TOWARDS CELL-AWARE TEST FOR SMALL DELAY FAULTS

Institute for Computer Engineering and Computer Architecture
University of Stuttgart

STUDY PROJECT

Suraj Rao Bappanadu
INFOTECH

Advisor: Zahra Paria Najafi Haghi
Professor: Dr. Hans-Joachim Wunderlich

May 31, 2021

Abstract

Hidden delay defects are a subsection of small delay defects. It escapes the manufacturing test as the delay due to these defects is well within the timing margins. At the field level, this delay gets stronger and causes early life failure (ELF) of the device. To reduce the ELF, an analysis on such defects should be performed. One of the main sources for these delay additions are physical defects like resistive bridges and resistive opens in the cell. Therefore, in this work, a tool is developed to generate the defect models with resistive defects (resistive bridges and resistive opens) for a given cell using its layout (GDS/GDSII file) and netlist (SPICE netlist). Cell layouts and netlists are obtained from an open cell library based on Nangate FinFET FreePDK 15nm technology. The information of layout is extracted using the python library "python-gdsii". A complete use-case of an Inverter is explained in the below sections and the results of the same are included. For additional proof, few outputs of NAND cell are also included.

Keywords - Early Life Failure, Small Delay Defects, Cell-Aware Test, Resistive Defects, Inverter

Contents

List of Figures

List of Tables

1 Introduction

Nowadays, more transistors are integrated on a single chip, as it contains both analog and digital subsystems. This is directly proportional to the increase in the number of defects [1]. Hence, the demand for higher quality tests is necessary for the various stages of development. In the manufacturing test, there is a certain level of the threshold set for the delay measured in the transistors. If the delay of the transistor exceeds this level, then it is said to be defective and the total defectiveness is measured per million, given as defective parts per million (DPPM). As technology scales down, a lot of defects can be observed which are mainly related to timing. Small delay defects (SDD) are a form of timing-related defect. Hidden delay defects are a subsection of small delay defects. These defects cause a small amount of extra delay in the circuit, which is less than the threshold set in the manufacturing test and hence escapes that test. **This delay might increase in the field level due to various parameters and becomes stronger and results in early life failure (ELF) of the device [2]**. The main sources for the delay addition are process variation effects, crosstalk effects, power supply noise effects and physical defects like resistive open (resistive cuts) and resistive bridge (resistive shorts) [3].

As per research, the majority of timing-related defects are found in the standard cell library and hence cell internal defects are considered as the most predominant [4]. As a result, testing is needed using the information available in the standard cell libraries. Cell-aware test (CAT) is a method of testing a cell by extracting its layout information, and it has proven to be the successor of Inductive Fault Analysis (IFA) [5]. It is considered as the new era in the defect modeling and defect circuit analysis.

The main focus of this Study Project will be on physical defects such as resistive bridges and resistive opens that cause a slight delay in the cell. It is very important to analyse such small delays due to resistive defects even though the delay is well within the timing margins employed at the test [6]. The cell layout from the open cell library, known as a GDS/GDSII file, will be used as the input for the tool in the first stage. This file contains cell layout information and is in binary format, so it will be converted to a text file (human-readable format) [F1]. The required information to insert resistive bridges and resistive opens, taken from the text file[F2] and the SPICE netlist of that specific cell are used as input in the second stage. This results in the automatic generation of defect models related to resistive bridges and resistive opens of that cell [F3]. These defect models can be used further for the analysis of small delays in the cell.

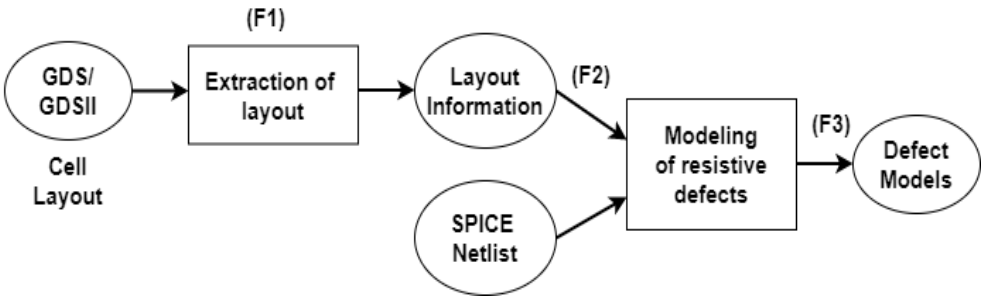


Figure 1: Focus of Study Project

Preliminaries and all necessary background information are explained in section 2 followed by tool overview in section 3. Results and their discussions are explained in section 4. Finally, the conclusions have been noted in section 5.

2 Preliminaries

2.1 Open Cell Library

Open Cell Library (OCL) is one of the most used libraries for independent Electronic Design Automation (EDA) flow testing, development of cell-based design flows, EDA tools and academic research. This helps universities and other research institutions to have access to the information needed to perform the research and design for advanced technology nodes. OCL can be roughly defined as the physical design kit (PDK) for an integrated circuit design.
In this Study Project, we are interested in the OCL based on Nangate FinFET FreePDK 15nm technology. The 15nm OCL is based on a generic predictive state-of-the-art technology node [7] - it mainly contains a package of standard cells with their corresponding information. It consists of 76 cells which includes sequential cells such as flip-flops, scan flops, latches and combinational cells such as AND, Inverter, AOI (AND-OR-Invert) at different drive strengths. It is being released with the necessary views in the front end and back end to perform the tasks involved in the integrated circuit design. The main focus of this work is on the information of the cell available in cell layouts and cell netlists.

2.1.1 Cell Layout (GDS/GDSII File)

GDS/GDSII file is a file in the binary format which is platform-independent and represents text labels, planar geometric shapes and other information about the layout of the cell/circuits in the hierarchical form [8]. GDSII format is a sequential list of records; each record is of variable length and contains a header to tell what information is in the record as shown in Fig. 2.

