Electrically Aware Design Flow for Sub-micron Technologies

Major Project Report

Submitted in fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Jagrut B Joshi
(14MECV08)

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2016

Electrically Aware Design Flow for Submicron Technologies

Major Project Report

Submitted in fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Jagrut Joshi
(14MECV08)

Under the guidance of

External Project Guide:	Internal Project Guide:
Mr. Indushekhar Benjwal Engineering Manager, Intel India Technology Pvt. Ltd., Bangalore.	**Dr. Usha Mehta** Professor (VLSI Design), Institute of Technology, Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2016

NIRMA
UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED ‘A’ GRADE

Certificate

This is to certify that the Major Project entitled **“Electrically Aware Design Flow fro Submicron Technologies ”** submitted by **Jagrut Joshi (14MECV08)**, towards the fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven’t been submitted to any other university or institution for award of any degree or diploma.

Date:	Place: Ahmedabad
Internal Guide	**Program Co-ordinator**
Dr. Usha Mehta (Professor,EC)	**Dr. N. M. Devashrayee** (Professor,EC)
HOD	**Director**
Dr. P. N. Tekwani (Head of EE Dept.)	**Dr. P. N. Tekwani** (Director, IT-NU)

Intel Technology India Pvt. Ltd.

Certificate

This is to certify that the Major Project entitled **"Electrically Aware Design Flow fro Submicron Technologies "** submitted by **Jagrut Joshi (14MECV08)**, towards the fulfillment of the requirements for the degree of Master of Technology in VLSI Design, 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 major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Bangalore

Mr. Indushekhar Benjwal
Engineering Manager
Intel Technology India Pvt. Ltd.,
Bangalore.

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.

2. Due acknowledgment has been made in the text to all other material used.

- Jagrut Joshi
14MECV08

Acknowledgements

First and foremost, sincere gratitude to my manager Mr. Indushekhar Benjwal. Also I want to thank Intel Technology India Private Limited, Bangalore for assigning me such project and guide me through.

I would like to express my gratitude and sincere thanks to my mentor Mrs. Pooja Melinamani at Intel Technology India Private Limited, Bangalore for her valuable guidance throughout the internship, she has given me valuable advices and support for my project work which I am very lucky to benefit from.

I would also thank to my Program Coordinator, Dr. N. M. Devashrayee, Professor, VLSI Design, Institute of Technology, Nirma University, Ahmedabad for giving valuable the advice and support throughout the academic period.

I would also thank to my Project Coordinator, Prof. Usha Mehta, Professor, EC Engineering Department, Institute of Technology, Nirma University, Ahmedabad for giving valuable support for the project work.

I would also like to express my gratitude to all faculty members of Nirma University for providing encouragement, exchanging knowledge during my post-graduate program.

I also owe my colleagues in the Intel, special thanks for helping me on this path and for making project at Intel more enjoyable.

- Jagrut Joshi
14MECV08

Abstract

With increase in design complexity and stringent Time to Market (TTM) requirements, it is important that productivity improvement measures are taken to help designers to be in pace with the demands. At each phase of the design there are utilities which help improve the productivity by incorporating design automation and information feed forwarding. This project is aimed at enabling an electrically aware backend design flow which will try to improve the productivity of Mask Designers by feed forwarding electrical parameters early in the design flow so that the number of iterations in the backend design flow is minimized, and better turnaround time is obtained.

Contents

List of Figures

Abbreviations

EAD	Electrically Aware Design
VXL	Virtuoso XL
GFS	Generate From Source
TTM	Time To Market
ModGen	Module Generator
TAT	Turn Around Time
SPEF	Standard Parasitic Exchange Format
RV	Reliability Verification
EM	Electro-Migration

Chapter 1

Introduction

1.1 Overview

The advancement in technology and continuous scaling down has been delivering nanoscale devices which are capable of performing a wide variety of functions. Moore's law states that the number of transistors per square inch on integrated circuits will double every year. This law holds good and advance technologies have been able to satisfy this law. Revolution brings along opportunities as well as obstacles to the design engineers. Lot of factors have to be considered in the design of integrated circuits at lower technology nodes. Designer faces a large number of challenges during the implementation phase. There are a number of approaches to aid the implementation flow. One such approach is the use of automatic tools for Placement and Routing which constitutes a majority of the design flow. There are lot of advantages associated with it which are explained in the later sections. A thorough understanding and the experimental approach followed to implement the flow has been explained. Different blocks have been considered for the implementation of the flow and complete evaluation of the flow has been done.

1.2 Project Motivation

Intel Corporation being one of the worlds largest and highest valued semiconductor company has been continuously striving towards working on lower technology nodes and successfully introducing leading edge process which have a lot of advantages associated with it.Following Moore's law, Intel is now working towards lower technology nodes. There has been a lot of improvement in terms of power dissipation, area and run time due to smaller channel length, width and thinner metals being used. These advantages bring along a few challenges with them. Certain issues such as short channel effects, crosstalk, Electromigration due to the migration of metal ions, increased interconnect delay, congestion while routing the tracks, low operating voltage to limit power dissipation which results in higher leakage currents and

DRC violations pose problems to the design. As the transistor size reaches deep submicron process, implementation of the design becomes a greater challenge. In order to overcome these challenges and address all the issues faced in the lower technology nodes, there have been many methods and tools to facilitate the entire design process. Since extraction and Electro-migration accounts for a major part of information flow in the back end flow, using feed forwarding of information in this phase can help improve mask designers productivity and save costly iterations of the backend flow.

1.3 Project Objective

The main objective of this project is to enable electrically aware design flow in Intel's submicron process. Different blocks have been considered for the evaluation purpose. During the implementation of the flow, lot of challenges have been faced and methods to overcome these challenges have been explained in the later sections. EAD model generation and RC extraction have been enabled. Results are compared with sign-off tool via simulation and the results have been compared. The evaluation of the flow has been thoroughly analyzed and feedback has been given to the respective teams for improvement.

Chapter 2

Background Study

This chapter provides a brief introduction of the VLSI custom back-end flow. It also prepares the base for parasitic extraction flow and the importance of real-time parasitics in the VLSI design flow. The Chapter gives an introduction of EAD

2.1 Custom design flow

The CMOS circuit design process consists of defining circuit inputs and outputs, hand calculations, circuit simulations, circuit layout, simulations including parasitics, re-evaluation of circuit inputs and outputs, fabrication, and testing. A flowchart of this process is shown in Fig. 2.1. The circuit specifications are rarely set in concrete; that is, they can change as the project matures. This can be the result of trade-offs made between cost and performance, changes in the marketability of the chip, or simply changes in the customer's needs. In almost all cases, major changes after the chip has gone into production are not possible.

The task of laying out the IC is often given to a layout designer. However, it is extremely important that the engineer can lay out a chip (and can provide direction to the layout designer on how to layout a chip) and understand the parasitics involved in the layout. Parasitics are the stray capacitances, inductances, pn junctions, and bipolar transistors, with the associated problems (breakdown, stored charge, latch-up, etc.). A fundamental understanding of these problems is important in precision/high-speed design.

The iteration of layout and physical verification is very expensive as it involves processes like LVS cleaning, DRC cleaning, Parasitic extraction and simulation which consumes lot of time of the designer. Hence it is important that we minimize the number of iterations so as to improve turn-around time.

There are DRC aware and connectivity driven placement and routing utilities which help prevent physical errors inline with the design flow. However parasitic extraction can be done only if LVS clean design is available, for that it is important that the layout design is completed. Avalability of parasitic information early in the design flow can ease the work of mask designer and reduce the turn around time for a given design. Electrically aware design flow provides this information to the layout designer while laying out the design.

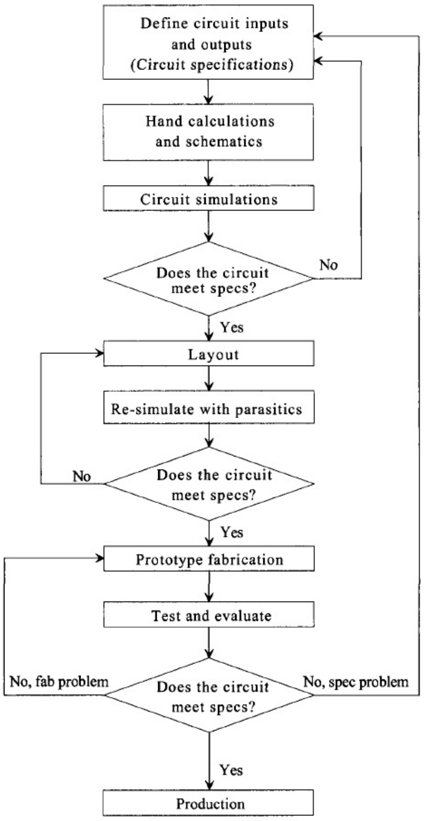


Figure 2.1: Flow chart of CMOS IC design process

Electrically Aware Design (EAD) flow, helps minimize costly design iterations by allowing you to consider physical effects up-front and in-design, without requiring a completed, LVS-clean layout view. As part of this flow, you can capture the current data from simulations, use it to perform electro-migration (EM) checks with the actual physical design layout, and update the layout to correct any violations. DC, Average, RMS and Peak EM checks are supported.

In addition, an extraction engine lets you extract and visualize RC parasitics on nets during design creation and to specify parasitic constraints for layout generation. An integrated high precision solver lets you perform critical net analysis by extracting capacitance and resistance on critical nets with a high degree of accuracy. These parasitics can then be used to perform early simulations to verify the design and circuit specifications.

