cell1	cell1	cell2	cell2	cell2
cell1	cell1	cell1	cell2	cell2	cell2
cell1	cell1	cell1	cell2	cell2	cell2
cell1	cell1	cell1	cell2	cell2	cell2

Figure 4.10: Memory cell

If we look at the legacy JAVA code of the developing the above structure. Its look like below:

Given below is the problem which is written in TEXT file and parse through TEXT PARSER

```
BEGIN STRUCTURE BLK_MAIN
          DIRECTION          "LEFT_TO_RIGHT}"
          NAME          "ALIAS(ALIAS_BLK1)"
          NAME          "ALIAS(ALIAS_BLK2)"
ENDS BLK_MAIN
```

```
BEGIN  ALIAS  ALIAS_BLK1
           NAME            "BLOCK(BLK_SUB1)"
ENDS  ALIAS_BLK21

BEGIN  ALIAS  ALIAS_BLK2
          NAME             "BLOCK(BLK_SUB2)"
ENDS  ALIAS_BLK2

BEGIN  BLOCK  BLK_SUB1
          NAME             "BLOCK(BLK_C)"
          ITERATE               "1:4"
ENDS  BLK_SUB2

BEGIN  BLOCK  BLK_C
          NAME             "cell1"
          ITERATE                 "1:3"
ENDS  BLK_C

BEGIN  BLOCK  BLK_SUB2
          NAME                 "BLOCK(BLK_E)"
          ITERATE            "1:4"
ENDS  BLK_SUB2

BEGIN  BLOCK  BLK_E
          NAME             "cell2"
          ITERATE  "1:3"
ENDS  BLK_E
```

Below is the legacy code in JAVA:

```
Class BLK_MAIN {
BLK_MAIN() {
            for( int i = 1 ; i <= 2 ; i++ ) {
                 for( int j = 0 ; j < 4  ; j++) {
                     for( int k = 0 ; k < 3 ; k++ ){
                            ABUTRT( âĂIJcellâĂİ + i );
                        }
                 }
           }
   }
}
```

Here is the code which is written in text file and it is optimized one.

```
BEGIN BLOCK BLK_A()
        NAME                   "BLK_B(x=UNI_COUNTER)"
        ITERATE                "1:2"
ENDS BLK_A
BEGIN BLOCK BLK_B(x)
        NAME                   "BLK_C(x=x)âĂİ
        ITERATE                "1:4"
ENDS BLK_B
BEGIN BLOCK BLK_C(x)
        NAME                   (x\%2)==0?"cell1":"cell2"
        ITERATE                "1:3"
ENDS BLK_C
```

Chapter 5

Validation

5.1 Introduction

The previous existing architecture is used to generate the compiler. It is trustworthy but manual. It is taking the investment of time. So to save the time and make more efficient we are introduced with new architecture that is **UNIBE**. Now, The product is developed from the **UNIBE** architecture which is time efficient and flexible from the user perspective.

Now the problem is that how to validate the product? We need something which evaluate the thing and produce the desirable result by comparing the compiler generated form old and new architecture.

5.2 Product Diff Tool

The product diff tool take input as two files which has path regarding product produced with old approach and new approach. It also contain the exhaustive cutlist and BELib path. Both are common for old and new approach. The product validation tool contains four parameter

- Validation Script : It contains Script like BEproddiif, gds2txt, memvalidkit.
- Output : Contain generalized and cutwise report.
- Local Product : contain exhaustive cutlist and BELib.
- Confiles : It is input file to validation scripts.