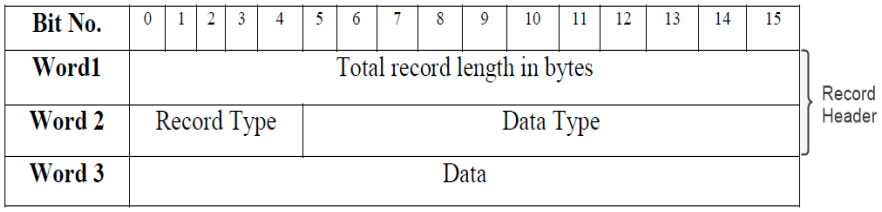


Figure 2: Record Fields

The first two bytes of the record header contain the total record length. The third byte is the record type - there are around 57 different record types. The fourth byte is the data type - there are around 7 different data types. The fifth byte onwards contains the data.

GDS file imported in Cadence Virtuoso environment with the FreePDK15 layer map file is as shown in the below Fig. 3 and the layer description is as shown in Table 1.

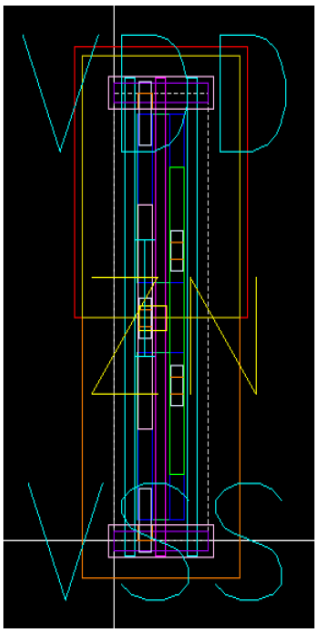


Figure 3: Layout of Inverter

Layer No.	Layer Name	Description
00	NW	N-well , Single well layer
01	ACT	Active area for fin definition
05	NIM	n-implant, N-type FET
06	PIM	p-implant, P-type FET
07	GATEA	Gate metal region, color A
08	GATEB	Gate metal, dummy gates for protection, color B
09	GATEC	Gate metal, Gate cut mask layer
11	AIL1	Active Interconnect layer, Layer 1
12	AIL2	Active Interconnect layer, Layer 2
13	GIL	Gate Interconnect Layer
14	V0	Via 0: connect between interconnectlayers and metals
15	M1A	Gate metal, First level of Interconnect metal, color A
16	M1B	Gate metal, First level of Interconnect metal, color B
200	prBoundary	Boundary of the cell

Table 1: Layer Description

An example of few layers of an Inverter observed in Cadence Virtuoso tool is shown in Fig. 4.

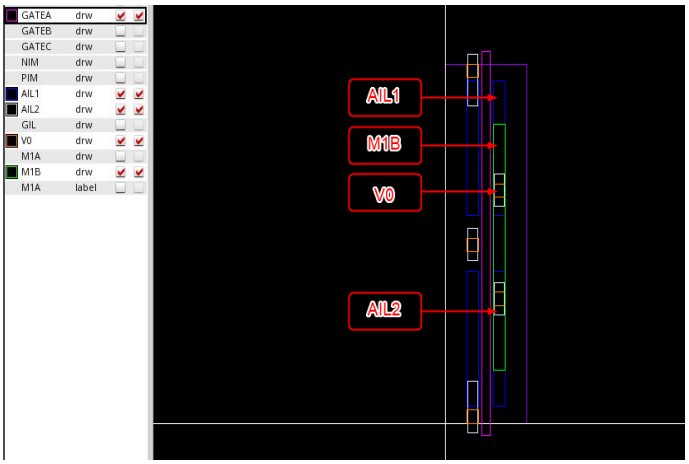


Figure 4: Layers in GDS File

2.1.2 Cell Netlist (SPICE File)

The elements in the circuit are described in the form of text statements, frequently called a netlist [9]. In the OCL, cell netlists are described in the form of subcircuits. A subcircuit is a method of defining a circuit block so that the main program can instantiate it any number of times by a single netlist line.

2.2 FinFET Technology

Integrated Circuits fabricated using FinFET technology involve a lot of technical challenges including limitation of the optical lithography process [10]. The presence of fins and the introduction of specific interconnect layers between the metal and fins are the main differences in FinFET technology when compared to other standard CMOS fabrication process.

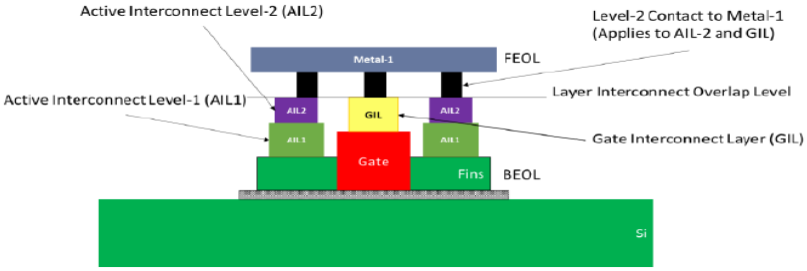


Figure 5: Cross Section of FinFET
[10]

Three local interconnect layers:
1. AIL1: Used for connecting individual fins of FinFET
2. AIL2: Used for connecting AIL1 to the metal through a via
3. GIL: Used for connecting gate of the device to the metal

2.3 Small Delay Defects

Small delay defects are one type of timing defect, which introduces a small amount of extra delay in the circuit [3]. Although the delay introduced is small, the overall impact can be significant if the sensitized path is long and might result in early life failure of the device. Therefore, it is necessary to include tests which cover these defects. One of the main sources for such delay additions are physical defects like resistive bridges and resistive opens.