2.2 Introduction to EAD

The reducing sizes of advanced integrated circuits means that wire widths are becoming smaller and the lengths of connecting wires longer. In addition, there is an exponential increase in the density of the current flowing through the wires. As a result, integrated circuits become more vulnerable to electro-migration (EM), which is the gradual dislocation of metal atoms in a conductor.

To produce a reliable and sustainable integrated circuit, it is therefore essential that you perform comprehensive electromigration checks and ensure that the physical design of every design component is electrically correct by construction and optimized to meet the design intent.

Electrically Aware Design (EAD) flow lets you capture the current data from design simulations, extract and visualize RC parasitics as you edit the layout, perform EM checks and fix violations. You can further extract parasitics from a partial or a complete layout and rerun simulations to check if the output specifications are met.

Figure 2.2 shows the electrically aware design flow for custom IC design. You begin by creating a schematic, running simulations, and modifying your design so that it meets the desired specifications. When the specifications are met, you then configure the testbench to save the current data (average, RMS, and peak) that will be used for EM checking in Layout EAD. EM depends on various factors, such as the local current density (current per unit area), the homogeneity of the current flow through a region, and the blech length. All these factors are captured for different conductor layers and vias as rules in technology files for each process technology specification. Before performing the EM checking, you launch Layout EAD and configure the EAD setup to load the required process settings from technology files. These settings are used during parasitic extraction and EM checking.

As you create or modify the layout of your design, an extraction engine in Layout EAD lets you extract and visualize resistance and capacitance parasitics on nets. An integrated high precision solver also lets you perform critical net analysis by extracting capacitance and resistance on critical nets with a high degree of accuracy.

You can now run EM checks to compare the actual current data with the standard EM limits specified in the technology files. The EAD Browser in Layout EAD shows detailed reports to indicate the pass or fail status of the EM checks for each net in the layout. You can then fix the reported EM violations by modifying the placement and routing of components in the layout or by changing the geometric properties of the nets, and then rerun the EM

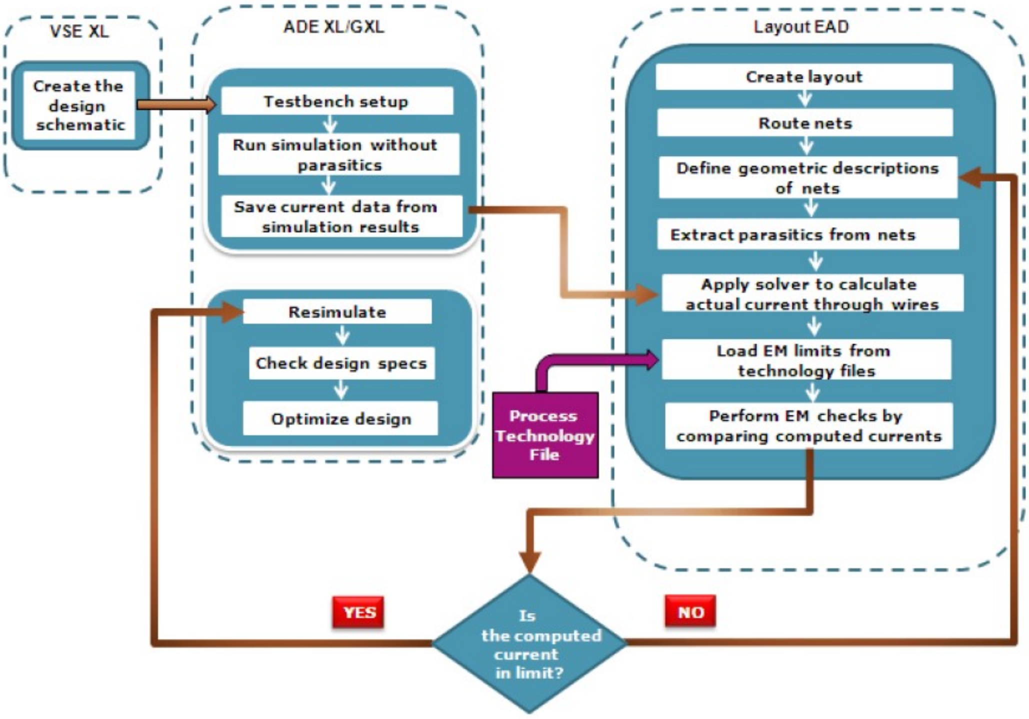

VSE XL
Create the design schematic
ADE XL/GXL
Testbench setup
Run simulation without parasitics
Save current data from simulation results
Resimulate
Check design specs
Optimize design
Process Technology File
Layout EAD
Create layout
Route nets
Define geometric descriptions of nets
Extract parasitics from nets
Apply solver to calculate actual current through wires
Load EM limits from technology files
Perform EM checks by comparing computed currents
YES
NO
Is the computed current in limit?

Figure 2.2: EAD Flow(courtesy of Cadence Design Systems INC.)

checks to verify the results.

One of the main benefits of EAD is that it can be used to analyze parasitic information for both partially and fully completed layouts. Not all devices and nets need to be routed in order to benefit from EAD and you can place and route devices while the application is running.

At any point in the flow, you can take the parasitics from your partially complete or completed layout and resimulate your design to check if the output specifications are still met. You can compare the results of an ideal simulation with those of a parasitic resimulation and, based on those results, further optimize your design and verify the performance and reliability of every physical design component to ensure that the layout is electrically correct by construction.

Chapter 3

EAD Evaluation and Parasitic Extraction

This chapter gives the detailed information about the Real-time parasitic extraction engine use cases. It covers various test cases for parasitic extraction at device level and interconnect level.

3.1 EAD flow evaluation

For EAD to work with intended process, a tech file needs to be generated using eadModelGen tool. It takes in ICT tech file and generates an EAD tech file out of it. The generated tech file needs to be feed to the EAD setup window under extraction corner tab. It also needs a layer map definition to map virtuoso layers with tech file layers. The entire evaluation process is divided into two phases,

1. EAD RC extraction
2. EAD EM analysis

For successful completion of phase 1, it is important to evaluate the accuracy of EAD extracted values as compared to sign-off tool (QRC). Also features of EAD like connectivity info in EAD browser, detailed parasitic tab and point to point resistance calculation were needed to be evaluated. A skill code for layer mapping in EAD setup is developed. There were bugs reported regarding hierarchical net display and net connectivity between layout window and navigator.

The test case for EAD evaluation were generated from scratch which involved learning placement and routing methodologies, achieving device matching by placement and signal matching by routing. Physical verification steps were also performed on the completed layout.

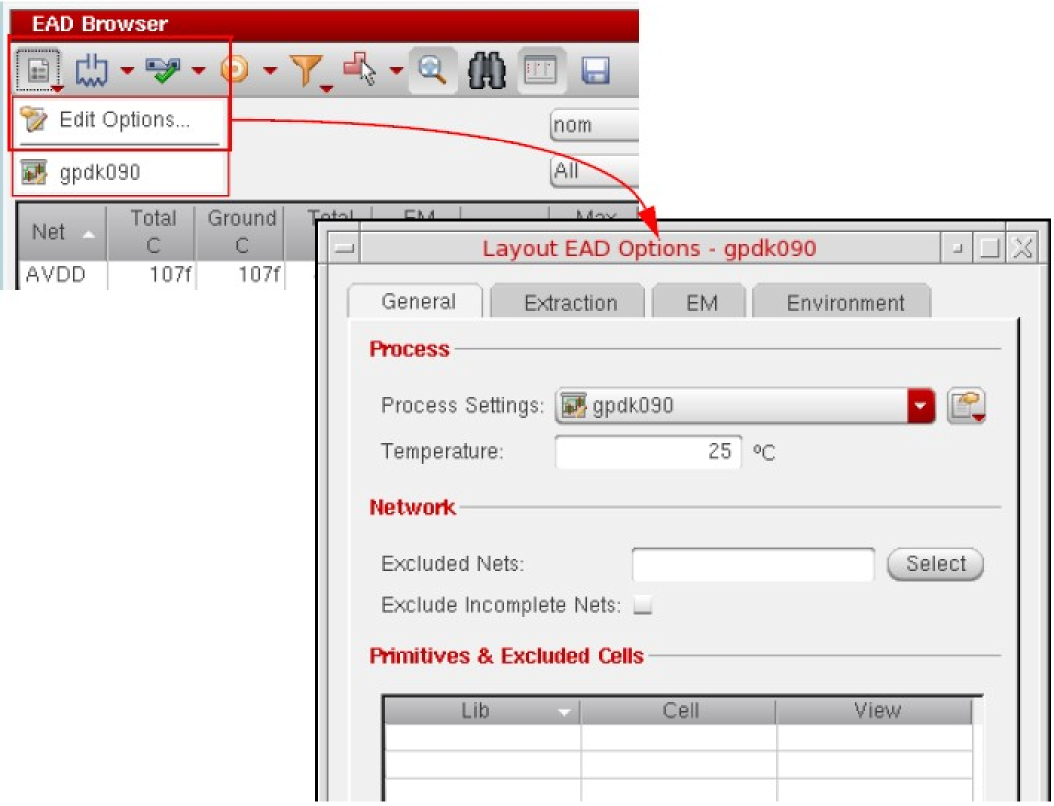
BASH scripts are created to ease the work flow for running sign-off extraction tool for the evaluation purpose. In order to evaluate RC extracted values vs sign-off, test cases of different complexities were taken. Complexities varying from a single leaf cell to an entire IP were evaluated for RC extraction run time and extraction accuracy.

