VERY LARGE SCALE INTEGRATION CELL BASED PATH EXTRACTOR FOR PHYSICAL TO LAYOUT MAPPING IN FAULT ISOLATION WORK

MATTHEW PRAGASAM

UNIVERSITY SAINS MALAYSIA
2017

VERY LARGE SCALE INTEGRATION CELL BASED PATH EXTRACTOR FOR PHYSICAL TO LAYOUT MAPPING IN FAULT ISOLATION WORK

By

MATTHEW PRAGASAM

Thesis submitted in fulfillment of the requirements for the degree of Master of Science (Electronic Systems Design Engineering)

August 2017

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my academic and research supervisor Assoc. Prof. Ir. Dr. Mohamad Kamarol Mohd. Jamil for his guidance and constant support in helping me to conduct and complete this work during and after working hours. I would also like to thank him for his ever-ready support in consulting me throughout completing my research.

I am grateful for the facilities and workspace provided by University Sains Malaysia, Penang, which had enabled me to carry out most of my discussions there. Many thanks to all the people I have come to know during my studies in USM Penang, whose friendship I cherish and experiences I will always remember. I owe my sincere appreciation to my family who have provided me constant emotional support and had encouraged me throughout my year of studies. I want to especially thank USM Penang for offering me this opportunity to study the programme of my greatest interest. Finally, I would like to extend my profound appreciation to my beloved parents for their love, affection and invaluable support throughout many late nights completing my work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

API	Application Programming Interface
APR	Auto Place and Route
ASIC	Application Specific Integrated Circuit
BFS	Breadth-First Search
CAD	Computer Aided Design
CBD	Cell-Based Design
DEF	Design Exchange Format
DFS	Depth-First Search
DoS	Depth of Search
DUT	Design Under Test
EDA	Electronic Design Automation
FA	Failure Analysis
FI	Fault Isolation
IC	Integrated Circuit
IM	Information Model
OA	OpenAccess
OS	Operating System

LITAR EKSTRAKTOR INTEGRASI SKALA SANGAT BESAR BERASASKAN SEL UNTUK FIZIKAL KE TATALETAK PEMETAAN DALAM KERJA KESALAHAN PENGASINGAN

ABSTRAK

Siasatan dalam jawatan-silikon semakin mencabar kemajuan teknologi dalam keupayaan Pemetaan Fizikal ke Tataletak. Bidang yang memerlukan inovasi tersebut adalah kesalahan kerja pengasingan dalam analisis kegagalan peranti semikonduktor pada peringkat pasca silikon. Sejak kerja kesalahan pengasingan bermula di peringkat "*Register Transfer Level"* (RTL) untuk membentuk sempadan yang disyaki yang terdiri daripada pelbagai logik dari satu hujung ke hujung yang lain, "*Electronic Design Automation*" (EDA) membantu mengenal pasti kesalahan dalam sempadan yang dinyatakan. Oleh itu, program ekstraktor litar yang mampu mengekstrak semua laluan mungkin dari isyarat awal hingga akhir boleh menjimatkan masa jurutera dalam mengesan komponen yang terlibat antara garis kesalahan dalam skematik. Untuk mendapat semua isyarat yang mungkin terlibat dalam sempadan yang disyaki adalah carian masalah pengiraan popular. Disebabkan itu, program litar ekstraktor yang dicadangkan menggabungkan ciri-ciri algoritma "*Depth-First Search*" (DFS) dengan mengambil kira spesifikasi reka bentuk berasaskan sel. Objektif dicapai dalam kajian ini terbukti dengan keputusan pengekstrakan jalan konsisten walaupun dengan manipulasi "*Depth of Search*" (DoS). Prestasi berbeza purata 12.6 % (kiraan lelaran) dengan menjaga kedalaman maksimum yang dibenarkan carian berterusan. Laluan urutan bersih adalah konsisten sepanjang pengesahan program jalan pemerah. Perkembangan ini dan kajian kaedah ekstrak jalan membawa kepentingan dalam bidang EDA dan kerja siasatan.

VERY LARGE SCALE INTEGRATION CELL BASED PATH EXTRACTOR FOR PHYSICAL TO LAYOUT MAPPING IN FAULT ISOLATION WORK

ABSTRACT

Debug and diagnosis in post-silicon challenges the technological advancement in Physical-to-Layout Mapping capabilities. Areas that require such innovation are fault isolation work in failure analysis of semiconductor devices, at post-silicon stage. Since fault isolation work begins at Register Transfer Level (RTL) level to form a suspected boundary consisting of multiple logics from one end to the other, layout to schematic mapping automation tool helps to identify fault in design within given boundary. Therefore the development of a path extractor program which is capable of extracting all possible paths from these start to end signals can save engineers time in tracing components involved between a fault line. This feature is extremely significant in Electronic Design Automation (EDA) as it can provide results of net name sequences stored in a database of mapper files. These mapper files can be used in layout design debug as the net sequence represents schematic signals. To be able to retrieve all possible signals involved within a suspected boundary is a popular search computational problem. Therefore the path extractor program proposed incorporates the characteristics of a depth-first search algorithm by considering the specifications of a cell-based design. The objectives achieved in this research are proven reliable with path extraction results consistent even with search depth manipulation. Performance differs an average of 12.6 % (iteration count) with keeping maximum allowable depth of search constant. Paths of net sequences were consistent throughout the verification of the path extractor program. This development and study of the path extract method carries significance in areas of EDA and debug diagnosis work.

CHAPTER 1

INTRODUCTION

1.1 Background

In Electronic Design Automation (EDA) which refers to electronic computer-aided designs for electronic systems commonly used in the semiconductor design flow by design engineers (Sifakis, 2015). These software tools are used to design and validate full semiconductor chips within the design flow cycle from Register Transfer Level (RTL) to layout in pre-silicon and debug & diagnosis in post-silicon. These EDA tools are useful since modern fabrication technology enables semiconductor chips to contain billions of transistors (Courtland, 2017) as technology node reduces and number of transistors (components) in a semiconductor chips doubles every 18 months according to Moore's Law (Bai, 2016). Since a full semiconductor design such as a microchip may contain billion of transistors, cell-based design methodology are widely used in EDA tools to provide abstract representation of components in terms of physical layout or diagrams (Xiu, 2007).