2.4 Resistive Bridges/Resistive Opens

A resistive open, also known as resistive cut, is an imperfect circuit connection between two nodes in the netlist. These nodes were supposed to be connected ideally but now have a resistor between them. Examples of resistive open defects are thin wires [11], ill-formed contacts (vias) [12], resistance between input and gate terminal of transistor. These open defects do not change the output state, but can introduce a finite or infinite amount of delay for signal transition [13]. A resistive bridge, also known as resistive short, is a short whose resistance is not that high enough such that it can be ignored. Along with resistive bridge, another resistive open has to be added, so that the current flows through the path with the resistive bridge. Examples of resistive bridge defects include shorts between transistor gates and drain, gate and source, drain and source, and short between interconnections [14].

An example of few possible sites at which resistive bridge (RB) and resistive open (RO) can be introduced at an Inverter is as shown in the below Fig. 6.

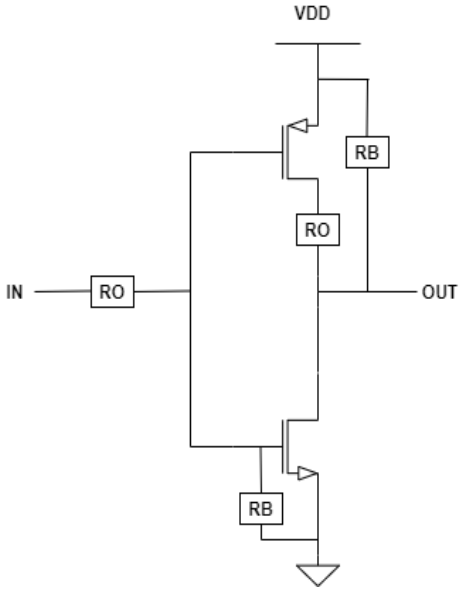


Figure 6: Resistive Bridge/Open: Inverter

3 Tool Overview

GDS/GDSII is a binary file that includes information on the cell layout. In this work, Cadence Virtuoso tool is used to open the GDSII file and to obtain the summary on the properties of the layout. HSPICE is a tool to perform the different types of analysis such as transient analysis, direct current (DC) analysis of the circuit at the electrical level. The two main inputs to the tool which is developed in this work are the cell layout (GDS/GDSII file) and the cell netlist (SPICE netlist). Cell-Aware Defect Modeling is the name given to the tool. Among the generated output files, the main files are the text file which contains complete information on the cell layout and the defect models (SPICE netlist) with resistive bridges and resistive opens. A complete step-by-step procedure of the tool is shown below in the Fig .7

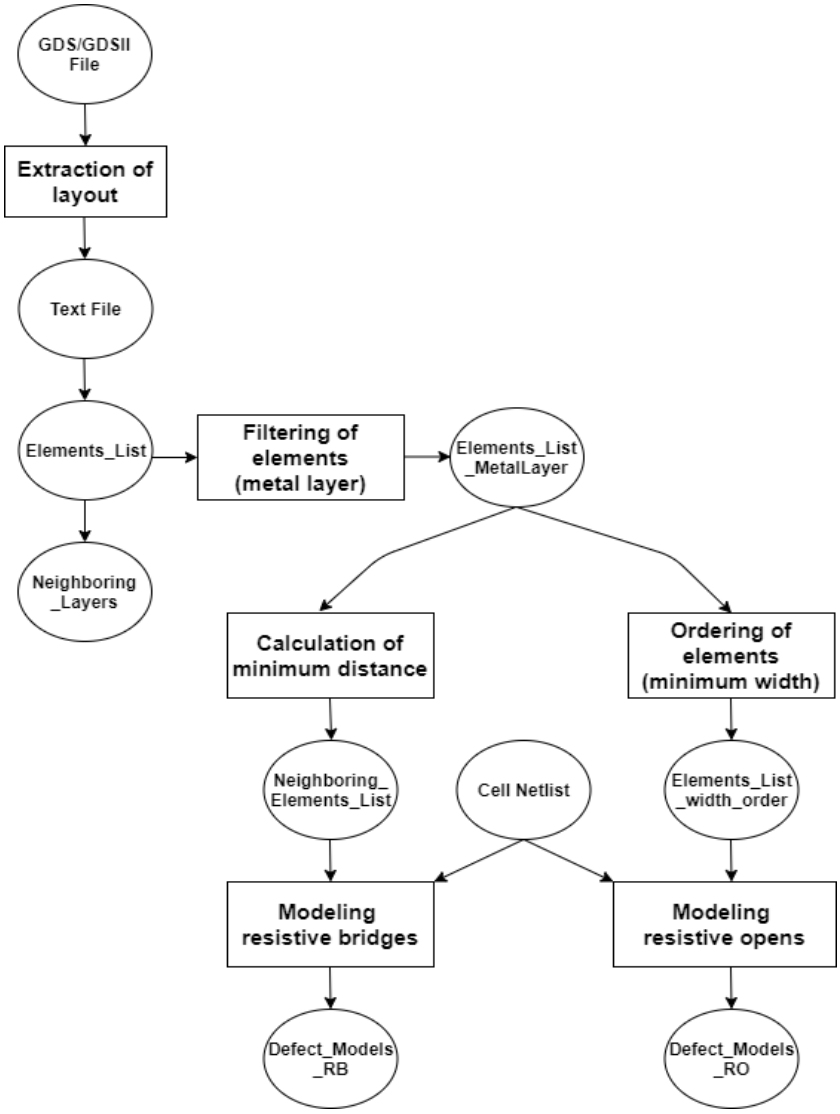


Figure 7: Step-By-Step Procedure of Tool

3.1 Layout Information Extraction

As explained earlier, GDS/GDSII file is a binary format file. To be able to use the information in it, it is converted to human-readable format (text file). The information of layout is extracted using the python library "python-gdsii". It is a standard library that is used to create, read, modify and save gds/gdsii files [15]. Classes defined under this library are used to convert the binary information in the gds file to readable format.

