

ABSTRACT

Varma, Ambrish Kant. Computer - Aided Tools for Seamless High Density Interconnects. (Under the direction of Paul D. Franzon)

This thesis presents the tool-set designed to demonstrate the possibility of using the Cadence tools to design, verify and extract circuitry on the substrate along with the on-chip design. This circuitry could be an inter-chip connection that connects two different chips or an intra-chip connection where a long interconnect is taken off from the active area of the chip to the substrate and back on to the same chip.

To be able to do this task, the work for this project is broadly classified into four different categories. These are writing

- The technology file and the display.drf file
- The Design Rule Check deck
- The Layout Verses Schematic deck
- The Extraction deck

After having completed the above-mentioned tasks, the tool-set was also tested and implemented on a circuit.

COMPUTER – AIDED TOOLS
FOR
SEAMLESS HIGH DENSITY INTERCONNECTS

by
AMBRISH KANT VARMA

A thesis submitted to the Graduate Faculty
of
North Carolina State University
In partial fulfillment of the requirement for the Degree in
Master of Science in Computer Engineering

Department of Electrical and Computer Engineering

Raleigh
2001

Approved by:

Dr. Paul D Franzon
Chairman, Advisory Committee

Dr. Greg Byrd

Dr. Eric Rotenberg

BIOGRAPHY

Ambrish Kant Varma was born in Allahabad, India in the year 1974. He attended his high school at Allahabad and went on to attend the University of Allahabad pursuing the Bachelors of Science program in Statistics. After completing one year in this university, he had the opportunity to go to Sydney, Australia to complete his undergraduate studies in Electrical Engineering. After graduating with honors from the University of Western Sydney, Nepean in 1999, Ambrish decided to take his academic qualification a step further and was admitted to the University of Louisiana at Lafayette in the fall of 1999. He decided to take a transfer to North Carolina State University in spring 2000 where he began work under the guidance of Dr. Paul. D. Franzon. He worked as an intern at the IBM facility at Research Triangle Park, North Carolina in the summer of 2000. His major interest lies in ASIC Design and VLSI.

ACKNOWLEDGEMENTS

There is never a job in the world that can be done single handedly. This work was no exception. Let me begin by naming my advisor, Dr. Paul D Franzon, who was always available for any doubt or question, howsoever elementary or trivial they might be. Without his insight and guidance, needless to say, this project would have never got completed. Also, many thanks to my committee members, Dr. Byrd and Dr. Rotenberg, for their valuable input, encouragement and for reviewing my thesis.

Special thanks to Alan Glaser, who, in spite of being extremely busy with his own research and thesis work, never got irritated with all the terribly silly questions that I hurled at him. There was not one email he didn't reply to and there was not one question he didn't have an answer for. Thank you Alan for all you've done.

Thanks also to Stephen Mick, John Damiano, Pronita Mehrotra and Nishith Rohatgi for helping me out whenever I got stuck. Whether it was using Cadence, programming in Skill or simple printer problems, I didn't have to look far for help. Needless to say, they deserve accolades as much as I do.

Guorong Ma helped me in this project as though his life depended on it. Your sincerity, dedication and hard work were a source of inspiration to me. Thank you for your late nights and for tolerating me.

Thanks to the Department of Electrical and Computer Engineering for EGRC 438. It will always have a special place in my heart and indeed, it will be a sacred place to me for the rest of my life. Thanks also to Michelle Joyner, who never let me worry about anything except my research. Thanks Michelle.

Thanks also to my EGRC 438 labmates for creating such an exciting and relaxed environment to work in. Good luck in your endeavors.

Lastly, my sincerest thanks to my family, who even though far away, never lost hope in me, and to God. Who keeps bailing me out – each time, everywhere.

TABLE OF CONTENTS

List of Tables	vi
List of Figures.....	vii
1 INTRODUCTION AND OVERVIEW	1
1.1 Research objectives and Approach	3
2 BACKGROUND AND LITERATURE REVIEW	5
2.1 SHOCC and MCM.....	6
2.2 Past Work	8
2.2.1 The NCSU Cadence Design Kit (NCSU CDK).....	8
3 THE TECHNOLOGY FILE	10
3.1 Features.....	11
3.2 To Add a Layer	13
3.3 The Display Resource file (display.drf)	15
4 THE DIVA DECK	16
4.1 Design Rule Check.....	16
4.1.1 Key Features of the DRC rule File.....	17
4.1.1.1 Separation Between Same Metal Layer	17
4.1.1.2 Separation Between Two Different Layers	17
4.1.1.3 Metal - Via Enclosure	18
4.1.2 Design Rule Check – A Demonstration.....	19
4.1.3 MCM DRC Rule list.....	23
4.2 Extraction.....	25
4.2.1 Why Transmission Lines?.....	25
4.2.2 Key Features of the Extraction File.....	26
4.2.2.1 The Solder Bump.....	26
4.2.2.2 The Y model of the U element	26
4.2.2.3 The X model of the U element	27
4.2.2.4 Connecting the Pieces	27
4.2.2.5 Calculate Lengths of Transmission Line	28
4.2.3 How Models Are Linked	28
4.3 Layout Vs Schematic.....	30
4.3.1 Key Features of LVS	30

4.3.1.1	permuteDevice.....	30
4.3.1.2	compareDeviceProperty.....	31
4.3.1.3	removeDevice.....	31
4.3.1.4	ignoreTerminal.....	31
4.3.1.5	Series and Parallel Reduction.....	31
4.3.1.6	MOS Reduction.....	32
4.3.2	The Error Files.....	32
5	CASE STUDY – DRIVER RECEIVER CIRCUIT.....	35
5.1	Layout and DRC.....	36
5.2	Extraction.....	38
5.3	LVS.....	40
5.4	Netlist Generation.....	40
5.5	Simulation.....	43
6	CONCLUSION AND FUTURE WORK.....	44
	References.....	48
	Appendix A – The Technology File.....	51
	Appendix B - display.drf file.....	67
	Appendix C - divaDRC.rul file.....	68
	Appendix D - D Extraction Rule File.....	73
	Appendix E - Layer Definition file.....	83
	Appendix F – The HSPICE Netlist.....	95
	Appendix G – Transmission Line Models.....	97
	UMODEL X.....	97
	UMODEL Y.....	97
	BUMP.....	97
	Appendix H – divaLVS.rul file.....	98

List of Tables

TABLE 1: NEW LAYERS FOR SHOCC 11
TABLE 2: MCM DRC RULE LIST 24

List of Figures

FIGURE 1: THE SHOCC STRUCTURE.....	5
FIGURE 2: A HIGH PERFORMANCE SHOCC IC (REFERENCE [7]).....	7
FIGURE 3: LSW WINDOW	12
FIGURE 4: THE SUBSTRATE LAYER STRUCTURE (CROSS SECTION VIEW).....	14
FIGURE 5: THE SUBSTRATE LAYER STRUCTURE (TOP VIEW).....	14
FIGURE 6: DRC SEPARATION (SAME LAYER)	17
FIGURE 7: DRC SEPARATION (DIFFERENT LAYERS).....	17
FIGURE 8: DRC ENCLOSURE COMMAND.....	18
FIGURE 9: FIGURE WITH DRC ERROR.....	19
FIGURE 10: FIGURE WITH ERROR REMOVED.....	19
FIGURE 11: FIGURE WITH ERROR	21
FIGURE 12: FIGURE WITH ERROR REMOVED.....	21
FIGURE 13: THE SHOCC PARADIGM.....	25
FIGURE 14 EXTRACTED BUMP SYMBOL.....	26
FIGURE 15 EXTRACTED UMODEL	27
FIGURE 16 THE BUMP SUBCIRCUIT	29
FIGURE 17: DRIVER AND RECEIVER CIRCUIT.....	35
FIGURE 18: THE DRIVER - INVERTER LAYOUT	36
FIGURE 19: THE EXTRACTED CIRCUIT.....	39
FIGURE 20: SCHEMATIC OF THE DRIVER - INVERTER CIRCUIT	40
FIGURE 21: SIMULATION OF THE DRIVER - RECEIVER CIRCUIT	43

1 INTRODUCTION AND OVERVIEW

“Seamless High Off-Chip connectivity (SHOCC) is a combined packaging, interconnect, and IC design philosophy and associated enabling technology that aims to shift the device fabrication paradigm from today’s single die approach to a parallel manufacturing scheme that utilizes yield-optimized IC elements packaged using a high-performance interconnect element” [1]. The main idea behind the concept is that the interconnects on the chip in a one die system are divided between on-chip and off-chip interconnects. This eliminates the possibility of having long lossy transmission lines on the chip. Consider that interconnects on the chip are aluminum which has a high resistance ($\rho = 4.5\mu\Omega\text{-cm}$). Interconnects off the chip can be fabricated using copper that has low resistance ($\rho = 1.67\mu\Omega\text{-cm}$), hence reducing the RC losses as per the equation below).

$$R = \frac{\rho l}{A}$$

Also, as designers are hard pressed to reduce the size of the chip, the area of the interconnects on the chips are bound to decrease, resulting in an increase in the resistance of the interconnects as can be seen by the above equation. Another

improvement that can be achieved is smaller chip size as a lot of area occupying interconnects is now brought off the chip on the substrate.

Also the number of inputs/outputs that the chip can handle can improve drastically considering that I/O can be done using the solder-pads on the chip that are distributed all over the area of the chip (in a multi-chip module) as compared to a single die where all the inputs/outputs have to be brought out to the perimeter of the chip.

The SHOCC interconnects are modeled as *lossy transmission* line elements. This is done so that the designer can obtain accurate chip-chip delay and cross-talk noise estimates when evaluating the performance of the chip. This gains importance when the rise time of the signals reduces as compared to the flight time of the signal through the chip.

This project is meant to put into practice the idea of SHOCC and to test it using a practical circuit. This thesis aims at documenting the work done in writing the technology files that support the extension of the traditional layout methodology to the level of SHOCC and the rules file that have to be written in order to test and extract the design. Chapter 3 covers the technology file and the display resource file that describes the layers physical appearance. The technology file permits independent or co-layout of the chip, substrate and solder bumps in the Cadence Virtuoso layout package.

Chapter 4 covers the Diva Deck – a set of rule files that allow complete design rule check, extraction of the design and a Layout versus Schematic test that checks the correctness of the extracted design against a schematic provided by the user. The extraction deck permits co-extraction of the IC and the SHOCC interposer, producing a HSPICE file in which on chip circuits are represented by transistors and parasitic capacitance and off chip SHOCC lines are captured as U element transmission lines

Chapter 5 discusses the circuit that has been used to test the technology file and the diva deck. It is a driver – receiver circuit with substrate interconnects. All the design

rules file have been put to test. This chapter also contains the netlist that is generated and the result of the simulations.

Chapter 6 discusses the future work that can be added on to the current work and also serves as a conclusion to this thesis.

The thesis is written in a manner in which a complete newcomer in this field would be able refer to it as a guide to grasp the subject matter and, if required, add to or modify the current work. Each section is accompanied by an extensive explanation of the SKILL and DIVA functions that have been used as well as illustrated with examples.

A webpage has also been designed that includes user manuals, installation instructions, and all the scripts corresponding to this project. The URL of the website is:

http://www.ece.ncsu.edu/cadence/SHOCC_Kit/SHOCC_home.html

1.1 RESEARCH OBJECTIVES AND APPROACH

The main objective of this research is to develop a CAD tool set useful for the co-design and analysis of chips and substrate employing the SHOCC paradigm. The traditional chip design process involves design capture, checking the geometric viability of the design, and extraction of the design to a form that can be understood by a fabrication house such as MOSIS¹. The intention with this project was to have a kit within the Cadence environment that could do all of the above as well as have the capability of utilizing the substrate layers for the placement of long interconnects for intra-chip connections as well as interconnects for inter-chip connections. The designer can choose to only look at the IC design, I/O pad design or substrate design as he pleases. Also the design can be interactively modified as seen fit. As mentioned before, a design rule checker can be invoked to check if the design is fit to be manufactured and all rules are being obeyed. Along with the design rule checking, our objective was also to write a Layout versus Schematic procedure so that a designer

¹ <http://www.mosis.org/>

will be able to check that the final layout is equivalent connection-wise to a supplied schematic.

The design kit was to be developed within the Cadence IC environment to permit co-design and analysis of SHOCC designed chips and substrates. The new technology files and the rules files would be developed by using the SKILL programming language within the Cadence environment, as would be the extraction process that would permit the co-extraction of the on-chip parasitics into a high fidelity HSPICE file.

Performance verification will be conducted by modifying the extraction routines so that the SHOCC interconnects are extracted as transmission lines and are represented as U models in an appropriate HSPICE file.

A few other objectives were decided upon before the start of the project, but these were subject to the availability of time and resources. One of them was to investigate the possibility of having a Post Extraction Filter. This filter would allow us to extract only a certain net or a specific area of the circuit instead of extracting the entire circuit. Another capability that the tool-set could have was the auto-routing feature. Currently connections on the on-chip circuit that are on the same electrical net can be auto-routed. A similar feature can be extended to the on-chip and the off-chip connections that are on the same electrical net.

2 BACKGROUND AND LITERATURE REVIEW

A substantial amount of research has been done in the area of off chip connectivity. Authors have talked in detail about the advent, need, and advantages of this technology [1], [6] and [7] and have gone as far as specifying the implementation of the computer-aided design (CAD) tools that support the SHOCC methodology as one of the long-term objectives of their project [6].

An interesting comparison is made between the performance of SHOCC interconnects with that of typical on-chip interconnects [7]. Figure 1 shows the various elements of the transmission line model of the SHOCC paradigm.

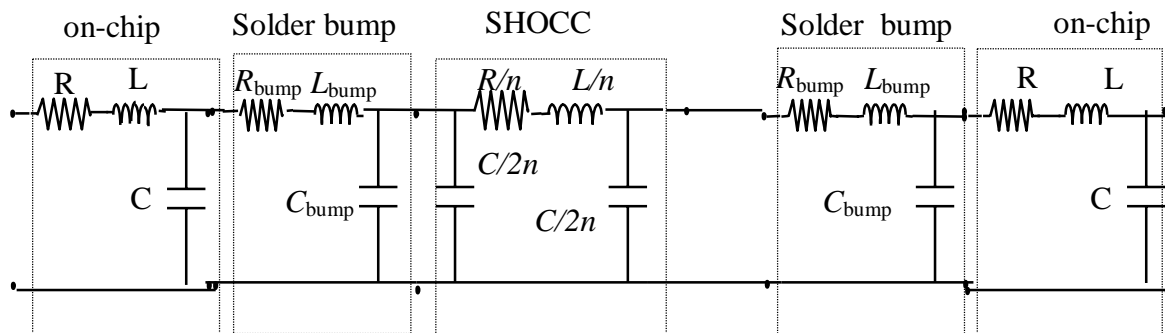


Figure 1: The SHOCC Structure

It clearly shows the way the on-chip and the off-chip circuit elements are modeled. The on-chip elements are represented as lumped L models whereas the off-chip circuit for the SHOCC line is represented by subdividing the line into 16 subsections. Shown in the figure is a π model of one of the subsections.

The authors modeled a driver – receiver circuit using Maxwell tools and extracted R, L and C for the SHOCC and its related structures. Simulation of the circuits was done using Maxwell Spice and PSPICE. Using the R, L and C values, transmission line spice model for the SHOCC structure was prepared. The authors draw several conclusions. An interesting results is that the rise time of the output waveform for on-chip lines is severely degraded as the line length is increased whereas for the SHOCC line there is no degradation. Also the delay in a SHOCC line increases linearly with length, whereas it varies exponentially for the on-chip line. A lot of mathematical results that the authors have derived using simulations in this paper have been used in this research work.

2.1 SHOCC AND MCM

MCMs can be said to be the originator of the idea of SHOCC. Authors talk about how performance and cost advantage can be gained if a chip-set is optimally redesigned to take advantage of the high wire density, fast interconnect details, and high pin counts available in MCM-D / flip-chip technology [6]. However, the authors say there are very few commercial digital system design that have migrated to this technology. The reason is that current design methods optimize chips for single chip packaging – hence the designers under-utilize the potential of MCM-D / flip-chip technology.

This project is a step forward in removing this drawback. We have demonstrated that multi-chip packaging is possible with wider and better substrate interconnects that can be made of copper ($\rho = 1.67\mu\Omega\text{-cm}$) instead of aluminum/copper ($\rho = 4.05\mu\Omega\text{-cm}$). The paper talks about partitioning a large, low-yielding chip to a set of smaller high-yielding chips. Even though the idea of SHOCC was not developed during those days, the authors, nevertheless, were moving in that direction.

Several advantages and other issues are discussed regarding the paradigm shift towards ‘multilithic’ system design as opposed to a monolithic – single chip

packaging. Of particular interest is the reduction in delay between on-chip interconnects and off-chip interconnects. Also, the global power and ground distribution could be moved off-chip, thus saving valuable on chip resources and real estate. Also, the MCM inter-chip interconnects can be built using substantially smaller drivers than those used traditionally for inter-chip signaling.

Even though SHOCC and MCM are fundamentally similar, there are some key differences between these two technologies [1]. The most notable difference is that MCM can be made with traditionally designed chips whereas the SHOCC technology can be implemented using only specifically designed chips. This is because interconnects in SHOCC environment can connect two points on the same chip as well as two points on two different chips using the substrate.

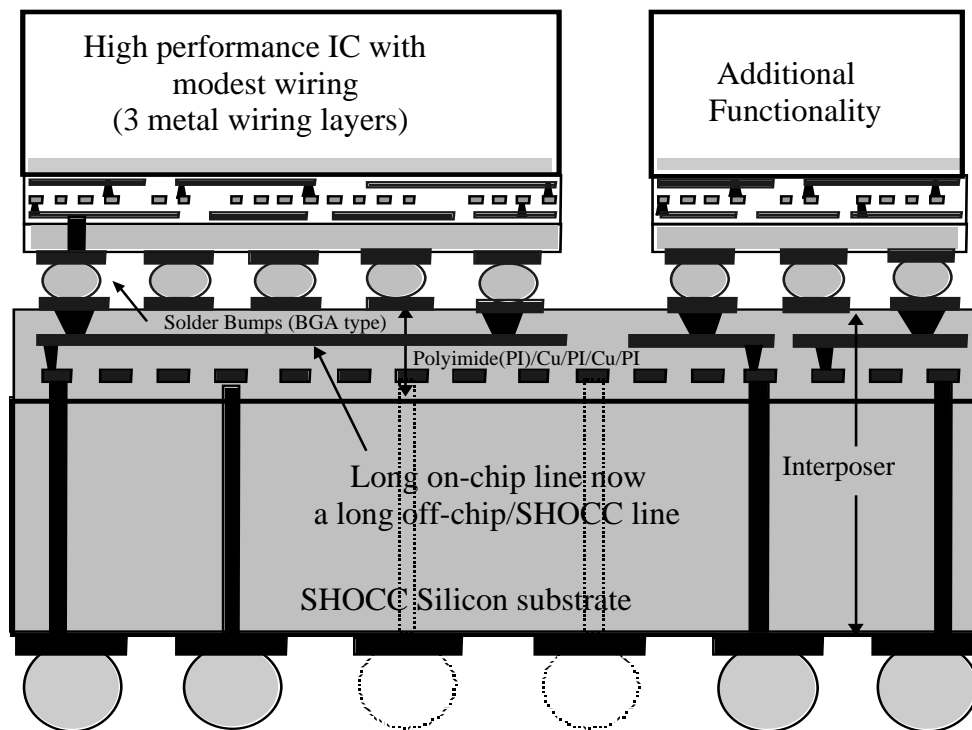


Figure 2: A High performance SHOCC IC (reference [7])

The figure 2 above (taken from [7]) shows a long on-chip line that has been shifted to the substrate. Apart from having intra-chip connections, SHOCC also allows inter-chip connection.

2.2 PAST WORK

A lot of work that proved to be the foundation to this project was done when researchers at NC State University developed the NCSU Cadence Design Kit (CDK). The Cadence Design Kit was put together to support the scalable MOSIS rule-set for IC design within Cadence. The kit and the associated flow tool were developed while research students worked on their research projects. The following section discusses the CDK.

2.2.1 The NCSU Cadence Design Kit (NCSU CDK)

The NCSU CDK is used for teaching and research purposes at NC State University and various other universities. The Cadence Design Kit has been customized with several technology files and a fair amount of skill² code [3]. These files contain information useful for full-custom CMOS IC design via MOSIS. The CDK has been used to design and fabricate working chips [4]. The kit, which can be downloaded freely off the Internet from <http://www.ece.ncsu.edu/cadence/CDK.html>, contains

- Technology files and technology libraries. These files define the masks that are available in different processes as well as the layer available. They also define the value of the lambda for that specific technology.
- Diva Rules files. These files are specifically written for verification.
 - The Design Rule Check (DRC) checks the dimensions, distances and the validity of the geometry of the structures that are built using the layout editor. All rules from the MOSIS SCMOS User's manual are checked. All rules are a function of Lambda - which is different for each process. The value of lambda for each process is stored in the file globaldata.il.
 - The Extraction File extracts FETs, vertical NPNs PN/NP diodes, poly-metall/thin-ox/poly-cap capacitors and parasitic capacitors

- Layout Vs Schematic (LVS) files compares the netlist from the schematic that a designer draws with the netlist that Analog Artist (used for circuit simulation) generates from the layout.
 - Standard Parts Libraries. The standard parts libraries contain common analog and digital parts symbols, Verilog primitives and example sheet borders. A few more complex but commonly used parts such as the multiplexor and the flip-flops are also included in the standard parts libraries.
 - Device Models contains the transistor model files that are obtainable from the MOSIS website.
 - Skill code. A good amount of skill code is used to interface with the Cadence design environment. It includes custom skill code for forms, menus, CDF (component descriptor format) callbacks and parameterized cell definitions.
- A detailed description of the NCSU CDK is also available [3] and [5].

² Skill is a lisp like language.

3 THE TECHNOLOGY FILE

The technology file is the foundation of the entire design. To run a Cadence design session, known as design framework II (DF II) session, we must define technology data in one or more technology files and display resource files. Every DF II design uses a technology library, and several design libraries can share the same technology library. Cadence online help [8], under the chapter, *Technology File and Display Resource File*, provides a deep insight of the technology file, what they contain and how they can be written. The technology file defines the materials and rules we can use in the IC fabrication process. It contains

- Layer Definitions,
- Device Definitions,
- Layer, Physical and Electric Rules and
- Rules specific to individual Cadence applications.

To permit independent or co-layout of the chip, substrate and solder bumps in the Cadence Virtuoso layout package, a new technology file was written. This section of the thesis discusses the structure of the technology file and a few key features, and then goes on to describe how the technology file can be modified if a new layer needs to be added or a feature of the present layer needs to be modified. In the last section,

the display resource file is discussed. The display resource file goes hand in hand with the technology file as it defines the physical properties of the layers that are defined in the technology file. The technology file and the display resource file together tell the design software how to display each layer on a specific display device [8]. The technology file and the display resource file are linked together by the display packet name that is defined in the technology file.

3.1 FEATURES

The technology file written for the co-design and analysis of chips and substrate employing the SHOCC paradigm has a total of five additional layers and four via layers that serve as connection between these layers. New layers that are added to permit co-layout of chips are:

Layer	Description
mcm0	Ground plane (A shape on mcm0 signifies a hole on the layer)
mcm1	Power plane (A shape on mcm0 signifies a hole on the layer)
mcm2	X Metal on the Substrate
mvmV1	Via Between mcm2 and mcm0
mcmV2	Via Between mcm2 and mcm1
mcm3	Y Metal on the Substrate
mcmV3	Via Between mcm2 and mcm3
mcm4	Top pad metal
mcmV4	Via Between mcm3 and mcm4

Table 1: New Layers for SHOCC

Apart from defining new layers, the technology file also lays down rules for proper layout. Classes like **Layer Rules class**, **Physical Rules class** and **Electrical Rules class** define design rules and constraints.

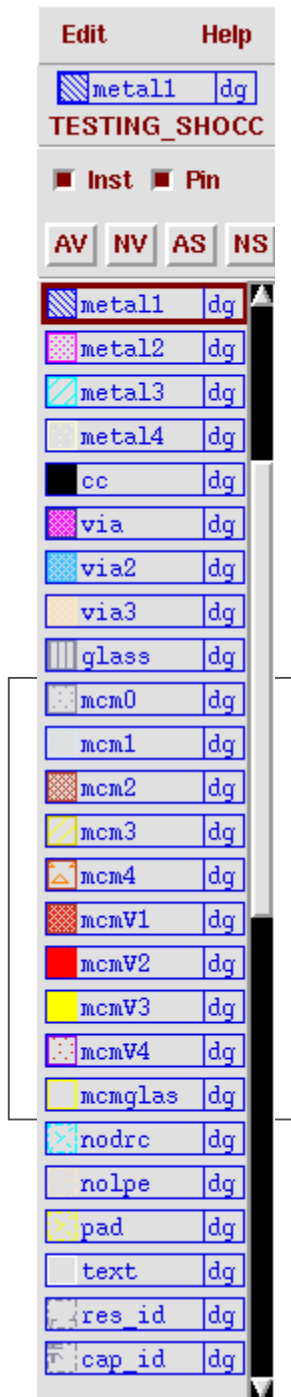


Figure 3 shows the Layer Select Window (LSW) and points out the new layers that have been added to the already present layers.

Layers mcm0 and mcm1 are digitized hole layers – i.e. when extracted, the drawn layers in the layout would actually represent a hole in layers.

The display.drf (display recourse file) is also written to represent all of the layers in the tech file. The technology file and the display resource file together tell the design software how to display each layer on a specific display device. The tech file assigns a display packet, by name, to each layer. The display resource file assigns a display packet definition, with a display packet name, to each display device.

Figure 3: LSW window

3.2 TO ADD A LAYER

- In the layer definition section add the necessary layer name/names, the layer numbers and the abbreviations

For example

```
;( LayerName      Layer#      Abbreviation )
( mcm0            84          mcm0          ) ; mcm layer (ground plane)
( mcm1            85          mcm1          ) ; mcm layer (power plane)
( dummymcm1      94          d_mcm1        ) ; mcm layer
```

- In the techLayerPurposePriorities section, mention the layer and the purposes such as drawing, label, net, pin and boundary.
- In the techDisplays section, each layer – purpose pair must be associated with a packet that is defined in the display resource file (display.drf). Also, the layer-purpose pair must specify the values for five properties that determine their behavior. The properties are ‘Visible’ (sets the objects visible), ‘Selectable’ (sets the objects selectable), ‘Changed Layer’ (enables Diva software tracks changes to objects in incremental verification), ‘Drag’ (enables the objects to be displayed as it moves) and ‘Valid’ (enables the layer-purpose pair to be displayed on the Layer Select Window (LSW)).
- In the **layerRules** class and the subclass **viaLayers**, define the layers that conduct between two other layers.