The use of cell-based design methodology enables design engineers to work on smaller partitions of various cells at different level of abstraction. As an example, an engineer may choose to focus on logical function of a particular circuit whereas a layout engineer is more interested in its physical layout instead. Having a cell-based design model of both physical and logical abstraction helps engineers to work on individual or group of cells based on patterns, design requirements or even its architecture (Xiu, 2007). This methodology also enables engineers to reuse cells known as “instances” in more complex designs. This technique does not require understanding of all the implementation details. An instance could be anything from a single transistor to a resistor or capacitor. In some cases, a cell represents a block of integrated circuit chip design.

Complex designs contains many instances and hence, the connectivity of these cells increases as the design becomes more dense, stacked within multiple hierarchies (Srivastava, Winter 2001) as seen in Figure 1.1.

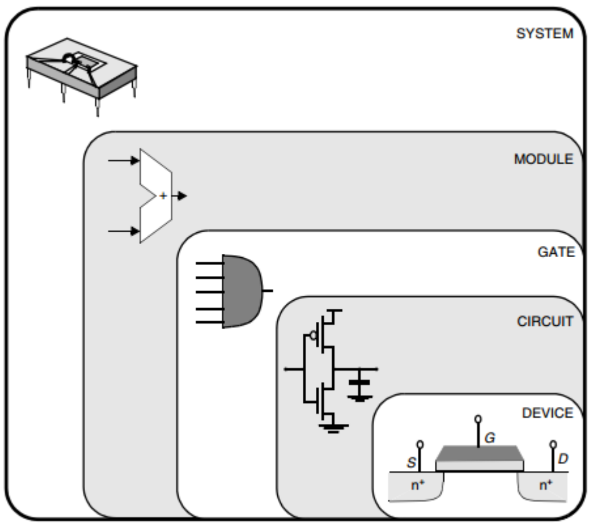


Figure 1.1: Design abstraction levels in digital circuits (Chandrakasan, Nikolić, & Rabaey, 1995).

In a cell based design (CBD), cells that are within a another cell are known as a child and the parent is known as a master cell. Both these cells are interconnected and can be referenced through its input and output signals. Moreover, a parent cell may contain multiple child cells that may have other instances nested in them hence, tracing the connectivity becomes a challenge as circuit designs becomes increasingly complex with increased depth. These depths are known as hierarchical design defined in the design netlist which is a description of the circuit connectivity.

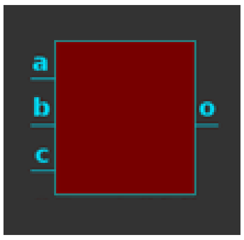


Figure 1.2: Primitive cell with 3 input and 1 output terminals.

The description of the connectivity includes input and output terminals to a particular cell also known as pin connections shown in Figure 1.2. Each cell can be expanded to unfold instances or components that are nested within. Shown in Figure 1.3 are the cell's interconnects unfolded from the cell in Figure 1.2.

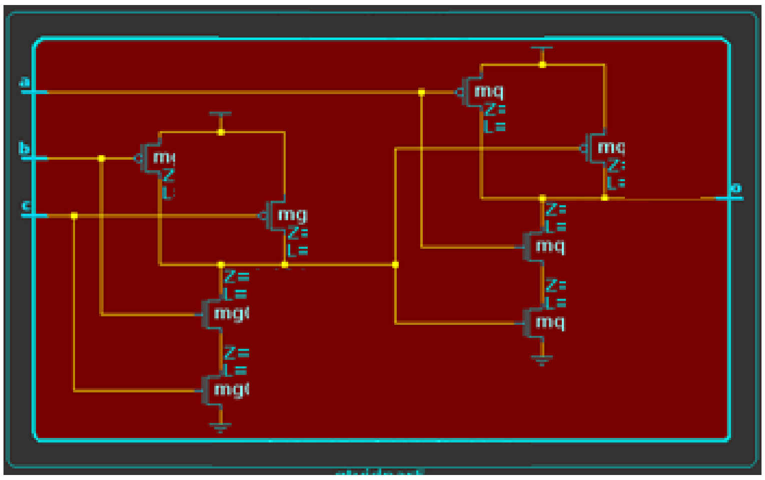


Figure 1.3: Components inside a Primitive cell (Flat view).

Standard cells are direct representation of simplest logical functions of a combinational group of transistors which its structure cannot be further expanded, identified as a "flat view". For example, a Standard cell represents Boolean functions of inverters, AND, OR, XOR, XNOR or storage functions such as flops and latches. This research proposes a solution to extract connectivity information tracing from a single driver pin to a receiver pin throughout the schematic design based on logical functions.

1.2 Problem Statements

In accordance to Moore’s law (Moore, 1965), transistor count is predicted to double every 2 years. Transistor count in semiconductor chips has exponentially increased up to 2.6 billion within a span of 30 years since 1971 depicted in Figure 1.4.