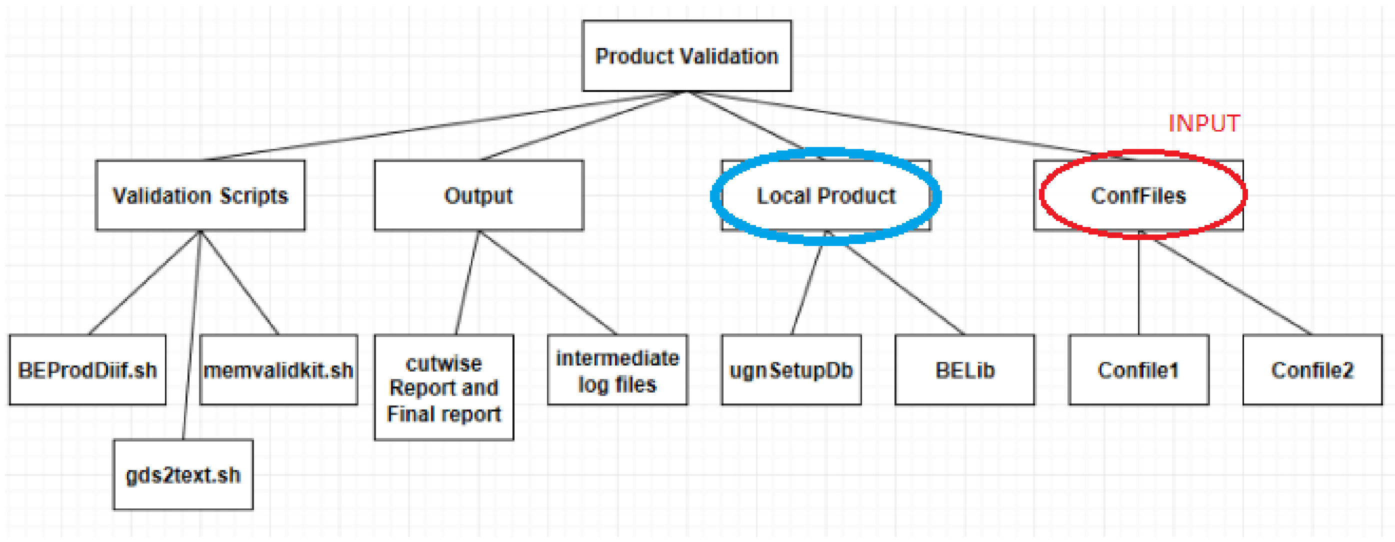


Figure 5.1: Structure diagram of Product diff Tool

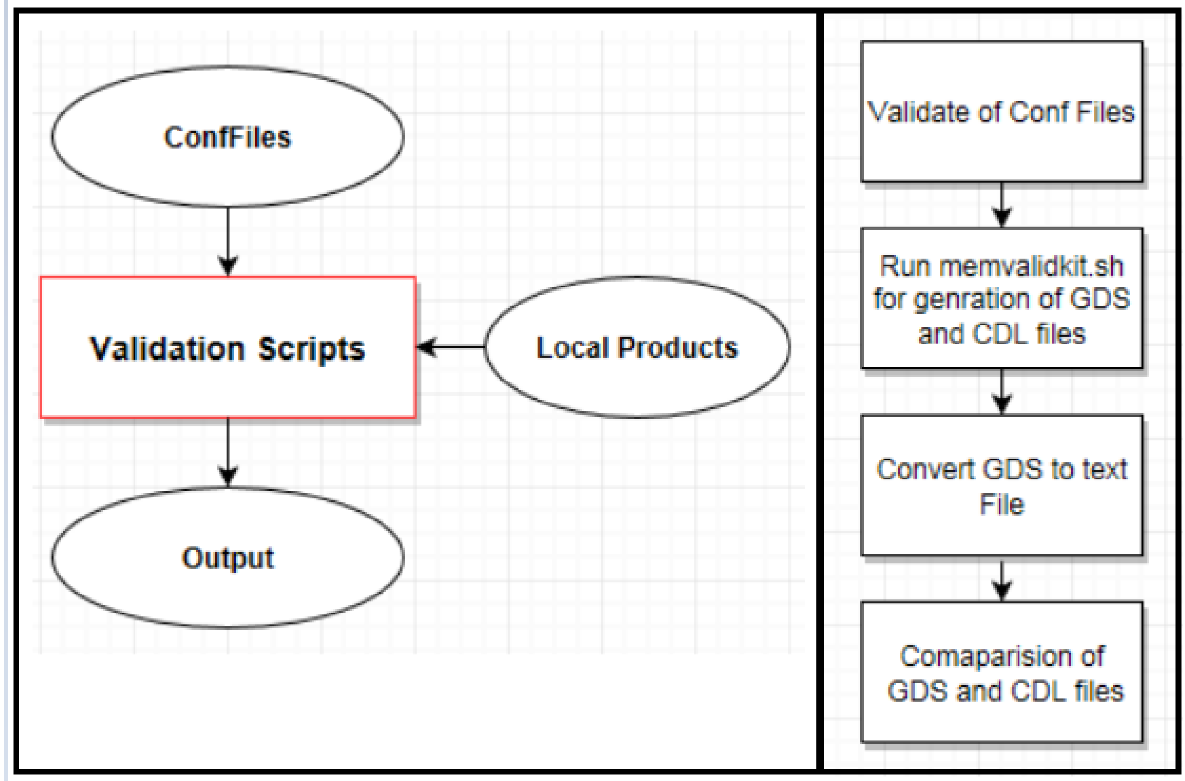
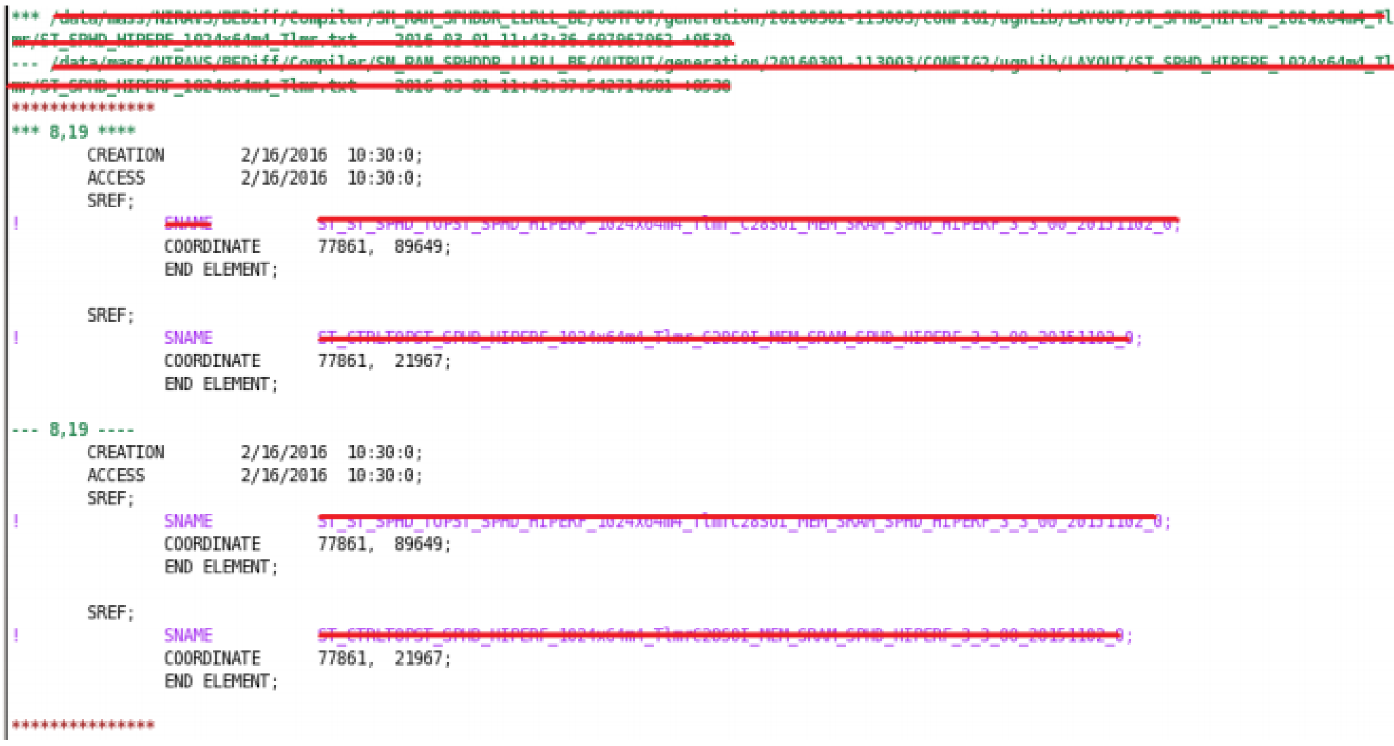