For example

```
;( layer1      viaLayer      layer2      )
( mcm0        mcmV1         mcm2        )
( mcm1        mcmV2         mcm2        )
( mcm2        mcmV3         mcm3        )`
```

- In the **streamLayers** subclass, list the stream translation data for the layer-purpose pairs.
- The **Physical Rules Class** has three subclasses, the **orderedSpacingRules**, the **spacingRules** and the **mfgGridResolution**. In the **ordereSpacingRules**, the order of layers is important. In this subclass, we specify the **minEnclosure** rule that gives the distance by which an object must be enclosed by another object. **MinEnclosure** is demonstrated by the following piece of code:

```

( minEnclosure "mcm1" "mcmV1" ( 40 ) ) ;
( minEnclosure "mcm2" "mcmV1" ( 20 ) ) ;
( minEnclosure "dummymcm1" "mcmV2" ( 20 ) ) ;

```

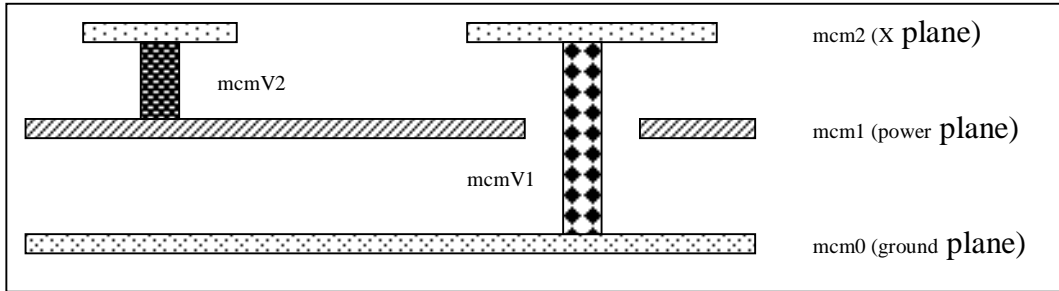


Figure 4: The Substrate Layer Structure (cross section view)

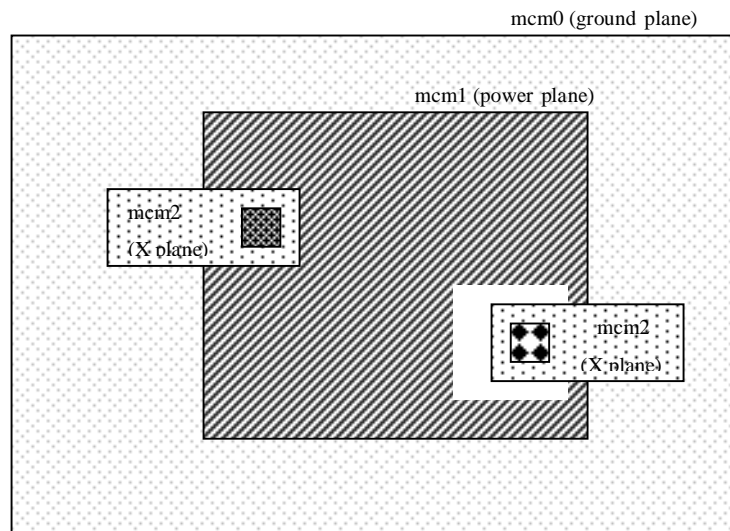


Figure 5: The Substrate Layer Structure (Top view)

As shown in figures 4 and 5, the hole in mcm1, which is the power plane and is digitised hole³ layer, should enclose mcmV1 by 40 microns. mcm2 should enclose mcmV1 by 20 microns. The above diagram shows the cross-section of the three lowest SHOCC layers – the ground plane, the power plane and the X metal plane. It also shows that if a connection needs to be made from mcm2 (X plane) to the ground plane, there should be a hole made in mcm1 for the via to pass through.

- The **spacingRules** subclass specifies the minNotch (minimum distance between the outside facing edges of a notch drawn in an object), minSpacing (distance

³ Digitised hole layer indicates that a drawing on the layer would represent a hole in the final circuit.

between objects drawn on the specified layer) and minWidth (minimum width of a path on the specified layer) values for the layers. Any new layers added would need to have these values specified. The **mfGridResolution** subclass specifies multiple for grid snapping.

- The **Devices** class of the technology file defines the devices we would use with Virtuoso layout. We could also create user-defined devices in the Device class. The main subclasses that we use in this class are **symContactDevice** and **symPinDevice**. The **symContactDevice** subclass of the Devices class declares contact devices. An example of the **symContactDevice** device statement for the layers in SHOCC is:

```
(mcm2_mcm0 mcmV1 drawing
      mcm2 drawing
      mcm0 drawing (mcm1 drawing 20)
      20 20 (1 1 1 1 center center) 20 20 _NA_)
```

The **symPinDevice** subclass of the Devices class declares pin devices.

The final technology file is attached as an appendix at the end of the thesis.

3.3 THE DISPLAY RESOURCE FILE (DISPLAY.DRF)

The display resource file, as described above, groups display data in display packets that it assigns to display devices. A separate display.drf file was created for the various packets associated with the SHOCC layers. We can have multiple display resource files at various locations but each one should be named display.drf.

The following lines show the way display.drf is coded:

```
drDefinePacket(
;( DisplayName   PacketName      Stipple   LineStyle   Fill       Outline)
( display       mcm0             dot4      solid       slate      slate )
( display       mcm1             dot3      solid       silver     silver.)
( display       mcm2             cross     solid       brown     brown )
```

The final display.drf is attached as an appendix to this thesis.

4 THE DIVA DECK

This section of the thesis deals with the Interactive Verification inside Design Automation (DIVA) rule decks. There are, primarily, three rule files that have been created to permit designs that can be fabricated. These rules are Design Rule Checking (DRC), Extraction rules and Layout versus Schematic (LVS) rules. Each of the rules are discussed separately in separate subsections in this chapter.

4.1 DESIGN RULE CHECK

This rule file will permit design rule checking of the substrate and solder bumps against a geometric set of manufacturing design rules. A designer would run the Cadence Diva package to perform these checks. The rules that have been implemented in this project are from MicroModule Systems MCM-D Technology Kit. The kit is provided to support the multi-chip module (MCM) designer to accurately develop an MCM design. The complete DRC rules list is presented in section 4.1.3 of this chapter. The rules file that has been written will only check design rules for the SHOCC layers and not the on chip circuit. For the on chip circuit, previously written rules were used.

Most of the rules that have been included in the divaDRC.rul file have been specified in the MCM Technology kit from MMS[9]. They are originally written in

DRACULA⁴. For this project, these rules were translated to DIVA. Some of the key expressions in both the languages are discussed in the next section.

4.1.1 Key Features of the DRC rule File

Before writing a design rule check file, we need to consider the various scenarios within the design that needs to be checked and verified. Some of them like the width of a path or piece of metal, separation between two like metals, separation between two different metal layers, etc., are fairly easy to understand – but some of the DRC checks are subtle and need some understanding of how the layers are represented.

In the rest of this section, some of the DRC scenarios are presented.

4.1.1.1 Separation Between Same Metal Layer

Separation between the same layer can be checked by the **drc sep** command.

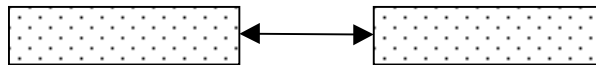


Figure 6: DRC Separation (same layer)

The figure 6 above shows the basic definition of the command.

The example below demonstrates the drc sep command. L25DR2 is the output layer,

The saveDerived command displays the output.

```
L25DR2 = drc(mcm3 sep < 19.40 )
saveDerived(L25DR2 "mcm3 separation < 19.40 ")
```

4.1.1.2 Separation Between Two Different Layers

Separation between two different layers can be done in much the same way except that this time we need to also figure out whether or not the two layers are overlapping.

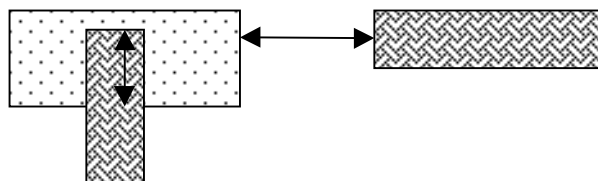


Figure 7: DRC Separation (different layers)

⁴ Dracula is a suite of software products used for verification of integrated circuits – similar to DIVA.

As shown in figure 7, both possibilities must be investigated.

The next piece of code demonstrates how the separation and overlap of two metal layers can be checked by DRC.

```
L22DR4A = drc(mcmV2 mcmV4 sep < 30.00 )
saveDerived(L22DR4A "mcmV2 to mcmV4 separation <30.00")
L22DR4B = geomOverlap(mcmV2 mcmV4)
saveDerived(L22DR4B "mcmV2 and mcmV4 overlapping!!")
```

4.1.1.3 Metal - Via Enclosure

Vias should be completely enclosed by the layers that are been connected by the via. To check for such design errors, we find out by how much is the edge of the via layer enclosed by the edge of the metal layer (done by the **enc** command). If its less than

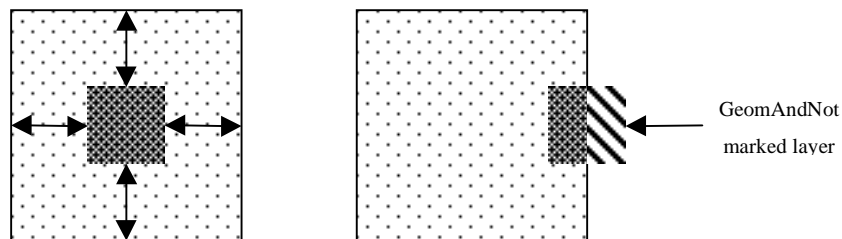


Figure 8: DRC Enclosure command

the specified limit, then an error is registered. We need another command to find out if the via layer is enclosed by the metal layer on all sides. For that, the **geomAndNot** command is used. It marks the area of the via layer that is not overlapped by the metal layer. Refer figure 8.

The code to represent the above mentioned drc is as follows:

```
L23DR3A = drc(mcm2Edge mcmV2Edge enc < 19.4)
saveDerived(L23DR3A "mcm2 and mcmV2 enclosure check (mcmV2 should be
enclosed by mcm2 by 19.4 u on each sides)")
L23DR3B = geomAndNot(mcmV2 mcm2)
saveDerived(L23DR3B "mcm2 and mcmV2 enclosure check (mcmV2 is not
completely covered by mcm2)")
```

4.1.2 Design Rule Check – A Demonstration

In this section, a few screenshots from Cadence Virtuoso program show a few of the DRC errors and the corresponding messages followed by another screenshot that shows the corrected layout.

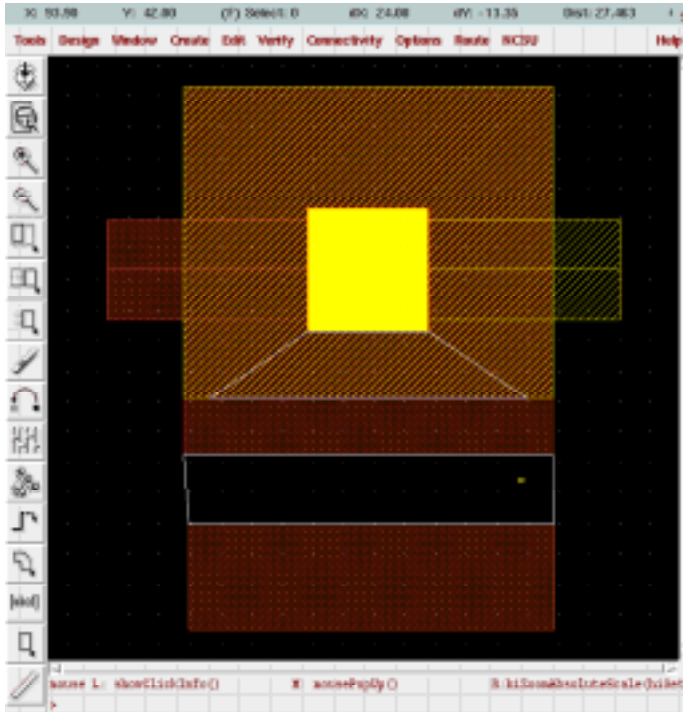


Figure 9: figure with DRC error

- Errors in figure 9
- 1) mcmV3 enclosure check (mcmV3 should be enclosed by mcm3 by 19.4 u on each sides)
 - 2) mcm2 separation < 19.40

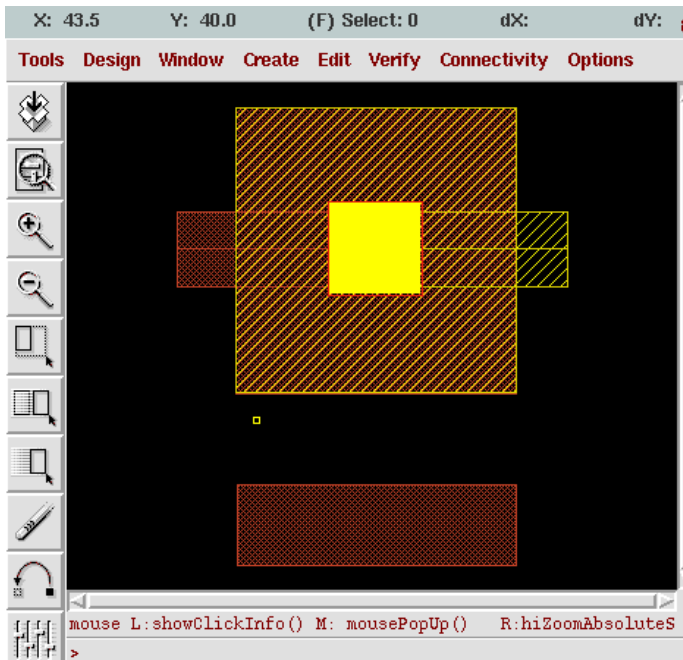


Figure 10: figure with error removed

The entry in the log file for the DRC run for the above figures.

```
DRC started at Wed Dec 20 17:07:49 2000
library: TESTING_SHOCC
cell:    DRC_TEST2
view:    layout
Rules come from library TESTING_SHOCC.
Rules path is divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Dec 20 17:07:49 2000
  completed ....Wed Dec 20 17:07:55 2000
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
**** Summary of rule violation for cell "DRC_TEST2 layout" ****
# errors  Violated Rules
   1  mcm3 and mcmV3 enclosure check (mcmV3 should be enclosed by...
   1  mcm2 seperation < 19.40
   2  Total errors found
```

The entry in the log file for the DRC run for the above figures – after the error was removed

```
DRC started at Wed Dec 20 17:11:00 2000
library: TESTING_SHOCC
cell:    DRC_TEST2
view:    layout
Rules come from library TESTING_SHOCC.
Rules path is divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Dec 20 17:11:00 2000
  completed ....Wed Dec 20 17:11:06 2000
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
**** Summary of rule violation for cell "DRC_TEST2 layout" ****
  Total errors found: 0
```

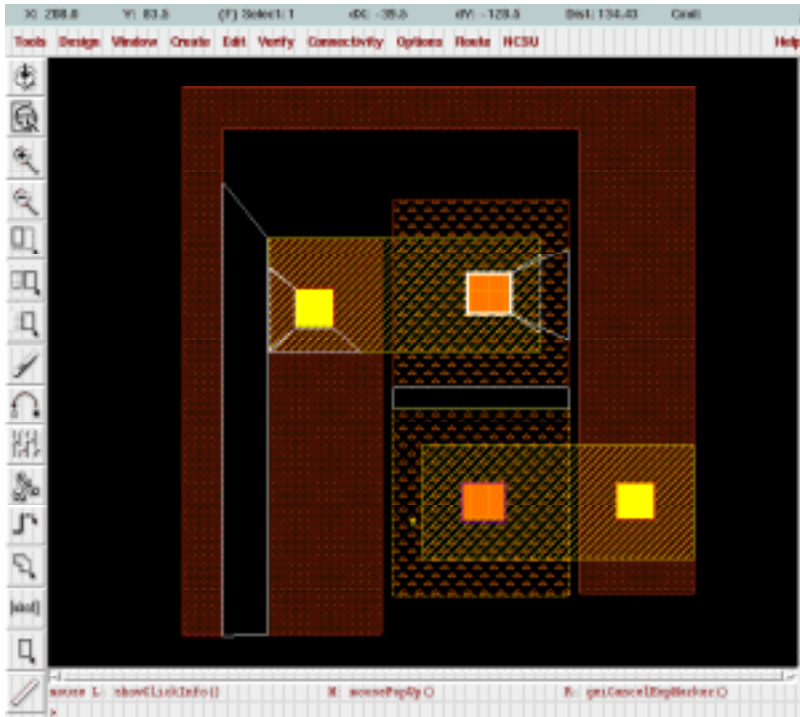


Figure 11: figure with error

Figure 11 has the following errors

1. mcm3 and mcmV3 enclosure check (mcmV3 should be enclosed by mcm3 by 19.4 u on each sides)
2. mcm2 and mcmV3 enclosure check (mcmV3 should be enclosed by mcm2 by 19.4 u on each sides)
3. mcm2 and mcmV3 enclosure check (mcmV3 should be enclosed by mcm2 by 19.4 u on each sides)

4. mcm4 and mcmV4 enclosure check (mcmV4 should be enclosed by mcm4 by 35.4 u on each sides).
5. mcm3 and mcmV4 enclosure check (mcmV4 should be enclosed by mcm3 by 19.4 u on each sides).
6. mcm2 sep < 40.0

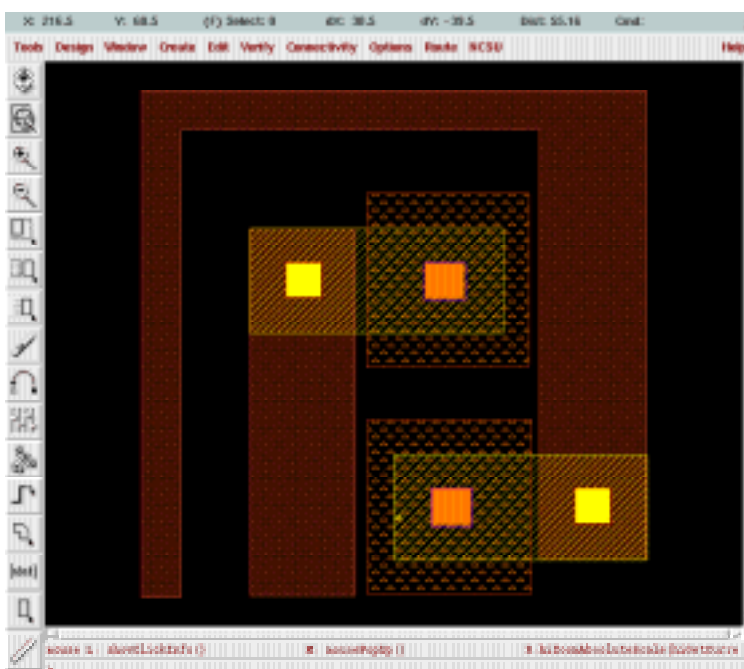


Figure 12: figure with error removed

The entry in the log file for the DRC run for the above figures.

```
DRC started at Wed Dec 20 17:28:37 2000
library: TESTING_SHOCC
cell:    DRC_TEST
view:    layout
Rules come from library TESTING_SHOCC.
Rules path is divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Dec 20 17:28:37 2000
  completed ....Wed Dec 20 17:28:43 2000
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
***** Summary of rule violation for cell "DRC_TEST layout" *****
# errors          Violated Rules
  1  mcm3 and mcmV4 enclosure check (mcmV4 should be enclosed by...
  2  mcm3 and mcmV3 enclosure check (mcmV3 should be enclosed by...
  1  mcm4 seperation < 20.0
  1  mcm2 and mcmV3 enclosure check (mcmV3 should be enclosed by...
  1  mcm4 and mcmV4 enclosure check (mcmV4 should be enclosed by...
      6 Total errors found
```

The entry in the log file for the DRC run for the above figures – after the error was removed

```
DRC started at Wed Dec 20 17:32:26 2000
library: TESTING_SHOCC
cell:    DRC_TEST
view:    layout
Rules come from library TESTING_SHOCC.
Rules path is divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Dec 20 17:32:26 2000
  completed ....Wed Dec 20 17:32:32 2000
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
***** Summary of rule violation for cell "DRC_TEST layout" *****
Total errors found: 0
```

4.1.3 MCM DRC Rule list

This table is a collection of all the DRC rules that have been implemented in the divaDRC.rul file for the SHOCC – co-layout package.

19.0 Ground Plane	mcm0
19DR1	mcm0 separation < 50.00
19DR2	mcm0 width < 25.00
19DR3A	mcmV1 to mcm0 separation <30.00
19DR3B	mcmV1 and mcm0 overlapping
19DR4A	mcmV2 to mcm0 separation <30.00
19DR4B	mcmV2 and mcm0 overlapping
19DR5A	mcmV3 to mcm0 separation <30.00
19DR5B	mcmV3 and mcm0 overlapping
19DR6A	mcmV4 to mcm0 separation <30.00
19DR6B	mcmV4 and mcm0 overlapping
20.0 Via1	mcmV1
20DR1	mcmV1 separation < 80.00
20DR2A	mcmV1 to mcmV2 separation <30.00
20DR2B	mcmV1 and mcmV2 overlapping
20DR3	mcmV1 width < 19.60
20DR4A	mcmV1 to mcmV3 separation <30.00
20DR4B	mcmV1 and mcmV3 overlapping
20DR5A	mcmV1 to mcmV4 separation <30.00
20DR5B	mcmV1 and mcmV4 overlapping
21.0 Power plane	mcm1
21DR1	mcml separation < 50.00
21DR2	mcml width < 25.0
21DR3A	mcml and mcmV1 enclosure check (mcmV1 should be enclosed by mcml by 30 u on each sides)
21DR3B	mcml and mcmV1 enclosure check (mcmV1 is not completely covered by mcml)
21DR4A	mcmV2 to mcml separation <30.00
21DR4B	mcmV2 and mcml overlapping
21DR5A	mcmV3 to mcml separation <30.00
21DR5B	mcmV3 and mcml overlapping
21DR6A	mcmV4 to mcml separation <30.00
21DR6B	mcmV4 and mcml overlapping
22.0 Via2	mcmV2
22DR1	McmV2 separation < 80.00
22DR2A	McmV2 to mcmV3 separation <30.00
22DR2B	mcmV2 and mcmV3 overlapping
22DR3	mcmV2 width < 19.6
22DR4A	mcmV2 to mcmV4 separation <30.00

22DR4B	mcmV2 and mcmV4 overlapping
23.0 X Metal	Mcm2
L23DR1	mcm2 separation < 40.0 and Error Flags length > 600.0
23DR2	mcm2 separation < 19.40
23DR3A	mcm2 and mcmV2 enclosure check (mcmV2 should be enclosed by mcm2 by 19.4 u on each sides)
23DR3A	mcm2 and mcmV2 enclosure check (mcmV2 is not completely covered by mcm2)
23DR4	mcm2 width < 16.0
23DR5A	mcm2 and mcmV3 enclosure check (mcmV3 should be enclosed by mcm2 by 19.4 u on each sides)
23DR5B	mcm2 and mcmV3 enclosure check (mcmV3 is not completely covered by mcm2)
24.0 Via3	mcmV3
24DR1	McmV3 separation < 50.00
24DR2A	McmV3 to mcmV4 separation <30.00
24DR2B	mcmV3 and mcmV4 overlapping
24DR3	mcmV3 width < 19.6
25.0 Y Metal	mcm3
25DR1	mcm3 sep < 40.0 and Error Flags length > 450.0
25DR2	mcm3 separation < 19.40
25DR3A	mcm3 and mcmV3 enclosure check (mcmV3 should be enclosed by mcm3 by 19.4 u on each sides)
25DR3B	mcm3 and mcmV3 enclosure check (mcmV3 is not completely covered by mcm3)
25DR4	mcm3 width < 16.0
25DR5A	mcm3 and mcmV4 enclosure check (mcmV4 should be enclosed by mcm3 by 19.4 u on each sides)
25DR5B	mcm3 and mcmV4 enclosure check (mcmV4 is not completely covered by mcm3)
26.0 Via4	mcmV4
26DR1	mcmV4 separation < 50.00
26DR2	mcmV4 width < 19.6
26DR3A	mcm2 and mcmV4 enclosure check (mcmV4 should be enclosed by mcm2 by 19.4 u on each sides)
26DR3B	mcm2 and mcmV4 enclosure check (mcmV4 is not completely covered by mcm2)
27.0 Top Pad Metal	mcm4
27DR1	mcm4 separation < 20.0
27DR2A	mcm4 and mcmV4 enclosure check (mcmV4 should be enclosed by mcm4 by 35.4 u on each sides)
27DR2B	mcm4 and mcmV4 enclosure check (mcmV4 is not completely covered by mcm4)
27DR3	mcm4 width < 50.0

Table 2: MCM DRC Rule List

4.2 EXTRACTION

The process of extraction for the SHOCC layers involves understanding of how the SHOCC layers have been modeled. As mentioned in the introduction chapter, the SHOCC lines are modeled as transmission lines. Why we consider long off-chip interconnects as transmission lines is discussed in the next section. Signal from the on-chip interconnect is passed on to the substrate through a solder pad. These solder pads are distributed uniformly throughout the chip surface (There are approximately 13 bump locations in a 0.2 mm radius [7].) The pad layer in SHOCC is represented by mcm4. As such, whenever metal3 and mcm4 overlap, a solder bump is placed in the extracted view. Once on the substrate, we have 2 metal layers to propagate the signal - X route (mcm2) and Y route (mcm3). The two layers are modeled as U elements (lossy transmission lines). The X and Y models are named UmodelX and UmodelY in the extraction view.

After traversing the substrate interconnects, the signal hits another solder bump before it jumps back on the chip. This could be the same chip (intra-chip) or a different chip (inter-chip for MCMs). The following figure illustrates the SHOCC structure.

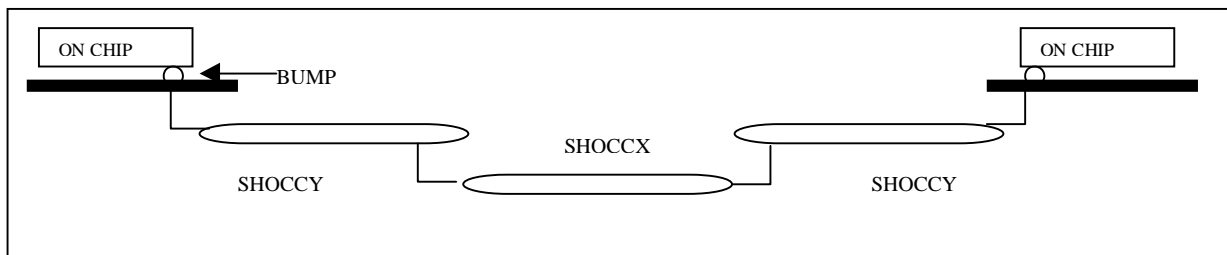


Figure 13: The SHOCC Paradigm

4.2.1 Why Transmission Lines?

To obtain accurate chip to chip delays and cross-talk noise estimates, interconnects, bonding wires and pins should be modeled as transmission lines [10]. This is because if the interconnect are sufficiently long (its inductance becomes larger) or the circuits

are sufficiently fast such that the rise time of the waveform is comparable to the time of flight across the line (resulting in a larger $L \frac{di}{dt}$ and also more crosstalk), the inductance of the circuit interconnects also becomes important. In these circumstances, both the distributed inductance and capacitance must be taken into account. It important to take the inductance into account as it is the cause of the reverse electromotive force that limits the amount of current that can be applied to the circuit.

4.2.2 Key Features of the Extraction File

To place the desired components such as the solder bumps and transmission lines in the extracted view, the divaEXT.rul file was written. This file extracts the on-chip components such as the transistors and capacitors as well as the off-chip components like the solder bump and the U element transmission line. Here is a brief description of how the off-chip devices are extracted.

4.2.2.1 The Solder Bump.

The bump is extracted using the following code:

```
extractDevice(bump metal3("IN") mcm4("OUT") "bump ivpcell TEST" physical)
```

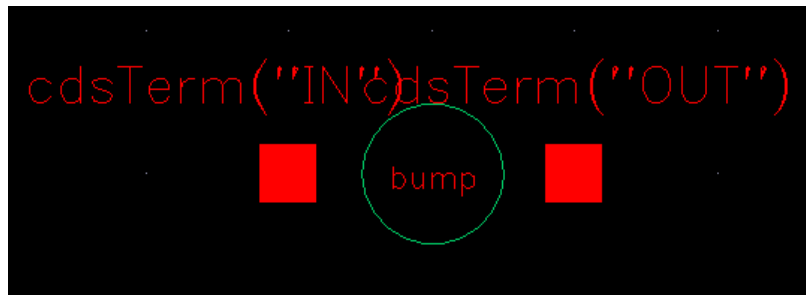


Figure 14 Extracted BUMP Symbol

Figure 14 shows the extracted view of the BUMP. The R, L and C values for the bump sub-circuit are given in the section 4.2.3.

The input to the bump is metal 3 and the output is mcm4. The ivpcell cell-view of bump is looked up in the cell TEST to be placed in the extracted view.

4.2.2.2 The Y model of the U element

The transmission line model for the Y layer is extracted using the following code:

```
extractDevice( shoccY mcmV4("in") pBulk("refin") groundY("refout") shoccXY
```

```
terminal("out") "ulwireSY ivpcell TEST" physical)
```

The input of this device is mcmV4 and the output is shoccXYterminal.