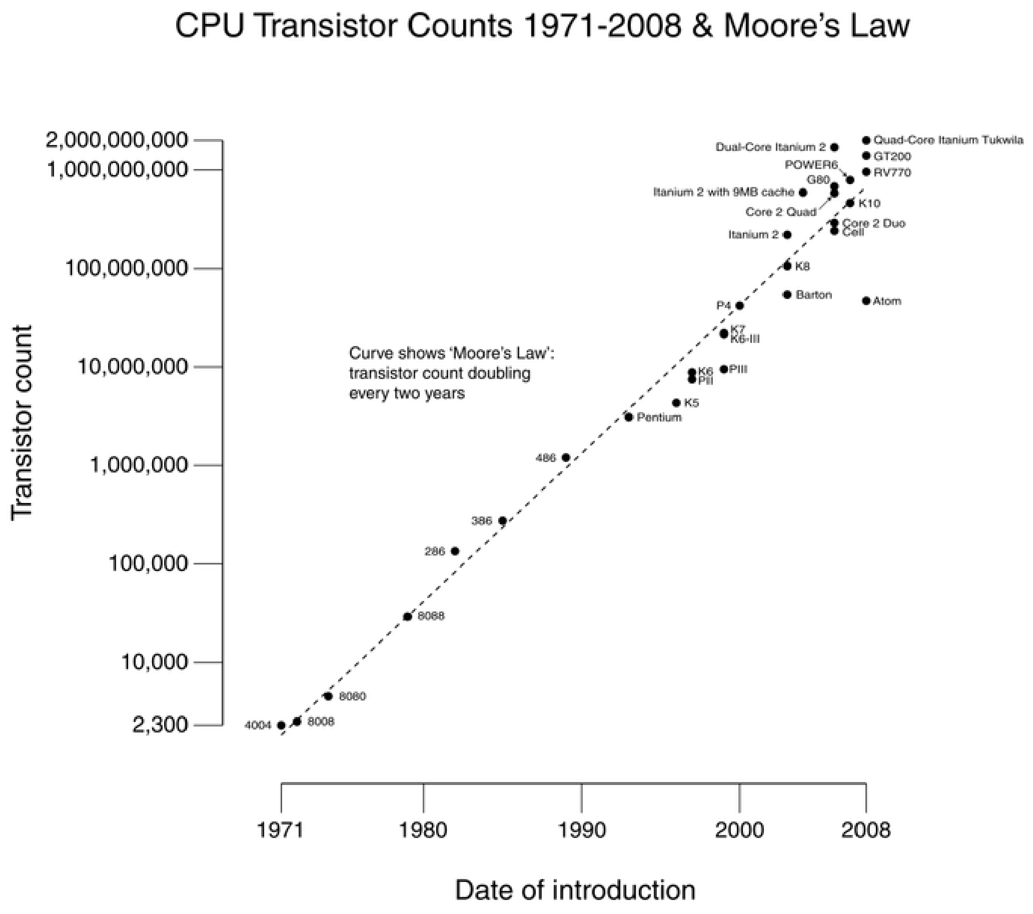


Figure 1.4: Microprocessor Transistor Counts from 1971 to 2011 (Klopfer, 2012).

Increased complexity in full semiconductor chips means that more transistors of different functions and architecture can be grouped and synthesized into different type of cells. This causes the cell counts within a design to increase modelled and used in EDA tools (Dunlop, Evans, & Rigge, 1997).

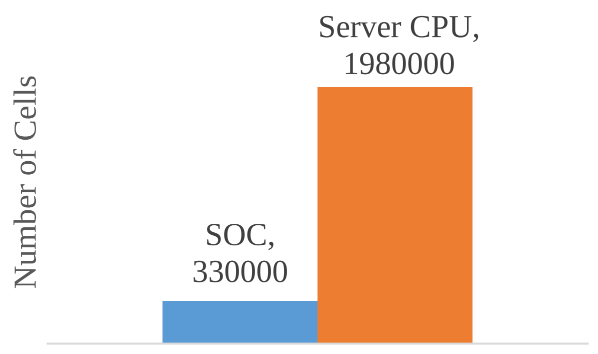


Figure 1.5: Average cell count in 10 nm semiconductor chips.

Figure 1.5 shows the average cell count from standard to netlist conversion of 10 nm design to OpenAccess (OA) model experience as of year 2017. Comparison was made in an SOC design as compared to server CPUs that are modeled in OA to form cell based IC design through spice netlist conversion. The OpenAccess Coalition (Blanchard et al., 2002) delivers and provide a broad industry standard Application Programming Interface (API) and Information Model (IM) essential for modern IC design systems (Blanchard, 2017). Computational search challenges arises in tracing fault affected components in a cell-based design schematic as more and more cells are fitted into a design and cell instances being embedded in multiple hierarchies.

Due to the hierarchical structure of Cell-Based Design (CBD), tracing signal paths from one terminal to the other across multiple depths of hierarchy manually is time consuming. This is because engineers are required to identify, analyze and trace each individual connection otherwise known as nets, traversing through cell and instance pins to carry out fault isolation work (Liu, 2012). Determining components between a start and end signal are time consuming for cells that has bi-directional pin where it can be treated as fan-in or fan-out of a particular cell as seen in Figure 1.6.

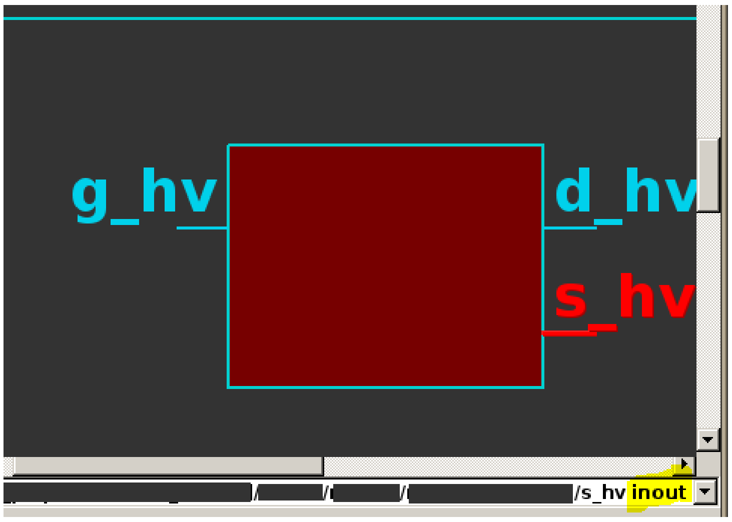


Figure 1.6: Cell with all 3 inout pins.

The fan-in and fan-out sequence of a particular cell has to be taken into consideration when traversing through a hierarchy (Tala, 2006). As an example, a cell with 5 terminals such as a register may have 3 strict input pins which are known as fan-in of a cell, whereas the remaining in-out pins that drives other cells are known as fan-out. The number of pins are multiplied for registers with higher resolutions such as a 100-bit register. Manual efforts to trace signals through such components would be time consuming as not all terminals would have paths leading to the receiver pin.

Apart from that, cell based designs are loaded on demand as user expands and view the interconnects of a particular cell to work on. For a complex full chip design, traversing through the entire design to identify a particular signal path from a driver to receiver pin requires complete load-up of the object model which greatly affects performance. As an example, the complete load-up of a cell-based design that only contains 34 cells at current depth requires a duration of 2 minutes and 18 seconds based on work experience in Intel. Therefore an automated path extractor program to traverse through all available cells with returned valid paths can save users effort and time.