Changes were proposed for tech file generation to improve the accuracy of EAD RC extracted values. With this, EAD design flow was stitched to work with intended process.

3.2 EAD setup

To prepare an EAD Setup,

1. Choose Options ->Edit Options from the EAD Browser toolbar to open the Layout EAD



EAD Browser

Edit Options...

gpdk090

nom

All

Net	Total C	Ground C
AVDD	107f	107f

Layout EAD Options - gpdk090

General | Extraction | EM | Environment

Process

Process Settings: gpdk090

Temperature: 25 °C

Network

Excluded Nets: Select

Exclude Incomplete Nets: □

Primitives & Excluded Cells

Lib	Cell	View

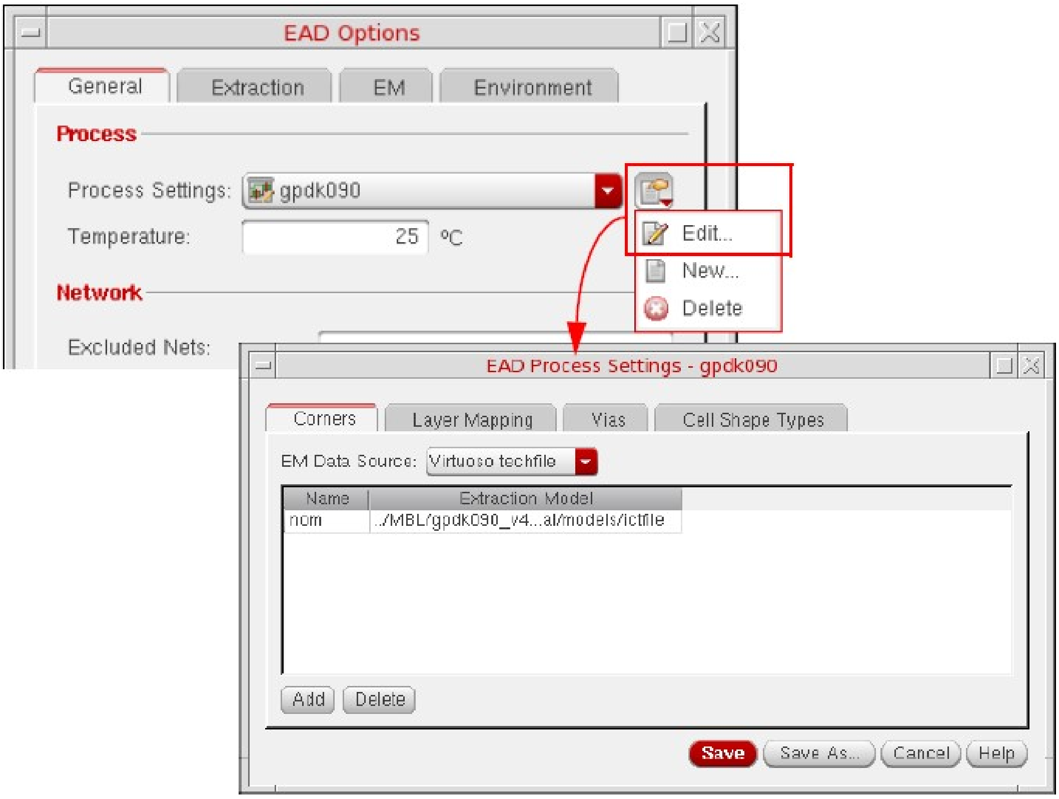
Figure 3.1: EAD Setup form from browser

2. Set the options on each of the four tabs to meet your requirements.

- General lets you specify the process settings to be used in the current session, the extraction temperature to be used, and the nets and cells to be excluded from parasitic extraction.
- Extraction lets you specify the main extraction engine controls, resistance and capacitance thresholds, and high precision C and R extractor settings.
- EM lets you specify the settings to be used for electro-migration checking, including temperature, default and minimum current values, scaling factor, and lifetime values.
- Environment lets you specify how violations are highlighted and control whether nets highlighted and selected in the EAD Browser are also highlighted and selected in the layout view.

3.3 Editing Process Settings

Process settings contain information about the extraction corners and the models to be used for each during EM checking, the mappings between layers in the ICT file and those in the Virtuoso technology file, and how vias and shapes in sub-cells are handled during extraction.



Process Settings: gpdk090

Temperature: 25 °C

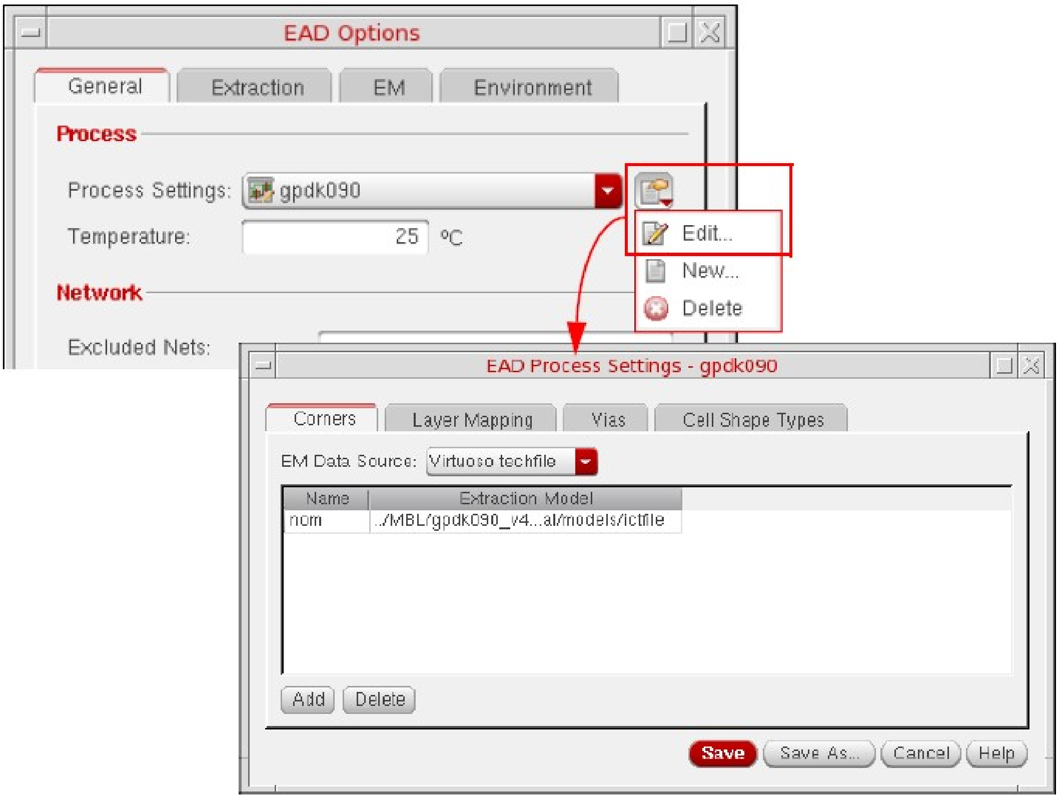
EM Data Source: Virtuoso techfile

Name	Extraction Model
nom	../MBL/gpdk090_v4...al/models/ictfile

Figure 3.2: Process setup form

Corners is where you associate your extraction corners with the ICT file to be used for extraction. The EM Data Source field lets you specify the source of the EM models and rules; either a Virtuoso technology file, an ICT file, or an EM data file.

You use the Layout EAD Options form to define the setups required to drive parasitic extraction and EM analysis in Layout EAD. A setup includes a pointer to the process settings required for parasitic extraction and EM checking, as well as user-customizable tool options. When you have specified the options you require, you can use the controls in the Setups section at the bottom of the form to save them in the system for later use.



Process Settings: gpdk090

Temperature: 25 °C

Excluded Nets:

EM Data Source: Virtuoso techfile

Name	Extraction Model
nom	../MBL/gpdk090_v4...al/models/ictfile

Figure 3.3: Process setup form

Corners is where you associate your extraction corners with the ICT file to be used for extraction. The EM Data Source field lets you specify the source of the EM models and rules; either a Virtuoso technology file, an ICT file, or an EM data file.

You use the Layout EAD Options form to define the setups required to drive parasitic extraction and EM analysis in Layout EAD. A setup includes a pointer to the process settings required for parasitic extraction and EM checking, as well as user-customizable tool options. When you have specified the options you require, you can use the controls in the Setups section at the bottom of the form to save them in the system for later use.

We typically define two EAD setups; one for use in EM analysis and one for parasitic extraction to be used in re-simulation.