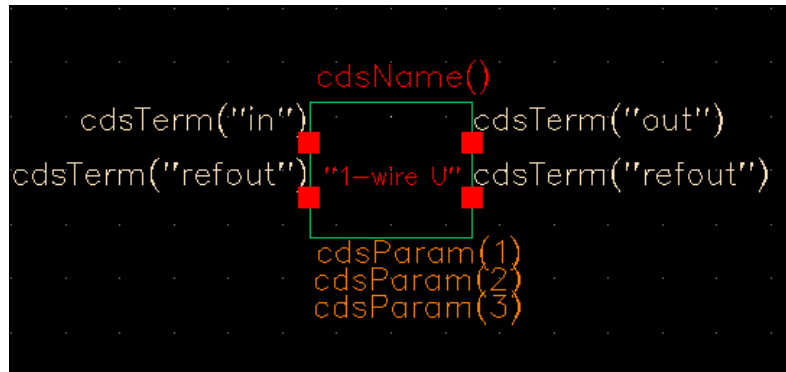


Figure 15 Extracted UModel

4.2.2.3 The X model of the U element

The transmission line model for the X layer is extracted using the following code:

```
ExtractDevice{shoccX shoccXYterminal("in") pBulk("refin") groundX("refout")
) shoccXYterminal("out") "ulwireSX ivpcell TEST" physical)
```

where shoccXYterminal is a derived layer of mcmV3. This layer is the input and the output to the extracted U model. Again, the ivpcell cell-view from library TEST is placed in the extracted view. The R, L and C values of the X and Y models are given in the section 4.2.3.

4.2.2.4 Connecting the Pieces

Now that the bump and the transmission lines have been extracted, the only issue left is to connect these. **GeomConnect** statements are used to join the extracted devices.

An example of the geomConnect statement is:

```
geomconnect(via(mcm4ViamcmV4 mcm4 mcmV4)
            via(shoccXYterminal mcm3 mcmV3)
            )
```

where the vias mcm4ViamcmV4 and shoccXYterminal connect the layers mcm4 and mcmV4 and mcm3 and mcmV3 respectively.

4.2.2.5 Calculate Lengths of Transmission Line

To get the corresponding resistance, capacitance and inductance of the transmission line elements, the lengths of the elements needs to be determined. The following code determines the length of the shoccY and saves it in shoccLengthY using the saveParameter skill function

```
shoccLengthY=measureParameter( perimeter shoccY 0.5e-6)
saveParameter( shoccLengthY "1")
```

4.2.3 How Models Are Linked

When a SHOCC line is identified by the extractDevice command, the cell view U1wireSX ivpcell from library TEST will be put in the extracted view. When a netlist is desired using HSPICE simulator, Analog Environment looks for the U1wireSX HSPICE cell view in library TEST. From there the Component Description Format (CDF) information HSPICE is extracted where the umodelX is specified as the model name linked to the X line. Analog Environment will look for the file umodelx.m in the path specified while performing setup. If SpectreS were used for simulation purposes, we would need to have the SpectreS view.

The umodelx.m file consists of:

```
.lib umodelX

.MODEL umodelX U level=3
+plev=1
+elev=2
+r11=7.6e2
+cr1=150e-12
+l11=508e-9

.endl umodelX
```

where the Level = 3 selects the lossy transmission line model. Elev = 2 selects the pre-computed parameters that allow specification of up to five signal conductors and a reference conductor. The conductor that we are using here are resistance of the line per unit length (r11), capacitance of the line with reference to the reference plane per unit length, (cr1) and the self inductance of the line per unit length (l11). The values that we are using here have been picked up from [7] where the authors had modeled

SHOCC interconnects and used MAXWELL Quick 3-D parameter extractor to get the R, L and C values.

Bumps are linked to the macro **SUBCKT** in the CDF and a circuit with the R, L and C values as of those specified in the CDF is created in the extracted view.

The circuit looks like the figure 16 below:

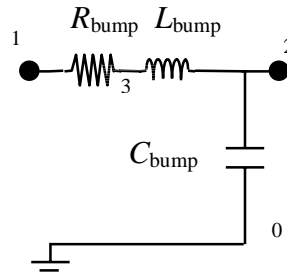


Figure 16 The Bump Subcircuit

The SUBCKT in HSPICE is as follows:

```
.SUBCKT &1 1 2  
  
Rbump 1 3 &2  
Lbump 3 2 &3  
Cbump 2 0 &4  
  
.ENDS &1
```

The rest of the circuit connects the sub-circuit by nodes 1 and 2. The Resistance is placed between nodes 1 and 3, the Inductance between nodes 3 and 2 and capacitance between nodes 0 and 2. The R_{bump} , L_{bump} and C_{bump} values that have been used for the bump sub-circuit are taken from [7]. These values have been extracted using MAXWELL quick 3-D. A more descriptive explanation on how sub-circuits and transmission line wire models are chosen is provided in the HSPICE manual. The complete extraction rule file is attached as an appendix.

4.3 LAYOUT VS SCHEMATIC

The Layout versus Schematic (LVS) program compares two versions of a circuit and isolates any differences. We can use it to compare two layouts, two schematics, or a layout and a schematic [8]. For our purpose, we use LVS to compare the extracted version of the layout and the schematic as drawn by the designer or as provided by a customer or a third party.

It compares the netlist from the extracted view of the layout with the netlist from the schematic that has been drawn that represents the layout. The divaLVS rule file is incorporated with the existing tech files and the diva deck so that simultaneous design and verification can be performed.

In the SHOCC process, the transmission lines and the bumps are removed for the LVS process and are replaced by shorts. This is done because the transmission lines and the solder bumps do not have any effect on the circuit and are just conductors for the purpose of a net-list match between schematic and layout.

4.3.1 Key Features of LVS

Some of the LVS features that have been utilized in this project are described in this section. More extensive explanations can be found in the *Assura Diva Verification Reference* chapter of the Cadence Openbook reference [8].

4.3.1.1 permuteDevice

The permuteDevice function simplifies a specific type of device depending on the specific skill function to perform simplification – for example, combine parallel resistance or series resistance or combine parallel FET. It can also combine MOS multiple-transistor configurations into single gate-function devices with permutable inputs. Below is a sample of the permuteDevice command as used in the divaLVS.rul file.

```
permuteDevice(parallel "nfet" combineParallelFET)
```

4.3.1.2 compareDeviceProperty

The compareDeviceProperty function is used to compare properties of devices matched in the layout and the schematic. This function calls another function that will compare the two devices. An example of the compareDeviceProperty function as used in the divaLVS.rul file is:

```
compareDeviceProperty("res" compareResistor)
compareDeviceProperty("nfet" compareFET)
compareDeviceProperty("pfet" compareFET)
```

where compareResistor and compareFET are two skill functions that do the comparing.

4.3.1.3 removeDevice

This function removes the device from the circuit and replaces the device with an open circuit or a short circuit. This function was used to remove the transmission line and the bump devices from the extracted view and short the terminals.

The function as used in the divaLVS.rul file is as follows:

```
removeDevice("ulwireSX" short("in" "out"))
removeDevice("bump" short("in" "out"))
```

Here, 'ulwireSX' and 'bump' are removed and the terminals 'in' and 'out' are shorted. This is done because in effect, the transmission lines and the bumps make no difference on how the circuit works.

4.3.1.4 ignoreTerminal

The ignoreTerminal command specifies which terminal types on which device types to ignore during the comparison. This is done because sometimes all the terminals in a device are not used in layout but are specified in the schematic, for example, the back gate of an N channel transistor. This may or may not be connected to power or ground in the layout but is always connected to the either of the two in schematic. As such it is best to ignore the back gate in the verification process as it might be erroneous to include it. An example is presented of the ignoreTerminal command is included here.

```
ignoreTerminal("nfet" "B" )
```

4.3.1.5 Series and Parallel Reduction

Devices that are connected in series or parallel can be combined by series or parallel reduction. This is done to cut down the number of devices that have to be compared with as resistors, capacitance and MOS devices can be reduced if they are connected

in series or in parallel. A more detailed explanation of Series and Parallel reduction can be found in the chapter on LVS in *Assura Diva Reference Guide*.

4.3.1.6 MOS Reduction

MOS reduction is also possible if the transistors are of the same type and form a logical function [8].

4.3.2 The Error Files

The LVS program generates files that store the errors that the program encountered. The error files that are created and a brief description of what errors they store are mentioned next.

netbad.out	Lists all unmatched nets
devbad.out	Lists all unmatched devices
prunenet.out	Lists all ignored nets
prunedev.out	Lists all ignored devices
mergenet.out	Lists all merged nets
termbad.out	Lists all unmatched terminals
audit.out	Lists all unmatched parameters

A sample output file that was created after running LVS on a design is presented next.

```
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Net-list summary for
/afs/unity.ncsu.edu/users/a/akvarma/LVS/layout/netlist
count
  15      nets
   4      terminals
   2      bump
  10      pmos
   1      ulwireSX
   2      ulwireSY
  10      nmos
Net-list summary for
/afs/unity.ncsu.edu/users/a/akvarma/LVS/schematic/netlist
count
  17      nets
   4      terminals
   2      bump
```

```

5          pmos
1          ulwireSX
2          ulwireSY
5          nmos
Terminal correspondence points
1  gnd!
2  in
3  out
4  vdd!

```

The net-lists match.

	layout	schematic
	instances	
un-matched	0	0
rewired	0	0
size errors	0	0
pruned	0	0
active	25	15
total	25	15

	nets	
un-matched	0	0
merged	0	0
pruned	0	0
active	15	17
total	15	17

	terminals	
un-matched	0	0
matched but		
different type	0	0
total	4	4

Probe files from /afs/unity.ncsu.edu/users/a/akvarma/LVS/schematic
devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

I /I4

? Device removed because of user's 'removeDevice' request.

I /I7

? Device removed because of user's 'removeDevice' request.

I /U2

? Device removed because of user's 'removeDevice' request.

I /U3

? Device removed because of user's 'removeDevice' request.

I /U1

? Device removed because of user's 'removeDevice' request.

audit.out:

Probe files from /afs/unity.ncsu.edu/users/a/akvarma/LVS/layout

```
devbad.out:
netbad.out:
mergenet.out:
termbad.out:
prunenet.out:
prunedev.out:
+4
? Device removed because of user's 'removeDevice' request.
I /+3
? Device removed because of user's 'removeDevice' request.
I /+2
? Device removed because of user's 'removeDevice' request.
I /+1
? Device removed because of user's 'removeDevice' request.
I /+0
? Device removed because of user's 'removeDevice' request.
audit.out:
```

5 CASE STUDY – DRIVER RECEIVER CIRCUIT

To demonstrate that the designed tool-set works satisfactorily, a driver and a receiver (which is simply an inverter) circuit were connected via off chip MCM layers. Figure 17 (modified from Afonso, et al [7]) shows the Driver – Receiver circuit and where

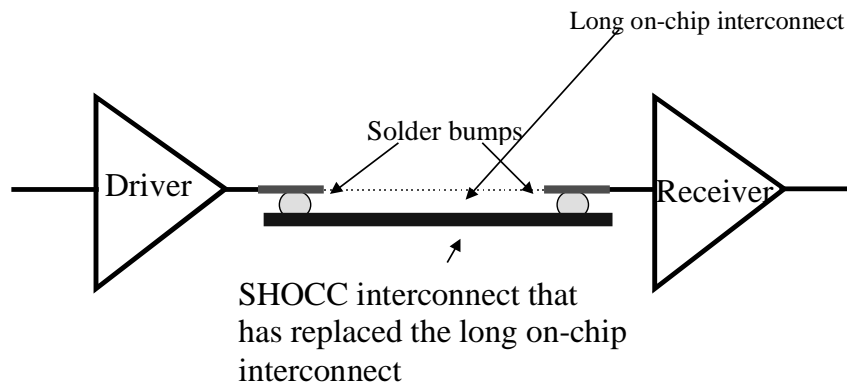


Figure 17: Driver and Receiver circuit

the interconnect attaches the two.

Signal from the driver was brought onto the substrate. The signal was propagated using the X and Y layers and was brought back on the chip to be connected to the

inverter. This circuit was checked for design rule errors, extracted and then the extracted netlist matched against the schematic netlist thus checking for any Layout Versus Schematic errors. This chapter discusses in detail the case study of the driver and the receiver circuit.

5.1 LAYOUT AND DRC

Layout was done by creating an instance of the driver (that was laid out previously for another project) and an instance of an inverter. The output of the driver is connected

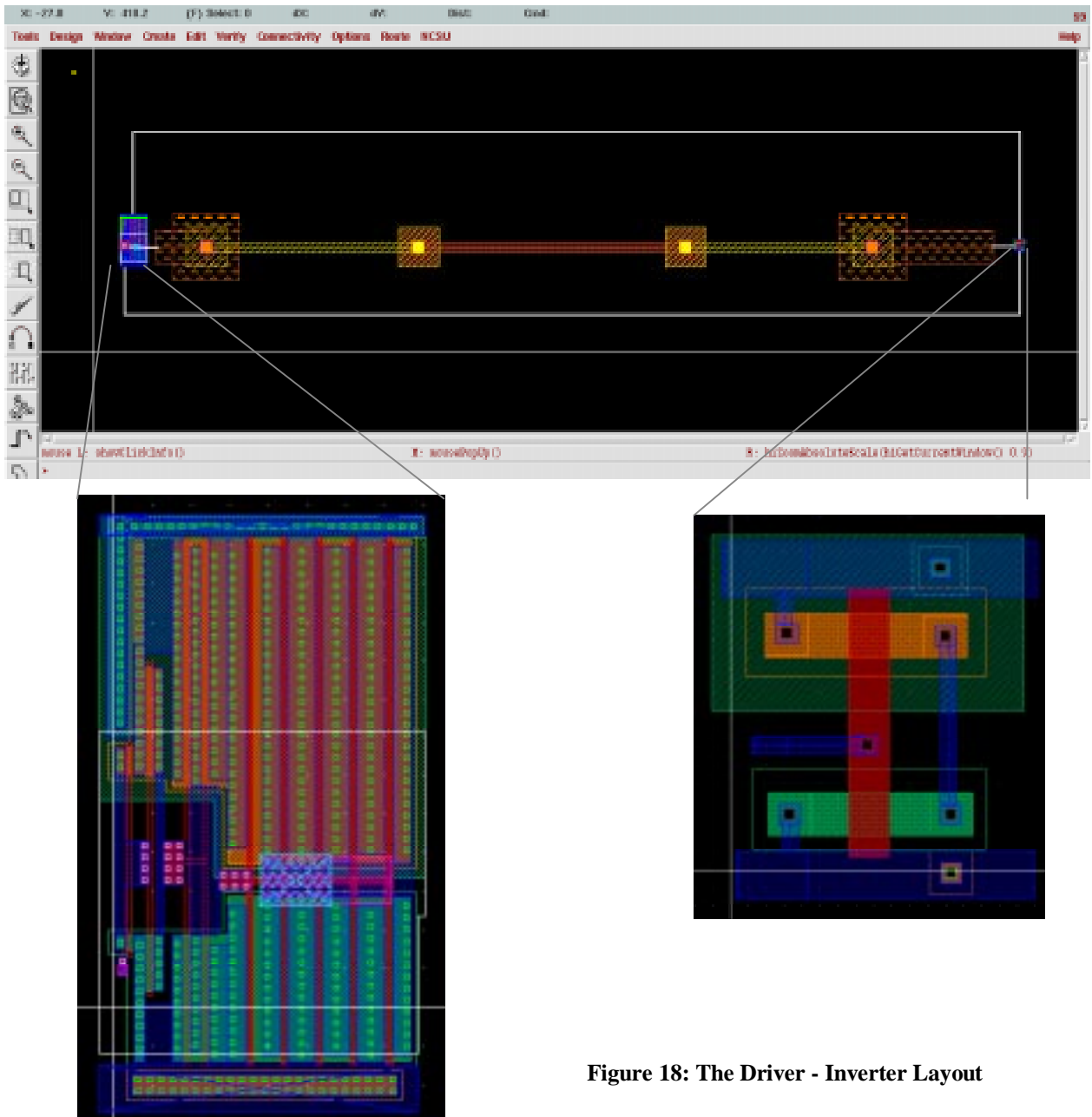


Figure 18: The Driver - Inverter Layout

to the input of the inverter via the substrate layers. Metal 3 at the output of the driver is laid out in such a way such that it overlaps mcm4 layer (a SHOCC top pad layer). From there, the signal is passed through to mcm3 (which is the Y Plane on the substrate) via the contact, mcmV4. The signal is then passed on to mcm2 (which is the X Plane on the substrate) via the contact layer mcmV3. The signal is then brought back to mcm3 from where it is connected to mcm4 and back to metal 3 to the input of the inverter. The layout described above is drawn out in figure 18. The two figures either side indicate the location of the driver and the inverter. The two figures following the complete layout are the figures of the driver and the inverter layout.

The driver and the inverter are subjected to individual design rule checks. Once they are connected together with the SHOCC layers, they are again tested for design rule violation. Presented below are the results of the individual DRC runs for the driver and the inverter and then the result of the combined DRC run.

Log file entry for DRC run for driver:

```
DRC started at Wed Feb 21 12:23:05 2001
library: DESchip
cell:    outputdriver2
view:    layout
Rules source is a simple file.
Rules path is /afs/eos.ncsu.edu/dist/cad445/local/techfile/divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Feb 21 12:23:05 2001
  completed ....Wed Feb 21 12:23:12 2001
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:07
**  Summary of rule violation for cell "outputdriver2 layout"  **
  Total errors found: 0
```

Log file entry for DRC Run for the inverter

```
DRC started at Wed Feb 21 12:24:46 2001
library: TEST_SHOCC_3
cell:    inv_hp
view:    layout
Rules source is a simple file.
Rules path is /afs/eos.ncsu.edu/dist/cad445/local/techfile/divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
```

```
Flat mode
Full checking.
DRC started.....Wed Feb 21 12:24:46 2001
  completed ....Wed Feb 21 12:24:52 2001
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
*****  Summary of rule violation for cell "inv_hp layout"  *****
  Total errors found: 0
```

Log file entry for DRC Run for the entire circuit

```
DRC started at Wed Feb 21 12:26:30 2001
library: TEST_SHOCC_3
cell:    SHOCC_TEST
view:    layout
Rules source is a simple file.
Rules path is ~/erl/SHOCC/final_code/divaDRC.rul.
Inclusion limit is set to 1000.
Running drclayout analysis
Flat mode
Full checking.
DRC started.....Wed Feb 21 12:26:30 2001
  completed ....Wed Feb 21 12:26:36 2001
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:06
*****  Summary of rule violation for cell "SHOCC_TEST layout"  *****
  Total errors found: 0
```

5.2 EXTRACTION

Extraction of the Driver- Inverter circuit results in the figure 19 on the next page. Shown is the complete extracted circuit with expanded views of the bump symbol, the UmodelX symbol and the UmodelY symbol. Also shown are the extracted circuits of the driver and the inverter. Unlike the DRC run, where we had to run DRC for the driver and the inverter separately, extraction of the entire circuit can be performed in one run.

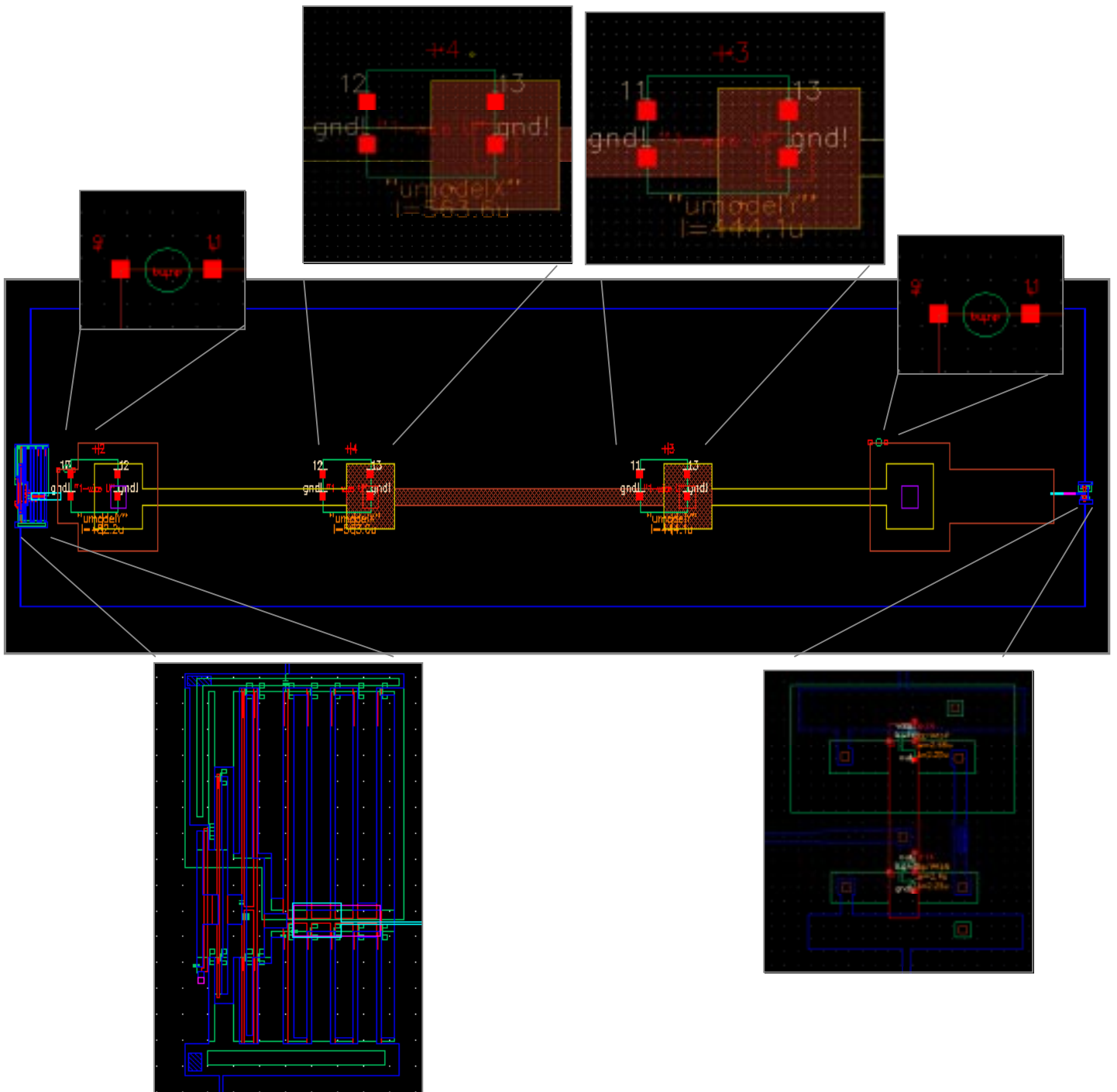


Figure 19: The Extracted Circuit

The complete extraction and the layer definition file are attached as appendix at the end of the report.

5.3 LVS

Once we have extracted the Driver – Inverter circuit, we can obtain the HSPICE netlist of the extracted view using Analog Circuit Design Environment and choosing HSPICE as the simulator. Layout Versus Schematic program compares the obtained netlist from the extracted view of the layout with the netlist from the schematic that has been drawn that represents the layout (which can be drawn by the designer or can be provided). The schematic is drawn using the symbols that have been created. The schematic of the Driver – Receiver circuit is shown in figure 20 below.

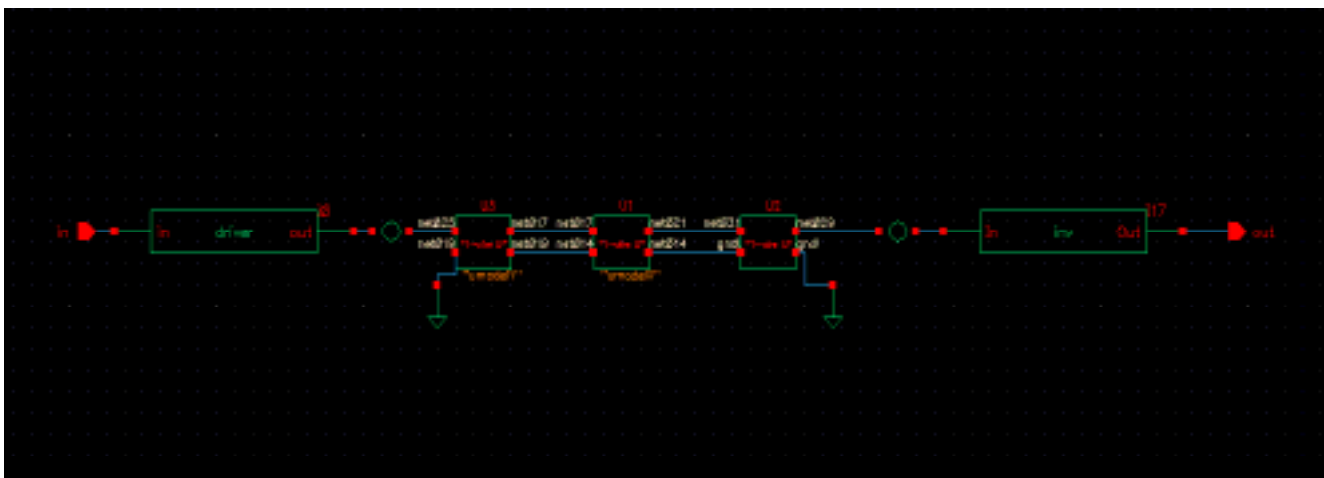


Figure 20: Schematic of the Driver - Inverter circuit

An important point to note in the LVS procedure is that the transmission line models and the solder bumps that were extracted out from the layout are removed from the LVS procedure and the terminals are replaced with short circuit. The output file created after running the LVS procedure is attached as an appendix.

5.4 NETLIST GENERATION

Once we have completed extraction, we need to get the netlist out for HSPICE simulation. The netlist is obtained using Cadence Analog Environment where after feeding the correct model paths and selecting the correct simulator (hspice), the request to generate the complete netlist is given. The correct environment options, such as the Vdd voltage and the input pulse, needs to be provided as well.