1.3 Objectives

The objectives of this research are:

I. To develop a cell-based design circuit signal path extractor driven by a single driver and receiver pin.

II. To analyse a suitable search algorithm method used to identify sequence of nets involved during path extraction.

1.4 Research Scope

The critical area involved in this research are the search algorithm and parameters used in identifying accurate paths from a single driver pin to a receiver pin. The search algorithm involved in development will cover hierarchical search and traversal method in respect to performance measured in unit time. The structure of the cell based design circuit represented by OpenAccess (OA) model libraries are studied and covers the methodology that enables the application of the search algorithm.

Conditions such as the depth of search which manipulates the path extraction performance are analyzed and discussed. The search algorithm is heavily dependent on the depth of search as more cells are involved as the search area expands. Similar to a linked tree data structure involving graphs, search algorithms used in these areas are experimented, analyzed and discussed. In this research, each cell and instance are treated as object nodes in a search pattern, the branches to these nodes represents the connectivity information in sequence. Since digital circuit designs used are represented as cell-based designs, analogue or electrical characteristics of the research are not covered as it does not affect the search algorithm for Cell Based Design (CBD) path extraction.

1.5 Research Contributions

This research contributes to improved turn-around-time in fault isolation work for Failure Analysis (FA) Engineers in layout to schematic mapping. The path extractor can be used to extract the possible paths with given start and end signal received from post-silicon debug tool that formed a suspected boundary consisting of multiple logics from one end to the other.

Apart from that, the path extractor can be used to acquire schematic signals to create a mapper file containing all connectivity information. As an example, a 100 bits register from bit 99 until bit 0 contains inverters and buffers in between each bit, path extractor can be used to retrieve all connectivity information by only providing the input and output signals. Therefore, these information can used in post-silicon debug tools where FA engineers carry out physical to layout mapping.

1.6 Thesis Outline

This thesis is organized into five main chapters whereby in chapter 2, the topic of path extraction in Very Large Scale Integration (VLSI) designs are being studied and discussed especially in areas where solutions can be applied to electronic design automation tools. Besides that, search algorithm such as the depth-first search (DFS) and breadth-first search (BFS) are discussed to review strength and weakness of individual methods. Applications of VLSI cell based methodology and application of these algorithm are also being discussed in this chapter.

In chapter 3, the methodology of the research is charted. The development stages of the research from search algorithm to manipulative parameters and implementation of conditions due to cell structures are discussed in detail. The search algorithm presents the method that is used in traversing through the hierarchical tree structure of the cell based design. For manipulative parameters, depth of search is discussed as to how the search algorithm depends on. Since not all terminals are bidirectional pins, therefore how the path extraction method identifies and respond to these cell structure and signal connectivity are thoroughly discussed here.

In chapter 4, careful analysis of collected results from experiments are discussed and presented in this section. These experimental results involves topics of performance in unit time measurement, specifications of the search algorithm, connectivity information and calculative changes that was retrieved by manipulating the depth of search parameter. This chapter also focuses on the algorithm which is used in distinctive designs with different cell pin configuration where similarities and differences of results obtained are discussed and analyzed.

Finally in chapter 5 concludes the content of the study completed in this research. Final remarks, overall summary and future improvements that could be enhanced were proposed and justified in this chapter. Opinions related to future works interpreted from result collection are emphasized in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The development of the cell based design path extractor involves a multitude of areas which are related to Electronic Design Automation (EDA). Hence, this literature review section provides an in-depth discussion of present methodology and result analysis in areas of cell based design methodology, type of data extractions, search and circuit extraction algorithms, motivation and works that are involved within the topic. Besides that, the current problems and gaps of these areas are discussed and reviewed so to provide an in-depth knowledge of the current technological solutions available. Because this research contributes to the advancement of EDA, many references are taken from journals and publications originating from this categories, with scholars contributing to areas of advancing design integration technologies for Computer Aided Design (CAD) designers and software developers in areas of semiconductor Integrated Circuit (IC) design.

The structure of this chapter is categorized into 6 sections in sequential order which discusses in detail of current and past researches. A summary of critical analysis was tabulated at the end of the chapter to justify disadvantages, strengths and gaps of the path extraction topic in Cell Based Design (CBD).

2.2 Cell Based Hierarchical Design Methodology

To be able to develop a circuit path extraction method, understanding the hierarchical structure and how this methodology is being used in current and previous research and work is important. As an example, OpenAccess (OA) provides industry standard database for cell based design methodology in the form of libraries (Hahn, 2012). According to Inoue, OA has the capacity and performance standards required for today's largest semiconductor design. The reason being, OA provides standard C++ Application Programming Interface (API) that facilitates simple and swift access to a unified data model for physical and logical designs alike (Inoue, 2006). Popular methodologies and details are discussed in the following subsection.

2.2.1 Cell-Based Design Structure and Hierarchical Netlist Extraction

To be able to acquire the hierarchical structure of a cell based design, information of these designs are extracted from the gate-level netlist. The gate-level netlist acquired from Logic Synthesis within the Very Large Scale Integration (VLSI) Integrated Circuit (IC) (Thakur, 2016) design flow helps validation engineers to understand the connectivity of the silicon design that is to be fabricated.