Figure 5.2: Flow of Product diff Tool

Above figure shows the structure diagram of Product Diff Tool:

The confFiles which is the data to the script contains the path of product and other parameter. The primary script BEprodDiff.sh accept the path and built up environment to run the script. It calls the memvalidkit.sh script which create the GDS and CDL files for both old and also new approach with gave through cutlist.

As GDS record is binary so specifically we can't look at it, so we have to change over into text content through gds2text.sh script. The principle script compare the cutwise GDS and CDL document and create the generalized and in addition cutwise report. The yield output is appeared in beneath figure.



```
*** [redacted]
--- [redacted]
***************
*** 8,19 ****
        CREATION        2/16/2016  10:30:0;
        ACCESS          2/16/2016  10:30:0;
        SREF;
!               [redacted]      [redacted]
                COORDINATE      77861,  89649;
                END ELEMENT;

        SREF;
!               SNAME           [redacted];
                COORDINATE      77861,  21967;
                END ELEMENT;

--- 8,19 ----
        CREATION        2/16/2016  10:30:0;
        ACCESS          2/16/2016  10:30:0;
        SREF;
!               SNAME           [redacted];
                COORDINATE      77861,  89649;
                END ELEMENT;

        SREF;
!               SNAME           [redacted];
                COORDINATE      77861,  21967;
                END ELEMENT;

***************
```

Figure 5.3: Demo Output of Product Diff Tool

Chapter 6

Conclusion and Future Scope

6.1 Conclusion and Future Scope

As we know quality time is more important in microelectronics.so it is require to make the thing automate and as possible as optimize.By making a unique framework, it gives platform to develop the compiler for all technology and architecture supporting combination of all configuration parameter.Moreover optimization in Layout code and input file saves quality time.change in single parameter doesn't effect others thing. Hence it reduces the time and effort to develop one mature compiler.