Modules in the library:
- gdsii.library.Library - GDSII library class.
  load(stream) - returns the name of the library
- gdsii.types - contains definitions of GDSII data types
- gdsii.record.Record - GDSII record with data attached to it.
  iterate(stream) - iterates over all the records in the GDSII file and returns Record object.

Based on the data types, the records can be differentiated in the stream file. The available data types with their values are shown in Table 2.

DataType	Value
No Data	00
Bit Array	01
Two byte signed integer	02
Four byte signed integer	03
Four byte real	04
Eight byte real	05
ASCII string	06

Table 2: DataTypes Description

Example of Records:
1. RecordType: LIBNAME
   DataType: ASCII
   Data: NanGate_15nm_OCL
   Description: LIBNAME is a type of record which gives the information on the name of the library in the GDS file. The data in the record is of datatype ASCII (Value 06). Data field contains the information/data of the record. Here it is of type ASCII string. Value is "NanGate_15nm_OCL"

2. RecordType: LAYER
   DataType: INT2
   Data: (5,)

Description: LAYER is a type of record which gives information on the layer number in which element is located. The data in the record which is always a number, is of datatype INT2 (Value 02). Data field: Value = 05. Element is found in layer 05.

As observed in the converted text file, the initial information in the GDS/GDSII file is all about the name of the standard cell, name of the library, time and date of creation/modification of standard cell. Then the information about the layout is explained in the form of boundary elements and text elements in different layers.

3.2 Elements List

From the text file, the relevant information needed is the list with the elements only. Every boundary element is defined in a particular layer and is in rectangular shape. The following properties (records) are defined for every boundary element:
- Layer: The number in which the element is.
- Datatype: Datatype of the element (0: NO_DATA)
- XY: Position of the element. 5 XY pair coordinates are defined.
- XY = (x1y1, x2y2, x3y3, x4y4, x5y5)
- First(x1y1) and last(x5y5) coordinates to remain same to make sure element is closed explicitly.
- ENDEL: Element is finished with ENDEL record.

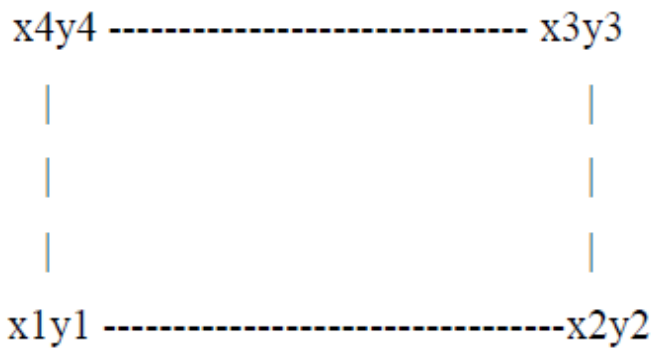


Figure 8: Boundary Element

Similar to the boundary element, text element is defined, which contains the text/string related to the boundary element. The below properties (records) is defined for the text element:
- LAYER: In which layer the text is found
- TEXTTYPE: Type of text
- PRESENTATION: FONT, Vertical/Horizontal, Middle/Bottom
- STRANS: Related to angle of the text
- MAG: Magnification factor
- XY: Position of the text
- STRING: Text to be displayed

- ENDEL: Element is finished with ENDEL record

From the list of elements, a new list with the elements arranged based on increasing order of width is obtained. Such a list is necessary for the analysis of resistive opens and will be explained in detail in further subsections. Width calculation in the script is as explained below:
For the given boundary element as shown in Fig. 8, minimum width is calculated as:
minimum_width = minimum(x2 - x1, y3 - y2)
minimum() - Function gives the minimum value of the given input parameters.

3.2.1 Elements List Metal Layer

From the list with elements arranged based on its width order, the list with elements only in the metal layer is obtained because defects like resistive bridges and resistive shorts can be introduced in the elements in the metal layer only. This is achieved by filtering the list with the elements in the layer no. 15 and layer no. 16 because metal elements are found in this layer.

As mentioned earlier, boundary element is associated with a set of XY pairs which denotes its position, whereas text element is associated with a single XY pair which denotes its position. These XY pairs can all be denoted as points in a 2D coordinate system. With the boundary element XY pairs, a polygon is constructed and if the single XY pair point of the text element is inside the polygon, then that particular string (data) of the text element is mapped to that respective boundary element. This helps in the mapping of the respective text element with the boundary element in the metal layer. Python classes Shapely, Polygon and Point are used.

3.2.2 Neighboring Elements List

As shown in Table 1, a layer number is assigned to each layer. With this information, a list with neighboring layers for each layer is generated.

As mentioned earlier, it is necessary to add the resistive bridge between two elements in the metal layer. For any given element in the metal layer, its distance from all the neighboring elements belonging to the metal layer are calculated and the one with the closest value is chosen and a resistive bridge is added between them. Elements could be placed in one of the possible ways as shown in Fig. 9. Minimum distance calculation between the elements would be different in each of the cases and is given below:

1. Case 01: Minimum distance is given by d.
2. Case 02: Minimum distance is 0.
3. Case 03: Minimum distance is given by d.
4. Case 04: Minimum distance is calculated using Pythagorean theorem, d = $\sqrt{a^2 + b^2}$