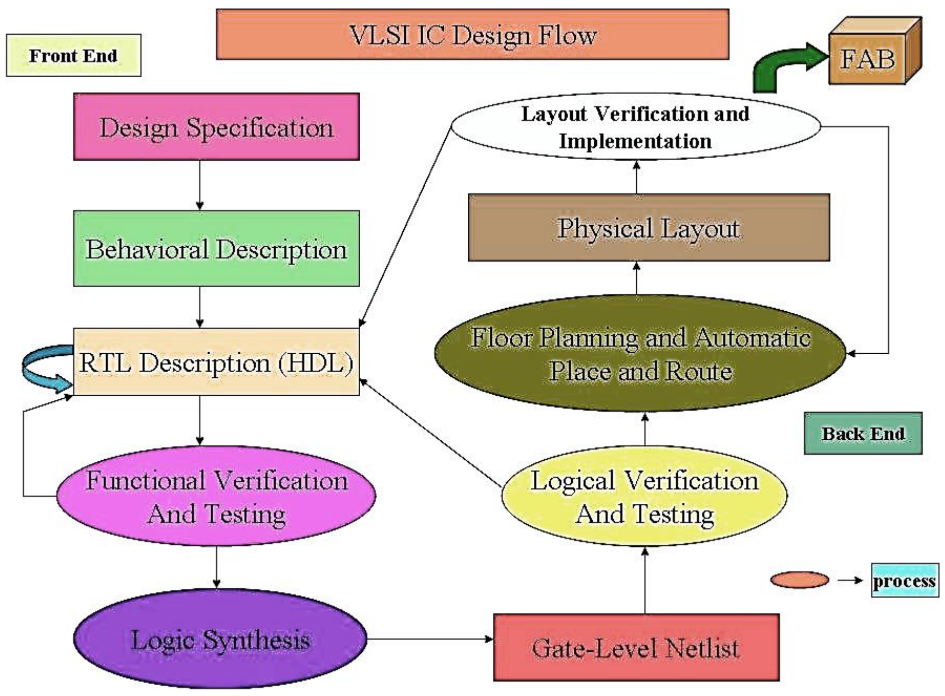


Figure 2.1: VLSI Custom Design Flow (Thakur, 2016)

Figure 2.1 shows the netlist information could be used for floor planning, Auto Place and Route (APR) and visualization of schematics and physical designs. Connectivity information is generated into a netlist file which contains information of each particular cell in a design including each instances. According to L.G. Chen et al. from the connectivity information acquired, cells and its interconnects is known (Chen, Lee, & Wang, 1987). The cells identified from the netlist file contains the following properties shown in Table 2.1:

Table 2.1: Properties of cell identified from netlist.

Properties that could be identified from netlist to model extraction.
Each cell are distinctively different from its name.
All connections to a cell should be connected through the input or output pins.
All input and output pins should be named with pin types which represents the pin as input or output pin type.

For particular cells, different cell pins can result in different number of fan-in or fan-outs of a cell. However, these pin can also be known as bidirectional pins representing both input and output pins alike depending on signal direction.

2.2.2 Effect of Different Cell-Based Methodologies

In this paper which was discussed by B. Kick et al. proposed a standard cell based methodology for high-performance support chips (Kick, Baur, Koehl, Ludwig, & Pflueger, 1997). The paper describes the methodology used in the design of a set of CMOS support chips used in IBM server chips. The logical design aspect is based on functional units of each cell. The majority of the logics implemented by using standard cell components placed and routed flat are determined using timing-driven techniques.

Custom library components are used whenever required for optimizing performance. Using this technique, comparable cell density to those of present-day custom designs with reasonable turnaround times were discussed thoroughly in this paper.

2.2.3 Data Model and Standard Cell Libraries

OpenAccess (OA) database is commonly supported by Electronic Design Automation (EDA) applications due to the standardization it provides for interoperable representation in semiconductor IC design as mentioned by Mark Hahn from Cadence Design Systems (Hahn, 2012). In the paper published by Shishpal S. Rawat from Intel and Sumit DasGupta from Si2 organisation (Rawat & DasGupta, 2012), discusses details of the OA API standards and reference database such as the embedded module hierarchy and constraint classes that aids in silicon design. The embedded module hierarchy represents the cell-based design methodology that is widely used in representing embedded components in a chip design of multiple hierarchies or modules as shown in Figure 2.2. In simple terms, OA is a complete, next-generation database for representing IC designs with its vast functionality as described by Joseph T. Santos (Santos, 2004). OA provides the data model that captures both implementation and intent of an Integrated Circuit (IC) design specification as it comprises of three integrated databases shown in Figure 2.2.

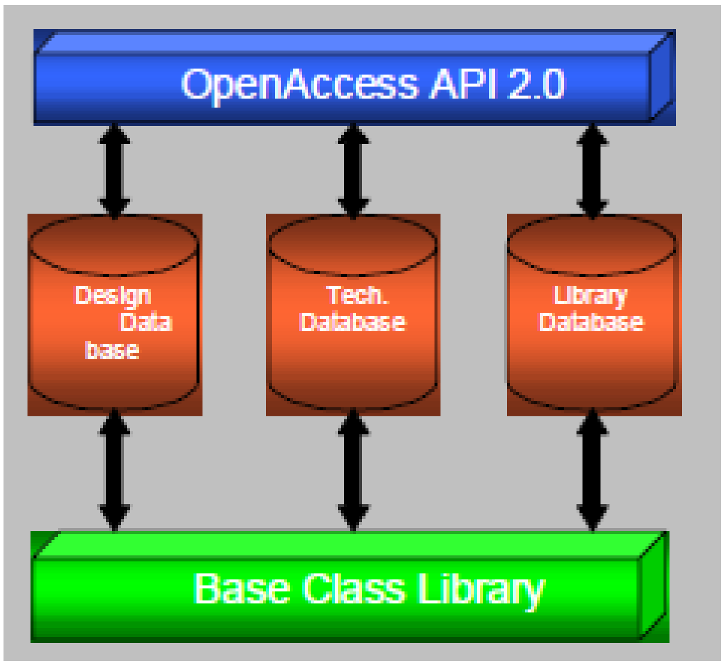


Figure 2.2: OpenAccess API 2.0 database structure.

The structure of these integrated database features high-performance access to design data while consuming minimal memory ideal for use in EDA applications. Cells contain definition of other primitive cells or standard cells in Gate-level functions at the lowest hierarchy. Besides that, OpenAccess data model also covers design data that includes structural netlist connectivity, placement and routing, and hierarchical information. Similar representation can also be seen in other EDA design formats such as Verilog, Design Exchange Format (DEF) introduced by Cadence and the popular Milkyway database available from Synopsys (Cadence Design Systems Inc., 2009).