- For EM analysis, you need extract only R parasitics and include the shapes inside devices and pcells
- For re-simulation, you would typically perform both R and C extraction and device and pcell shapes (because these are already modeled in simulator models)

3.4 Overview of Parasitic Extraction in Layout EAD

3.4.1 Extracting parasitics on interconnect wires between devices

Figure 3.3 given below illustrates the default extraction behavior for wires used to connect devices:

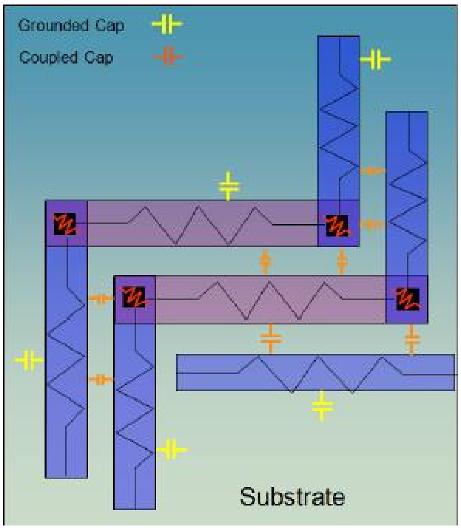


Figure 3.4: Interconnect parasitic extraction

Note the following points:

- Resistance is extracted for all layers defined as conductor layers in the ICT file and for vias/contacts (typically poly, contact, and metal layers and vias). If the layer_type is define as diffusion, resistance is not extracted for that layer.
- Coupling capacitance is extracted between all conductor layers (i.e. Metal1 to Metal1, Metal1 to Metal2, and so on)
- Grounded capacitance is extracted between conductor layers and substrate. During re-simulation the grounded node is specified in the Parasitics – Setup form.

3.4.2 Extracting Parasitics inside Devices

Figure 3.5 below illustrates the default extraction behavior inside devices:

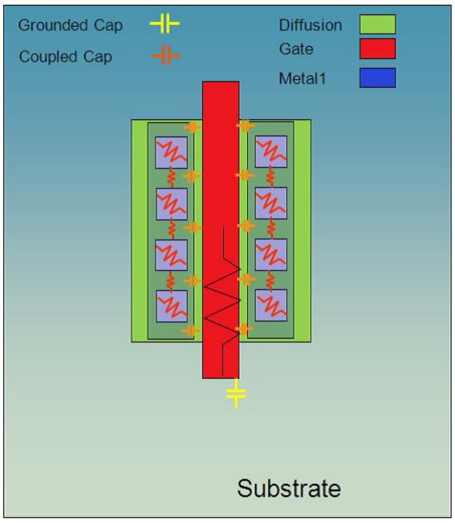


Figure 3.5: Parasitics inside device

Note the following points:

- Resistance is extracted for the terminals on layers specified as conductor in the ICT file and for vias/contacts (typically, metal and contacts, but excluding diffusion)
- Coupling capacitance is extracted between all conductor layers (e.g. Metal1 to Poly, Metal1 to Metal1, and so on) Note: Coupling capacitance is not extracted for
    - gate-to-contact
    - metal-to-contact
    - contact-to-contact
    - gate-to-diffusion fringing cap
- Grounded capacitance is extracted between conductor layers and substrate (poly to substrate). During re-simulation, the grounded node is specified in the Parasitics – Setup form.

3.4.3 Extracting Parasitics for Devices and Interconnect

When extracting parasitics between devices and interconnect wires, the coupling capacitance is extracted between interconnect outside the device and interconnect inside the device, as shown in the figure 3.6

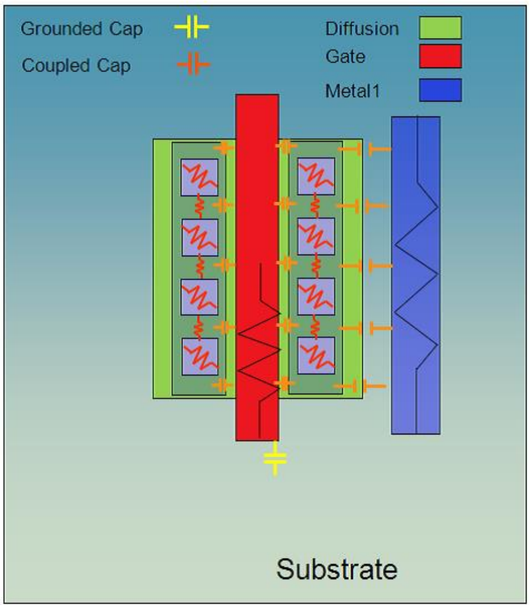


Figure 3.6: Parasitic extraction for device and interconnect

3.4.4 Extracting Parasitics for Devices and Overlapping Interconnect

Figure 3.7 below illustrates the default extraction behavior for devices and overlapping interconnect

Note the following points:

- Coupling capacitance is extracted between interconnect outside the device (defined as conductor in the ICT file) and interconnect inside the device (defined as conductor in ICT file).
- Grounded capacitance is extracted to any shape that has a net, but where the technology file defines no RC extraction on that shape; for example, substrate or diffusion.

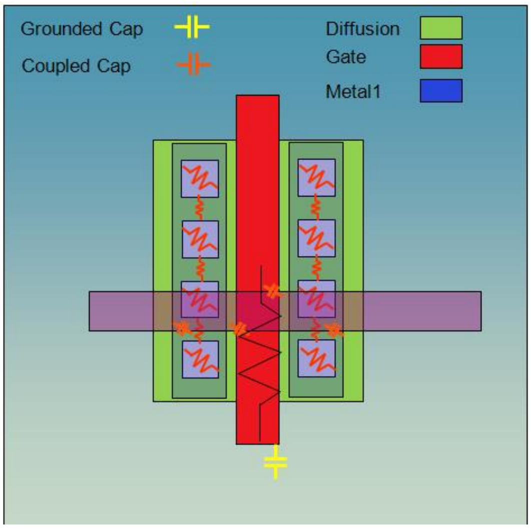


Figure 3.7: parasitic extraction for device and overlapping interconnect

3.4.5 Extracting Parasitics for Local Interconnect Layers

When two shapes on different local interconnect layers overlap and also touch by abutment in the vertical dimension, that is, top-to-bottom, then these shapes are connected by a resistor, as shown in the figure given below.

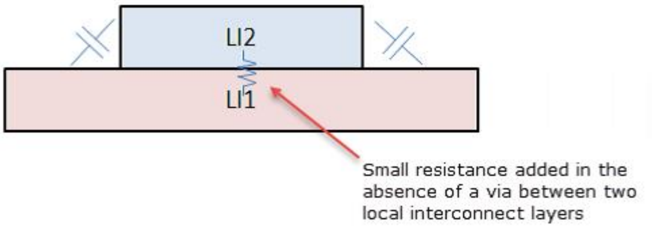

LI2
LI1
Small resistance added in the absence of a via between two local interconnect layers

Figure 3.8: Parasitic extraction for local interconnect layers

Typically, the resistance value between layers is defined by a via definition in the ICT file. However, if there is no via definition, as shown in the case above, EAD uses a small resistance value.

Chapter 4

Test case Development

The Developed EAD flow was put to test on various design projects to get the performance improvement in the layout productivity. This chapter provides detailed information about the different test-case development for EAD flow.

4.1 Layout Design Exercise

Test cases are required for testing the EAD flow. We can take any layout and try to extract the parasitics with EAD and compare it with the extraction from sign-off tool. The accuracy of EAD extracted values rely on the binding of layout instances, nets and pins with those present in schematic, in other words we require virtuoso XL compliance. Most of the legacy layout are not VXL compliant, so we have to create the test case for EAD. There are two ways to approach this problem, we can convert legacy layouts to new VXL compliant layout which can then be used as test case for EAD. The legacy layouts have polygons spread across many level of hierarchy and some of them are make cells which are formed by combining many instances in schematic which are repeated in the design. The conversion of such layouts to make it VXL compliant will be very time consuming. The other option is to come up with small designs which can be made from scratch and used as test case. This also has an advantage that we can see how incremental layout parasitics effects the design TAT.

For these case we will be considering small design which cover most aspects of layout design to serve as a test case for EAD. We will be considering designs like ring oscillators, diff-amps and clock receivers, thus will cover different level of complexities and variety of design sizes. we will observe the change in TAT for the proposed flow with EAD.

4.2 Ring Oscillator Test case

The Ring Oscillator test case consist of inverter chains and a AND gate to act as a switch to enable the circuit. To create a layout for this test case we first start with creating layouts for inverter and AND gate. The instances of these cells will form the top level layout. The layout design started with a completed schematic for std-gates. To maintain VXL compliance we did GFS. Then used Vituoso Layout XL to complete the design. This will ensure that the proper binding between schematic instance and layout instances is maintained.

The layout of the gates was done using Layout EAD. The parasitics for nets in the design were available up-front. This helped in achieving design constraints like drive strength and with parasitic re-simulation we were able to determine the timing performance of the cell. Once all the gate cells were layed out with proper VXL bindings, those cells were instantiated in top level.

The circuit for ring oscillator consist of multiple instances of inverter cells. To improve the layout productivity a layout regeneration utility was used to mimic the placement of cells giving reference to an existing layout. A skill utility was developed to change naming convention of source layout instances. The new naming convention was in accordance with GFS utility. For the placement of cells in the source it was necessary to filter out instances from the rest of the objects in the selected set which may include nets, pins and text. A skill code was developed for filtering out instances from the rest of the selection. Once the placement of the cells is done automated routing utility was used to route the design. The routing utility takes into consideration all the routing constraints set in the constraint manager. Also we can specify any local constraints to the router for a particular run like matching between nets etc.

The top level schematic for a general Ring oscillator is as shown in the figure 4.1. This design consist of cells with fixed height, i.e std-cells. Thus row based placement can be utilised in this case. The row based placement flow ensures that minimum area is utilized for placement of the cells. Row-based placement is a structured placement type that enables you to place specific component types within rows. In addition to providing a structured placement, row-based placement enables you to constrain the components within a row to specific orientations and alignments. Given the uniform and structured placement it helps to obtain, row-based placement is considered a more economical and optimal placement type. The below figure 4.2 shows the row based placement form for specifying different parameters for the placer.