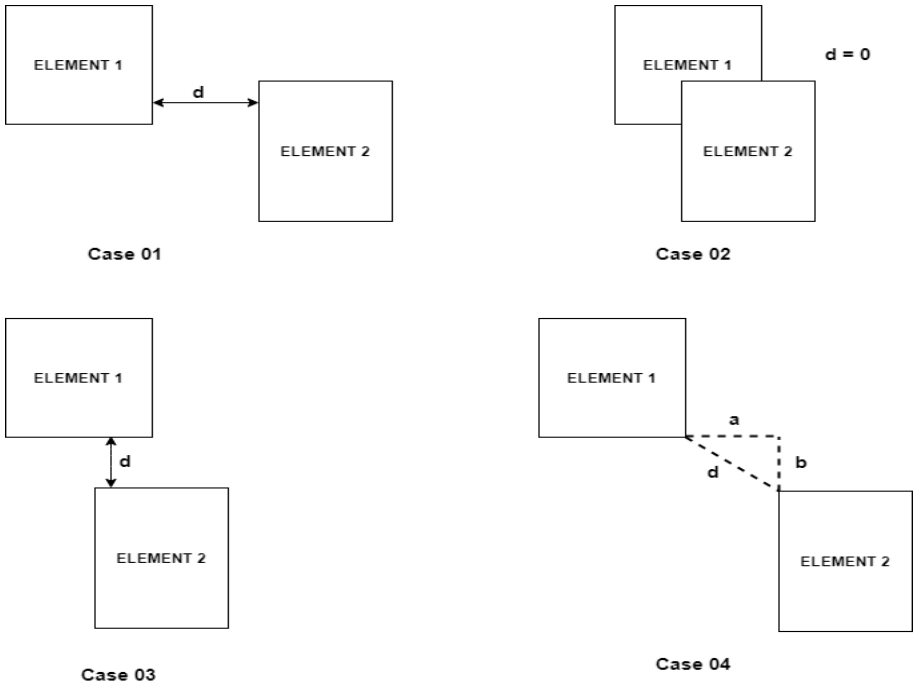


Figure 9: Neighboring Elements Distance Calculation

3.3 Defect Models

Defect models with resistive bridges will be created for each element in the metal layer using the list of closest neighboring element. The cell's SPICE netlist is one of the inputs to the tool that will be considered in this step. Along with resistive bridge, it is mandatory to add another resistive open with a small resistance value, to avoid the current flow from passing through the path with no resistance and to make sure the current flows through the path with the resistive bridge.

From the list in which elements in the metal layer are arranged based on their increasing width order, a sub list will be obtained with the elements whose width will be less than the threshold set by user. Using this list and the cell's SPICE netlist, defect models with resistive opens will be created.

The default resistance would be "x0=1000" in both the cases and it is also added in the ".SUBCKT" line so that whenever the cell is instantiated, the resistance value can be varied and it would be helpful for further delay analysis.

4 Results

The two main inputs to be considered for the tool are GDSII file of the cell and SPICE netlist of the cell. The generated output files are the text file, elements list, elements list with increasing width order, elements list with metal layer only, neighboring layers, neighboring elements list, defect models with resistive bridges and defect models with resistive opens.

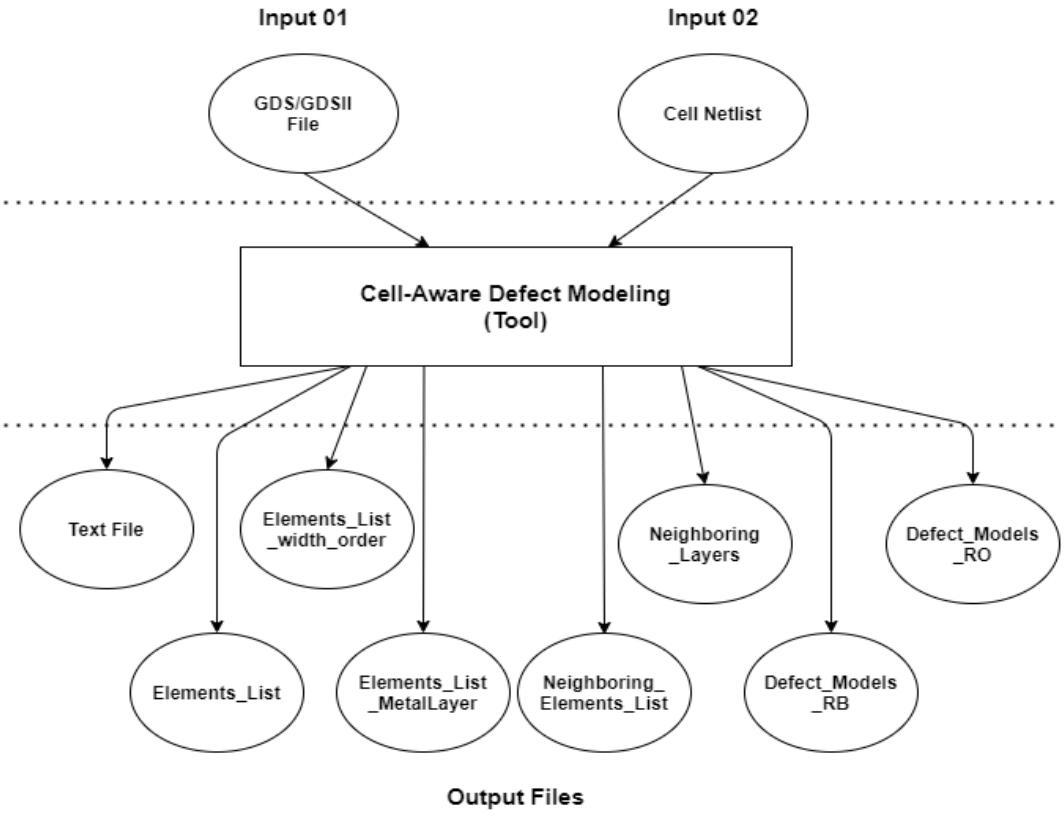


Figure 10: Inputs and Outputs of the Tool

Most of the results include text files generated from python script. As text files can't be copied completely, the necessary screenshots from the output of python code and the text file will be included and explained in detail for one particular cell (Inverter Cell). For additional proof, the necessary output of another cell will also be included (NAND Cell).

4.1 Text File Information

The generated text file gives the complete information of the cell layout. Most of the results in the text file can be validated with the information of the layout from the Cadence Virtuoso tool. An example of a boundary element and text element is shown below in Fig. 11 and Fig. 12 respectively.