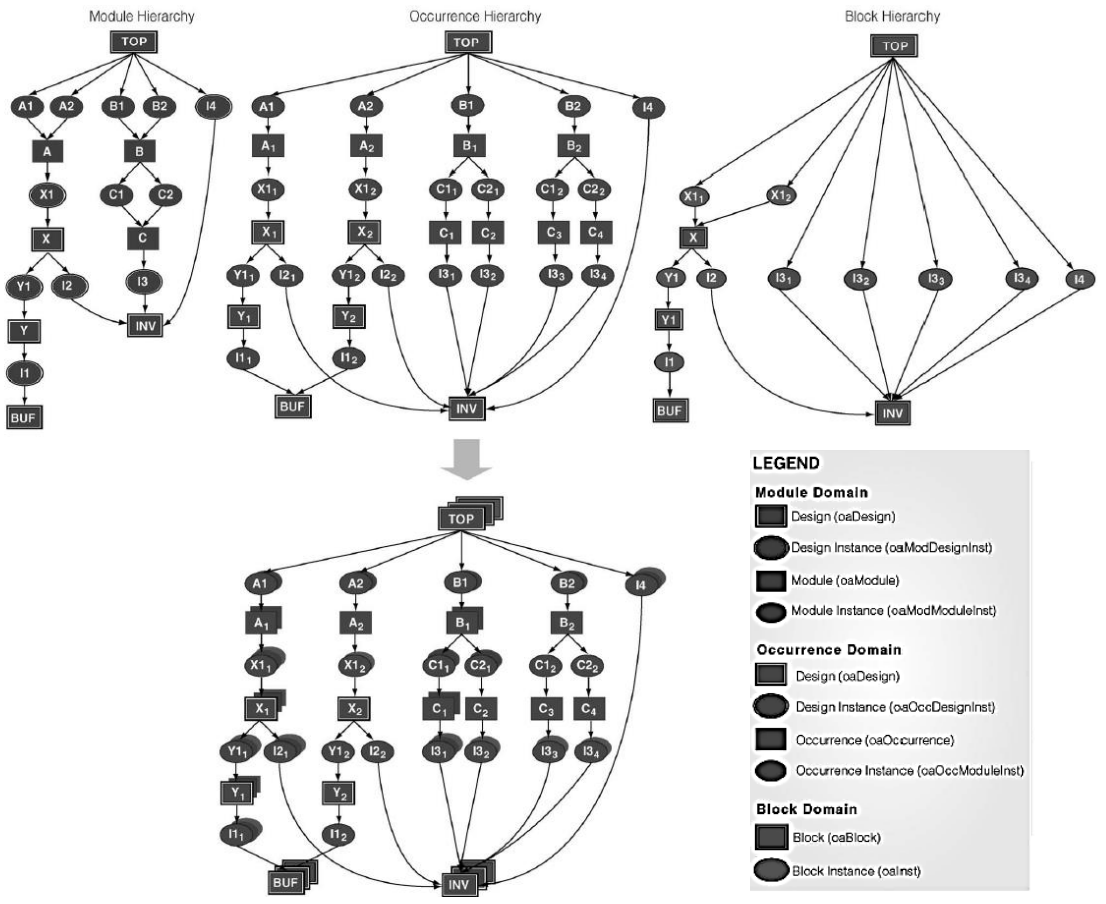


Figure 2.3: Embedded module hierarchy (Hahn, 2012).

In a custom design flow, the database helps in providing a library model of the design representations as shown in Figure 2.3. The involvement of these libraries in the path extractor program from sequential design flow as described by Lavago et al. is after the routing stage referred to Figure 2.4.

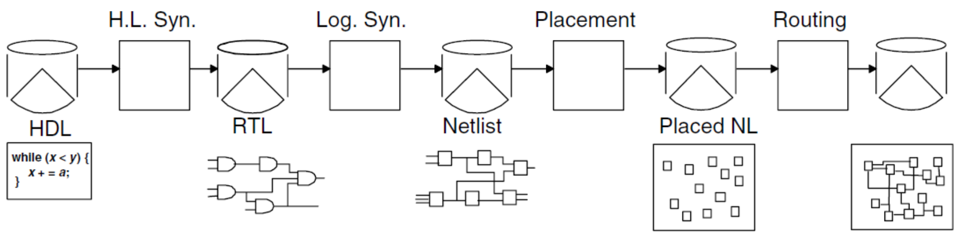


Figure 2.4: Sequential design flow (Lavagno, Martin, & Scheffer, 2006).

The hierarchical representation of connectivity in a cell-based design is defined in the netlist. Proven and described in the textbook (Lavagno, Martin, & Scheffer, 2006), EDA for IC Implementation Circuit Design and Process Technology.

2.3 Data Extraction Method in Very Large Scale Integration Integrated Circuit Designs

This section discusses about recent and past research works in the areas of information extraction of VLSI circuit designs. These extraction methods and analytical results were studied in detail to understand more on the parameters and information retrieved from interconnect extractions. This literature review section aims to provide in-depth understanding in areas of a VLSI data extractor that interacts with a design database and the applications that motivates the development of such a feature.

2.3.1 Parametric Path Extraction Method

A 3D Global interconnect parameter extractor known as (GIPER) was developed by S. Y. Oh et al. to provide a practical EDA extraction tool for a full-chip global critical path analysis that extracts resistance and capacitance (R,C) parameters (Oh, et al., 1996). A typical global interconnect parameter extraction of a net takes several minutes on a HP 9000/755 workstation within 5 % accuracy comparable to a full 3D numerical simulation as discussed in the publication. As mentioned by S. Y. Oh et al, five assumptions are used to simplify and make the 3D interconnect model library generation practical with occupying reasonable CPU time and accuracy. However, three assumptions are taken into review due to its involvement in hierarchical circuit design in a full custom chip. Table 2.2 presents the assumptions carried out in GIPER for interconnect model library generation.

Table 2.2 Assumptions of GIPER in interconnect library generation.

Assumptions to simple 3D interconnect model library generation
It is enough to include only cross-overs in 3D interconnect model library for global interconnect parameter extraction.
The cross-section of hierarchical interconnect is determined to be rectangular and planar.
Adjacent metal layers routing directions are orthogonal to each other.