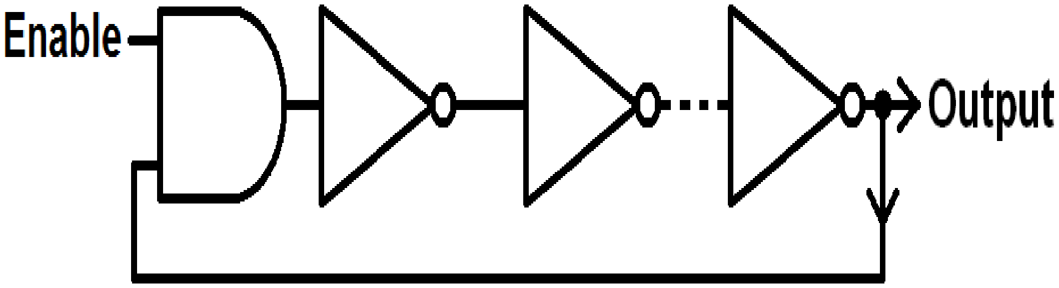
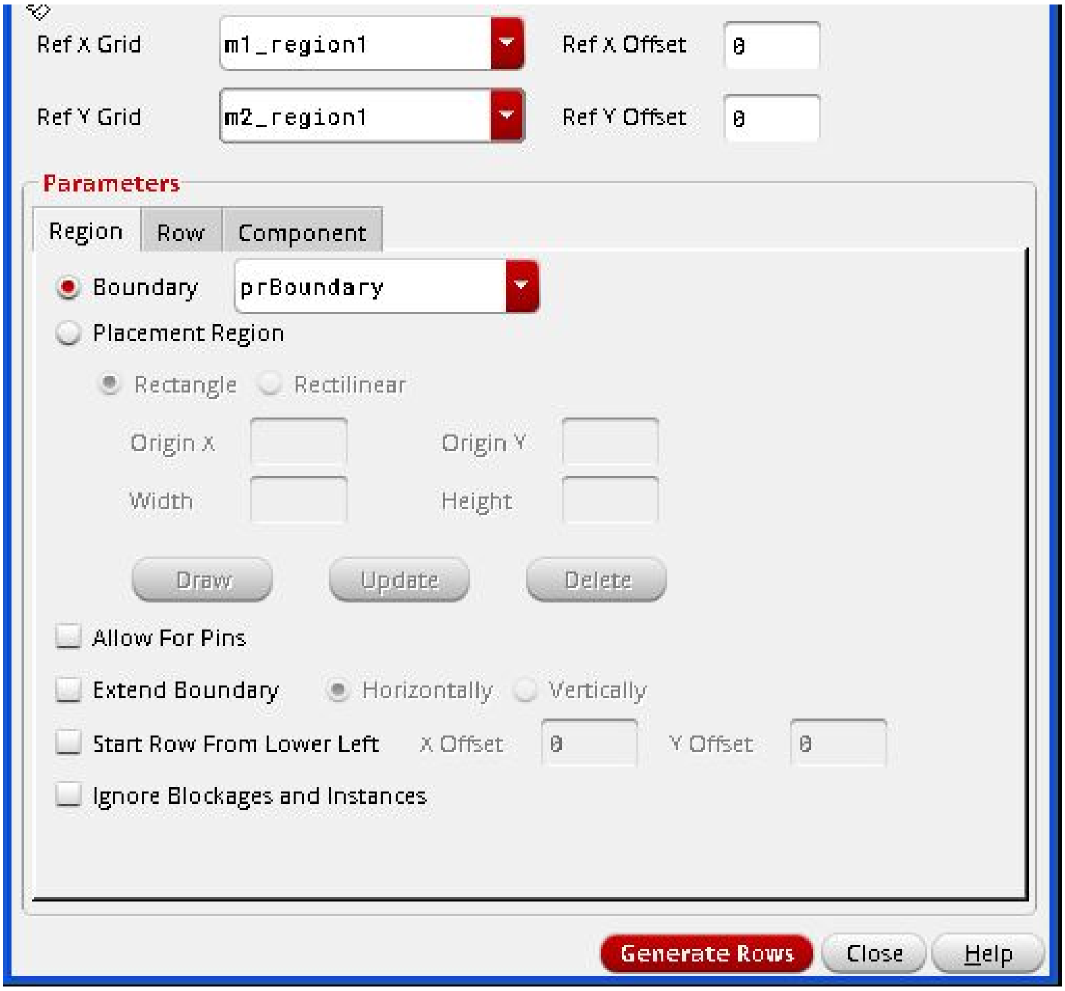


Figure 4.1: Ring Oscillator Circuit

The constraints to be set for placement planning are the reference grids and their offsets if any for row creation, the region were the rows can be created, the type of cells to be placed in the row, and number of rows required. The row height is calculated automatically, either it is the largest size of the cell to be placed or it is the standard height if all cells are have height in multiple of a standard height. The orientation of the cells in a row is decided based on the sharing of power and ground grids if possible. If the power grids are shared which is generally the case, the outcome of row based placement is similar to the one shown in figure 4.3. If all cells are placed in single row the outcome of row based placement is similar to the one shown in figure 4.4.



Ref X Grid: m1_region1 Ref X Offset: 0

Ref Y Grid: m2_region1 Ref Y Offset: 0

Parameters

Region | Row | Component

● Boundary: prBoundary

○ Placement Region

● Rectangle ○ Rectilinear

Origin X Origin Y

Width Height

Draw Update Delete

□ Allow For Pins

□ Extend Boundary ● Horizontally ○ Vertically

□ Start Row From Lower Left X Offset: 0 Y Offset: 0

□ Ignore Blockages and Instances

Generate Rows Close Help

Figure 4.2: Placement Planning Form

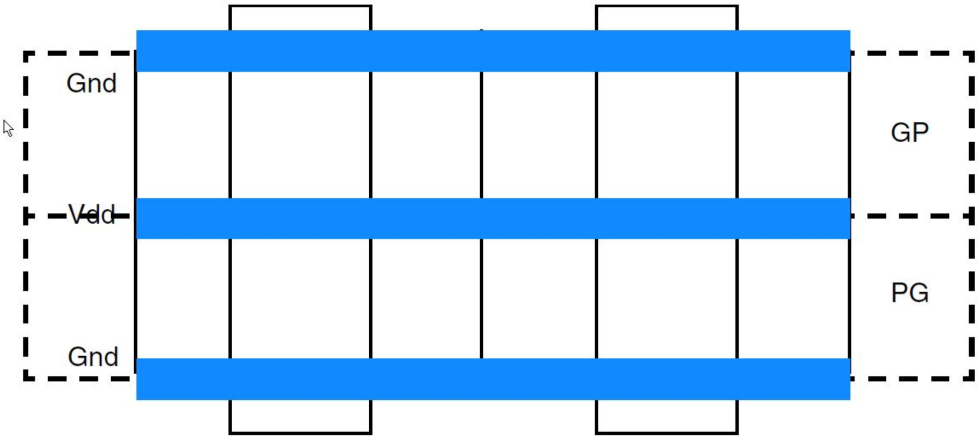


Figure 4.3: Row based placement outcome with shared power rails

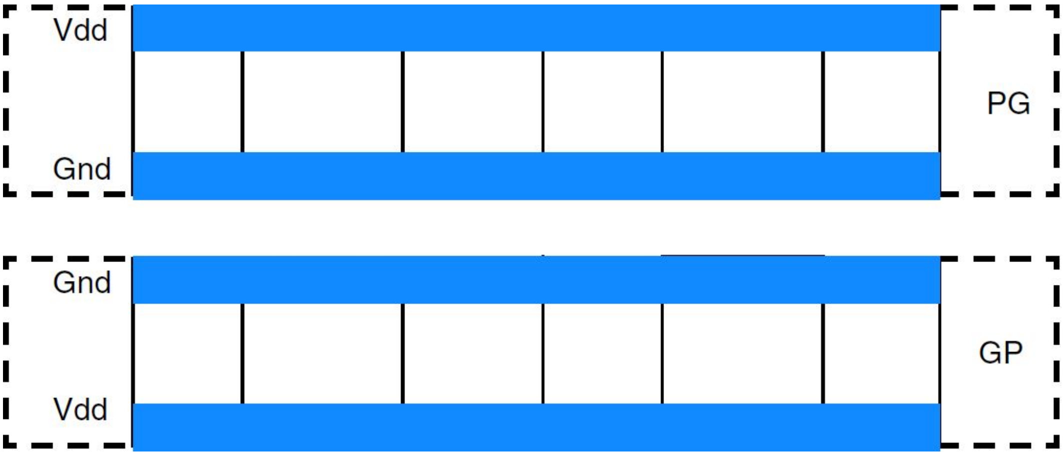


Figure 4.4: Row based placement outcome

Here the number of instances are less but the aspect ratio for the macro size is such that we will need to make two rows and the remaining space is filled with filler cells to maintain continuity in the well structure and to maintain density in base layer. Once the cells are placed they are routed using assisted routing in Virtuoso Layout XL.