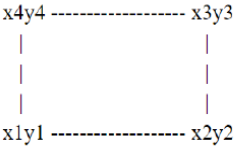
TEXT FILE

RecordType:BOUNDARY
DataType:NODATA
Data:None

RecordType:LAYER
DataType:INT2
Data:(0,)
LayerNumber:0
Layer Name: NW

RecordType:XY
DataType:INT4
Data:(-80, 384, 272, 384, 272, 848, -80, 848, -80, 384)
Area:163328.0U

RecordType:ENDEL
DataType:NODATA
Data:None



CADENCE TOOL

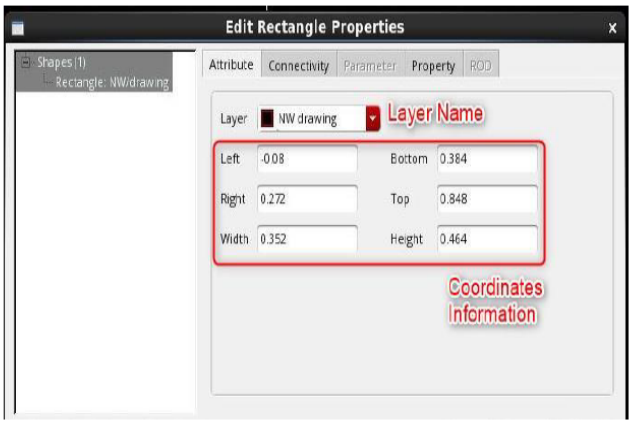


Figure 11: Comparison of Boundary Element

TEXT FILE

RecordType:TEXT
DataType:NODATA
Data:None

RecordType:LAYER
DataType:INT2
Data:(15,)
LayerNumber:15
Layer Name: M1A

RecordType:XY
DataType:INT4
Data:(93, 768)

RecordType:STRING
DataType:ASCII
Data:VDD

RecordType:ENDEL
DataType:NODATA
Data:None

CADENCE TOOL

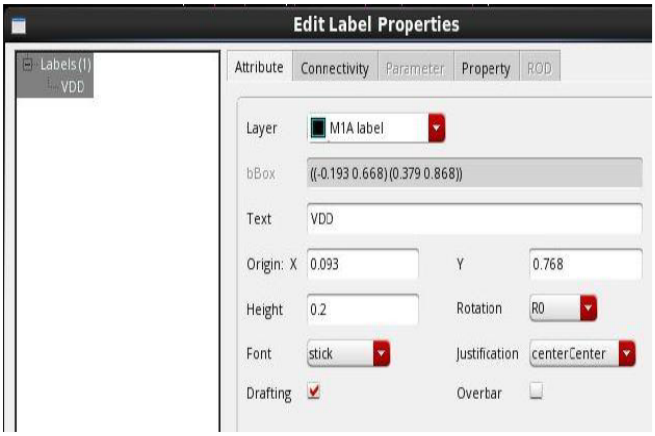


Figure 12: Comparison of Text Element

4.2 Element List Information

The generated element list file contains information on the layer, layer number, minimum width and the position of the elements. A part of the output for an Inverter is as below:

GDS:FileName:INV_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position
1: NW: 0: 352.0: ['-80', '384', '272', '384', '272', '848', '-80', '848', '-80', '384']
2: ACT: 1: 96.0: ['48', '36', '144', '36', '144', '324', '48', '324', '48', '36']
3: ACT: 1: 96.0: ['48', '444', '144', '444', '144', '732', '48', '732', '48', '444']

Similarly for a NAND cell is as below:

GDS:FileName:NAND2_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position
9: GATEC: 9: 32.0: ['0', '752', '256', '752', '256', '784', '0', '784', '0', '752']
10: NIM: 5: 384.0: ['-64', '-64', '320', '-64', '320', '384', '-64', '384', '-64', '-64']
11: PIM: 6: 384.0: ['-64', '384', '320', '384', '320', '832', '-64', '832', '-64', '384']
12: AIL1: 11: 28.0: ['50', '36', '78', '36', '78', '324', '50', '324', '50', '36']

The generated element list width order file contains the same information as the element list file, but the elements are arranged in increasing order of width. A part of the output for an Inverter is as below:

GDS:FileName:INV_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position
8: GATEC: 9: 32.0: ['0', '752', '192', '752', '192', '784', '0', '784', '0', '752']
20: GIL: 13: 44.0: ['52', '360', '108', '360', '108', '404', '52', '404', '52', '360']
27: M1A: 15: 56.0: ['-10', '-28', '202', '-28', '202', '28', '-10', '28', '-10', '-28']

Similarly for a NAND cell is as below:

GDS:FileName:NAND2_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position
37: M1A: 15: 56.0: ['-10', '740', '266', '740', '266', '796', '-10', '796', '-10', '740']
24: AIL2: 12: 68.0: ['139', '348', '236', '348', '236', '416', '139', '416', '139', '348']
2: ACT: 1: 160.0: ['48', '36', '208', '36', '208', '324', '48', '324', '48', '36']
10: NIM: 5: 384.0: ['-64', '-64', '320', '-64', '320', '384', '-64', '384', '-64', '-64']

Elements in the metal layer for an Inverter is shown in the Fig. 13. Minimum width calculation for VDD and ZN is shown below:
minimum_width (VDD) = minimum (212,56) = 56
minimum_width (ZN) = minimum (28,526) = 28