The final netlist that is generated appears like the one presented here:

```

* # FILE NAME: /AFS/UNITY.NCSU.EDU/USERS/A/AKVARMA/CADENCE/SIMULATION/
* SHOCC_TEST/hspiceS/extracted/netlist/SHOCC_TEST.C.raw
* Netlist output for hspiceS.
* Generated on Feb 21 19:35:05 2001
* File name: TEST_SHOCC_3_SHOCC_TEST_extracted.S.
* Subcircuit for cell: SHOCC_TEST.
* Generated for: hspiceS.
* Generated on Feb 21 19:35:05 2001.
C15 0 3 2.72160008051513E-15 M=1.0
C17 0 2 3.42629994065343E-15 M=1.0
C19 3 4 5.51700002414217E-15 M=1.0
C21 2 3 3.38814004758844E-15 M=1.0
C23 0 3 2.25783006013562E-15 M=1.0
C25 0 1 2.12688003703952E-15 M=1.0
C27 VDD! 0 164.090719094109E-15 M=1.0
M29 VDD! IN 1 VDD! HP14TBP L=600E-9 W=3.6E-6 AD=21.0150005036658E-12
+AS=5.39999989168649E-12 PD=23.5499992413679E-6 PS=6.59999977870029E-6 M=1
M31 2 1 VDD! VDD! HP14TBP L=600E-9 W=14.1E-6 AD=21.1500001540132E-12
+AS=21.0150005036658E-12 PD=17.0999992405996E-6 PS=23.5499992413679E-6 M=1
M33 VDD! 2 3 VDD! HP14TBP L=600E-9 W=32.1E-6 AD=93.9150007761569E-12
+AS=28.8899997674674E-12 PD=10.1249997896957E-6 PS=1.799999499807E-6 M=1
M35 3 2 VDD! VDD! HP14TBP L=600E-9 W=32.1E-6 AD=28.8899997674674E-12
+AS=48.1500013471692E-12 PD=1.799999499807E-6 PS=35.1000016962644E-6 M=1
M37 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=101.159997045741E-12
+AS=75.8699977843058E-12 PD=46.9500009785406E-6 PS=3.599998999614E-6 M=1
M39 VDD! 3 4 VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M41 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M43 VDD! 3 4 VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M45 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=93.9150007761569E-12 PD=3.599998999614E-6 PS=10.1249997896957E-6 M=1
M47 OUT 9 VDD! VDD! HP14TBP L=2.25E-6 W=2.55E-6 AD=11.0925001822748E-12
+AS=11.8575002247545E-12 PD=11.2500001705484E-6 PS=11.850000191771E-6 M=1
M49 0 IN 1 0 HP14TBN L=600E-9 W=1.8E-6 AD=11.0699999514297E-12
+AS=2.6999994584325E-12 PD=13.5000000227592E-6 PS=4.80000016978011E-6 M=1
M51 2 1 0 0 HP14TBN L=600E-9 W=7.2E-6 AD=10.799999783373E-12
+AS=11.0699999514297E-12 PD=10.2000003607827E-6 PS=13.5000000227592E-6 M=1
M53 0 2 3 0 HP14TBN L=600E-9 W=16.2E-6 AD=47.2499990522568E-12
+AS=14.5800004014429E-12 PD=7.57499992687372E-6 PS=1.799999499807E-6 M=1
M55 3 2 0 0 HP14TBN L=600E-9 W=16.2E-6 AD=14.5800004014429E-12
+AS=24.2999995125892E-12 PD=1.799999499807E-6 PS=19.2000006791204E-6 M=1
M57 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=50.7600003696318E-12
+AS=38.0699985425004E-12 PD=25.949999326258E-6 PS=3.599998999614E-6 M=1
M59 0 3 4 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M61 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1

```

```

M63 0 3 4 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M65 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=47.2499990522568E-12 PD=3.599998999614E-6 PS=7.57499992687372E-6 M=1
M67 OUT 9 0 0 HP14TBN L=2.25E-6 W=2.4E-6 AD=11.1600000074485E-12
+AS=10.799999783373E-12 PD=11.699999590917E-6 PS=11.4000004032278E-6 M=1
U69 12 0 13 0 UMODELX L=563.599984161556E-6
U71 11 0 13 0 UMODELY L=444.050005171448E-6
U73 10 0 12 0 UMODELY L=482.150004245341E-6
X75 4 10 BUMP_BSUBCKT_1
X77 9 11 BUMP_BSUBCKT_1
.lib "/ncsu/cadence/local/models/hspice/hpl4tb/hsp_nom_lib" hpl4tbp
.lib "~/cadence/umodelY.m" umodelY
.lib "~/cadence/umodelX.m" umodelX
.lib "/ncsu/cadence/local/models/hspice/hpl4tb/hsp_nom_lib" hpl4tbn
* Include files
VDD VDD! 0 3.3
V IN 0 0 PULSE (0 3.3 0N 0.5N 0.5N 3N 6N)
.TRAN 0.1N 15N
.OPTIONS LIST NODE POST
* End of Netlist
.SUBCKT BUMP_BSUBCKT_1 1 2
RBUMP 1 3 17.2M
LBUMP 3 2 8.94P
CBUMP 2 0 18.4F
.ENDS BUMP_BSUBCKT_1
.TEMP 25.0000
.END

```

The contents of “~/cadence/umodelY.m” and “~/cadence/umodelX.m” are described discussed in section 4.2.3. The contents are also listed in Appendix G.

5.5 SIMULATION

Once the simulation is completed using HSPICE, the output is plotted using *Avant Waves*.

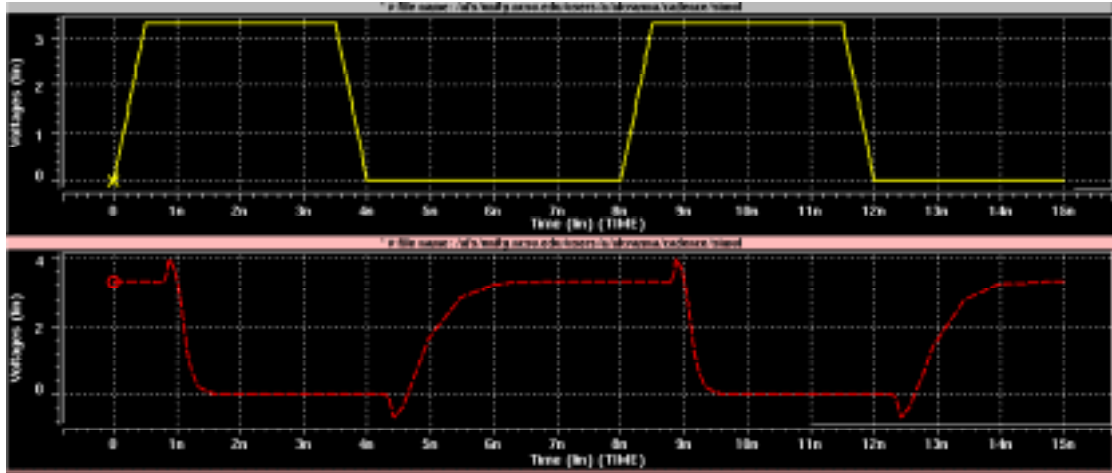


Figure 21: Simulation of the Driver - Receiver Circuit

In figure 21, the input to the receiver is depicted in the upper of the two waveforms whereas the lower waveform shows the output from the inverter.

A close inspection of the waveforms shows us that the circuit is indeed working, as the output from the inverter is an inverted version of the input.

The test case was performed with the designed tool-set which was installed on the NCSU Cadence Design Kit.

6 CONCLUSION AND FUTURE WORK

This project accomplished its goal of demonstrating that interconnects on the substrate can be co-designed with the rest of the chip. Before we could test our design, there were four very distinct pieces of this project we had to work on.

We started with introducing new substrate layers in the **technology file**. Ground, power, X and Y layers are added as mcm substrate layers. Apart from describing their physical appearance, the technology file that was created also defined some physical rules such as minimum spacing between layers, minimum width of the layers, minimum enclosure rules (enclosure of a layer by another layer). The technology file also defined devices such as the mcm2_mcm1 and mcm3_mcm2 that are contact devices that Virtuoso uses when joining two layers. Other physical properties of the layers were defined in the **display.drf** file.

The next part of the project was to write the **divaDRC.rul (Design Rule Check)** file. This file would check for common design errors that the designer would commit such as the distance between two layers, whether the contact layers are properly covered by the contacted layers, whether two contact layers are overlapping each other and so on. The divaDRC.rul file written is to be used specifically for the substrate layers. One of

the future jobs to be done includes the integration of the on-chip and off-chip design rule checks.

Extraction of the on-chip and off chip circuitry was a big challenge. To model the off-chip interconnects as U element (Lossy) transmission lines and to extract the solder bumps was our main objective. HSPICE models for the transmission lines and the solder bump sub-circuit were written and integrated with the extraction procedure. Past NCSU-CDK research resulted in extraction of on-chip parasitic capacitance. We tried obtaining the parasitic resistance for the on-chip circuitry but due to time constraints this job is added to the list of future things to do.

Now that we had the extracted circuit, **Layout Versus Extraction** netlist matching was the next issue that was worked upon in the project. For this we needed to have a schematic view of the circuit. This program generated the netlist from the schematic and the extracted view and compared the two, thus finding the similarity between the two views of the circuit.

After having completed the four crucial parts of the project, work on the last and the most important part of testing the entire tool-set was started. We decided to test the decks using a driver, receiver circuit. The driver circuit was a pre-designed four-stage inverter circuit with a total of 4 inverters of different W/L ratios of the gates. The receiver circuit was simply an inverter connected on the other end of the substrate.

An important item in the future things to do is to have the **Post Extraction Filter** added to the tool-set. Often it is desirable to extract just a set of specified nets or to conduct a timing verification run and sort the nets by criticality. It will be good to investigate alternative means for delivering this functionality and determine a reasonable approach to doing so.

Another feature that could be investigated is to have **auto-routing** capability. Presently, Cadence allows auto-routing facility with nets that are on the same electrical connection. The SHOCC auto-router would allow marking nets that are on the same electrical connection not only on the substrate, but also nets that connect on-

chip connection with off-chip connection. This would go a long way in assisting designers of multi-chip modules.

In the current tool-set, we have files that are not integrated with a standard library such as tsmc03 (TSMC 0.3 μm process) or hp06 (HP 0.6 μm process). Another future task would be to incorporate the tool-set with the existing tech files and the diva deck so that simultaneous design and verification can be performed.

References

References

- [1]. “*Seamless High Off-Chip Connectivity.*” L Schaper, M Dibbs & P Garrou, CC Chau, Y So, D Frye, J Wagner, J Ousley, G Baugher, R Picard, G Connor, D Winn, P Deane, R Eden, R Sands. Proc. Int. Symp. Microelectron. Oct 1997.
- [2]. *Multichip Module – Technologies and Alternatives: The Basics.* Daryl Ann Doane, Paul D. Franzon, VAN NOSTRAND REINHOLD 1993.
- [3]. “*Infrastructure and course progression for complex IC design education.*” Franzon, P.D.; Wentai Lui; Gloster, C.; Schaffer, T.; Glaser, A.; Stanaski, A. Microelectronic Systems Education, 1999. MSE’99. IEEE International Conference, 1999 Page(s): 88 -89
- [4]. “*A flip-chip implementation of the Data Encryption Standard (DES).*” Schaffer, T.; Glaser, A.; Rao, S.; Franzon, P. Multi-Chip Module Conference, 1997. MCMC '97., 1997 IEEE , 1997 Page(s): 13 –17
- [5]. *The NCSU Design Kit for IC Fabrication through MOSIS.* Toby Schaffer Andy Stanaski Alan Glaser, Paul Franzon. available at <http://www.ece.ncsu.edu/cadence/CUG1998/index.html>
- [6]. “*System Design Optimization for MCM-D/Flip-Chip.*” Franzon, P.D.; Stanaski, A.; Tekmen, Y.; Banerjia, S. Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on, Volume: 18 Issue: 4 , Nov. 1995, 620 –627
- [7]. “*Modeling and electrical analysis of seamless high off-chip connectivity (SHOCC) interconnects.*” Afonso, S.; Schaper, L.W.; Parkerson, J.P.; Brown, W.D.; Ang, S.; Naseem, H.A. Advanced Packaging, IEEE Transactions on [see also Components, Packaging and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on] , Volume: 22 Issue: 3 , Aug. 1999; 309 –320
- [8]. *Cadence Openbook – online documentation for cadence specific topics.*
- [9]. *MCM-D Technology Kit – MicroModule Systems and Mentor Graphics Rev 1.01* 1 –Sep 1994.
- [10]. *Circuits, Interconnections and Packaging for VLSI.* H. B. Bakaglu. Addison Wesley, 1989
- [11]. *Computer Circuits Electrical Design.* Poon, Ron K., Prentice Hall, Englewood Cliffs, NJ, ISBN 0-13-213471-3, 1995.

- [12]. "When are transmission-line effects important for on-chip interconnects ?,"
A. Deutsch, G.V. Kopcsay, P.J. Restle, H.H. Smith, G. Katopis, W.D. Becker,
P.W. Coteus, C.W. Surovic, B.J.Rubin, R.P. Dunne, Jr., T. Gallo, K.A. Jenkins,
L.M. Terman, R.H. Dennard, G.A. Sai-Halasz, B.L. Krauter, and D.R. Knebel,
IEEE Trans. Microwave Theory &Tech., vol.45, no.10, pp.1836--1844, Oct. 1997.
- [13]. "High-Speed Signal Propagation on Lossy Transmission Lines," A. Deutsch,
G. V. Kopcsay, V. A. Ranieri, J. K. Cataldo, E. A. Galligan, W. S. Graham, R. P.
McGouey, S. L. Nunes, J. R. Paraszczak, J. J. Ritsko, R. J. Serino, D. Y. Shih, and
J. S. Wilczynski, IBM J. Res. Develop. 34, No. 4, 601-615 (1990).
- [14]. "Design and Technology Guidelines for Short, Medium, and Long On-Chip
Interconnections," A. Deutsch, W. D. Becker, G. A. Katopis, H. Smith, P. J.
Restle, P. W. Coteus, C. W. Surovic, G. V. Kopcsay, B. J. Rubin, R. P. Dunne, T.
Gallo, K. A. Jenkins, L. M. Terman, R. H. Dennard, and D. R. Knebel,
Proceedings of the 1996 IEEE Electrical Performance of Electronic Packaging
Conference, 1996, pp. 30-33.
- [15]. *Principles of CMOS VLSI Design*. Neil H. E. Weste, Karman Eshragian,
Addison-Wesley, 1993
- [16]. MOSIS Homepage:
<http://www.mosis.org/>
- [17]. Online HSPICE Manual. :
http://www.eos.ncsu.edu/software/software_index/spice.html

Appendices

Appendix A – The Technology File

```
; Technology File NCSU_TechLib_SHOCC_tsmc02
; created on Aug 28 17:27:29 2000

;*****
; CONTROLS
;*****
controls(
  techParams(
    ;( parameter          value          )
    ;( -----          -----          )
    ( lambda              0.2 )
    ( technology          "HP_GMOS10QA" )
    ( metal3Available t   )
    ( metal4Available t   )
  ) ;techParams
  techPermissions(
    ;( class              (read-only users) (read & write users) )
    ;( -----          -----          -----          )
  ) ;techPermissions
) ;controls

;*****
; LAYER DEFINITION
;*****
layerDefinitions(
  techPurposes(
    ;( PurposeName          Purpose#   Abbreviation )
    ;( -----          -----   ----- )
    ;User-Defined Purposes:
    ;System-Reserved Purposes:
    ( warning              234       wng          )
    ( tool1                 235       t11          )
    ( tool0                 236       t10          )
    ( label                 237       lb1          )
    ( flight                238       flt          )
    ( error                 239       err          )
    ( annotate              240       ant          )
    ( drawing1              241       dr1          )
    ( drawing2              242       dr2          )
    ( drawing3              243       dr3          )
    ( drawing4              244       dr4          )
    ( drawing5              245       dr5          )
    ( drawing6              246       dr6          )
    ( drawing7              247       dr7          )
    ( drawing8              248       dr8          )
    ( drawing9              249       dr9          )
    ( boundary              250       bnd          )
    ( pin                   251       pin          )
    ( drawing               252       drw          )
    ( net                   253       net          )
    ( cell                  254       cel          )
    ( all                   255       all          )
  ) ;techPurposes

  techLayers(
    ;( LayerName          Layer#   Abbreviation )
    ;( -----          -----   ----- )
    ;User-Defined Layers:
    ( default              0       default     )
    ( nactive              1       nactive     )
  ) ;techLayers
)
```

```

( pactive          2      pactive      )
( active          3      active      )
( pwell           6      pwell      )
( via2            7      via2      )
( via             8      via      )
( via3            9      via3      )
( nwell          12      nwell      )
( gwell          18      gwell      )
( pad            29      pad      )
( poly           35      poly      )
( glass          36      glass      )
( nselect        39      nselect   )
( pselect        40      pselect   )
( gselect        41      gselect   )
( metall         45      metall    )
( metal3         46      metal3    )
( metal4         47      metal4    )
( metal2         50      metal2    )
( ca             56      ca      )
( cp             57      cp      )
( cc             59      cc      )
( nodrc          80      nodrc    )
( cap_id         81      cap_id    )
( res_id         82      res_id    )
( nolpe          83      nolpe    )
( hdrc           126     hdrc     )

( mcm0            84      mcm0      ) ; mcm layer (ground plane)
( mcm1            85      mcm1      ) ; mcm layer (power plane)
( dummymcm1      94      d_mcm1    ) ; mcm layer
( mcmV1           86      mcmV1     ) ; mcm layer (via between
mcm2 and mcm0)
( mcm2            87      mcm2      ) ; mcm layer (X plane)
( mcmV2           88      mcmV2     ) ; mcm layer (via between
mcm2 to mcm1)
( mcm3            89      mcm3      ) ; mcm layer (Y plane)
( mcmV3           90      mcmV3     ) ; mcm layer (via between
mcm3 to mcm2)
( mcm4            91      mcm4      ) ; mcm layer (Top pad
metal)
( mcmV4           92      mcmV4     ) ; mcm layer (via between
mcm4 to mcm3)
( mcmglass        93      mcmglass  ) ; mcm layer

;System-Reserved Layers:
( Unrouted        200     Unroute   )
( Row             201     Row      )
( Group           202     Group    )
( Cannotoccupy   203     Cannotoc )
( Canplace       204     Canplac  )
( hardFence      205     hardFen  )
( softFence      206     softFen  )
( y0             207     y0      )
( y1             208     y1      )
( y2             209     y2      )
( y3             210     y3      )
( y4             211     y4      )
( y5             212     y5      )
( y6             213     y6      )
( y7             214     y7      )
( y8             215     y8      )
( y9             216     y9      )
( designFlow     217     designF  )
( stretch       218     stretch )
( edgeLayer      219     edgeLay  )
( changedLayer   220     changed  )

```

```

( unset                221        unset        )
( unknown              222        unknown      )
( spike                223        spike        )
( hiz                  224        hiz          )
( resist               225        resist       )
( drive                226        drive       )
( supply               227        supply      )
( wire                 228        wire        )
( pin                  229        pin         )
( text                 230        text        )
( device               231        device      )
( border               232        border      )
( snap                 233        snap        )
( align                234        align       )
( prBoundary           235        prBound     )
( instance             236        instanc     )
( annotate             237        annotat    )
( marker               238        marker      )
( select               239        select      )
( grid                 251        grid        )
( axis                 252        axis        )
( hilite               253        hilite      )
( background           254        backgro     )
) ;techLayers

```

```

techLayerPurposePriorities(
;layers are ordered from lowest to highest priority
;( LayerName           Purpose       )
;( -----            -)
( pwell                drawing   )
( nwell                drawing   )
( active               drawing   )
( nactive              drawing   )
( pactive              drawing   )
( nselect              drawing   )
( pselect              drawing   )
( poly                 drawing   )
( metall               drawing   )
( metal2               drawing   )
( metal3               drawing   )
( metal4               drawing   )
( cp                   drawing   )
( ca                   drawing   )
( cc                   drawing   )
( via                  drawing   )
( via2                 drawing   )
( via3                 drawing   )
( glass                drawing   )

;( solderbump          drawing   )
( mcm0                 drawing   )
( mcm1                 drawing   )
( dummmcm1             drawing   )
( mcm2                 drawing   )
( mcm3                 drawing   )
( mcm4                 drawing   )
( mcmV1                drawing   )
( mcmV2                drawing   )
( mcmV3                drawing   )
( mcmV4                drawing   )
( mcmglass             drawing   )

( background           drawing   )
( grid                 drawing   )
( grid                 drawing1  )
( annotate              drawing   )
( annotate              drawing1  )

```

```

( annotate          drawing2 )
( annotate          drawing3 )
( annotate          drawing4 )
( annotate          drawing5 )
( annotate          drawing6 )
( annotate          drawing7 )
( annotate          drawing8 )
( annotate          drawing9 )
( default          drawing )
( instance         drawing )
( instance         label )
( prBoundary       drawing )
( prBoundary       boundary )
( prBoundary       label )
( gselect          drawing )
( gwell            drawing )
( align            drawing )
( hardFence        drawing )
( softFence        drawing )
( nodrc            drawing )
( nolpe            drawing )
( pad              drawing )
( text             drawing )
( text             drawing1 )
( text             drawing2 )
( res_id           drawing )
( cap_id           drawing )
( border           drawing )
( device           drawing )
( device           label )
( device           drawing1 )
( device           drawing2 )
( device           annotate )
( wire             drawing )
( wire             label )
( wire             flight )
( pin              label )
( pin              drawing )
( pin              annotate )
( axis             drawing )
( edgeLayer        drawing )
( edgeLayer        pin )
( snap             drawing )
( stretch         drawing )
( y0               drawing )
( y1               drawing )
( y2               drawing )
( y3               drawing )
( y4               drawing )
( y5               drawing )
( y6               drawing )
( y7               drawing )
( y8               drawing )
( y9               drawing )
( hilite           drawing )
( hilite           drawing1 )
( hilite           drawing2 )
( hilite           drawing3 )
( hilite           drawing4 )
( hilite           drawing5 )
( hilite           drawing6 )
( hilite           drawing7 )
( hilite           drawing8 )
( hilite           drawing9 )
( select           drawing )
( drive            drawing )
( hiz              drawing )

```

```

( resist          drawing      )
( spike          drawing      )
( supply         drawing      )
( unknown        drawing      )
( unset          drawing      )
( designFlow     drawing      )
( designFlow     drawing1     )
( designFlow     drawing2     )
( designFlow     drawing3     )
( designFlow     drawing4     )
( designFlow     drawing5     )
( designFlow     drawing6     )
( designFlow     drawing7     )
( designFlow     drawing8     )
( designFlow     drawing9     )
( changedLayer   tool0        )
( changedLayer   tool1        )
( marker         warning      )
( marker         error        )
( Row            drawing      )
( Row            label         )
( Group          drawing      )
( Group          label         )
( ca             label         )
( cc             label         )
( cp             label         )
( metall         label         )
( metal2         label         )
( metal3         label         )
( metal4         label         )
( poly           label         )
( via            label         )
( via2           label         )
( via3           label         )

( mcm0           label        )
( mcm1           label        )
( mcm2           label        )
( mcm3           label        )
( mcm4           label        )
( mcmV1          label        )
( mcmV2          label        )
( mcmV3          label        )
( mcmV4          label        )

( pwell          net          )
( nwell          net          )
( gwell          net          )
( active         net          )
( nactive        net          )
( pactive        net          )
( poly           net          )
( metall         net          )
( metal2         net          )
( metal3         net          )
( metal4         net          )
( ca             net          )
( cc             net          )
( cp             net          )
( via            net          )
( via2           net          )
( via3           net          )

( mcm0           net          )
( mcm1           net          )
( mcm2           net          )
( mcm3           net          )

```

```

(mcm4 net )
(mcmV1 net )
(mcmV2 net )
(mcmV3 net )
(mcmV4 net )

( active pin )
( ca pin )
( cc pin )
( cp pin )
( gwell pin )
( metall pin )
( metal2 pin )
( metal3 pin )
( metal4 pin )
( nactive pin )
( nwell pin )
( pactive pin )
( poly pin )
( pwell pin )
( via pin )
( via2 pin )
( via3 pin )

(mcm0 pin )
(mcm1 pin )
(mcm2 pin )
(mcm3 pin )
(mcm4 pin )
(mcmV1 pin )
(mcmV2 pin )
(mcmV3 pin )
(mcmV4 pin )

( Cannotoccupy drawing )
( Cannotoccupy boundary )
( Canplace drawing )
( ca boundary )
( cc boundary )
( cp boundary )
( metall boundary )
( metal2 boundary )
( metal3 boundary )
( metal4 boundary )
( poly boundary )
( via boundary )
( via2 boundary )
( via3 boundary )
( hdrc boundary )

(mcm0 boundary )
(mcm1 boundary )
(mcm2 boundary )
(mcm3 boundary )
(mcm4 boundary )
(mcmV1 boundary )
(mcmV2 boundary )
(mcmV3 boundary )
(mcmV4 boundary )

( Unrouted drawing )
( Unrouted drawing1 )
( Unrouted drawing2 )
( Unrouted drawing3 )
( Unrouted drawing4 )
( Unrouted drawing5 )
( Unrouted drawing6 )

```

```

( Unrouted          drawing7  )
( Unrouted          drawing8  )
( Unrouted          drawing9  )
) ;techLayerPurposePriorities

```

```

techDisplays(
;( LayerName      Purpose      Packet          Vis Sel Con2ChgLy DrgEnbl
Valid )
;( ---  -----  ---  -----  ----- )
( pwell          drawing      pwell          t t t t t )
( nwell          drawing      nwell          t t t t t )
( active         drawing      active         t t t t t )
( nactive        drawing      nactive        t t t t t )
( pactive        drawing      pactive        t t t t t )
( nselect        drawing      nselect        t t t t t )
( pselect        drawing      pselect        t t t t t )
( poly           drawing      poly           t t t t t )
( metall         drawing      metall         t t t t t )
( metal2         drawing      metal2         t t t t t )
( metal3         drawing      metal3         t t t t t )
( metal4         drawing      metal4         t t t t t )
( cp             drawing      cp             t t t t nil )
( ca             drawing      ca             t t t t nil )
( cc             drawing      cc             t t t t t )
( via            drawing      via            t t t t t )
( via2           drawing      via2           t t t t t )
( via3           drawing      via3           t t t t t )
( glass          drawing      glass          t t t t t )

( mcm0           drawing      mcm0           t t t t t )
( mcm1           drawing      mcm1           t t t t t )
( dummymcm1     drawing      d_mcm1         t t t t nil )
( mcm2           drawing      mcm2           t t t t t )
( mcm3           drawing      mcm3           t t t t t )
( mcm4           drawing      mcm4           t t t t t )
( mcmV1          drawing      mcmV1          t t t t t )
( mcmV2          drawing      mcmV2          t t t t t )
( mcmV3          drawing      mcmV3          t t t t t )
( mcmV4          drawing      mcmV4          t t t t t )

( background     drawing      background     t nil t nil nil )
( grid           drawing      grid           t nil t nil nil )
( grid           drawing1     grid1          t nil t nil nil )
( annotate        drawing      annotate        t t t t nil )
( annotate        drawing1     annotate1       t t t t nil )
( annotate        drawing2     annotate2       t t t t nil )
( annotate        drawing3     annotate3       t t t t nil )
( annotate        drawing4     annotate4       t t t t nil )
( annotate        drawing5     annotate5       t t t t nil )
( annotate        drawing6     annotate6       t t t t nil )
( annotate        drawing7     annotate7       t t t t nil )
( annotate        drawing8     annotate8       t t t t nil )
( annotate        drawing9     annotate9       nil t t t nil )
( default        drawing      default        t t t t nil )
( instance       drawing      instance       t t t t nil )
( instance       label         instanceLbl    t t t t nil )
( prBoundary     drawing      prBoundary     t t t t nil )
( prBoundary     boundary     prBoundaryBnd t t t t nil )
( prBoundary     label         prBoundaryLbl  t t t t nil )
( gselect        drawing      gselect        t t t t nil )
( gwell          drawing      gwell          t t t t nil )
( align          drawing      align          t t t t nil )
( hardFence      drawing      hardFence     t t t t nil )
( softFence      drawing      softFence     t t t t nil )
( nodrc          drawing      nodrc         t t t t t )
( nolpe          drawing      nolpe         t t t t t )
( pad            drawing      pad            t t t t t )