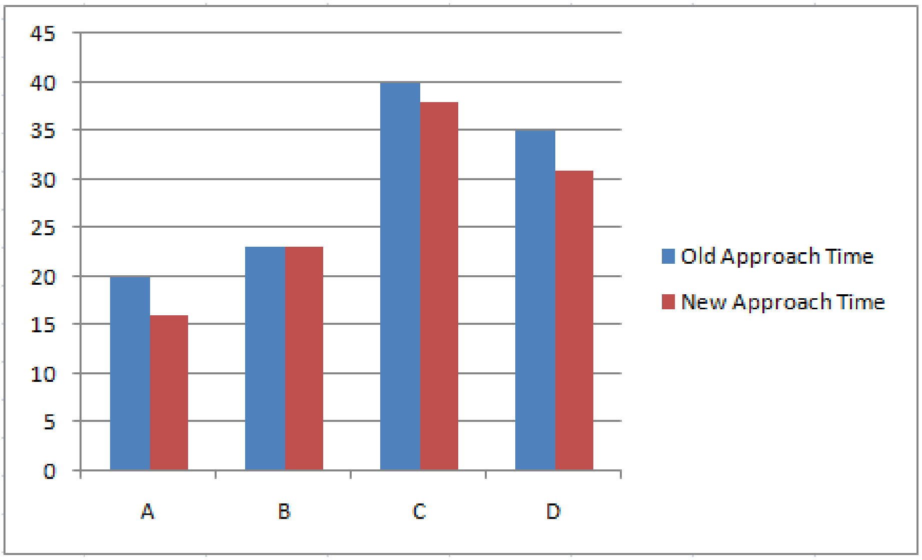


Figure 6.1: Execution Graph : Compiler $\rightarrow Time$

If we look at the Execution graph for the different technology compiler,the time is taken by the old approach is more than the new approach.

By looking at the validation graph, It also convey that the new approach is batter

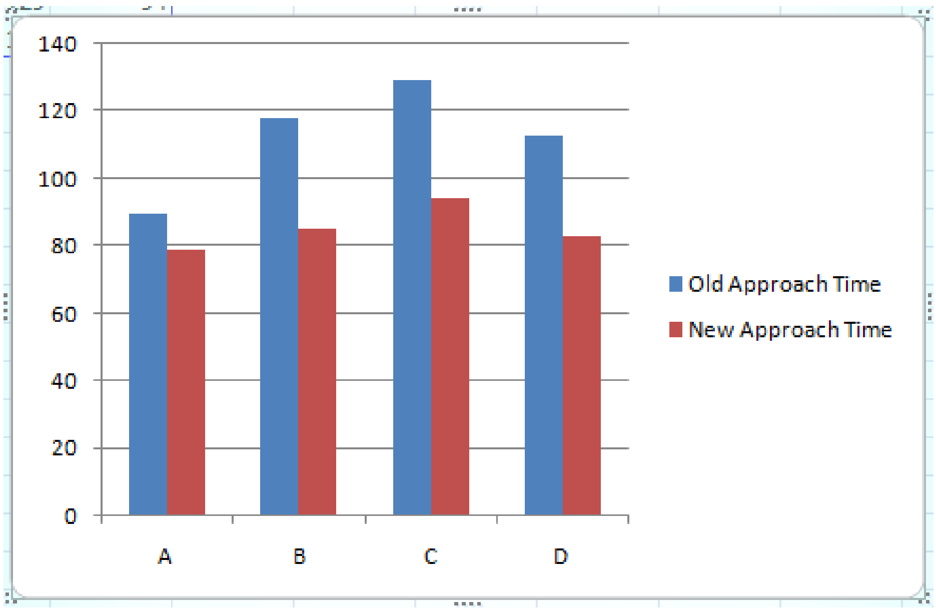


Figure 6.2: Validation Graph : Compiler $\rightarrow$ *Time*

for validation of the compiler,But there is still scope for improvement for the validation script which minimize the validation time at fabrication. here still design time is almost same but it improve the quality steps to develop the SoC.

As enhancement we can reduce the time of validation of input File and also provide input in form of GUI. we can also reduce the time of validation script which enhance the overall performance of the SoC development.

References

[1] STMicroelectronics, "St tools," pp. 1 – 8, October 2015.

[2] STMicroelectronics, "St documents on cut generation," pp. 1 – 17, October 2015.

[3] STMicroelectronics, "St internal documents on memories and training manuals," pp. 314 – 317, October 2015.

[4] Vogella, "www.vogella.com/tutorials/java.html," Jannuary 2016.

[5] Oracle, "docs.oracle.com/javase/7/docs/technotes/guides/scripting/programmergui," January 2016.