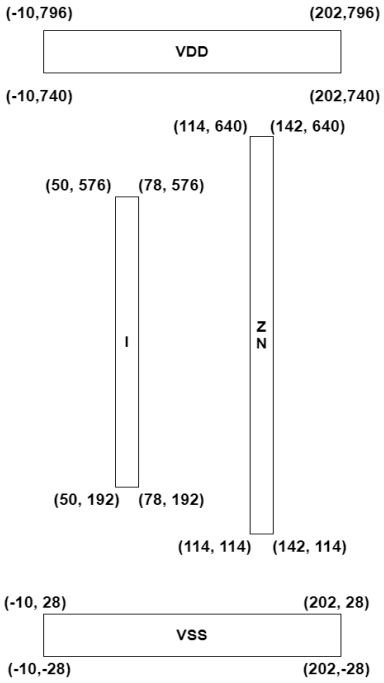


Figure 13: Elements of Inverter in Metal Layer

4.2.1 Element List Metal Layer Information

The generated element list metal layer file contains information on the layer, layer number, minimum width, text string and the position of the elements in the metal layer only. An output for an Inverter is as below:

GDS:FileName:INV_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position : Text String
26: M1A: 15: 28.0: ['50', '192', '78', '192', '78', '576', '50', '576', '50', '192']: I
29: M1B: 16: 28.0: ['114', '114', '142', '114', '142', '640', '114', '640', '114', '114']: ZN
27: M1A: 15: 56.0: ['-10', '-28', '202', '-28', '202', '28', '-10', '28', '-10', '-28']: VSS
28: M1A: 15: 56.0: ['-10', '740', '202', '740', '202', '796', '-10', '796', '-10', '740']: VDD
Here I - Input, ZN - Output, VSS - Ground and VDD - Power Supply.

Similarly for a NAND cell is as below:

GDS:FileName:NAND2_X1.gds
Sl. No : Layer Name : Layer No : Minimum_Value(nm) : XY Position : Text String
34: M1A: 15: 28.0: ['50', '256', '78', '256', '78', '574', '50', '574', '50', '256']: A2
35: M1A: 15: 28.0: ['178', '256', '206', '256', '206', '574', '178', '574', '178', '256']: A1
Here A1 and A2 are two inputs of NAND cell.
The boundary element and text element belonging to the metal layer will be mapped in this step.

Example:
Boundary element: (-10, 740, 202, 740, 202, 796, -10, 796, -10, 740)
Text element: (93, 768), STRING: VDD

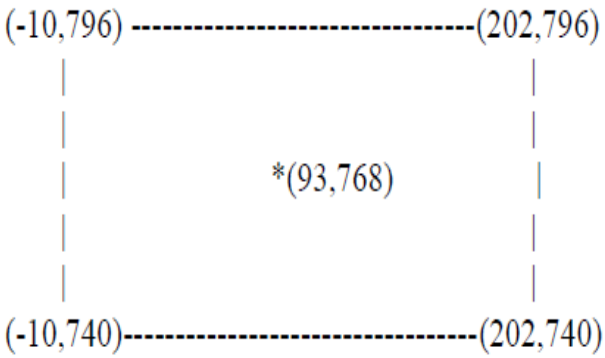


Figure 14: Mapping of Boundary and Text Element

4.2.2 Neighboring Layers

The generated neighboring layers file contains information on the neighboring layers to each layer in the cell layout. A part of the output is as below:
Current Layer : Previous Layer : Next Layer
GATEB: GATEA: GATEC
AIL2: AIL1: GIL

4.2.3 Neighboring Elements List Information

The generated neighboring elements list file contains information on the minimum distance of the neighboring elements from each element and gives the closest element to that particular element in the metal layer.
Example:
1. For the input (I) of an Inverter:
   Distance between the elements in the metal layer :
   I and ZN is 36.0 nm
   I and VSS is 164.0 nm
   I and VDD is 164.0 nm
   Closest Element to I-ZN
2. For the ground (VSS) of a NAND cell:
   Distance between the elements in the metal layer :
   VSS and A2 is 228.0 nm
   VSS and A1 is 228.0 nm
   VSS and ZN is 105.0 nm
   VSS and VDD is 712.0 nm
   Closest Element to VSS-ZN

```
surajraob0108@surajraob:~/Suraj/Study_Project/Text_Files/Inverter$ python3 gds_t
o_text.py
Minimum distance between the elements :712.0 nm
```

Figure 15: Minimum Distance between VDD and VSS of an Inverter

4.3 Defect Models

In the final step, defect models with resistive bridge and resistive open are generated. For an Inverter, 4 elements are found in the metal layer, so 4 defect models with resistive bridges are generated. An input threshold of ”40 nm” is set, thus 2 defect models with resistive opens are generated.

4.3.1 Resistive Bridges

From the result of neighboring elements list, input (I) is closest to output (ZN) in an Inverter. Therefore, a resistive bridge is added with an additional node ZN_0 with default resistance ”x0=1000”. Resistive open is added between ZN_0 and ZN with resistance value = 50.

```
.SUBCKT INV_X1 I ZN VDD VNW VPW VSS x0=1000
*.PININFO I:I ZN:O VDD:P VNW:P VPW:P VSS:G
*.ORDERING_0_PUP M_i_1 VDD ZN
*.ORDERING_0_PDN M_i_0 VSS ZN
*.EQN ZN_0 =!I
M_i_0 ZN_0 I VSS VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_1 ZN_0 I VDD VNW pfet W=0.288000U L=0.020000U nfin=7
RB1 I ZN_0 x0
RO1 ZN_0 ZN 50
.ENDS
```

Figure 16: Netlist of an Inverter with Resistive Bridge

```
.SUBCKT NAND2_X1 A1 A2 ZN VDD VNW VPW VSS x0=1000
*.PININFO A1:I A2:I ZN:O VDD:P VNW:P VPW:P VSS:G
*.ORDERING_0_PUP M_i_3 VDD ZN_0 M_i_2 ZN_0 VDD
*.ORDERING_0_PDN M_i_1 VSS net_0 M_i_0 net_0 ZN
*.EQN ZN_0 =!(A1 * A2 )
M_i_1 net_0 A2 VSS VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_0 ZN_0 A1 net_0 VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_3 ZN_0 A2 VDD VNW pfet W=0.288000U L=0.020000U nfin=7
M_i_2 VDD A1 ZN_0 VNW pfet W=0.288000U L=0.020000U nfin=7
RB2 A1 ZN_0 x0
RO1 ZN_0 ZN 50
.ENDS
```