4.3 Diff-amp Test case

The Diff-amp test case is completely analog test case. The device matching is key in this test case and also the routing to be used should be matched in terms of its parasitics. The design consists of a current mirror module along with common source differential amplifier. The circuit diagram of the same is shown in figure 4.5

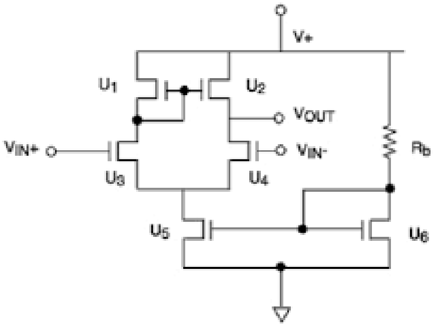
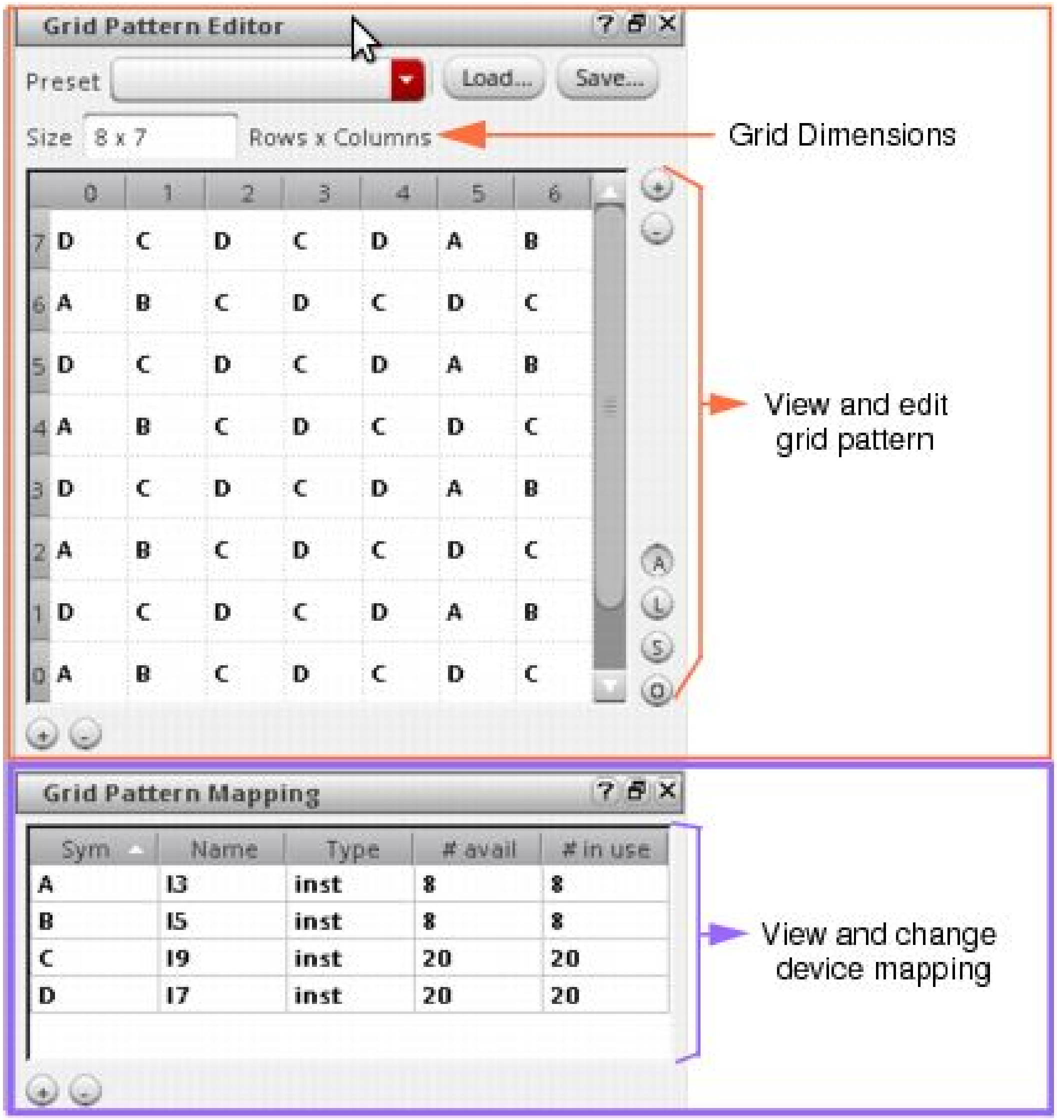


FIGURE 11

Figure 4.5: Diff-Amp test case circuit

The layout design process for this circuit can be made easy by using modgen utility. This creates a structured pattern for component to be placed in a pattern specified by the pattern editor. All instances in a modgen are part of a fig group so once it is created it can be relocated and/or copied multiple time with all the routing and dummies if they are present.

We have utilized the property of grided pattern placement planning of modgen to ease the place and route of this design. We will utilize the common-centroid method of placement to minimize the effect of process variation on different devices. Also we utilize folding of larger width devices to achieve width matching. The pattern for common-centroid device placement can be specified using modgen pattern editor form. The pattern editor form is shown in the figure 4.2.



Preset

Size 8 x 7 Rows x Columns

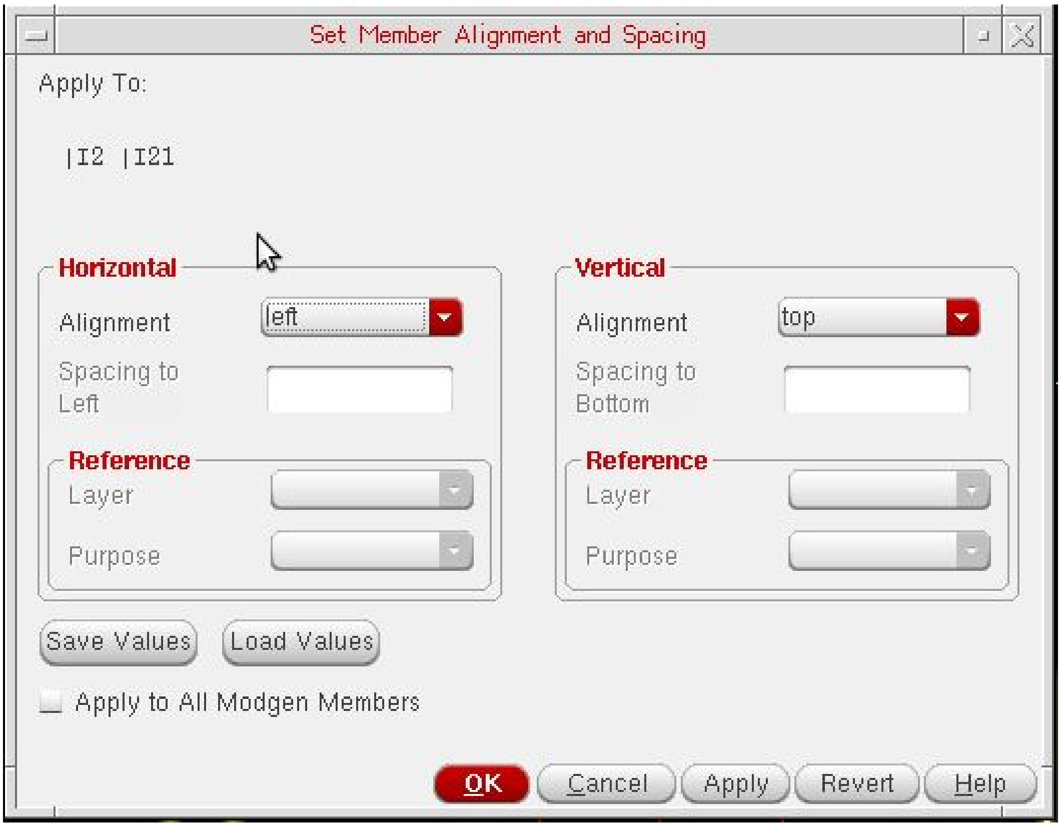
	0	1	2	3	4	5	6
7	D	C	D	C	D	A	B
6	A	B	C	D	C	D	C
5	D	C	D	C	D	A	B
4	A	B	C	D	C	D	C
3	D	C	D	C	D	A	B
2	A	B	C	D	C	D	C
1	D	C	D	C	D	A	B
0	A	B	C	D	C	D	C

Sym	Name	Type	# avail	# in use
A	I3	inst	8	8
B	I5	inst	8	8
C	I9	inst	20	20
D	I7	inst	20	20

Figure 4.6: Grid Pattern Editor for Modgen

As shown in the figure 4.6 the grid pattern editor have options for changing the number of rows and columns in a modgen, the pattern in which the devices are placed in a modgen and the mapping for devices and dummies used in the pattern. we can save a particular pattern and then use it in some other design if required.

Once the devices are placed in their respective locations they are abutted to each other to optimize on area and form a chain of interleaved devices. By default the distance between each instances in a modgen is 1DG. We will need to add space between two rows of devices for routing space. This spacing can be customized by using the member alignment form. The member alignment form is shown in the figure 4.7



Set Member Alignment and Spacing

Apply To:

|I2 |I21

Horizontal

Alignment: left

Spacing to Left:

Vertical

Alignment: top

Spacing to Bottom:

Figure 4.7: Member Alignment Form

Once the devices are placed we route the design taking care of constraints like matching between differential pairs and total cap on output net. The real-time parasitic engine in EAD comes handy to check for net parasitics while routing the design.