```

```

( text          drawing      text          t t t t t )
( text          drawing1     text1         t t t t nil )
( text          drawing2     text2         t t t t nil )
( res_id        drawing      res_id        t t t t t )
( cap_id        drawing      cap_id        t t t t t )
( border        drawing      border        t t t t nil )
( device        drawing      device        t t t t nil )
( device        label        deviceLbl     t t t t nil )
( device        drawing1     device1       t t t t nil )
( device        drawing2     device2       t t t t nil )
( device        annotate     deviceAnt     t t t t nil )
( wire          drawing      wire          t t t t nil )
( wire          label        wireLbl       t t t t nil )
( wire          flight       wireFlt       t t t t nil )
( pin           label        pinLbl        t t t t nil )
( pin           drawing      pin           t t t t nil )
( pin           annotate     pinAnt        t t t t nil )
( axis          drawing      axis          t nil t t nil )
( edgeLayer     drawing      edgeLayer     t t t t nil )
( edgeLayer     pin         edgeLayerPin  t t t t nil )
( snap          drawing      snap          t t t t nil )
( stretch      drawing      stretch      t t t t nil )
( y0            drawing      y0            t t t t nil )
( y1            drawing      y1            t t t t nil )
( y2            drawing      y2            t t t t nil )
( y3            drawing      y3            t t t t nil )
( y4            drawing      y4            t t t t nil )
( y5            drawing      y5            t t t t nil )
( y6            drawing      y6            t t t t nil )
( y7            drawing      y7            t t t t nil )
( y8            drawing      y8            t t t t nil )
( y9            drawing      y9            t t t t nil )
( hilite        drawing      hilite        t t t t nil )
( hilite        drawing1     hilite1       t t t t nil )
( hilite        drawing2     hilite2       t t t t nil )
( hilite        drawing3     hilite3       t t t t nil )
( hilite        drawing4     hilite4       t t t t nil )
( hilite        drawing5     hilite5       t t t t nil )
( hilite        drawing6     hilite6       t t t t nil )
( hilite        drawing7     hilite7       t t t t nil )
( hilite        drawing8     hilite8       t t t t nil )
( hilite        drawing9     hilite9       t t t t nil )
( select        drawing      select        t t t t nil )
( drive         drawing      drive         t t t t nil )
( hiz           drawing      hiz           t t t t nil )
( resist        drawing      resist        t t t t nil )
( spike         drawing      spike         t t t t nil )
( supply        drawing      supply        t t t t nil )
( unknown       drawing      unknown       t t t t nil )
( unset         drawing      unset         t t t t nil )
( designFlow    drawing      designFlow    t t t nil nil )
( designFlow    drawing1     designFlow1   t t t nil nil )
( designFlow    drawing2     designFlow2   t t t nil nil )
( designFlow    drawing3     designFlow3   t t t nil nil )
( designFlow    drawing4     designFlow4   t t t nil nil )
( designFlow    drawing5     designFlow5   t t t nil nil )
( designFlow    drawing6     designFlow6   t t t nil nil )
( designFlow    drawing7     designFlow7   t t t nil nil )
( designFlow    drawing8     designFlow8   t t t nil nil )
( designFlow    drawing9     designFlow9   t t t nil nil )
( changedLayer  tool0        changedLayerTl0 nil nil t nil nil )
( changedLayer  tool1        changedLayerTl1 nil nil t nil nil )
( marker        warning     markerWarn    t t t t nil )
( marker        error       markerErr     t t t t nil )
( Row           drawing      Row           t t t t nil )
( Row           label        RowLbl        t nil t t nil )
( Group         drawing      Group         t t t t nil )

```

```

( Group      label      GroupLbl      t nil t t nil )
( ca        label      caLbl        t t t nil nil )
( cc        label      ccLbl        t t t nil nil )
( cp        label      cpLbl        t t t nil nil )
( metall1   label      metall1Lbl   t t t nil nil )
( metal2    label      metal2Lbl    t t t nil nil )
( metal3    label      metal3Lbl    t t t nil nil )
( metal4    label      metal4Lbl    t t t nil nil )
( poly      label      polyLbl     t t t nil nil )
( via       label      viaLbl      t t t nil nil )
( via2      label      via2Lbl     t t t nil nil )
( via3      label      via3Lbl     t t t nil nil )

( mcm0      label      mcm0Lbl     t t t nil nil )
( mcm1      label      mcm1Lbl     t t t nil nil )
( mcm2      label      mcm2Lbl     t t t nil nil )
( mcm3      label      mcm3Lbl     t t t nil nil )
( mcm4      label      mcm4Lbl     t t t nil nil )
( mcmV1     label      mcmV1Lbl    t t t nil nil )
( mcmV2     label      mcmV2Lbl    t t t nil nil )
( mcmV3     label      mcmV3Lbl    t t t nil nil )
( mcmV4     label      mcmV4Lbl    t t t nil nil )

( pwell     net        pwellNet    t t t nil t )
( nwell     net        nwellNet    t t t nil t )
( gwell     net        gwellNet    t t t nil nil )
( active    net        activeNet   t t t nil t )
( nactive   net        nactiveNet  t t t nil t )
( pactive   net        pactiveNet  t t t nil t )
( poly      net        polyNet     t t t nil t )
( metall1   net        metall1Net  t t t nil t )
( metal2    net        metal2Net  t t t nil t )
( metal3    net        metal3Net  t t t nil t )
( metal4    net        metal4Net  t t t nil t )
( ca        net        caNet      t t t nil nil )
( cc        net        ccNet      t t t nil t )
( cp        net        cpNet      t t t nil nil )
( via       net        viaNet     t t t nil t )
( via2      net        via2Net    t t t nil t )
( via3      net        via3Net    t t t nil t )

( mcm0      net        mcm0Net    t t t nil t )
( mcm1      net        mcm1Net    t t t nil t )
( mcm2      net        mcm2Net    t t t nil t )
( mcm3      net        mcm3Net    t t t nil t )
( mcm4      net        mcm4Net    t t t nil t )
( mcmV1     net        mcmV1Net   t t t nil t )
( mcmV2     net        mcmV2Net   t t t nil t )
( mcmV3     net        mcmV3Net   t t t nil t )
( mcmV4     net        mcmV4Net   t t t nil t )
( mcmglass  net        mcmglassNet t t t nil t )

( active    pin        activePin   t t t nil nil )
( ca        pin        caPin      t t t nil nil )
( cc        pin        ccPin      t t t nil nil )
( cp        pin        cpPin      t t t nil nil )
( gwell     pin        gwellPin   t t t nil nil )
( metall1   pin        metall1Pin  t t t nil nil )
( metal2    pin        metal2Pin  t t t nil nil )
( metal3    pin        metal3Pin  t t t nil nil )
( metal4    pin        metal4Pin  t t t nil nil )
( nactive   pin        nactivePin  t t t nil nil )
( nwell     pin        nwellPin   t t t nil nil )
( pactive   pin        pactivePin  t t t nil nil )
( poly      pin        polyPin    t t t nil nil )
( pwell     pin        pwellPin   t t t nil nil )
( via       pin        viaPin     t t t nil nil )

```

```

( via2          pin          via2Pin          t t t nil nil )
( via3          pin          via3Pin          t t t nil nil )

( mcm0          pin          mcm0Pin           t t t nil nil )
( mcm1          pin          mcm1Pin           t t t nil nil )
( mcm2          pin          mcm2Pin           t t t nil nil )
( mcm3          pin          mcm3Pin           t t t nil nil )
( mcm4          pin          mcm4Pin           t t t nil nil )
( mcmV1         pin          mcmV1Pin          t t t nil nil )
( mcmV2         pin          mcmV2Pin          t t t nil nil )
( mcmV3         pin          mcmV3Pin          t t t nil nil )
( mcmV4         pin          mcmV4Pin          t t t nil nil )

( Cannotoccupy drawing      Cannotoccupy      t t t t nil )
( Cannotoccupy boundary    CannotoccupyBnd t t t t nil )
( Canplace          drawing  Canplace          t t t t nil )
( ca                boundary caBnd              t t t nil nil )
( cc                boundary ccBnd              t t t nil nil )
( cp                boundary cpBnd              t t t nil nil )
( metall1          boundary metall1Bnd         t t t nil nil )
( metal2           boundary metal2Bnd         t t t nil nil )
( metal3           boundary metal3Bnd         t t t nil nil )
( metal4           boundary metal4Bnd         t t t nil nil )
( poly             boundary polyBnd           t t t nil nil )
( via              boundary viaBnd            t t t nil nil )
( via2            boundary via2Bnd            t t t nil nil )
( via3            boundary via3Bnd            t t t nil nil )
( hdrc            boundary hdrcBnd            t nil t nil nil )

( mcm0            boundary mcm0Bnd             t t t nil nil )
( mcm1            boundary mcm1Bnd             t t t nil nil )
( mcm2            boundary mcm2Bnd             t t t nil nil )
( mcm3            boundary mcm3Bnd             t t t nil nil )
( mcm4            boundary mcm4Bnd             t t t nil nil )
( mcmV1           boundary mcmV1Bnd            t t t nil nil )
( mcmV2           boundary mcmV2Bnd            t t t nil nil )
( mcmV3           boundary mcmV3Bnd            t t t nil nil )
( mcmV4           boundary mcmV4Bnd            t t t nil nil )

( Unrouted        drawing  Unrouted          t t t t nil )
( Unrouted        drawing1 Unrouted1         t t t t nil )
( Unrouted        drawing2 Unrouted2         t t t t nil )
( Unrouted        drawing3 Unrouted3         t t t t nil )
( Unrouted        drawing4 Unrouted4         t t t t nil )
( Unrouted        drawing5 Unrouted5         t t t t nil )
( Unrouted        drawing6 Unrouted6         t t t t nil )
( Unrouted        drawing7 Unrouted7         t t t t nil )
( Unrouted        drawing8 Unrouted8         t t t t nil )
( Unrouted        drawing9 Unrouted9         t t t t nil )
) ;techDisplays

techLayerProperties(
;( PropName          Layer1 [ Layer2 ]          PropValue )
( sheetResistance   nwell              1355 )
( sheetResistance   poly                3 )
( areaCap           default            poly                90 )
( areaCap           default            metall1            29 )
( areaCap           default            metal3             7 )
( areaCap           default            metal4             5 )
( areaCap           default            metal2             13 )
( perimeterCap     default            metall1            42 )
( perimeterCap     default            metal3             19 )
( perimeterCap     default            metal4             26 )
( perimeterCap     default            metal2             21 )
( areaCap           active             metall1            44 )
( areaCap           active             metal3             18 )
( areaCap           active             metal4             10 )

```

```

( areaCap          active      metal2          44 )
( areaCap          poly        metal1          67 )
( areaCap          poly        metal3          10 )
( areaCap          poly        metal4           7 )
( areaCap          poly        metal2          17 )
( perimeterCap     poly        metal1          60 )
( perimeterCap     poly        metal3          30 )
( perimeterCap     poly        metal4          24 )
( perimeterCap     poly        metal2          39 )
( areaCap          metall      metal3          14 )
( areaCap          metall      metal4           9 )
( areaCap          metall      metal2          37 )
( perimeterCap     metall      metal3          36 )
( perimeterCap     metall      metal4          27 )
( perimeterCap     metall      metal2          62 )
( areaCap          metal3      metal2          37 )
( areaCap          metal3      metal4          48 )
( perimeterCap     metal3      metal2          57 )
( perimeterCap     metal3      metal4          48 )
( areaCap          metal4      metal2          14 )
( perimeterCap     metal4      metal2          35 )
)

) ;layerDefinitions

;*****
; LAYER RULES
;*****
layerRules(

viaLayers(
;( layer1          viaLayer      layer2          )
;( -----        - - - - -      - - - - -      )
( poly            cp             metal1          )
( poly            cc             metal1          )
( active          ca             metal1          )
( active          cc             metal1          )
( nactive         ca             metal1          )
( nactive         cc             metal1          )
( pactive         ca             metal1          )
( pactive         cc             metal1          )
( metall          via            metal2          )
( metal2          via2           metal3          )
( metal3          via3           metal4          )
( mcm0            mcmV1          mcm2            )
( mcm1            mcmV2          mcm2            )
( mcm2            mcmV3          mcm3            )
( mcm3            mcmV4          mcm4            )

) ;viaLayers

streamLayers(
;( layer          streamNumber    dataType          translate )
;( -----        - - - - -      - - - - -      - - - - - )
( gwell          53              0                t          )
( nwell          42              0                t          )
( pwell          41              0                t          )
( active         43              0                t          )
( nactive        43              0                t          )
( pactive        43              0                t          )
( gselect       54              0                t          )
( nselect       45              0                t          )
( pselect       44              0                t          )
( poly          46              0                t          )
( metall        49              0                t          )
( ca            25              0                t          )
( cp            25              0                t          )

```

```

( cc          25          0          t          )
( metal2     51          0          t          )
( via        50          0          t          )
( glass      52          0          t          )
( pad        26          0          t          )
( metal3     62          0          t          )
( via2       61          0          t          )
( metal4     31          0          t          )
( via3       30          0          t          )
( mcm0       47          0          0          t          )
( mcmV1      48          0          0          t          )
( mcm1       55          0          0          t          )
( mcmV2      56          0          0          t          )
( mcm2       57          0          0          t          )
( mcmV3      58          0          0          t          )
( mcm3       59          0          0          t          )
( mcmV4      60          0          0          t          )
( mcm4       62          0          0          t          )
) ;streamLayers

) ;layerRules

;*****
; PHYSICAL RULES
;*****
physicalRules(
  orderedSpacingRules(
    ;( rule          layer1          layer2          value          )
    ;( ----          -)          -)          -)          )
    ( minEnclosure  "nselect"      "active"      (times 0.2 2.0)
      )
    ( minEnclosure  "pselect"      "active"      (times 0.2 2.0)
      )
    ( minEnclosure  "nselect"      "ca"          (times 0.2 1.0)
      )
    ( minEnclosure  "pselect"      "ca"          (times 0.2 1.0)
      )
    ( minEnclosure  "nselect"      "cc"          (times 0.2 1.0)
      )
    ( minEnclosure  "pselect"      "cc"          (times 0.2 1.0)
      )
    ( minEnclosure  "active"        "ca"          (times 0.2 1.0)
      )
    ( minEnclosure  "active"        "cc"          (times 0.2 1.0)
      )
    ( minEnclosure  "poly"          "cp"          (times 0.2 1.0)
      )
    ( minEnclosure  "poly"          "cc"          (times 0.2 1.0)
      )
    ( minEnclosure  "active"        "via"         (times 0.2 2.0)
      )
    ( minEnclosure  "poly"          "via"         (times 0.2 2.0)
      )
    ( minEnclosure  "metal1"        "ca"          (times 0.2 1.0)
      )
    ( minEnclosure  "metal1"        "cp"          (times 0.2 1.0)
      )
    ( minEnclosure  "metal1"        "cc"          (times 0.2 1.0)
      )
    ( minEnclosure  "metal1"        "via"         (times 0.2 1.0)
      )
    ( minEnclosure  "metal2"        "via"         (times 0.2 1.0)
      )
    ( minEnclosure  "nwell"         "nactive"     (times 0.2
3.0)
      )
    ( minEnclosure  "pwell"         "pactive"     (times 0.2
3.0)
      )
  )
)

```

```

    ( minEnclosure      "nwell"                "active"      (times 0.2
6.0)    )
    ( minEnclosure      "pwell"                "active"      (times 0.2
6.0)    )
    ( minEnclosure      "metal2"              "via2"        (times 0.2 1.0)
    )
    ( minEnclosure      "metal3"              "via2"        (times 0.2 1.0)
    )
    ( minEnclosure      "metal3"              "via3"        (times 0.2 1.0)
    )
    ( minEnclosure      "metal4"              "via3"        (times 0.2 2.0)
    )

    ( minEnclosure      "mcm1"                "mcmV1"
(40)    ) ;new (Here we actually want a hole and mcm is a dig hole layer)
    ( minEnclosure      "mcm2"                "mcmV1"
(20)    ) ;new
    ( minEnclosure      "dummymcm1"          "mcmV2"
(20)    ) ;new (the dummymcm1 layer represent a solid base)
    ( minEnclosure      "mcm2"                "mcmV2"
(20)    ) ;new
    ( minEnclosure      "mcm2"                "mcmV3"
(20)    ) ;new
    ( minEnclosure      "mcm3"                "mcmV3"
(20)    ) ;new
    ( minEnclosure      "mcm3"                "mcmV4"
(20)    ) ;new
    ( minEnclosure      "mcm4"                "mcmV4"
(40)    ) ;new

) ;orderedSpacingRules

spacingRules(
;( rule          layer1          layer2          value          )
;( ----          - - - - -      - - - - -      - - - - -      )
    ( minNotch    "nwell"                (times 0.2 6.0)
    )
    ( minNotch    "pwell"                (times 0.2
6.0)    )
    ( minSpacing  "active"                (times 0.2 3.0)    )
    ( minNotch    "active"                (times 0.2 3.0)    )
    ( minWidth    "active"                (times 0.2 3.0)    )
    ( minSpacing  "nselect"               (times 0.2 2.0)    )
    ( minSpacing  "pselect"               (times 0.2 2.0)    )
    ( minNotch    "nselect"               (times 0.2 2.0)    )
    ( minNotch    "pselect"               (times 0.2 2.0)    )
    ( minWidth    "nselect"               (times 0.2 2.0)    )
    ( minWidth    "pselect"               (times 0.2 2.0)    )
    ( minWidth    "poly"                  (times 0.2 2.0)    )
    ( minSpacing  "metal1"                (times 0.2 3.0)    )
    ( minNotch    "metal1"                (times 0.2 3.0)    )
    ( minWidth    "metal1"                (times 0.2 3.0)    )
    ( minWidth    "metal2"                (times 0.2 3.0)    )
    ( minSpacing  "via"                    (times 0.2 3.0)    )
    ( minSpacing  "metal2"                (times 0.2 3.0)    )
    ( minNotch    "metal2"                (times 0.2 3.0)    )
    ( minSpacing  "ca"                    (times 0.2 3.0)    )
    ( minSpacing  "cp"                    (times 0.2 3.0)    )
    ( minSpacing  "cc"                    (times 0.2 3.0)    )
    ( minSpacing  "poly"                  (times 0.2 3.0)    )
    ( minWidth    "nwell"                (times 0.2 12.0)   )
    ( minWidth    "pwell"                (times 0.2 12.0)   )
    ( minSpacing  "nwell"                (times 0.2 18.0)   )
    ( minSpacing  "pwell"                (times 0.2 18.0)   )
    ( minSpacing  "metal4"                (times 0.2 6.0)    )
    ( minNotch    "metal4"                (times 0.2 6.0)    )
    ( minWidth    "metal4"                (times 0.2 6.0)    )

```

```

    ( minSpacing      "via3"      (times 0.2 4.0)      )
    ( minSpacing      "metal3"     (times 0.2 3.0)      )
    ( minNotch        "metal3"     (times 0.2 3.0)      )
    ( minWidth        "metal3"     (times 0.2 3.0)      )
    ( minSpacing      "via2"      (times 0.2 3.0)      )
    ( minSpacing      "nwell"      "active"      (times 0.2 3.0)
      )
    ( minSpacing      "nwell"      "nactive"     (times 0.2 5.0)
      )
    ( minSpacing      "poly"       "ca"         (times 0.2 2.0)      )
    ( minSpacing      "poly"       "cc"         (times 0.2 2.0)      )

    ( minSpacing      "mcm1"       50           ) ;new (from
DRC specs)
    ( minWidth        "mcm1"       25           ) ;new
    ( minSpacing      "mcm2"       40           ) ;new
    ( minNotch        "mcm2"       (times 0.2 3.0)      )
;new
    ( minWidth        "mcm2"       16           ) ;new

    ( minSpacing      "mcm3"       40           ) ;new
    ( minNotch        "mcm3"       (times 0.2 3.0)      )
;new
    ( minWidth        "mcm3"       16           ) ;new

    ( minSpacing      "mcm4"       20           ) ;new
    ( minNotch        "mcm4"       (times 0.2 3.0)      )
;new
    ( minWidth        "mcm4"       50           ) ;new

    ( minSpacing      "mcmV1"      80           ) ;new
(from DRC specs)
    ( minWidth        "mcmV1"      19.6        ) ;new

    ( minSpacing      "mcmV2"      80           ) ;new
(from DRC specs)
    ( minWidth        "mcmV2"      19.6        ) ;new

    ( minSpacing      "mcmV3"      50           ) ;new
(from DRC specs)
    ( minWidth        "mcmV3"      19.6        ) ;new
    ( minSpacing      "mcmV4"      50           ) ;new
(from DRC specs)
    ( minWidth        "mcmV4"      19.6        ) ;new
) ;spacingRules
mfgGridResolution(
    ( (times 0.2 0.5) )
) ;mfgGridResolution
) ;physicalRules
;*****
; DEVICES
;*****
devices(
tcCreateCDSDeviceClass()
;
; no syEnhancement devices
;
;
; no syDepletion devices
;
symContactDevice(
; (name viaLayer viaPurpose layer1 purpose1 layer2 purpose2
; w l (row column xPitch yPitch xBias yBias) encByLayer1 encByLayer2
legalRegion)
    (M1_P cc drawing pactive drawing (pselect drawing 0.4) metall drawing
    0.4 0.4 (1 1 1 1 center center) 0.2 0.2 (inside nwell drawing))
    (M1_N cc drawing nactive drawing (nselect drawing 0.4) metall drawing

```

```

0.4 0.4 (1 1 1 1 center center) 0.2 0.2 (outside nwell drawing))
(NTAP cc drawing
  nwell drawing (nselect drawing -0.2)
  metall drawing (nactive drawing 0)
  0.4 0.4 (1 1 1 1 center center) 0.8 0.2 (inside nwell drawing))
(M1_POLY cc drawing poly drawing metall drawing
0.4 0.4 (1 1 1 1 center center) 0.2 0.2 _NA_)

(M2_M1 via drawing metall drawing metal2 drawing
0.4 0.4 (1 1 1 1 center center) 0.2 0.2 _NA_)

(M4_M3 via3 drawing metal3 drawing metal4 drawing
0.4 0.4 (1 1 1.2 1.2 center center) 0.2 0.4 _NA_)

(M3_M2 via2 drawing metal2 drawing metal3 drawing
0.4 0.4 (1 1 1 1 center center) 0.2 0.2 _NA_)

(mcm2_mcm0 mcmV1 drawing ;new
  mcm2 drawing ;new
  mcm0 drawing (mcm1 drawing 20) ;new
20 20 (1 1 1 1 center center) 20 20 _NA_) ;new

(mcm2_mcm1 mcmV2 drawing dummysmcm1 drawing mcm2 drawing ;new
20 20 (1 1 1 1 center center) 20 20 _NA_) ;new

(mcm3_mcm2 mcmV3 drawing mcm2 drawing mcm3 drawing ;new
20 20 (1 1 1 1 center center) 20 20 _NA_) ;new

(mcm4_mcm3 mcmV4 drawing mcm3 drawing mcm4 drawing ;new
20 20 (1 1 1 1 center center) 20 40 _NA_) ;
)
symPinDevice(
; (name maskable layer1 purpose1 w1 layer2 purpose2 w2 legalRegion)
(nwell nil nwell drawing 0.4 _NA_ _NA_ _NA_ _NA_)
(nactive nil nactive drawing 0.4 _NA_ _NA_ _NA_ _NA_)
(pactive nil pactive drawing 0.4 _NA_ _NA_ _NA_ _NA_)
(active nil active drawing 0.4 _NA_ _NA_ _NA_ _NA_)
(poly nil poly drawing 0.4 _NA_ _NA_ _NA_ _NA_)
(metall nil metall drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(metal2 nil metal2 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(metal3 nil metal3 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(metal4 nil metal4 drawing 0.6 _NA_ _NA_ _NA_ _NA_)

(mcm0 nil mcm0 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(mcm1 nil mcm1 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(mcm2 nil mcm2 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(mcm3 nil mcm3 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
(mcm4 nil mcm4 drawing 0.6 _NA_ _NA_ _NA_ _NA_)
)
;
; no ruleContact devices
;
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
; Opus Symbolic Device Class Definition
;
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
; no other device classes
;
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
; Opus Symbolic Device Declaration
;
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;

```

```

; no other devices
;
) ;devices
;*****
; LX RULES
;*****
lxRules(

  lxExtractLayers(
    ;( list of layers or layer/purpose pairs )
    ;( ----- )
      ( gwell          pwell          nwell          active          nactive
        pactive        metall         metal2        via              ca              cp
          cc           metal3         metal4        mcm0             mcm1
mcm2                mcm3             mcm4)
    ) ;lxExtractLayers

  lxNoOverlapLayers(
    ;( forbidden overlaps in LX )
    ;( ----- )
      ( poly          active      )
      ( poly          nactive     )
      ( poly          pactive     )
      ( via           ca          )
      ( via           cp          )
      ( via           cc          )
    ) ;lxNoOverlapLayers

) ;lxRules

```

Appendix B - display.drf file

```
;display.drf for the SHOCC package.  
; Last updated Nov 29 by akvarma  
; Last updated Dec 18 by akvarma
```

```
drDefinePacket(  
( DisplayName PacketName Stipple LineStyle Fill Outline )  
( display mcm0 dot4 solid slate slate )  
( display mcm0Net blank solid slate slate )  
( display mcm0Pin X solid slate slate )  
( display mcm0Llbl blank solid slate slate )  
( display mcm0Bnd blank solid slate slate )  
(display d_mcm1 blank solid slate maroon )  
( display mcm1 dot3 solid silver silver )  
( display mcm1Net blank solid silver silver )  
( display mcm1Pin X solid silver silver )  
( display mcm1Llbl blank solid silver silver )  
( display mcm1Bnd blank solid silver silver )  
( display mcm2 cross solid brown brown )  
( display mcm2Net blank solid brown brown )  
( display mcm2Pin X solid brown brown )  
( display mcm2Llbl blank solid brown brown )  
( display mcm2Bnd blank solid brown brown )  
( display mcm3 halfslash solid yellow gold )  
( display mcm3Net blank solid yellow gold )  
( display mcm3Pin X solid yellow gold )  
( display mcm3Llbl blank solid yellow gold )  
( display mcm3Bnd blank solid yellow gold )  
( display mcm4 triangle solid orange brown )  
( display mcm4Net blank solid orange brown )  
( display mcm4Pin X solid orange brown )  
( display mcm4Llbl blank solid orange brown )  
( display mcm4Bnd blank solid orange brown )  
( display mcmV1 invCross solid maroon brown )  
( display mcmV1Net blank solid maroon brown )  
( display mcmV1Pin X solid maroon brown )  
( display mcmV1Llbl blank solid maroon brown )  
( display mcmV1Bnd blank solid maroon brown )  
( display mcmV2 solid solid red maroon )  
( display mcmV2Net blank solid brown maroon )  
( display mcmV2Pin X solid brown maroon )  
( display mcmV2Llbl blank solid brown maroon )  
( display mcmV2Bnd blank solid brown maroon )  
( display mcmV3 solid solid yellow maroon )  
( display mcmV3Net blank solid purple maroon )  
( display mcmV3Pin X solid purple maroon )  
( display mcmV3Llbl blank solid purple maroon )  
( display mcmV3Bnd blank solid purple maroon )  
( display mcmV4 solid solid orange purple )  
( display mcmV4Net blank solid maroon purple )  
( display mcmV4Pin X solid maroon purple )  
( display mcmV4Llbl blank solid maroon purple )  
( display mcmV4Bnd blank solid maroon purple )  
)
```

Appendix C - divaDRC.rul file

```
; divaDRC rule file for the SHOCC package
; last updated Dec 20, 2000 by akvarma

drcExtractRules(
ivIf(switch("drc?") then

/*****
*****

DERIVED LAYER DEFINITIONS

*****
*****/

mcm0 = geomOr("mcm0")
mcmV1 = geomOr("mcmV1")
mcm1 = geomOr("mcm1")
mcmV2 = geomOr("mcmV2")
mcm2 = geomOr("mcm2")
mcmV3 = geomOr("mcmV3")
mcm3 = geomOr("mcm3")
mcmV4 = geomOr("mcmV4")
mcm4 = geomOr("mcm4")

/* **Get edges for enclosure checks ** */
/*****      Metal      *****/
mcm1Edge = geomGetEdge(mcm1)
mcm2Edge = geomGetEdge(mcm2)
mcm3Edge = geomGetEdge(mcm3)
mcm4Edge = geomGetEdge(mcm4)

/*****      Via      *****/
mcmV1Edge = geomGetEdge(mcmV1)
mcmV2Edge = geomGetEdge(mcmV2)
mcmV3Edge = geomGetEdge(mcmV3)
mcmV4Edge = geomGetEdge(mcmV4)
/***** Overlap Area Check*****/
/*??*****/More derived Layers*****/ ??*/

V1M0 = geomAnd(mcmV1 mcm0)
V2M0 = geomAnd(mcmV2 mcm0)
V3M0 = geomAnd(mcmV3 mcm0)
V4M0 = geomAnd(mcmV4 mcm0)
NV1M1 = geomAndNot(mcmV1 mcm1)
V2M1 = geomAnd(mcmV2 mcm1)
V3M1 = geomAnd(mcmV3 mcm1)
V4M1 = geomAnd(mcmV4 mcm1)
NV2M2 = geomAndNot(mcmV2 mcm2)
NV3M2 = geomAndNot(mcmV3 mcm2)
NV3M3 = geomAndNot(mcmV3 mcm3)
NV4M2 = geomAndNot(mcmV4 mcm2)
NV4M3 = geomAndNot(mcmV4 mcm3)
NV4M4 = geomAndNot(mcmV4 mcm4)
V1V2 = geomAnd(mcmV1 mcmV2)
V1V3 = geomAnd(mcmV1 mcmV3)
V1V4 = geomAnd(mcmV1 mcmV4)
V2V3 = geomAnd(mcmV2 mcmV3)
V2V4 = geomAnd(mcmV2 mcmV4)
V3V4 = geomAnd(mcmV3 mcmV4)

/*****      *****/
*****/DRC Rule Checks*****/

;19.0 Ground plane(mcm0)
```

```

L19DR1 = drc(mcm0 sep < 50.00 )
saveDerived(L19DR1 "mcm0 separation < 50.00")
L19DR2 = drc(mcm0 width < 25.00 )
saveDerived(L19DR2 "mcm0 width < 25.00")

L19DR3A = drc(mcmV1 mcm0 sep < 30.00 )
saveDerived(L19DR3A "mcmV1 to mcm0 separation <30.00")
L19DR3B = geomOverlap(mcmV1 mcm0)
saveDerived(L19DR3B "mcmV1 and mcm0 overlapping!!")

L19DR4A = drc(mcmV2 mcm0 sep < 30.00 )
saveDerived(L19DR4A "mcmV2 to mcm0 separation <30.00")
L19DR4B = geomOverlap(mcmV2 mcm0)
saveDerived(L19DR4B "mcmV2 and mcm0 overlapping!!")

L19DR5A = drc(mcmV3 mcm0 sep < 30.00 )
saveDerived(L19DR5A "mcmV3 to mcm0 separation <30.00")
L19DR5B = geomOverlap(mcmV3 mcm0)
saveDerived(L19DR5B "mcmV3 and mcm0 overlapping!!")

L19DR6A = drc(mcmV4 mcm0 sep < 30.00 )
saveDerived(L19DR6A "mcmV4 to mcm0 separation <30.00")
L19DR6B = geomOverlap(mcmV4 mcm0)
saveDerived(L19DR6B "mcmV4 and mcm0 overlapping!!")

;20.0 Vial (mcmV1)

L20DR1 = drc(mcmV1 sep < 80.00 )
saveDerived(L20DR1 "mcmV1 separation < 80.00")

L20DR2A = drc(mcmV1 mcmV2 sep < 30.00 )
saveDerived(L20DR2A "mcmV1 to mcmV2 separation <30.00")
L20DR2B = geomOverlap(mcmV1 mcmV2)
saveDerived(L20DR2B "mcmV1 and mcmV2 overlapping!!")

L20DR3 = drc(mcmV1 width < 19.60 )
saveDerived(L20DR3 "mcmV1 width < 19.60")

L20DR4A = drc(mcmV1 mcmV3 sep < 30.00 )
saveDerived(L20DR4A "mcmV1 to mcmV3 separation <30.00")
L20DR4B = geomOverlap(mcmV1 mcmV3)
saveDerived(L20DR4B "mcmV1 and mcmV3 overlapping!!")

L20DR5A = drc(mcmV1 mcmV4 sep < 30.00 )
saveDerived(L20DR5A "mcmV1 to mcmV4 separation <30.00" )
L20DR5B = geomOverlap(mcmV1 mcmV4)
saveDerived(L20DR5B "mcmV1 and mcmV4 overlapping!!")

;21.0 Power plane(mcm1)

L21DR1 = drc(mcm1 sep < 50.00 )
saveDerived(L21DR1 "mcm1 separation < 50.00")
L21DR2 = drc(mcm1 width < 25.0 )
saveDerived(L21DR2 "mcm1 width < 25.0")

L21DR3A = drc(mcm1Edge mcmV1Edge enc < 30.0 )
saveDerived(L21DR3A "mcm1 and mcmV1 enclosure check (mcmV1 should be
enclosed by mcm1 by 30 u on each sides)")
L21DR3B = geomAndNot(mcmV1 mcm1)
saveDerived(L21DR3B "mcm1 and mcmV1 enclosure check (mcmV1 is not completely
covered by mcm1)")

L21DR4A = drc(mcmV2 mcm1 sep < 30.00 )
saveDerived(L21DR4A "mcmV2 to mcm1 separation <30.00")
L21DR4B = geomOverlap(mcmV2 mcm1)
saveDerived(L21DR4B "mcmV2 and mcm1 overlapping!!")