Apart from that, the hierarchical circuit of an IC netlist design should be taken into consideration also. In a paper regards to hierarchical IC layout circuit extraction as discussed in a paper by Ahsan Bootehsaz et al. described a set of heuristic methods for a hierarchical circuit extractor (Bootehsaz & Cottrell, 1986). The key points to note in this research is that the strength of an extraction algorithms depends on the capability of a software in exploiting the natural hierarchical structure of an IC layout design. Besides that, the handling in overlapping cells also known as instances without creating partial devices are important treated as one of the parameters in optimizing the extraction algorithm.

In this paper, Ahsan Bootehsaz et al. mentioned that all technology dependent information are kept in a user accessible file external to the program which is also used to configure or manipulate the extent of the parameter extraction shown in Figure 2.5.

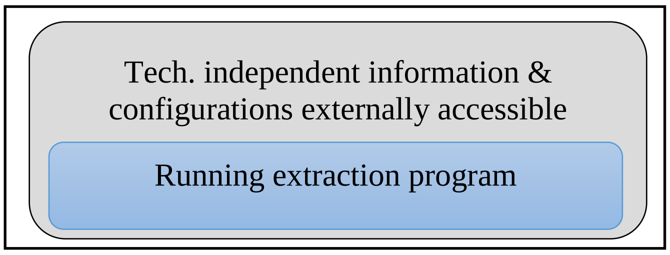
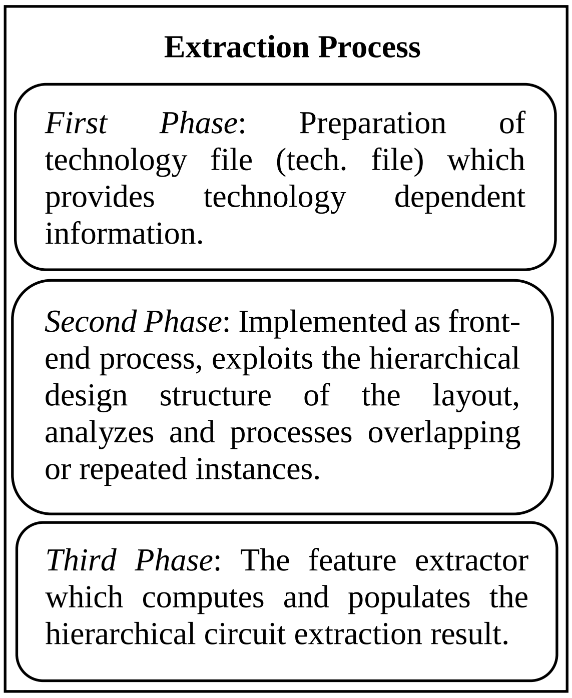

Tech. independent information &
configurations externally accessible
Running extraction program

Figure 2.5: Extraction methodology used by Ahsan Bootehsaz et al. in hierarchical circuit extraction.

The circuit extraction proposed is performed in a bottom-up manner which produces a netlist description similar with the hierarchical structure of the layout. The extraction process as proposed in this paper is categorized into three categories shown in Figure 2.6.


Extraction Process

First Phase: Preparation of technology file (tech. file) which provides technology dependent information.

Second Phase: Implemented as front-end process, exploits the hierarchical design structure of the layout, analyzes and processes overlapping or repeated instances.

Third Phase: The feature extractor which computes and populates the hierarchical circuit extraction result.

Figure 2.6: Three categories contained within the extraction process (Bootehsaz & Cottrell, 1986).

2.4 Circuit Analysis Algorithm Used in Electronic Design Automation

This section discusses about the previous works relating to circuit analysis in EDA tools. Previous researches involving search algorithms, hierarchical connectivity, cell placement as well as existing design traversal method are thoroughly discussed in this section.

2.4.1 Depth-First (DFS) and Breadth-First Search (BFS) Algorithm

In the research that studies the depth-first search algorithm involved in a communication network by S.A.M. Makki et al. discusses about embedding information "forward and return" messages which helps to explore the linked tree more efficiently (Makki & Havas, 1994). The key improvement in their development of the depth-first search tree algorithm is to reduce the number of "return" messages by using dynamic backtracking. This was made possible by including message information that communicates to the receiver the potential return message address which may not necessarily be its parent node.

In the algorithm, a node is defined to be a "split point" if it has two or more unvisited neighbors or it is a root node when visited. This way, nodes that are not split points could be bypassed by return messages. This paper discusses heavily in providing the program more information during traversal of the DFS tree which helps to improve performance and extract more information of the tree structure at runtime. For analysis purposes, the sample graph in Figure 2.7 was studied.

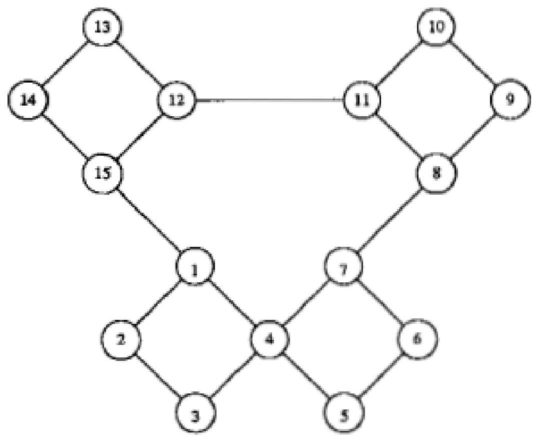


Figure 2.7: Sample graph (Makki & Havas, 1994).

Assume that node 1 in the graph is the root node and total number of nodes involve depends on the route chosen by the program. The routing as described by S.A.M. Makki et al. is in order in which the nodes receive the forward message equals to 14. For example, if the program chooses the route (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) or (1, 4, 7, 8, 11, 12, 15, 14, 13, 10, 9, 6, 5, 3, 2), the number of return messages known as nodes is 6. However if the route taken is (1, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2), the number of return messages is 8. The worst-case discussed in this paper of 14 return messages arises from the route (1, 15, 12, 13, 14, 11, 8, 9, 10, 7, 4, 5, 6, 3, 2).

Apart from that, in a research conducted by Xinguo D. et al. involving the combination of breadth-first search (BFS) with depth-first search (DFS) algorithm for shortest VLSI routing was studied (Deng, et al. 2010). This study discusses about the hybrid method that utilizes capability of BFS algorithm which requires less time to find nodes closer to root, and DFS algorithm to find the shortest path between two nodes in wire routing. In this study, BFS algorithm was used to compute the shortest distance between every node and the start node. This is proven to consume high bandwidth as it utilises a huge amount of computer memory as mentioned in this paper. The depth-first search algorithm was used to traverse through all the shortest paths received from the BFS algorithm from the end to the start position.