Chapter 5

Implementations

The EAD flow is of use only if it can be used with existing automated tools. The chapter provides details about the integration of EAD flow with existing automated tools.

5.1 EAD based Automatic Routing flow

It is important that any new utility addition in the existing eco-system must be integrated with the existing tools. This ensures the usability of that utility. EAD is a value add for mask designers and hence, its integration with existing tools can be of a great value add. Routing takes up most of the mask designers time. To ease their effort there are many automatic tools which helps in placement, routing and automations like automatic via droping, automatic width correction etc. are available. Most of the automated tools takes into consideration physical effects like DRC and also connectivity issues like shorts. This ease the efforts in the physical verification stage.

There are 3 main stages in physical verification

1. Design Rule check
2. Layout vs Schematic check
3. Parasitic extration

As mentioned earlier, current tools supports awareness in DRC and LVS issues, but they are not parasitic aware. EAD can be used to fill in this gap. The extracted values from EAD can be feed in to placement and routing tools as an added dimension to their capabilities. Amongst Place & route, effect of parasitics on routing is more prominent and hence we target automatic routing.

As shown in figure 5.1 the integration of EAD with existing auto-routers is done by creating a wrapper utilizing the auto-router engine. The reference values are added through cadence constraint manager. These values are set by the circuit designer based on the specifications.

The routing will start with the critical nets first, and amongst them the net which has the most stringent constraint to achieve. Also there is a optional text file input which has net routing order priority in it. It is possible to give only critical net routing order in the text file, rest of the nets will be taken automatically based on the constraints. For cases like

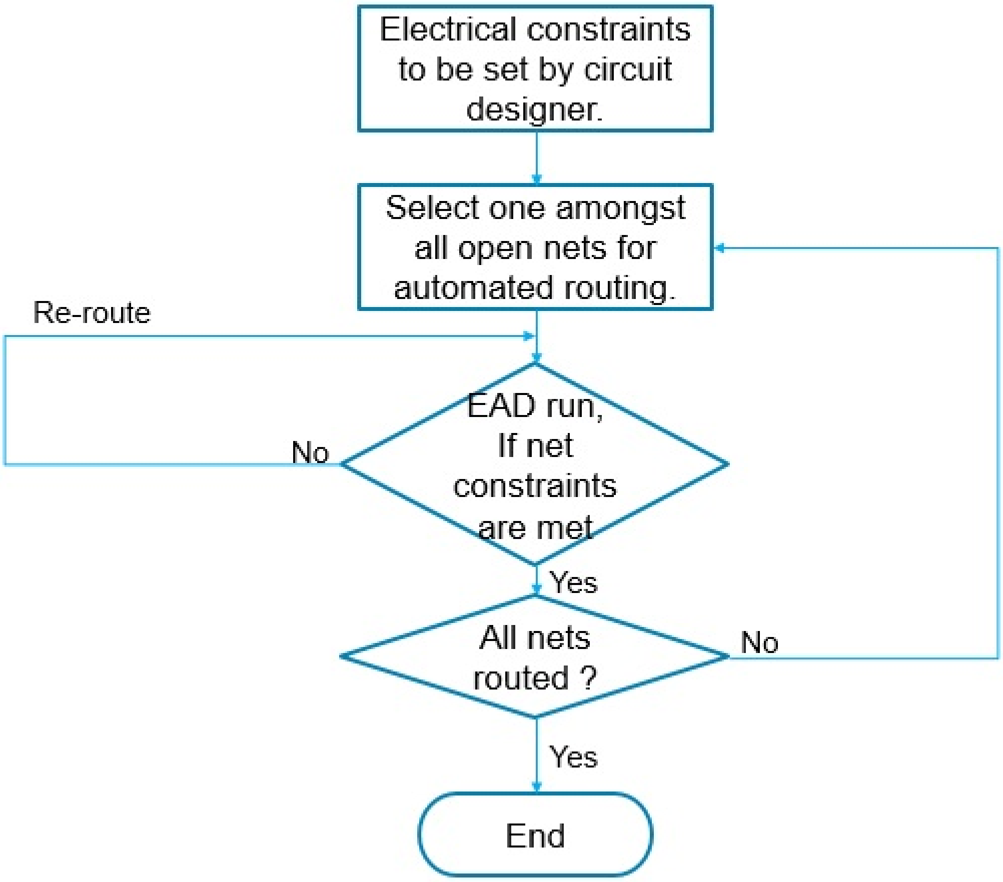

Electrical constraints to be set by circuit designer.
Select one amongst all open nets for automated routing.
Re-route
No
EAD run, If net constraints are met
Yes
All nets routed ?
No
Yes
End

Figure 5.1: EAD based automated Routing flow

match length routing constraint on nets, all the nets in that constraint group will be routed before running EAD engine on them.

There is a provision for guard band region for total cap and resistance of a net. If the values exceed this guard band region it should flag as a warning. The values for guard band region on individual nets can be given with the text file along with the routing priority. This feature is for scalability of the design. for example, the total cap value of a net is the addition of ground cap and coupled cap. The value of coupled cap can change with addition of polygons nearby. This polygons can be from fill,or nearby instance when integrated at top level. The guard band region ensures that there is room for integration of design at top level.

5.2 Real-Time EM Analysis

Electromigration is a general term used to describe failure mechanisms in the metal wires of a chip caused by the movement of metal atoms in a wire because of high current stress. As electrons move through a metal wire, they collide with the atoms in that wire. These collisions cause wires to become heated and, if enough electrons collide with a metal atom over a period of time, the metal atom can move in the direction of the electron flow, potentially creating

- An open circuit if the wire breaks
- A short to an adjacent metal wire if enough atoms move to the same location

EM checking is performed to analyze the power grid and signals of the design for any density violations of the DC and AC current through the metal conductor lines, vias, and contacts. These violations can then be corrected by limiting current density in the affected areas of the design.

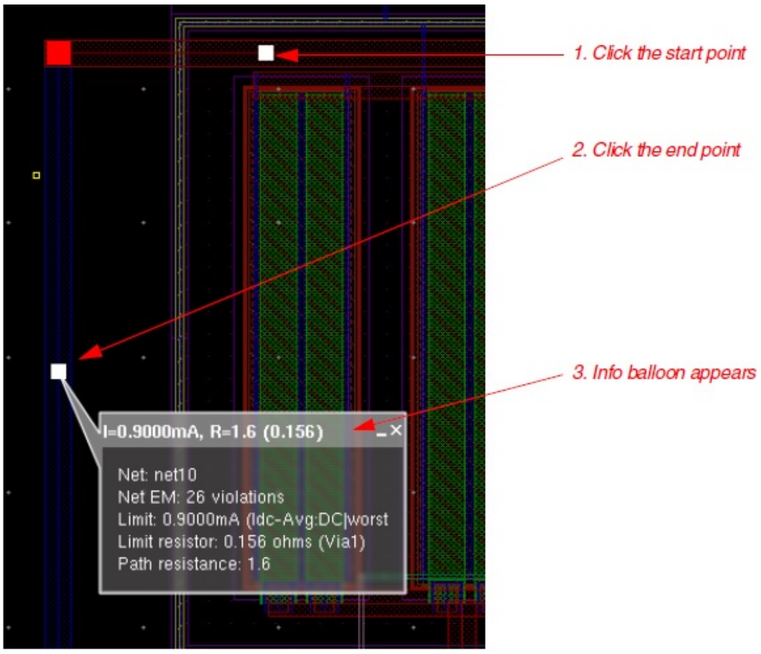
You can use Layout EAD to

- Run EM checks
- Analyze the results and identify violations in the EAD Browser
- Highlight problem areas and display detailed EAD-related information in the layout canvas

You can then edit your design directly in the layout window to address the issues found and rerun the analysis either automatically or on-demand to ensure that the problems were resolved.

Real-time EM capabilities is the second most important feature of EAD. It utilizes the feature of back-annoting the extracted values to simulation thus finding the current densities in the nets. These current values are converted into EM datasets which are fed to EAD browser for EM analysis. The tool reads the maximum allowed current density values from the tech file and flags error once your current density exceeds that value. This is the biggest advantage of EAD which has the most impact on TAT. It is estimated to have around 50% improvement in TAT. This is because the designer don't have to complete the entire design to run EM checks on the layout.

5.3 Point-to-point Info Balloons



1. Click the start point

2. Click the end point

3. Info balloon appears

I=0.9000mA, R=1.6 (0.156)

Net: net10
Net EM: 26 violations
Limit: 0.9000mA (Idc-Avg:DC|worst
Limit resistor: 0.156 ohms (Via1)
Path resistance: 1.6

Figure 5.2: EAD Info Balloon

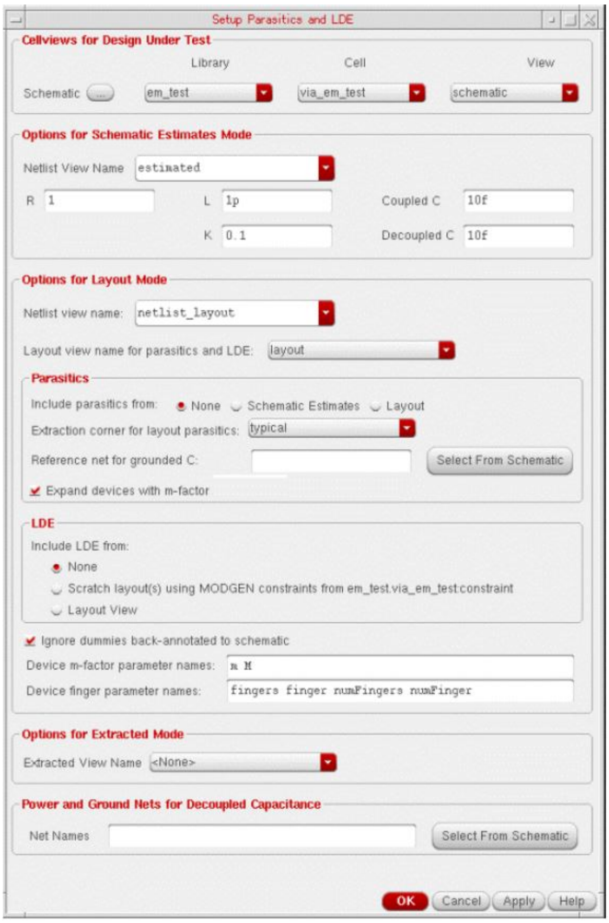
Point-to-point info balloon is an interactive way to view EAD parameters for the segment between any two points in the design. They remain visible on the canvas until they are closed. An info balloon appears showing resistance and EM violation information between the two points. It also visually identifies any limiting resistor with an overlay.