```

```

L21DR5A = drc(mcmV3 mcm1 sep < 30.00 )
saveDerived(L21DR5A "mcmV3 to mcm1 separation <30.00")
L21DR5B = geomOverlap(mcmV3 mcm1)
saveDerived(L21DR5B "mcmV3 and mcm1 overlapping!!")

L21DR6A = drc(mcmV4 mcm1 sep < 30.00 )
saveDerived(L21DR6A "mcmV4 to mcm1 separation <30.00")
L21DR6B = geomOverlap(mcmV4 mcm1)
saveDerived(L21DR6B "mcmV4 and mcm1 overlapping!!")

;22.0 Via2 (mcmV2)

L22DR1 = drc(mcmV2 sep < 80.00 )
saveDerived(L22DR1 "mcmV2 separation < 80.00")
L22DR2A = drc(mcmV2 mcmV3 sep < 30.00 )
saveDerived(L22DR2A "mcmV2 to mcmV3 separation <30.00")
L22DR2B = geomOverlap(mcmV2 mcmV3)
saveDerived(L22DR2B "mcmV2 and mcmV3 overlapping!!")

L22DR3 = drc(mcmV2 width < 19.6)
saveDerived(L22DR3 "mcmV2 width < 19.6")
L22DR4A = drc(mcmV2 mcmV4 sep < 30.00 )
saveDerived(L22DR4A "mcmV2 to mcmV4 separation <30.00")
L22DR4B = geomOverlap(mcmV2 mcmV4)
saveDerived(L22DR4B "mcmV2 and mcmV4 overlapping!!")

;23.0 X Metal(mcm2)

mcm2SEP = drc(mcm2 sep<40.0)
L23DR1 = geomGetLength(mcm2SEP length > 600.0)
saveDerived(L23DR1 "Error Flags length > 600.0")

L23DR2 = drc(mcm2 sep < 19.40 )
saveDerived(L23DR2 "mcm2 separation < 19.40")

L23DR3A = drc(mcm2Edge mcmV2Edge enc < 19.4)
saveDerived(L23DR3A "mcm2 and mcmV2 enclosure check (mcmV2 should be
enclosed by mcm2 by 19.4 u on each sides)")
L23DR3B = geomAndNot(mcmV2 mcm2)
saveDerived(L23DR3B "mcm2 and mcmV2 enclosure check (mcmV2 is not completely
covered by mcm2)")

L23DR4 = drc(mcm2 width < 16.0)
saveDerived(L23DR4 "mcm2 width < 16.0")

L23DR5A = drc(mcm2Edge mcmV3Edge enc < 19.4 )
saveDerived(L23DR5A "mcm2 and mcmV3 enclosure check (mcmV3 should be
enclosed by mcm2 by 19.4 u on each sides)")
L23DR5B = geomAndNot(mcmV3 mcm2)
saveDerived(L23DR5B "mcm2 and mcmV3 enclosure check (mcmV3 is not completely
covered by mcm2)")

;24.0 Via3 (mcmV3)

L24DR1 = drc(mcmV3 sep < 50.00 )
saveDerived(L24DR1 "mcmV3 separation < 50.00")

L24DR2A = drc(mcmV3 mcmV4 sep < 30.00 )
saveDerived(L24DR2A "mcmV3 to mcmV4 separation <30.00")
L24DR2B = geomOverlap(mcmV3 mcmV4)
saveDerived(L24DR2B "mcmV3 and mcmV4 overlapping!!")

L24DR3 = drc(mcmV3 width < 19.6 )
saveDerived(L24DR3 "mcmV3 width < 19.6")

;25.0 Y Metal(mcm3)

```

```

mcm3SEP = drc(mcm3 sep<40.0)
L25DR1 = geomGetLength(mcm3SEP length > 450.0)
saveDerived(L25DR1 "Error Flags length > 450.0")

L25DR2 = drc(mcm3 sep < 19.40 )
saveDerived(L25DR2 "mcm3 separation < 19.40")
L25DR3A = drc(mcm3Edge mcmV3Edge enc < 19.4 )
saveDerived(L25DR3A "mcm3 and mcmV3 enclosure check (mcmV3 should be
enclosed by mcm3 by 19.4 u on each sides)")
L25DR3B = geomAndNot(mcmV3 mcm3)
saveDerived(L25DR3B "mcm3 and mcmV3 enclosure check (mcmV3 is not completely
covered by mcm3)")

L25DR4 = drc(mcm3 width < 16.0 )
saveDerived(L25DR4 "mcm3 width < 16.0")

L25DR5A = drc(mcm3Edge mcmV4Edge enc < 19.4 )
saveDerived(L25DR5A "mcm3 and mcmV4 enclosure check (mcmV4 should be
enclosed by mcm3 by 19.4 u on each sides)")
L25DR5B = geomAndNot(mcmV4 mcm3)
saveDerived(L25DR5B "mcm3 and mcmV4 enclosure check (mcmV4 is not completely
covered by mcm3)")

;26.0 Via4 (mcmV4)

L26DR1 = drc(mcmV4 sep < 50.00 )
saveDerived(L26DR1 "mcmV4 separation < 50.00")

L26DR2 = drc(mcmV4 width < 19.6 )
saveDerived(L26DR2 "mcmV4 width < 19.6")

;27.0 Top Pad Metal(mcm4)

L27DR1 = drc(mcm4 sep < 20.0 )
saveDerived(L27DR1 "mcm4 separation < 20.0")

L27DR2A = drc(mcm4Edge mcmV4Edge enc < 35.0 )
saveDerived(L27DR2A "mcm4 and mcmV4 enclosure check (mcmV4 should be
enclosed by mcm4 by 35.4 u on each sides)")
L27DR2B = geomAndNot(mcmV4 mcm4)
saveDerived(L27DR2B "mcm4 and mcmV4 enclosure check (mcmV4 is not completely
covered by mcm4)")

L27DR3 = drc(mcm4 width < 50.0 )
saveDerived(L27DR3 "mcm4 width < 50.0")

; These are illegal geometry checks.

L19A0=drc(mcm0 0 < area < 8200.0)
L20A0=drc(mcmV1 0 < area < 320.0)
L20A1=drc(mcmV1 0 < area < 320.0)
L21A0=drc(mcm1 0 < area < 7250.0)
L22A0=drc(mcmV2 0 < area < 320.0)
L22A1=drc(mcmV2 0 < area < 320.0)
L23A0=drc(mcm2 0 < area < 2900.0)
L24A0=drc(mcmV3 0 < area < 320.0)
L24A1=drc(mcmV3 0 < area < 320.0)
L25A0=drc(mcm3 0 < area < 2900.0)
L26A0=drc(mcmV4 0 < area < 320.0)
L26A1=drc(mcmV4 0 < area < 320.0)
L27A0=drc(mcm4 0 < area < 10000.0)

L19A1=drc(V1M0 0 < area < 99999.0)
L19A2=drc(V2M0 0 < area < 99999.0)
L19A3=drc(V3M0 0 < area < 99999.0)

```

L19A4=drc(V4M0 0 < area < 99999.0)
L20A2=drc(NV1M1 0 < area < 99999.0)
L22A1=drc(V2M1 0 < area < 99999.0)
L24A1=drc(V3M1 0 < area < 99999.0)
L26A1=drc(V4M1 0 < area < 99999.0)
L22A2=drc(NV2M2 0 < area < 99999.0)
L24A2=drc(NV3M2 0 < area < 99999.0)
L24A3=drc(NV3M3 0 < area < 99999.0)
L26A2=drc(NV4M2 0 < area < 99999.0)
L26A3=drc(NV4M3 0 < area < 99999.0)
L26A4=drc(NV4M4 0 < area < 99999.0)
L20A3=drc(V1V2 0 < area < 99999.0)
L20A4=drc(V1V3 0 < area < 99999.0)
L20A5=drc(V1V4 0 < area < 99999.0)
L22A3=drc(V2V3 0 < area < 99999.0)
L22A4=drc(V2V4 0 < area < 99999.0)
L24A4=drc(V3V4 0 < area < 99999.0)
)
)

Appendix D - D Extraction Rule File

```
/*
*
*                               Diva Extraction Rules                               *
*
*/

let( ( lambda gridRes techdesc wellType submicronAvailable
      cwellAvailable metal3Available metal4Available metal5Available
      metal6Available metalcapAvailable npnAvailable ccdAvailable
elecAvailable
      sblockAvailable PadType highresAvailable stackedViasAvailable
techfile
      modelPrefix errMesg useCapIVPcell )
;load( prependNCSUCDKInstallPath( "techfile/divaTechdesc.il" ) )
techdesc="HP_AMOS14TB"
drcExtractRules(
load( "/afs/eos.ncsu.edu/project/erl/paulf_group/user/akvarma/SHOCC/EXT/shocc
LayerDef.il" )
;-----
      ; shocc line
      ;-----
bump=geomOr(mcm4)
extractDevice( bump metal3("IN") mcm4("OUT") "bump ivpcell TEST" physical)
shoccY=geomOr(mcm3)
      shoccX=geomOr(mcm2)
      groundY=geomAndNot(pBulk mcm3)
      groundX=geomAndNot(pBulk mcm2)
      extractDevice( shoccY mcmV4("in") pBulk("refin") groundY("refout")
shoccXYterminal("out") "ulwireSY ivpcell TEST" physical)
      shoccLengthY=measureParameter( perimeter shoccY 0.5e-6)
      saveParameter( shoccLengthY "1")
      extractDevice( shoccX shoccXYterminal("in") pBulk("refin")
groundX("refout") shoccXYterminal("out") "ulwireSX ivpcell TEST" physical)
      shoccLengthX=measureParameter( perimeter shoccX 0.5e-6)
      saveParameter( shoccLengthX "1")
      ;-----
      ; mosfets
      ;-----
      extractMOS( nChannelTran poly("G") nDiff("S" "D") pBulk("B") "nmos4
ivpcell
NCSU_Analog_Parts" "l" "w" 1.0e-6 )
      saveProperty( nChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["nmos"] ) )
      extractMOS( pChannelTran poly("G") pDiff("S" "D") nBulk("B") "pmos4
ivpcell
NCSU_Analog_Parts" "l" "w" 1.0e-6 )
      saveProperty( pChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["pmos"] ) )
      if( hvAvailable then
          extractMOS( hvnChannelTran poly("G") nDiff("S" "D") pBulk("B")
"nmos4_hv ivpcell NCSU_Analog_Parts" "l" "w" 1.0e-6 )
          saveProperty( hvnChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["nmos_hv"] ) )
          extractMOS( hvpChannelTran poly("G") pDiff("S" "D") nBulk("B")
"pmos4_hv ivpcell NCSU_Analog_Parts" "l" "w" 1.0e-6 )
          saveProperty( hvpChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["pmos_hv"] ) )
      )

      if( elecAvailable then
          unless( member( techdesc list( "TSMC_CMOS035_4M2P"
                                          "TSMC_CMOS035_3M2P"
                                          "AMI_C5N"
                                          ) )
          ) )
      ) )
```

```

        extractMOS( nElecChannelTran elec( "G" ) nDiff( "S" "D" ) pBulk("B" )
                    "nmos4_elec ivpcell NCSU_Analog_Parts" "1" "w" 1.0e-6 )
        saveProperty( nElecChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["nmos_elec"] ) )

        extractMOS( pElecChannelTran elec( "G" ) nDiff( "S" "D" ) nBulk("B" )
                    "pmos4_elec ivpcell NCSU_Analog_Parts" "1" "w" 1.0e-6 )
        saveProperty( pElecChannelTran "model" strcat( modelPrefix
NCSU_modelSuffix["pmos_elec"] ) )
    )
    nDiffArea = measureParasitic( area (nDiff not_over nolpe) 1.0e-12 figure
)
    attachParasitic( nDiffArea ("as" "S") ("ad" "D") nChannelTran shared )
    pDiffArea = measureParasitic( area (pDiff not_over nolpe) 1.0e-12 figure
)
    attachParasitic( pDiffArea ("as" "S") ("ad" "D") pChannelTran shared )
    if( hvAvailable then
        attachParasitic( nDiffArea ("as" "S") ("ad" "D") hvnChannelTran shared
)
        attachParasitic( pDiffArea ("as" "S") ("ad" "D") hvpChannelTran shared
)
    )
; could have problem here if there's a process w/both elec and hv,
; and it allows elec transistors

    if( elecAvailable && (member( techdesc list( "TSMC_CMOS035_4M2P"
"TSMC_CMOS035_3M2P" "AMI_C5N" ) ) == nil) then
        attachParasitic( nDiffArea ("as" "S") ("ad" "D") nElecChannelTran shared )
        attachParasitic( pDiffArea ("as" "S") ("ad" "D") pElecChannelTran shared )

        nDiffPerimeter = measureParasitic( length (nDiff outside poly
outside elec not_over nolpe) 1.0e-6 figure )
        attachParasitic( nDiffPerimeter ("ps" "S") ("pd" "D") nChannelTran shared )
        attachParasitic( nDiffPerimeter ("ps" "S") ("pd" "D")
nElecChannelTran shared )
        pDiffPerimeter = measureParasitic( length (pDiff outside poly
outside elec not_over nolpe) 1.0e-6 figure )
        attachParasitic( pDiffPerimeter ("ps" "S") ("pd" "D") pChannelTran shared )
        attachParasitic( pDiffPerimeter ("ps" "S") ("pd" "D") pElecChannelTran
shared )
        else
            nDiffPerimeter = measureParasitic( length (nDiff outside poly not_over
nolpe) 1.0e-6 figure )
            attachParasitic( nDiffPerimeter ("ps" "S") ("pd" "D") nChannelTran
shared )
            pDiffPerimeter = measureParasitic( length (pDiff outside poly not_over
nolpe) 1.0e-6 figure )
            attachParasitic( pDiffPerimeter ("ps" "S") ("pd" "D") pChannelTran shared )
            if( hvAvailable then
                attachParasitic( nDiffPerimeter ("ps" "S") ("pd" "D")
hvnChannelTran shared )
                attachParasitic( pDiffPerimeter ("ps" "S") ("pd" "D")
hvpChannelTran shared )
            )
        )
    )
;-----
; npns
;-----

; hardcode the NPN model name to "Generic_NPN", since generally
; different models are used for different geometry BJTs the user
; will have to edit the netlist and replace it with the correct
; model name (a tip of the hat to John Galbraith at Arizona for his
; help with this issue)

    if( npnAvailable then

```

```

        extractDevice( npnTran (npnCollector "C") (npnBaseTap "B")
(npnEmitter
"E") "npn ivpcell NCSU_Analog_Parts" )
        saveProperty( npnTran "model" "Generic_NPN" )
    )
;-----
; resistors
;-----
; poly/elec
extractDevice( polyRes (poly "PLUS" "MINUS") "res ivpcell
NCSU_Analog_Parts" )
    if( sblockAvailable then
        extractDevice( polySRes (poly "PLUS" "MINUS") "res ivpcell
NCSU_Analog_Parts" )
    )
    if( elecAvailable then
        extractDevice( elecRes (elec "PLUS" "MINUS") "res ivpcell
NCSU_Analog_Parts" )
    )
    if( highresAvailable then
        extractDevice( elecHighres (elec "PLUS" "MINUS") "res ivpcell
NCSU_Analog_Parts" )
    )
; nwell
extractDevice( nwellRes (nBulk "PLUS" "MINUS") "res ivpcell
NCSU_Analog_Parts" )
; resistor parameter extraction
let( (sheetRes resWidth resPerim resLength res fullCorners)
    ; for each full corner (ie, 90 degree turn), subtract 0.5 a square
    ; of resistance (see W&E 2nd ed, p.178)
    ; poly
    sheetRes = techGetLayerProp( techfile list( "poly" "drawing")
"sheetResistance" )
    when( sheetRes
        resWidth = measureParameter( length (polyRes butting poly) 1.0e-6 )
        resPerim = measureParameter( perimeter polyRes 1.0e-6 )
        resLength = calculateParameter( (resPerim - resWidth) / 2.0 )
        fullCorners = measureParameter( bends_full polyRes )
        res = calculateParameter( sheetRes * (resLength/(resWidth/2.0) -
fullCorners*0.5) )
        saveParameter( res "r" )
    )
)
; poly + sblock
if( sblockAvailable then
    sheetRes = techGetLayerProp( techfile list( "sblock" "drawing")
"sheetResistance" )
    when( sheetRes
        resWidth = measureParameter( length (polySRes butting poly) 1.0e-6 )
        resPerim = measureParameter( perimeter polySRes 1.0e-6 )
        resLength = calculateParameter( (resPerim - resWidth) / 2.0
)
        fullCorners = measureParameter( bends_full polySRes )
        res = calculateParameter( sheetRes * (resLength/(resWidth/2.0)
- fullCorners*0.5) )
        saveParameter( res "r" )
    )
)
; elec
if( elecAvailable then
    ; regular
    sheetRes = techGetLayerProp( techfile list( "elec" "drawing")
"sheetResistance" )
    when( sheetRes
        resWidth = measureParameter( length (elecRes butting elec) 1.0e-6 )
        resPerim = measureParameter( perimeter elecRes 1.0e-6 )
        resLength = calculateParameter( (resPerim - resWidth) / 2.0
)
)
)

```

```

        fullCorners = measureParameter( bends_full elecRes )
        res = calculateParameter( sheetRes * (resLength/(resWidth/2.0)
- fullCorners*0.5) )
        saveParameter( res "r" )
    )
; high-resistance
    if( highresAvailable then
        sheetRes = techGetLayerProp( techfile list( "highres"
"drawing") "sheetResistance" )
        when( sheetRes
            resWidth = measureParameter( length (elecHighres butting elec) 1.0e-6 )
            resPerim = measureParameter( perimeter elecHighres 1.0e-6 )
            resLength = calculateParameter( (resPerim - resWidth) / 2.0
)
                fullCorners = measureParameter( bends_full elecHighres )
                res = calculateParameter( sheetRes *
(resLength/(resWidth/2.0) - fullCorners*0.5) )
                saveParameter( res "r" )
            )
        )
sheetRes = techGetLayerProp( techfile list( "nwell" "drawing")
"sheetResistance" )
    when( sheetRes
        resWidth = measureParameter( length (nwellRes butting nBulk) 1.0e-6
)
            resPerim = measureParameter( perimeter nwellRes 1.0e-6 )
            resLength = calculateParameter( (resPerim - resWidth) / 2.0 );
            fullCorners = measureParameter( bends_full nwellRes )
            res = calculateParameter( sheetRes * (resLength/(resWidth/2.0) -
fullCorners*0.5) )
            saveParameter( res "r" )
        )
    )
;-----
; capacitors
;-----
; first we do all the caps that use process-optional layers. we
; always extract these, since we assume they're "intentional".
; poly - polycap capacitors
if( polycapAvailable then
    extractDevice( polycapCap (poly "PLUS") (polycap "MINUS") "cap ivpcell
NCSU_Analog_Parts")
        saveRecognition( polycapCap "poly" )
        let( ( areaCap capacitance )
            areaCap = techGetTwoLayerProp( techfile
                list( "polycap" "drawing")
                list( "poly" "drawing")
                "areaCap" )
            when( areaCap
                capacitance = measureParameter( area polycapCap 1.0e-18 * areaCap )
                saveParameter( capacitance "c" )
            )
        )
)
; metal - metalcap capacitors
if( metalcapAvailable then
    let( ( areaCap capacitance )
        cond(
            ( metal6Available
                extractDevice( metalcapCap (metalcap "PLUS") (metal5
"MINUS") "cap ivpcell NCSU_Analog_Parts")
                saveRecognition( metalcapCap "metalcap" )
                areaCap = techGetTwoLayerProp( techfile
                    list( "metal5" "drawing")
                    list( "metalcap" "drawing")
                    "areaCap" )

```

```

    )
    ( metal5Available
    extractDevice( metalcapCap (metalcap "PLUS") (metal4
"MINUS") "cap ivpcell NCSU_Analog_Parts")
        saveRecognition( metalcapCap "metalcap" )
        areaCap = techGetTwoLayerProp( techfile
            list( "metal4" "drawing")
            list( "metalcap" "drawing")
            "areaCap" )
    )
    )
    when( areaCap
    capacitance = measureParameter( area metalcapCap 1.0e-18 * areaCap )
    saveParameter( capacitance "c" )
    )
    )
)
; poly - cwell thinox capacitors
if( cwellAvailable then
    extractDevice( lcCap (poly "PLUS") (lcDiff "MINUS") "cap ivpcell
NCSU_Analog_Parts")
    saveRecognition( lcCap "poly" )
    let( ( areaCap capacitance )
        areaCap = techGetTwoLayerProp( techfile
            list( "poly" "drawing")
            list( "cwell" "drawing")
            "areaCap" )
        when( areaCap
            capacitance = measureParameter( area lcCap 1.0e-18 * areaCap
)
            saveParameter( capacitance "c" )
        )
    )
)
; poly - elec caps
if( elecAvailable then
    extractDevice( CapacitorElec (elec "PLUS") (poly "MINUS") "cap
ivpcell
NCSU_Analog_Parts" )
    saveRecognition( CapacitorElec "elec" )
    let( ( areaCap capacitance )
        areaCap = techGetTwoLayerProp( techfile
            list( "elec" "drawing")
            list( "poly" "drawing")
            "areaCap" )
        when( areaCap
            capacitance = measureParameter( area CapacitorElec 1.0e-18*areaCap )
            saveParameter( capacitance "c" )
        )
    )
)
; The rest of the capacitance extraction (including "intentional"
; plain old metal caps and parasitic caps) is performed in the file
; divaMultiLevel.il via the multiLevelParasitic command.
;
; The regular metal, poly, and other layers are modified below
; according to the Extract_parasitic_caps switch. It is these
; modified layers on which the multiLevelParasitic command operates.
;
; Make scale factor specify values in aF/um^2 for areacaps and aF/um
; for perimcaps. This will make it easier to read values straight
; from MOSIS parametric reports and Magic tech files.
;
; See local/techfile/layerDefintions.tf for the actual value of the
; parasitic capacitances.
; "intentional" caps (i.e. metal with cap_id over it) are always extracted
if( metal6Available then

```

```

        metal6_ext =      geomAnd( metal6 cap_id )
    )
    if( metal5Available then
        metal5_ext =      geomAnd( metal5 cap_id )
    )
    if( metal4Available then
        metal4_ext =      geomAnd( metal4 cap_id )
    )
    if( metal3Available then
        metal3_ext =      geomAnd( metal3 cap_id )
    )
    if( elecAvailable then
        elec_ext =        geomAnd( elec cap_id )
    )
    metal2_ext =      geomAnd( metal2 cap_id )
    metall_ext =      geomAnd( metall cap_id )
    poly_ext =        geomAnd( poly cap_id )
    nDiff_ext =       geomAnd( nDiff cap_id )
    pDiff_ext =       geomAnd( pDiff cap_id )
    nOhmic_ext =      geomAnd( nOhmic cap_id )
    pOhmic_ext =      geomAnd( pOhmic cap_id )
    nNotOhmic_ext =   geomAnd( nNotOhmic cap_id )
    pNotOhmic_ext =   geomAnd( pNotOhmic cap_id )
    nBulk_ext =       geomAnd( nBulk cap_id )
    pBulk_ext =       geomAnd( pBulk cap_id )
useCapIVPcell = t
load( prependNCSUCDKInstallPath( "techfile/divaMultiLevel.il" ) )
; for any time when Extract_parasitic_caps is selected. note that we
; don't want to extract any "intentional" caps as parasitics
ivIf( switch("Extract_parasitic_caps") then
if( metal6Available then
        metal6_ext =      geomAndNot( metal6 geomOr( nolpe cap_id ) )
    )
    if( metal5Available then
        metal5_ext =      geomAndNot( metal5 geomOr( nolpe cap_id ) )
    )
    if( metal4Available then
        metal4_ext =      geomAndNot( metal4 geomOr( nolpe cap_id ) )
    )
    if( metal3Available then
        metal3_ext =      geomAndNot( metal3 geomOr( nolpe cap_id ) )
    )
    if( elecAvailable then
        elec_ext =        geomAndNot( elec geomOr( nolpe cap_id ) )
    )
    metal2_ext =      geomAndNot( metal2 geomOr( nolpe cap_id ) )
    metall_ext =      geomAndNot( metall geomOr( nolpe cap_id ) )
    poly_ext =        geomAndNot( poly geomOr( nolpe cap_id ) )
    nDiff_ext =       geomAndNot( nDiff geomOr( nolpe cap_id ) )
    pDiff_ext =       geomAndNot( pDiff geomOr( nolpe cap_id ) )
    nOhmic_ext =      geomAndNot( nOhmic geomOr( nolpe cap_id ) )
    pOhmic_ext =      geomAndNot( pOhmic geomOr( nolpe cap_id ) )
    nNotOhmic_ext =   geomAndNot( nNotOhmic geomOr( nolpe cap_id ) )
    pNotOhmic_ext =   geomAndNot( pNotOhmic geomOr( nolpe cap_id ) )
    nBulk_ext =       geomAndNot( nBulk geomOr( nolpe cap_id ) )
    pBulk_ext =       geomAndNot( pBulk geomOr( nolpe cap_id ) )
useCapIVPcell = nil
load( prependNCSUCDKInstallPath( "techfile/divaMultiLevel.il" ) )
)
;-----
; diodes
;-----
;extractDevice( NPdiode (pBulk "PLUS") (nDiff "MINUS") "diode ivpcell
NCSU_Analog_Parts" )
;saveProperty( NPdiode "model" strcat( modelPrefix
NCSU_modelSuffix["npdiode"] ) )