Figure 17: Netlist of NAND cell with Resistive Bridge

4.3.2 Resistive Opens

From the list of elements with width order and the user input, defect model with resistive open is generated at the input (I) with the default resistance "x0=1000" for an Inverter.

```
.SUBCKT INV_X1 I ZN VDD VNW VPW VSS x0=1000
*.PININFO I:I ZN:O VDD:P VNW:P VPW:P VSS:G
*.ORDERING_0_PUP M_i_1 VDD ZN
*.ORDERING_0_PDN M_i_0 VSS ZN
*.EQN ZN=!I
M_i_0 ZN I VSS VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_1 ZN I VDD VNW pfet W=0.288000U L=0.020000U nfin=7
RO1 I N1 x0
.ENDS
```

Figure 18: Netlist of an Inverter with Resistive Open

```
.SUBCKT NAND2_X1 A1 A2 ZN VDD VNW VPW VSS x0=1000
*.PININFO A1:I A2:I ZN:O VDD:P VNW:P VPW:P VSS:G
*.ORDERING_0_PUP M_i_3 VDD ZN M_i_2 ZN VDD
*.ORDERING_0_PDN M_i_1 VSS net_0 M_i_0 net_0 ZN
*.EQN ZN=!(A1 * A2)
M_i_1 net_0 A2 VSS VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_0 ZN A1 net_0 VPW nfet W=0.288000U L=0.020000U nfin=7
M_i_3 ZN A2 VDD VNW pfet W=0.288000U L=0.020000U nfin=7
M_i_2 VDD A1 ZN VNW pfet W=0.288000U L=0.020000U nfin=7
RO1 A2 N1 x0
.ENDS
```

Figure 19: Netlist of NAND cell with Resistive Open

5 Conclusion

Increase in the number of transistors on a single chip is directly proportional to an increase in the number of defects. Before the chip is sent to the field, manufacturing test helps to find the transistors in which the delay crosses certain threshold level. This helps to avoid the failure of device and reduces DPPM. However, if there is a small delay which is less than the threshold, it escapes the manufacturing test and is sent to the field. Due to various parameters at the field level, these small delays get stronger and will result in early life failure (ELF) of the device. One of the main sources for such delay additions are physical defects like resistive bridges and resistive opens. To reduce the ELF, an analysis on these defects should be performed. Therefore, in this work, a tool is developed to generate the defect models with resistive defects (resistive bridges and resistive opens) for a given cell using its layout (GDS/GDSII file) and netlist (SPICE netlist). The first step is to extract the layout information from the cell layout into a text file. The information in the text file is validated with the layout information obtained from the Cadence Virtuoso tool. Resistive defects affect the elements in the metal layer. By filtering the necessary information on the metal layer elements in the text file and with the help of cell netlist, defect models with resistive bridges and resistive opens are generated. A complete use-case of an Inverter is considered and explained in above sections and results of the same are included. For additional proof, few outputs of a NAND cell are also included.

As a next step, with the help of defect models, cells can be instantiated with different resistance values and transient analysis can be carried out with the help of SPICE tool. This helps in analysis of small delay due to insertion of resistive bridges and resistive opens in the cell.

References

[1] Dennis R. E. Gnad, J.K., Tahoori, M.B.: Leaky noise: New side-channel attack vectors in mixed-signal iot devices. IACR Transactions on Cryptographic Hardware and Embedded Systems (2019) 1–3

[2] Haghi, Z.P.N., et al: Variation-aware defect characterization at cell level. (May 2020) 1–6

[3] Tehranipoor, M., ke Peng Krishnendu Chakrabarty: Test and diagnosis for small-delay defects. (2011)

[4] Hapke, F., et al: Cell-aware test. IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems (vol. 33, no. 9) (Sept 2014) 1396–1409

[5] J. P. Shen, W.M., Ferguson, F.J.: Inductive fault analysis of mos integrated circuits. IEEE Design Test of Computers (vol. 2, no. 6) (Dec 1985) 13–26

[6] Haihua Yan, A.D.S.: A delay test to differentiate resistive interconnect faults from weak transistor defects. 18th International Conference on VLSI Design (2005) 1–6

[7] Mayler Martins, Jody Maick Matos, J.M., Rech, L.: Open cell library in 15nm freepdk technology. (April 2015) 1–8

[8] Information, G.F.: (www.boolean.klaasholwerda.nl/interface/bnf/gdsformat.html)

[9] Synopsys: Hspice user guide: Basic simulation and analysis. (Version M-2017.03-SP1) (June 2017)

[10] Bhanushali, K.: Design rule development for freepdk15: An open source predictive process design kit for 15nm finfet devices. (2014)

[11] Yeoh, E., We, M.: The application of novel failure analysis techniques for advanced multi layered cmos devices. Proc. Int. Test Conf.,Washington, DC (1997) 304–309

[12] K. Baker, G. Gronthoud, M.L.I.S., Hawkins, C.: Defect-based delay testing of resistive vias-contacts. Proc. Int. Test Conf., Atlantic City, NJ (1999) 467–476

[13] Nachiket Rajderkar, Marco Ottavi, S.P., Han, J., Lombardi, F.: On the effects of intra-gate resistive open defects in gates at nanoscaled cmos. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (2011) 1–7

[14] Li, J.C.M., McCluskey, E.J.: Diagnosis of resistive-open and stuck-open defects in digital cmos ics. IEEE Conference (2005) 1–12

[15] Library, P.G.: (www.pypi.org/project/python-gdsii/)