You can edit any segment between the two points specified directly in the canvas. If Automatic Update Mode is set, EAD automatically updates the layout overlay as well as the information shown in the balloon.If Manual Update Mode is set, you must click the Update Parasitics EM button on either of the EAD toolbars to update the display and information shown in the info balloon.This feature helps in interactive debug of EM issues which reduces the time taken to solve EM violations.

5.4 Parasitic Re-simulation

The parasitic information of the design gets updated dynamically while you run electromigration analysis and make changes in the layout to resolve violations. You can use this information, which you can get from a partial or complete layout, to regenerate a netlist and then rerun simulations to check if the output specifications are still met.If required, you can further modify design variables and parameters or the layout to meet the desired results.

A netlist with the name specified in ”Netlist View Name” field is created which contains the parasitic information extracted using EAD. These netlist is included final simulation netlist along with the schematic netlist. The stiching of parasitic information is done auto-



Setup Parasitics and LDE

Cellviews for Design Under Test

Schematic: Library em_test, Cell via_em_test, View schematic

Options for Schematic Estimates Mode

Netlist View Name: estimated

R: 1 L: 1p Coupled C: 10f

K: 0.1 Decoupled C: 10f

Options for Layout Mode

Netlist view name: netlist_layout

Layout view name for parasitics and LDE: layout

Parasitics

Include parasitics from: ● None ○ Schematic Estimates ○ Layout

Extraction corner for layout parasitics: typical

Reference net for grounded C:

☑ Expand devices with m-factor

LDE

Include LDE from:

● None

○ Scratch layout(s) using MODGEN constraints from em_test.via_em_test.constraint

○ Layout View

☑ Ignore dummies back-annotated to schematic

Device m-factor parameter names: m M

Device finger parameter names: fingers finger numFingers numFinger

Options for Extracted Mode

Extracted View Name: <None>

Power and Ground Nets for Decoupled Capacitance

Net Names:

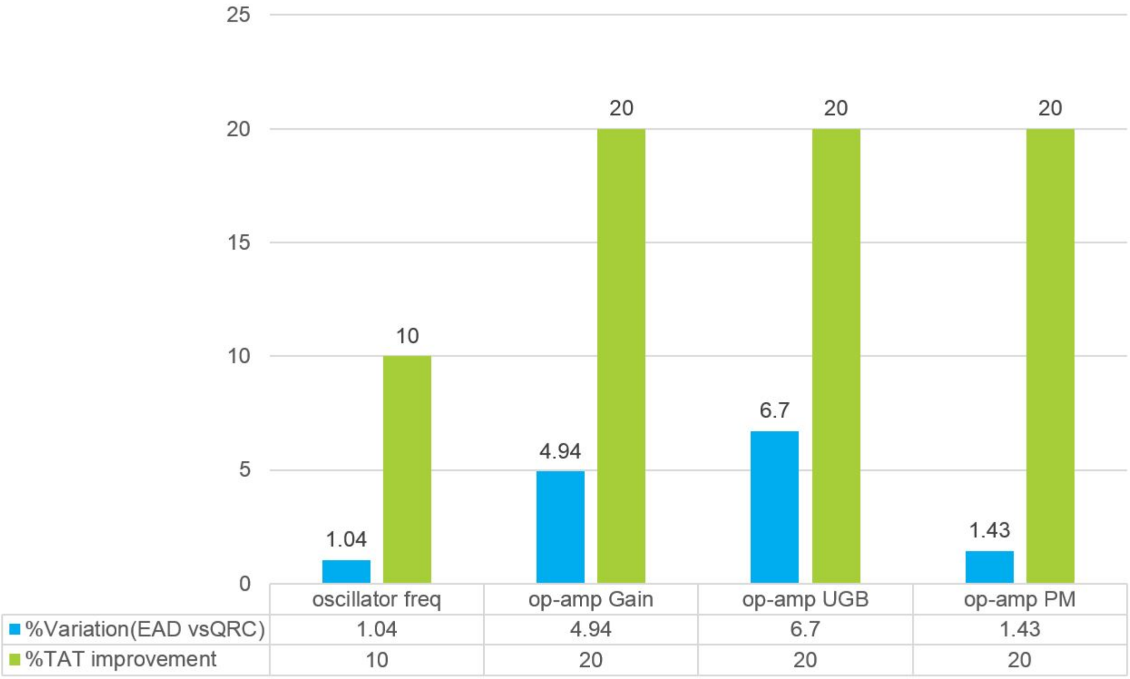
Figure 5.3: Setup Parasitics

matically using the connectivity information from schematic and layout (because they are VXL complaint).

Chapter 6

Results

Implementation of EAD in custom design flow have improved TAT. The % variation in TAT compared to classical custom design flow depends on the many factors including design complexity and size. The chart in figure 6.1 shows the %variation in TAT for two different test cases of different size and complexity and shows overall 15% to 20% gain in TAT with just the extracted values available upfront. The results for EM analysis can improve this number even further. It is expected that with the real-time EM analysis, the time taken to close on specification for a typical RV case would improve by 50%.



	oscillator freq	op-amp Gain	op-amp UGB	op-amp PM
🟦 %Variation(EAD vsQRC)	1.04	4.94	6.7	1.43
🟩 %TAT improvement	10	20	20	20

Figure 6.1: EAD vs QRC comparision and improvement in TAT

The graph in figure 5.1 also shows the % accuracy of EAD extracted values with sign-off extraction tool, in our case it is PVS-QRC. The % variation in extracted values are all within +/-10%.

Chapter 7

Conclusion

The parasitic aware design flow was developed. For examining the extraction accuracy of EAD, test cases were developed and the parasitic aware design flow was utilized in it. The parasitic extracted values were compared with sign-off. With the implementation of parasitic aware design flow in the regular custom design flow we see improvements in the TAT. Also the ability to debug interactively along with designing add great value to the existing design eco-system. The %variation of EAD extracted values are within the acceptable tolerance range for the specified test cases.

Existing automated tools taking advantage of parasitic awareness is a plus. Hence we are looking forward to implement the EAD based automated routing flow.

References

[1] Cadence Design Systems *Virtuoso Electrically Aware Design Flow Guide Product, Version ICADV12.1, August 2014.*

[2] http://community.cadence.com/cadence_blogs_8/b/ii/archive/2011/03/14/how-parasitic-aware-design-improves-custom-analog-productivity

[3] http://community.cadence.com/cadence_blogs_8/b/ii/archive/2013/07/10/virtuoso-electrically-aware-design-ead-a-new-approach-to-custom-analog-layout

Appendix A

EDA Tools used throughout this text

There are various tools used throughout the project. Those tools and a short description of them are covered in the following section.

1. **Cadence Virtuoso**

   - Developed by Cadence Design systems Inc. One of the most popular platforms for custom IC design.
   - It is popular for its easy interface, Library management and rich SKILL API's.
   - It host many utilities for productivity improvement. Modgen, Qbert and EAD to name a few. We can also make our own utilities using the skill interface and add it to the GUI.
   - It is one of the oldest platforms which is still running successfully in the industry, Hence the support infrastructure is huge.

2. **Modgen**

   - Developed by Cadence Design systems Inc. It is part of Cadence Virtuoso Platform.
   - It is a constraint that is applied on one or more instances in a layout or schematic.
   - It helps in creating a regular pattern of device placement through its easy GUI.
   - The most important feature of it begin the ability to generate physical modules of predefined circuits like current mirror and diff-amps from an abstract description.
   - It has support for trunk to pin routing, can manage dummy device addition and abutment and un-abutment of devices

3. **Qbert**

   - It is a Analog/digital block placer in the virtuoso platform.
   - It supports row based placement for std-cells as-well as device placement.
   - It respects the physical constraints like alignment and spacing to name a few.

- It has a wide variety of features like support for substrate contact addition, constraint aware placement, pin placement and planning etc.

4. **EAD**

    - It is a real-time extraction utility for layout productivity improvement in the Virtuoso Platform.
    - It supports both RC coupled and decoupled mode of extraction using the fast extraction mode or a more accurate 3D field solver mode.
    - It can also check for EM violations in the layout on the fly. This helps closing on the Reliability-Verification issues quickly.
    - It works directly on the OA database, so the integration of the tool with the layout editor is seamless.