```

```

;extractDevice( PNdiode (pDiff "PLUS") (nBulk "MINUS") "diode ivpcell
NCSU_Analog_Parts" )
;saveProperty( PNdiode "model" strcat( modelPrefix
NCSU_modelSuffix["pndiode"] ) ) ;disable 01/25

;-----
; old-style moscap extraction (deprecated)
;-----

ivIf( switch("Use_old_moscap_extraction") then
    extractDevice( nChannelCap (poly "G") (nDiff "S") (pBulk "B")
"nmoscap
ivpcell NCSU_Analog_Parts" )
    let( ( capWidth capArea capLength )
        capArea = measureParameter( area nChannelCap 1.0e-12 )
        capWidth = measureParameter( length (nChannelCap coincident
poly)
1.0e-6 )
        capLength = calculateParameter( 2 * capArea / capWidth )
saveProperty( nChannelCap "m" 0.5)
        saveParameter( capWidth "w" )
        saveParameter( capLength "l" )
    )
extractDevice( pChannelCap (poly "G") (pDiff "S") (nBulk "B") "pmoscap
ivpcell NCSU_Analog_Parts" )
    let( ( capWidth capArea capLength )
        capArea = measureParameter( area pChannelCap 1.0e-12 )
        capWidth = measureParameter( length (pChannelCap coincident poly)
1.0e-6 )
        capLength = calculateParameter( 2 * capArea / capWidth )
saveProperty( pChannelCap "m" 0.5)
        saveParameter( capWidth "w" )
        saveParameter( capLength "l" )
    )
if( !cwellAvailable then
saveProperty( nChannelCap "model" strcat( modelPrefix
NCSU_modelSuffix["nmos"] ) )
    saveProperty( pChannelCap "model" strcat( modelPrefix
NCSU_modelSuffix["pmos"] ) )
)
if( elecAvailable then
    extractDevice( nElecChannelCap (elec "G") (nDiff "S") (pBulk
"B")
"nmoscap ivpcell NCSU_Analog_Parts" )
    let( ( capWidth capLength capArea )
        capArea = measureParameter( area nElecChannelCap 1.0e-12 )
        capWidth = measureParameter( length (nElecChannelCap coincident
poly) 1.0e-6 )
        capLength = calculateParameter( 2 * capArea / capWidth )
saveProperty( nElecChannelCap "m" 0.5)
        saveParameter( capWidth "w" )
        saveParameter( capLength "l" )
    )
    saveProperty( nElecChannelCap "model" strcat( modelPrefix
NCSU_modelSuffix["nmos_elec"] ) )

        extractDevice( pElecChannelCap (elec "G") (nDiff "S") (nBulk "B")
"pmoscap ivpcell NCSU_Analog_Parts" )
        let( ( capWidth capLength capArea )
            capArea = measureParameter( area pElecChannelCap 1.0e-12 )
            capWidth = measureParameter( length (pElecChannelCap coincident
poly) 1.0e-6 )
            capLength = calculateParameter( 2 * capArea / capWidth )
saveProperty( pElecChannelCap "m" 0.5 )
            saveParameter( capWidth "w" )
            saveParameter( capLength "l" )
        )
    )

```

```

        saveProperty( pElecChannelCap "model" strcat( modelPrefix
NCSU_modelSuffix["pmos_elec"] ) )
    )
    ;-----
    ; saveRecognition and saveInterconnect statements for everything
    ;-----
saveRecognition( nChannelTran "poly" )
    saveRecognition( pChannelTran "poly" )
if( hvAvailable then
    saveRecognition( hvnChannelTran "poly" )
    saveRecognition( hvpChannelTran "poly" )
)
if( elecAvailable then
    unless( member( techdesc list( "TSMC_CMOS035_4M2P"
                                  "TSMC_CMOS035_3M2P"
                                  "AMI_C5N"
                                  ) )
            saveRecognition( nElecChannelTran "elec" )
            saveRecognition( pElecChannelTran "elec" )
        )
    saveInterconnect( (elec "elec") )
    saveInterconnect( (ce "cc") )
)
if( npnAvailable then
    saveInterconnect( (nnpCollectorContact "cc") )
    saveInterconnect( (nnpCollector "nwell") )
    saveInterconnect( (nnpEmitterContact "cc") )
    saveInterconnect( (nnpBaseContact "cc") )
    saveInterconnect( (nnpBase "pbase") )
)
case( wellType
    ( "P"
        saveInterconnect( (pBulk "pwell")) )
    ( "E"
        saveInterconnect( (nBulk "nwell"))
        saveInterconnect( (pBulk "pwell")) )
    ( t
        saveInterconnect( (nBulk "nwell")) )
)
saveInterconnect( (nOhmic "active") )
saveInterconnect( (pOhmic "active") )
saveInterconnect( (nDiff "active") )
saveInterconnect( (pDiff "active") )
saveInterconnect( (poly "poly") )
saveInterconnect( (metall "metall") )
saveInterconnect( (nOhmicContact "cc") )
saveInterconnect( (pOhmicContact "cc") )
saveInterconnect( (nDiffContact "cc") )
saveInterconnect( (pDiffContact "cc") )
saveInterconnect( (cp "cc") )
saveInterconnect( (metal2 "metal2") )
saveInterconnect( (via "via") )
if( cwellAvailable then
    saveInterconnect( (lcDiff "active") )
    saveInterconnect( (lcContact "cc") )
)
if( metal3Available then
    saveInterconnect( (metal3 "metal3") )
    saveInterconnect( (via2 "via2") )
)
if( metal4Available then
    saveInterconnect( (metal4 "metal4") )
    saveInterconnect( (via3 "via3") )
)
if( metal5Available then
    saveInterconnect( (metal5 "metal5") )

```

```

        saveInterconnect( (via4 "via4") )
    )
    if( metal6Available then
        saveInterconnect( (metal6 "metal6") )
        saveInterconnect( (via5 "via5") )
    )
    saveInterconnect( (nwellRes "res_id") )
    saveInterconnect( (polyRes "res_id") )
saveInterconnect( (mcmV4 "mcmV4") )
    saveInterconnect( (mcm4 "mcm4") )
    saveInterconnect( (mcm2 "mcm2") )
    saveInterconnect( (mcm3 "mcm3") )
    saveInterconnect( (mcmV3 "mcmV3") ) ;; add 01 25 5 lines
if( sblockAvailable then
    saveInterconnect( (polySRes "res_id") )
)
    if( elecAvailable then
        saveInterconnect( (elecRes "res_id") )
    )
    if( highresAvailable then
        saveInterconnect( (elecHighres "res_id") )
    )
)
;-----
; Add some layers to the "excell" view for hierarchical extraction.
; This makes it easier to connect upper level "paint" to the pins
; of a lower-level cell, since more of the lower-level cell shows
; up in the upper-level cell's extracted cellview.
;-----
saveDerived( metall ("metall" "net") cell_view )
saveDerived( metal2 ("metal2" "net") cell_view )
saveDerived( via ("via" "net") cell_view )
if( metal3Available then
    saveDerived( metal3 ("metal3" "net") cell_view )
    saveDerived( via2 ("via2" "net") cell_view )
)
if( metal4Available then
    saveDerived( metal4 ("metal4" "net") cell_view )
    saveDerived( via3 ("via3" "net") cell_view )
)
if( metal5Available then
    saveDerived( metal5 ("metal5" "net") cell_view )
    saveDerived( via4 ("via4" "net") cell_view )
)
if( metal6Available then
    saveDerived( metal6 ("metal6" "net") cell_view )
    saveDerived( via5 ("via5" "net") cell_view )
)
)
;-----
; Some routines to convert back and forth between the mask layers
; and convenience layers and to create select layers from the
; convenience layers...
;-----
ivIf( switch("Layer_create_select_around_field_poly") then
    let( ( fieldpoly )
        fieldpoly = geomSize( "poly" (lambda * 1.0) raw )
        saveDerived( geomAnd( nBulk geomAndNot( fieldpoly geomOr( "pselect"
"nselect" ) ) ) ("pselect" "drawing") )
        saveDerived( geomAnd( pBulk geomAndNot( fieldpoly geomOr( "pselect"
"nselect" ) ) ) ("nselect" "drawing") )
    )
)
ivIf( switch("Layer_convert_[n]pactive_to_active") then
    saveDerived( geomCat("nactive" "pactive") ("active" "drawing") )
    geomErase( "nactive" "drawing")
    geomErase( "pactive" "drawing")
)
ivIf( switch("Layer_create_nselect_around_nactive") then

```

```

        let( ( nactive )
            nactive = geomSize( "nactive" (lambda * 2.0) raw )
            saveDerived( geomCat( geomOutside( nactive "pactive")
            geomAndNot( geomStraddle( nactive "pactive")"pactive" ) )
                ("nselect" "drawing" ) )
        )
    )
ivIf( switch("Layer_create_pselect_around_pactive") then
    let( ( pactive )
        pactive = geomSize( "pactive" (lambda * 2.0) raw )
        saveDerived( geomCat( geomOutside( pactive "nactive")
        geomAndNot( geomStraddle( pactive "nactive")"nactive" ) )
            ("pselect" "drawing" ) )
    )
)

ivIf( switch("Layer_convert_active_to_[np]active") then
    case( wellType
        ( "P"
            saveDerived( geomCat( geomOutside( "active" "pselect")
            geomAndNot( geomStraddle( "active" "pselect") "pselect" ) )
                ("nactive" "drawing" ) )
        )
        ( t
            saveDerived( geomCat( geomInside( "active" "nselect")
            geomAnd( geomStraddle( "active" "nselect") "nselect" ) )
                ("nactive" "drawing" ) )
        )
    )
)
case( wellType
    ( "N"
        saveDerived( geomCat( geomOutside( "active" "nselect")
        geomAndNot( geomStraddle( "active" "nselect") "nselect" ) )
            ("pactive" "drawing" ) )
    )
    ( t
        saveDerived( geomCat( geomInside( "active" "pselect")
        geomAnd( geomStraddle( "active"
"pselect" ) "pselect" ) )
            ("pactive" "drawing" ) )
    )
)
)

;-----
; have (text) labels show up in extracted view
;-----
ivIf( switch("Keep_labels_in_extracted_view") then
    copyGraphics( ("text" "drawing") all )
)
) ; drcExtractRules
) ; let
; vim:ts=4:columns=132:set tw=0:

```

Appendix E - Layer Definition file

```
/*
*****
Diva Derived Layer Definitions
*****
*/

* Note that some of these variables are set here and some are set
* above. The ones below are set here because any variable with the
* value nil becomes unbound when you enter the drcExtractRules()
* function. Some have to be set above because functions like
* error() can't be called inside drcExtractRules(). Weird.
*/

wellType = substring( NCSU_techData[ techdesc ]->mosisCode 3 1 )
submicronAvailable = NCSU_techData[ techdesc ]->submicronRules
deepAvailable = NCSU_techData[ techdesc ]->deepRules
stackedViasAvailable = NCSU_techData[ techdesc ]->stackedVias
lambda = atof( NCSU_techData[ techdesc ]->lambda )
gridRes = NCSU_techData[ techdesc ]->gridRes
modelPrefix = NCSU_techData[ techdesc ]->fetModelPrefix

techfile = techGetTechFile( geGetEditCellView() )
cwellAvailable = techGetLayerNum( techfile "cwell" )
polycapAvailable = techGetLayerNum( techfile "polycap" )
nnpAvailable = techGetLayerNum( techfile "pbase" )
ccdAvailable = techGetLayerNum( techfile "ccd" )
metal3Available = techGetLayerNum( techfile "metal3" )
metal4Available = techGetLayerNum( techfile "metal4" )
metal5Available = techGetLayerNum( techfile "metal5" )
metal6Available = techGetLayerNum( techfile "metal6" )
metalcapAvailable = techGetLayerNum( techfile "metalcap" )
hvAvailable = techGetLayerNum( techfile "tactive" )
elecAvailable = techGetLayerNum( techfile "elec" )
memsAvailable = techGetLayerNum( techfile "pstop" )
sblockAvailable = techGetLayerNum( techfile "sblock" )
highresAvailable = techGetLayerNum( techfile "highres" )

/*
*****
DERIVED LAYER DEFINITIONS
*****
*/

ivIf( switch("drc?") then
    nodrc = geomOr( "nodrc" )
)

ivIf( switch("extract?") then
    nodrc = geomAndNot( "nodrc" "nodrc" )
)

bkgnd = geomBkgnd()
gwell = geomOr( geomAndNot( ( "gwell" "drawing" ) nodrc ) )
nwell = geomOr( geomAndNot( ( "nwell" "drawing" ) nodrc ) )
pwell = geomOr( geomAndNot( ( "pwell" "drawing" ) nodrc ) )
nactive = geomOr( geomAndNot( ( "nactive" "drawing" ) nodrc ) )
pactive = geomOr( geomAndNot( ( "pactive" "drawing" ) nodrc ) )
active = geomOr( geomAndNot( ( "active" "drawing" ) nodrc ) nactive
pactive )
gselect = geomOr( geomAndNot( ( "gselect" "drawing" ) nodrc ) )
nselect = geomOr( geomAndNot( ( "nselect" "drawing" ) nodrc ) )
pselect = geomOr( geomAndNot( ( "pselect" "drawing" ) nodrc ) )
poly = geomOr( geomAndNot( ( "poly" "drawing" ) nodrc ) )
metall = geomOr( geomAndNot( ( "metall" "drawing" ) nodrc ) )
cc = geomOr( geomAndNot( ( "cc" "drawing" ) nodrc ) )
metal2 = geomOr( geomAndNot( ( "metal2" "drawing" ) nodrc ) )
via = geomOr( geomAndNot( ( "via" "drawing" ) nodrc ) )
glass = geomOr( geomAndNot( ( "glass" "drawing" ) nodrc ) )
```

```

pad      = geomOr( geomAndNot( ( "pad" "drawing" ) nodrc ) )
nolpe   = geomOr( "nolpe" )
cap_id  = geomOr( "cap_id" )
res_id  = geomOr( "res_id" )

mcm0 = geomOr( "mcm0" )
mcmV1 = geomOr( "mcmV1" )
mcm1 = geomOr( "mcm1" )
mcmV2 = geomOr( "mcmV2" )
mcm2 = geomOr( "mcm2" )
mcmV3 = geomOr( "mcmV3" )
mcm3 = geomOr( "mcm3" )
mcmV4 = geomOr( "mcmV4" )
mcm4 = geomOr( "mcm4" )

shoccXYterminal=geomOr(mcmV3)
mcm4ViamcmV4=geomAnd(mcm4 mcmV4)
  if( metal3Available then
    metal3 = geomOr( geomAndNot( ( "metal3" "drawing" ) nodrc ) )
    via2   = geomOr( geomAndNot( ( "via2" "drawing" ) nodrc ) )
  )
  if( metal4Available then
    metal4 = geomOr( geomAndNot( ( "metal4" "drawing" ) nodrc ) )
    via3   = geomOr( geomAndNot( ( "via3" "drawing" ) nodrc ) )
  )
  if( metal5Available then
    metal5 = geomOr( geomAndNot( ( "metal5" "drawing" ) nodrc ) )
    via4   = geomOr( geomAndNot( ( "via4" "drawing" ) nodrc ) )
  )
  if( metal6Available then
    metal6 = geomOr( geomAndNot( ( "metal6" "drawing" ) nodrc ) )
    via5   = geomOr( geomAndNot( ( "via5" "drawing" ) nodrc ) )
  )
  if( metalcapAvailable then
    metalcap = geomOr( geomAndNot( ( "metalcap" "drawing" ) nodrc ) )
  )
  if( ccdAvailable then
    ccd = geomOr( geomAndNot( ( "ccd" "drawing" ) nodrc ) )
  )
  if( cwellAvailable then
    cwell = geomOr( geomAndNot( ( "cwell" "drawing" ) nodrc ) )
  )
  if( polycapAvailable then
    polycap = geomOr( geomAndNot( ( "polycap" "drawing" ) nodrc ) )
    cpolycap = geomAnd( polycap geomOr( cc cp ) )
  )
  if( sblockAvailable then
    sblock = geomOr( geomAndNot( ( "sblock" "drawing" ) nodrc ) )
  )
  if( highresAvailable then
    highres = geomOr( geomAndNot( ( "highres" "drawing" ) nodrc ) )
  )
  if( memsAvailable then
    pstop = geomOr( geomAndNot( ( "pstop" "drawing" ) nodrc ) )
    open = geomOr( geomAndNot( ( "open" "drawing" ) nodrc ) )
  )
(elec
-> poly -> active).
  */
  if( elecAvailable then
    elec = geomOr( geomAndNot( ( "elec" "drawing" ) nodrc ) )
    ce = geomOr( geomOr( geomAndNot( ( "ce" "drawing" ) nodrc ) )
geomAnd( cc elec ) )
    cp = geomOr( geomOr( geomAndNot( ( "cp" "drawing" ) nodrc ) )
geomAnd( cc geomAndNot( poly ce ) ) )
    ca = geomOr( geomOr( geomAndNot( ( "ca" "drawing" ) nodrc ) )

```

```

geomAnd( cc geomAndNot( active geomOr( ce cp ) ) ) )
else
  cp = geomOr( geomOr( geomAndNot( ( "cp" "drawing" ) nodrc ) )
geomAnd( cc poly ) )
  ca = geomOr( geomOr( geomAndNot( ( "ca" "drawing" ) nodrc ) )
geomAnd( cc geomAndNot( active cp ) ) )
)
if( npnAvailable then
  pbase = geomOr( geomAndNot( ( "pbase" "drawing" ) nodrc ) )
  cactive = geomOr( geomAndNot( ( "cactive" "drawing" ) nodrc ) )
  ca = geomOr( ca geomAnd( cc cactive ) )
)

nActive = geomAnd( active nselect )
pActive = geomAnd( active pselect )
if( ccdAvailable then
  nActive = geomAndNot( nActive ccd )
  pActive = geomAndNot( pActive ccd )
)
if( cwellAvailable then
  nActive = geomAndNot( nActive cwell )
  pActive = geomAndNot( pActive cwell )
)
if( hvAvailable then
  tactive = geomOr( geomAndNot( ( "tactive" "drawing" ) nodrc ) )
)

/*
* This next section basically sets how the well enclosure of
* source/drain active, substrate/well contact and capacitor
* electrode are interpreted. In a pwell process, everything that
* doesn't have the "pwell" layer drawn on it is assumed to be
* N-type bulk. Similarly, in an nwell process, everything that
* doesn't have the "nwell" layer drawn on it is assumed to be
* P-type bulk. If it's an "either-well" process, only the areas
* with drawn "pwell" and "nwell" are assumed to be P-type bulk or
* N-type bulk, respectively.
*
* Since MOSIS only has N-type processes for now, this is assumed to
* be the default.
*/
case( wellType
( "P"
  nBulk = geomOr( geomNot( pwell ) geomAndNot( nwell pwell ) )
  pBulk = geomOr( pwell )
  nOhmic = geomAndNot( nActive pwell )
  pOhmic = geomAnd( pActive pwell )
  nNotOhmic = geomAnd( nActive pwell )
  pNotOhmic = geomAndNot( pActive pwell )
)
( "E"
  nBulk = geomOr( nwell )
  pBulk = geomOr( pwell )
  nOhmic = geomAnd( nActive nwell )
  pOhmic = geomAnd( pActive pwell )
  nNotOhmic = geomAnd( nActive pwell )
  pNotOhmic = geomAnd( pActive nwell )
)
( t
  if( npnAvailable then
    nBulk = geomOutside( nwell cactive )
  else
    nBulk = geomOr( nwell )
  )
  pBulk = geomOr( geomNot( nwell ) geomAndNot( pwell nwell ) )
  nOhmic = geomAnd( nActive nwell )
  pOhmic = geomAndNot( pActive nwell )
)

```

```

        nNotOhmic = geomAndNot( nActive nwell )
        pNotOhmic = geomAnd( pActive nwell)
    )
)

if( elecAvailable then
    nDiff = geomAndNot( nNotOhmic geomOr( poly elec ) )
    pDiff = geomAndNot( pNotOhmic geomOr( poly elec ) )
    nChannel = geomOutside( geomAnd( nNotOhmic poly ) elec )
    pChannel = geomOutside( geomAnd( pNotOhmic poly ) elec )
    nElecChannel = geomOutside( geomAnd( nNotOhmic elec ) poly )
    pElecChannel = geomOutside( geomAnd( pNotOhmic elec ) poly )
    nElecChannelTran = geomButting( nElecChannel nDiff keep == 2 )
    pElecChannelTran = geomButting( pElecChannel pDiff keep == 2 )
    nElecChannelCap = geomButting( nElecChannel nDiff keep == 1 )
    pElecChannelCap = geomButting( pElecChannel pDiff keep == 1 )
    Space = geomNot( geomOr( active poly elec ) )
else
    nDiff = geomAndNot( nNotOhmic poly )
    pDiff = geomAndNot( pNotOhmic poly )
    nChannel = geomAnd( nNotOhmic poly )
    pChannel = geomAnd( pNotOhmic poly )
    Space = geomNot( geomOr( active poly ) )
)
nChannelTran = geomButting( nChannel nDiff keep == 2 )
pChannelTran = geomButting( pChannel pDiff keep == 2 )
nChannelCap = geomButting( nChannel nDiff keep == 1 )
pChannelCap = geomButting( pChannel pDiff keep == 1 )
if( hvAvailable then
    hvnChannelTran = geomAnd( nChannelTran tactive )
    nChannelTran = geomAndNot( nChannelTran hvnChannelTran )
    hvpChannelTran = geomAnd( pChannelTran tactive )
    pChannelTran = geomAndNot( pChannelTran hvpChannelTran )
)

nDiffContact = geomAnd( ca nDiff )
pDiffContact = geomAnd( ca pDiff )
nOhmicContact = geomAnd( ca nOhmic )
pOhmicContact = geomAnd( ca pOhmic )
Gate = geomAnd( geomOr( nNotOhmic pNotOhmic) poly )
fieldPoly = geomAvoiding( poly Gate )
mlpCap = geomAnd( geomAnd( poly metall ) cap_id )
mlsCap = geomAnd( geomAndNot( metall poly ) cap_id )
m2mlCap = geomAnd( geomAnd( metall metal2 ) cap_id )
if( metal3Available then
    m3m2Cap = geomAnd( geomAnd( metal2 metal3 ) cap_id )
)
if( metal4Available then
    m4m3Cap = geomAnd( geomAnd( metal3 metal4 ) cap_id )
)
if( metal5Available then
    m5m4Cap = geomAnd( geomAnd( metal4 metal5 ) cap_id )
)
if( metal6Available then
    m6m5Cap = geomAnd( geomAnd( metal5 metal6 ) cap_id )
)
if( metalcapAvailable then
    /*
    * metalcap is between top-layer metal and next-to-top layer metal. for *
    drc, we need the whole metal shape of the bottom plate, not just the *
    metal-metalcap overlap.
    */
    cond(
        ( metal6Available
            metalcapBottom = geomStraddle( metal5 metalcap )
            metalcapCap = geomAnd( metalcap metal5 )
            via5metalcap = geomAnd( via5 metalcapBottom )
        )
    )
)

```

```

        via5 = geomAndNot( via5 via5metalcap )
    )
    ( metal5Available
        metalcapBottom = geomStraddle( metal4 metalcap )
        metalcapCap = geomAnd( metalcap metal4 )
        via4metalcap = geomAnd( via4 metalcapBottom )
        via4 = geomAndNot( via4 via4metalcap )
    )
)
)
)
if( elecAvailable then
    elecGate = geomAnd( geomOr( nNotOhmic pNotOhmic) elec )
    fieldElec = geomAvoiding( elec elecGate )
    CapacitorElec = geomInside( elec poly )
    TransistorElec = geomOverlap( elec geomNot( poly ) )
)
if( polycapAvailable then
    polycapCap = geomAnd( poly polycap )
)
if( sblockAvailable then
    sGateWidthCheck = geomSize( geomSize( geomAnd( Gate sblock ) -2.9 )
2.9
)
)
if( npnAvailable then
    npnCollector = geomAnd( nwell geomAnd( nselect cactive ) )
    npnCollectorContact = geomAnd( ca npnCollector )
    npnBaseImplant = geomAnd( nwell pbase )
    npnEmitter = geomAnd( nselect npnBaseImplant )
    npnBase = geomAndNot( npnBaseImplant npnEmitter )
    npnBaseTap = geomAnd( npnBase pselect )
    npnBaseContact = geomAnd( geomOr( cc ca ) npnBaseTap )
    npnEmitterContact = geomAnd( geomOr( cc ca ) npnEmitter )
    npnTran = geomEnclose( nwell geomOr( npnCollector
npnBase npnEmitter ) )
)
if( ccdAvailable then
    ccdDiff = geomAnd( active ccd )
    ccdContact = geomAnd( ca ccdDiff )
)
if( cwellAvailable then
    lcDiff = geomStraddle( cwell active )
    lcContact = geomAnd( ca lcDiff )
    lcCap = geomAnd( poly geomAnd( lcDiff active ) )
)
)
if( sblockAvailable then
    fieldPoly = geomAndNot( fieldPoly geomOr( sblock res_id ) )
    polySRes = geomButting( geomAnd( sblock poly ) fieldPoly keep == 2 )
    polyRes = geomButting( geomAndNot( geomAnd( res_id poly ) polySRes )
fieldPoly keep == 2 )
    poly = geomAndNot( poly geomOr( sblock res_id ) )
else
    fieldPoly = geomAndNot( fieldPoly res_id )
    polyRes = geomButting( geomAnd( res_id poly ) fieldPoly keep == 2 )
    poly = geomAndNot( poly res_id )
)
)
; res_id/elec
if( elecAvailable then
    ; currently can only do highres over elec (poly2)
    if( highresAvailable then
        fieldElec = geomAndNot( fieldElec geomOr( res_id highres ) )
        elecRes = geomButting( geomAnd( res_id elec ) fieldElec keep == 2 )
    elecHighres = geomButting( geomAnd( highres elec ) fieldElec keep == 2 )
        elec = geomAndNot( elec geomOr( res_id highres ) )