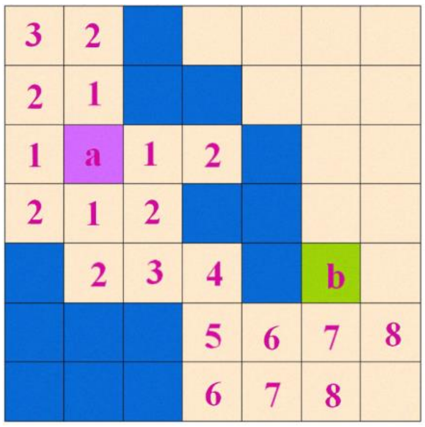


Figure 2.8: Example of Breadth-First Search structure (Deng, et al. 2010).

Illustrated by Xinguo D. et. al. in Figure 2.8, to find the shortest path between two point, BFS algorithm begins search at root node "a" and labels its reachable neighbor such at point "a" is 1. Followed by the labelling next reachable square with increment of 1. This labelling process is continued until the algorithm reaches point "b" or until it has no nodes to traverse any further. Hence, the final number reaching point "b" is 8 indicating it's distance. For the DFS algorithm, the tracing begins at the end point which is the "b" node and moves to any neighbor with 1 value less than the previous b's label until it reaches node "a".

2.4.2 Hierarchical Path Planning Algorithm

In a recent research proposed by Tosmate C. et al. the use of DFS worst-fit search based for data center networks resulting in multipath routing was introduced (Cheocherngngarn, et. al. 2012). In this paper, the multipath routing problems in data center networks (DCNs) were studied. The depth-first search algorithm was proposed as opposed to an integer linear program as it is not suitable for fast runtime routing calculations. The main reason behind the use of the DFS algorithm is to find a sequence of worst-fit links to connect source and destination of a flow. Since DCNs topologies are defined in hierarchies, the depth-first search algorithm can quickly traverse between the hierarchical layers to find a path.

Understanding the hierarchical path planning method as discussed by Yong-Jie M. et al. helps to understand the aspect of path planning in an ant colony algorithm and genetic algorithm (Ma & Hou, 2010). The path planning method referred to Table 2.3 algorithm type was used to tackle the path planning feature aimed at urban road traffic by analyzing traffic restrictions.

Table 2.3: Algorithm used urban traffic path-planning (Ma & Hou, 2010).

Algorithm type	**Description**
Ant Colony	Employed at bottom to calculate selected subnets for local optimization in parallel.
Genetic	Genetic algorithm is used at the top for global optimization.

This research helps to identify the best solution to a converged global optimization method in which the advantages of an Ant colony algorithm is stronger in local search capability and faster convergence speed whereas Generic algorithm has stronger global search capability. The disadvantages of these two algorithm are that an Ant colony algorithm is weaker at global search capability and Genetic algorithm is slow at convergence speed (Cristina Martinez, 2008). The algorithm introduces different methods of algorithm usage at each hierarchical layer running these programs in parallel to improve search efficiency. Apart from that hybrid genetic algorithm was also introduced by Xiongfeng C. et. al. in solving VLSI standard cell placement problem which the paper presents an adaptive hybrid genetic algorithm in standard cell VLSI placement which belong to NP-hard combinatorial optimization problem (Chen, et. al. 2016).

2.4.3 Interconnect Traversal Algorithm

The research proposed by R. Sharathkumar et al. discusses about a practical algorithm for connectivity extraction Very Large Scale Integration (VLSI) layouts (Sharathkumar, et al. 2006). As Moore's law continues to strive, the number of transistors increases hence, connectivity information becomes increasingly large. The problem discussed in this paper relates to processing large VLSI layouts being too large for computer's main memory, communication between the internal memory which is faster and the slower external memory is often the performance bottleneck. The traversal algorithm and data structures designed under the assumption of the single level memory may not be meaningful.

R. Sharathkumar et al. designed and implemented a practical external memory algorithm which extracts connectivity information from a VLSI layout design. The first solution to this implementation is using a "UNION-FIND" (Lu, et al. 2016) data structure and second, the author discusses about targeting the problem with a "square-root" rule counter problem present in VLSI layout design. In the result of this implementation, the paper list that the standard main memory algorithm uses the operating system's (OS) memory management routine to access the disk but fails to perform well as the program does not have control over how the OS perform disk accesses.

2.5 Physical to Layout Debug Method and Analysis

As this dissertation research aids in silicon debug method, physical-to-layout mapping plays an important role in fault isolation debug in maintaining silicon design quality. Figure 2.9 shows an example of a "physical" 800 nm process E-fuse bits on a polysilicon layer.

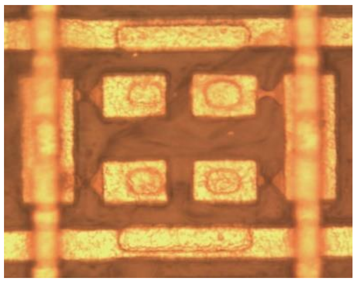


Figure 2.9: Polysilicon layer of a 800nm process IC (McMaster, 2014).

Silicon architecture mapping (Physical to Layout) in post-silicon validation (Mishra, et al. 2017) discusses about pin-pointing the location which issue was found probing between two nodes on a silicon tracing to layout schematic pins in an Electronic Design Automation (EDA) tool. As designs becomes more complex, understanding these faults and recent debug methods becomes increasingly challenging (Shafique, et al. 2014) and important in debug diagnosis EDA tools.

2.5.1 Fault Isolation and Failure Analysis

In general, emission microscopy (EMMI) is able to detect photon emission event induced by crystal defect but is not always reliable due to thick dense metallization and high doped substrate as discussed by N.S. Lee et al. (Lee & Yong, 2016). In publication, the author proposed fault isolation of die level crystal defect failure mechanism through "OBIRCH" analysis and micro-probing. Electrical failure