```

```

else
    fieldElec = geomAndNot( fieldElec res_id )
    elecRes = geomButting( geomAnd( res_id elec ) fieldElec keep == 2 )
    elec = geomAndNot( elec res_id )
)
)

; res_id/nwell
nBulk = geomAndNot( nBulk res_id )
nwellRes = geomButting( geomAnd( res_id nwell ) nBulk keep == 2 )
nwell = geomAndNot( nwell res_id ) ; w/o this, whole well is connected

cond(
    ( metal6Available
        if( metalcapAvailable then
            geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
                via( pOhmicContact pOhmic pwell pBulk metall )
                    via( nDiffContact nDiff metall )
                    via( pDiffContact pDiff metall )
                    via( cp poly metall )
                    via( via metall metal2 )
                    via( via2 metal2 metal3 )
                    via( via3 metal3 metal4 )
                    via( via4 metal4 metal5 )
                    via( via5 metal5 metal6 )
                    via( via5metalcap metalcap metal6 )
                label( "text" metal6 metalcap metal5 metal4 metal3
                    metal2 metall poly pDiff nDiff )
            )
        else
            geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
                via( pOhmicContact pOhmic pwell pBulk metall )
                    via( nDiffContact nDiff metall )
                    via( pDiffContact pDiff metall )
                    via( cp poly metall )
                    via( via metall metal2 )
                    via( via2 metal2 metal3 )
                    via( via3 metal3 metal4 )
                    via( via4 metal4 metal5 )
                    via( via5 metal5 metal6 )
                label( "text" metal6 metal5 metal4 metal3 metal2
                    metall poly pDiff nDiff )
            )
        )
    )
    ( metal5Available
        if( metalcapAvailable then
            geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
                via( pOhmicContact pOhmic pwell pBulk metall )
                    via( nDiffContact nDiff metall )
                    via( pDiffContact pDiff metall )
                    via( cp poly metall )
                    via( via metall metal2 )
                    via( via2 metal2 metal3 )
                    via( via3 metal3 metal4 )
                    via( via4 metal4 metal5 )
                    via( via4metalcap metalcap metal5 )
                label( "text" metal5 metalcap metal4 metal3 metal2
                    metall poly pDiff nDiff )
            )
        else
            geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
                via( pOhmicContact pOhmic pwell pBulk metall )
                    via( nDiffContact nDiff metall )
                    via( pDiffContact pDiff metall )
                    via( cp poly metall )
                    via( via metall metal2 )
            )
        )
    )
)

```

```

        via( via2 metal2 metal3 )
        via( via3 metal3 metal4 )
        via( via4 metal4 metal5 )
        label( "text" metal5 metal4 metal3 metal2 metall
poly pDiff nDiff )
    )
)
( metal4Available
  if( elecAvailable then
    geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
      via( pOhmicContact pOhmic pwell pBulk metall )
      via( nDiffContact nDiff metall )
      via( pDiffContact pDiff metall )
      via( cp poly metall )
      via( ce elec metall )
      via( via metall metal2 )
      via( via2 metal2 metal3 )
      via( via3 metal3 metal4 )
      label( "text" metal4 metal3 metal2 metall elec
poly pDiff nDiff )
    )
  else
    geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
      via( pOhmicContact pOhmic pwell pBulk metall )
      via( nDiffContact nDiff metall )
      via( pDiffContact pDiff metall )
      via( cp poly metall )
      via( via metall metal2 )
      via( via2 metal2 metal3 )
      via( via3 metal3 metal4 )
      via(mcm4ViamcmV4 mcm4 mcmV4)
      via(shoccXYterminal mcm3 mcmV3) ;add two line 01/25
      label( "text" metal4 metal3 metal2 metall poly
pDiff nDiff )
    )
  )
)
( metal3Available && cwellAvailable
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( lcContact lcDiff metall )
    via( cp poly metall )
    via( via metall metal2 )
    via( via2 metal2 metal3 )
    label( "text" metal3 metal2 metall poly lcDiff pDiff nDiff )
  )
)
( metal3Available && elecAvailable
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( cp poly metall )
    via( ce elec metall )
    via( via metall metal2 )
    via( via2 metal2 metal3 )
    label( "text" metal3 metal2 metall elec poly pDiffnDiff )
  )
)
( metal3Available
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )

```

```

        via( nDiffContact nDiff metall )
        via( pDiffContact pDiff metall )
        via( cp poly metall )
        via( via metall metal2 )
        via( via2 metal2 metal3 )
        label( "text" metal3 metal2 metall poly pDiff nDiff )
    )
)
( npnAvailable && ccdAvailable && elecAvailable
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( ccdContact ccdDiff metall )
    via( npnEmitterContact npnEmitter metall )
    via( npnBaseContact npnBaseTap npnBase metall )
    via( npnCollectorContact npnCollector metall )
    via( cp poly metall )
    via( ce elec metall )
    via( via metall metal2 )
    label( "text" metal2 metall elec poly pDiff nDiff
ccdDiff npnCollector npnEmitter npnBase )
  )
)
( npnAvailable && elecAvailable
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( npnEmitterContact npnEmitter metall )
    via( npnBaseContact npnBaseTap npnBase metall )
    via( npnCollectorContact npnCollector metall )
    via( cp poly metall )
    via( ce elec metall )
    via( via metall metal2 )
    label( "text" metal2 metall elec poly pDiff nDiff
npnCollector npnEmitter npnBase )
  )
)
( polycapAvailable
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( cp poly metall )
    via( cpolycap polycap metall )
    via( via metall metal2 )
    label( "text" metal2 metall poly polycap pDiff nDiff )
  )
)
( t
  geomConnect( via( nOhmicContact nOhmic nwell nBulk metall )
    via( pOhmicContact pOhmic pwell pBulk metall )
    via( nDiffContact nDiff metall )
    via( pDiffContact pDiff metall )
    via( cp poly metall )
    via( via metall metal2 )
    label( "text" metal2 metall poly pDiff nDiff )
  )
)
)
ivIf( switch("drc?") then
ivIf( ( !switch("hier?") || switch("currentCell?") ) then
dubiousData( ( "gwell" "drawing" ) "Improperly formed shape - gwell")
dubiousData( ( "nwell" "drawing" ) "Improperly formed shape - nwell")
dubiousData( ( "pwell" "drawing" ) "Improperly formed shape - pwell")
dubiousData( ( "active" "drawing" ) "Improperly formed shape - active,

```

```

nactive or pactive" )
  dubiousData( ( "gselect" "drawing" ) "Improperly formed shape - gselect" )
  dubiousData( ( "nselect" "drawing" ) "Improperly formed shape - nselect" )
  dubiousData( ( "pselect" "drawing" ) "Improperly formed shape -pselect" )
    dubiousData( ( "poly" "drawing" ) "Improperly formed shape - poly" )
  dubiousData( ( "metall" "drawing" ) "Improperly formed shape - metall" )
    dubiousData( ( "ca" "drawing" ) "Improperly formed shape - ca" )
    dubiousData( ( "cp" "drawing" ) "Improperly formed shape - cp" )
  dubiousData( ( "metal2" "drawing" ) "Improperly formed shape - metal2" )
    dubiousData( ( "via" "drawing" ) "Improperly formed shape - via" )
    dubiousData( ( "glass" "drawing" ) "Improperly formed shape - glass" )
saveDerived( geomGetNon45( gwell ) "Non-Manhattan shape - gwell" )
  saveDerived( geomGetNon45( nwell ) "Non-Manhattan shape - nwell" )
  saveDerived( geomGetNon45( pwell ) "Non-Manhattan shape - pwell" )
  saveDerived( geomGetNon45( active ) "Non-Manhattan shape - active,
nactive or pactive" )
  saveDerived( geomGetNon45( gselect ) "Non-Manhattan shape - gselect" )
  saveDerived( geomGetNon45( nselect ) "Non-Manhattan shape - nselect" )
  saveDerived( geomGetNon45( pselect ) "Non-Manhattan shape - pselect" )
  saveDerived( geomGetNon45( poly ) "Non-Manhattan shape - poly" )
  saveDerived( geomGetNon45( metall ) "Non-Manhattan shape - metall" )
  saveDerived( geomGetNon45( ca ) "Non-Manhattan shape - ca" )
  saveDerived( geomGetNon45( cp ) "Non-Manhattan shape - cp" )
  saveDerived( geomGetNon45( metal2 ) "Non-Manhattan shape - metal2" )
  saveDerived( geomGetNon45( via ) "Non-Manhattan shape - via" )
offGrid( gwell gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( nwell gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( pwell gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( active gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( gselect gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( nselect gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( pselect gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( poly gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( metall gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( ca gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( cp gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( metal2 gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( via gridRes "(SCMOS Inst) Edge not on grid" )
if( npnAvailable then
  dubiousData( ( "pbase" "drawing" ) "Improperly formed shape -pbase" )
  saveDerived( geomGetNon45( pbase ) "Non-Manhattan shape - pbase" )
  offGrid( pbase gridRes "(SCMOS Inst) Edge not on grid" )
  dubiousData( ( "cactive" "drawing" ) "Improperly formed shape -cactive" )
  saveDerived( geomGetNon45( cactive ) "Non-Manhattan shape -cactive" )
  offGrid( cactive gridRes "(SCMOS Inst) Edge not on grid" )
  )
  if( ccdAvailable then
    dubiousData( ( "ccd" "drawing" ) "Improperly formed shape -ccd" )
    saveDerived( geomGetNon45( ccd ) "Non-Manhattan shape - ccd"
  )
    offGrid( ccd gridRes "(SCMOS Inst) Edge not on grid" )
  )
  if( metal3Available then
    dubiousData( ( "metal3" "drawing" ) "Improperly formed shape -metal3" )
    dubiousData( ( "via2" "drawing" ) "Improperly formed shape -via2" )
    saveDerived( geomGetNon45( metal3 ) "Non-Manhattan shape - metal3" )
    saveDerived( geomGetNon45( via2 ) "Non-Manhattan shape - via2" )
    offGrid( metal3 gridRes "(SCMOS Inst) Edge not on grid" )
    offGrid( via2 gridRes "(SCMOS Inst) Edge not on grid" )
  )
  if( metal4Available then
    dubiousData( ( "metal4" "drawing" ) "Improperly formed shape -metal4" )
    dubiousData( ( "via3" "drawing" ) "Improperly formed shape -via3" )
    saveDerived( geomGetNon45( metal4 ) "Non-Manhattan shape - metal4" )
    saveDerived( geomGetNon45( via3 ) "Non-Manhattan shape - via3" )
    offGrid( metal4 gridRes "(SCMOS Inst) Edge not on grid" )
    offGrid( via3 gridRes "(SCMOS Inst) Edge not on grid" )
  )

```

```

)
  if( metal5Available then
dubiousData( ( "metal5" "drawing" ) "Improperly formed shape -metal5" )
dubiousData( ( "via4" "drawing" ) "Improperly formed shape -via4" )
saveDerived( geomGetNon45( metal5 ) "Non-Manhattan shape - metal5" )
saveDerived( geomGetNon45( via4 ) "Non-Manhattan shape - via4" )
  offGrid( metal5 gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( via4 gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( metal6Available then
dubiousData( ( "metal6" "drawing" ) "Improperly formed shape -metal6" )
dubiousData( ( "via5" "drawing" ) "Improperly formed shape -via5" )
saveDerived( geomGetNon45( metal6 ) "Non-Manhattan shape - metal6" )
saveDerived( geomGetNon45( via5 ) "Non-Manhattan shape - via5" )
  offGrid( metal6 gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( via5 gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( metalcapAvailable then
dubiousData( ( "metalcap" "drawing" ) "Improperly formed shape -metalcap" )
saveDerived( geomGetNon45( metalcap ) "Non-Manhattan shape -metalcap" )
offGrid( metalcap gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( elecAvailable then
dubiousData( ( "elec" "drawing" ) "Improperly formed shape -elec" )
dubiousData( ( "ce" "drawing" ) "Improperly formed shape - ce" )
saveDerived( geomGetNon45( elec ) "Non-Manhattan shape - elec" )
  saveDerived( geomGetNon45( ce ) "Non-Manhattan shape - ce" )
  offGrid( elec gridRes "(SCMOS Inst) Edge not on grid" )
  offGrid( ce gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( cwellAvailable then
dubiousData( ( "cwell" "drawing" ) "Improperly formed shape -cwell" )
saveDerived( geomGetNon45( cwell ) "Non-Manhattan shape - cwell" )
  offGrid( cwell gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( polycapAvailable then
dubiousData( ( "polycap" "drawing" ) "Improperly formed shape -polycap" )
saveDerived( geomGetNon45( polycap ) "Non-Manhattan shape -polycap" )
  offGrid( polycap gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( sblockAvailable then
dubiousData( ( "sblock" "drawing" ) "Improperly formed shape -sblock" )
saveDerived( geomGetNon45( sblock ) "Non-Manhattan shape -sblock" )
  offGrid( sblock gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( memsAvailable then
dubiousData( ( "pstop" "drawing" ) "Improperly formed shape -pstop" )
saveDerived( geomGetNon45( pstop ) "Non-Manhattan shape - pstop" )
  offGrid( pstop gridRes "(SCMOS Inst) Edge not on grid" )
dubiousData( ( "open" "drawing" ) "Improperly formed shape -open" )
saveDerived( geomGetNon45( open ) "Non-Manhattan shape - open" )
)
  offGrid( open gridRes "(SCMOS Inst) Edge not on grid" )
)
  if( hvAvailable then
dubiousData( ( "tactive" "drawing" ) "Improperly formed shape -tactive" )
saveDerived( geomGetNon45( tactive ) "Non-Manhattan shape -tactive" )
  offGrid( tactive gridRes "(SCMOS Inst) Edge not on grid" )
)
)
gwellEdge = geomGetEdge( gwell )
nwellEdge = geomGetEdge( nwell )
pwellEdge = geomGetEdge( pwell )
activeEdge = geomGetEdge( active )
gselectEdge = geomGetEdge( gselect )
nselectEdge = geomGetEdge( nselect )
pselectEdge = geomGetEdge( pselect )

```

```

polyEdge          = geomGetEdge( poly      )
metal1Edge        = geomGetEdge( metall   )
ccEdge            = geomGetEdge( cc       )
caEdge            = geomGetEdge( ca       )
cpEdge            = geomGetEdge( cp       )
metal2Edge        = geomGetEdge( metal2   )
viaEdge           = geomGetEdge( via      )
glassEdge         = geomGetEdge( glass    )
padEdge           = geomGetEdge( pad      )
if( cwellAvailable then
    cwellEdge      = geomGetEdge( cwell    )
)
if( npnAvailable then
    pbaseEdge      = geomGetEdge( pbase    )
    cactiveEdge    = geomGetEdge( cactive  )
)
if( ccdAvailable then
    ccdEdge        = geomGetEdge( ccd      )
)
if( polycapAvailable then
    polycapEdge    = geomGetEdge( polycap  )
    cpolycapEdge   = geomGetEdge( cpolycap )
)
if( sblockAvailable then
    sblockEdge     = geomGetEdge( sblock   )
)
if( highresAvailable then
    highresEdge    = geomGetEdge( highres  )
)
if( memsAvailable then
    pstopEdge      = geomGetEdge( pstop    )
    openEdge       = geomGetEdge( open     )
)
if( elecAvailable then
    ceEdge         = geomGetEdge( ce       )
    elecEdge       = geomGetEdge( elec     )
)
if( metal3Available then
    metal3Edge     = geomGetEdge( metal3   )
    via2Edge       = geomGetEdge( via2     )
)
if( metal4Available then
    metal4Edge     = geomGetEdge( metal4   )
    via3Edge       = geomGetEdge( via3     )
)
if( metal5Available then
    metal5Edge     = geomGetEdge( metal5   )
    via4Edge       = geomGetEdge( via4     )
)
if( metal6Available then
    metal6Edge     = geomGetEdge( metal6   )
    via5Edge       = geomGetEdge( via5     )
)
if( metalcapAvailable then
    metalcapEdge   = geomGetEdge( metalcap )
    metalcapBottomEdge = geomGetEdge( metalcapBottom )
    metalcapCapEdge = geomGetEdge( metalcapCap )
    cond(
        ( metal6Available
            via5metalcapEdge = geomGetEdge( via5metalcap )
        )
        ( metal5Available
            via4metalcapEdge = geomGetEdge( via4metalcap )
        )
    )
)
if( hvAvailable then

```

```

        tactiveEdge      = geomGetEdge( tactive )
    )
nBulkEdge      = geomGetEdge( nBulk )
pBulkEdge      = geomGetEdge( pBulk )
nOhmicEdge     = geomGetEdge( nOhmic )
pOhmicEdge     = geomGetEdge( pOhmic )
nNotOhmicEdge  = geomGetEdge( nNotOhmic )
pNotOhmicEdge  = geomGetEdge( pNotOhmic )
GateEdge       = geomGetEdge( Gate )
fieldPolyEdge  = geomGetEdge( fieldPoly )
if( elecAvailable then
    CapacitorElecEdge = geomGetEdge( CapacitorElec )
    TransistorElecEdge = geomGetEdge( TransistorElec )
)
if( npnAvailable then
    npnCollectorEdge = geomGetEdge( npnCollector )
    npnCollectorContactEdge = geomGetEdge( npnCollectorContact )
    npnBaseImplantEdge = geomGetEdge( npnBaseImplant )
    npnEmitterEdge = geomGetEdge( npnEmitter )
    npnBaseEdge = geomGetEdge( npnBase )
    npnBaseTapEdge = geomGetEdge( npnBaseTap )
    npnBaseContactEdge = geomGetEdge( npnBaseContact )
    npnEmitterContactEdge = geomGetEdge( npnEmitterContact )
)
if( ccdAvailable then
    ccdDiffEdge = geomGetEdge( ccdDiff )
    ccdContactEdge = geomGetEdge( ccdContact )
)
if( cwellAvailable then
    lcDiffEdge = geomGetEdge( lcDiff )
    lcCapEdge = geomGetEdge( lcCap )
)
if( polycapAvailable then
    polycapCapEdge = geomGetEdge( polycapCap )
)
if( sblockAvailable then
    sGateWidthCheckEdge = geomGetEdge( sGateWidthCheck )
)
if( sblockAvailable then
    polySResEdge = geomGetEdge( polySRes )
)
polyResEdge = geomGetEdge( polyRes coincident poly )
nwellResEdge = geomGetEdge( nwellRes coincident nwell )
if( elecAvailable && highresAvailable then
    elecHighresEdge = geomGetEdge( elecRes coincident elec )
)
) ; close ivIf( switch( "?drc" ) )

```

Appendix F – The HSPICE Netlist

```
* # FILE NAME: /AFS/UNITY.NCSU.EDU/USERS/A/AKVARMA/CADENCE/SIMULATION/
* SHOCC_TEST/hspiceS/extracted/netlist/SHOCC_TEST.C.raw
* Netlist output for hspiceS.
* Generated on Feb 21 19:35:05 2001
* File name: TEST_SHOCC_3_SHOCC_TEST_extracted.S.
* Subcircuit for cell: SHOCC_TEST.
* Generated for: hspiceS.
* Generated on Feb 21 19:35:05 2001.
C15 0 3 2.72160008051513E-15 M=1.0
C17 0 2 3.42629994065343E-15 M=1.0
C19 3 4 5.51700002414217E-15 M=1.0
C21 2 3 3.38814004758844E-15 M=1.0
C23 0 3 2.25783006013562E-15 M=1.0
C25 0 1 2.12688003703952E-15 M=1.0
C27 VDD! 0 164.090719094109E-15 M=1.0
M29 VDD! IN 1 VDD! HP14TBP L=600E-9 W=3.6E-6 AD=21.0150005036658E-12
+AS=5.39999989168649E-12 PD=23.5499992413679E-6 PS=6.59999977870029E-6 M=1
M31 2 1 VDD! VDD! HP14TBP L=600E-9 W=14.1E-6 AD=21.1500001540132E-12
+AS=21.0150005036658E-12 PD=17.0999992405996E-6 PS=23.5499992413679E-6 M=1
M33 VDD! 2 3 VDD! HP14TBP L=600E-9 W=32.1E-6 AD=93.9150007761569E-12
+AS=28.8899997674674E-12 PD=10.1249997896957E-6 PS=1.7999999499807E-6 M=1
M35 3 2 VDD! VDD! HP14TBP L=600E-9 W=32.1E-6 AD=28.8899997674674E-12
+AS=48.1500013471692E-12 PD=1.7999999499807E-6 PS=35.1000016962644E-6 M=1
M37 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=101.159997045741E-12
+AS=75.8699977843058E-12 PD=46.9500009785406E-6 PS=3.599998999614E-6 M=1
M39 VDD! 3 4 VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M41 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M43 VDD! 3 4 VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=75.8699977843058E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M45 4 3 VDD! VDD! HP14TBP L=900E-9 W=42.15E-6 AD=75.8699977843058E-12
+AS=93.9150007761569E-12 PD=3.599998999614E-6 PS=10.1249997896957E-6 M=1
M47 OUT 9 VDD! VDD! HP14TBP L=2.25E-6 W=2.55E-6 AD=11.0925001822748E-12
+AS=11.8575002247545E-12 PD=11.2500001705484E-6 PS=11.850000191771E-6 M=1
M49 0 IN 1 0 HP14TBN L=600E-9 W=1.8E-6 AD=11.0699999514297E-12
+AS=2.69999994584325E-12 PD=13.5000000227592E-6 PS=4.80000016978011E-6 M=1
M51 2 1 0 0 HP14TBN L=600E-9 W=7.2E-6 AD=10.799999783373E-12
+AS=11.0699999514297E-12 PD=10.2000003607827E-6 PS=13.5000000227592E-6 M=1
M53 0 2 3 0 HP14TBN L=600E-9 W=16.2E-6 AD=47.2499990522568E-12
+AS=14.5800004014429E-12 PD=7.57499992687372E-6 PS=1.7999999499807E-6 M=1
M55 3 2 0 0 HP14TBN L=600E-9 W=16.2E-6 AD=14.5800004014429E-12
+AS=24.2999995125892E-12 PD=1.7999999499807E-6 PS=19.2000006791204E-6 M=1
M57 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=50.7600003696318E-12
+AS=38.0699985425004E-12 PD=25.949999326258E-6 PS=3.599998999614E-6 M=1
M59 0 3 4 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M61 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
```

```

+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M63 0 3 4 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=38.0699985425004E-12 PD=3.599998999614E-6 PS=3.599998999614E-6 M=1
M65 4 3 0 0 HP14TBN L=900E-9 W=21.15E-6 AD=38.0699985425004E-12
+AS=47.2499990522568E-12 PD=3.599998999614E-6 PS=7.57499992687372E-6 M=1
M67 OUT 9 0 0 HP14TBN L=2.25E-6 W=2.4E-6 AD=11.160000074485E-12
+AS=10.799999783373E-12 PD=11.6999999590917E-6 PS=11.4000004032278E-6 M=1
U69 12 0 13 0 UMODELX L=563.599984161556E-6
U71 11 0 13 0 UMODELY L=444.050005171448E-6
U73 10 0 12 0 UMODELY L=482.150004245341E-6
X75 4 10 BUMP_BSUBCKT_1
X77 9 11 BUMP_BSUBCKT_1
.lib "/ncsu/cadence/local/models/hspice/hp14tb/hsp_nom_lib" hp14tbp
.lib "~/cadence/umodelY.m" umodelY
.lib "~/cadence/umodelX.m" umodelX
.lib "/ncsu/cadence/local/models/hspice/hp14tb/hsp_nom_lib" hp14tbn
* Include files
VDD VDD! 0 3.3
V IN 0 0 PULSE (0 3.3 0N 0.5N 0.5N 3N 6N)
.TRAN 0.1N 15N
.OPTIONS LIST NODE POST
* End of Netlist
.SUBCKT BUMP_BSUBCKT_1 1 2
RBUMP 1 3 17.2M
LBUMP 3 2 8.94P
CBUMP 2 0 18.4F
.ENDS BUMP_BSUBCKT_1
.TEMP 25.0000
.END

```

Appendix G – Transmission Line Models

UMODEL X

```
.lib umodelX

.MODEL umodelX U level=3
+plev=1
+elev=2
+r11=7.6e2
+cr1=150e-12
+l11=508e-9

.endl umodelX
```

UMODEL Y

```
.lib umodelY

.MODEL umodelY U level=3
+plev=1
+elev=2
+r11=7.6e2
+cr1=116e-12
+l11=508e-9

.endl umodelY
```

BUMP

```
.SUBCKT &1 1 2

Rbump 1 3 &2
Lbump 3 2 &3
Cbump 2 0 &4

.ENDS &1
```

Appendix H – divaLVS.rul file

```
/*
 * LVS rules file for SHOCC package
 * Last edit Feb 2' 2001
 */

lvsRules(
procedure(combineParallelFET(m1 m2)
  prog((mt m_m1 m_m2)
    (mt = list(nil))
    if(((m1->l) != nil) && ((m2->l) != nil)) then
      if(((m1->l) != (m2->l))
        return("doNotCombine")
      )
      (mt->l = (m1->l))
    )
    if((m1->m) != nil) then
      (m_m1 = (m1->m)) else
      (m_m1 = 1)
    )
    if((m2->m) != nil) then
      (m_m2 = (m2->m)) else
      (m_m2 = 1)
    )
    if(((m1->w) != nil) && ((m2->w) != nil)) then
      (mt->w = ((m_m1 * (m1->w)) + (m_m2 * (m2->w))))
    )
    return(mt)
  )
)
procedure(compareFET(m1 m2)
  prog((errMsg err m)
    (errMsg = "")
    (err = nil)
    if((m1->l) == nil) then
      (errMsg = strcat(errMsg "Missing transistor length property \"l\"
(layout)\n"))
      (err = t)
    )
    if((m2->l) == nil) then
      (errMsg = strcat(errMsg "Missing transistor length property \"l\"
(schematic)\n"))
      (err = t)
    )
    if((m1->w) == nil) then
      (errMsg = strcat(errMsg "Missing transistor width property \"w\"
(layout)\n"))
      (err = t)
    )
    if((m2->w) == nil) then
      (errMsg = strcat(errMsg "Missing transistor width property \"w\"
(schematic)\n"))
      (err = t)
    )
    if(err then
      return(errMsg)
    )
    if(((abs(((m1->l) - (m2->l))) * 1000000.0) > 0.01) then
      (errMsg = strcat(errMsg
sprintf(nil "Transistor length mismatch: layout %.2f um,
schematic %.2f um\n"
          ((m1->l) * 1000000.0)
          ((m2->l) * 1000000.0)
        )
      )
    )
  ))
)
```



```

procedure(combineParallelCap(c1 c2)
  prog((ct)
    (ct = list(nil))
    if((((c1->c) != nil) && ((c2->c) != nil))
      (ct->c = ((c1->c) + (c2->c)))
    )
  return(ct)
)
)
procedure(combineSeriesCap(c1 c2)
  prog((ct)
    (ct = list(nil))
    if((((c1->c) != nil) && ((c2->c) != nil))
      (ct->c = (((c1->c) * (c2->c)) / ((c1->c) + (c2->c))))
    )
  return(ct)
)
)
removeDevice("ulwireSX" short("in" "out"))
removeDevice("bump" short("in" "out"))
removeDevice("ulwireSY" short("in" "out"))
permuteDevice(parallel "nfet" combineParallelFET)
permuteDevice(parallel "pfet" combineParallelFET)
permuteDevice(parallel "nmos" combineParallelFET)
permuteDevice(parallel "pmos" combineParallelFET)
permuteDevice(parallel "nmos4" combineParallelFET)
permuteDevice(parallel "pmos4" combineParallelFET)
permuteDevice(parallel "res" combineParallelRes)
permuteDevice(series "res" combineSeriesRes)
NCSU_LVSResSlack = 0.10
compareDeviceProperty("res" compareResistor)
permuteDevice(parallel "cap" combineParallelCap)
permuteDevice(series "cap" combineSeriesCap)
compareDeviceProperty("nfet" compareFET)
compareDeviceProperty("pfet" compareFET)
compareDeviceProperty("nmos" compareFET)
compareDeviceProperty("pmos" compareFET)
compareDeviceProperty("nmos4" compareFET)
compareDeviceProperty("pmos4" compareFET)
permuteDevice(MOS "nfet")
permuteDevice(MOS "pfet")
permuteDevice(MOS "nmos")
permuteDevice(MOS "pmos")
permuteDevice(MOS "nmos4")
permuteDevice(MOS "pmos4")
ignoreTerminal("nfet" "B" )
ignoreTerminal("pfet" "B" )
ignoreTerminal("nmos" "B" )
ignoreTerminal("pmos" "B" )
ignoreTerminal("nmos4" "B" )
ignoreTerminal("pmos4" "B" )
)

```